UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARIA CECÍLIA MATOS CORRÊA

Comparing classification methods for point of interest categorization

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

Advisor: Prof[a]. Dr. Viviane Moreira
Coadvisor: M.Sc. Luciana Bencke

Porto Alegre
October 2022

“The end is not an apocalyptic explosion.
There may be nothing so quiet as the end.”
— MILAN KUNDERA

ABSTRACT

Urban geography is of fundamental importance for the understanding of space and the way in which it is transformed. In order to analyse the urban space, detailed land use data is an essential resource. Since 2011 the Brazilian Institute of Geography and Statistics has made available around 78 million records of addresses with land use descriptions in natural language. In this work, we compared different methods to automatically classify these records according to the economic activity they perform based on the short natural language descriptions. These descriptions are short, ambiguous, and often misspelled – posing challenges to classification algorithms. The classification methods we developed include a rule-based classifier that relies on human intervention and four ML classifiers that learn from training data. Our main research question is whether the ML classifiers can achieve a performance that is close to the rule-based classifier's. The results of our experiments using 41 classes showed that a classifier built using a state-of-the-art language model was able to achieve results that are not statistically different from the results of the rule-based classifier.

Keywords: CNEFE. classification. natural language processing. BERT. mapping. census. IBGE.

Comparação de métodos de classificação de texto para a categorização de pontos de interesse do CNEFE

RESUMO

A geografia urbana é de fundamental importância para a compreenção do espaço e da maneira como ele se transforma. Para conceber os estudo do espaço urbano, dados detalhados do uso do solo são um recurso essencial. Desde 2011 o Instituto Brasileiro de Geografia e Estatística (IBGE) tem disponibilizado cerca de 78 milhões de registros de endereços com descrição em linguagem natural do uso da terra. Neste trabalho, comparamos diferentes de métodos para classificar esses registros de acordo com a atividade econômica que eles exercem baseados nas curtas descrições em linguagem natural. Essas descrições são curtas, ambíguas e frequentemente possuiem erros ortográficos – apresentando desafios para os algoritmos de classificação. Os métodos de classificação desenvolvidos incluem um classificador baseado em regras heurísticas que necessita de intervenção humana e quatro classificadores de aprendizado de máquina que aprendem a partir dos dados de treinamento. Nossa principal questão é se as abordagens dos classificadores de aprendizado de máquina conseguem atingir uma performance que se aproxima do classificador baseado em regras. A possibilidade de classificar os dados com abordagens de aprendizado de máquina Os resultados dos experimentos usando 41 classes mostraeam que um classificador construído usando um modelo de linguagem de do estado-da-arte foi capaz de alcançar resultados que não são estatisticamente diferentes dos resultados do classificador baseado em regras.

Palavras-chave: CNEFE, classificação, processamento de linguagem natural, BERT, mapeamento, censo, IBGE.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag of Words
CNAE	Classificação Nacional de Atividades Econômicas
CNEFE	Cadastro Nacional de Endereços para Fins Estatísticos
ELMo	Embeddings from Language Models
F1	Weighted F-measure
IBGE	Instituto Brasileiro de Geografia e Estatística
IPEA	Instituto de Pesquisa Econômica Aplicada
LSTM	Long-Short Term Memory
M-BERT	Multilingual BERT
ML	Machine Learning
NLP	Natural Language Processing
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency

CONTENTS

1 INTRODUCTION

In 2018, the United Nations stated that 55% of the world's population lives in urban areas (UN, 2018). One of the reasons people choose to live in cities is because there is usually better access to healthcare, education, employment, and efficient infrastructure, which positively affect their quality of life. To understand the dynamics and processes that are responsible for the production and reproduction of urban space, detailed land use data is of paramount importance.

The Brazilian Institute of Geography and Statistics (IBGE) performs a demographic census every ten years. In 2011, as a product of the 2010 census, CNEFE, the National Register of Addresses for Statistical Purposes (IBGE, 2011), was released, with around 78 million urban and rural addresses. This data provides information on the locations of various activities for the entire country.

Every record on CNEFE holds a description of the land use in natural language, as well as an identification code that classifies the land into seven groups (*e.g.,* residential units, teaching units, and health units, among others). All of this was integrated in 2010 into a series of urban and rural maps, the Territorial Base, which even made georeferencing possible. It is through the census that Brazilian public policies are conceived, and with the CNEFE data, the horizon is even broader. More than a way of knowing how many residences would be affected in the event of major infrastructure works or natural disasters, the land use description could closely map out the country's economic activity.

In order to extract this knowledge from the CNEFE data, the records need to be classified. Manually classifying the entire database is unfeasible. Thus, employing an automatic classification scheme becomes necessary.

The goal of this work is to design, implement, and evaluate classification models for CNEFE data. In order to achieve that, this work employed a process that started with the definition of the classes of interest, followed by the manual labeling of a sample of records with their class. The result was an annotated dataset with 44k instances together with their ground truth classes. This dataset was used to create different classification models using supervised machine learning (ML) algorithms. Both traditional algorithms (such as random forests, logistic regression, and support vector machines) and state-of-the-art natural language processing models (such as BERT) were implemented. In addition, during the labeling process, the human annotators also identified words and expressions that were associated with each class (*i.e.,* creating a lexicon). The lexicon

was used to create a rule-based classification system.

Classifying records based on their natural language description is a very challenging task for a number of reasons. Misspellings are one of the most prevalent issues, affecting a large portion of the records. On many occasions, descriptions are ambiguous or they may be inconsistent with their identification code. To make matters worse, all these problems can occur together in a single record.

We carried out experiments comparing the quality of the different classification algorithms in assigning the correct class label among 41 possibilities. The results in terms of macro-F1 score ranged between 65 and 71%, and the best ML method was able to achieve a performance with difference not statistically significant from the rule-based classifier and. These scores were achieved despite the dataset being very unbalanced. The results also showed that some classes can be very difficult to learn. This was the case of the undefined class (designed for descriptions too vague or impossible to understand) or the retail trade classes that have too many different patterns. It is also noted the scarceness of instances in some classes penalizes the classifiers.

The CNEFE data classified by our algorithms are available at the project's repository[1], along with the experiment implementations.

The remainder of this work is organized as follows: Chapter 2 covers the background work on the techniques used to process and classify our data. Chapter 3 reports on related work in real-world data classification and imbalanced dataset evaluations. Chapter 4 describes how training instances were generated and our general approach with classifiers. In Chapters 5 and 6, we present the evaluation and compare the classifiers. In Chapter 7, we summarize our observations and touch on directions for future work.

[1] <https://github.com/cixcore/cnefe-poi-classification>

2 BACKGROUND

In this chapter, we describe the techniques used to classify the points of interest. We will cover how to compare the distance between two strings, data representation models, and text classification algorithms.

2.1 String distance metrics

When dealing with human written datasets, it is common to find misspelled words. The CNEFE base was no different – mistakes ranged from simple missing letters, presumably due to lack of attention on the census taker part when writing things like "ESCRTORIO" instead of "ESCRITORIO", to descriptions completely divergent from the standard norm, like "CANPO DE FOTIBAU" representing "CAMPO DE FUTEBOL". As humans, it is easy to figure out which words are correct and which are variants, because to us, it is clear how close they are. In this section, we cover some methods to automatically identify how distant different strings are.

2.1.1 Levenshtein Distance

Levenshtein or Edit Distance, presented by the mathematician Vladimir Levenshtein (LEVENSHTEIN et al., 1966), is a metric of how many operations are necessary to turn one string into another. The operations allowed are insertion, deletion, and substitution of characters, where each operation can have its own cost, usually 1 (NAVARRO, 2001). With two strings α, of length $|\alpha|$, and β, of length $|\beta|$, a naive recursive implementation of $lev(\ \alpha,\ \beta)$ can be defined by:

$$lev(\alpha,\beta) = \begin{cases} |\alpha|, & \text{if } |\beta| = 0 \\ |\beta|, & \text{if } |\alpha| = 0 \\ lev(tail(\alpha),\ tail(\beta)), & \text{if } |\alpha| = |\beta| \\ \min \begin{cases} lev(tail(\alpha),\ \beta) + cost_{insertion} \\ lev(\alpha,\ tail(\beta)) + cost_{deletion} \\ lev(tail(\alpha),\ tail(\beta)) + cost_{substitution} \end{cases}, & \text{otherwise} \end{cases} \tag{2.1}$$

To obtain the cost of editions, we work with a modified Levenshtein distance that considers the position of the two letters in a *QWERTY* keyboard when calculating the substitution cost. Thus, words written with misspelled characters that are very close to the correct one are more valued than characters that are far away, meaning they have a higher chance of being a misspelling of the same term. In addition, rather than using the raw distance values, it is usual to normalize them so that they yield a proportion that reflects the dissimilarity between the strings, *e.g*, the words *construcao* and *contrucao* are 10% dissimilar, considering the cost of deletion/insertion as 1, but the words *lancha* and *lanche* are 33% dissimilar, because the cost of replacing *a* for *e* is 2 in the character distance measure for the *QWERTY* keyboard.

2.1.2 Phonetic distance

Phonetic algorithms represent words by a codification of their pronunciation. This codification allows us to represent misspelled words that have variations of letters with equal phonetic value or the original, and by having the same codification, both terms are considered close. Soundex and Metaphone are two of the most notorious phonetic algorithms, and both were initially designed with heuristics for the English language.

Metaphone (PHILIPS, 1990) is actually an improvement on the Soundex algorithm. Essentially, it removes all vowels, except if it is the first letter, and encodes the consonants according to some pattern. The implementation in Brazilian Portuguese, for example, will encode a "G" if followed by "A", "O" or "U" to *G*, and to *J* if followed by "E" or "I". It can be used to assume that "CABELEIRO" should be the same as "CABELEIREIRO", but it will also assume that the term "VAZIO" is the same as "VISÃO". Table 2.1 shows some translations of the Metaphone for real terms appearing on CNEFE.

2.2 Representation

Word vectorization is how words and meanings are represented by our classifiers. It consists of encoding the words of the documents as vectors, and by projecting these words into a vector space. We find that words that share similar contexts usually have vectors close to each other. This form of semantics is the standard way to represent word

Table 2.1 – Phonetic representation with Brazilian Portuguese Metaphone

Term	*Phonetic representation*
xiqueiro	XKR
chiqueiro	XKR
xícara	XKR
escritório	ESRT
escrotório	ESRT
fotibau	FTB
futebol	FTB
futibol	FTB

Source: The Author

meaning in Natural Language Processing (JURAFSKY; MARTIN, 2009).

