UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JULIANE DA ROCHA ALVES

Eye and Skin Color Prediction for Brazilian Population using Single Nucleotide Polymorphisms

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Engineering

Advisor: Prof. Dr. Marcio Dorn

Porto Alegre
October 2022

“To the stars who listen and the dreams that are answered.”

— SARAH J. MAAS, A COURT OF MIST AND FURY

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my family, especially my mother, Cleonice Feix da Rocha, for always believing I was capable of getting here. Thank you for supporting all my decisions and being my emotional support. We had to deal with a lot of difficulties in the past few years, and there were times when it seemed this day would never come, but it did, and I would never be able to get here without you.

I am also grateful to my friends, Tainara Silva, Pedro Braga, Jonas Bohrer, Daniel Einloft, Brian Seggebruch, and Gabriel Lando, for all the positive words and thoughts. Thank you for always asking me if I needed any help or if there was something you could do to help me. Your support during this process was essential to me. Also, thank you for reminding me that I was more than capable to go through all this.

I would like to acknowledge all my friends at HP and Poatek. Thank you for celebrating this achievement with me and hearing me talking about how tired I was almost every day. You were very supportive and understanding when I had to skip some happy hours or be absent to focus on this work.

Thanks should also go to Prof. Dr. Marcio Dorn for the guidance and for accepting to be my advisor. I appreciate all the teachings and support. I also want to thank Dr. Mariel Barbachan for helping me with the theoretical basis of Generative Adversarial Networks and clarifying all my doubts about this subject.

Lastly, I would like to acknowledge everybody that somehow made a contribution to this journey.

ABSTRACT

When a crime is under investigation, especially when too many questions are unanswered, it is necessary to reduce the number of suspects to be able to solve the investigation. To reduce the number of suspects, any detail found at the crime scene is important, such as a strand of hair, DNA, or even a fingerprint. When the DNA found does not have the complete information to be able to determine the identity of the suspect, some information can still be extracted from it, like the information of eye color or skin color. This work presents the application of Machine Learning algorithms, such as Random Forest, and Support Vector Machine to determine the pigmentation of the eye and skin using Single Nucleotide Polymorphisms (SNPs) from a DNA sample for forensics use. The following chapters will present the necessary studies to investigate a solution for the proposed problem. Genetic and machine learning theoretical basis are presented, as well as related works, experiments, and results. Each dataset contains sixty-six SNPs and three classes: Blue, Intermediate, and Dark Brown are the classes related to eye color, and White, Intermediate, and Brown are the classes related to skin color. 144 experiments were executed (72 for eye and 72 for skin classification), combining different approaches of feature selection, class balanced, and classifiers to define the best solution. The data used for this study were collected from the Southern Brazilian population. The final results showed that 4 SNPs can be used to predict Blue and Dark Brown classes. For skin classification, 56 SNPs can be used when SMOTE is applied to balance the classes, but a further investigation is necessary to understand if the SMOTE is impacting the selection of the SNPs. Using 36 SNPs without class balance also achieved a close result. All the experiments had a bad performance for the Intermediate classes. For future work, a better investigation of intermediate colors is necessary.

Keywords: Single Nucleotide Polymorphisms. Forensic. Eye color. Skin color.

Previsão da cor dos olhos e da pele para a população brasileiros utilizando Polimorfismos de Nucleotídeo Único

RESUMO

Quando um crime está sob investigação, especialmente quando muitas perguntas não são respondidas, é necessário reduzir o número de suspeitos para poder resolver a investigação. Para reduzir o número de suspeitos, qualquer detalhe encontrado na cena do crime é importante, como um fio de cabelo, DNA ou até uma impressão digital. Quando o DNA encontrado não possui as informações completas para poder determinar a identidade do suspeito, algumas informações ainda podem ser extraídas dele, como a informação da cor dos olhos ou da pele. Este trabalho apresenta a aplicação de algoritmos de Aprendizado de Máquina, como Random Forest e Support Vector Machine para determinar a pigmentação do olho e da pele usando Polimorfismos de Nucleotídeo Único (SNPs) a partir de uma amostra de DNA para uso forense. Os capítulos seguintes apresentarão os estudos necessários para investigar uma solução para o problema proposto. São apresentadas as bases teóricas de genéticas e de aprendizado de máquina, bem como trabalhos relacionados, experimentos e resultados. Cada conjunto de dados contém sessenta e seis SNPs e três classes: Azul, Intermediário e Marrom Escuro são as classes relacionadas à cor dos olhos, e Branco, Intermediário e Marrom são as classes relacionadas à cor da pele. Foram executados 144 experimentos (72 para olho e 72 para classificação de pele), combinando diferentes abordagens de seleção de *features*, balanceamento de classe e classificadores para definir a melhor solução. Os dados utilizados para este estudo foram coletados da população do Sul do Brasil. Os resultados finais mostraram que 4 SNPs podem ser utilizados para prever as classes Azul e Marrom Escuro. Para classificação da pele, 56 SNPs podem ser utilizados quando SMOTE é aplicado para equilibrar as classes, mas é necessária uma investigação mais aprofundada para entender se o SMOTE está impactando na seleção dos SNPs. O uso de 36 SNPs sem balanceamento de classe também obteve um resultado próximo. Todos os experimentos tiveram um desempenho ruim para as classes Intermediárias. Para trabalhos futuros, é necessária uma melhor investigação de cores intermediárias.

Palavras-chave: Polimorfismos de Nucleotídeo Único. Forense. Cor do olho. Cor da pele.

LIST OF ABBREVIATIONS AND ACRONYMS

1DCNN	1D Convolutional Neural Network
ADASYN	Adaptive Synthetic Sampling Approach
AHCY	Adenosylhomocysteinase
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
ASD	Autism
ASIP	Agouti Signaling Protein
AUC	Area Under The Curve
AUPRC	Area Under the Precision-Recall Curve
BC	Breast Cancer
BD	Bipolar Disorder
BILSTM	Bidirectional LSTM
BM-ELM	Boundary Movement Extreme Learning Machine
BNC2	Basonuclin 2
CADD	Combined Annotation–Dependent Depletion
CC	Colorectal Cancer
CD	Crohn's Disease
CGAN	Conditional Generative Adversarial Network
CMIM	Conditional Mutual Information Maximization
CNN	Condensed Nearest Neighbor
DDB1	Damage Specific DNA Binding Protein 1
DeepSEA	Deep Learning–based Algorithmic
DNA	Deoxyribose Nucleic Acid
ELM	Extreme Learning Machine

FCBF	Fast Correlation Based Feature Selection
FN	False Negative
FP	False Positive
FPR	False Positive Rate
G-mean	Geometric Mean
GAN	Generative Adversarial Network
GEBV	Genomic Estimated Breeding Values
GI	Gini Impurity
GRU	Gated Recurrent Unit
GWAS	Genome-Wide Association
GWAVA	Genome-Wide Annotation of Variants
HapMap	Haplotype Map of the human genome
HERC2	Hect Domain and RLD 2
HT	Hypertension
IRF4	Interferon Regulatory Factor 4
KDE	Kernel Density Estimation
KITLG	KIT ligand
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
LVQ	Learning Vector Quantization
LYST	Lysosomal trafficking regulator
MC1R	Melanocortin 1 Receptor (alpha melanocyte stimulating hormone receptor)
MFNN	Multilayer Feed-forward Neural Network
MR	Mental Retardation

mRMR	Minimum Redundancy Maximum Relevance
MYEF2	Myelin Expression Factor 2
MYO5A	Myosin VA
NB	Naïve Bayes
PCA	Principal Component Analysis
PIGU	Phosphatidylinositol Glycan Anchor Biosynthesis Class U
RA	Rheumatoid Arthritis
RBF	Radial Basis Function
RF-RFE	Recursive Feature Elimination with Random Forest
RF	Random Forest
RFE	Recursive Feature Elimination
ROS	Random Over-sampling
RUS	Random Under-sampling
SDV	Synthetic Data Vault Project
SLC24A4	Solute Carrier Family 24 Member 4
SLC24A5	Solute Carrier Family 24 Member 5
SLC45S2	Solute Carrier Family 45 Member 2
SMOTE	Synthetic Minority Over-sampling Technique
SMOTEENN	Synthetic Minority Over-sampling Technique and Edited Nearest Neighbours
SNP	Single Nucleotide Polymorphisms
SVM-RFE	Support Vector Machine Recursive Feature Elimination
SVM	Support Vector Machine
TC	Thyroid Cance
TMEM138	Transmembrane Protein 138
TN	True Negative

TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
TTC3	Tetratricopeptide Repeat Domain 3
TUBB3	Tubulin Beta 3 Class III
TYR	Tyrosinase
TYR1	Tyrosinase-Related protein 1
T1D	Type 1 and type 2 diabetes
T2D	Type 2 diabetes
UGT1A6	UDP glucuronosyltransferase 1 family, polypeptide A6
UGT1A7	UDP glucuronosyltransferase 1 family, polypeptide A7
UGT1A8	UDP glucuronosyltransferase 1 family, polypeptide A8
UGT1A9	UDP glucuronosyltransferase 1 family, polypeptide A9
UGT1A10	UDP glucuronosyltransferase 1 family, polypeptide A10

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

When a DNA found at a crime scene has no match in the police database, or when it is necessary to endorse the eyewitness's testimony with more pieces of evidence, alternatives are necessary to avoid "cold cases", and the information of skin and eye color could be one of the alternatives. There are many studies in the forensic field (HART et al., 2013) (WALSH et al., 2011) (CHAITANYA et al., 2018) trying to find a state-of-the-art solution using Single Nucleotide Polymorphisms (SNPs) to determine pigmentation traits, such as skin and eye color.

Single Nucleotide Polymorphisms (SNPs) are mutations at a single nucleotide position that occurred during evolution and were passed on through heredity, accounting for most of the genetic variation among different individuals (SAEYS; INZA; LARRANAGA, 2007). According to Kwok (2003), the human genome has 10 to 30 million SNPs, but only some of them are related to external traits, making the task of using SNPs to determine pigmentation traits even harder. Given the number of SNPs to analyze, it is important to reduce the dimensionality of the problem to have a better understanding of what SNPs are actually relevant.

IrisPlex is a tool developed by Walsh et al. (2011) to predict blue, intermediate, and brown eye color for forensic use, using only six SNPs. The SNPs were selected based on previous studies in the literature. In another work of Walsh et al. (2017), 36 SNPs were chosen to predict skin color based on the Akaike Information Criterion (AIC). The AIC estimates the quality of statistical models for a given dataset, estimating how much information a model loses. Because there is no consensus on the best method to find the optimal SNPs, each work has different approaches in order to maximize the prediction of pigmentation traits using only relevant SNPs.

Besides finding the optimal SNPs, some studies have raised the fact that imbalanced data can degrade classification tasks. Guan and Zhang (2022) presented a study to predict diabetes using genotype SNP data and phenotype data. The dataset used for the task was highly skewed, having much more healthy samples. After some experiments, the final conclusion was the imbalanced data had a poor performance if compared with the experiments in a balanced dataset.

Most of the related works published use data collected from Europeans, and the main challenge of this work is to find a solution using data collected from the Brazilian population. The focus of this work is to understand which SNPs are relevant for the pro-

posed problem and build a classifier for forensic use. The paper is organized as follows: Chapter 2 will give an introduction to the genetic theoretical basis. Chapter 3 will present some machine learning theoretical bases, such as feature selection, predictive models, class imbalance problems, and Generative Adversarial Networks. Chapter 4 will review the use of SNPs for phenotype prediction, and present the proposed problem. Chapter 5 will present experiments and results. Lastly, Chapter 6 will present the conclusion of the work.

2 GENETIC THEORETICAL BASIS

2.1 Deoxyribose Nucleic Acid

Deoxyribose nucleic acid (DNA) is a biopolymer constructed from four different nucleotides attached to a long backbone structure of deoxyribose sugar molecules bonded to phosphate groups (CRAWFORD, 2017). The order of the nucleotides in the DNA is the blueprint for all enzymes and proteins. The DNA is composed of a pair of sugar-phosphate backbone fused to a set of purine and pyrimidine bases. These strands are joined together through hydrogen bonds between thymine and adenine bases and cytosine and guanine bases (BISHOYI, 2021).

The genome (the complete DNA sequence), can be divided into two different classes, based on known functional properties: coding and non-coding regions. The coding regions are sequence of nucleotides that actually code for protein while non-coding regions do not code for protein (FOWLER et al., 1988) (AHMAD; JUNG; BHUIYAN, 2017).

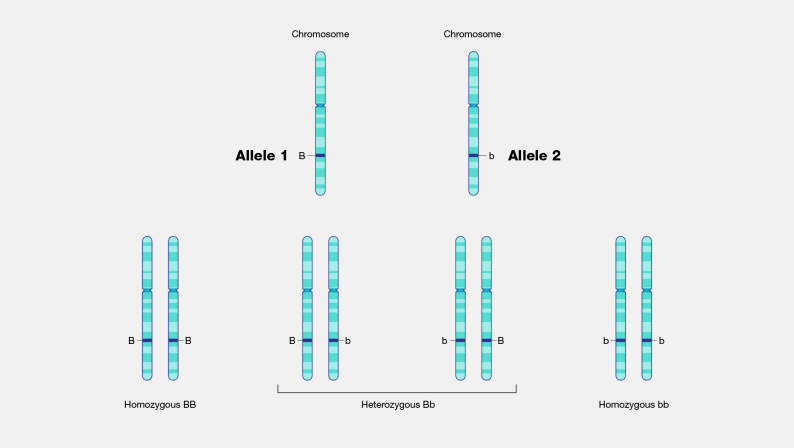
Crawford (2017) defined the chromosome as a composition of proteins and DNA, which carry thousands of genes. In a healthy human cell, there are 23 pairs of chromosomes, and in each pair, one chromosome comes from the father and the other one comes from the mother. In the monogenic inheritance a recessive trait can be observed in individuals with the dominance of an allele. When the allele is non-dominant in idividuals, they do not have the recessive trait (BISHOYI, 2021).

A pair of alleles at a locus forms the genotype, while the resultant physical trait is known as the phenotype. The alleles are replicas of any specific gene, where each allele in the pair was inherited by each parent (BISHOYI, 2021). Figure 2.1 shows the allele at a locus in each chromosome. When the individual inherited different alleles from their parents, they are called heterozygous. When the individual inherited the same alleles from their parents, they are called homozygous.

2.2 Single Nucleotide Polymorphisms

Using the definition presented by Brookes (1999), Single Nucleotide Polymorphisms, or SNPs (pronounced “snips”), are single pair positions in genomic DNA at which different sequence alternatives (alleles) exist in normal individuals in some population(s),

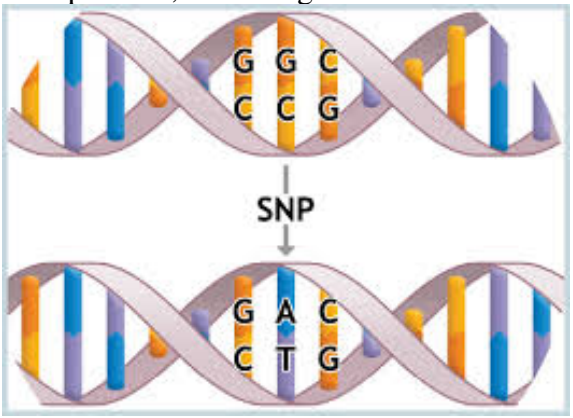
Figure 2.1: The figure shows the allele located at a locus in each chromosome. The heterozygous is determined by different alleles. The homozygous is determined by equal alleles.



Source: National Human Genome Research Institute (https://www.genome.gov/genetics-glossary/Allele)

wherein the least frequent allele has an abundance of 1% or greater. Or, in other words, SNPs are the places in the genome where people differ. For example, if a major part of a population has the nucleotide C (cytosine) in a specific genome position and a minority of the same population has the nucleotide A (adenine) in this same genome position, that indicates that an SNP occurs in that genome position. Most of the SNPs are located in parts of the genome with no critical function. But some of the SNPs determine individual characteristics such as eye color, hair color, and skin color (phenotype). Figure 2.2 shows an example of an SNP.

Figure 2.2: The figure shows two DNA strands. Both DNA strands differ because the one on the top has the alleles G and C in a particular position, and the one below has the alleles A and T in the same position, indicating that a SNP occurs in this specific location.



Source: Society for Mucosal Immunology (https://www.socmucimm.org/news-media/single-nucleotide-polymorphism-snp-allele-frequency-dna-pools/)

From the information of the genes collected, it is possible to obtain SNPs. Each

gene has a set of SNPs. Genes are elements within the genome of a living organism that controls the transmission of a hereditary characteristic by specifying the structure of a particular protein or by controlling the function of other genetic material. A gene is a fundamental unit of heredity and contains instructions necessary for the synthesis of its product which is the RNA (CRAWFORD, 2017).

Many studies are trying to find a state-of-the-art solution to determine the phenotype of an individual using SNPs. In the work presented by Walsh et al. (2011), a tool named IrisPlex was developed to predict the Blue and Brown eye colors for forensic use. Hart et al. (2013) presented a solution for eye color and skin color prediction using 8 SNPs, to improve the 7-Plex system, that utilizes 7 SNPs (rs12913832, rs1545397, rs16891982, rs1426654, rs885479, rs6119471, and rs12203592). Chapter 4 will present a review about the human phenotype prediction using SNPs.

2.3 Chapter Conclusion

The chapter presented the genetic theoretical basis, reviewing DNA and SNPs concepts. The DNA is composed of a pair of sugar-phosphate backbones fused to a set of purine and pyrimidine bases. (BISHOYI, 2021). SNPs, are single pair positions in genomic DNA at which different sequence alternatives (alleles) exist in normal individuals in some population(s) (BROOKES, 1999).

The next chapter will present the machine learning theoretical basis.

3 MACHINE LEARNING THEORETICAL BASIS

Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment (NAQA; MURPHY, 2015). It is a research field at the intersection of statistics, artificial intelligence, and computer science (MüLLER; GUIDO, 2017). Some real-world applications that utilize machine learning are facial recognition, spam detector in emails, and product recommendations (FACELI et al., 2011). Techniques based on machine learning have been applied in different fields, such as finance, computational biology, and medical applications (NAQA; MURPHY, 2015).

Machine learning is about extracting knowledge from data (MüLLER; GUIDO, 2017). The data is usually tabular, a matrix with attribute-value, where each row represents an object (the instance), and each column represents the attribute (features). The attributes can be split into predictive attributes, where the values describe the feature of the object, and target attributes, where the value labels the object (FACELI et al., 2011).

The following sections will give an introduction to some machine learning theoretical bases, such as predictive models, feature selection, and class imbalanced problems.

3.1 Predictive Models

A broad range of machine learning algorithms have been employed in previous studies in the Bioinformatics field to predict pigmentation traits, such as Random Forests (MUNEEB; HENSCHEL, 2021) (ZAORSKA; ZAWIERUCHA; NOWICKI, 2019) and Support Vector Machine (KATSARA et al., 2021) (KUKLA-BARTOSZEK et al., 2021). The K-Nearest Neighbors is most used in the Bioinformatics field to validate Future Selection (ALZUBI et al., 2018b) or for data imputation (ROBERTS et al., 2007). The use of the K-Nearest Neighbors to classify pigmentation traits is very unusual. Our work focused on those three algortithms to predict the eye and skin color. The following subsections will present an introduction to those algorithms.

3.1.1 Random Forest

Breiman (2001) defined Random Forests as a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input **x**. Or, in other words, Random Forest consists of a pre-defined number of Decision Trees, and after a large number of trees is generated, they vote for the most popular class. According to Müller and Guido (2017), Decision Trees learn a hierarchy of if/else questions, leading to a decision. Jiang et al. (2019) introduced the Decision Tree elements as follows: each interior node of the tree corresponds to one of the input variables, each edge to children denotes one possible value of the input variable and each leaf represents the value of the target variable that is represented by the path from the root to the leaf.

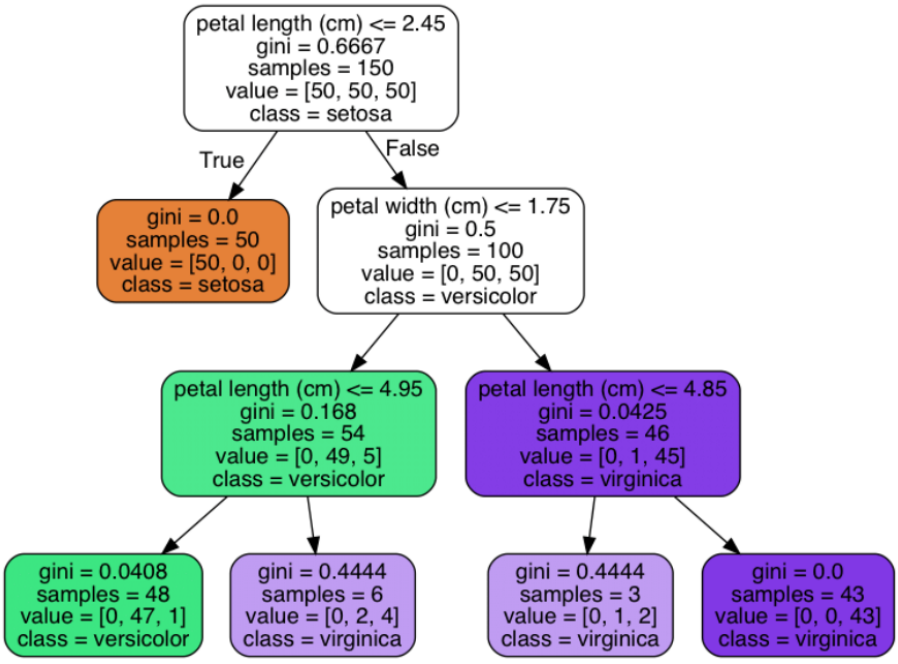
Figure 3.1 shows the Decision Tree built for the Iris flowers classification problem. The image was extracted from Scikit-Learn's page [1] about Decision Tree. The root of the tree makes a decision based on the pental length (one of the input variables). If the pental length is less or equal to 2.45cm, then the final class is Setosa, otherwise, a new decision will be made based on the petal width. For each node, a feature is selected to maximize the information gain, or, using Jiang et al. (2019) description, the splitting process is recursively repeated on each desired subset of features. The recursion stops when the subset at a node has all the same value as the target, or when splitting no longer contributes to the predictions. Using Figure 3.1 as an example, the root used the pental length information and generated a node that could make a final decision about the classification in case the pental length is less or equal to 2.45cm.

In the work presented by Zaorska, Zawierucha and Nowicki (2019), 14 SNPs were analyzed to predict skin pigmentation traits using a dataset collected from the Polish population. The data contains 222 samples (90 males and 132 females) of unrelated individuals from Poland. The traits were graded for skin color: dark (olive)/medium/light (pale), for tanning/skin sensitivity to the sun: high susceptibility to sunburns/initial sunburns (but turning brown)/moderate tanning (without sunburns)/quick tanning, for freckling: severe freckling/moderate freckling/non-freckled skin. For the purpose of binomial estimation, the phenotype categories were adjusted as follows: for skin color: dark vs. non-dark (comprising moderate and light/pale), for tanning: sunburns (comprising high susceptibility and initial sunburns) vs. non-sunburns (comprising moderate and quick tanning) and

[1] https://scikit-learn.org/

for freckling: freckled skin (comprising severe and moderate freckling) vs. non-freckled skin.

Figure 3.1: The figure shows the Decision Tree for the Iris classification problem, where each interior node corresponds to a feature of the dataset, and the leafs correspond to a target).



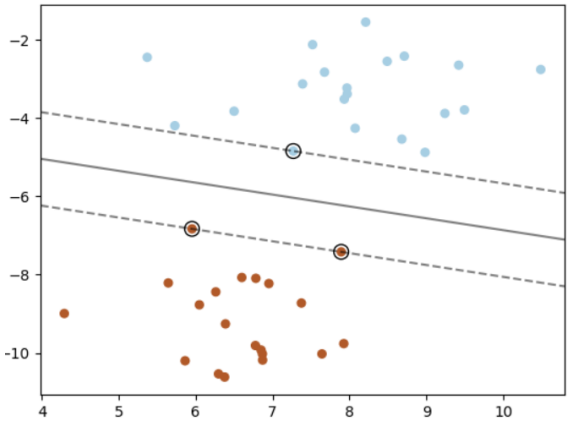
Source: Scikit-learn (https://scikit-learn.org/stable/modules/tree.html)

The 14 SNPs selected for their work are the rs12913832 in the hect domain and RCC1-like domain 2 (HERC2) gene, rs1800407, rs7495174, rs4778241, and rs4778138 in the oculocutaneous albinism II (OCA2) gene, rs12896399 in solute carrier family 24, member 4 (SLC24A4) gene, rs16891982 in solute carrier family 45, member 2 (SLC45A2) gene, rs12203592 in interferon regulatory factor 4 (IRF4) gene, rs1393350 in tyrosinase (TYR) gene, rs731236 in vitamin D receptor (VDR) gene, rs6058017, rs1015362 and rs4911414 in Agouti signaling protein (ASIP) gene, rs1805007 in melanocortin 1 receptor (MC1R) gene. Three algorithms were compared to predict skin pigmentation traits: General Linear Model based on logistic regression, Random Forest, and Neural Network. The final results showed that the Random Forest was the most accurate algorithm for 3 and 4 category estimations (total of 58.3% correct calls for skin color prediction, 47.2% for tanning prediction, 50% for freckling prediction).

3.1.2 Support Vector Machine

Using the definition presented by Xia (2020), the goal of the SVM algorithm is to use a training set of objects (samples) separated into classes to find a hyperplane in the data space that produces the largest minimum distance (called margin) between the objects (samples) that belong to different classes. According to Müller and Guido (2017), a subset of the data lies on the border between classes in the plane. Those data points are called support vectors and the maximum distance between the support vectors and the hyperplane defines the decision boundary between the classes. Figure 3.2 shows an example of how the SVM separates the classes. The blue points represent class A, the brown points represent class B, the dashed line represents the margin, the continuous line is separating the two hyperplanes created (the decision boundary), and the double circles in the dashed line represent the support vectors.

Figure 3.2: The figure shows the creation of two hyperplanes to separate the samples from two different classes, represented by the blue dots (class A) and by the brown dots (class B).



Source: Scikit-learn (https://scikit-learn.org/stable/modules/svm.html)

To predict a new data point, the distance between the test sample and the support vectors is calculated using a kernel function: Polynomial, RBF, or Sigmoidal (FACELI et al., 2011). Equation 3.1 represents the RBF Kernel, where $||x_1 - x_2||^2$ is the Euclidean distance.

$$k_{rbf}(x_1, x_2) = \exp\left(\gamma||x_1 - x_2||^2\right) \tag{3.1}$$

The algorithm is deterministic, meaning that it will always return the same result even if the data is presented in a different order. However, the results of the model are not easy to interpret (FACELI et al., 2011).

In the work presented by Katsara et al. (2021), an evaluation of supervised machine-learning methods for predicting appearance traits using SNPs was applied. Three popular machine learning classifiers were used: support vector machines, random forest, and artificial neural networks. The evaluation was focused on classifying eye, hair, and skin color by using the previously established SNPs from the IrisPlex (WALSH et al., 2011), HIrisPlex (WALSH et al., 2013), and HIrisPlex-S (CHAITANYA et al., 2018) systems (see Section 2.2). The dataset used contains 1,095 samples for eye, 1,702 for hair, and 1,318 for skin color prediction, originating from Europeans, Americans, South and East Asians, Africans, Middle Eastern, and a few admixed samples. The eye color was classified into three categories: blue, intermediate, brown; hair color into four categories: red, blond, brown, black; and skin color was classified into five categories: very pale, pale, intermediate, dark, and dark to black.

For the eye color, hair color, and skin color, 6 SNPs from the previously established IrisPlex model were selected, 22 SNPs were used for hair color prediction from the previously reported HIrisPlex model, and 36 SNPs were applied for the skin color prediction from the previously described HIrisPlex-S model. The results showed that all classification methods had a similar performance, with no method being considered superior to the others for any of the traits, meaning that any of the classifiers, including the SVM, can be used to predict pigmentation traits.

IrisPlex performs pretty well for Europeans, but not so good for the Brazilian population. The authors tested the prediction of Blue eye color in Europeans and one individual from Brazil. For the Europeans, the probability of the prediction was 0.86, while for the individual from Brazil, the probability was 0.69. The main challenge of this work is to find the SNPs to accurately classify skin and eye color, and build a classifier using data collected from the Brazilian population.

3.1.3 K-Nearest Neighbors

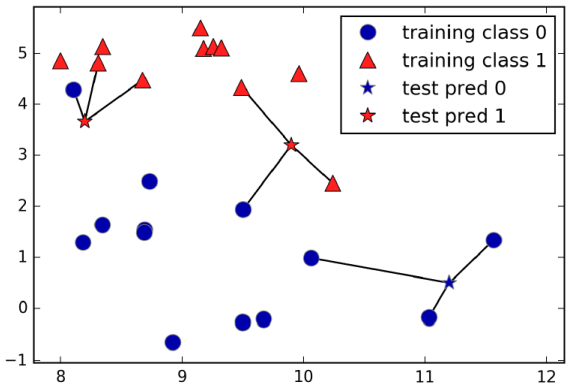
Zhang et al. (2018) defined that the principle of the K-Nearest Neighbors (KNN) algorithm is that the most similar samples belonging to the same class have a high probability. The KNN takes into consideration that instances from the same class have similar behavior and their data points are close to each other in the plane. To classify new instances, the algorithm will search for the K nearest neighbors and decide the classification for the data using the majority class between the neighbors.

The nearest data points can be calculated using a set of different distance metrics such as Euclidean Distance, Manhattan Distance, and Hamming Distance. Equation 3.2 shows the Euclidean Distance, where d is the number of attributes.

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{d} (x_i^l - x_i^l)^2} \tag{3.2}$$

For each test instance, the algorithm calculates the distance for each feature between the test sample and the train instances. After the distances are calculated, they are ranked and the classes of the K first elements are taken. The final classification will be the majority class of the K nearest instances. In case of a tie, a random class is chosen (FACELI et al., 2011). Figure 3.3 shows an example of the prediction of a new data sample considering three nearest neighbors. Class 0 is represented by the blue dots, while class 1 is represented by the red triangles, and the stars represent the new data samples to be classified. Considering the three nearest neighbors, the first star at the top left of the figure was classified as 1, because there are 2 nearest points from class 1 and only one nearest data point from class 0.

Figure 3.3: Predictions made by the three-nearest-neighbors model on the forge dataset.



Source: (MüLLER; GUIDO, 2017), p.36

The number of K is a hyperparameter of the model. It is usually necessary to run the algorithm a couple of times with different values of K to find the best value for it that minimizes the prediction error. The KNN is a lazy algorithm because it stores the values and only calculates the distance in the classification moment.