Vectors can be sparse and contain as little information as how many times a word appears on a document, which we call Bag-of-words (BOW), or even weights associated with the terms. A common algorithm used to generate weighted vectors is **TF-IDF**, based on term-document-matrices with weights that represent the importance of a word to a document. TF-IDF can be understood as, if a word appears many times within a document, it must have a meaningful relation to it, so it receives importance; however, many words are very common in many contexts and are of no good use to distinguish documents, so we use this frequency to reduce their importance. To generate the weights, the tokens of the documents are extracted to create a dictionary with a more efficient representation of these tokens. From the dictionary, the occurrences of the token are used to build the bags of words and subsequently calculate the TF-IDF weights.

Moreover, vectors can also be dense and contain shorter, but powerful representations, which is what we call embeddings. Fixed embeddings do not consider the context of words during vectorization. In this topic, Mikolov et al. (2013) came up with *word2vec*, an algorithm that represents words by a set of the other words that appear nearby in a given window size. This concept origins two models *Skip-gram*, which tries to predict the words in the set of an entry center word, and Continuous Bag of Words (CBOW), which does the opposite, predicting the center word having a bag of its context words.

One downside of word2vec is representing polysemic words because it represents them with the same vectors. To get around that, contextualized embeddings with ELMo (PETERS et al., 2018) were developed. ELMo comes from Embeddings from Language Models, and that is because it uses vectors derived from a bidirectional LSTM – the process of making a neural network have the sequence information both forward and backward – that is trained with a coupled language model (LM). A limitation of ELMo is

that, although it assimilates context information into word embeddings, it can not consider contextual information from both left-to-right and right-to-left at the same time. In Section 2.3.4, we introduce BERT, a new milestone after ELMo that is able to incorporate contextual information from both directions at the same time.

2.3 Classification Algorithms

According to Manning, Raghavan and Schütze (2009, Chapter 13), text classification is, given a set of target classes, the task of determining which class (or classes) a given object belongs to. It can be accomplished through manually labeling every object, though it is very expensive, or using standing queries, which can be seen as an assemblage of rules, also often manually designed. A third way of classifying objects, in our case, classifying text documents, is employing machine learning methods.

Here, we present supervised learning algorithms used to classify CNEFE's points of interest. Supervised learning algorithms do not discard the need for manual labeling, for they use manually annotated data to define the decision criteria of the classifiers. This building of decision criteria can be defined as, given a set of document-class pairs, which we call *training set*, we wish to generate a function λ that receives a document d and returns the class c it belongs to.

2.3.1 Logistic Regression

Logistic Regression classifiers are statistical models that fit the training instances by seeking to generate a linear combination that best represents the relation between the entry features and the class output. The goal of Logistic Regression is to define the best bias and combination of weights associated with each feature that minimizes the *Loss function* for all instances.

To train the model, we start from arbitrarily initialized values and iteratively adjust the weights using a *cost function*. To minimize the cost function, the *gradient descent* strategy is used.

The intuition behind gradient descent is that, in order to find a local minimum of a function, we can find out in which direction – in the space of the parameters – the function's slope is steepest, and move in that direction. The gradient of a function is a

vector pointing in the direction of the greatest increase in a function.

At every iteration, the updated weight is given by the previous weight minus a *learning rate* multiplied by the gradient. A higher value of learning rate means that we should move the w more on each step. If the learning rate is too high, the weight might change too much and miss the local minimum, never meeting convergence before the maximum defined iterations – we call this *overshooting*. Having as Loss function L that is represented by the difference between the real value y and the observed value from the combinations of weights w of each feature x in the function f, and a learning rate η Jurafsky and Martin (2009) define the updated weight as:

$$w_{t+1} = w_t - \eta \frac{d}{dw} L(f(x; w), y) \tag{2.2}$$

2.3.2 Random Forests

Random forests are a type of ensemble technique that combines the prediction of several weak learners, notably **decision trees**, in order to predict the class of the dataset. Weak learners are algorithms that tend to have high bias or variance, but accuracy at least better than random guessing, that can be combined to generate "strong" learners. Ensemble techniques usually combine this weak learners with one of three main strategies known as **bagging**, **boosting** and **stacking**. With bagging, the main goal is to reduce the variance of the model.

Random forests were properly introduced by Breiman (2001), and work by exposing different decision trees to different slices of the training data – either by bootstrap aggregation or by some other sampling method – during the learning phase, and combining the final prediction by voting for the most popular class among the decision trees. The voting can be uniform, where every classifier has the same decision-making power as the others, or weighted, where some votes have more influence than others. Random Forests can be compared to the concept of the Wisdom of Crowds, which refers to the knowledge that comes from a collective decision often performs better than one coming from a single individual, even if the individual is a specialist.

2.3.3 Support Vector Machines

Support Vector Machines, or SVMs, are methods based on statistical learning frameworks proposed by Cortes and Vapnik (1995). The SVM model maps training instances to non-linear points in space using its features, and then tries and trace vectors between the groups of points looking to maximize the width of the gap between each group, or rather, each category. To define the frontier of the groups, the algorithm generates **support vectors** and uses the support vectors to find the **optimal hyperplane**, that is the vector that generates the **optimal margin** – this is the concept of Maximal Margin Classifiers. New instances are then mapped into that space and predicted to belong to a category based on which side of the optimal hyperplane they fall.

In order to avoid errors generated by the Maximal Margin Classifiers' high sensitivity to outliers, SVM lets us define a "soft margin" that configures the trade-off between bias and variance, allowing the misclassification of outliers. This approach, the Support Vector Classifier, only models linearly separable instances, and to be able to perform non-linear classification, SVM also allows us to configure what is called the *kernel trick*. The kernel trick is a function that calculates the relation between each of the instance points, using the dot product, as if they were projected in a higher dimension space. This relation may be fitted into a Support Vector Classifier. A very common kernel used in SVM is the Radial Basis Function, a kernel able to work with infinite dimensions of features.

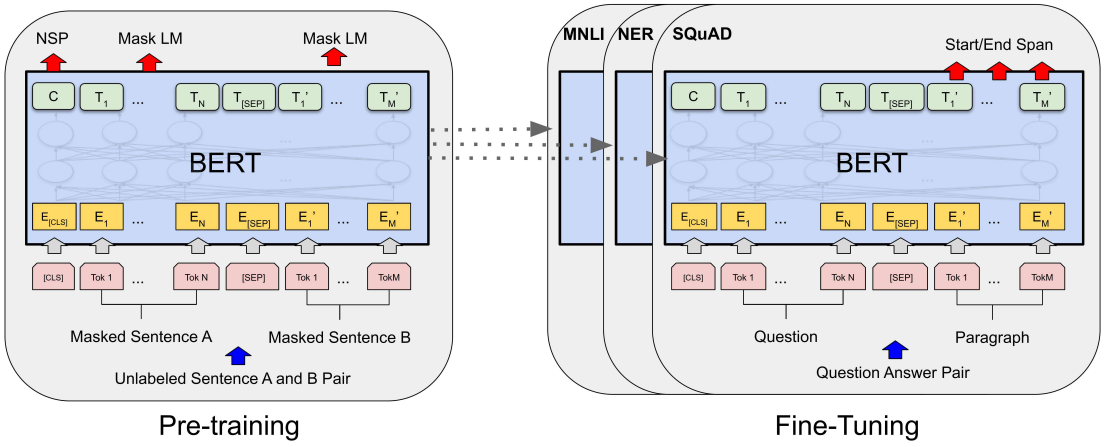
SVMs are known for their high generalization ability and robust results when applied to large datasets, as opposed to other methods that tend to under or overfit (LORENA et al., 2011).

2.3.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) is an open-source library developed by Google that obtained new state-of-the-art results on at least eleven NLP tasks (DEVLIN et al., 2018). Its model architecture is a multi-layer bidirectional Transformer encoder-based implementation proposed by Vaswani et al. (2017).

BERT introduces bidirectionality of context by using a *masked language model* that randomly masks some tokens from the input and tries to predict it again based only on its context; this provides us with a representation that merges the left and the right contexts. Subsequently from being unsupervisedly *pre-trained* with the masked model,

Figure 2.1 – Overall pre-training and fine-tuning procedures for BERT



Source: Devlin et al. (2018)

BERT is trained with a *next sentence prediction* task to learn relationships between sentences. The last step in the pre-training stage is to pre-train with data; BERT uses, in its release, the BooksCorpus[1] and English Wikipedia.

After the pre-training is done, BERT moves on to the stage of *fine-tuning*. This stage consists of fitting the pre-trained model to inputs from the specific task we are trying to accomplish. In this stage, it is possible to make the new fine-tuned model available to the community as a pre-trained exemplary to fit inputs from new tasks. A very useful pre-trained language model is M-BERT (Multilingual BERT), making for a straightforward solution to cross-language conversion. M-BERT is BERT trained with additional 102 languages.

2.4 CNEFE

In November 2011, IBGE launched the National Register of Addresses for Statistical Purposes (CNEFE). It is a registry with 78,056,411 urban and rural addresses that began to be produced in the 2000 Census, was improved in 2007, and was consolidated in the 2010 Census. In the beginning, the census takers had to work with printed maps and write down the addresses to apply the questionnaires as they traveled through the route. The great technological leap forward took place in the 2010 Census, when census takers went to the field with handheld computers containing the digital mesh of the urban sectors with addresses associated, and were able to update them. CNEFE is the first public

[1] <https://github.com/soskek/bookcorpus>

archive of its kind in the country, and the graphic files associated with the registry were made available by IBGE throughout 2012[2].

The CNEFE records are divided by Federative Unit and contain the municipality code, type of road, a code representing whether the address is urban (1) or rural (2), and full addresses organized by the parts that form it, like street name, complement, house number *etc.*. Besides, every entry contains a land use id that indicates in general lines the use of the location, like whether the address is a residential home or educational establishment. The ids are explained in Table 2.2.