The work presented by Adam Roberts et al. (ROBERTS et al., 2007) shows a new approach to impute missing values in SNPs panels. They transform the biallelic SNPs into an array with three values. They choose to represent the majority allele of the SNP as a '0', the minority allele as a '1', and an unknown value as a '?'. With the vectors with three values, they created a pairwise mismatch vector of each SNP and generated a matrix

with all mismatched vectors. This data structure created supports fast K-nearest neighbor (KNN) searches over sliding windows in the matrix generated. To evaluate their solution, they randomly inserted missed calls in a dataset and ran 5 simulations with 5, 10, 15, 20, and 25% of unknown values. The final imputation was compared with the original data and the final results showed that the method implemented is very efficient.

3.2 Feature Selection

The Feature Selection approach is mostly used in machine learning to reduce and identify which features have a big impact on predictive models. The reduction of the dimensionality helps to increase the speed of learning algorithms, improve the accuracy of the classifier, and to remove the irrelevant and redundant features from the dataset (V; PUTHIYEDTH; R, 2019).

The work presented by Nicole Dalia Cilia et al (CILIA et al., 2019) shows the evaluation of a set of machine learning algorithms to predict different types of cancer (Breast, Colon, Leukemia, Lymphone, Lung, and Ovarian), training multiple classifiers with a different number of features. The results showed that the models trained with less number of features had a higher precision than when they were trained with a bigger number of features. And, in most cases, the results using fewer features were very close to the results using more features. As an example, one of the experiments using a Random Forest for Breast Cancer had a recognition rate of 89.70% using 200 features, and a recognition rate of 89.44% using 50 features. These results show that reducing the dimensionality of the problem can improve classification precision and the inerpretability (GRISCI; KRAUSE; DORN, 2021).

Ang et al. (2016) categorized Feature Selection into five categories: filter, wrapper, embedded, ensemble, and hybrid. The following subsections will give an introduction to these methods.

3.2.1 Filters

The filter examines the features based on the intrinsic characteristics of the data. A score is calculated for each feature, correlating to the expected output, and they are ranked based on the score. The score can be calculated using a statistical approach, like the *p*-

value, *r-squared*, *t-test*, ANOVA, and chi-squared (O'NEIL; SCHUTT, 2014). The main advantage of filter methods is the fact they are independent of any machine learning algorithm. Also, filters are largely used for being very efficient and for being computationally fast(ANG et al., 2016) (LAZAR et al., 2012).

The work presented by Nina Zhou and Lipo Wang (ZHOU; WANG, 2007) compares the results of using a *t-test*, modified for the problem, and the F-statistics to classify population groups using the SNPs information. The dataset used for them is the HapMap (haplotype map of the human genome) dataset, which has four population groups with about ten million SNPs. To rank the features, first, they ranked features in each chromosome separately. Then they combined the 22 ranking lists for the 22 chromosomes together and ranked again to obtain the total ranking list, from which they selected 5, 10, 50, 100, 200, 300, 400, 500, and 1,000 top features. After that, a classifier based on SVM was used to train using the 9 different feature sets at a time. The final results showed that using the group of 400 SNPs, the accuracy, on average, was 99.29% for the modified *t-test*, and 99.57% for the F-statistics. Therefore, it is possible to conclude that only 400 SNPs or so, from the ten million SNPs in the original data, are actually very important for differentiating the populations.

3.2.2 Wrappers

The wrapper methods select a subset of the features by minimizing the prediction error for a specific machine learning algorithm. The subsets of features are generated and validated by training and testing a specific classification model, selecting the subset that minimizes the prediction error. To search the space of all features subsets, a search algorithm is wrapped around the classification model (O'NEIL; SCHUTT, 2014) (SAEYS; INZA; LARRANAGA, 2007).

One of the approaches for wrapper methods is the Forward Feature Selection. It adds features in the subset until the best performance model is found (O'NEIL; SCHUTT, 2014). The approach starts with finding the single feature that minimizes the prediction error. Then, it generates a two-dimensional array containing the single feature selected in the previous step and adds one more feature, selecting the best combination of two features. The process continues until a subset of features that minimize the prediction error is found (XIONG; FANG; ZHAO, 2001). Another similar approach is Backward Feature Elimination, but in this case, features are removed from the subset. It starts using

all features and for each step, it eliminates one feature. The Exhaustive Feature Selection method tries all possible combinations of features. This approach has a high computational cost because it needs to test all possible subsets of features.

Chang et al. (2013) presented a study to classify osteoporosis based on SNPs, in a Taiwanese women population. In their work, three algorithms were applied: multilayer feed-forward neural network (MFNN), Naïve Bayes, and logistic regression. The method they choose to apply Feature Selection to reduce the number of SNPs was the wrapper method, searching forward for potential feature subsets. The results showed that the classifiers had a better performance using a feature selection than without the feature selection. In the case of their study, the MFNN using the wrapper method demonstrated a superior prediction performance using 4 SNPs, from a total of 22.

3.2.3 Embedded

The embedded method is a built-in feature selection mechanism that embeds the feature selection in the learning algorithm and uses its properties to guide feature evaluation (ANG et al., 2016). Examples of predictive methods that perform embedded feature selection are LASSO (MUTHUKRISHNAN; ROHINI, 2016), Random Forests (SYLVESTER et al., 2017) and Gradient Boost (JIANG et al., 2019) (BOMMERT et al., 2020).

To understand LASSO, first, it is important to understand regression. Regression expresses the relationship between the features and the expected output through a mathematical expression (O'NEIL; SCHUTT, 2014). For Linear Regression (assuming there is a linear relationship between the features and the output), for example, the Equation 3.3 represents the formula for a linear model, where x_0 to x_p denotes the features, *p* is the number of features, *w* is the weight of each feature and *b* is the y-axis offset. *b* and *w* are the parameter of the model that are learned (MüLLER; GUIDO, 2017).

$$y = w_0 * x_0 + w_1 * x_1 + ... + w_p * x_p + b \tag{3.3}$$

The LASSO method will penalize the weights of some features to force them to be exactly zero. This means some features will be entirely ignored by the model (MüLLER; GUIDO, 2017). Equation 3.4 shows the formula for LASSO, where β_0 is the constant coefficient (y-axis offset), $\beta := (\beta_1, \beta_2, \ldots, \beta_p)$ is the coefficient vector (the weights of

each feature), y_i is the output, $x_i := (x_1, x_2, \ldots, x_p)_i$ is the features vector for the i^{th} case, and t is a prespecified free parameter that determines the degree of regularization (MUTHUKRISHNAN; ROHINI, 2016).

$$\begin{aligned} \beta = arg\ min \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2, \\ subject\ to \sum_{j=1}^{p} |\beta_j| \leq t \end{aligned} \tag{3.4}$$

Muthukrishnan and Rohini (2016) experiment used a dataset for Diabetes to compare the prediction results for the measure of the growth of the disease using three different approaches for feature selection: LASSO, Ridge Regression (MCDONALD, 2009), and Ordinary Least Squares (OLS) (BURTON, 2021). The results showed that LASSO works better than the other methods, and can be used as an alternative for feature selection.

USAI, GODDARD and HAYES (2009) presented a study for genomic selection using LASSO. Two datasets were used in the experiment: a simulated dataset with 5,865 individuals and 6,000 Single Nucleotide Polymorphisms (SNPs) and a mouse dataset with 1,885 individuals genotyped for 10,656 SNPs. The prediction equation applied to estimate the effectiveness of the genomic selection was the Equation 3.5.

$$GEBV_c = X_c \beta_r \tag{3.5}$$

Where GEBV means genomic estimated breeding values, X_c is the design matrix allocating the marker genotypes in the candidate population and β_r is the SNP effects vector estimated in the reference population. The results showed that for both datasets, LASSO often outperformed, and for the candidate population, LASSO reached an accuracy of 89% using 156 SNPs. The conclusion was that the LASSO approach is a good alternative method to estimate marker effects for genomic selection.

Random Forests consist of a pre-defined number of Decision Trees (see Section 3.1). Each Decision Tree will be generalized with a random subset of the features using the bagging method (bootstrap aggregating) with replacement. The features selected are the ones that maximize the information gain. The information gain is a criterion used to determine the quality of a split (NOWOZIN, 2012). The number of Decision Trees and the number of features to be selected are hyperparameters of the model.

The work presented by Sylvester et al. (2017) shows the use of SNPs for fine-scale salmon population assignment. Two different datasets were used: Alaskan Chinook salmon, with 10,944 SNPs, and Atlantic salmon, with 220,000 SNPs. To reduce the number of SNPs in the datasets, Random Forest, Regularized Random Forest, guided Regularization Random Forest, and F_{ST} rank were applied to create panels of 40 to 700 SNPs. Random Forest methods often outperformed the F_{ST} rank. For the Atlantic salmon data, in the small panel sizes (50-100 SNPs), F_{ST} rank had better or comparable self-assignment accuracy than Random Forest, which performed better with small to medium panels (101-200 SNPs). For the Alaskan Chinook salmon data, the Random Forest performed better with small to medium panels (up to 200 SNPs). Across all panel sizes, the Random Forest had an accuracy, on average, bigger than 80% for each dataset.

Gradient Boost methods usually use a combination of weak algorithms to improve prediction. Weak algorithms are the algorithms with low accuracy when used solo, like Decision Trees. The gradient boosted regression tree is a method that combines multiple Decision Trees to create a more powerful model. It works by building trees in a serial manner, where each tree tries to correct the mistakes of the previous one. The feature importances of the gradient boosted trees are somewhat similar to the feature importances of the random forests, though the gradient boosting completely ignored some of the features (MüLLER; GUIDO, 2017).

Jiang et al. (2019) presented a study to predict a certain genetic disease using the SNPs information from data collected by the Southeastern University of China. The dataset has the information of 1,000 individuals (500 cases and 500 controls) with 9,000 SNPs each. In their experiment, they applied a Gradient Boosting algorithm slightly modified by introducing Gini Impurity as a strategy to choose candidate SNPs, and Decision Trees was chosen as the weak learner. Jiang et al. (2019) defined the gain of Gini Impurity (GI) as being a measure of how often the element would be correctly labeled if it is randomly labeled. The Gini impurity is calculated as shown in Equation 3.6, where p is the tree node, p_i is the fraction of items labeled with class i in the set, and J is the total number of classes.

$$GI(p) = \sum_{i=1}^{J} p_i(1 - p_i) = 1 - \sum_{i=1}^{J} p_i^2 \tag{3.6}$$

The gain of GI is computed according to Equation 3.7, where N and GI represents

the quantity of individuals at a node and its GI, respectively. p is the parent node and d is its child.

$$Gain(p) = GI(p) - \sum_{d \in p} \frac{N_d}{N_p} \times GI(d) \tag{3.7}$$

Comparing the results cross all algortithms applied for the classification (Naïve Bayes, SVM, Random Forest, and Gradient boosting), the results from the experiment showed that the Gradient boosting algorithm had a better performance in the classification than the other algorithms applied, with a precision of 92.92%.

3.2.4 Ensemble

Ang et al. (2016) presented the ensemble method as a method that aims to construct a group of feature subsets and then produce an aggregate result out of the group. It is purposely designed to tackle the instability and perturbation issues in many feature selection algorithms. Yang et al. (2013) described the two most commonly used approaches for feature selection using ensemble: Ensemble-based on data perturbation and Ensemble-based on different data partitioning.

The idea behind the ensemble based on the data perturbation approach is to disturb the data by creating many versions of it and apply a feature selection method in each version created for the data. An aggregation of the results will define the features to be selected. Pengyi Yang et al. used as an example the use of a filter method to rank the features and the bootstrapping as the perturbation method to create many versions of the same data. The final rank of the features selected is the average of the rank results.

The ensemble-based on different data partitioning is based on partitioning the training and testing data differently, generating different training and test samples for the same data, and applying a wrapper method to select the features. The final feature subset is determined by calculating the frequency of each feature that was selected from each partitioning, where the features with a higher frequency will be in the final feature set.

The work by Abeel et al. (2009) presented a study in biomarker identification for cancer diagnosis using ensemble for feature selection. The algorithm chosen by the authors was a linear SVM (see Section 3.1). The advantage of the SVM is the fact that SVMs contain an embedded capability for feature selection. The method chosen to re-

duce the features was backward elimination. Using the weights calculated by the SVM model for each feature, the features with lower weights are removed from the feature set, and the model is trained again with the remaining features. The process stops when all features have been removed or a desired number of features is reached. For the ensemble method, the SVMs were trained with different subsets of all features and the output was aggregated to return the final result. The final results showed that using ensemble feature selection techniques could improve around 15% the classification performance, other than that the solution showed an increase of up to almost 30% in the robustness of the selected biomarkers.

3.2.5 Hybrid

The hybrid method can be either the combination of two different methods, two methods of the same criterion, or two feature selection approaches. By combining multiple methods, the hybrid approach can take advantage of the associated techniques (ANG et al., 2016). Alzubi et al. (2018a) proposed a solution for feature selection using a fusion of filter and wrapper to detect the most informative SNPs. The filter approach chosen was the Conditional Mutual Information Maximization (CMIM) (FLEURET, 2004), and the wrapper approach chosen was the Support Vector Machine Recursive Feature Elimination (SVM-RFE) (GUYON et al., 2002). The CMIM selects features that maximize their mutual information with the class to predict, achieving the balance between individual power and independence through the comparison of the new feature with the features that have already been selected. Equation 3.8 shows how the CMIM is calculated for each new feature, where S are the features already selected, Y is the target, X_j is a feature that was already picked ($j \in S$), X_n is a new candidate feature to be selected, and H represents the conditional entropy. A feature X_0 is considered good, if $I(Y; X_0|X)$ is large for every X already selected, meaning that X_0 is carrying information about Y that has not been captured yet.

$$\begin{aligned} CMIM(X_n) &= \min_{j \in S} I(X_i; Y|X_j) \\ I(X_i; Y|X_j) &= H(X_i|X_j) - H(X_i|X_j, Y) \end{aligned} \tag{3.8}$$

The SVM-FRE adopts a backward feature elimination. It begins with all features

set and eliminates the features that are least important to the SVM classifier, creating a ranked list where the top features can be selected to obtain an optimal subset of features. In each iteration, a new SVM is trained and a new feature is removed. The rank is created by sorting the features by order of elimination, where the first removed features are the less important ones. The solution proposed by the authors first applies the CMIM to pre-filter the most relevant SNPs and then applies the SVM-FRE to obtain the optimal SNPs subset. The CMIM is first applied once the feature space is huge and it would require a high computational cost for the SVM-FRE to process all the SNPs. Besides that, the SVM-FRE does not take into consideration the redundancy among SNPs, so it is necessary to remove the irrelevant and redundant information through the CMIM.

To evaluate the solution, five datasets were chosen from the NCBI GEO repository: Thyroid Cancer (TC) (with 1,000,000 SNPs), Autism (ASD) (with 250,000 SNPs), Colorectal Cancer (CC) (with 250,000 SNPs), Mental Retardation (MR) (with 250,000 SNPs), and Breast Cancer (BC) (with 500,000 SNPs). The proposed novel was compared against four different feature selection approaches: Minimum Redundancy Maximum Relevance (mRMR) (PENG; LONG; DING, 2005), ReliefF (ROBNIK-ŠIKONJA; KONONENKO, 2003), Fast Correlation Based Feature Selection (FCBF) (YU; LIU, 2003), and CMIM. Four classifiers were chosen to compare the feature selection approaches: Support Vector Machine (SVM) (HEARST et al., 1998), K-Nearest Neighbors (KNN) (FIX; HODGES, 1989), Naïve Bayes (NB) (RISH et al., 2001), and Linear Discriminant Analysis (LDA) (XANTHOPOULOS; PARDALOS; TRAFALIS, 2013). The results showed that for the ASD dataset, the proposed novel could achieve the best accuracy (89.50) using only 100 SNPs and the SVM classifier. For the BC dataset, the solution achieved the best accuracy (96.39) using only 50 SNPs and the SVM classifier. For the MR dataset, the solution achieved the best accuracy (85.00) using only 50 SNPs and the SVM classifier. For the MR dataset, the solution could achieve the best accuracy (85.00) using only 50 SNPs and the SVM classifier. The TC dataset achieved the best accuracy (90.37) using only 100 SNPs and the SVM classifier.

3.3 Class Imbalance Problem

The class imbalance problem is one of the main challenges in the machine learning field. The problem happens when a big number of instances in a dataset are labeled as one class, called majority class, while fewer are labeled as the other class, called minority

class. The imbalance degrades the performance of machine learning algorithms because the decision-making will be biased to the majority class (ELRAHMAN; ABRAHAM, 2013) (GUO et al., 2008). In many real-world scenarios, the prediction of the minority class is very important for cases such as medical diagnosis, for example, where the number of patients with a rare disease is much lower than the number of patients that do not have the disease. According to Guo et al. (2008), from the view of the applications, the nature of the imbalanced class can be because the data are naturally imbalanced or the data are not naturally imbalanced but it is too expensive to obtain data of the minority class.

Kaur, Pannu and Malhi (2020) split the imbalance class approach into four techniques: pre-processing approaches, cost-sensitive learning methods, algorithmic centered approaches, and hybrid methods. Besides these four approaches, new studies have used Generative Adversarial Networks (GANs) to generate synthetic samples for the minority class to deal with the imbalance class problem (CAI et al., 2019a) (DOUZAS; BACAO, 2018) (SANTOS; ARANHA, 2019). The following subsections will address the techniques presented by Kaur, Pannu and Malhi (2020), with the difference that cost-sensitive learning methods are classified for many authors as an algorithmic centered approache (GUO et al., 2008) (ELRAHMAN; ABRAHAM, 2013) (SPELMEN; PORKODI, 2018). Because of this consideration, it was decided to discuss this method in the algorithmic-centered approach subsection, rather than discussing it as a separate topic. The section 3.4 will present the theoretical concept of GANs and the studies for its use for the imbalance class problem.

3.3.1 Pre-processing approaches

The pre-processing approaches are performed on the training data, changing the class distribution to reduce the ratio between them. The most popular way to pre-processing the data is the sampling methods, which include under-sampling, over-sampling, and hybrid sampling (KAUR; PANNU; MALHI, 2020) (ELRAHMAN; ABRAHAM, 2013).

- *Over-sampling*: Guo et al. (2008) describes over-sampling as a non-heuristic method that aims to balance the class distribution by random replicating the samples of the minority class. The main shortcoming of this approach is the model over-fitting since it makes copies of the minority samples. Nitesh et al. (2002) introduced a

heuristic solution called Synthetic Minority Over-sampling Technique (SMOTE), where the main idea is to create synthetic samples rather than over-sampling the minority class with replacement. It first randomly selects a point in the minority class. Second, it searches for the k nearest neighbors of the same class. It finally selects a new point between each closest neighbor and the random point selected. These new data points are randomly located in the vector between the k nearest neighbors and the point selected in the first step (SANTOS; ARANHA, 2019). SMOTE avoids the over-fitting problem since it does not replicate the minority samples. He et al. (2008) presented a solution called Adaptive Synthetic Sampling Approach (ADASYN), which generates more synthetic data for the minority examples. The ADASYN method is very similar to the SMOTE, where it searches for the k nearest neighbors of a selected data point to generate synthetic data. But the key difference between the two methods is that ADASYN uses a density distribution as a criterion to automatically decide the number of synthetic data that needs to be generated, rather than pick up one data point between the first random point selected and each neighbor.

- *Under-sampling*: Guo et al. (2008) describes under-sampling as a non-heuristic method that aims to balance the class distribution by eliminating samples from the majority class. The shortcoming of this approach is the loss of useful information, especially when the training data is small. Hart (1968) introduced the Condensed Nearest Neighbor Rule, where the algorithm sets up bins called STORE and GRABBAG. First, one sample of the data is placed in the STORE bin. Second, it picks a second sample of the data and classifies this sample using the nearest neighbor rule comparing it with the samples in STORE. If this sample is classified correctly, it is placed in the GRABBAG bin, otherwise, it is placed in STORE. After one passes through the original sample set, the procedure continues to loop through GRABBAG until termination. The termination can occur in two different ways: All the samples from the GRABBAG were transferred to STORE (in this case, the consistent subset found in the entire original set); no samples were transferred to STORE because the underlying decision surface has not been changed. Finally, the samples in the GRABBAG are discarded and the set in STORE is used as reference points. The main idea is that the Condensed Nearest Neighbor Rule will pick out points near the boundary between classes. Typically points deeply embedded within a class will not be transferred to STORE. Kubat and Matwin (1997) presented the

One-Sided Selection approach, a very similar solution to the Condensed Nearest Neighbor Rule. Let *S* be the original training set. Initially, *C* contains all the samples from the minority class from *S* and one randomly selected sample from the majority class. It first classifies *S* with the 1 nearest neighbor rule, using the examples in *C*, and compares the assigned labels with the original ones. Then, it moves all the misclassified examples into *C*. Finally, it removes from *C* all the examples of the majority class that is distant from the decision boundary, since these examples might be considered less relevant for learning (GUO et al., 2008).

- *Hybrid sampling*: Kaur, Pannu and Malhi (2020) describes hybrid sampling as a method that applies both resampling techniques, over-sampling, and under-sampling. Qian et al. (2014) presented a solution for the imbalance class problem where the minority classes are over-sampled and majority classes are under-sampled. It first splits the training set into different classes. Second, it applies an under-sampling technique, randomly selecting samples from the majority class, and applies the SMOTE technique in the minority class. Wang (2014) presented an approach for resampling the training set using a hybrid solution based on SVM to address the imbalance problem. The proposed approach first uses the SVM method to generate a classification hyperplane and applies an under-sampling technique to reduce the majority of samples. For that, it deletes some samples far away from the hyperplane according to the calculated distances between the samples and the hyperplane. Then, it divides the training dataset into several subsets, in which it synthesizes new samples for the minority class using SMOTE technique.

Hasibuan, Kusuma and Suwamo (2014) presented the study for identification of SNP in cultivated soybean using SVM in an imbalanced dataset. Under-sampling and over-sampling were applied to obtain balanced data. The SNPs were labeled as $+1$ for positive SNP, or -1 for negative SNP. The genomic data used was limited to chromosome number 16, which has 1,524,576 candidate SNPs. 1,500 random samples were extracted from the total candidates for the positive SNPs, and 15,000 random samples were extracted from the total candidates for the negative SNPs. For the under-sampling technique, the negative SNPs were grouped into 10 clusters, using the K-means algorithm, and from each cluster, the negative SNPs were randomly selected, according to the Equation 3.9, where $m \times Size_{MI}$ is the total number of selected majority class samples that it is supposed to have in the final training dataset, $\sum_{i=1}^{K} Size_{MA}^{i}/Size_{MI}^{i}$ is the total ratio of the number of majority class samples to the number of minority class samples in

all clusters, K is the number of clusters, and $Size^i_{MA}/Size^i_{MI}$ is the ratio of the number of majority class samples to the number of minority class samples in the ith cluster. MI and MA means the minority class and majority class, respectively.

$$SSize^i_{MA} = (m \times Size_{MI}) \times \frac{Size^i_{MA}/Size^i_{MI}}{\sum_{i=1}^{K} Size^i_{MA}/Size^i_{MI}} \tag{3.9}$$

For the over-sampling technique, after taking 25% of the negative data by the under-sampling, the positive data was replicated to obtain balance data. The SVM was trained using three different samplings and the results were compared with the SVM trained with the imbalance data. The dataset generated by under-sampling with $m = 1$ (Equation 3.9) had an accuracy of 87.10%, true positive rate (TPR) of 0.96, and false positive rate (FPR) of 0.21. The dataset generated by under-sampling with $m = 2$ (Equation 3.9) had an accuracy of 85.82%, true positive rate (TPR) of 0.88, and false positive rate (FPR) of 0.15. The dataset generated by over-sampling had an accuracy of 88.97%, a true positive rate (TPR) of 0.96, and a false positive rate (FPR) of 0.18. The classifier trained with the original dataset had an accuracy of 93.21%, a true positive rate (TPR) of 0.51, and a false positive rate (FPR) of 0.02.

The final results showed that the classifier trained with a balanced dataset could identify SNPs better than the one trained using the original data.

3.3.2 Algorithmic centered approaches

The algorithmic centered approaches deal with the bias produced by the imbalanced data by improving existing classifiers (HAIXIANG et al., 2017). Many classifier algorithms produces a score that represents the degree to which an examples is a mesmber of a class (GUO et al., 2008). Changing how these classifiers define the treashold between the classes is a very commom approach to deal with the imbalanced data (YU et al., 2015) (WU et al., 2016). Spelmen and Porkodi (2018) classified the algorithmic centered approaches into ensemble-based methods, threshold methods, one class learning, cost sensitive learning, and active learning methods.

- *Ensemble-based methods*: Elrahman and Abraham (2013) defined the ensemble-based method as a combination of multiple classifiers to improve the generalization ability and increase the prediction accuracy. The most popular ensemble method is boosting and bagging. Boosting convert weak learning models into a learning

model with better generalization (GANAIE et al., 2021), where each classifier is dependent on the previous one, and focuses on the previous one's error. Data samples that are misclassified in the previous classifiers are chosen more often or weighted more heavily (ELRAHMAN; ABRAHAM, 2013). AdaBoost, introduced by Freund and Schapire (FREUND; SCHAPIRE, 1997), trains the classifiers serially and after each round it updates the weights of the instances, giving more focus to the misclassified ones. First, all examples start with the same weight and for each iteration, the weights are adjusted, increasing for the instances that are harder to classify and decreasing for the ones that are correctly classified. Also, each classifier has a different weight depending on its accuracy, where more confidence is given to more accurate ones. When a new instance is submitted, each classifier gives a weighted vote, and the class label is selected by the majority (GALAR et al., 2011). Whereas, bagging, also known as bootstrap aggregation, trains different learning models by replicating the original training examples. A new dataset is randomly created (with replacement, meaning that the same sample can be chosen more than once for the same new dataset) using the instances from the original dataset. The new examples generated usually maintain the original data size. When an unknown instance is presented to each classifier, a majority or weighted vote, from all classifiers, is used to infer the class (GALAR et al., 2011). The Random Forest (see Section 3.1) uses the bagging strategy for improving the predictions of the base classifier which is a decision tree. Each decision tree will learn with a different subset of features and for each tree, a new dataset is generated using bootstrap with replacement (GANAIE et al., 2021). When an unknown instance is presented, a majority vote from all the trees is used to infer the class.

- *Threshold methods*: According to Esposito et al. (2021), thresholding methods aim to identify the optimal decision threshold for classification (once the decision boundary can be biased towards the majority class), typically by maximizing a balanced accuracy metric on a validation set through cross-validation or bootstrapping. Yu et al. (2015) proposed a support vector machine-based solution to iteratively search the optimal position for the classification hyperplane. Since the classification accuracy is skewed, other specific evaluation metrics are considered in the solution to evaluate the performance of the learner. Table 3.1 shows the confusion matrix used for specific metrics, where the columns represent the predicted classes and the rows represent the actual classes. The positive class is the class of inter-

est, in this case, the minority class, and the negative class is the other class, in this case, the majority class. The true positive value is the number of instances in the positive class that was correctly classified. The false positive value is the number of instances in the negative class that was misclassified as the positive class. The false negative value is the number of instances in the positive class that was misclassified as the negative class. And the true negative value is the number of instances in the negative class that was correctly classified as the negative class.

Table 3.1: Confusion matrix

	Predicted positive class	**Predicted negative class**
Actual positive class	TP (True Positive)	FN (False Negative)
Actual negative class	FP (False Positive)	TN (True Negative)

The Equation 3.10 calculates the precision, also known as the positive predictive value, of the classifier. The Equation 3.11 calculates the recall, also known as sensitivity or true positive rate (TPR). The Equation 3.12 calculates the harmonic mean of precision and recall. Equation 3.13 calculates the true negative rate (TNR). And the Equation 3.14 calculates the geometric mean of sensitivity and specificity, reflecting the balance between the accuracy of the minority and majority classes.

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

$$Recall(TPR) = \frac{TP}{TP + FN} \tag{3.11}$$

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{3.12}$$

$$TNR = \frac{TN}{TN + FP} \tag{3.13}$$

$$G - mean = \sqrt{TPR \times TNR} \tag{3.14}$$

The iterative search algorithm proposed considers the output values for all misclassified instances to generate candidate positions for the optimal classification hyperplane. Each candidate position is adjusted to the middle between the corresponding

misclassified minority class instance (x_i) and its nearest neighbor instance (nx_i) belonging to the majority class that lies far from the original classification hyperplane. The adjusted distance θ_i is calculated according to Equation 3.15, where h is the original decision function.

$$\theta_i = \frac{-h(x_i) - h(nx_i)}{2} \tag{3.15}$$

The G-mean value of all candidate positions is compared to find the optimal position and its adjusted distance. Then all the G-mean values are ranked in descending order and the optimal position ($\theta_{optimal}$) corresponding to the highest G-mean is obtained. The final decision threshold h' is calculated using Equation 3.16, where h is the original decision threshold.

$$h'(x) = h(x) + \theta_{optimal} \tag{3.16}$$

- *One class learning*: One class learning aims to build models using only a single class of data, predicting whether an instance belongs to the target class or is an outlier. The most common approach for it is to use the statistical distribution of the data from a single class and classify the unknown instances as belonging to the target class (high-density values) or the set of outlier classes (low-density values) (BELLINGER; SHARMA; JAPKOWICZ, 2012). From a Bayesian perspective, the probability density function of the given target class can be represented by Equation 3.17.

$$Classification(x) = \begin{cases} target & \text{if } p(x|\omega) \geq \tau \\ outlier & \text{otherwise} \end{cases} \tag{3.17}$$

Where $p(x|\omega)$ is the probability of class ω, for a given example x, and τ is a threshold defined for the classification. Hempstalk, Frank and Witten (2008) presented a solution where artificial data is generated to take the role of a second class, converting the one-class problem into a binary classification. The solution first obtains a rough estimation of the density of the target class and then it generates artificial data that is close as possible to the target class, using the Equation 3.18.

$$P(X|T) = \frac{(1 - P(T))P(T|X)}{P(T)(1 - P(T|X))} P(X|A) \tag{3.18}$$

Where T denotes the target class, X is the instance and A is the artificial class. To use the equation in practice, the value for $P(X|A)$ is chosen and the amount of data to be generated is specified by the user. After the generation of the artificial class, the problem can be solved as a binary classification.

- *Cost-sensitive learning methods*: Kaur, Pannu and Malhi (2020) defined cost-sensitive learning as a cost-specific technique that finds costs associated with misclassified examples. The cost learning techniques take the misclassification error cost into its account by assigning a higher cost to the minority class (ELRAHMAN; ABRAHAM, 2013). The method can be incorporated both at the data level and at the algorithm level. Compared with resampling methods, cost-sensitive learning is more computationally efficient, but less popular (HAIXIANG et al., 2017). The cost model takes the form of a cost matrix (Table 3.2), where the diagonal elements are zero, meaning that correct classification has no cost, and the cost of misclassifying is the entries C_{ij} and C_{ji} (GUO et al., 2008).