Table 2.2 – Explanation of land use ids from CNEFE

Id	*Meaning*
1	private domicile
2	collective domicile
3	agricultural establishment
4	educational establishment
5	health establishment
6	other purposes establishment
7	building under construction

Source: The Author

Some ids have a direct relation to the final classes used in this work to label CNEFE. We can associate id 3 with the class designed for agriculture, livestock, forestry production, fisheries, and aquaculture, while id 4 is related to the Education label and id 5 to Human health and social services.

Lastly, the records have a short description written by the census takers in natural language. The descriptions have on average between 2 and 3 words and may repeat between entries, and some records are missing descriptions. There are even some examples of incomprehensible descriptions formed of single letters or numbers that not always can be defined by their land use id associated. In addition, many records have misspelled or very abbreviated words.

[2]Available at <http://www.censo2010.ibge.gov.br/cnefe/>

3 RELATED WORK

In this chapter, we discuss other works that focused on labeling real datasets by employing different classification algorithms.

The motivation for our work stems from the infeasibility of manually classifying the records from CNEFE. Surprisingly, Battistam (2015) shows that the data had a portion that was manually classified. In a work that took place between 2012 and 2015, the central areas of the cities Marília, São Carlos and São José do Rio Preto were labeled according to a similar structure we chose in this work, and that data was applied to produce thematic maps. This illustrates exactly what use can be given to the results we present. It also shows how burdensome it is to generate this classification manually, even for a reduced region.

Moving to automatic classification, Wei et al. (2018) compares two supervised ML algorithms (SVM and convolutional neural networks) on the task of classifying documents in the legal field. The problem presented in the work is a binary classification problem, and it focuses on comparing how the size of the training set affects the performance of the classifiers. The results show that, when the training sets had a size approximate to what we present in this work, the neural network classifier performed better on the dataset of legal documents.

Another way the training data size can impact predictions from supervised classifiers is when there is an imbalance in the number of instances that form the classes. Common methods for handling imbalanced datasets are *undersampling*, when a portion of the majority class is excluded from training, and *oversampling*, where the idea is to adjust the distribution of classes by artificially filling an under-represented class with data points. Both oversampling and undersampling involve introducing a bias to select more samples from one class than from another, to compensate for the data imbalance. Padurariu and Breaban (2019) explored different oversampling techniques and how data imbalance affects the classification of documents from the Human Resource area. It concludes that oversampling methods improve performance, and that data imbalance and text representation are correlated when it comes to the prediction performance of the classifiers.

Wei et al. (2018) and Padurariu and Breaban (2019) both test multiple classification algorithms like the present work. While Wei et al. (2018) only performs tests on binary classification problems, Padurariu and Breaban (2019) performed multi-class classification on imbalanced datasets, the same problems we face with the CNEFE data. This

work does not approach methods for balancing the dataset, nor did it compare different text representation algorithms for the same classifiers.

4 METHODOLOGY

The goal of this work is to automatically classify records in the CNEFE data based on their land use id and descriptions. More specifically, we are interested in addresses devoted to businesses, health, education, agriculture, public administration, and other economic activities.

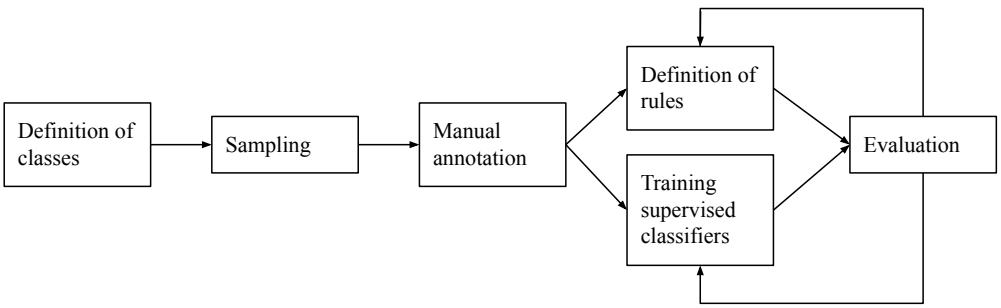
Our methodology includes the application of different types of classification algorithms, namely

- Rule-based Classifiers
- Supervised Classifiers
  - Traditional algorithms
  - BERT-based

Figure 4.1 shows an overview of the methodology employed in this work. The first step is to decide on the classes that will be used to annotate the instances. The classes are also used to define lexicons of words that can be used to identify the class. The lexicons are used by the rule-based classifier and the annotated instances are used to train supervised classifiers. Finally, we evaluate the quality of the classification models we obtained.

We provide further information on the performances in Chapter 6. The phases to accomplish the task of classifying CNEFE's points of interest are detailed in the following sections.

Figure 4.1 – Outline of the methodology for classifying points of interest



Source: The Author

4.1 Definition of classes

Before we are able to classify the instances, we need to know the class labels that will be assigned. CNEFE's dataset does not come with a set of target classes. Thus, our first task was to define those classes.

Initially, we considered using The British Ordnance Survey classification scheme[1]. Our motivation was that this scheme had already been used to classify points of interest. However, the Ordnance Survey does not represent well enough what we wished to map with CNEFE. For instance, on one hand, it presents excessive details on attractions, sports, and entertainment, and on the other hand, it does not have enough detail on retail options. Therefore, we decided to move on to another alternative: the National Classification of Economic Activities (CNAE) scheme[2]. CNAE is also maintained by IBGE and it is the Brazilian standardization of economic activity codes and schematization criteria.

CNAE has a hierarchical classification with 21 sections. Out of those, 17 were maintained as they are. The parent sections "G - COMÉRCIO; REPARAÇÃO DE VEÍCULOS AUTOMOTORES E MOTOCICLETAS", "H - TRANSPORTE, ARMAZENAGEM E CORREIO" and "I - ALOJAMENTO E ALIMENTAÇÃO" were replaced with their child divisions, adding 10 classes to the scheme. The internal divisions of "G.47 - COMÉRCIO VAREJISTA" had 8 subgroups that were also added. The section "T - SERVIÇOS DOMÉSTICOS" is simply removed. We also added the same two classes that were added before with the Ordnance Survey scheme, "IGREJAS, TEMPLOS E ATIVIDADES RELIGIOSAS" and "DESOCUPADO" in Portuguese. At last, the classes "OBRAS", designed for instances that were wrongfully not labeled as ongoing constructions by the land use id, and "NÃO DEFINIDO", for everything that simply can not be labeled, were introduced. With that, we have a total of 41 classes that can be seen in Table 4.1.

Table 4.1 – Distribution of annotated instances per class in $\mathcal{A}$ and $\mathcal{U}'$ in decreasing order of frequency in $\mathcal{A}$

Id	*Label*	$\mathcal{A}$	$\mathcal{U}'$
29	Alimentação	7,032	70

Table 4.1 – Continued on next page

[1] Available at <https://www.ordnancesurvey.co.uk/documents/product-support/support/points-of-interest-classification-scheme.pdf>. Last accessed on September 5th, 2022

[2] Available at <https://cnae.ibge.gov.br/?view=estrutura>. Last accessed on September 5th, 2022

Table 4.1 – Continued from previous page

Id	*Label*	$\mathcal{A}$	$\mathcal{U}'$
1	Agricultura, pecuária, produção florestal, pesca e aquicultura	6,221	65
7	Igrejas, templos e atividades religiosas	5,360	40
20	Comércio varejista de produtos novos não especificados anteriormente e de produtos usados	5,357	37
14	Comércio varejista de produtos alimentícios, bebidas e fumo	2,474	67
19	Comércio varejista de produtos farmacêuticos, perfumaria e cosméticos e artigos médicos, ópticos e ortopédicos	2,070	36
37	Saúde humana e serviços sociais	1,845	63
13	Comércio varejista de mercadorias em geral, com predominância de produtos alimentícios - minimercados, mercearias e armazéns	1,805	46
40	Outras atividades de serviços	923	97
8	Comércio e reparação de veículos automotores e motocicletas	921	61
36	Educação	911	61
38	Artes, cultura, esporte e recreação	870	56
21	Desocupado	846	48
9	Comércio por atacado, exceto veículos automotores e motocicletas	780	51
18	Comércio varejista de artigos culturais, recreativos e esportivos	679	33
30	Informação e comunicação	470	49
22	Transporte terrestre	425	37
17	Comércio varejista de equipamentos de informática e comunicação; equipamentos e artigos de uso doméstico	417	47
15	Comércio varejista de combustíveis para veículos automotores	414	25
28	Alojamento	403	39

Table 4.1 – Continued on next page

Table 4.1 – Continued from previous page

Id	*Label*	$\mathcal{A}$	$\mathcal{U}'$
11	Comércio varejista não especializado	377	46
16	Comércio varejista de material de construção	376	56
35	Administração pública, defesa e seguridade social	375	40
3	Indústrias de transformação	371	82
32	Atividades imobiliárias	369	47
6	Construção	349	30
31	Atividades financeiras, de seguros e serviços relacionados	321	39
25	Armazenamento e atividades auxiliares dos transportes	287	61
33	Atividades profissionais, científicas e técnicas	227	77
5	Água, esgoto, atividades de gestão de resíduos e descontaminação	215	44
34	Atividades administrativas e serviços complementares	141	35
4	Eletricidade e gás	112	29
10	Comércio varejista	65	32
2	Indústrias extrativas	57	27
0	Obras	56	28
27	Não definido	25	68
26	Correio e outras atividades de entrega	23	34
12	Comércio varejista de mercadorias em geral, com predominância de produtos alimentícios - hipermercados e supermercados	22	50
24	Transporte aéreo	19	23
39	Organismos internacionais e outras instituições extraterritoriais	15	35
23	Transporte aquaviário	6	48
Total		**44,031**	**1926**

Source: The Author

4.2 Sampling

As described in Section 2.4, the complete dataset has 78 million records. Our focus was on addresses that perform an economic activity. Thus, all instances in which the land use id corresponds to households (land use id = 1) or ongoing constructions (land use id = 7) were removed. The resulting dataset has 11 million instances. We will refer to this as our complete dataset $\mathcal{C}$.

It is unfeasible to manually label all these instances. Thus, we generated a representative sample by randomly selecting five percent of the addresses from each Federative Unit. We then removed duplicate land use descriptions that also have the same land use id. This sample was submitted to the manual annotation process described in Section 4.3.

4.3 Manual annotation

Once the classes were defined and a sample was selected, the annotation phase could start. As is expected of human-produced data, there are some inconsistencies and ambiguity in the records. Here, we mention how we addressed these issues.

4.3.1 Mismatched land use id and description

The land use ids and their labels are shown in Table 2.2. Looking at the data, we noticed that some instances have mismatched land use description and id, as can be seen in Table 4.2. For example, in the first row, the description refers to a barber shop but the id 3 is for agriculture. Similarly, in the second row, a bar was assigned the land use id of educational institutions. To solve this issue we decided that the land use description takes precedence over the id.

4.3.2 Multiple purposes in the same address

Another issue is that some addresses have more than one use, for example, "CONSULTORIO ODONTOLOGICO E BAR" (*dental office and bar*) that could go either into classes or 29 or 37, or "BAZAR E FERRAGEM" (*bazaar and hardware store*), that also could go into classes 16 and 20.

Handling these multiple purposes in the same address would require using multi-label classification algorithms. However, multi-label classification presents additional challenges, such as an exponential number of possible label sets and capturing dependencies between labels. As a result, at this moment, we modeled the task as a *single-label multi-class classification problem*. To deal with instances with more than one purpose, in the manual annotation phase, we adopted the convention that the class would be that of the first use appearing in the description.

Table 4.2 – Example of CNEFE instances with mismatched land use description and id

Description	*Actual id*	*Expected id*	*Possible classes*
BARBEARIA	3	6	1, 40 (agriculture, other service activities)
BAR	4	6	29, 36 (eating places, education)
ABATEDOURO	5	6	3, 37 (manufacturing industries, human health and social services)
ATO ESCOLA	6	4	36 (only education, but wrong id informed)

Source: The Author

4.3.3 Unclear class coverage

An even more complex issue is to do with what exactly is covered in each of the classes. Should transport and logistic services provided by shipping companies go into class 22 (Ground transportation), 25 (Storage and auxiliary transport activities), or 26 (Mail and delivery services)? Pet shops, that in Brazil usually sell items for animals, but can also provide bathing and grooming services, would go into class 20 or 40? Does a supermarket warehouse belong in a supermarket category or in a warehouse category? Cases like this, in the end, are inherently ambiguous and can only be solved with cautious discussion over what is more significant when using this data for mapping. After discussion, we assigned shipping companies to the *ground transportation* class (label 22), pet shops to the *other new and second-hand product retail* (label 22) and supermarket warehouses to class 12 of supermarkets – in this case, if the economic activity the warehouses serves to can not be defined, than it would go to class 25 of storage activities.

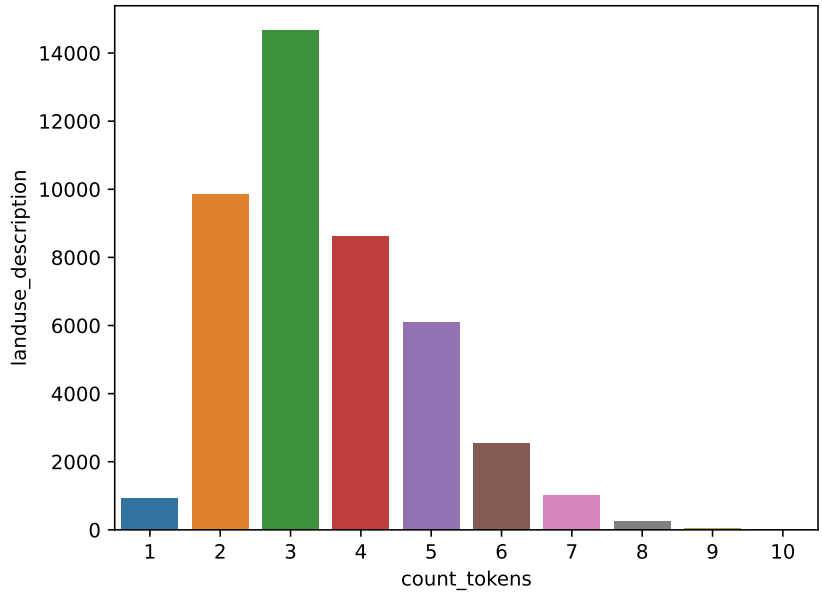
4.3.4 Vague descriptions

The data also contains vague descriptions. There is an average of 2.5 words per description, and that very short length indicates the potential for lack of information.

Figure 4.2 shows the distribution of *tokens* by description in the annotated dataset.

An example of a vague description in CNEFE is "TUBO EXTRA" (*extra tube*): it seems to express some relation to tubes or pipes, but even if this assumption is correct, is this a store or a distributor? Or does it provide plumbing services? That is definitely too little information on the description to enable us to classify them with certainty. In this case, we simply decided to label it as *undefined*. Several instances required deliberation to obtain their final label.

Figure 4.2 – Distribution of tokens per description



Source: The Author

4.3.5 The final datasets

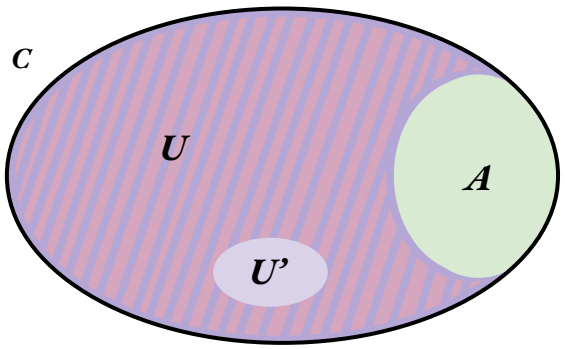
The labeling process was carried out by five annotators who interacted to solve disagreements and try to standardize the annotation methodology. At the end of the process, nearly 44 thousand instances were manually labeled with their *ground truth* classes, making up our annotated dataset $\mathcal{A}$.

Finally, human annotation was also carried out on a new sample taken from $\mathcal{C}$ completely unseen before by the human annotators and by the classifiers. This sample is referred to as $\mathcal{U}'$. It has 1926 instances. The process for generating this sample is

described in Section 4.6 as it has to do with the outputs of the classification algorithms.

The relationship between the datasets can be seen in Figure 4.3. $\mathcal{C}$ is the complete dataset with 11 million instances which includes all other datasets. $\mathcal{A}$ and $\mathcal{U}$ are disjoint samples (*i.e.,* $\mathcal{A} \cap \mathcal{U} = \emptyset$). $\mathcal{U}'$ is a subset of $\mathcal{U}$.

Figure 4.3 – Diagram showing the relationships between the datasets



Source: The Author

Details on the annotated datasets $\mathcal{A}$ and $\mathcal{U}'$ are shown in Table 4.1. We can see that the most common class in $\mathcal{A}$, *i.e.* "29 - Alimentação" (Eating and drinking) has 7,032 instances, while the least common class, *i.e.* "23 - Transporte Aquaviário" (Water transportation), has only six annotated instances. There was an effort into trying not to have such unbalanced classes, but this distribution reflects what we find in the real world. Dataset $\mathcal{U}'$ is much more balanced because the selection process (explained in Section 4.6.2) aimed for a more balanced distribution by attempting to extract the same number of instances for every predicted class.

4.4 Designing a Rule-Based Classifier

The annotation rules used in the rule-based classifier are based on string matching that is built upon knowledge acquired from the manual annotation phase. The functions first sanitize the land use descriptions and then try to find patterns. Sanitization consists in transforming the description to lower case and replacing some symbols, like correcting numbers that were meant to be used as letters – the word "oficina", for example, appeared as "0ficina" in some records, with a zero instead of an 'o' – and also removing diacritics. The pattern matching step is designed as a series of methods that are executed one after another, in order of precedence.

When creating the pattern-matching methods, it is important to pay attention to

overlapping patterns. In the dataset, cases of words belonging to the contexts of multiple target classes are not rare – the word "construção" (*construction*) appears in "materiais de construção" (*building materials*), class of building material retail trade, and can also appear in "casa em construção" (*house under construction*), which belongs to the ongoing constructions class. Thus, it is crucial to get the precedence of these overlapping patterns correctly. Bearing this in mind, the highest priority function was assigned the rule that matches terms like "vazio", "antiga", "desocupado", "a venda" (*empty, old, vacant, for sale*), and classifies them as belonging to class 21 for vacant addresses.

As we have already mentioned in previous examples, there are several misspelled words in the dataset. Some functions try and apply a few known variations of the words in the lexicon, but it is not feasible to detect all possible ways a word can be misspelled. To address this issue, we relied on string distance metrics (described in Section 2.1.1). The string distance metrics are added with the least priority to avoid errors. The order was set to first execute all functions based on known terms and expressions, then the functions considering edit distance, and finally the functions considering the phonetic distance.

Rather than applying the raw distance functions, we used their normalized versions (as described in subsection 2.1.1). Whenever two words are at most 25% dissimilar, then they are considered a match. Many words had missing or misplaced letters and edit distance works very well in those cases. Phonetic distance, on the other hand simply encodes the words of the description with the words associated with a class using the Metaphone algorithm for Brazilian Portuguese, if they have the same codification, then they are considered a match. In cases of misspellings where the wrong letters are picked to represent a given phoneme, both words will end up with the same phonetic codification.

At the end of the process, we ended up with 122 labeling functions. The more functions we got, the more difficult it became to maintain the quality of the results. Each new term added to a function can have side effects that can only be discovered when re-analyzing the labeling output in the sample. This becomes problematic when executing the functions with edit distance, as they add considerable overhead in terms of time. Also, the side effects include unpredictable outcomes on the ordering of the functions for the sake of adding some patterns. Testing these side effects becomes impractical.

The creation of labeling functions also helps the manual annotation phase. Having to constantly evaluate the performance of the rule-based classifier allowed us to find patterns that had not been annotated before, or misspelled instances that we considered interesting to feed to the other classifiers. In that sense, the phases of rule definition

and manual annotation feed each other. Both take considerable manual effort to build and improve. An important question we wish to answer is whether models automatically trained with the annotated data (without any further human intervention) can approach the performance of the rule-based classification.