Table 3.2: Cost-sensitive learning matrix

		Prediction	
		Class i	**Class j**
True Class	**Class i**	0	C_{ij}
	Class j	C_{ji}	0

Domingos (1999) presented a method to make a classifier cost-sensitive (MetaCost), where the main idea behind MetaCost is to relabel the training examples with their optimal classes. To do that, the algorithm uses a bootstrap with replacement in the training examples and applies a learning classifier in each bootstrap. Using the output of each bootstrap, it estimates the class's probability by the fraction of votes that it receives from the ensemble of all outputs. Finally, it relabels each example of the training set using the equation 3.19, where $P(j|x)$ is the probability of each class j, for a given example x, and $C(i, j)$ is the cost-sensitive matrix. The Bayes optimal prediction ($R(i|x)$) for x is the class i that minimizes the expected cost of predicting that x belongs to class i. The matrix and the learning classifier are parameters for the MetaCost algorithm.

$$R(i|x) = \arg\min_i \sum_j P(j|x)C(i,j) \tag{3.19}$$

- *Active learning methods*: The active learning method can actively choose the training data, and it is usually used to label unknown instances to create a training set since manually labeling a huge number of instances can be time-consuming and costly (TONG; KOLLER, 2001). Ertekin et al. (2007) proposed a solution using SVM and the active learning method to deal with the class imbalance problem. SVM-based active learning can pick up instances by checking their distances to the hyperplane since instances close to the hyperplane are more informative for learning. The empirical solution proposed by the authors is that the imbalance ratio of the classes within the margin in real-world data is generally much lower than the entire data. This means that the solution proposed will provide the learner with more balance classes picking up instances close to the hyperplane. The algorithm works as follows: First, an SVM learner is trained using all the existing training data. Second, it selects the closest instance to the hyperplane. Then, this newly selected instance is added to the training set and the SVM is trained again. Because each iteration needs to recompute the distance to the new hyperplane, the solution proposes a selection method that does not search through the entire dataset. Picking a random instance, with 95% of probability that is among the top 5% closest instance, and using the Equation 3.20, where $p\%$ is the top p percent closest instances with probability $1 - \eta$, the size of the random sampling is 59. This means that 59 random instances will be picked and the closest instance to the hyperplane will be selected from the 59 samples in the second step of the iteration.

$$L = \log \eta / \log(1 - p\%) \tag{3.20}$$

Cheng et al. (2015) presented a solution for the class imbalanced problem in bioinformatics using boundary movement-based, called BM-ELM. The algorithm proposed can be divided into three stages: first, it uses an Extreme Learning Machine (ELM) to train a classifier with the original training set. ELM is a fast algorithm to train single hidden layer forward networks that randomly generates the weights and bias between the input layer and the hidden layer, then uses the least-square algorithm to get the solution of the hidden layer output weights. Second, all the instances are projected on a one-dimensional space according to the distance between each example and the initial hyperplane. It uses the kernel density estimation (KDE) approach to obtain the probability density distribution curves of the two different classes. Third, it finds the intersecting point of the two

density curves. The distance between the intersecting point and the original point denotes the optimal movement distance of the original classification hyperplane.

The solution was evaluated using four imbalanced bioinformatics datasets: MicroRNA precursors (XUE et al., 2005), SNP (GENG; YU-QUAN; YANG, 2018), Box H/ACA snoRNA, and Box C/D snoRNA (HERTEL; HOFACKER; STADLER, 2008). The MicroRNA precursors dataset has 8,687 instances and a class imbalance ratio of 44.01. The SNP dataset has 3,074 instances and a class imbalance ratio of 15.80. The Box H/ACA snoRNA dataset has 8,510 instances and a class imbalance ratio of 129.92. The Box C/D snoRNA has 45,515 instances and a class imbalance ratio of 147.74. The classifiers used to compare against the BM-ELM were the ELM, weighted ELM, ELM with random under-sampling (RUS), ELM with random over-sampling (ROS), ELM with SMOTE, and SVM with RUS. The final results were very similar between all classifiers using a class imbalance approach. Compared with the ELM without any class imbalance approach, the other classifiers outperformed by comparing the sensitivity and G-mean. For the SNP dataset, the BM-ELM obtained a sensitivity of 0.61, a specificity of 0.81, and a G-mean of 0.70. The ELM obtained a sensitivity of 0.01, a specificity of 0.99, and a G-mean of 0.06. For the MicroRNA precursors dataset, the BM-ELM obtained a sensitivity of 0.88, a specificity of 0.91, and a G-mean of 0.90. The ELM obtained a sensitivity of 0.04, a specificity of 1.00, and a G-mean of 0.20. For the Box H/ACA snoRNA, the BM-ELM obtained a sensitivity of 0.95, a specificity of 0.95, and a G-mean of 0.95. The ELM obtained a sensitivity of 0.00, a specificity of 1.00, and a G-mean of 0.00. For the Box C/D snoRNA, the BM-ELM obtained a sensitivity of 0.96, a specificity of 0.95, and a G-mean of 0.95. The ELM obtained a sensitivity of 0.00, a specificity of 1.00, and a G-mean of 0.00.

The final results showed that the BM-ELM, and the other classifiers, outperformed the ELM in sensitivity and G-mean. The BM-ELM could guarantee a small loss of specificity if compared to the ELM trained without any class imbalance approach.

3.3.3 Hybrid methods

Spelmen and Porkodi (2018) describes the hybrid method as a combination of the data level and the algorithmic level method, to overcome the problems in both methods, and also to have a better classification accuracy. Seiffert et al. (2010) proposed a solution called RUSBoost using random under-sampling (RUS) and AdaBoost technique. The al-

gorithm works as follows: First, all the instances are weighted initially as $1/m$, where m is the number of examples in the training set. Second, T weak learners are iteratively trained. For each iteration, RUS is applied to randomly remove examples from the majority class until the class ratio achieves the desired ratio. After that, this new training set generated by RUS is passed to the weak learner with their respective weights, and a weak hypothesis is obtained from the learner. With the hypothesis obtained, a pseudo loss $\in_t$ is calculated using the Equation 3.21 based on the original training set, where i represents the ith example, t represents the tth iteration, $D_t(i)$ is the weight of the ith example in the tth iteration, h_t is the hypothesis from the weak learner, x_i is the ith example, y_i is the class of the ith example, and y is the class different from y_i.

$$\in_t = \sum_{(i,y):y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y)) \tag{3.21}$$

With the $\in_t$ calculated, the weight update parameter α_t is obtained as $\in_t /(1- \in_t)$. Next, the weight distribution for the next iteraction D_{t+1} is uptaed (Equation 3.22) and normalized (Equation 3.23).

$$D_{t+1}(i) = D_t(i)\alpha_t^{\frac{1}{2}(1+h_t(x_i,y_i)-h_t(x_i,y:y\neq y_i))} \tag{3.22}$$

$$\begin{aligned} Z_t &= \sum_i D_{t+1}(i) \\ D_{t+1}(i) &= \frac{D_{t+1}(i)}{Z_t} \end{aligned} \tag{3.23}$$

After the T iterations, the final hypothesis is obtained as the weighted vote of the T weak hypotheses, as shown in Equation 3.24.

$$H(x) = arg\ max \sum_{t=1}^{T} h_t(x, y) \log \frac{1}{\alpha_t} \tag{3.24}$$

Schubach et al. (2017) presented a novel method for imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants, called hyperSMURF. The solution proposed can be split into three phases: first, simultaneous over-sampling and under-sampling are applied. The negative examples (majority class) are subdivided into n non-overlapping partitions, and each partition is randomly subsampled to reduce the number of negative samples. For the over-sampling, the SMOTE is

applied in the positive examples, and, for each partition, the oversampled data is added, resulting in balanced datasets. Second, each partition generated in the first step is used to train n Random Forests (RF). Any ensemble of learning machines can be used, but the authors choose RF. Third, the results of the n classifiers are combined by averaging across the probabilities estimated by each Random Forest. The combination of the Random Forests (ensemble of decision trees), and the ensemble in the third step, results in a hyper-ensemble (an ensemble of ensembles).

To evaluate the solution, two datasets were used: Mendelian data (SMEDLEY et al., 2016) and GWAS data (MA et al., 2015). The Mendelian dataset is extremely imbalanced, by, approximately, one positive regulatory Mendelian mutation to every 36,000 negative non-deleterious variants, and the GWAS dataset has an imbalance of $\sim$1:700. The state-of-the-art methods for scoring variants (Combined Annotation–Dependent Depletion (CADD) (KIRCHER et al., 2014), Genome-Wide Annotation of variants (GWAVA) (RITCHIE et al., 2014), Deep Learning–based Algorithmic framework (DeepSEA) (ZHOU; TROYANSKAYA, 2015), and Eigen (IONITA-LAZA et al., 2016)) were used for performance comparison, and the metric for the evaluation was the Area Under the Precision-Recall Curve (AUPRC). For the Mendelian database, the hyperSMURF had an AUPRC of 0.42, the CADD had an AUPRC of 0.09, and the Eigen had an AUPRC of 0.01, the GWAVA had an AUPRC of 0.15, and the DeepSEA had an AUPRC of 0.05. For the GWAS database, the hyperSMURF had an AUPRC of 0.635, the CADD had an AUPRC of 0.03, the Eigen had an AUPRC of 0.00, the GWAVA had an AUPRC of 0.40, and the DeepSEA had an AUPRC of 0.23. The final results showed that the hyperSMURF achieved significantly better results than the state-of-the-art methods.

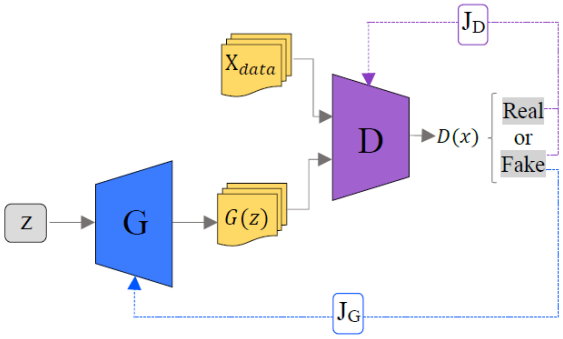
3.4 Generative Adversarial Networks

The generative adversarial network (GAN) was first introduced by Goodfellow et al. (2014) as a framework for estimating generative models via an adversarial process. The GAN architecture consists of two networks: the generator and the discriminator. The generator's goal is to learn the statistical distribution of the real data to be able to generate fake data that is indistinguishable from real-world data. The discriminator is often a binary classifier that discriminates real data from the fake data generated by the generator. The adversarial in the name comes from the fact that both networks are trained simultaneously, and in competition with each other (SALEHI; CHALECHALE; TAGHIZADEH,

2020) (WANG et al., 2017). Gui et al. (2021) describes the optimization of GANs as a minimax problem, where the goal is to reach the Nash equilibrium. According to Holt and Roth (2004), the Nash equilibrium can be interpreted as a potential stable point of a dynamic adjustment process in which individuals adjust their behavior based on the strategy of other players in the game, searching for a strategy that will give them better results. This means that a participant cannot change their strategy without altering the strategy of other participants. The generator and the discriminator reach a state where one cannot progress without changing the other. The minimax refers to minimizing the loss in the generator and maximizing the loss in the discriminator (SALEHI; CHALECHALE; TAGHIZADEH, 2020). The training process of GANs involves finding the parameters for the discriminator that maximize the classification accuracy and finding the parameters for the generator that maximally confuse the discriminator (CRESWELL et al., 2018).

Figure 3.4 shows the architecture of the GAN, where X_{data} represents the real data, $G(z)$ represents the fake data generated by the generator, G is the generator, D is the discriminator, $D(x)$ is the output from the discriminator, z is a noise vector with uniform distribution or Gaussian distribution, J_D and J_G are the loss functions that update the learning process of the discriminator and the generator, respectively (SALEHI; CHALECHALE; TAGHIZADEH, 2020).

Figure 3.4: The architecture of the GAN.



Source: (SALEHI; CHALECHALE; TAGHIZADEH, 2020), p.4

During the training process, the generator tries to capture the distribution of true examples and generates new data using the noise vector z received as input, mapping the representation space z, called latent space, to the space of the real data (GUI et al., 2021) (CRESWELL et al., 2018). According to Wang et al. (2017), eventually, when the discrimination ability of D has been improved to a high level but cannot discriminate the data correctly, the generator had captured the distribution of the real data.

Equation 3.27 represents the GAN optimization strategy, as a minimax problem.

Salehi, Chalechale and Taghizadeh (2020) broke down the Equation 3.27 into Equation 3.25 and Equation 3.26, for a better understanding. According to Equation 3.25, if $X = X_{data}$, where X is the discriminator input, and X_{data} is the real data, the discriminator should display a numeric value close to 1 in the output ($D(X) \rightarrow 1$), since the input is the real data, maximizing $V(D, G)$ (maximizing the classification). According to Equation 3.26, if $X = G(Z)$, where X is the input of the discriminator and $G(Z)$ is the fake data generated by the generator, the discriminator has two possible outputs: a close number to 0, classifying the input as fake, or a close number to 1, classifying the input as real. If the discriminator correctly predicts that the input data is fake ($X = G(Z)$), it maximizes $V(D, G)$ (maximizing the classification). If the discriminator misclassifies the input as real, the generator minimizes $V(D, G)$, once the main goal of the generator is to fool the discriminator.

$$\text{if } X = X_{data} \implies D(X) \rightarrow 1 \implies \max_{D} V(D, G) = E_{x \sim p_{data}(x)}[\log(D(x))] \tag{3.25}$$

$$\text{if } X = G(Z) \implies \begin{cases} D(X) \rightarrow 0; \text{for D} \implies \max_D V(D, G) = E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \\ D(X) \rightarrow 1; \text{for G} \implies \min_G V(D, G) = E_{x \sim p_{data}(x)}[\log(D(x))] \end{cases} \tag{3.26}$$

$$\min_{G} \max_{D} V(D, G) = E_{x \sim p_{data}(x)}[\log(D(x))] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.27}$$

From a mathematical point of view, Equation 3.27 shows a minimax game, where G represents the generator, D represents the discriminator, z is the noise vector, x is the real data, p_z is the probability density function of the noise vector, p_{data} is the probability density function of the real data, $D(x)$ represents the probability that x came from the real data, and $G(z)$ represents the probability that z came from the fake data.

Giving the noise vector z, the GAN generates a random output that is indistinguishable from real-world data. From the current network, it is not possible to control the data that is being generated (SAXENA; CAO, 2022). To overcome the problem, Mirza and Osindero (2014) introduced the Conditional Generative Adversarial Network (CGAN). CGANs can generate examples by conditioning the model on additional information to direct the data generation process. The generative adversarial nets can be

extended to a conditional model if the generator and discriminator are conditioned on some extra information y, where y can be any extra information such as class labels or other types of data. Equation 3.28 is an updated version of Equation 3.27 for the CGAN, considering the extra information y.

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log(D(x|y))] + E_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (3.28)$$

3.4.1 Class imbalance problem with GANs

To overcome the class imbalance problem, recent works have used GANs to generate synthetic data to create more samples for the minority class (DOUZAS; BACAO, 2018) (SANTOS; ARANHA, 2019) (CAI et al., 2019b) (MARIANI et al., 2018). Santos and Aranha (2019) presented a study comparing the use of GANs, SMOTE, and ADASYN to synthesize datasets as a solution for two main problems: imbalanced data and to avoid the use of the original data for privacy reasons. Three different datasets were used during the experimentation: Pima Indians Diabetes (SMITH et al., 1988) [2], Breast Cancer Wisconsin (Diagnostic) from UCI machine learning repository [3], and Credit Card Fraud Detection (POZZOLO et al., 2017). All the labels for the datasets used for the experiment are binary, where 0 represents the majority class label, and 1 represents the minority class label. Table 3.3 shows a summary of the datasets.

Table 3.3: Summary of the databases used in the Santos and Aranha (2019) research

Database Name	Number of features	Size	Label Distribution
Pima Indians Diabetes Database	9	768	No diabetes: 500, Diabetes: 268
Breast Cancer Wisconsin (Diagnostic)	32	569	Benign: 357, Malignant: 212
Credit Card Fraud Detection	31	284807	Non-frauds: 284315, Frauds: 492

[2] https://www.kaggle.com/uciml/pima-indians-diabetes-database
[3] http://archive.ics.uci.edu/ml/index.php

The first experiment trained a Decision Tree classifier using only synthetic data generated by the GAN for the Cancer and Diabetes databases. The experiment had three steps: first, GAN was trained with the full original dataset, second, they used the trained GAN to generate a new synthetic dataset with the exact size of the original, third, the Decision Tree was trined using only the synthetic data, and then the classifier was evaluated using the original database. The results showed that the GAN with 2 hidden layers with sizes of 256 and 512 each, outperformed the classifier trained using the original Cancer dataset in accuracy and precision. Using the original dataset, the accuracy and precision for the original data were 0.888 and 0.679, respectively. For the synthetic data generated by the GAN, the accuracy and precision were 0.935 and 0.853, respectively. The classifier trained using the Diabetes dataset with the original data outperformed the classifier trained using the synthetic data generated by the GAN with 2 hidden layers with sizes of 256 and 512 each. The accuracy, precision, and recall for the original data were 0.748, 0.784, and 0.367, respectively. For the synthetic data, the accuracy, precision, and recall were 0.706, 0.582, and 0.584, respectively.

The second experiment trained a Decision Tree classifier using a balanced dataset for Fraud detection. To generate the balanced data, they first separated the dataset on the target classes, generating a database only with class 0, and another one only with class 1. A GAN was trained using only the data from the minority class. The trained GAN was used to generate data for the minority class until the original dataset becomes balanced. For comparison, the classifier was trained using the imbalanced data and the balance data. In both cases, GAN, ADASYN, and SMOTE were applied to over-sampling the minority class. Using the imbalanced dataset, the GAN with 1 hidden layer with a size of 256 outperformed the accuracy (0.986) and precision (0.077) compared with SMOTE, where the accuracy was 0.958 and the precision was 0.026, and ADASYN, where the accuracy was 0.958 and the precision was 0.026. But for the original imbalanced data, without any under-sampling, the classifier outperformed the SMOTE, ADASYN, and the GAN, with an accuracy of 0.999 and precision of 0.896. Using the balanced data generated by the GAN, and using GAN, SMOTE, and ADASYN for under-sampling the minority class, the results were very similar between ADASYN, SMOTE, and the GAN. SMOTE had an accuracy of 0.912, a precision of 0.959, and a recall of 0.861. ADASYN had an accuracy of 0.921, precision of 0.979, and a recall of 0.861. The GAN with 1 hidden layer with a size of 256 had an accuracy of 0.894, precision of 0.998, and recall of 0.789. The balanced dataset with no over-sampling technique had an accuracy of 0.782, precision of 1.0, and

recall of 0.565.

The results showed that the use of GAN to generate synthetic data in the balanced scenario showed better accuracy and precision than training on the original dataset. For the imbalanced scenario, the GAN synthetic data performed better than the original data but did not outperform SMOTE and ADASYN.

Guan and Zhang (2022) presented a study to predict diabetes using genotype SNP data and phenotype data. The dataset is highly skewed with healthy samples with a ratio of 20. For comparison, two sampling techniques were chosen and the data was augmented by GAN. The proposed work has two neural networks, one is a genotype neural network, and the other is a phenotype neural network. For the phenotype neural network, SMOTE was chosen for the under-sampling method, random under-sampling (RUS) for the under-sampling method, and a Spearman correlation was used for feature selection. The genotype neural network predicts diabetes using only SNPs data. The number of features was reduced by filtering the genes through Genome-Wide Association Studies (GWAS) Catalog with features and biomarkers that are known highly related to diabetes. For the resampling, the same techniques used for the phenotype were applied. The results for the phenotype network compared the SMOTE, random under-sampling, and GAN. The under-sampling technique had an accuracy of 0.90, specificity (TNR) of 0.91, and a sensitivity of 0.88. The over-sampling technique had an accuracy of 0.88, specificity (TNR) of 0.94, and a sensitivity of 0.83. The GAN had an accuracy of 0.89, specificity (TNR) of 0.94, and a sensitivity of 0.84. The genotype network performed poorly compared with the phenotype network. The SMOTE had an accuracy of 0.60, specificity of 0.7, and Sensitivity of 0.5. The GAN had an accuracy of 0.55, specificity of 0.7, and sensitivity of 0.5.

The overall results showed that the under-sampling technique has the best performance for the problem proposed by the authors, and the phenotype network could achieve state-of-the-art results. It was also observed that GAN is more suitable to generate continuous data than discrete data.

3.5 Chapter Conclusion

The chapter presented machine learning theoretical basis, such as feature selection, predictive models, class imbalance problems, and Generative Adversarial Networks. The feature selection can be categorized into five categories: filter, wrapper, embedded,

ensemble, and hybrid Ang et al. (2016). The predictive models addressed in the chapter were Random Forest, SVM, and K-Nearest Neighbors.

The imbalance class degrades the performance of machine learning algorithms because the decision-making will be biased toward the majority class (ELRAHMAN; ABRAHAM, 2013) (GUO et al., 2008). Kaur, Pannu and Malhi (2020) split the imbalance class approach into four techniques: pre-processing approaches, cost-sensitive learning methods, algorithmic centered approaches, and hybrid methods. Besides those approaches, new studies have used Generative Adversarial Networks to overcome the problem.

The next chapter will present the studies for human phenotype prediction using SNPs and the proposed problem for this work.

4 HUMAN PHENOTYPE PREDICTION USING SNPS

The use of SNPs to determine pigmentation traits has many studies trying to find a state-of-the-art solution. There is no consensus on the best approach for the problem, and each study tries a different way to solve it (WALSH et al., 2011) (MUNEEB; HENSCHEL, 2021). The SNPs that accurately predict the phenotype is also under discussion. Walsh et al. (2011) developed a tool to predict eye coloration using six SNPs, while Hart et al. (2013) presented a solution for eye coloration using eight SNPs. Studies in the field are still necessary to build a robust solution for the problem.

IrisPlex was developed by Walsh et al. (2011) to predict Blue, Intermediate, and Brown eye colors for forensic use. For the creation of the tool, six SNPs were used: rs12913832, rs1800407, rs12896399, rs16891982, rs1393350 and rs12203592 from the HERC2, OCA2, SLC24A4, SLC45A2, TYR and IRF4 genes, respectively. They used the information of 6168 Dutch Europeans to establish that the six SNPs selected carry the most important eye color information. The IrisPlex presented an AUC (Area Under The Curve) of 0.93 for brown eyes and 0.91 for blue eyes. But, according to the paper, intermediate eye colors were more challenging to define using the presented prediction model and the available SNPs.

Hart et al. (2013) presented a solution for eye color and skin color prediction using 8 SNPs, to improve the 7-Plex system, that utilizes 7 SNPs (rs12913832, rs1545397, rs16891982, rs1426654, rs885479, rs6119471, and rs12203592). The solution adds the rs12896399 SNP. The training set used for them has 803 training samples, and the classes for eye color are Blue, Brown, and Green, while the classes for skin are Dark, Medium, and Light. The process for eye color prediction occurs in two steps: The first step will classify the sample as Not Brown or Not Blue using the rs12913832 SNP. Then, in the second step, it will classify the eye as being Blue, Brown, or Green using the rs12203592, rs16891982, and rs6119471 SNP.

The eye classification occurs according to the alleles in each SNP. Light and medium skin color are predicted by any of the two following alleles: G/G at rs12913832, G/G at rs16891982, A/A at rs1426654, T/T at rs1545397, or A/A at rs885479. Light skin color is predicted by more stringent conditions: G/G at rs12913832, plus G/G at rs16891982, and A/A at rs1426654. Non-light skin color, like medium or dark, is predicted by G/G at rs6119471. Using the European data for test, the call rate for the solution was approximately 94%, and no errors occurred for eye prediction.

Muneeb and Henschel (2021) work presented an experiment to classify the eye color and Type-2 diabetes using 9 types of classifiers: Random Forest, Extreme Gradient boosting, Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (BILSTM), 1D Convolutional Neural Network (1DCNN), ensembles of ANN, and ensembles of LSTM. The dataset used for eye color was randomly split into 540 samples for training and 266 samples for test, maintaining the classes proportion (Brown and Green). Each algorithm was trained using different numbers of SNPs. The results of all models were very close, but in their case, the ensembles of LSTM had the higher accuracy (96%) using 1560 SNPs. It is important to highlight that the ensembles of LSTM trained using 3 SNPs, for their experiment, had an accuracy of 90%, a close result using fewer attributes.

Chaitanya et al. (2018) presented a tool called HIrisPlex-S to predict eye, hair, and skin color. The solution utilizes 41 SNPs, where 17 SNPs are used for skin color, and 24 SNPs are used for eye and hair color. Of these 24 SNPs, 17 also contribute to skin color prediction. The HIrisPlex-S comprises the IrisPlex tool developed previously and has 3 eye, 4 hair, and 5 skin color categories. The eye categories are Blue, Intermediate, and Brown. The hair categories are Blond, Brown, Red, and Black. The skin categories are Very Pale, Pale, Intermediate, Dark, and Dark-Black. The 17 SNPs added in the solution for skin coloration are: rs3114908, rs1800414, rs10756819, rs2238289, rs17128291, rs6497292, rs1129038, rs1667394, rs1126809, rs1470608, rs1426654, rs6119471, rs1545397, rs6059655, rs12441727, rs3212355, and rs8051733. To validate the solution, a comparison with 194 individuals from 17 different populations was made, and the final results showed an Area Under the Curve (AUC) of 0.75 for Very Pale, 0.73 for Pale, 0.75 for Intermediate, 0.84 for Dark, and 0.98 for Dark-Black skin color.

Most of the solutions proposed so far use data collected from Europeans and only a few admixed samples. The goal of this work is to understand what are the most relevant SNPs to determine skin and eye traits to build a machine learning solution for forensics use, using data collected from the Southern Brazilian population, focused on the state of Rio Grande do Sul. To achieve this goal, many experiments were applied in the datasets provided to find the best solution for the proposed problem, dealing with the class imbalance problem and feature selection. Chapter 5 will present all the experiments and results found.

For this work, the data was collected from the Southern Brazilian population (see Sections 4.1 and 4.2), and for the study, it was selected sixty-six SNPs in twenty-one

genes reported in the literature as associated with human pigmentation: ASIP, BNC2, DDB1, EXOC2, HERC2, IRF4, KITLG, LYST, MC1R, MFSD12, MYO5A, NPLOC4, OCA2, SLC24A4, SLC24A5, SLC45A2, TMEM138, TTC3, TYR, TYRP1, UTG1A6 (see Table 4.1). The information related to the gene and chromosome can be found in SNPedia[1], dbSNP[2], GWAS Catalog[3], and Infinome[4].

Table 4.1: The sixty-six SNPs selected for the study of this work

SNP	Gene	Chromosome
rs3768056	LYST	1
rs2070959	UGT1A6, UGT1A7, UGT1A8, UGT1A9, UGT1A10	2
rs16891982, rs28777, rs183671, rs13289	SLC45S2	5
rs12203592	IRF4	6
rs4959270	LOC105374875	6
rs13289810	RNU-47P	9
rs1325127	TYR	9
rs2733832, rs683	LURAP1L-AS1, TYRP1	9
rs10756819	BNC2	9
rs11230664	DDB1	11
rs7948623	TMEM138	11
rs1042602, rs1393350	LOC107984363, TYR	11
rs10777129, rs642742, rs12821256	KITLG	12
rs12896399	LOC105370627	14
rs2402130	SLC24A4	14

[1]https://www.snpedia.com/index.php/SNPedia
[2]https://www.ncbi.nlm.nih.gov/snp/
[3]https://www.ebi.ac.uk/gwas/home
[4]https://www.infino.me/welcome

rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037280, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241	OCA2	15
rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs4932620, rs8039195, rs16950987	HERC2	15
rs1426654	MYEF2, SLC24A5	15
rs1724630	MYO5A	15
rs3212345	LOC101927910	16
rs1805005, rs1805006, rs1110400, rs885479	MC1R	16
rs1805009	MC1R, TUBB3	16
rs9894429	NPLOC4	17
rs10424065	MFSD12	19
rs6119471	ASIP	20
rs2424984	AHCY, ASIP	20
rs2378249	PIGU	20
rs2835630	TTC3	21

Two datasets were provided with information for eye and skin color. Each dataset contains sixty-six SNPs (Table 4.1) and three classes: Blue, Intermediate, and Dark Brown are the classes related to eye color, and White, Intermediate, and Brown are the classes related to skin color. The dataset for eye classification has 653 samples (Table 4.2), where 154 samples were classified as Blue, 158 samples were classified as intermediate

and 341 samples were classified as Dark Brown. The Intermediate class corresponds to green and hazel eyes. The Dark Brown class corresponds to light brown, dark brown, and black eyes.

The dataset for skin classification has 652 samples (Table 4.3), where 467 samples were classified as White, 107 samples were classified as Intermediate, and 78 samples were classified as Brown. The White class corresponds to white and pale skin colors, the Intermediate class corresponds to beige and light brown skin colors, and the Brown class corresponds to medium and dark brown skin colors.

Table 4.2: Data distribution for eye color dataset

Class	**Eye Color**	**Number of samples**
Blue	Blue	467
Intermediate	Green	107
	Hazel	
Dark Brown	Black	78
	Dark Brown	

Table 4.3: Data distribution for skin color dataset

Class	**Skin Color**	**Number of samples**
White	White	467
	Pale	
Intermediate	Beige	107
	Light Brown	
Brown	Medium Brown	78
	Dark Brown	

The data provided for the study is highly skewed, as shown in tables 4.2 and 4.3. For the skin dataset, most samples were classified as White, while fewer were classified as Brown or Intermediate. The same occurs with the eye skin dataset, where most samples were classified as Dark Brown, while fewer were classified as Blue or Intermediate.

The experiments presented in Chapter 5 will focus on finding the SNPs that accurately predict eye and skin color and build a classifier for the proposed problem. The two main challenges of this work are the class imbalance problem and finding the SNPs

that will correcttly classify the three classes for each dataset. As presented in Chapter 3.3, the class imbalance can directly impact the performance of machine learning algorithms. Besides that, as presented in the related works, many solutions do not perform well for the Intermediate classes because is not an easy task to find the right SNPs for this class.

4.1 Samples and Phenotypical Characterization

An informed written consent and a survey form on sex, age, self-declared biogeographical ancestry, place of birth and residency was completed and signed by all voluntary participants. Oral swabs were collected from 653 individuals from Southern Brazilian population. Digital photographs of subjects' eyes were taken and their colors were classified as: Blue (154), Green (100), Hazel (58), Light Brown (80), Dark Brown (187), and Black (74).

Amounts of red (R), green (G), and blue (B) color values were measured by analyzing each photo with COLORS software (OTAKA et al., 2002) to confirm such classification. Eye color categorization was performed according to overall accepted perception for local Southern Brazilian population, and category assignment was independently performed by three different collaborators.

Each participant skin color was identified using the Fitzpatrick Score (Types 1 to 6) and classified independently by at least three collaborators as: White (245), Pale (222), Beige (67), Light Brown (40), Medium Brown (55), or Dark Brown (23). Amounts of RGB color values were measured in an inner and hairless portion of the right arm (below elbow) using a color analyzer equipment ACR-1023 (Instrutherm, Brazil) (PARRA et al., 2003; FITZPATRICK, 1988).