4.5 Training Supervised Classifiers

The chosen classifier models are Logistic Regression, Random Forest, SVM, and BERT. Logistic Regression classifiers are efficient on training and evaluating new data, and are somewhat simpler models; they, however, tend to overfit on high dimensional datasets, especially without regularization (JURAFSKY; MARTIN, 2009). Random Forests, although generating more complex models and requiring more computing, generally provide high accuracy and balance the bias-variance trade-off well. SVMs require greater training time but are highly regarded when it comes to NLP tasks. BERT is the most sophisticated classifier we use, it currently is the dominant topic when it comes to NLP innovation and performance, and also is the most computationally expensive model.

Almost every ML process starts with a pre-processing stage, where data is refined and encoded to better serve the model. In Section 4.4, we describe a string sanitization process that transforms to lowercase and removes numbers and diacritics; likewise, this procedure was applied to the remaining classifiers. Instances with empty descriptions were removed and, for the traditional algorithms, we remove instances that are composed of only numerical values and normalize the text so all instances have the same sequence of code points.

BERT transforms the data to be trained and predicted using a pre-trained tokenizer. All tokenizers from BERT extend this main method: converting token strings to ids and back, and encoding/decoding, tokenizing (splitting strings in sub-word token strings), adding new tokens to the vocabulary, and managing special tokens.

Even though the training data is unbalanced, we did not apply any form of over or undersampling. Due to time constraints, we leave this for future work.

4.6 Evaluation Procedure

The aim of the evaluation procedure is to assess the quality of the classification algorithms and their generalization capabilities. Our evaluation procedure considered two scenarios.

- **Scenario 1** – using the annotated dataset $\mathcal{A}$.
- **Scenario 2** – the entire $\mathcal{A}$ is used for training and the predictions are made on dataset $\mathcal{U}$.

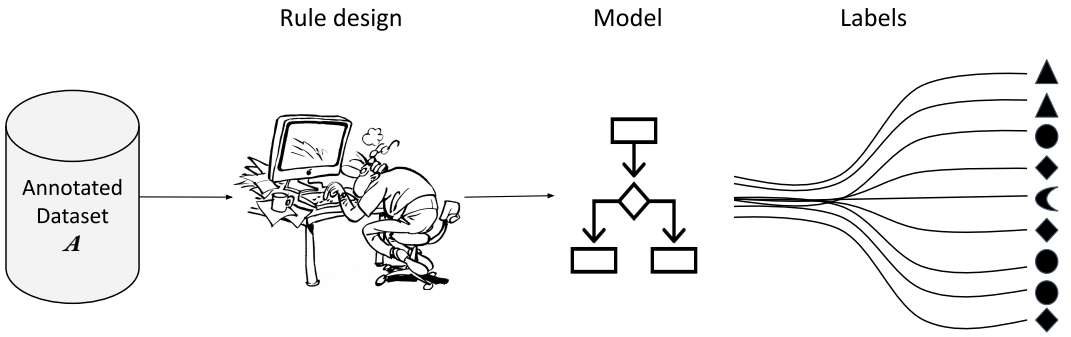
These scenarios are described in the next sections.

4.6.1 Scenario 1

Our goal with Scenario 1 is to assess the classification algorithms using the annotated dataset $\mathcal{A}$.

Rule-based classifier Taking into account that the rule-based classifier does not require any training, the rules can be directly applied to the records and the results are compared with the manual labels from dataset $\mathcal{A}$. The methodology used in the rule-based classifier is shown in Figure 4.4.

Figure 4.4 – Overview of the methodology for the rule-based classifier in Scenario 1



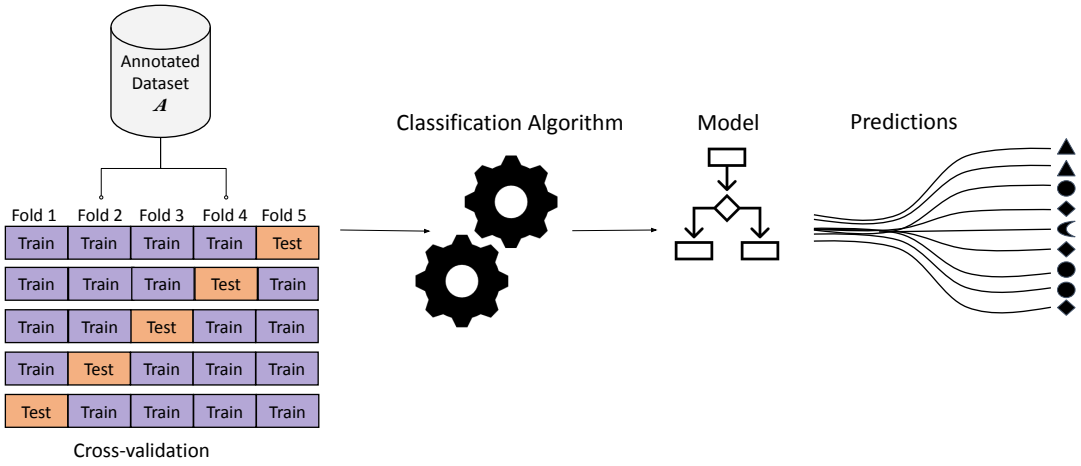
Source: The Author

Traditional classifiers

The standard way to compare the performance of different ML models is through *cross-validation*. K-Fold Cross-validation is a method for evaluating algorithms based on split-

ting the annotated data in k *folds*, typically 5 or 10, and iterating through them training the model with $k-1$ folds while using the remaining fold as test. We define $k=5$ folds and use stratified cross-validation, where the splits seek to preserve the proportion of classes between each fold. The process used for k-fold cross-validation on the traditional algorithms (Random Forest, SVM, and Logistic Regression) is shown in Figure 4.5. Dataset $\mathcal{A}$ is used to train k models on the training folds and the predictions are done on the test fold. With that, every instance has only one predicted class that is compared with the true class (using the labels in $\mathcal{A}$) to calculate the evaluation metrics.

Figure 4.5 – Overview of the methodology for the traditional algorithms in Scenario 1

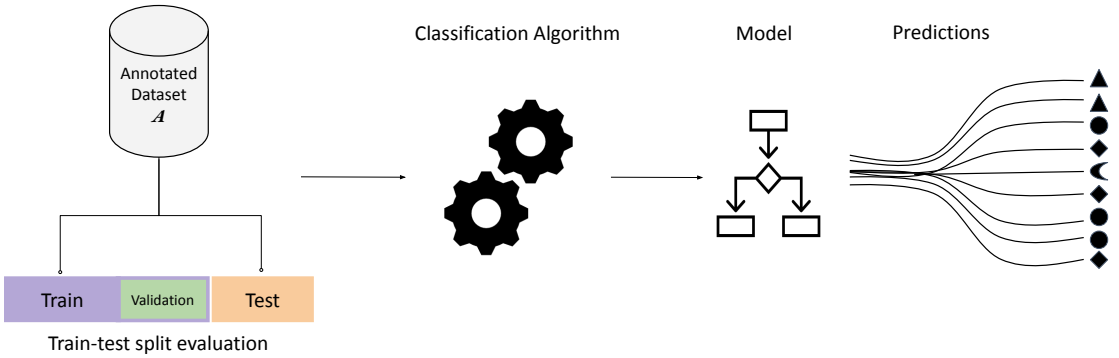


Source: The Author

BERT-based classifiers

For BERT, due to time restrictions, cross-validation was not performed. Also, in addition to the train and test folds, BERT needs a validation fold (commonly required in deep learning algorithms) that is used for model optimization. The process used for BERT is shown in Figure 4.6. The dataset is split in 80% to generate training instances and the remaining 20% become test instances to extract metrics from. The validation fold is a sample taken from the training fold (*i.e.,* 20% of the training fold). We fine-tune BERT setting the parameters with values in the range described by Devlin et al. (2018). During fine-tuning, we observed a decrease in training and validation losses, as is expected.

Figure 4.6 – Overview of the methodology for the BERT-based classifier in Scenario 1



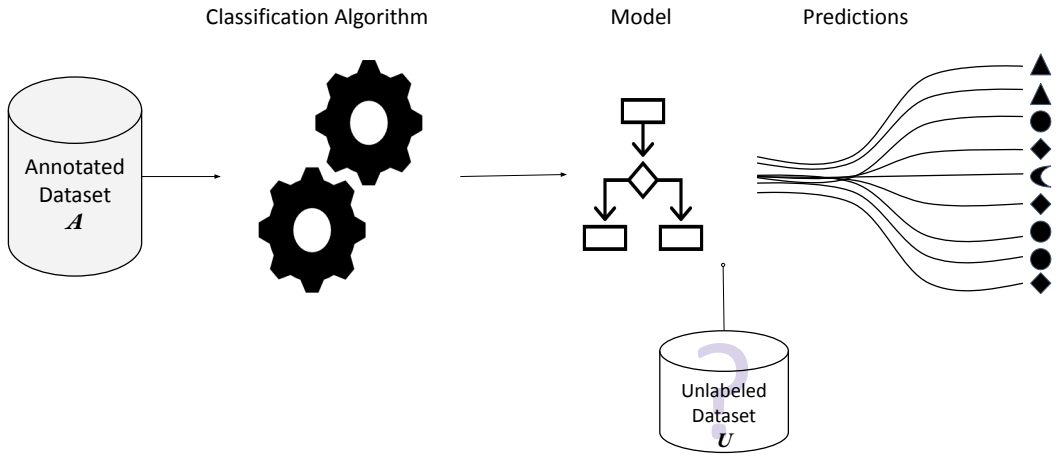
Source: The Author

4.6.2 Scenario 2

In the second scenario, our goal is to assess the classifiers on completely unseen data ($\mathcal{U}$). The rationale is to have an intuition of their generalization capability in a real-world situation.

In this scenario, described in Figure 4.7, the traditional classifiers were trained on the entire annotated dataset ($\mathcal{A}$). BERT was trained on 80% of $\mathcal{A}$ and the remaining 20% were used for validation. The classification model resulting from the training is used to predict the labels for the entire dataset $\mathcal{U}$ (*i.e.,* the one with 11 million instances). For the rule-based classifier, we simply used the rules to generate the labels for $\mathcal{U}$.

Figure 4.7 – Overview of the methodology for all classifiers in Scenario 2



Source: The Author

Then, to assess the performance of the classifiers, we need an annotated sample.