4.2 DNA Genotyping

Genomic DNA from oral swabs was extracted using the NucleoSpin Tissue kit (Macherey-Nagel Inc.) following manufacturer instructions or by organic extraction method (LIU, 2009). Samples were sent to Liggins Institute, from Auckland University, New Zealand, for SNPs genotyping using iPLEX® Pro reagentes on the MassARRAY® system and MassARRAY Typer application software (PROTOCOLS, ; GABRIEL; ZIAUGRA; TABBAA, 2009).

4.3 Chapter Conclusion

The chapter presented studies for human phenotype prediction using SNPs and the proposed problem for this work. Each work presented has different ways to solve the proposed problem, and there is no consensus related to the best approach. The main challenges of this study are class imbalance and finding the SNPs to correctly classify skin and eye color.

The next chapter will present all the experiments performed to find the best approach to balance the classes, the selection of the SNPs, and the classifiers construction.

5 EXPERIMENTS AND RESULTS

The focus of this work is to build a classifier to predict eye and skin color for forensic use, using the data collected from the Southern Brazilian population. The experiments will focus on finding the SNPs to accurately predict the phenotype for each problem and find the best classifier. As presented in Chapter 4, the datasets provided for the study are skewed, and to overcome the problem, different approaches for class imbalance were tested.

To perform the experiments, the datasets were first split into train and test datasets, and the proportion of the train size is 70% of the original size. The train size for the skin dataset is 456 samples, while the test size is 196 samples. For the eye dataset, the train size is 457 samples, and the test size is 196 samples. Table 5.1 and Table 5.2 show the distribution for each class for the train and test set for eye and skin data, respectively.

Table 5.1: Data distribution for eye color dataset for train and test set

Class	**Number of samples in the train set**	**Number of samples in the test set**
Blue	111	43
Intermediate	120	38
Dark Brown	226	115

Table 5.2: Data distribution for skin color dataset for train and test set

Class	**Number of samples in the train set**	**Number of samples in the test set**
White	331	136
Intermediate	76	31
Brown	49	29

All the machine learning models were trained using the same train set and tested using the same test set. For the training and test process, each dataset was prepared using the additive model data encoding (see Section 5.1). Random Forest and Support

Vector Machine were the algorithms selected for the experimentations, based on previous studies (KATSARA et al., 2021) (ZAORSKA; ZAWIERUCHA; NOWICKI, 2019). The method used for Feature Selection was Recursive Feature Elimination, as a wrapper method. Python was used for the experiments and the Scikit-Learn library (version 1.1.2) was used to apply the Machine Learning algorithms. To find the best hyperparameters of each classifier, a Grid Search was applied to test a set of different hyperparameters for each classifier. Also, Leave One Out cross validation model was applied in all training processes. The Grid Search exhaustively searches for the best parameters for an estimator, given a set of predefined parameters to execute the search. The Leave One Out splits the entire train set into train/test sets, where each sample of the entire train set is used at least once as a test set. This approach guarantees the classifiers will be trained using different combinations of the samples, and a better evaluation of the predictor's behavior can be done (FACELI et al., 2011).

For the class imbalance problem, it was compared four different approaches: SMOTE for over-sampling, SMOTE and Edited Nearest Neighbours (SMOTEENN) for hybrid sampling, CNN (Condensed Nearest Neighbor) for under-sampling, and Conditional GAN to generate synthetic data. A Conditional GAN was chosen because it is possible to determine the number of synthetic data necessary to generate as a condition of the model. The library used to train the CGAN was the SDV (Synthetic Data Vault Project) library (version 0.15.0). The SMOTE, SMOTEENN, and CNN algorithms are from the imbalanced-learn library[1] (version 0.9.1). It was also evaluated the performance of the classifiers without applying any feature selection or any class balanced approach. Table 5.3 shows the possible values of parameters tested for each algorithm. For the experiments, E4 for skin classification using the data generated by the CGAN using the SDV encoding, E4 and E7 for skin classification using the CNN, E7 for eye classification using the CNN, E7 for eye classification using the data generated by the CGAN, and E7 for eye classification using the data generated by the CGAN for additive encoding, a small change was made in the parameters for the SVM. Because the experiments were not running due to many values to test in the degree parameter, for those cases, it was necessary to remove the degrees 21 and 27.

For the RFE algorithm, the metric used to be maximized using the features selected was the recall, to minimize the number of samples in the Intermediate classes (for both datasets) misclassified as the other two classes, as well as minimizing the number of

[1] https://imbalanced-learn.org/stable/

samples from the other two classes misclassified as Intermediate. Figures 5.1, 5.2, 5.3, 5.4, and 5.5 show the pipeline for the experiments using SMOTE, SMOTEENN, CNN, CGAN, and with no class balancing, respectively. Tables 5.4, 5.5, 5.6, 5.7, 5.8, and 5.9 shows a summary of all experiments and their respective identification, where SVM-RFE is Recursive Feature Elimination with SVM, and RF-RFE is Recursive Feature Elimination with RF.

Table 5.3: Paremeter optmization values for each algorithm

Algorithm	**Parameter**	**Values**
Random Forest	bootstrap	True, False
	max_depth	10, 50, 80, 90, 100, 110
	class_weight	balanced, balanced_subsample
	n_estimators	50, 100, 200, 300, 1000
	criterion	gini, entropy
SVM	kernel	linear, poly, rbf, sigmoid
	degree	1, 3, 5, 11, 13, 21, 27
	gamma	scale, auto
	class_weight	balanced
	decision_function_shape	ovo, ovr
RFECV	estimator	Random Forest, SVM
	step	1
	cv	Leave One Out
	scoring	Recall
	min_features_to_select	1
GridSearchCV	cv	Leave One Out

Table 5.4: Summary of the experiments with SMOTE

ID	**SMOTE**
E1	Feature Selection: SVM-RFE Classifier: RF with Grid Search
E2	Feature Selection: SVM-RFE Classifier: RF without Grid Search

E3	Feature Selection: SVM-RFE Classifier: SVM without Grid Search
E4	Feature Selection: SVM-RFE Classifier: SVM with Grid Search
E5	Feature Selection: RF-RFE Classifier: RF with Grid Search
E6	Feature Selection: RF-RFE Classifier: RF without Grid Search
E7	Feature Selection: RF-RFE Classifier: SVM with Grid Search
E8	Feature Selection: RF-RFE Classifier: SVM without Grid Search
E9	Feature Selection: Not applied Classifier: RF with Grid Search
E10	Feature Selection: Not applied Classifier: RF without Grid Search
E11	Feature Selection: Not applied Classifier: SVM with Grid Search
E12	Feature Selection: Not applied Classifier: SVM without Grid Search

Table 5.5: Summary of the experiments with SMOTEEN

ID	**SMOTEEN**
E1	Feature Selection: SVM-RFE Classifier: RF with Grid Search
E2	Feature Selection: SVM-RFE Classifier: RF without Grid Search
E3	Feature Selection: SVM-RFE Classifier: SVM without Grid Search
E4	Feature Selection: SVM-RFE Classifier: SVM with Grid Search

E5	Feature Selection: RF-RFE Classifier: RF with Grid Search
E6	Feature Selection: RF-RFE Classifier: RF without Grid Search
E7	Feature Selection: RF-RFE Classifier: SVM with Grid Search
E8	Feature Selection: RF-RFE Classifier: SVM without Grid Search
E9	Feature Selection: Not applied Classifier: RF with Grid Search
E10	Feature Selection: Not applied Classifier: RF without Grid Search
E11	Feature Selection: Not applied Classifier: SVM with Grid Search
E12	Feature Selection: Not applied Classifier: SVM without Grid Search

Table 5.6: Summary of the experiments with CNN

ID	CNN
E1	Feature Selection: SVM-RFE Classifier: RF with Grid Search
E2	Feature Selection: SVM-RFE Classifier: RF without Grid Search
E3	Feature Selection: SVM-RFE Classifier: SVM without Grid Search
E4	Feature Selection: SVM-RFE Classifier: SVM with Grid Search
E5	Feature Selection: RF-RFE Classifier: RF with Grid Search
E6	Feature Selection: RF-RFE Classifier: RF without Grid Search

E7	Feature Selection: RF-RFE Classifier: SVM with Grid Search
E8	Feature Selection: RF-RFE Classifier: SVM without Grid Search
E9	Feature Selection: Not applied Classifier: RF with Grid Search
E10	Feature Selection: Not applied Classifier: RF without Grid Search
E11	Feature Selection: Not applied Classifier: SVM with Grid Search
E12	Feature Selection: Not applied Classifier: SVM without Grid Search

Table 5.7: Summary of the experiments with no class balancing applied

ID	No class balancing applied
E1	Feature Selection: SVM-RFE Classifier: RF with Grid Search
E2	Feature Selection: SVM-RFE Classifier: RF without Grid Search
E3	Feature Selection: SVM-RFE Classifier: SVM without Grid Search
E4	Feature Selection: SVM-RFE Classifier: SVM with Grid Search
E5	Feature Selection: RF-RFE Classifier: RF with Grid Search
E6	Feature Selection: RF-RFE Classifier: RF without Grid Search
E7	Feature Selection: RF-RFE Classifier: SVM with Grid Search
E8	Feature Selection: RF-RFE Classifier: SVM without Grid Search

E9	Feature Selection: Not applied Classifier: RF with Grid Search
E10	Feature Selection: Not applied Classifier: RF without Grid Search
E11	Feature Selection: Not applied Classifier: SVM with Grid Search
E12	Feature Selection: Not applied Classifier: SVM without Grid Search

Table 5.8: Summary of the experiments and CGAN with the SDV encoding

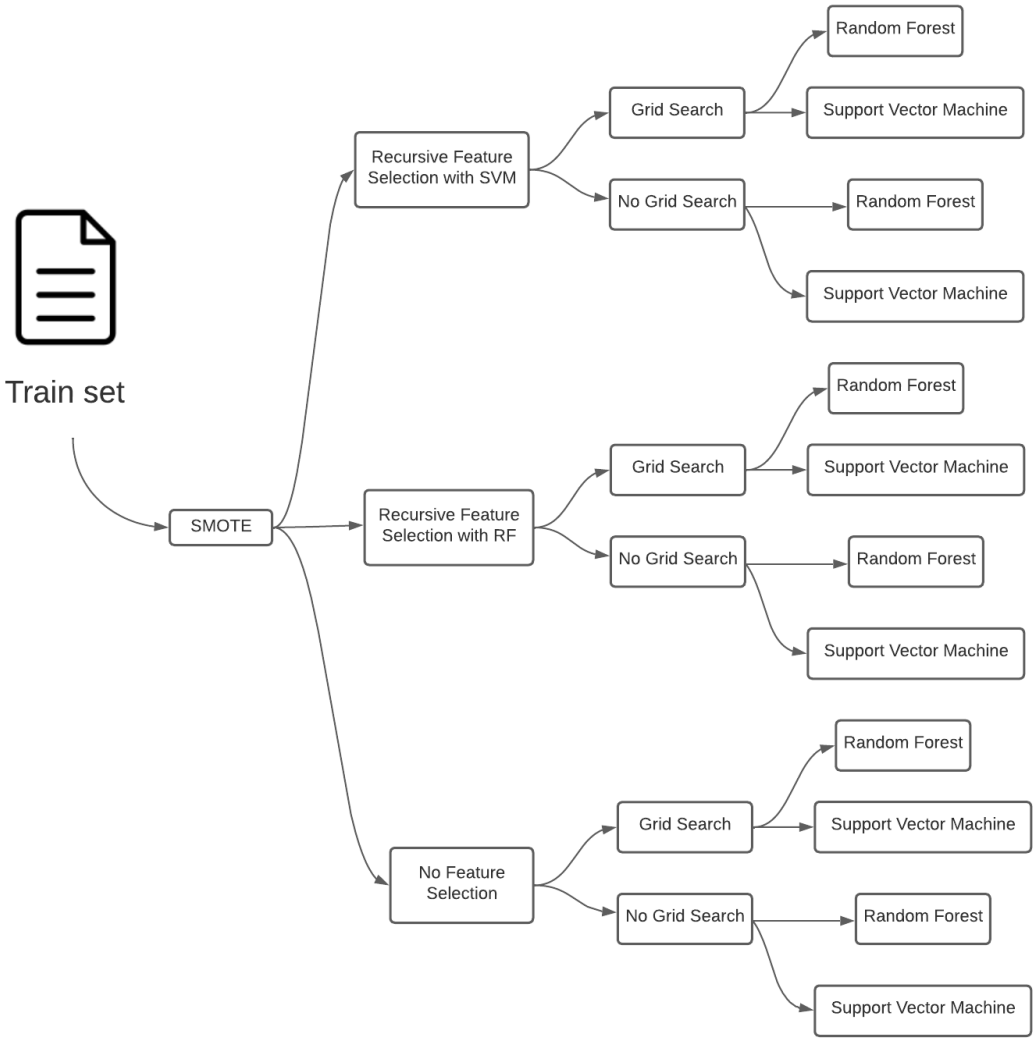
ID	**CGAN with the SDV encoding**
E1	Feature Selection: SVM-RFE Classifier: RF with Grid Search
E2	Feature Selection: SVM-RFE Classifier: RF without Grid Search
E3	Feature Selection: SVM-RFE Classifier: SVM without Grid Search
E4	Feature Selection: SVM-RFE Classifier: SVM with Grid Search
E5	Feature Selection: RF-RFE Classifier: RF with Grid Search
E6	Feature Selection: RF-RFE Classifier: RF without Grid Search
E7	Feature Selection: RF-RFE Classifier: SVM with Grid Search
E8	Feature Selection: RF-RFE Classifier: SVM without Grid Search
E9	Feature Selection: Not applied Classifier: RF with Grid Search
E10	Feature Selection: Not applied Classifier: RF without Grid Search

E11	Feature Selection: Not applied Classifier: SVM with Grid Search
E12	Feature Selection: Not applied Classifier: SVM without Grid Search

Table 5.9: Summary of the experiments with CGAN and the SDV encoding

ID	CGAN with the SDV encoding
E1	Feature Selection: SVM-RFE Classifier: RF with Grid Search
E2	Feature Selection: SVM-RFE Classifier: RF without Grid Search
E3	Feature Selection: SVM-RFE Classifier: SVM without Grid Search
E4	Feature Selection: SVM-RFE Classifier: SVM with Grid Search
E5	Feature Selection: RF-RFE Classifier: RF with Grid Search
E6	Feature Selection: RF-RFE Classifier: RF without Grid Search
E7	Feature Selection: RF-RFE Classifier: SVM with Grid Search
E8	Feature Selection: RF-RFE Classifier: SVM without Grid Search
E9	Feature Selection: Not applied Classifier: RF with Grid Search
E10	Feature Selection: Not applied Classifier: RF without Grid Search
E11	Feature Selection: Not applied Classifier: SVM with Grid Search
E12	Feature Selection: Not applied Classifier: SVM without Grid Search

Figure 5.1: The pipeline for the experiment using SMOTE. Table 5.4 shows the experiment's details.



The metrics collected for each class using the classifiers outputs are the Precision (Equation 3.10), Recall (Equation 3.11) and F1-Score (Equation 3.12) (see Section 3.3, Subsection 3.3.2). The following sections will present the data preparation for the experiments, the training proccess of the CGAN for both datasets, and the final results for eye and skin classification.

5.1 Data Preparation

In the work presented by Mittag, Römer and Zell (2015), the experiment consisted of the preparation of the SNPs and the evaluation of predictive models. The SNPsk

Figure 5.2: The pipeline for the experiment using SMOTEENN. Table 5.5 shows the experiment's details.

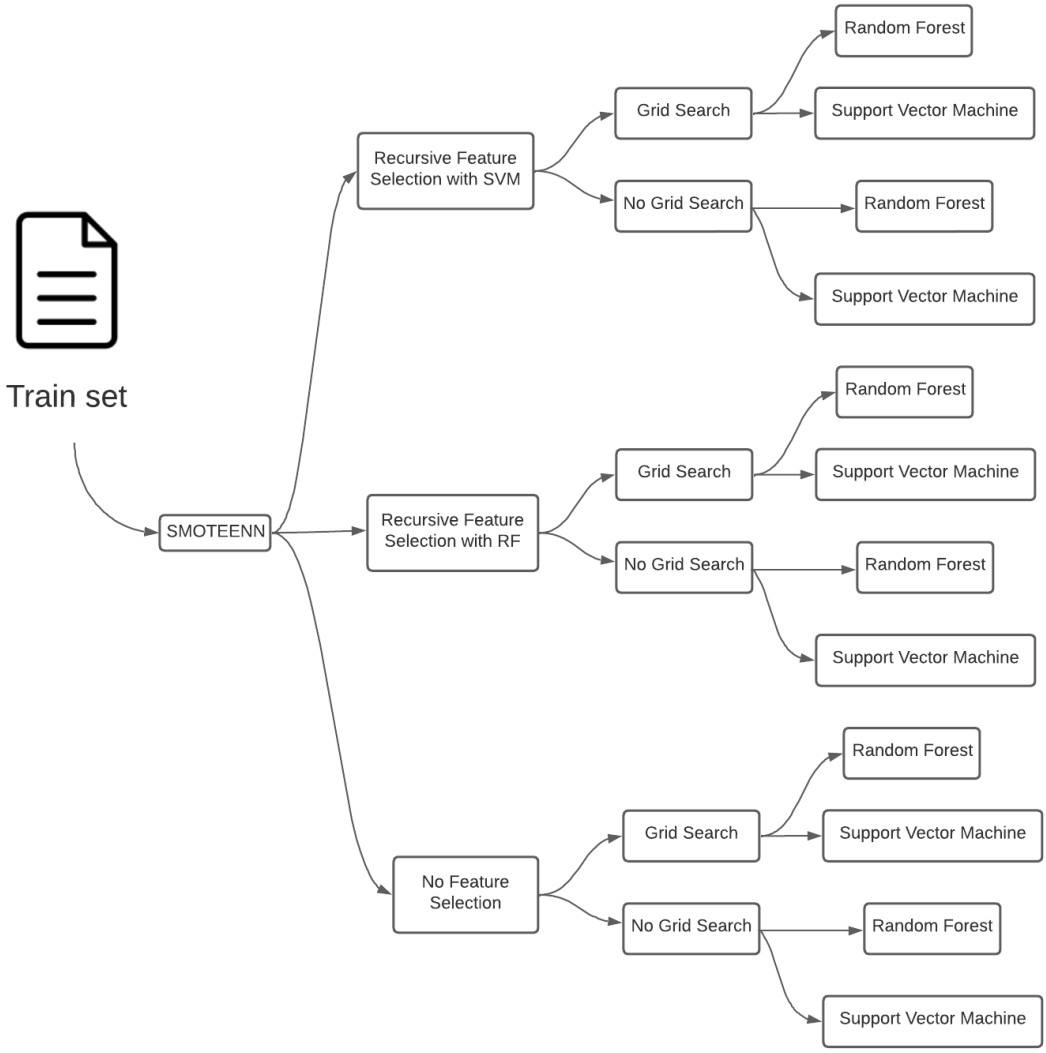


Figure 5.3: The pipeline for the experiment using CNN. Table 5.6 shows the experiment's details.

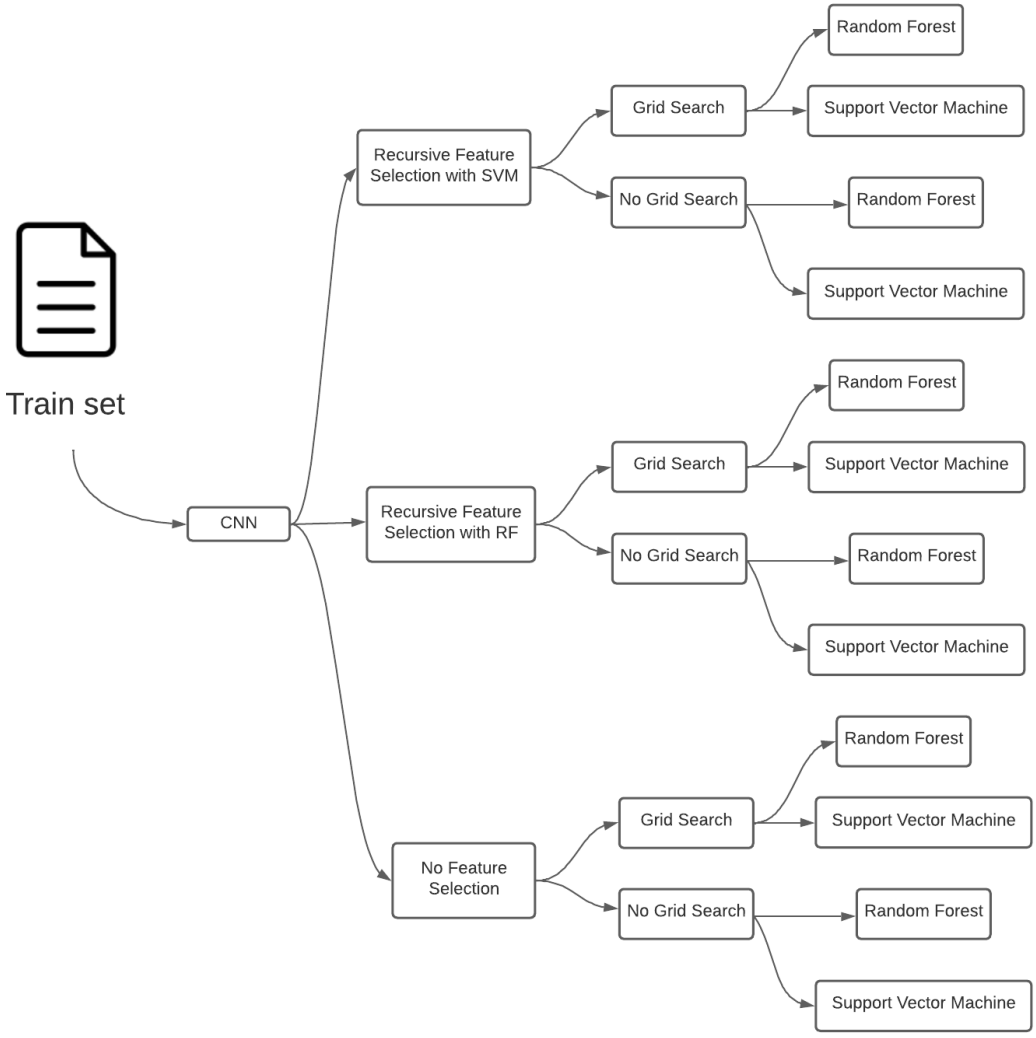


Figure 5.4: The pipeline for the experiment using CGAN. Tables 5.8 and 5.9 show the experiment's details.

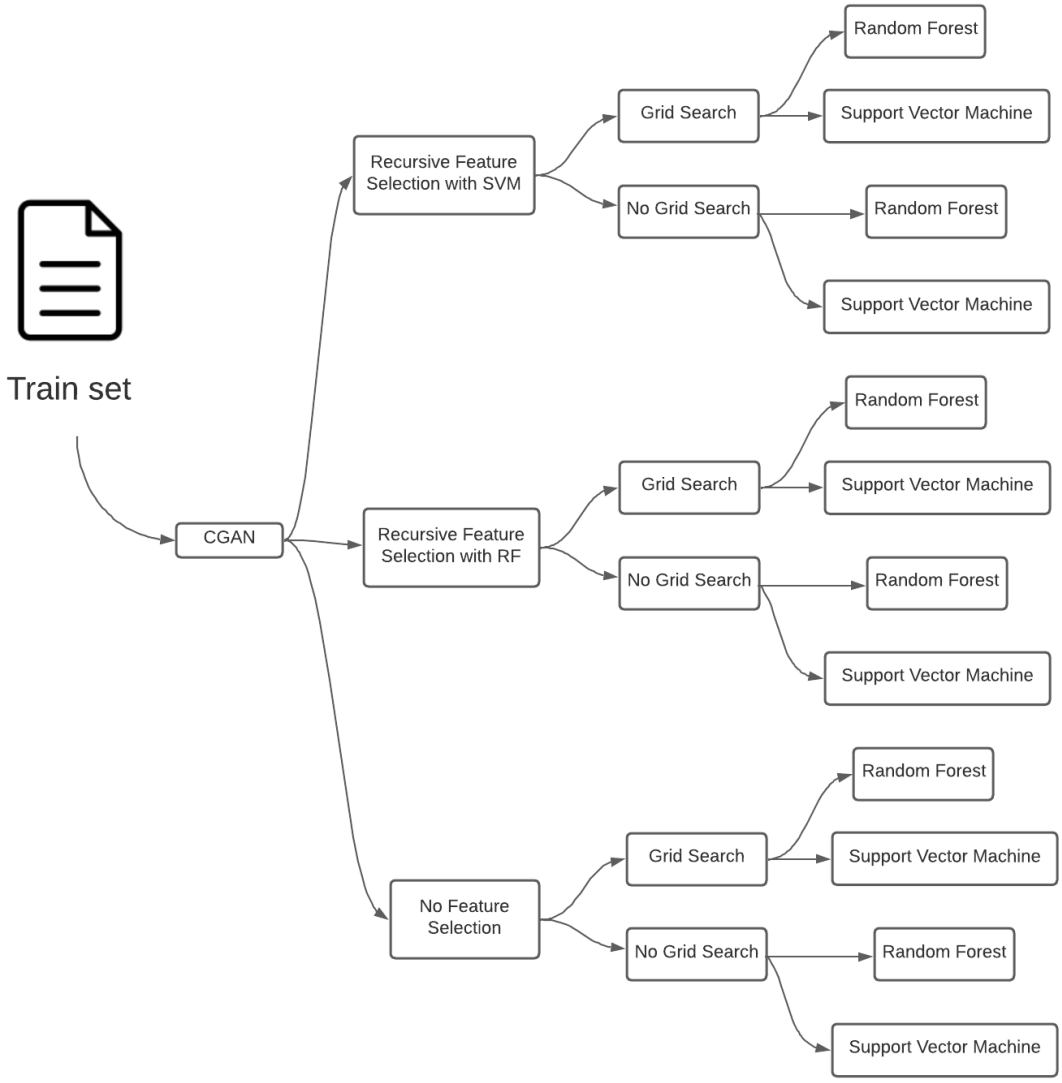
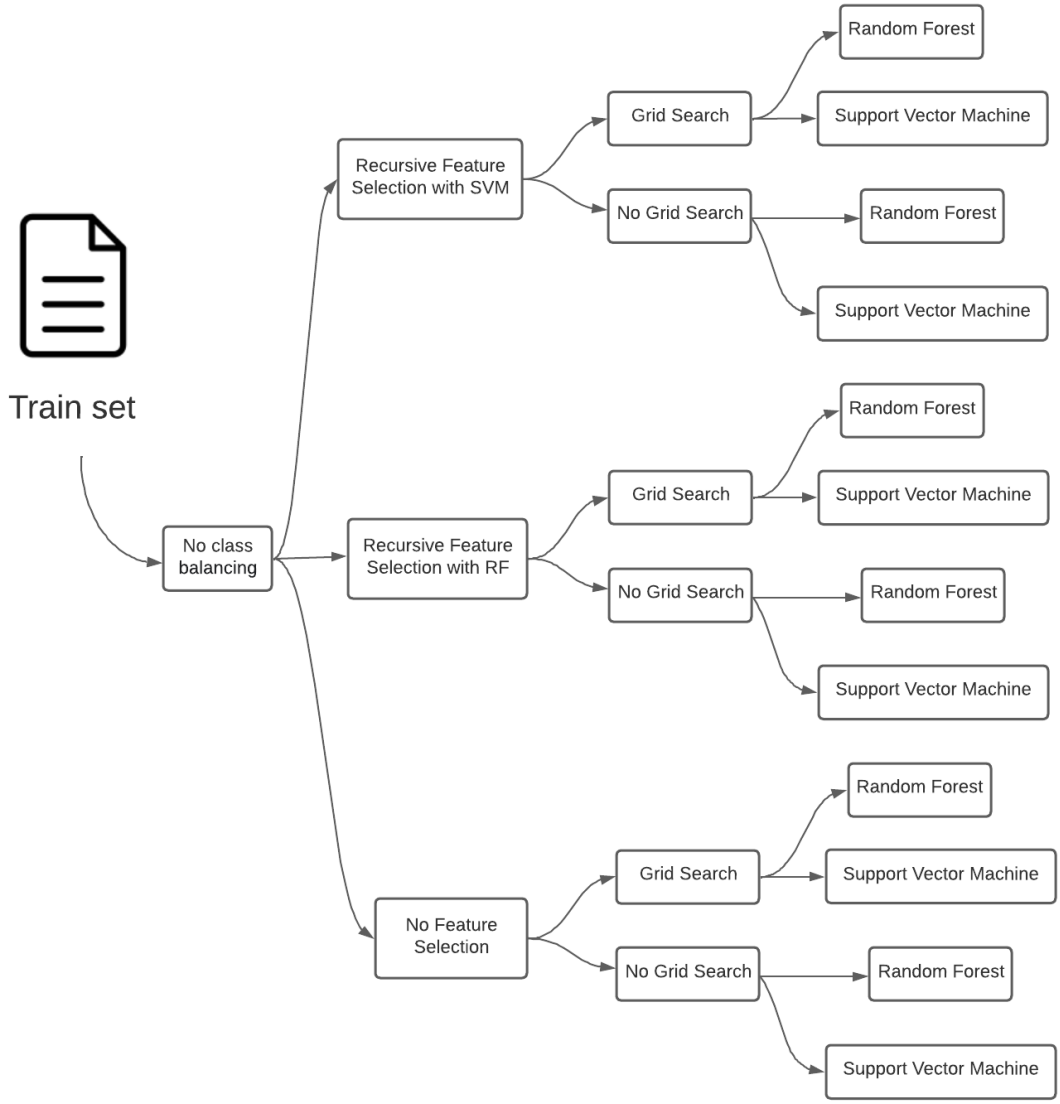


Figure 5.5: The pipeline for the experiment without any class balancing. Table 5.7 shows the experiment's details.



were prepared by the authors in three different ways: Additive model, One-hot encoding (called genotypic by the authors), and Recessive/dominant model. The algorithms were trained using the three different data preparation at a time, to determine what data preparation improves the classification. The algorithms used by the experiment were Decision Tree, Random Forest, SVM (linear), SVM (RBF), kNN, Multilayer Perceptron, and LVQ (Learning vector quantization). The datasets used for the experiment was bipolar disorder (BD), Crohn's disease (CD), coronary heart disease (CAD), hypertension (HT), rheumatoid arthritis (RA), type 1 and type 2 diabetes (T1D and T2D).

The additive model consists of encoding the SNP as a single numeric feature that reflects the number of minor alleles. Homozygous major, heterozygous and homozygous minor is encoded as 0, 1, and 2, respectively. As an example, an SNP with possible genotypes being AA, AB, and BB are encoded as 0, 1, and 2, respectively. The recessive/dominant model creates two columns for each SNP, where each column represents an allele. The column of each allele is set to 0 if the corresponding allele is not presented and set to 1 if it is presented. The last one is the One-hot encoding, where the alleles are represented in a binary way. This model creates a column for each possible genotype (three new columns) for each SNP. The genotype is set to 1 if is presented and it is set to 0 if it is not presented. Figure 5.6 show the representation additive encoding (Add count), recessive/dominant (Rec) encoding, and genotypic (Gen) encoding.