The procedure we used was as follows. For each of the five classifiers, we randomly picked ten instances from each target label (based on their predictions) – or as many as possible if the classifier could not find ten of a given class. When selecting the instances, we excluded descriptions that had appeared on the manually labeled sample. The idea was to select truly unseen data. The process ended up with 1926 instances. These instances were submitted to manual labeling by four human annotators. To have an impartial annotation, we hid the classifiers and predicted labels from the humans, and had every instance labeled by two different annotators. An ambiguity column is added as well, in which any of the two annotators could mark whether they thought another label aside for the one they chose fits a description. To facilitate the annotation, we also filtered repeated descriptions. The annotated dataset with 1926 labels is referred to as $\mathcal{U}'$ and it is used as the ground truth for Scenario 2. This scenario allows a fair comparison among the classification methods.

4.7 Summary

This chapter described the steps took to accomplish the task of classifying points of interest from CNEFE. We started by defining the target classes and annotating a sample $\mathcal{A}$ used to train three traditional supervised classifiers and one BERT-based. During the labeling process, the human annotators extracted expressions used to design a rule-based classifier.

The classifiers were evaluated in two scenarios. The first uses only the annotated sample $\mathcal{A}$ to apply cross-validation on the traditional classifiers and train-test split on the BERT-based, and also compare the results with the labels from the rules. The second scenario uses the final predictions of the classifiers, after training them with the complete dataset $\mathcal{A}$, to predict the labels for $\mathcal{U}'$, a sample with descriptions that have not occurred in $\mathcal{A}$. The dataset $\mathcal{U}'$ provides a fairer comparison among the classifiers.

Chapter 5 details the methods used to implement and evaluate the classifiers.

5 EXPERIMENTAL SETUP

In this chapter, we describe the tools and resources used in the experiments.

5.1 Classifiers

The classifiers used in our experiments can be divided into three groups: (i) the rule-based classifier, (ii) the traditional algorithms (SVM, Random Forest, and Logistic Regression), and (iii) the BERT-based classifier. The next subsections describe how these classifiers were implemented.

5.1.1 Rule-based Classifier

The *Snorkel* library[1] was used to generate and apply the classification based on rules. Snorkel is a framework that offers a set of functionalities to handle training data for models, among other features. The rule-based classifier uses Snorkel's *labeling functions*, a very useful tool that, to its full extent, can be used for data augmentation and even train built-in ML classifiers. In this work, the labeling functions were sufficient for the primary goal of analyzing the dataset using the 122 functions[2] that we designed (see Section 4.4) and applying the class labels. As the order of execution was set on the priority of the functions, it was defined that the first label returned from a function would be the assigned label. However, Snorkel still executes all labeling functions and generates a summarized analysis of the coverage, overlap, and conflicts among functions.

There was roughly one function per target label, plus some functions based on the land use ids 3, 4, and 5 that can be designated to specific classes (described in Section 2.4) and some words that generate high conflict and had their labeling separated from other terms of the same class, totaling 48 labeling functions based on regular expressions and finding substrings. In addition, there were 37 labeling functions for testing keywords with Levenshtein distance and 37 with phonetic distance. The application of labeling functions on the manually labeled dataset ($\mathcal{A}$) took between 30 and 40 minutes without the distance functions. Using the edit distance, the time increased to 4 hours. This happens because the number of necessary comparisons increases greatly if we account for string variations. It

[1] <https://www.snorkel.org/>

[2] Available at <https://github.com/cixcore/cnefe-poi-classification/blob/main/labeling/lf.py>

took 29 hours to apply the Snorkel functions on the unlabelled ($\mathcal{U}$) dataset with 11 million records without the edit distance functions.

The Levenshtein distance with different weights for characters that appear close by on the *QWERTY* keyboard was made using the implementation from *clavier*[3], an MIT licensed repository. A threshold of 25% for dissimilarity was used. As for the phonetic distance, the implementation from the *metaphone-ptbr*[4] package was used; as we had stated in Section 4.4, the phonetic distance functions only apply the label if the description has a word that, when encoded, matched exactly with the codification of one of the classes keywords.

5.1.2 Traditional Classifiers

We used the *scikit-learn*[5] library to implement RandomForestClassifier[6], SVC (the SVM implementation)[7], and LogisticRegression[8]. Scikit-learn is a widely-used open-source library in python with tools for predictive data analysis. The parameters we used were:

- RandomForestClassifier: **n_estimators** (number of decision trees used) = 100; **criterion** (function to measure the quality of a split) = "gini" (for the Gini impurity)
- LogisticRegression: **C** (inverse of regularization strength) = 1.0
- SVC: **C** (inverse of regularization strength) = 1.0; **kernel** = "rbf"

Scikit-learn offers the tools used for cross-validation, which was used to generate the metrics of performance over the annotated dataset ($\mathcal{A}$). We employed 5-fold cross-validation with each instance belonging to exactly one test set.

Vectorization of the input data into these classifiers was carried out using scikit-learn – **TfidfVectorizer** was used for land use description, and **OneHotEncoder** was used for land use id, and **ColumnTransformer** was used to combine both attributes.

[3]<https://github.com/MaxHalford/clavier>
[4]<https://github.com/carlosjordao/metaphone-ptbr>
[5]<https://scikit-learn.org/stable/>
[6]<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
[7]<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
[8]<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

5.1.3 BERT-based Classifier

The BERT model was trained using the *transformers*[9] official library. The **bert-base-multilingual-uncased** pre-trained model was used with 64 maximum tokens, batch size of 32, and 3 epochs. The default loss function (cross-entropy) and optimizer (AdamW (LOSHCHILOV; HUTTER, 2017)) were used with learning rate of 4×10^{-4}. With that, we instantiate a **BertForSequenceClassification** with hidden dropout probability of 0.1.

5.2 Environment Configuration

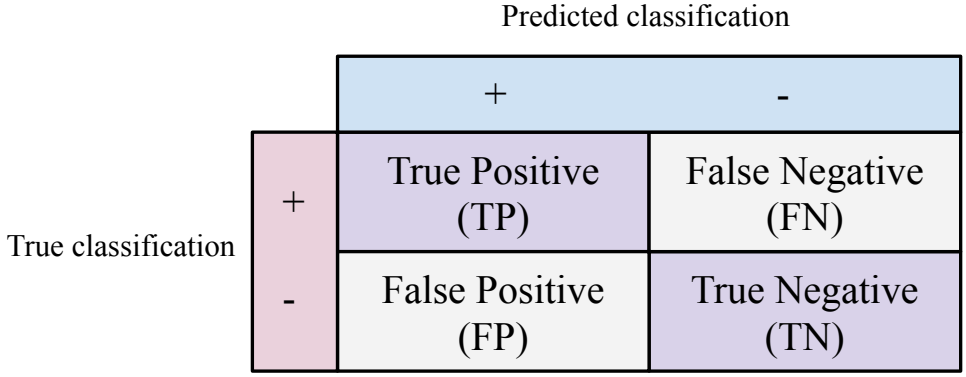
To train the traditional and rule-based classifiers, we use a machine with Intel Core i5-8265U processor, Intel UHD Graphics 620 GPU, and 8GB of memory.

BERT is trained in the *Google Colaboratory*[10] virtual environment.

5.3 Evaluation Metrics

The metrics we used in our evaluation can be explained based on the confusion matrix shown in Figure 5.1. To generate the confusion matrices presented in Chapter 6, we used once again a scikit-learn method, the **ConfusionMatrixDisplay.from_predictions**.

Figure 5.1 – Overview of a Confusion Matrix



Source: The Author

[9] <https://huggingface.co/docs/transformers/index>
[10] <https://colab.research.google.com/>

5.3.1 F1 Score

To explain macro- and micro-F1 scores, first we have to introduce *precision* and *recall*. Precision tells us how many of the predicted positives are actual true positives, while recall measures how many of the true positives were correctly predicted among the true predictions.

$$precision = \frac{TP}{TP + FP} \tag{5.1}$$

$$recall = \frac{TP}{TP + FN} \tag{5.2}$$

With that, we have the F1 score of a class as the harmonic mean between precision and recall.

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{5.3}$$

Finally, the *micro-F1 score* is simply the F1 formula applied to the global number of True Positives, False Positives, and False Negatives, instead of individually for each class. Micro-F1 assigns equal weights to all instances, so large classes end up having more weight, *i.e.,* a greater impact on the metric. Conversely, the *macro-F1 score* is the mean of the F1 scores calculated per class. In imbalanced datasets, it penalizes poor performance in smaller classes, for every class is given equal weight independently from its size.

$$F1_{macro} = \frac{F1_{class1} + F1_{class2} + ... + F1_{classN}}{N} \tag{5.4}$$

5.3.2 Significance Testing

To determine statistical differences between the classifiers, we analyzed the results of a **Wilcoxon signed-rank**. The Wilcoxon signed-rank test is a paired difference test used when the differences in measurements might not follow a normal distribution. Its goal is to determine whether the population mean ranks differ. This test compares the predictions of two classifiers and determines whether the difference in performance can be considered caused by randomness.

5.3.3 Correlation among classifiers

The Pearson correlation coefficient is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations. It tells us how correlated the predictions of different classifiers are. We apply this test to the predicted classes of the classifiers.

6 RESULTS

In this section, we present the results of our experimental analysis. We start reporting on the runtimes of the classification methods and then we analyze the results under the two classification scenarios described in Chapter 5.

Displaying the class labels in our figures and tables is impossible due to their long names. To aid the reader in understanding our analyzes, Table 6.1 repeats the list of target classes sorted by class id.

6.1 Runtimes

SVM, Logistic Regression, and Random Forest took less than a minute to train on dataset $\mathcal{A}$, and the prediction of the whole dataset needed twenty minutes to Random Forest and around 85 seconds to Logistic Regression. SVM took around 5 hours to finish predicting all 11 million instances. BERT is a very large model that generates a lot of data when predicting the classes for new instances, so we needed to paginate the dataset and append the predictions in the end. The rule-based classifier took around 30 hours to complete labeling the full dataset $\mathcal{U}$.

6.2 Evaluation of Scenario 1

This section shows the results obtained from the BERT, SVM, Random Forest, Logistic Regression, and the rule-based classifier – which we will refer to as *snorkel* for simplicity – using evaluation Scenario 1.