Figure 5.6: Illustration of the three different encoding schemes for SNP data.

SNP_i	Add count	Rec		Gen		
		A	B	AA	AB	BB
AA	0	1	0	1	0	0
AB	1	1	1	0	1	0
BB	2	0	1	0	0	1

Source: (MITTAG; RöMER; ZELL, 2015), p. 4

After training the machine learning algorithms using the three different data preparation. The maximum and the average area under the curve (AUC) were compared for the three different encodings for each dataset, and the additive had the highest values for AUC. The final result showed that additive encoding has an advantage in terms of predictive performance. Figure 5.7 shows the maximum and average AUCs for different encodings grouped by data set.

Considering the results from the study of Mittag, Römer and Zell (2015), the data for the experiments of this work was prepared using the additive model.

Figure 5.7: Maximum and average AUCs for different encodings grouped by data set.

AUC		BD	CAD	CD	HT	RA	T1D	T2D
Max	Add	0.5834	0.5843	0.6178	0.5617	0.7276	0.8682	0.5861
	Rec	0.5752	0.5753	0.6158	0.5517	0.7191	0.8558	0.5837
	Gen	0.5752	0.5771	0.6080	0.5559	0.7167	0.8521	0.5773
Avg	Add	0.5383	0.5571	0.5793	0.5226	0.6736	0.8031	0.5579
	Rec	0.5360	0.5551	0.5748	0.5210	0.6670	0.7931	0.5565
	Gen	0.5365	0.5549	0.5731	0.5214	0.6655	0.7887	0.5559

Source: (MITTAG; RöMER; ZELL, 2015), p.10

5.2 CGAN training process

The CGAN was trained using the SDV library with two different approaches: training the network without any data preparation and training the data using the additive model data encoding. The SDV already has its way to encode categorical data (one-hot encoding), but, as presented in Section 5.1, the encoding of the SNPs can directly affect the algorithm's performance. To verify what was the best method for the CGAN to learn the distribution of the real data, both approaches were tested. Figure 5.8 shows the skin data distribution for real data, represented by the dots in green, and synthetic data, represented by the dots in red. The figure on the left shows the distribution using the SDV encode process, and the figure on the right shows the data distribution using an additive model. Principal Component Analysis (PCA) was used to plot the data distribution. PCA's goal is to extract important information from the table and to display the pattern of similarity of the observations (ABDI; WILLIAMS, 2010). Looking at both pictures, it seems that both ways learned the same data pattern, even though the data generated by the network trained using the additive encoding has more data slightly centered to the left.

Figure 5.8: Skin data distribution generated by the CGAN. The figure on the left shows the distribution using the SDV encode process. The figure on the right show the data distribution using additive model.

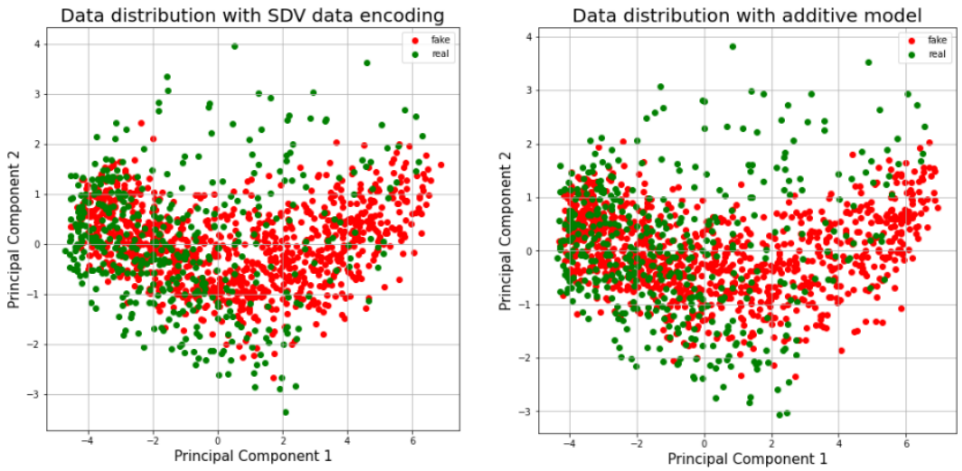
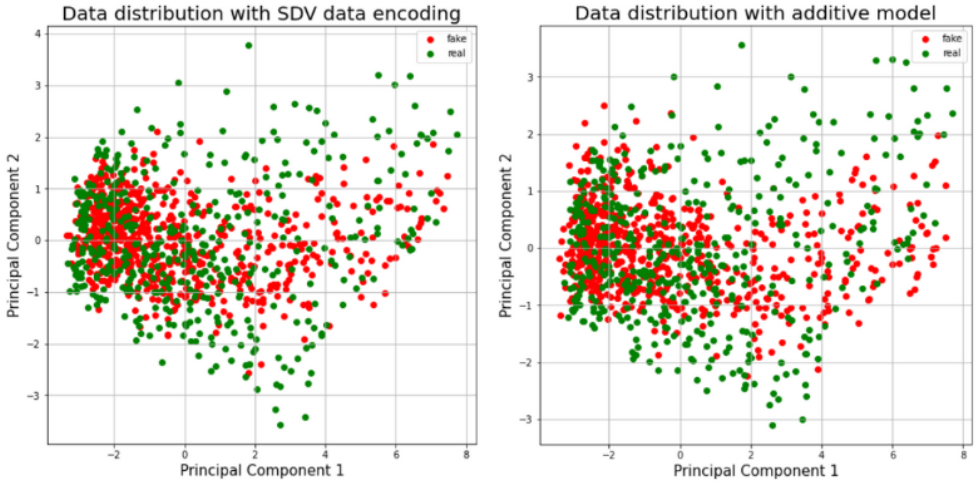


Figure 5.9 shows the eye data distribution for real data, represented by the dots in green, and synthetic data, represented by the dots in red. The figure on the left shows the distribution using the SDV encode process, and the figure on the right shows the data distribution using an additive model.

Figure 5.9: Eye data distribution generated by the CGAN. The figure on the left shows the distribution using the SDV encode process. The figure on the right show the data distribution using additive model.



To evaluate the data synthesized for skin and eye, it was used the evaluation metric provided by the SDV library. The metric provides a score from 0 to 1 to represent how similar the real data and the synthetic data are, being 0 the worst and 1 the best possible score. For the skin data generated, the score using the SDV encoding method was 0.89, and for the additive model was 0.77. For the eye data generated, the score using the SDV encoding method was 0.86, and for the additive model was 0.73.

As a condition of the GAN, 331 samples were generated for White, Intermediate, and Brown skin color. And 226 samples were generated for Blue, Intermediate, and Dark Brown eye colors. The final hyperparameters chosen for each model are described in Table 5.10. To find the best hyperparameters, a manual test was applied to change the values of the parameters, and the best ones had the highest evaluation score. The SDV does not provide something similar to Grid Search, so manual work was necessary.

Table 5.10: Hyperparameters used to train the CGAN

Dataset	Model using the SDV encoding method	Model using additive encoding method
Skin	epochs=3000, batch_size=100, log_frequency=False, generator_lr=0.00001, discriminator_lr=0.00002, generator_decay=0.001, discriminator_decay=0.002, generator_dim=(256, 256, 256), discriminator_dim=(256, 256, 256), embedding_dim=256, discriminator_steps=5	epochs=3000, batch_size=100, log_frequency=False, generator_lr=0.00001, discriminator_lr=0.00002, generator_decay=0.001, discriminator_decay=0.002, generator_dim=(512, 512, 512), discriminator_dim=(256, 256, 256), embedding_dim=256, discriminator_steps=5
Eye	epochs=3000, batch_size=100, log_frequency=False, generator_lr=0.00001, discriminator_lr=0.00002, generator_decay=0.001, discriminator_decay=0.002, generator_dim=(256, 256, 256), discriminator_dim=(256, 256, 256), embedding_dim=256, discriminator_steps=5	epochs=3000, batch_size=100, log_frequency=False, generator_lr=0.00002, discriminator_lr=0.00001, generator_decay=0.0002, discriminator_decay=0.001, generator_dim=(256, 256, 256), discriminator_dim=(256, 256, 256), embedding_dim=256, discriminator_steps=5

5.3 Eye color prediction results

For the experiment, we ran 3 replicates for each classifier. Table 5.11, Table 5.12, Table 5.13, Table 5.14, Table 5.15, and Table 5.16 show the metrics (average and standard deviation) for the overall results. Table 5.17, Table 5.18, Table 5.19, Table 5.20, Table 5.21, and Table 5.22, and Table 5.32 show the SNPs selected in each experiment.

All the experiments had poor performance for the Intermediate class, but the one using CNN had a very bad precision and recall in almost all combinations. It was also observed that a very different number and set of SNPs were selected when the class balance approach changed. For cases like SMOTE, SMOTEENN, and the CGANs, a large number of SNPs were selected, while in the CNN approach, only a few were chosen.

The best result for eye classification was achieved by the experiment without any class balancing (E11) using all the 66 SNPs. The E11 experiment does not use feature selection and the classifier was an SVM with GridSearch (Table 5.7). The precision and recall, on average, respectively were 0.76 and 0.79 for Blue, 0.41 and 0.63 for Intermediate, and 0.92 and 0.75 for Dark Brown. The experiment E1 without any class balancing had a very close result for Blue and Dark Brown classification, using only 4 SNPs (rs6497271, rs12913832, rs1426654, rs1805006), but the performance for the Intermediate was poor. The experiment E1 used as feature selection an SVM-RFE and an RF as the classifier (with GridSeach). The reason why using 4 SNPs had a very close result for Blue and Dark Brown happens it could be the fact that the SNP rs12913832 is directly linked to blue and brown eyes, and the SNP rs1426654 influences the skin pigmentation, indicating light-skinned West Eurasian ancestry. The information related to SNPs can be found in SNPedia[2]. The experiment E5 using SMOTE had also a close result using 57 SNPs. The experiment E5 uses as feature selection an RF-RFE and an RF (with GridSearch) as the classifier. The precision and recall, on average, respectively were 0.74 and 0.67 for Blue, 0.4 and 0.5 for Intermediate, and 0.86 and 0.82 for Dark Brown.

Figure 5.10, Figure 5.11, Figure 5.12, Figure 5.13, Figure 5.14, and Figure 5.15 show the confusion matrix for all classifiers. The columns of the confusion matrix are samples that were predicted by the classifier, and the rows show the actual samples of each class. The main diagonal of the matrix shows the number of samples correctly predicted by the model. Looking at the matrices, it is possible to notice that most of the classifiers confuse the Intermediate class with Blue and Dark Brown. The labels 1AZ, 2V3M, and 4CC5CE6PR represent Blue, Intermediate, and Dark Brown, respectively.

The performance of the classifiers using the data generated by the CGAN had bad performance. A hypothesis is that the CGAN training should be improved, maybe by using other hyperparameters or using more data for the training process. Because the results for both types of encodings were very similar, it is not possible to determine which encoding is the best approach to train the CGAN, but it is important to highlight the study

[2]https://www.snpedia.com/index.php/SNPedia

presented in Section 5.1 about the importance of the SNPs encoding.

Table 5.11: Results for eye classification using SMOTE

ID	Class	Precision	Recall	F1
E1	Blue	0.72±0.01	0.65±0.04	0.69±0.03
	Intermediate	0.39±0.03	0.47±0.05	0.43±0.04
	Dark Brown	0.85±0.01	0.82±0.00	0.84±0.00
E2	Blue	0.69±0.03	0.63±0.02	0.66±0.03
	Intermediate	0.36±0.03	0.45±0.05	0.40±0.03
	Dark Brown	0.85±0.00	0.83±0.00	0.84±0.00
E3	Blue	0.72±0.00	0.65±0.00	0.68±0.00
	Intermediate	0.33±0.00	0.47±0.00	0.39±0.00
	Dark Brown	0.85±0.00	0.77±0.00	0.80±0.00
E4	Blue	0.68±0.00	0.65±0.00	0.67±0.00
	Intermediate	0.29±0.00	0.39±0.00	0.33±0.00
	Dark Brown	0.83±0.00	0.75±0.00	0.79±0.00
E5	Blue	**0.74**±0.03	**0.67**±0.02	**0.71**±0.02
	Intermediate	**0.40**±0.00	**0.50**±0.01	**0.44**±0.00
	Dark Brown	**0.86**±0.01	**0.82**±0.00	**0.83**±0.00
E6	Blue	0.73±0.05	0.70±0.06	0.71±0.05
	Intermediate	0.38±0.01	0.50±0.04	0.43±0.02
	Dark Brown	0.86±0.01	0.80±0.02	0.84±0.01
E7	Blue	0.65±0.00	0.65±0.00	0.65±0.00
	Intermediate	0.36±0.00	0.45±0.00	0.40±0.00
	Dark Brown	0.83±0.00	0.77±0.00	0.80±0.00
E8	Blue	0.72±0.00	0.67±0.00	0.70±0.00
	Intermediate	0.35±0.00	0.5±0.00	0.41±0.00
	Dark Brown	0.86±0.00	0.77±0.00	0.81±0.00
E9	Blue	0.73±0.02	0.67±0.02	0.71±0.02
	Intermediate	0.38±0.01	0.47±0.02	0.42±0.01
	Dark Brown	0.85±0.00	0.83±0.01	0.84±0.00
E10	Blue	0.74±0.00	0.67±0.02	0.71±0.01
	Intermediate	0.38±0.01	0.47±0.00	0.42±0.01
	Dark Brown	0.85±0.00	0.82±0.01	0.84±0.00

E11	Blue	0.63±0.00	0.6±0.00	0.62±0.00
	Intermediate	0.31±0.00	0.42±0.00	0.36±0.00
	Dark Brown	0.83±0.00	0.75±0.00	0.79±0.00
E12	Blue	0.71±0.00	0.7±0.00	0.71±0.00
	Intermediate	0.36±0.00	0.5±0.00	0.42±0.00
	Dark Brown	0.87±0.00	0.77±0.00	0.81±0.00

Table 5.12: Results for eye classification using SMOTEENN

ID	Class	Precision	Recall	F1
E1	Blue	0.67±0.02	0.60±0.05	0.63±0.03
	Intermediate	0.25±0.01	0.63±0.01	0.36±0.01
	Dark Brown	0.96±0.00	0.48±0.01	0.64±0.00
E2	Blue	0.67±0.02	0.65±0.01	0.66±0.01
	Intermediate	0.25±0.00	0.63±0.01	0.36±0.00
	Dark Brown	0.97±0.00	0.49±0.02	0.65±0.01
E3	Blue	**0.69**±0.00	**0.77**±0.00	**0.73**±0.00
	Intermediate	**0.28**±0.00	**0.66**±0.00	**0.39**±0.00
	Dark Brown	**0.95**±0.00	**0.48**±0.00	**0.64**±0.00
E4	Blue	0.65±0.00	0.60±0.00	0.63±0.00
	Intermediate	0.25±0.00	0.66±0.00	0.36±0.00
	Dark Brown	0.96±0.00	0.47±0.00	0.63±0.00
E5	Blue	0.64±0.00	0.65±0.02	0.64±0.01
	Intermediate	0.27±0.00	0.63±0.01	0.38±0.01
	Dark Brown	0.98±0.01	0.51±0.02	0.67±0.05
E6	Blue	0.67±0.01	0.70±0.01	0.67±0.01
	Intermediate	0.26±0.00	0.63±0.01	0.37±0.00
	Dark Brown	0.98±0.01	0.50±0.01	0.66±0.01
E7	Blue	0.64±0.00	0.70±0.00	0.67±0.00
	Intermediate	0.24±0.00	0.55±0.00	0.33±0.00
	Dark Brown	0.92±0.00	0.48±0.00	0.63±0.00
E8	Blue	0.67±0.00	0.74±0.00	0.70±0.00
	Intermediate	0.29±0.00	0.68±0.00	0.40±0.00

	Dark Brown	0.95±0.00	0.47±0.00	0.63±0.00
E9	Blue	0.62±0.03	0.67±0.02	0.64±0.03
	Intermediate	0.26±0.00	0.63±0.02	0.37±0.00
	Dark Brown	0.97±0.00	0.50±0.02	0.66±0.02
E10	Blue	0.64±0.00	0.67±0.04	0.66±0.01
	Intermediate	0.26±0.00	0.66±0.03	0.38±0.01
	Dark Brown	0.98±0.00	0.48±0.00	0.64±0.00
E11	Blue	0.64±0.00	0.67±0.00	0.66±0.00
	Intermediate	0.23±0.00	0.55±0.00	0.33±0.00
	Dark Brown	0.92±0.00	0.48±0.00	0.63±0.00
E12	Blue	**0.67**±0.00	**0.77**±0.00	**0.72**±0.00
	Intermediate	**0.29**±0.00	**0.68**±0.00	**0.41**±0.00
	Dark Brown	**0.95**±0.00	**0.48**±0.00	**0.64**±0.00

Table 5.13: Results for eye classification using CNN

ID	Class	Precision	Recall	F1
E1	Blue	0.71±0.00	0.81±0.00	0.76±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.78±0.00	0.97±0.00	0.86±0.00
E2	Blue	0.71±0.00	0.81±0.00	0.76±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.77±0.00	0.96±0.01	0.86±0.00
E3	Blue	0.71±0.00	0.81±0.00	0.76±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.78±0.00	0.99±0.00	0.87±0.00
E4	Blue	0.71±0.00	0.81±0.00	0.76±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.78±0.00	0.99±0.00	0.87±0.00
E5	Blue	0.70±0.00	0.81±0.00	0.75±0.00
	Intermediate	0.19±0.00	0.18±0.00	0.19±0.00
	Dark Brown	0.81±0.00	0.77±0.00	0.79±0.00
E6	Blue	0.70±0.00	0.81±0.00	0.75±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00

	Dark Brown	0.78±0.00	0.99±0.00	0.87±0.00
E7	Blue	0.70±0.00	0.81±0.00	0.75±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.78±0.00	0.99±0.00	0.87±0.00
E8	Blue	0.70±0.00	0.81±0.00	0.75±0.00
	Intermediate	0.19±0.00	0.18±0.00	0.19±0.00
	Dark Brown	0.81±0.00	0.77±0.00	0.79±0.00
E9	Blue	0.69±0.01	0.81±0.02	0.74±0.01
	Intermediate	0.38±0.02	0.34±0.04	0.36±0.03
	Dark Brown	0.85±0.00	0.86±0.02	0.86±0.01
E10	Blue	**0.69**±0.00	**0.79**±0.03	**0.73**±0.01
	Intermediate	**0.46**±0.02	**0.34**±0.03	**0.39**±0.03
	Dark Brown	**0.87**±0.00	**0.88**±0.00	**0.87**±0.00
E11	Blue	0.67±0.00	0.72±0.00	0.70±0.00
	Intermediate	0.41±0.00	0.39±0.00	0.40±0.00
	Dark Brown	0.86±0.00	0.84±0.00	0.85±0.00
E12	Blue	0.58±0.00	0.88±0.00	0.70±0.00
	Intermediate	0.56±0.00	0.24±0.00	0.33±0.00
	Dark Brown	0.87±0.00	0.86±0.00	0.86±0.00

Table 5.14: Results for eye classification without class balancing

ID	Class	Precision	Recall	F1
E1	Blue	0.70±0.00	0.81±0.00	0.75±0.00
	Intermediate	0.28±0.00	0.45±0.00	0.35±0.00
	Dark Brown	0.90±0.00	0.67±0.00	0.77±0.00
E2	Blue	0.71±0.00	0.81±0.00	0.76±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.78±0.00	0.99±0.00	0.87±0.00
E3	Blue	0.71±0.00	0.81±0.00	0.76±0.00
	Intermediate	0.00±0.00	0.00±0.00	0.00±0.00
	Dark Brown	0.78±0.00	0.99±0.00	0.87±0.00
E4	Blue	0.70±0.00	0.81±0.00	0.75±0.00

	Intermediate	0.28±0.00	0.45±0.00	0.35±0.00
	Dark Brown	0.90±0.00	0.67±0.00	0.77±0.00
E5	Blue	0.70±0.01	0.70±0.04	0.70±0.01
	Intermediate	0.37±0.06	0.34±0.05	0.37±0.03
	Dark Brown	0.84±0.00	0.85±0.04	0.85±0.01
E6	Blue	0.72±0.00	0.67±0.02	0.70±0.01
	Intermediate	0.41±0.05	0.34±0.06	0.37±0.06
	Dark Brown	0.83±0.02	0.90±0.00	0.86±0.01
E7	Blue	0.65±0.00	0.72±0.00	0.68±0.00
	Intermediate	0.35±0.00	0.53±0.00	0.42±0.00
	Dark Brown	0.88±0.00	0.7±0.00	0.78±0.00
E8	Blue	0.72±0.00	0.77±0.00	0.74±0.00
	Intermediate	0.34±0.00	0.29±0.00	0.31±0.00
	Dark Brown	0.83±0.00	0.85±0.00	0.84±0.00
E9	Blue	0.71±0.01	0.68±0.02	0.70±0.00
	Intermediate	0.40±0.04	0.38±0.05	0.39±0.04
	Dark Brown	0.84±0.00	0.87±0.01	0.85±0.00
E10	Blue	0.72±0.01	0.72±0.02	0.72±0.02
	Intermediate	0.37±0.04	0.29±0.04	0.32±0.04
	Dark Brown	0.83±0.01	0.89±0.01	0.86±0.00
E11	Blue	**0.76**±0.00	**0.79**±0.00	**0.77**±0.00
	Intermediate	**0.41**±0.00	**0.63**±0.00	**0.50**±0.00
	Dark Brown	**0.92**±0.00	**0.75**±0.00	**0.83**±0.00
E12	Blue	0.69±0.00	0.72±0.00	0.70±0.00
	Intermediate	0.39±0.00	0.39±0.00	0.39±0.00
	Dark Brown	0.86±0.00	0.84±0.00	0.85±0.00

Table 5.15: Results for eye classification for the CGAN using SDV encoding

ID	Class	Precision	Recall	F1
E1	Blue	0.56±0.03	0.67±0.04	0.61±0.03
	Intermediate	0.24±0.08	0.29±0.11	0.26±0.09

	Dark Brown	0.84±0.03	0.73±0.01	0.77±0.02
E2	Blue	0.54±0.04	0.56±0.09	0.52±0.06
	Intermediate	0.20±0.00	0.26±0.04	0.24±0.01
	Dark Brown	0.86±0.02	0.77±0.04	0.81±0.01
E3	Blue	0.48±0.00	0.53±0.00	0.51±0.00
	Intermediate	0.23±0.00	0.32±0.00	0.26±0.00
	Dark Brown	0.87±0.00	0.72±0.00	0.79±0.00
E4	Blue	0.50±0.00	0.67±0.00	0.57±0.00
	Intermediate	0.24±0.00	0.26±0.00	0.25±0.00
	Dark Brown	0.85±0.00	0..71±0.00	0.78±0.00
E5	Blue	0.58±0.00	0.81±0.00	0.68±0.00
	Intermediate	0.26±0.00	0.47±0.00	0.34±0.00
	Dark Brown	0.91±0.00	0.54±0.00	0.68±0.00
E6	Blue	0.59±0.06	0.63±0.11	0.61±0.08
	Intermediate	0.26±0.09	0.29±0.04	0.30±0.05
	Dark Brown	0.84±0.01	0.74±0.07	0.79±0.04
E7	Blue	0.58±0.00	0.81±0.00	0.68±0.00
	Intermediate	0.38±0.00	0.21±0.00	0.27±0.00
	Dark Brown	0.84±0.00	0.84±0.00	0.84±0.00
E8	Blue	**0.69**±0.00	**0.81**±0.00	**0.74**±0.00
	Intermediate	**0.37**±0.00	**0.29**±0.00	**0.32**±0.00
	Dark Brown	**0.84**±0.00	**0.84**±0.00	**0.84**±0.00
E9	Blue	0.53±0.01	0.60±0.04	0.58±0.02
	Intermediate	0.25±0.05	0.34±0.04	0.29±0.05
	Dark Brown	0.87±0.05	0.70±0.04	0.77±0.02
E10	Blue	0.54±0.02	0.60±0.06	0.58±0.02
	Intermediate	0.23±0.01	0.26±0.05	0.23±0.03
	Dark Brown	0.88±0.01	0.74±0.02	0.80±0.02
E11	Blue	0.50±0.00	0.67±0.00	0.57±0.00
	Intermediate	0.23±0.00	0.26±0.00	0.24±0.00
	Dark Brown	0.86±0.00	0.70±0.00	0.78±0.00
E12	Blue	0.47±0.00	0.56±0.00	0.51±0.00
	Intermediate	0.24±0.00	0.32±0.00	0.27±0.00

	Dark Brown	0.87±0.00	0.71±0.00	0.78±0.00

Table 5.16: Results for eye classification for the CGAN using additive encoding

ID	Class	Precision	Recall	F1
E1	Blue	0.36±0.01	0.47±0.02	0.40±0.01
	Intermediate	0.29±0.00	0.39±0.01	0.34±0.00
	Dark Brown	0.86±0.00	0.66±0.01	0.75±0.00
E2	Blue	0.35±0.01	0.40±0.04	0.37±0.02
	Intermediate	0.29±0.02	0.45±0.04	0.35±0.03
	Dark Brown	0.85±0.01	0.67±0.01	0.75±0.00
E3	Blue	0.44±0.00	0.56±0.00	0.49±0.00
	Intermediate	0.28±0.00	0.34±0.00	0.31±0.00
	Dark Brown	0.83±0.00	0.69±0.00	0.75±0.00
E4	Blue	0.41±0.00	0.51±0.00	0.45±0.00
	Intermediate	0.26±0.00	0.37±0.00	0.30±0.00
	Dark Brown	0.84±0.00	0.64±0.00	0.73±0.00
E5	Blue	0.30±0.14	0.40±0.14	0.34±0.14
	Intermediate	0.21±0.08	0.21±0.18	0.21±0.12
	Dark Brown	0.62±0.13	0.56±0.05	0.59±0.08
E6	Blue	0.52±0.03	0.60±0.08	0.53±0.11
	Intermediate	0.27±0.01	0.45±0.00	0.34±0.00
	Dark Brown	0.87±0.00	0.59±0.04	0.70±0.02
E7	Blue	**0.61**±0.00	**0.72**±0.00	**0.66**±0.00
	Intermediate	**0.28**±0.00	**0.39**±0.00	**0.33**±0.00
	Dark Brown	**0.86**±0.00	**0.69**±0.00	**0.76**±0.00
E8	Blue	0.64±0.00	0.63±0.00	0.64±0.00
	Intermediate	0.30±0.00	0.53±0.00	0.38±0.00
	Dark Brown	0.86±0.00	0.66±0.00	0.75±0.00
E9	Blue	0.50±0.02	0.60±0.08	0.55±0.05
	Intermediate	0.24±0.07	0.32±0.11	0.27±0.09
	Dark Brown	0.87±0.03	0.73±0.02	0.77±0.01
E10	Blue	0.48±0.01	0.56±0.07	0.52±0.02

	Intermediate	0.23±0.01	0.34±0.06	0.27±0.04
	Dark Brown	0.87±0.01	0.68±0.01	0.76±0.01
E11	Blue	0.49±0.00	0.67±0.00	0.57±0.00
	Intermediate	0.24±0.00	0.29±0.00	0.27±0.00
	Dark Brown	0.86±0.00	0.69±0.00	0.76±0.00
E12	Blue	0.52±0.00	0.65±0.00	0.58±0.00
	Intermediate	0.24±0.00	0.29±0.00	0.26±0.00
	Dark Brown	0.86±0.00	0.72±0.00	0.79±0.00

Table 5.17: SNPs for eye classification selected for the SMOTE experiments

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	59	rs3768056, rs2070959, rs16891982, rs28777, rs13289, rs12203592, rs4959270, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602 rs1393350, rs642742, rs12821256, rs12896399, rs2402130, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196 rs895828, rs895829. rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977 rs4932620, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805006, rs1110400, rs885479 rs1805009, rs9894429, rs10424065, rs2424984, rs2378249, rs2835630

E5	57	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810 rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs1042602, rs1393350, rs10777129, rs642742 rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606 rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241 rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203 rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs885479, rs9894429 rs2424984, rs2378249, rs2835630
E6	48	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289 rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819 rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1375164 rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs916977, rs8039195, rs1426654 rs1724630, rs3212345, rs9894429, rs2424984, rs2378249, rs2835630

E7, E8	57	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs885479, rs9894429, rs2424984, rs2378249, rs2835630

Table 5.18: SNPs for eye classification selected for the SMOTEEN experiments

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	46	rs3768056, rs2070959, rs16891982, rs13289, rs12203592, rs1325127, rs2733832, rs683, rs10756819, rs1393350, rs10777129, rs12896399, rs2402130, rs2594935, rs1800407, rs1037208, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs3212345, rs1110400, rs885479, rs10424065, rs6119471, rs2424984, rs2378249, rs2835630

E5	54	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs9894429, rs2424984, rs2378249, rs2835630
E6	54	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs9894429, rs6119471, rs2424984, rs2378249, rs2835630

E7, E8	54	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs9894429, rs2424984, rs2378249, rs2835630

Table 5.19: SNPs for eye classification selected for the CNN experiments

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	5	rs7948623, rs1448484, rs12913832, rs1426654, rs1805006
E5	3	rs1129038, rs12913832, rs9894429
E6	1	rs12913832
E7, E8	3	rs1129038, rs12913832, rs9894429

Table 5.20: SNPs for eye classification selected for the experiments without class balancing

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	4	rs6497271, rs12913832, rs1426654, rs1805006

E5	47	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800404, rs4778232, rs1375164, rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs916977, rs8039195, rs1426654, rs1724630, rs3212345, rs1805005, rs9894429, rs2424984, rs2378249, rs2835630
E6	50	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12821256, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1375164, rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs916977, rs8039195, rs1426654, rs1724630, rs3212345, rs1805005, rs9894429, rs2424984, rs2378249, rs2835630
E7, E8	47	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800404, rs4778232, rs1375164, rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs916977, rs8039195, rs1426654, rs1724630, rs3212345, rs1805005, rs9894429, rs2424984, rs2378249, rs2835630

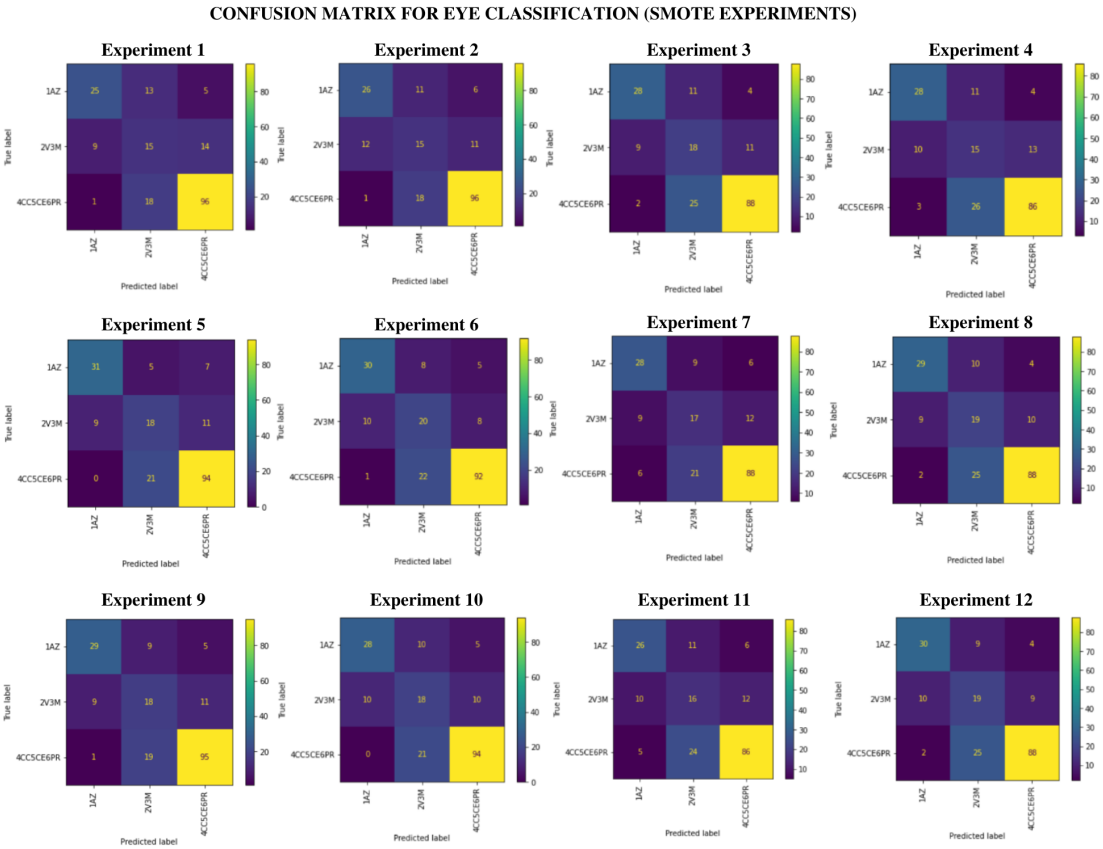
Table 5.21: SNPs for eye classification selected for CGAN experiments using additive encoding

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	13	rs13289, rs7948623, rs1393350, rs1900758, rs3794606, rs895829, rs1129038, rs3935591, rs1805006, rs1110400, rs885479, rs10424065, rs6119471
E5	19	rs2070959, rs16891982, rs4959270, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs12896399, rs2036213, rs3794606, rs4778137, rs1129038, rs7494942, rs12913832, rs3935591, rs3212345, rs9894429, rs2835630
E6	18	rs2070959, rs16891982, rs13289, rs4959270, rs2733832, rs683, rs10756819, rs1042602, rs12896399, rs2036213, rs3794606, rs1129038, rs7494942, rs12913832, rs3935591, rs3212345, rs9894429, rs2835630
E7, E8	19	rs2070959, rs16891982, rs4959270, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs12896399, rs2036213, rs3794606, rs4778137, rs1129038, rs7494942, rs12913832, rs3935591, rs3212345, rs9894429, rs2835630

Table 5.22: SNPs for eye classification selected for CGAN experiments using SDV encoding

ID	Number of SNPs	SNPs
E1, E3, E4	65	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs1393350, rs10777129, rs12821256, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs4932620, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs1805006, rs1110400, rs885479, rs1805009, rs9894429, rs10424065, rs6119471, rs2424984, rs2378249, rs2835630
E2	15	rs7948623, rs12821256, rs4778232, rs1448484, rs1375164, rs12913832, rs7170852, rs2238289, rs4932620, rs16950987, rs1805005, rs1805006, rs1110400, rs10424065, rs6119471
E5, E7, E8	4	rs1129038, rs7494942, rs12913832, rs2835630
E6	50	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs4778232, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs9894429, rs2424984, rs2378249, rs2835630

Figure 5.10: Confusion matrix for SMOTE experiments for eye classification.