Snorkel's evaluation in this scenario showed using edit distance functions, which had an execution time eight times higher than just using phonetic distance functions or no distance functions at all, actually did not perform much better. Using both groups of distance functions resulted in metrics around 1pp. better than just using phonetic distance or edit distance, and around 1pp. worse (for all metrics) if no distance functions were used. Using just phonetic distance had a performance less than 0.5pp worse than using just edit distance, so we opted for using only phonetic distance functions to predict the labels.

Table 6.2 shows the results for Macro- and Micro-F1 for the complete set of

Table 6.1 – List of classes

Id	*Label*
0	Obras
1	Agricultura, pecuária, produção florestal, pesca e aquicultura
2	Indústrias extrativas
3	Indústrias de transformação
4	Eletricidade e gás
5	Água, esgoto, atividades de gestão de resíduos e descontaminação
6	Construção
7	Igrejas, templos e atividades religiosas
8	Comércio e reparação de veículos automotores e motocicletas
9	Comércio por atacado, exceto veículos automotores e motocicletas
10	Comércio varejista
11	Comércio varejista não especializado
12	Comércio varejista de mercadorias em geral, com predominância de produtos alimentícios - hipermercados e supermercados
13	Comércio varejista de mercadorias em geral, com predominância de produtos alimentícios - minimercados, mercearias e armazéns
14	Comércio varejista de produtos alimentícios, bebidas e fumo
15	Comércio varejista de combustíveis para veículos automotores
16	Comércio varejista de material de construção
17	Comércio varejista de equipamentos de informática e comunicação; equipamentos e artigos de uso doméstico
18	Comércio varejista de artigos culturais, recreativos e esportivos
19	Comércio varejista de produtos farmacêuticos, perfumaria e cosméticos e artigos médicos, ópticos e ortopédicos
20	Comércio varejista de produtos novos não especificados anteriormente e de produtos usados
21	Desocupado
22	Transporte terrestre
23	Transporte aquaviário
24	Transporte aéreo
25	Armazenamento e atividades auxiliares dos transportes
26	Correio e outras atividades de entrega
27	Não definido
28	Alojamento
29	Alimentação
30	Informação e comunicação
31	Atividades financeiras, de seguros e serviços relacionados
32	Atividades imobiliárias
33	Atividades profissionais, científicas e técnicas
34	Atividades administrativas e serviços complementares
35	Administração pública, defesa e seguridade social
36	Educação
37	Saúde humana e serviços sociais
38	Artes, cultura, esporte e recreação
39	Organismos internacionais e outras instituições extraterritoriais
40	Outras atividades de serviços

Source: The Author

classes. The scores from the traditional supervised classifiers and from snorkel are very close to each other. Among the traditional algorithms, Logistic Regression was the worst performer. SVM and Random forest were superior to Logistic regression and even to snorkel. BERT obtained very high scores in this scenario. However, since we did not put

Table 6.2 – Results of Scenario 1

Model	*Macro-F1*	*Micro-F1*
Snorkel	0.731	0.859
Logistic Regression	0.706	0.874
Random Forest	0.745	0.861
SVM	**0.753**	**0.876**
BERT	**0.880**	**0.983**

Source: The Author

BERT through cross-validation, its results can not be compared on equal footing to the other classifiers. The next evaluation section helps us see more clearly the differences in their performances.

6.3 Evaluation of Scenario 2

Here, we present the results of the classifiers on a new sample with instances that had not been seen by any of the classifiers during training or rule creation (*i.e.,* dataset $\mathcal{U}$). The ground truth labels from dataset $\mathcal{U}'$ were used to calculate the evaluation metrics. In Section 6.3.1 we evaluate the overall results and in Section 6.3.2 we discuss some hypothesis for the classifiers performances in specific classes.

6.3.1 Overall Evaluation

The summary of the results for all classes is shown Table 6.3. We can see that the performance of all classifiers was noticeably lower in this scenario. SVM had the best performance among the traditional classifiers, but ended up among the worst performances in the second scenario. This is expected since the size of the annotated sample $\mathcal{A}$ is about 0.4% of $\mathcal{C}$ and is not enough to capture all the patterns that could occur. Still, when we compare the performance of the classifiers, we see a similar pattern (in relation to Table 6.2), with BERT being the best performer among the ML classifiers and Logistic Regression being the worst.

Table 6.4 shows whether the differences for all pairs of classifiers are statistically significant. We can see that nearly all differences were found to be significant with p-values $\ll 0.01$. The exceptions were between BERT and Snorkel and between SVM and Logistic Regression. To summarize our findings, we can say that: Logistic Regression

Table 6.3 – Results of the classifiers on $\mathcal{U}'$

Model	*Macro-F1*	*Micro-F1*
Snorkel	**0.710**	**0.694**
Logistic Regression	0.645	0.607
Random Forest	0.685	0.652
SVM	0.679	0.614
BERT	**0.694**	**0.703**

Source: The Author

was outperformed by all other classifiers; BERT and snorkel outperformed all others and are no different from each other; and Random forest was better than SVM.

Table 6.4 – Wilcoxon signed-rank test p-values for the classifiers' predictions made over $\mathcal{U}'$

	Random Forest	*Logistic Regression*	*Snorkel*	*BERT*
SVM	1.3E-05	**2.8E-01**	9.0E-12	1.4E-20
Random Forest	-	1.7E-07	2.5E-04	1.5E-07
Logistic Regression	-	-	9.1E-13	1.7E-24
Snorkel	-	-	-	**4.4E-01**

Source: The Author

Table 6.5 shows the correlations between all pairs of classifiers. A strong correlation was found between SVM and Logistic Regression as they tend to agree often on their predictions. All other correlations can be considered moderate.

Table 6.5 – Pearson correlation coefficient for the classifiers' predictions made over $\mathcal{U}'$

	Random Forest	*Logistic Regression*	*Snorkel*	*BERT*
SVM	0.69	**0.84**	0.43	0.63
Random Forest	-	0.70	0.44	0.60
Logistic Regression	-	-	0.39	0.65
Snorkel	-	-	-	0.41

Source: The Author

Since we had five classifiers, it made sense to investigate what their performance would be if the predictions were submitted to a simple majority voting system. Table 6.6 shows how many instances were correctly predicted by the number of classifiers that succeeded in their predictions. There were 279 instances (14% of $\mathcal{U}'$) that could not be correctly labeled by any of the classifiers. We verified that the accuracy of predictions would be 64% if there was a majority voting system, which is below the performances of BERT, Snorkel, and Random Forest alone. This leads us to the conclusion that a simple ensemble would not yield gains in classification quality.

Table 6.6 – Relation of how many classifiers predicted correctly the instances from $\mathcal{U}'$

Correctly predicted by N classifiers	*Instances*	*% of* $\mathcal{U}'$
none	279	14%
1	248	13%
2	130	7%
3	140	7%
4	270	14%
5	859	44%

Source: The Author

A more lenient evaluation

As pointed out in Section 4.3.2, one address can serve many purposes but our ground truth considered only the first economic activity in the description. We also performed a more lenient evaluation in which we considered the classifier's prediction correct if it fitted the following criteria:

1. The prediction was of a parent class, even though the instance was labeled in a child class, for example, a gas station is labeled as fuel retail trade, but the classifier puts it in the retail trade (*i.e.,* the parent class).
2. There was more than one economic activity with different target classes described in the instance, and the classifier was able to predict one of them correctly, but it was not the first that appeared in the description
3. If the description was ambiguous and the label assigned by the classifier could be considered correct.

As expected, the lenient evaluation provided better results for all the classifiers, although not by much, as we can see in Table 6.7. It was between 1 and 2.3pp. better for all classifiers. Even though misspelling, abbreviations, and vagueness presented a challenge when labeling and designing rules for the snorkel classifier, it appeared this was not so challenging for the supervised classifiers, as the lenient result did not differ so much from the performance over the original annotation on $\mathcal{U}'$.

Table 6.7 – Results of the classifiers on $\mathcal{U}'$ with lenient labeling and difference from the original

Model	*Macro F1*$_{lenient}$	Δ *Macro F1*	*Micro F1*$_{lenient}$	Δ *Micro F1*
Snorkel	**0.731**	0.021	**0.713**	0.019
Logistic Regression	0.656	0.011	0.618	0.011
Random Forest	0.695	0.010	0.664	0.012
SVM	0.702	**0.023**	0.636	**0.022**
BERT	**0.709**	0.015	**0.718**	0.015

Source: The Author

6.3.2 Results by class

In this section, we analyze the results for the individual classes. The scores for TP, TN and F1 are shown in Table 6.8. Confusion matrices for all classifiers are shown in Figures 6.1, 6.2, 6.3, 6.4, and 6.5.

The results show that neither of the classifiers could learn the undefined class (id 27) very well. This could be explained by the concept of the class itself, which is designed for instances not defined certainly by any other class. For example, lets take the description "ESCRITORIO DE COSMETICOS" (*cosmetics office*) present in the new sample: offices can usually be put in class "33 - Atividades profissionais, científicas e técnicas"; however, we do not have enough information to assume this is about an administrative center for a cosmetics company, or a place where they sell cosmetics (fitting class "19 - Comércio varejista de produtos farmacêuticos, perfumaria e cosméticos e artigos médicos, ópticos e ortopédicos"), or maybe even a trade representative headquarters (which falls into class "9 - Comércio por atacado, exceto veículos automotores e motocicletas"). At the same time that there is simply not an adequate amount of contextual information provided by the description to say what is most likely to be taking place in that location, it does happen that some words very distinctive of other classes are present. Not only neither of the classifiers performed so well in class 27, but Random Forests predicted many false positives in this class.

Random Forest, Logistic Regression, and SVM show some missclassifications in their confusion matrices for class "33 - Atividades profissionais, científicas e técnicas" that gets predicted as "6 - Construção". A common description from class 33 can look like "ESCRITORIO DE CONSTRUCAO CIVIL" (*civil construction office*) or contain the word "ENGENHARIA" (*engineering*), and these expressions have a strong relation to the context of class 6 (for construction services, contractors, and such).

We can also see the impact of having so few examples in class "23 - Transporte aquaviário". BERT and Logistic Regression classifiers did not learn how to apply this class and did not assign any instances from $\mathcal{U}$ to it. All the examples we found in this class had very similar patterns, *i.e.,* the presence of the word word "balsa" (*ferry*). However, SVM and Random Forest did overcome fairly well this adversity for many of the most undersampled classes and were able to learn how to recognize some of their instances, which reflects on their macro-F1 results.