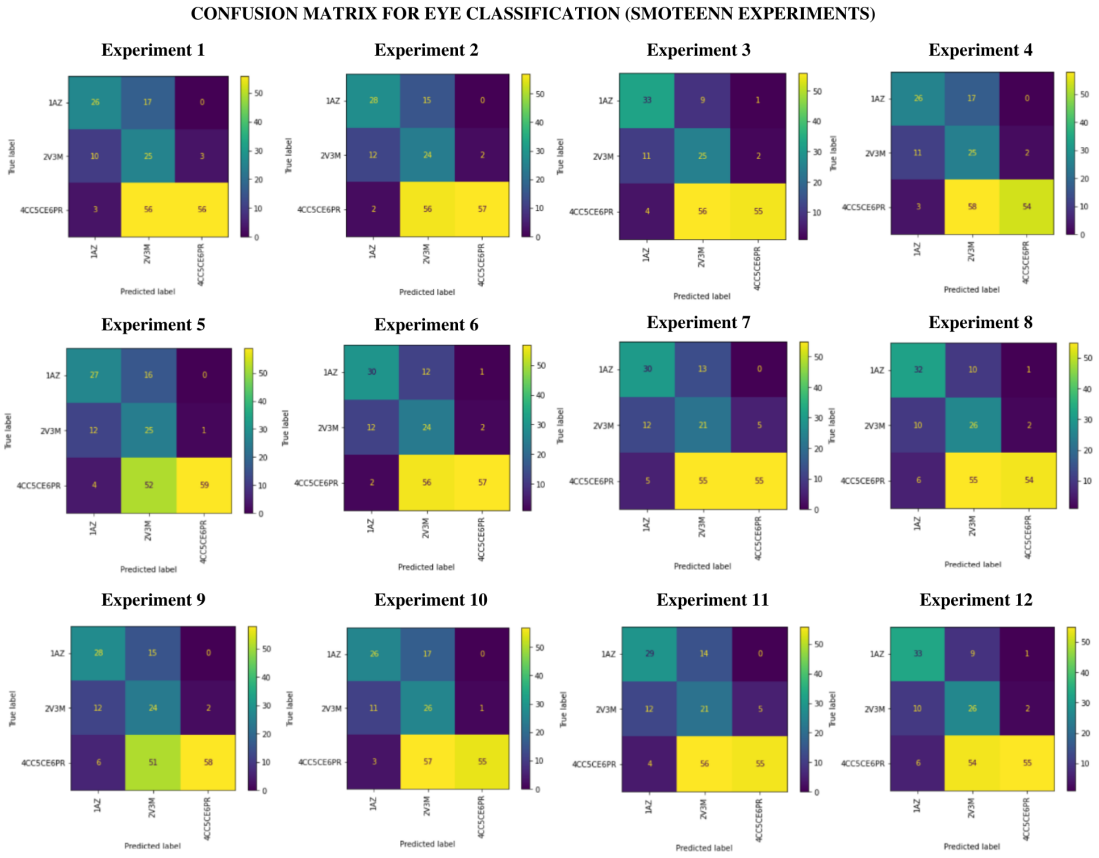
1AZ, 2V3M, and 4CC5CE6PR represent the Blue, Intermediate, and Dark Brown classes, respectively.

5.4 Skin color prediction results

For the experiment, we ran 3 replicates for each classifier. Table 5.23, Table 5.24, Table 5.25, Table 5.26, Table 5.27, and Table 5.28 show the metrics (average and standard deviation) for the overall results. Table 5.29, Table 5.30, Table 5.31, Table 5.32, Table 5.33, and Table 5.34 show the SNPs selected in each experiment.

All the experiments had a poor performance for the Intermediate class, and the results for the White and Brown classification were very similar. It was also observed that a very different number and set of SNPs were selected when the class balance approach changed. For cases like SMOTE, SMOTEENN, the CGANs, and without any class balancing, a large number of SNPs were selected, while in the CNN approach, only a few were chosen.

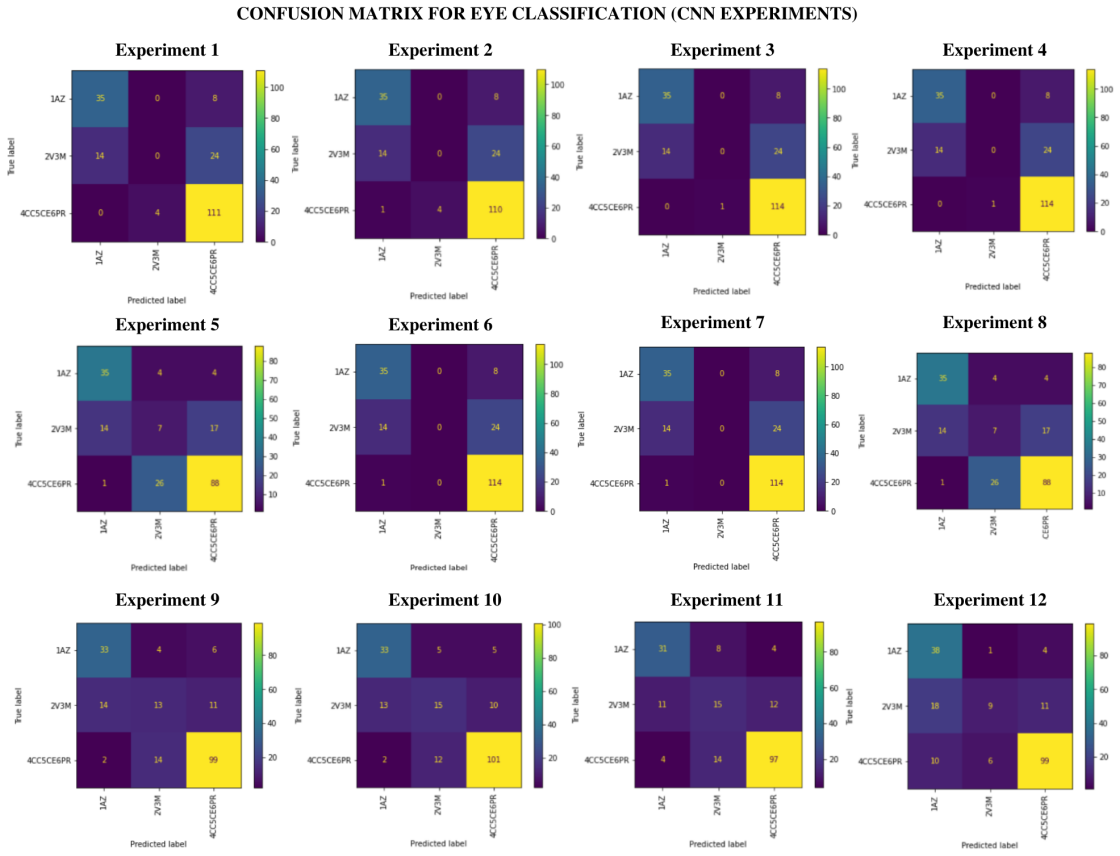
Figure 5.11: Confusion matrix for SMOTEENN experiments for eye classification.



1AZ, 2V3M, and 4CC5CE6PR represent the Blue, Intermediate, and Dark Brown classes, respectively.

The best result for skin classification was achieved by the experiment using SMOTE (E3) with 56 SNPs. The precision and recall, on average, respectively, were 0.90 and 0.94 for White, 0.56 and 0.48 for Intermediate, and 0.85 and 0.79 for Brown. The experiment E3 used as feature selection the SVM-RFE and an SVM as the classifier (without a GridSearch). The experience E6 without class balancing had a very similar result using 36 SNPs (Table 5.32). The experiment E6 used as feature selection the RF-RFE and an RF as the classifier (without a GridSearch). The precision and recall, on average, respectively, were 0.88 and 0.99 for White, 0.58 and 0.42 for Intermediate, and 0.90 and 0.66 for Brown. Of the 36 SNPs selected in experiment E6 without class balancing, 3 were not selected in experiment E3 using SMOTE (rs13289, rs642742, rs2424984). The SNP rs13289, rs642742, and rs2424984 are directly related to pigmentation traits. The study presented by Valenzuela et al. (2010) showed that one of the SNP rs2424984 is one of the SNPs with a high proportion of phenotypic variance of skin reflectance. Besides the experiment E3 using SMOTE had a slightly better performance, it seems the SMOTE could

Figure 5.12: Confusion matrix for CNN experiments for eye classification.



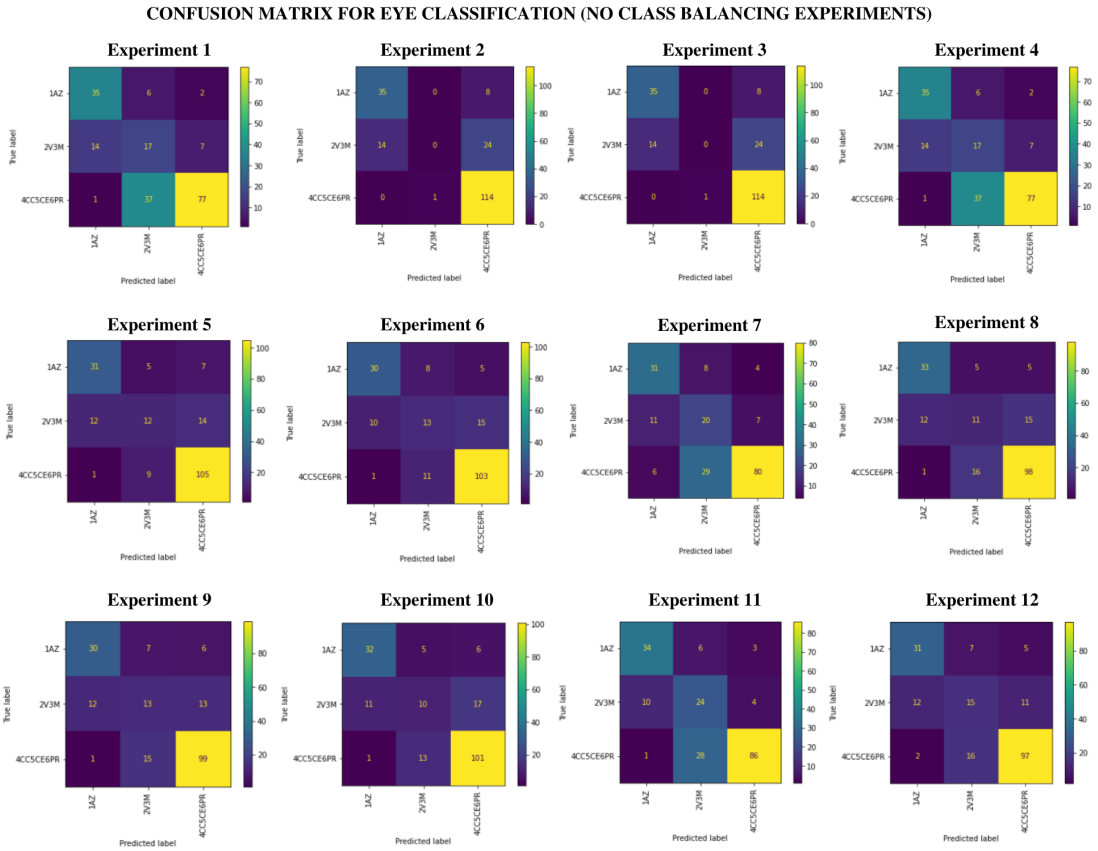
1AZ, 2V3M, and 4CC5CE6PR represent the Blue, Intermediate, and Dark Brown classes, respectively.

have affected the selection of the best SNPs for skin classification.

Figure 5.16, Figure 5.17, Figure 5.18, Figure 5.19, Figure 5.20, and Figure 5.21 show the confusion matrix for all classifiers. The columns of the confusion matrix are samples that were predicted by the classifier, and the rows show the actual samples of each class. The main diagonal of the matrix shows the number of samples correctly predicted by the model. Looking at the matrices, it is possible to notice that most of the classifiers confuse the Intermediate class with White and Brown. In the images, 1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

The performance of the classifiers using the data generated by the CGAN had a bad performance. Looking at the confusion matrices for the classifiers using the data generated by the CGAN, it is possible to notice the classifiers confuse the classes White and Intermediate much more frequently than in the other experiments. One of the hypotheses for it is that the CGAN generated very similar samples for both classes, and it

Figure 5.13: Confusion matrix eye classification without class balancing.



1AZ, 2V3M, and 4CC5CE6PR represent the Blue, Intermediate, and Dark Brown classes, respectively.

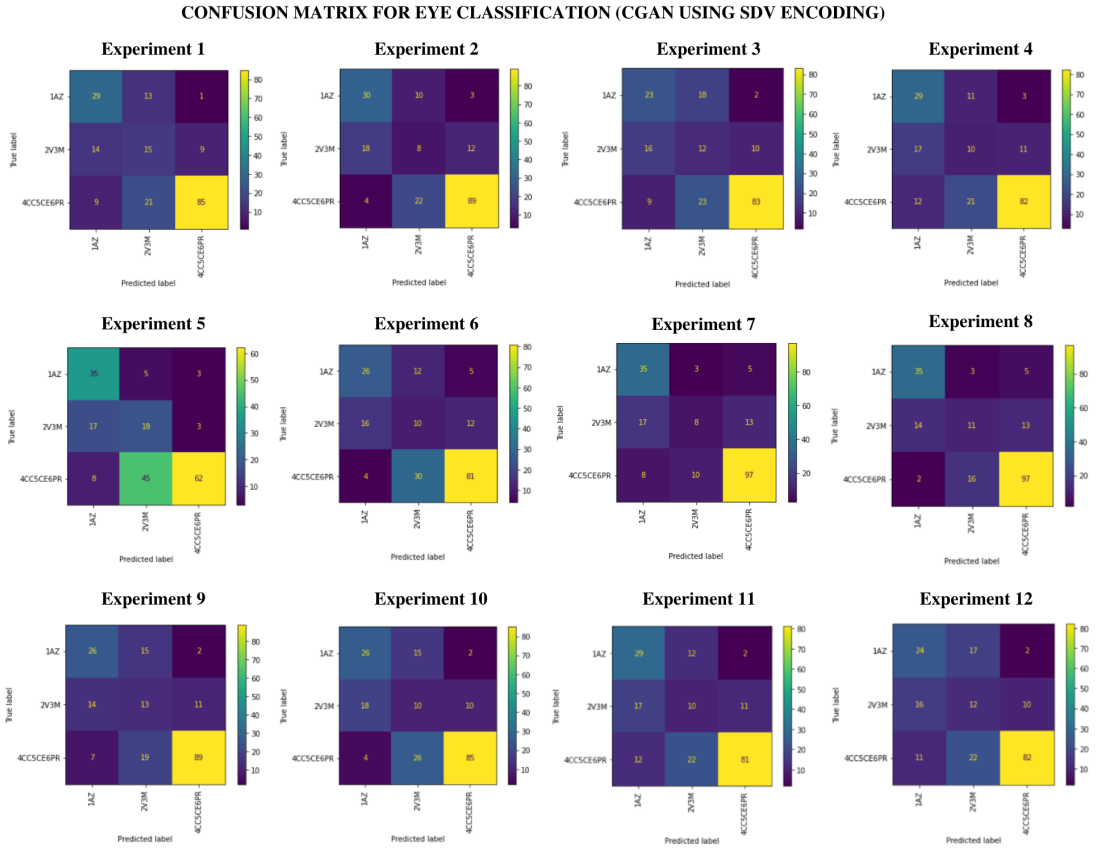
couldn't learn the difference between them. Another hypothesis is that the CGAN training should be improved, maybe by using other hyperparameters or using more data for the training process. Because the results for both types of encodings were very similar, it is not possible to determine which encoding is the best approach to train the CGAN, but it is important to highlight the study presented in Section 5.1 about the importance of the SNPs encoding.

Table 5.23: Results for skin classification using SMOTE

ID	Class	Precision	Recall	F1
E1	White	0.88±0.00	0.96±0.00	0.91±0.00
	Intermediate	0.46±0.03	0.39±0.02	0.42±0.02
	Brown	0.83±0.04	0.69±0.01	0.75±0.03
E2	White	0.90±0.00	0.94±0.00	0.92±0.00
	Intermediate	0.48±0.03	0.42±0.03	0.46±0.02

	Brown	0.85±0.00	0.79±0.04	0.82±0.02
E3	White	**0.90**±0.00	**0.94**±0.00	**0.92**±0.00
	Intermediate	**0.56**±0.00	**0.48**±0.00	**0.52**±0.00
	Brown	**0.85**±0.00	**0.79**±0.00	**0.82**±0.00
E4	White	0.88±0.00	0.92±0.00	0.90±0.00
	Intermediate	0.44±0.00	0.39±0.00	0.41±0.00
	Brown	0.81±0.00	0.76±0.00	0.79±0.00
E5	White	0.88±0.00	0.94±0.00	0.91±0.00
	Intermediate	0.46±0.01	0.35±0.02	0.40±0.01
	Brown	0.84±0.04	0.72±0.00	0.78±0.01
E6	White	0.88±0.01	0.94±0.00	0.91±0.00
	Intermediate	0.46±0.02	0.35±0.05	0.40±0.04
	Brown	0.85±0.04	0.76±0.03	0.82±0.02
E7	White	0.85±0.00	0.90±0.00	0.88±0.00
	Intermediate	0.29±0.00	0.23±0.00	0.25±0.00
	Brown	0.75±0.00	0.72±0.00	0.74±0.00
E8	White	0.89±0.00	0.91±0.00	0.90±0.00
	Intermediate	0.45±0.00	0.45±0.00	0.45±0.00
	Brown	0.85±0.00	0.76±0.00	0.80±0.00
E9	White	0.89±0.01	0.95±0.01	0.91±0.00
	Intermediate	0.50±0.05	0.45±0.07	0.47±0.06
	Brown	0.84±0.02	0.69±0.01	0.77±0.01
E10	White	0.90±0.00	0.94±0.01	0.92±0.00
	Intermediate	0.47±0.03	0.42±0.03	0.46±0.02
	Brown	0.81±0.02	0.72±0.02	0.76±0.02
E11	White	0.88±0.00	0.93±0.00	0.90±0.00
	Intermediate	0.44±0.00	0.35±0.00	0.39±0.00
	Brown	0.78±0.00	0.72±0.00	0.75±0.00
E12	White	0.89±0.00	0.93±0.00	0.91±0.00
	Intermediate	0.48±0.00	0.45±0.00	0.47±0.00
	Brown	0.85±0.00	0.76±0.00	0.80±0.00

Figure 5.14: Confusion matrix for CGAN experiments using SDV encoding for eye classification.



1AZ, 2V3M, and 4CC5CE6PR represent the Blue, Intermediate, and Dark Brown classes, respectively.

Table 5.24: Results for skin classification using SMOTEENN

ID	Class	Precision	Recall	F1
E1	White	0.95±0.00	0.81±0.01	0.87±0.00
	Intermediate	0.36±0.01	0.65±0.03	0.47±0.01
	Brown	0.81±0.02	0.72±0.03	0.78±0.02
E2	White	0.95±0.00	0.77±0.01	0.85±0.00
	Intermediate	0.35±0.01	0.65±0.03	0.45±0.01
	Brown	0.82±0.03	0.76±0.03	0.80±0.03
E3	White	0.94±0.00	0.81±0.00	0.87±0.00
	Intermediate	0.38±0.00	0.65±0.00	0.48±0.00
	Brown	0.81±0.00	0.76±0.00	0.79±0.00

E4	White	0.95±0.00	0.79±0.00	0.86±0.00
	Intermediate	0.36±0.00	0.58±0.00	0.44±0.00
	Brown	0.78±0.00	0.86±0.00	0.82±0.00
E5	White	**0.95**±0.00	**0.80**±0.01	**0.87**±0.01
	Intermediate	**0.37**±0.01	**0.71**±0.03	**0.49**±0.01
	Brown	**0.84**±0.03	**0.76**±0.02	**0.79**±0.01
E6	White	0.95±0.00	0.78±0.00	0.85±0.00
	Intermediate	0.35±0.00	0.65±0.02	0.45±0.00
	Brown	0.81±0.01	0.76±0.00	0.79±0.01
E7	White	0.96±0.00	0.80±0.00	0.87±0.00
	Intermediate	0.35±0.00	0.55±0.00	0.42±0.00
	Brown	0.73±0.00	0.83±0.00	0.77±0.00
E8	White	0.95±0.00	0.81±0.00	0.87±0.00
	Intermediate	0.40±0.00	0.68±0.00	0.50±0.00
	Brown	0.81±0.00	0.76±0.00	0.79±0.00
E9	White	0.95±0.00	0.79±0.01	0.86±0.00
	Intermediate	0.37±0.01	0.65±0.01	0.48±0.01
	Brown	0.81±0.02	0.76±0.02	0.79±0.02
E10	White	0.95±0.00	0.78±0.00	0.85±0.00
	Intermediate	0.34±0.02	0.65±0.03	0.45±0.02
	Brown	0.79±0.03	0.76±0.03	0.77±0.03
E11	White	0.96±0.00	0.79±0.00	0.87±0.00
	Intermediate	0.34±0.00	0.55±0.00	0.42±0.00
	Brown	0.73±0.00	0.83±0.00	0.77±0.00
E12	White	0.95±0.00	0.81±0.00	0.87±0.00
	Intermediate	0.39±0.00	0.68±0.00	0.49±0.00
	Brown	0.81±0.00	0.72±0.00	0.76±0.00

Table 5.25: Results for skin classification using CNN

ID	Class	Precision	Recall	F1
E1	White	0.90±0.00	0.88±0.00	0.89±0.00
	Intermediate	0.35±0.00	0.45±0.00	0.39±0.00

	Brown	0.79±0.00	0.66±0.00	0.72±0.00
E2	White	0.90±0.00	0.88±0.00	0.89±0.00
	Intermediate	0.35±0.00	0.45±0.00	0.39±0.00
	Brown	0.79±0.00	0.66±0.01	0.72±0.01
E3	White	**0.90**±0.00	**0.88**±0.00	**0.89**±0.00
	Intermediate	**0.36**±0.00	**0.45**±0.00	**0.40**±0.00
	Brown	**0.80**±0.00	**0.69**±0.00	**0.74**±0.00
E4	White	0.90 ±0.00	0.88±0.00	0.89±0.00
	Intermediate	0.36±0.00	0.45±0.00	0.40±0.00
	Brown	0.80±0.00	0.69±0.00	0.74±0.00
E5	White	0.90±0.00	0.88±0.00	0.89±0.00
	Intermediate	0.35±0.00	0.45±0.00	0.39±0.00
	Brown	0.79±0.00	0.66±0.00	0.72±0.00
E6	White	0.89±0.01	0.89±0.03	0.90±0.02
	Intermediate	0.36±0.04	0.45±0.01	0.39±0.02
	Brown	0.77±0.01	0.59±0.07	0.67±0.04
E7	White	0.90±0.00	0.88±0.00	0.89±0.00
	Intermediate	0.36±0.00	0.45±0.00	0.40±0.00
	Brown	0.80±0.00	0.69±0.00	0.74±0.00
E8	White	**0.90**±0.00	**0.88**±0.01	**0.89**±0.00
	Intermediate	**0.36**±0.00	**0.45**±0.01	**0.40**±0.00
	Brown	**0.80**±0.01	**0.69**±0.05	**0.74**±0.04
E9	White	0.88±0.01	0.96±0.01	0.92±0.00
	Intermediate	0.50±0.03	0.19±0.02	0.28±0.01
	Brown	0.73±0.06	0.83±0.04	0.78±0.02
E10	White	0.90±0.01	0.96±0.01	0.93±0.00
	Intermediate	0.69±0.09	0.29±0.01	0.41±0.00
	Brown	0.71±0.01	0.90±0.02	0.79±0.01
E11	White	0.89±0.00	0.97±0.00	0.93±0.00
	Intermediate	0.55±0.00	0.39±0.00	0.45±0.00
	Brown	0.85±0.00	0.76±0.00	0.80±0.00
E12	White	0.89±0.00	0.99±0.00	0.94±0.00
	Intermediate	0.78±0.00	0.23±0.00	0.35±0.00

	Brown	0.74±0.00	0.90±0.00	0.81±0.00

Table 5.26: Results for skin classification without class balancing

ID	Class	Precision	Recall	F1
E1	White	0.86±0.00	1.00±0.00	0.92±0.00
	Intermediate	0.55±0.05	0.19±0.03	0.29±0.04
	Brown	0.85±0.01	0.79±0.01	0.81±0.01
E2	White	0.89±0.00	1.00±0.00	0.94±0.00
	Intermediate	0.62±0.03	0.39±0.04	0.47±0.03
	Brown	0.81±0.02	0.72±0.03	0.76±0.03
E3	White	0.88±0.00	1.00±0.00	0.93±0.00
	Intermediate	0.69±0.00	0.29±0.00	0.41±0.00
	Brown	0.82±0.00	0.79±0.00	0.81±0.00
E4	White	0.88±0.00	0.90±0.00	0.89±0.00
	Intermediate	0.37±0.00	0.42±0.00	0.39±0.00
	Brown	0.87±0.00	0.69±0.00	0.77±0.00
E5	White	0.89±0.01	0.99±0.00	0.93±0.01
	Intermediate	0.53±0.06	0.35±0.08	0.42±0.08
	Brown	0.90±0.00	0.66±0.02	0.76±0.01
E6	White	**0.88**±0.00	**0.99**±0.01	**0.93**±0.00
	Intermediate	**0.58**±0.03	**0.42**±0.04	**0.50**±0.04
	Brown	**0.90**±0.00	**0.66**±0.02	**0.76**±0.01
E7	White	0.93±0.00	0.85±0.00	0.89±0.00
	Intermediate	0.41±0.00	0.61±0.00	0.49±0.00
	Brown	0.88±0.00	0.76±0.00	0.81±0.00
E8	White	0.88±0.00	0.99±0.00	0.93±0.00
	Intermediate	0.55±0.00	0.35±0.00	0.43±0.00
	Brown	0.87±0.00	0.69±0.00	0.77±0.00
E9	White	0.84±0.00	1.00±0.00	0.92±0.00
	Intermediate	0.57±0.01	0.19±0.06	0.29±0.08
	Brown	0.88±0.03	0.72±0.02	0.81±0.01
E10	White	0.87±0.00	1.00±0.00	0.93±0.00

	Intermediate	0.65±0.06	0.35±0.03	0.46±0.04
	Brown	0.88±0.02	0.72±0.01	0.79±0.02
E11	White	0.94±0.00	0.88±0.00	0.91±0.00
	Intermediate	0.50±0.00	0.71±0.00	0.59±0.00
	Brown	0.88±0.00	0.79±0.00	0.84±0.00
E12	White	0.87±0.00	1.00±0.00	0.93±0.00
	Intermediate	0.62±0.00	0.32±0.00	0.43±0.00
	Brown	0.91±0.00	0.72±0.00	0.81±0.00

Table 5.27: Results for skin classification for the CGAN using SDV encoding

ID	Class	Precision	Recall	F1
E1	Blue	0.85±0.00	0.65±0.00	0.74±0.00
	Intermediate	0.21±0.00	0.35±0.00	0.27±0.00
	Dark Brown	0.46±0.00	0.66±0.00	0.54±0.00
E2	Blue	0.85±0.01	0.62±0.02	0.72±0.01
	Intermediate	0.20±0.01	0.35±0.05	0.26±0.02
	Dark Brown	0.48±0.01	0.69±0.01	0.56±0.01
E3	Blue	0.85±0.00	0.63±0.00	0.73±0.00
	Intermediate	0.24±0.00	0.45±0.00	0.31±0.00
	Dark Brown	0.49±0.00	0.62±0.00	0.55±0.00
E4	Blue	0.88±0.00	0.61±0.00	0.72±0.00
	Intermediate	0.22±0.00	0.26±0.00	0.24±0.00
	Dark Brown	0.42±0.00	0.93±0.00	0.57±0.00
E5	Blue	**0.93**±0.02	**0.70**±0.00	**0.80**±0.01
	Intermediate	**0.32**±0.02	**0.58**±0.09	**0.41**±0.04
	Dark Brown	**0.58**±0.00	**0.76**±0.01	**0.66**±0.14
E6	Blue	0.95±0.02	0.67±0.01	0.78±0.00
	Intermediate	0.29±0.01	0.55±0.06	0.38±0.02
	Dark Brown	0.57±0.02	0.79±0.01	0.67±0.02
E7	Blue	0.91±0.00	0.71±0.00	0.80±0.00
	Intermediate	0.31±0.00	0.52±0.00	0.39±0.00

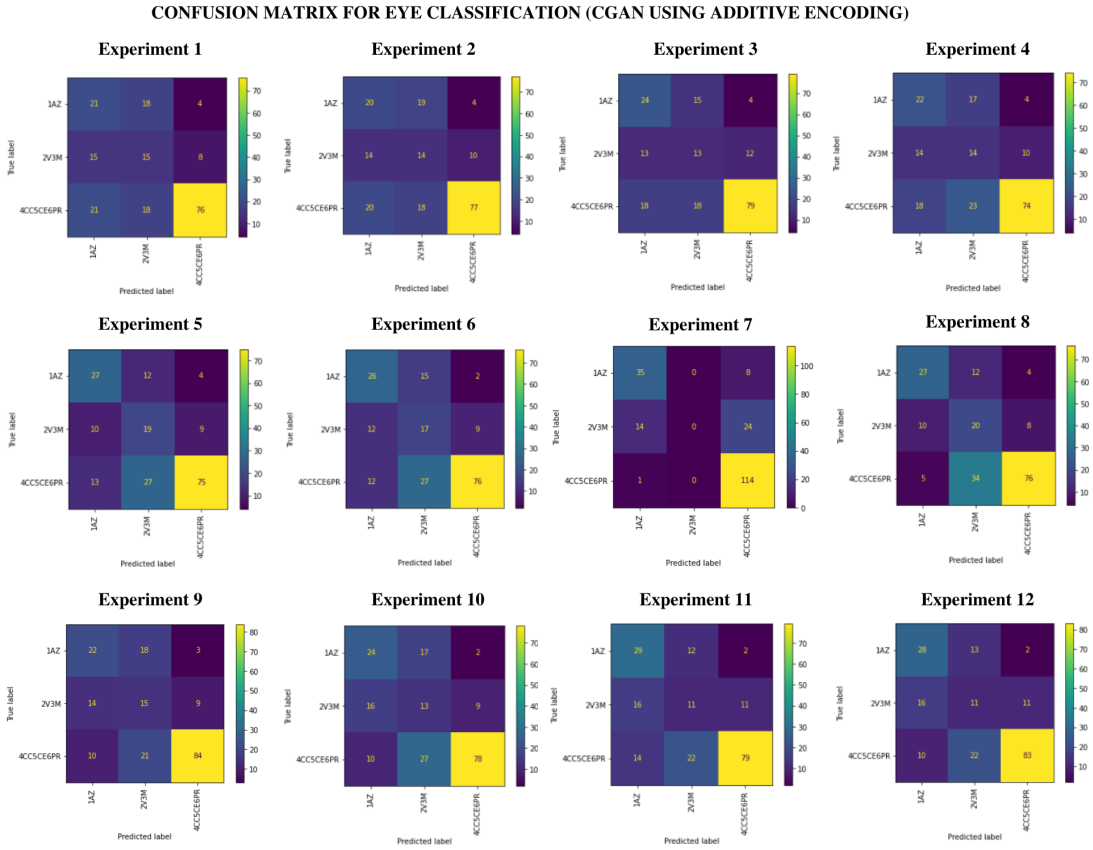
	Dark Brown	0.61±0.00	0.79±0.00	0.69±0.00
E8	Blue	0.91±0.00	0.71±0.00	0.80±0.00
	Intermediate	0.31±0.00	0.52±0.00	0.39±0.00
	Dark Brown	0.61±0.00	0.79±0.00	0.69±0.00
E9	Blue	0.92±0.00	0.70±0.00	0.79±0.00
	Intermediate	0.30±0.01	0.52±0.01	0.39±0.01
	Dark Brown	0.59±0.01	0.79±0.01	0.67±0.01
E10	Blue	0.91±0.00	0.71±0.01	0.79±0.01
	Intermediate	0.28±0.03	0.48±0.05	0.36±0.04
	Dark Brown	0.57±0.00	0.79±0.00	0.67±0.00
E11	Blue	0.92±0.00	0.71±0.00	0.80±0.00
	Intermediate	0.33±0.00	0.55±0.00	0.41±0.00
	Dark Brown	0.59±0.00	0.79±0.00	0.68±0.00
E12	Blue	0.92±0.00	0.71±0.00	0.80±0.00
	Intermediate	0.33±0.00	0.55±0.00	0.41±0.00
	Dark Brown	0.59±0.00	0.79±0.00	0.68±0.00

Table 5.28: Results for skin classification for the CGAN using additive encoding

ID	Class	Precision	Recall	F1
E1	Blue	0.89±0.00	0.74±0.00	0.81±0.00
	Intermediate	0.21±0.00	0.29±0.00	0.24±0.00
	Dark Brown	0.59±0.01	0.79±0.02	0.68±0.01
E2	Blue	0.91±0.00	0.73±0.01	0.81±0.00
	Intermediate	0.23±0.02	0.32±0.04	0.27±0.02
	Dark Brown	0.57±0.01	0.83±0.02	0.68±0.01
E3	Blue	0.92±0.00	0.60±0.00	0.73±0.00
	Intermediate	0.19±0.00	0.39±0.00	0.26±0.00
	Dark Brown	0.60±0.00	0.93±0.00	0.73±0.00
E4	Blue	0.90±0.00	0.73±0.00	0.80±0.00
	Intermediate	0.33±0.00	0.32±0.00	0.33±0.00
	Dark Brown	0.48±0.00	0.93±0.00	0.64±0.00
E5	Blue	0.92±0.02	0.66±0.01	0.76±0.00

	Intermediate	0.26±0.01	0.55±0.01	0.35±0.01
	Dark Brown	0.63±0.05	0.76±0.00	0.69±0.02
E6	Blue	0.92±0.01	0.65±0.01	0.76±0.01
	Intermediate	0.24±0.01	0.52±0.05	0.33±0.02
	Dark Brown	0.59±0.07	0.76±0.07	0.65±0.07
E7	Blue	0.92±0.00	0.72±0.04	0.81±0.02
	Intermediate	0.31±0.04	0.55±0.05	0.40±0.05
	Dark Brown	0.66±0.05	0.79±0.01	0.72±0.04
E8	Blue	0.92±0.00	0.72±0.00	0.81±0.00
	Intermediate	0.31±0.00	0.55±0.00	0.40±0.00
	Dark Brown	0.66±0.00	0.79±0.00	0.72±0.00
E9	Blue	0.92±0.01	0.68±0.00	0.78±0.01
	Intermediate	0.30±0.04	0.58±0.06	0.40±0.05
	Dark Brown	0.62±0.04	0.76±0.10	0.72±0.05
E10	Blue	0.91±0.03	0.64±0.00	0.75±0.00
	Intermediate	0.28±0.01	0.58±0.07	0.38±0.02
	Dark Brown	0.62±0.04	0.69±0.01	0.67±0.02
E11	Blue	0.93±0.00	0.71±0.00	0.80±0.00
	Intermediate	0.29±0.00	0.55±0.00	0.38±0.00
	Dark Brown	0.66±0.00	0.79±0.00	0.72±0.00
E12	Blue	**0.93**±0.00	**0.73**±0.00	**0.82**±0.00
	Intermediate	**0.32**±0.00	**0.55**±0.00	**0.40**±0.00
	Dark Brown	**0.65**±0.00	**0.83**±0.00	**0.73**±0.00

Figure 5.15: Confusion matrix for CGAN experiments using additive encoding for eye classification.



1AZ, 2V3M, and 4CC5CE6PR represent the Blue, Intermediate, and Dark Brown classes, respectively.

Table 5.29: SNPs for skin classification selected for SMOTE experiments

ID	Number of SNPs	SNPs
E1, E2, E3, E4	56	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs1393350, rs12821256, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs1448484, rs1375164, rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs885479, rs1805009, rs9894429, rs10424065, rs6119471, rs2378249, rs2835630
E5, E7, E8	48	rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800404, rs3794606, rs1448484, rs1375164, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs9894429, rs6119471, rs2424984, rs2378249, rs2835630

E6	59	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs9894429, rs6119471, rs2424984, rs2378249, rs2835630

Table 5.30: SNPs for skin classification selected for SMO-TEENN experiments

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	57	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs4932620, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs1805006, rs885479, rs6119471, rs2378249, rs2835630

E5	62	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs4932620, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs885479, rs9894429, rs10424065, rs6119471, rs2424984, rs2378249, rs2835630
E6	35	rs16891982, rs28777, rs183671, rs13289, rs4959270, rs2733832, rs683, rs11230664, rs7948623, rs1042602, rs10777129, rs2402130, rs2036213, rs7170989, rs1900758, rs1800404, rs3794606, rs1448484, rs895829, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs7170852, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs2835630
E7, E8	62	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs4932620, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs885479, rs9894429, rs10424065, rs6119471, rs2424984, rs2378249, rs2835630

Table 5.31: SNPs for skin classification selected for CNN experiments

ID	Number of SNPs	SNPs
E1, E2, E3, E4, E5, E7, E8	2	rs11230664, rs1426654
E6	1	rs1426654

Table 5.32: SNPs for skin classification selected for the experiments without class balancing

ID	Number of SNPs	SNPs
E1, E2, E3, E4	34	rs3768056, rs2070959, rs16891982, rs183671, rs12203592, rs13289810, rs1325127, rs683, rs10756819, rs11230664, rs7948623, rs1393350, rs12896399, rs7170989, rs1900758, rs1800404, rs3794606, rs4778232, rs1448484, rs4778241, rs3935591, rs11636232, rs2240203, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs885479, rs1805009, rs10424065, rs6119471, rs2378249
E5, E7, E8	23	rs16891982, rs28777, rs183671, rs2733832, rs10756819, rs11230664, rs1042602, rs642742, rs2036213, rs2594935, rs1800404, rs1448484, rs4778138, rs6497271, rs12913832, rs916977, rs1426654, rs1724630, rs3212345, rs9894429, rs6119471, rs2424984, rs2835630

E6	36	rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs2733832, rs683, rs10756819, rs11230664, rs7948623, rs1042602, rs642742, rs12896399, rs2036213, rs2594935, rs7170989, rs1900758, rs1800404, rs3794606, rs1448484, rs1375164, rs4778138, rs1129038, rs7494942, rs6497271, rs12913832, rs7170852, rs916977, rs1426654, rs1724630, rs3212345, rs9894429, rs6119471, rs2424984, rs2835630

Table 5.33: SNPs for skin classification selected for CGAN experiments using additive encoding

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	10	rs11230664, rs7948623, rs1448484, rs6497271, rs12913832, rs916977, rs1426654, rs1805006, rs1805009, rs10424065
E5, E7, E8	56	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs12203592, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs9894429, rs2424984, rs2378249, rs2835630

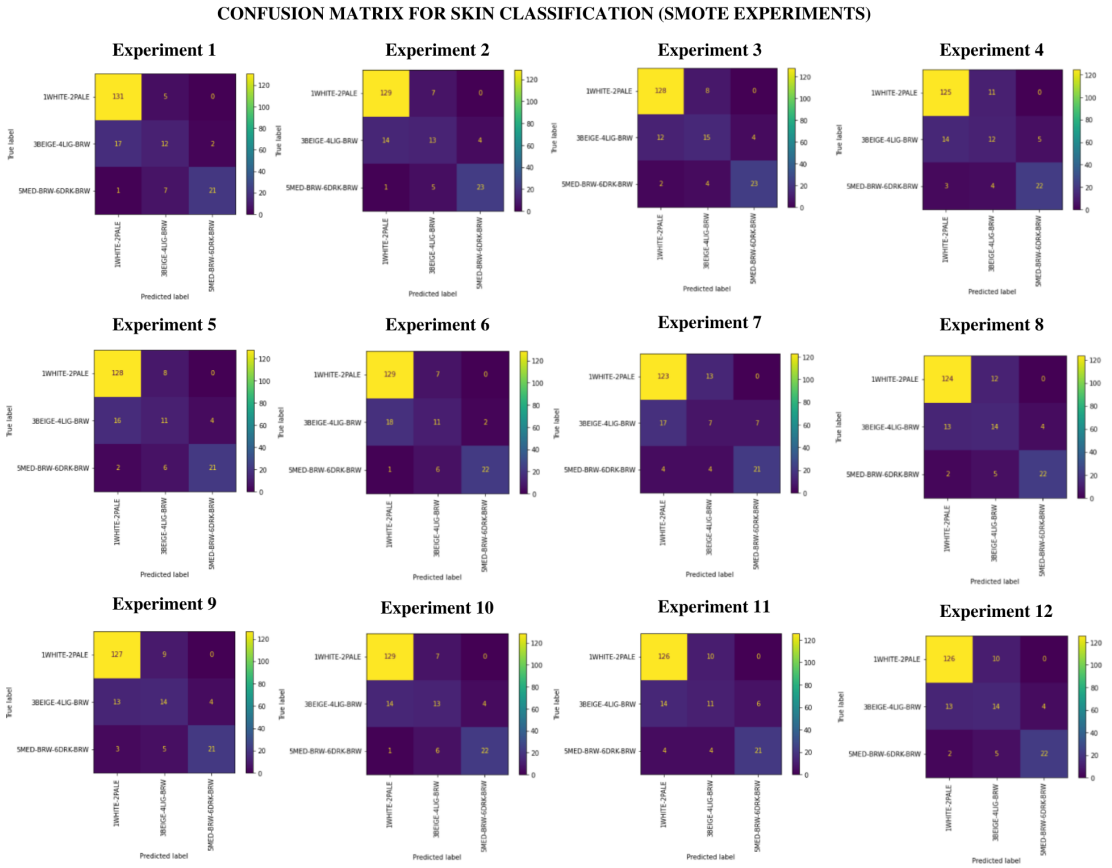
E6	50	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1037208, rs1800404, rs3794606, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs1129038, rs7494942, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs1426654, rs1724630, rs3212345, rs9894429, rs2424984, rs2378249, rs2835630

Table 5.34: SNPs for skin classification selected for CGAN experiments using SDV encoding

ID	**Number of SNPs**	**SNPs**
E1, E2, E3, E4	5	rs1900758, rs1129038, rs3935591, rs4932620, rs1110400
E5, E7, E8	57	rs3768056, rs2070959, rs16891982, rs28777, rs183671, rs13289, rs4959270, rs13289810, rs1325127, rs2733832, rs683, rs10756819, rs11230664, rs1042602, rs1393350, rs10777129, rs642742, rs12896399, rs2402130, rs2036213, rs2594935, rs7170989, rs1900758, rs1800407, rs1037208, rs1800404, rs3794606, rs4778232, rs1448484, rs1375164, rs1597196, rs895828, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs6497271, rs12913832, rs3935591, rs11636232, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs1426654, rs1724630, rs3212345, rs1805005, rs9894429, rs6119471, rs2424984, rs2378249, rs2835630

E6	34	rs16891982, rs28777, rs13289, rs4959270, rs1325127, rs2733832, rs683, rs642742, rs12896399, rs2036213, rs2594935, rs7170989, rs1900758, rs1800404, rs3794606, rs4778232, rs1375164, rs1597196, rs895829, rs4778137, rs4778138, rs4778241, rs1129038, rs7494942, rs12913832, rs3935591, rs7170852, rs2238289, rs2240203, rs916977, rs8039195, rs16950987, rs3212345, rs9894429

Figure 5.16: Confusion matrix for SMOTE experiments for skin classification.

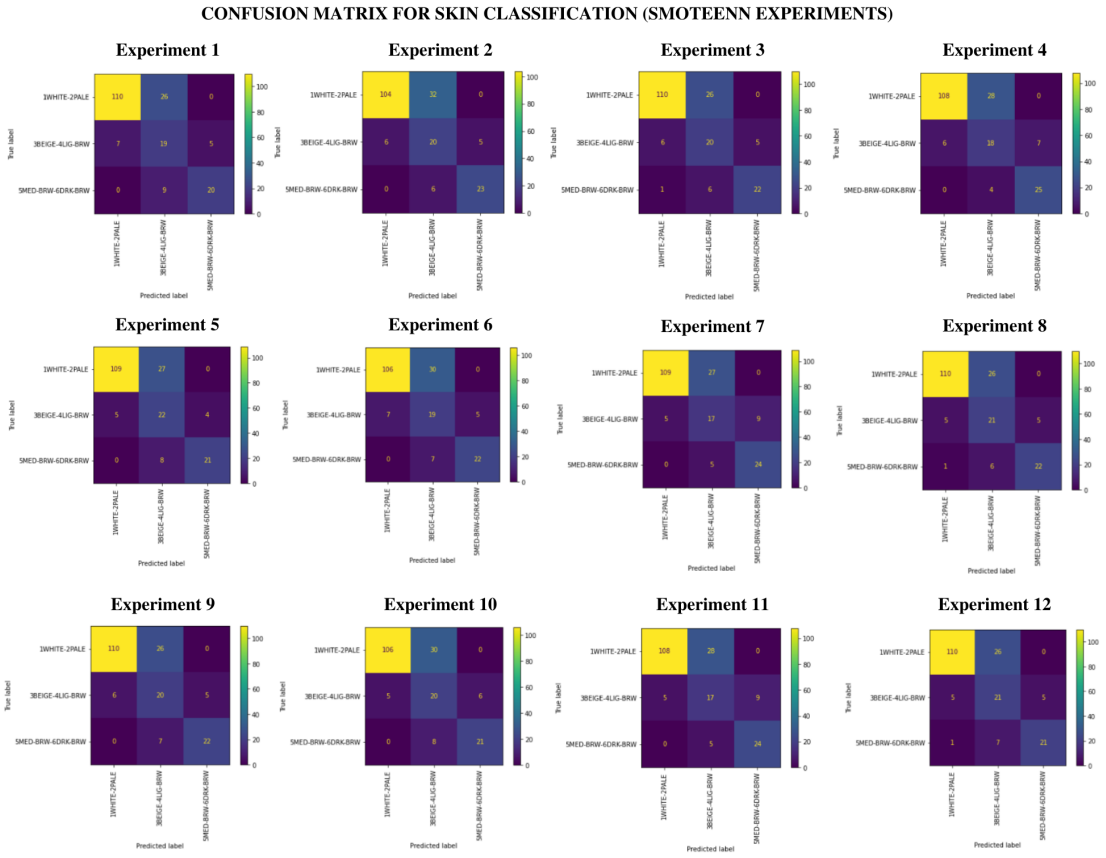


1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

5.5 Chapter Conclusion

This chapter presented the experiments for eye and skin classification using data from the Southern Brazilian population. A set of different approaches were tested for

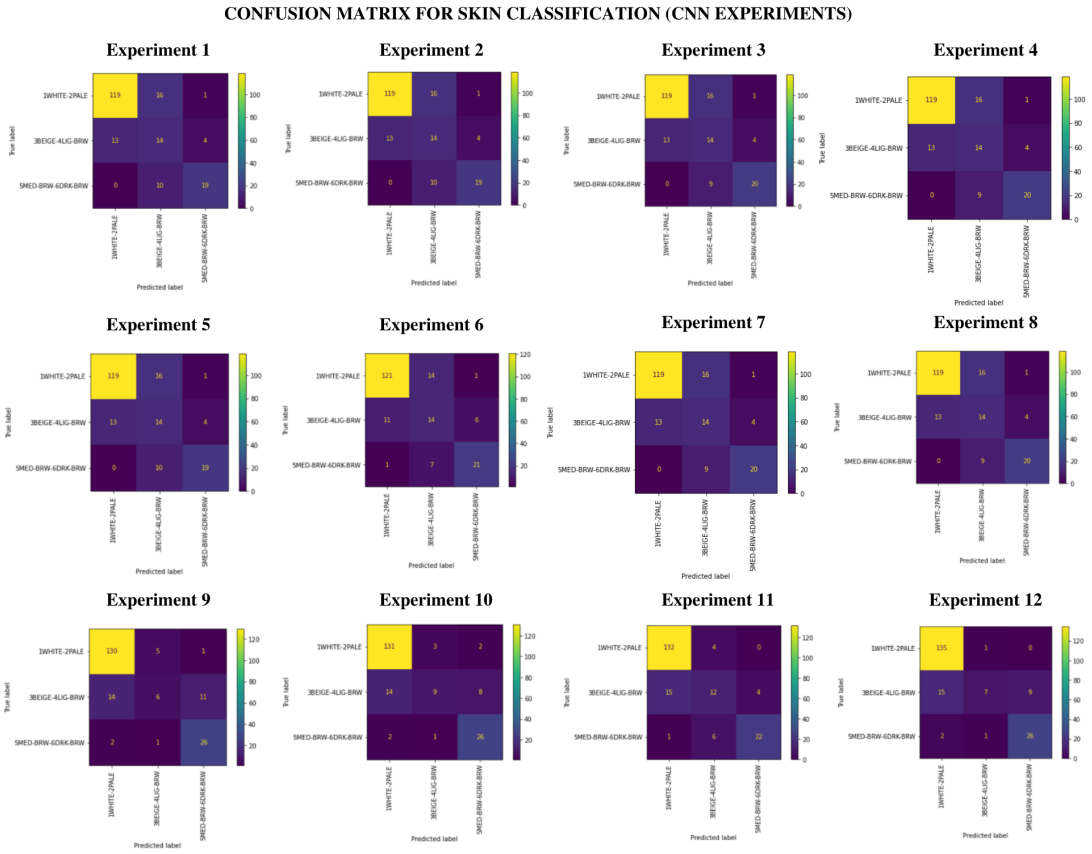
Figure 5.17: Confusion matrix for SMOTEENN experiments for skin classification.



1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

class balancing, feature selection, and classifiers to determine what is the best approach for the proposed problem. Combining all different approaches, it was 144 totalized experiments (72 for skin classification and 72 for eye classification). The final results showed that, for eye classification, only 4 SNPs are necessary to classify Blue and Dark Brown eyes (rs6497271, rs12913832, rs1426654, rs1805006), but only those 4 SNPs are not sufficient enough to classify the Intermediate class. The best result for eye classification was achieved by the experiment without any class balancing (E11) using all the 66 SNPs. The E11 experiment does not use feature selection and the classifier was an SVM with GridSearch (Table 5.7). The precision and recall, on average, respectively were 0.76 and 0.79 for Blue, 0.41 and 0.63 for Intermediate, and 0.92 and 0.75 for Dark Brown. As mentioned before, using only 4 SNPs a very close result can be obtained for Blue and Dark Brown using only 4 SNPs. The experiment E5 using SMOTE had also a similar result. The class imbalance seems to not affect the eye classifier as much as the selection of the SNPs. Both classifiers (SVM and RF) had similar performance, allowing the use of either

Figure 5.18: Confusion matrix for CNN experiments skin eye classification.

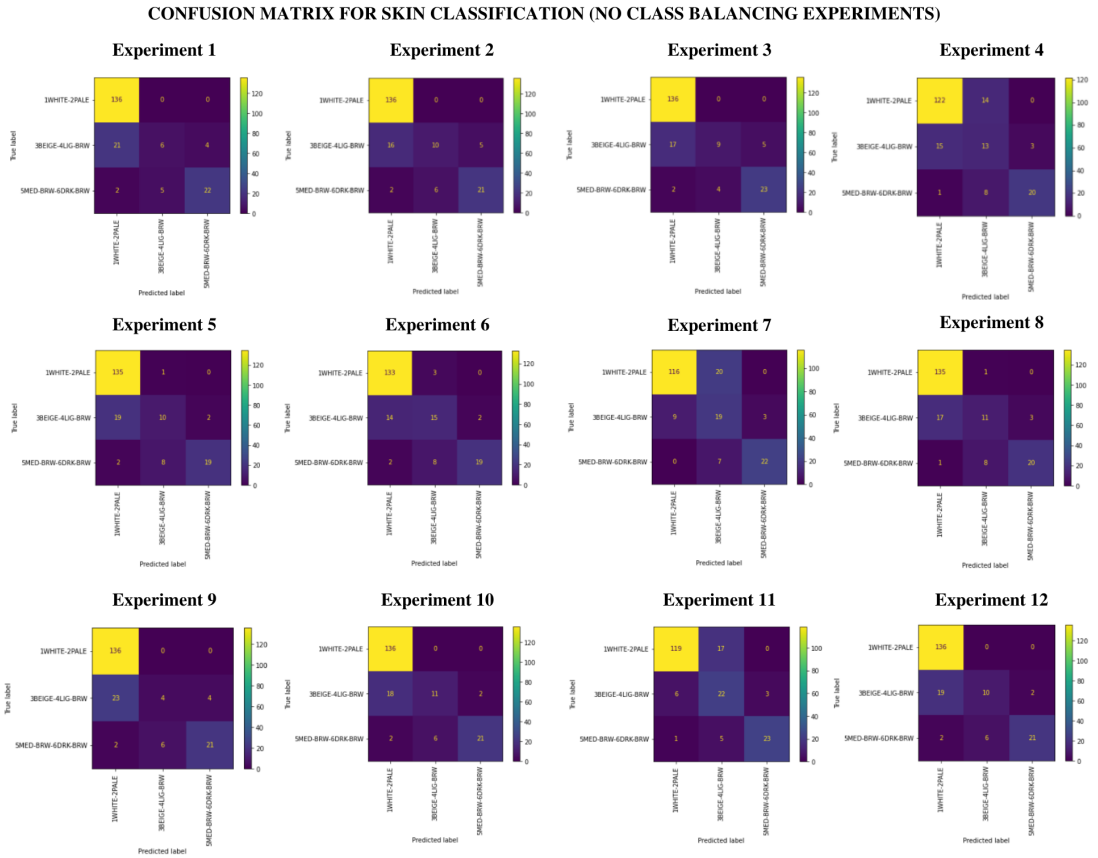


1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

of the two as the classifier.

For skin classification, the best result was achieved by the experiment using SMOTE (E3) with 56 SNPs. The precision and recall, on average, respectively, were 0.90 and 0.94 for White, 0.56 and 0.48 for Intermediate, and 0.85 and 0.79 for Brown. The experiment E3 used as feature selection the SVM-RFE and an SVM as the classifier (without a GridSearch). But it was noticed that the approach using SMOTE could have affected the selection of the SNPs because three SNPs that are known to be directly related to skin pigmentation were not chosen (rs13289, rs642742, rs2424984). A similar result was achieved by the experience E6 without class balancing using 36 SNPs (Table 5.32). The experiment E6 used as feature selection the RF-RFE and an RF as the classifier (without a GridSearch). The precision and recall, on average, respectively, were 0.88 and 0.99 for White, 0.58 and 0.42 for Intermediate, and 0.90 and 0.66 for Brown. Both classifiers (SVM and RF) had similar performance, allowing the use of either of the two as the classifier.

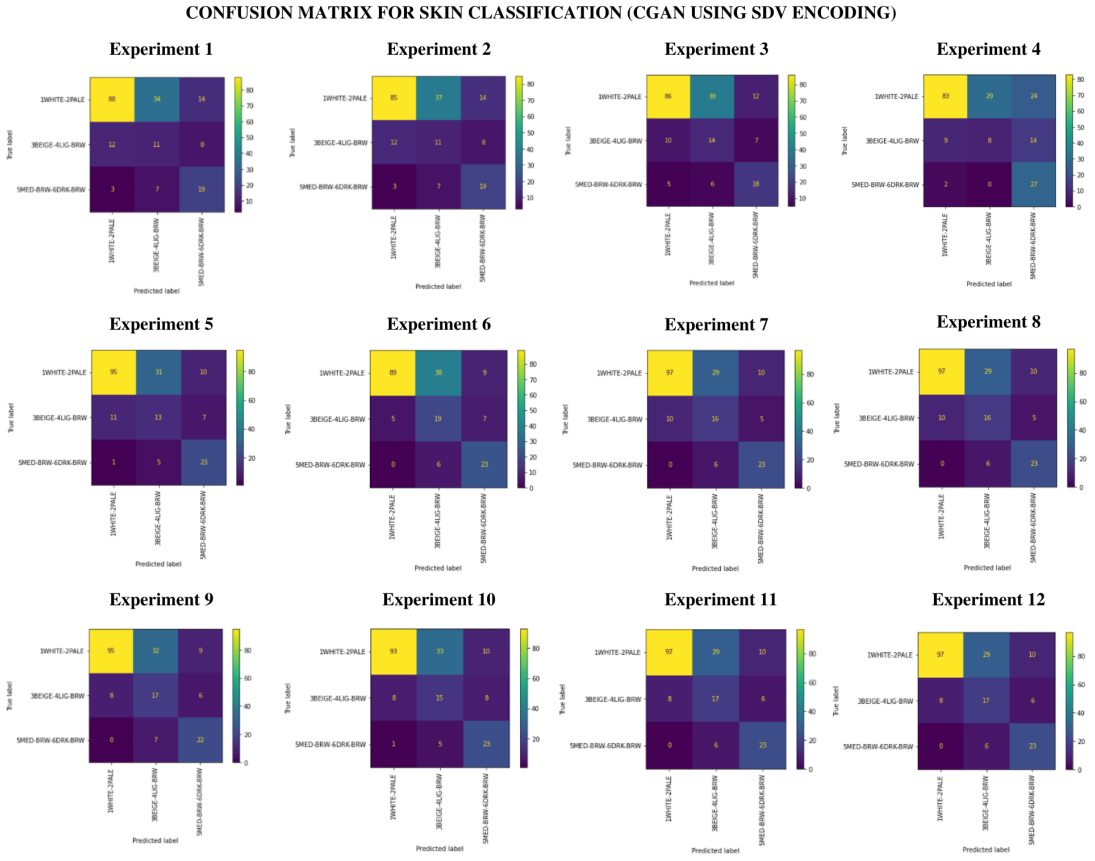
Figure 5.19: Confusion matrix skin classification without class balancing.



1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

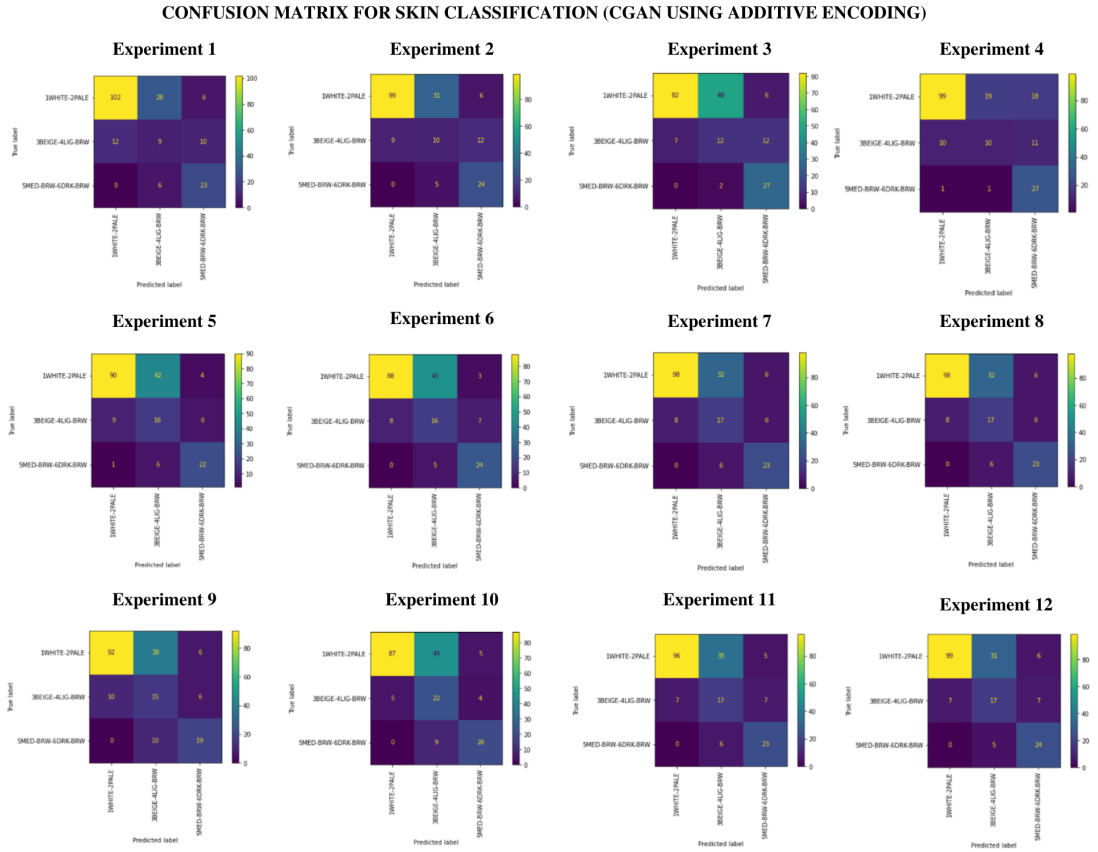
The next chapter will present the conclusion of the work.

Figure 5.20: Confusion matrix for CGAN experiments using SDV encoding for skin classification.



1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

Figure 5.21: Confusion matrix for CGAN experiments using additive encoding for skin classification.



1WHITE-2PALE, 3BEIGE-4LIG-BRW, and 5MED-BRW-9DRK-BRW represent the White, Intermediate, and Brown classes, respectively.

6 CONCLUSION

The proposed problem of this study was to find a solution for forensic use to predict eye and skin color using Single Nucleotide Polymorphisms for the Brazilian population. For the work, a study about the current solutions was made, as well as a theoretical basis study for biology and machine learning techniques. Most of the studies in the field have a solution using data from the Europen population, and each study has a different approach to solve the prediction of the phenotype problem, once there is no consensus about the best SNPs to use and the best approach for it. The main challenge of this work was to find the best solution using the data collected from the Southern Brazilian population.

For the proposed problem, 144 experiments were executed (72 for eye and 72 for skin classification). Each experiment had a different combination of techniques, to find the best approach. Two datasets were used in this study. The dataset for eye classification has 653 samples (Table 4.2), where 154 samples were classified as Blue, 158 samples were classified as Intermediate and 341 samples were classified as Dark Brown. The dataset for skin classification has 652 samples (Table 4.3), where 467 samples were classified as White, 107 samples were classified as Intermediate, and 78 samples were classified as Brown.

To deal with the class imbalanced problem, it was tested the SMOTE, SMOTEENN, CNN, and synthetic data generated by CGAN. Tests with no class balancing approach were also executed. The feature selection algorithm chosen was the recursive feature elimination, and some experiments without selecting the SNPs were also tested. For the classifiers, two algorithms were tested, Random Forest and Support Vector Machine. The most difficult task of this study was to find the best solution that accurately classifies the Intermediate class for eye and skin. The Intermediate class for both cases had poor performance. One hypothesis for the problem is the difficulty to define the SNPs that distinguish the intermediate colors from Blue and Dark Brown. Previous studies had raised the same difficulty. Walsh et al. (2011) mentioned that the Intermediate class was the most challenging to predict during the development of IrisPlex.

The final results showed that for the classifiers, both Random Forest and Support Vector Machine have a good performance to classify eye and skin color. For eye classification, 4 SNPs can be used (rs6497271, rs12913832, rs1426654, rs1805006) to predict the Blue and Dark Brown colors, and no class balance was necessary for this case. To have a better performance for the Intermediate, more SNPs are necessary, once only those

4 are not enough to distinguish the Intermediate from the other two classes. But even using more SNPs, the performance for the Intermediate was not good, as mentioned before. The best performance for eye classification had precision and recall, on average, respectively of 0.76 and 0.79 for Blue, 0.41 and 0.63 for Intermediate, and 0.92 and 0.75 for Dark Brown. The experiment using 4 SNPs with no class balancing had a precision and recall, on average, respectively of 0.70 and 0.81 for Blue, 0.28 and 0.45 for Intermediate, and 0.9 and 0.67 for Dark Brown. For skin classification, the best result was achieved by the experiment using SMOTE to deal with the class imbalance, using 56 SNPs. It was observed that 3 SNPs known to be directly related to skin pigmentation were not selected in the list of 56 SNPs using SMOTE. A further investigation is necessary to understand if the SMOTE technique has affected somehow the selection of the SNPs. The precision and recall, on average, respectively, were 0.90 and 0.94 for White, 0.56 and 0.48 for Intermediate, and 0.85 and 0.79 for Brown.

For future work, a deep investigation of the Intermediate classes is necessary. Having a better understanding of which SNPs define the intermediate colors for eye and skin is crucial and still very necessary to have better solutions.

REFERENCES

ABDI, H.; WILLIAMS, L. J. Principal component analysis. **Wiley Interdisciplinary Reviews: Computational Statistics**, Wiley, v. 2, n. 4, p. 433–459, jun. 2010. Available from Internet: <https://doi.org/10.1002/wics.101>.

ABEEL, T. et al. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. **Bioinformatics**, v. 26, n. 3, p. 392–398, 11 2009. ISSN 1367-4803. Available from Internet: <https://doi.org/10.1093/bioinformatics/btp630>.

AHMAD, M.; JUNG, L. T.; BHUIYAN, A.-A. From DNA to protein: Why genetic code context of nucleotides for DNA signal processing? a review. **Biomedical Signal Processing and Control**, v. 34, p. 44–63, 2017. ISSN 1746-8094. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1746809417300046>.

ALZUBI, R. et al. A Hybrid Feature Selection Method for Complex Diseases SNPs. **IEEE Access**, v. 6, p. 1292–1301, 2018.

ALZUBI, R. et al. A hybrid feature selection method for complex diseases SNPs. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 1292–1301, 2018. Available from Internet: <https://doi.org/10.1109/access.2017.2778268>.

ANG, J. C. et al. Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 5, p. 971–989, sep 2016. Available from Internet: <https://doi.org/10.1109/tcbb.2015.2478454>.

BELLINGER, C.; SHARMA, S.; JAPKOWICZ, N. One-Class versus Binary Classification: Which and When? In: **2012 11th International Conference on Machine Learning and Applications**. IEEE, 2012. p. 102—-106. Available from Internet: <https://doi.org/10.1109\%2Ficmla.2012.212>.

BISHOYI, A. K. Fundamental of genetics. In: **General Biology**. 9th. ed. Oakville, ON, Canada: Arcler Education, 2021.

BOMMERT, A. et al. Benchmark for filter methods for feature selection in high-dimensional classification data. **Computational Statistics & Data Analysis**, Elsevier BV, v. 143, p. 106839, mar 2020. Available from Internet: <https://doi.org/10.1016/j.csda.2019.106839>.

BREIMAN, L. Random Forests. **Machine Learning**, Springer Science and Business Media LLC, v. 45, n. 1, p. 5–32, 2001. Available from Internet: <https://doi.org/10.1023/a:1010933404324>.

BROOKES, A. J. The essence of SNPs. **Gene**, Elsevier BV, v. 234, n. 2, p. 177–186, jul 1999. Available from Internet: <https://doi.org/10.1016/s0378-1119(99)00219-x>.

BURTON, A. L. **OLS (Linear) Regression**. Wiley, 2021. 509–514 p. Available from Internet: <https://doi.org/10.1002/9781119111931.ch104>.

CAI, Z. et al. Supervised class distribution learning for GANs-based imbalanced classification. In: **2019 IEEE International Conference on Data Mining (ICDM)**. [S.l.]: IEEE, 2019. p. 41–50.

CAI, Z. et al. Supervised Class Distribution Learning for GANs-Based Imbalanced Classification. In: **2019 IEEE International Conference on Data Mining (ICDM)**. [S.l.: s.n.], 2019. p. 41–50.

CHAITANYA, L. et al. The hirisplex-s system for eye, hair and skin colour prediction from dna: Introduction and forensic developmental validation. **Forensic Science International: Genetics**, v. 35, p. 123–135, 2018. ISSN 1872-4973. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1872497318302205>.

CHANG, H.-W. et al. Comparison of classification algorithms with wrapper-based feature selection for predicting osteoporosis outcome based on genetic factors in a taiwanese women population. **International Journal of Endocrinology**, Hindawi Limited, v. 2013, p. 1–8, 2013. Available from Internet: <https://doi.org/10.1155/2013/850735>.

CHENG, K. et al. Classification of imbalanced bioinformatics data by using boundary movement-based ELM. **Bio-Medical Materials and Engineering**, IOS Press, v. 26, n. s1, p. S1855–S1862, aug 2015. Available from Internet: <https://doi.org/10.3233%2Fbme-151488>.

CILIA, N. et al. An experimental comparison of feature-selection and classification methods for microarray datasets. **Information**, MDPI AG, v. 10, n. 3, p. 109, mar 2019. Available from Internet: <https://doi.org/10.3390/info10030109>.

CRAWFORD, C. A. Gene flow. In: **Principles of biology**. 1st. ed. New York, NY: Salem Press, 2017, (Principle of Science). p. 255–262.

CRESWELL, A. et al. Generative Adversarial Networks: An Overview. **IEEE Signal Processing Magazine**, Institute of Electrical and Electronics Engineers (IEEE), v. 35, n. 1, p. 53–65, jan 2018. Available from Internet: <https://doi.org/10.1109\%2Fmsp.2017.2765202>.

DOMINGOS, P. MetaCost: A general method for making classifiers cost-sensitive. In: **Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 1999. p. 155–164.

DOUZAS, G.; BACAO, F. Effective data generation for imbalanced learning using conditional generative adversarial networks. **Expert Systems with Applications**, Elsevier BV, v. 91, p. 464–471, jan 2018. Available from Internet: <https://doi.org/10.1016\%2Fj.eswa.2017.09.030>.

ELRAHMAN, S. M. A.; ABRAHAM, A. A review of class imbalance problem. **Journal of Network and Innovative Computing**, v. 1, n. 2013, p. 332–340, 2013.

ERTEKIN, S. et al. Learning on the Border: Active Learning in Imbalanced Data Classification. In: **Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2007. p. 127—-136. ISBN 9781595938039. Available from Internet: <https://doi.org/10.1145/1321440.1321461>.

ESPOSITO, C. et al. Ghost: Adjusting the decision threshold to handle imbalanced data in machine learning. **Journal of Chemical Information and Modeling**, v. 61, n. 6, p. 2623–2640, 2021. Available from Internet: <https://doi.org/10.1021/acs.jcim.1c00160>.

FACELI, K. et al. Modelos preditivos. In: **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**. 2nd. ed. [S.l.]: LTC, 2011. p. 49–164.

FITZPATRICK, T. B. The validity and practicality of sun-reactive skin types i through vi. **Archives of dermatology**, American Medical Association, v. 124, n. 6, p. 869–871, 1988.

FIX, E.; HODGES, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. **International Statistical Review/Revue Internationale de Statistique**, JSTOR, v. 57, n. 3, p. 238–247, 1989.

FLEURET, F. Fast binary feature selection with conditional mutual information. **Journal of Machine learning research**, v. 5, n. 9, p. 1531—-1555, 2004.

FOWLER, J. C. S. et al. Repetitive deoxyribonucleic acid (dna) and human genome variation—a concise review relevant to forensic biology. **Journal of Forensic Science**, ASTM International, v. 33, n. 5, p. 1111–1126, 1988.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.

GABRIEL, S.; ZIAUGRA, L.; TABBAA, D. Snp genotyping using the sequenom massarray iplex platform. **Current protocols in human genetics**, Wiley Online Library, v. 60, n. 1, p. 2–12, 2009.

GALAR, M. et al. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 42, n. 4, p. 463–484, 2011.

GANAIE, M. A. et al. Ensemble deep learning: A review. **arXiv preprint arXiv:2104.02395**, 2021.

GENG, X.; YU-QUAN, Z.; YANG, Z. A novel classification method for class-imbalanced data and its application in microrna recognition. **International Journal Bioautomation**, Bulgarska Akademiya na Naukite/Bulgarian Academy of Sciences, v. 22, n. 2, p. 133, 2018.

GOODFELLOW, I. et al. Generative Adversarial Nets. In: GHAHRAMANI, Z. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2014. v. 27, p. 2672—-2680. Available from Internet: <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.

GRISCI, B. I.; KRAUSE, M. J.; DORN, M. Relevance aggregation for neural networks interpretability and knowledge discovery on tabular data. **Information Sciences**, Elsevier BV, v. 559, p. 111–129, jun. 2021. Available from Internet: <https://doi.org/10.1016/j.ins.2021.01.052>.

GUAN, H.; ZHANG, C. Predicting Diabetes in Imbalanced Datasets Using Neural Networks. In: **Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics**. New York, NY, USA: Association for Computing Machinery, 2022. (BCB 22). ISBN 9781450393867. Available from Internet: <https://doi.org/10.1145/3535508.3545540>.

GUI, J. et al. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. **IEEE Transactions on Knowledge and Data Engineering**, p. 1–1, 2021. Available from Internet: <https://arxiv.org/abs/2001.06937>.

GUO, X. et al. On the class imbalance problem. In: **2008 Fourth International Conference on Natural Computation**. IEEE, 2008. p. 192–201. Available from Internet: <https://doi.org/10.1109\%2Ficnc.2008.871>.

GUYON, I. et al. Gene selection for cancer classification using support vector machines. **Machine learning**, Springer, v. 46, n. 1, p. 389–422, 2002.

HAIXIANG, G. et al. Learning from class-imbalanced data: Review of methods and applications. **Expert Systems with Applications**, v. 73, p. 220–239, 2017. ISSN 0957-4174. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0957417416307175>.

HART, K. L. et al. Improved eye- and skin-color prediction based on 8 SNPs. **Croatian Medical Journal**, Croatian Medical Journals, v. 54, n. 3, p. 248–256, jun 2013. Available from Internet: <https://doi.org/10.3325/cmj.2013.54.248>.

HART, P. The condensed nearest neighbor rule. **IEEE transactions on information theory**, Citeseer, v. 14, n. 3, p. 515–516, 1968.

HASIBUAN, L. S.; KUSUMA, W. A.; SUWAMO, W. B. Identification of single nucleotide polymorphism using support vector machine on imbalanced data. In: **2014 International Conference on Advanced Computer Science and Information System**. [S.l.]: IEEE, 2014. p. 375–379.

HE, H. et al. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE. **2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)**. [S.l.], 2008. p. 1322–1328.

HEARST, M. A. et al. Support vector machines. **IEEE Intelligent Systems and their Applications**, v. 13, n. 4, p. 18–28, 1998.

HEMPSTALK, K.; FRANK, E.; WITTEN, I. H. One-class classification by combining density and class probability estimation. In: SPRINGER. **Joint European Conference on Machine Learning and Knowledge Discovery in Databases**. [S.l.], 2008. p. 505–519.

HERTEL, J.; HOFACKER, I. L.; STADLER, P. F. Snoreport: computational identification of snornas with unknown targets. **Bioinformatics**, Oxford University Press, v. 24, n. 2, p. 158–164, 2008.

HOLT, C. A.; ROTH, A. E. The nash equilibrium: A perspective. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy

of Sciences, v. 101, n. 12, p. 3999–4002, mar 2004. Available from Internet: <https://doi.org/10.1073\%2Fpnas.0308738101>.

IONITA-LAZA, I. et al. A spectral approach integrating functional genomic annotations for coding and noncoding variants. **Nature genetics**, Nature Publishing Group, v. 48, n. 2, p. 214–220, 2016.

JIANG, L. et al. Prediction of SNP sequences via gini impurity based gradient boosting method. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 7, p. 12647–12657, 2019. Available from Internet: <https://doi.org/10.1109/access.2019.2893269>.

KATSARA, M.-A. et al. Evaluation of supervised machine-learning methods for predicting appearance traits from DNA. **Forensic Science International: Genetics**, Elsevier BV, v. 53, p. 102507, jul 2021. Available from Internet: <https://doi.org/10.1016/j.fsigen.2021.102507>.

KAUR, H.; PANNU, H. S.; MALHI, A. K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 52, n. 4, p. 1–36, jul 2020. Available from Internet: <https://doi.org/10.1145\%2F3343440>.

KIRCHER, M. et al. A general framework for estimating the relative pathogenicity of human genetic variants. **Nature genetics**, Nature Publishing Group, v. 46, n. 3, p. 310–315, 2014.

KUBAT, M.; MATWIN, S. Addressing the curse of imbalanced training sets: One-Sided Selection. In: CITESEER. **Proceedings of the Fourteenth International Conference on Machine Learning**. [S.l.], 1997. v. 97, n. 1, p. 179–186.

KUKLA-BARTOSZEK, M. et al. Searching for improvements in predicting human eye colour from DNA. **International Journal of Legal Medicine**, Springer Science and Business Media LLC, v. 135, n. 6, p. 2175–2187, jul 2021. Available from Internet: <https://doi.org/10.1007/s00414-021-02645-5>.

KWOK, P.-Y. SNPs: Why Do We Care? In: **Single Nucleotide Polymorphisms : methods and protocols**. 1st. ed. Totowa, N.J: Humana Press, 2003. p. 1–11. ISBN 978-1-59259-327-9.

LAZAR, C. et al. A survey on filter techniques for feature selection in gene expression microarray analysis. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, n. 4, p. 1106–1119, jul 2012. Available from Internet: <https://doi.org/10.1109/tcbb.2012.33>.

LIU, D. **Handbook of nucleic acid purification**. [S.l.]: CRC Press, 2009.

MA, M. et al. Disease-associated variants in different categories of disease located in distinct regulatory elements. **BMC genomics**, BioMed Central, v. 16, n. 8, p. 1–13, 2015.

MARIANI, G. et al. Bagan: Data augmentation with balancing GAN. **arXiv preprint arXiv:1803.09655**, 2018.

MCDONALD, G. C. Ridge regression. **Wiley Interdisciplinary Reviews: Computational Statistics**, Wiley, v. 1, n. 1, p. 93–100, jul. 2009. Available from Internet: <https://doi.org/10.1002/wics.14>.

MIRZA, M.; OSINDERO, S. Conditional generative adversarial nets. **arXiv preprint arXiv:1411.1784**, 2014.

MITTAG, F.; RöMER, M.; ZELL, A. Influence of feature encoding and choice of classifier on disease risk prediction in genome-wide association studies. **PLOS ONE**, Public Library of Science (PLoS), v. 10, n. 8, p. e0135832, aug 2015. Available from Internet: <https://doi.org/10.1371/journal.pone.0135832>.

MUNEEB, M.; HENSCHEL, A. Eye-color and type-2 diabetes phenotype prediction from genotype data using deep learning methods. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 22, n. 1, p. 1–26, apr 2021. Available from Internet: <https://doi.org/10.1186/s12859-021-04077-9>.

MUTHUKRISHNAN, R.; ROHINI, R. LASSO: A feature selection technique in predictive modeling for machine learning. In: **2016 IEEE International Conference on Advances in Computer Applications (ICACA)**. IEEE, 2016. p. 18–20. Available from Internet: <https://doi.org/10.1109/icaca.2016.7887916>.

MüLLER, A. C.; GUIDO, S. Supervised Learning. In: **Introduction to Machine Learning with Python**. 1st. ed. [S.l.]: O'Reilly Media, 2017. p. 25–119.

NAQA, I. E.; MURPHY, M. J. What Is Machine Learning? In: ____. **Machine Learning in Radiation Oncology: Theory and Applications**. Cham: Springer International Publishing, 2015. p. 3–11. ISBN 978-3-319-18305-3. Available from Internet: <https://doi.org/10.1007/978-3-319-18305-3_1>.

NITESH, C. V. et al. Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

NOWOZIN, S. Improved information gain estimates for decision tree induction. **arXiv preprint arXiv:1206.4620**, 2012.

O'NEIL, C.; SCHUTT, R. Feature Selection. In: **Doing Data Science**. 3rd. ed. [S.l.]: O'Reilly Media, 2014. p. 176–193.

OTAKA, I. et al. Simple and inexpensive software designed for the evaluation of color. **American journal of ophthalmology**, Elsevier, v. 133, n. 1, p. 140–142, 2002.

PARRA, F. C. et al. Color and genomic ancestry in brazilians. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 100, n. 1, p. 177–182, 2003.

PENG, H.; LONG, F.; DING, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 27, n. 8, p. 1226–1238, 2005.

POZZOLO, A. D. et al. Credit card fraud detection: a realistic modeling and a novel learning strategy. **IEEE transactions on neural networks and learning systems**, IEEE, v. 29, n. 8, p. 3784–3797, 2017.

PROTOCOLS iPLEX P. R. Available from Internet: <https://agenacx.com/wp-content/uploads/2017/12/iPLEX-Pro-Reagents-User-Guide-1.pdf>.

QIAN, Y. et al. A resampling ensemble algorithm for classification of imbalance problems. **Neurocomputing**, v. 143, p. 57–67, 2014. ISSN 0925-2312. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0925231214007644>.

RISH, I. et al. An empirical study of the naive bayes classifier. In: **IJCAI 2001 workshop on empirical methods in artificial intelligence**. [S.l.: s.n.], 2001. v. 3, n. 22, p. 41–46.

RITCHIE, G. R. et al. Functional annotation of noncoding sequence variants. **Nature methods**, Nature Publishing Group, v. 11, n. 3, p. 294–296, 2014.

ROBERTS, A. et al. Inferring missing genotypes in large SNP panels using fast nearest-neighbor searches over sliding windows. **Bioinformatics**, Oxford University Press (OUP), v. 23, n. 13, p. i401–i407, jul 2007. Available from Internet: <https://doi.org/10.1093/bioinformatics/btm220>.

ROBNIK-ŠIKONJA, M.; KONONENKO, I. Theoretical and empirical analysis of relieff and rrelieff. **Machine learning**, Springer, v. 53, n. 1, p. 23–69, 2003.

SAEYS, Y.; INZA, I.; LARRANAGA, P. A review of feature selection techniques in bioinformatics. **Bioinformatics**, Oxford University Press (OUP), v. 23, n. 19, p. 2507–2517, aug 2007. Available from Internet: <https://doi.org/10.1093/bioinformatics/btm344>.

SALEHI, P.; CHALECHALE, A.; TAGHIZADEH, M. Generative adversarial networks (GANs): An overview of theoretical model, evaluation metrics, and recent developments. **arXiv preprint arXiv:2005.13178**, 2020.

SANTOS, F. H. K. T. dos; ARANHA, C. **Data Augmentation Using GANs**. arXiv, 2019. 1–16 p. Available from Internet: <https://arxiv.org/abs/1904.09135>.

SAXENA, D.; CAO, J. Generative adversarial networks (GANs): Challenges, solutions, and future directions. **ACM Computing Surveys**, Association for Computing Machinery (ACM), v. 54, n. 3, p. 1–42, apr 2022. Available from Internet: <https://doi.org/10.1145\%2F3446374>.

SCHUBACH, M. et al. Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants. **Scientific reports**, Nature Publishing Group, v. 7, n. 1, p. 1–12, 2017.

SEIFFERT, C. et al. Rusboost: A hybrid approach to alleviating class imbalance. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, v. 40, n. 1, p. 185–197, 2010.

SMEDLEY, D. et al. A whole-genome analysis framework for effective identification of pathogenic regulatory variants in mendelian disease. **The American Journal of Human Genetics**, Elsevier BV, v. 99, n. 3, p. 595–606, sep. 2016. Available from Internet: <https://doi.org/10.1016/j.ajhg.2016.07.005>.

SMITH, J. W. et al. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In: AMERICAN MEDICAL INFORMATICS ASSOCIATION. **Proceedings of the annual symposium on computer application in medical care**. [S.l.], 1988. p. 261.

SPELMEN, V. S.; PORKODI, R. A review on handling imbalanced data. In: IEEE. **2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)**. [S.l.], 2018. p. 1–11.

SYLVESTER, E. V. A. et al. Applications of random forest feature selection for fine-scale genetic population assignment. **Evolutionary Applications**, Wiley, v. 11, n. 2, p. 153–165, sep 2017. Available from Internet: <https://doi.org/10.1111/eva.12524>.

TONG, S.; KOLLER, D. Support vector machine active learning with applications to text classification. **Journal of machine learning research**, v. 2, n. Nov, p. 45–66, 2001.

USAI, M. G.; GODDARD, M. E.; HAYES, B. J. Lasso with cross-validation for genomic selection. **Genetics Research**, Cambridge University Press, v. 91, n. 6, p. 427–436, 2009.

V, A. C.; PUTHIYEDTH, N.; R, N. Feature selection methods for SNP analysis. In: **2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)**. IEEE, 2019. v. 1, p. 87–93. Available from Internet: <https://doi.org/10.1109/icicict46008.2019.8993273>.

VALENZUELA, R. K. et al. Predicting phenotype from genotype: Normal pigmentation. **Journal of Forensic Sciences**, Wiley, v. 55, n. 2, p. 315–322, mar. 2010. Available from Internet: <https://doi.org/10.1111/j.1556-4029.2009.01317.x>.

WALSH, S. et al. Global skin colour prediction from DNA. **Human Genetics**, Springer Science and Business Media LLC, v. 136, n. 7, p. 847–863, may 2017. Available from Internet: <https://doi.org/10.1007/s00439-017-1808-5>.

WALSH, S. et al. IrisPlex: A sensitive DNA tool for accurate prediction of blue and brown eye colour in the absence of ancestry information. **Forensic Science International: Genetics**, Elsevier BV, v. 5, n. 3, p. 170–180, jun 2011. Available from Internet: <https://doi.org/10.1016/j.fsigen.2010.02.004>.

WALSH, S. et al. The HIrisPlex system for simultaneous prediction of hair and eye colour from DNA. **Forensic Science International: Genetics**, Elsevier BV, v. 7, n. 1, p. 98–115, jan. 2013. Available from Internet: <https://doi.org/10.1016/j.fsigen.2012.07.005>.

WANG, K. et al. Generative adversarial networks: introduction and outlook. **IEEE/CAA Journal of Automatica Sinica**, v. 4, n. 4, p. 588–598, 2017.

WANG, Q. A hybrid sampling SVM approach to imbalanced data classification. **Abstract and Applied Analysis**, Hindawi Limited, v. 2014, p. 1–7, 2014. Available from Internet: <https://doi.org/10.1155\%2F2014%2F972786>.

WU, D. et al. Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. **Neurocomputing**, v. 190, p. 35–49, 2016. ISSN 0925-2312. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0925231216000072>.

XANTHOPOULOS, P.; PARDALOS, P. M.; TRAFALIS, T. B. Linear discriminant analysis. In: ____. **Robust Data Mining**. New York, NY: Springer New York, 2013. p. 27–33. ISBN 978-1-4419-9878-1. Available from Internet: <https://doi.org/10.1007/978-1-4419-9878-1_4>.

XIA, Y. Correlation and association analyses in microbiome study integrating multiomics in health and disease. In: **Progress in Molecular Biology and Translational Science**. Elsevier, 2020. v. 171, p. 309–491. Available from Internet: <https://doi.org/10.1016/bs.pmbts.2020.04.003>.

XIONG, M.; FANG, X.; ZHAO, J. Biomarker identification by feature wrappers. **Genome Research**, Cold Spring Harbor Laboratory, v. 11, n. 11, p. 1878–1887, nov 2001. Available from Internet: <https://doi.org/10.1101/gr.190001>.

XUE, C. et al. Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. **BMC Bioinformatics**, Springer Science and Business Media LLC, v. 6, n. 1, dec. 2005. Available from Internet: <https://doi.org/10.1186/1471-2105-6-310>.

YANG, P. et al. Stability of feature selection algorithms and ensemble feature selection methods in bioinformatics. **Biological knowledge discovery handbook: Preprocessing, mining and postprocessing of biological data**, John Wiley & Sons, v. 23, p. 333, 2013.

YU, H. et al. Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data. **Knowledge-Based Systems**, Elsevier BV, v. 76, p. 67–78, mar 2015. Available from Internet: <https://doi.org/10.1016\%2Fj.knosys.2014.12.007>.

YU, L.; LIU, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: **Proceedings of the 20th international conference on machine learning (ICML-03)**. [S.l.: s.n.], 2003. p. 856–863.

ZAORSKA, K.; ZAWIERUCHA, P.; NOWICKI, M. Prediction of skin color, tanning and freckling from DNA in polish population: linear regression, random forest and neural network approaches. **Human Genetics**, Springer Science and Business Media LLC, v. 138, n. 6, p. 635–647, apr 2019. Available from Internet: <https://doi.org/10.1007/s00439-019-02012-w>.

ZHANG, S. et al. A novel k NN algorithm with data-driven k parameter computation. **Pattern Recognition Letters**, Elsevier BV, v. 109, p. 44–54, jul 2018. Available from Internet: <https://doi.org/10.1016/j.patrec.2017.09.036>.

ZHOU, J.; TROYANSKAYA, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. **Nature methods**, Nature Publishing Group, v. 12, n. 10, p. 931–934, 2015.

ZHOU, N.; WANG, L. A modified t-test feature selection method and its application on the HapMap genotype data. **Genomics, Proteomics & Bioinformatics**, Elsevier BV, v. 5, n. 3-4, p. 242–249, 2007. Available from Internet: <https://doi.org/10.1016/s1672-0229(08)60011-x>.