Two other classes that also were mislabeled by the classifiers are "14 - Comér-

cio varejista de produtos alimentícios, bebidas e fumo" and "17 - Comércio varejista de equipamentos de informática e comunicação; equipamentos e artigos de uso doméstico", that were often classified as "20 - Comércio varejista de produtos novos não especificados anteriormente e de produtos usados". This could be related to the number of annotations of class 20. It is the fourth largest class and has almost twice the number of instances from the fifth largest. It is also a very diverse class, as it encompasses many different types of retail trade. There could also be some relation to the difference in quality between the annotations in $\mathcal{A}$ and $\mathcal{U}'$. The annotations in $\mathcal{U}'$ were more careful since each instance was annotated by two people who discussed the cases in which they disagreed. In $\mathcal{A}$, only one person labeled each instance and no revisions were made.

Table 6.8 – True positive, True negative, and F1 score by class for the predictions over $\mathcal{U}'$

	Snorkel			*LR*			*RF*			*SVM*			*BERT*		
Id	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*
0	0.61	1.00	0.68	0.54	1.00	0.59	0.64	0.99	0.64	0.75	1.00	0.75	0.75	1.00	**0.76**
1	0.82	0.99	0.76	0.82	1.00	0.84	0.82	1.00	**0.85**	0.80	0.99	0.81	0.66	0.99	0.64
2	0.85	0.99	0.60	0.85	1.00	0.87	0.93	1.00	**0.88**	0.93	1.00	**0.88**	0.85	1.00	0.79
3	0.57	1.00	**0.69**	0.26	1.00	0.40	0.37	1.00	0.51	0.26	1.00	0.40	0.46	0.99	0.57
4	0.83	0.99	0.68	0.86	1.00	0.85	0.90	1.00	0.85	0.90	1.00	**0.88**	0.90	0.99	0.79
5	0.68	1.00	0.76	0.77	1.00	**0.81**	0.73	1.00	0.76	0.75	1.00	0.80	0.80	1.00	0.80
6	0.37	0.99	0.40	0.47	0.98	0.36	0.40	0.99	0.36	0.40	0.99	0.36	0.53	0.99	**0.44**
7	0.78	1.00	0.86	0.93	0.97	0.59	0.90	0.99	0.72	0.93	0.98	0.65	0.93	1.00	**0.88**
8	0.77	0.99	**0.77**	0.59	0.99	0.66	0.64	0.99	0.71	0.62	1.00	0.70	0.67	0.99	0.67
9	0.80	1.00	0.86	0.80	1.00	0.86	0.80	1.00	0.84	0.80	1.00	0.87	0.86	1.00	**0.90**
10	0.78	0.98	0.52	0.66	0.99	0.65	0.66	1.00	**0.68**	0.66	0.99	0.59	0.62	1.00	**0.68**
11	0.67	1.00	0.73	0.67	1.00	0.78	0.72	1.00	**0.80**	0.70	1.00	0.78	0.67	1.00	0.72
12	0.98	1.00	**0.96**	0.22	1.00	0.36	0.86	1.00	0.92	0.56	1.00	0.72	0.92	1.00	0.94
13	0.78	1.00	0.82	0.76	0.99	0.73	0.67	0.99	0.67	0.74	0.99	0.72	0.89	1.00	**0.86**
14	0.39	1.00	0.54	0.51	0.99	0.56	0.48	0.99	0.55	0.45	1.00	0.58	0.51	0.99	**0.59**
15	0.96	0.99	0.72	0.96	0.99	0.73	0.92	0.99	0.78	0.96	0.99	**0.79**	0.96	0.99	0.74
16	0.54	1.00	0.67	0.71	0.99	0.73	0.71	0.99	0.71	0.68	0.99	0.72	0.79	0.99	**0.74**
17	0.40	0.99	0.47	0.34	0.99	0.44	0.30	1.00	0.41	0.36	1.00	0.47	0.55	0.99	**0.60**
18	0.79	1.00	**0.85**	0.88	0.99	0.74	0.82	0.99	0.76	0.88	0.99	0.76	0.94	0.99	0.68
19	0.89	1.00	**0.91**	0.92	0.99	0.82	0.97	0.99	0.81	0.89	1.00	0.89	1.00	0.99	0.85
20	0.57	0.99	**0.53**	0.68	0.79	0.11	0.65	0.93	0.26	0.76	0.76	0.11	0.65	0.97	0.39
21	0.81	1.00	0.88	0.88	1.00	**0.89**	0.73	0.99	0.70	0.77	1.00	0.85	0.92	0.99	0.82
22	0.76	1.00	**0.77**	0.68	0.99	0.60	0.68	0.99	0.61	0.62	0.99	0.64	0.68	0.99	0.66
23	0.93	0.99	0.70	-	1.00	-	0.60	1.00	0.64	0.67	1.00	**0.74**	-	1.00	-

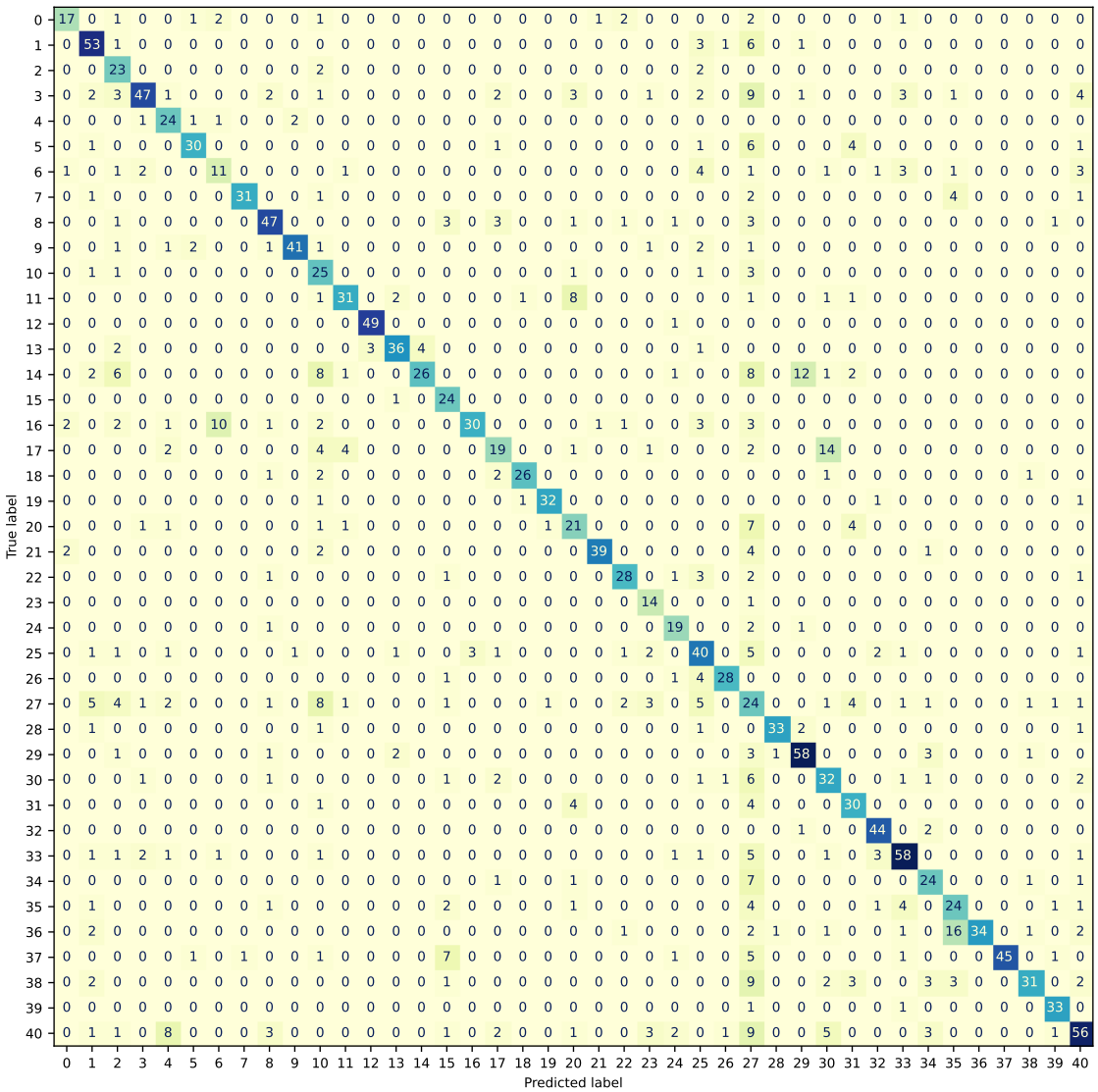
Table 6.8 – Continued on next page

Table 6.8 – Continued from previous page

	Snorkel			*LR*			*RF*			*SVM*			*BERT*		
Id	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*	*TP*	*TN*	*F1*
24	0.83	1.00	**0.75**	0.43	1.00	0.54	0.74	0.99	0.67	0.74	0.99	0.68	0.78	0.99	0.67
25	0.66	0.98	0.59	0.67	1.00	**0.74**	0.57	1.00	0.69	0.67	0.99	0.71	0.57	0.99	0.63
26	0.82	1.00	0.86	0.71	1.00	0.81	0.56	1.00	0.68	0.94	1.00	**0.94**	0.91	0.99	0.79
27	0.35	0.93	**0.22**	-	1.00	-	0.44	0.91	**0.22**	-	1.00	-	0.10	1.00	0.18
28	0.85	1.00	0.89	0.97	1.00	**0.96**	0.97	1.00	0.95	0.97	1.00	**0.96**	0.97	0.99	0.80
29	0.83	0.99	**0.79**	0.57	0.99	0.66	0.67	0.99	0.72	0.43	1.00	0.59	0.71	0.99	0.74
30	0.65	0.99	0.59	0.47	0.99	0.55	0.47	0.99	0.49	0.43	1.00	0.54	0.63	0.99	**0.68**
31	0.77	0.99	0.69	0.77	1.00	0.83	0.72	0.99	0.71	0.72	1.00	0.79	0.92	1.00	**0.87**
32	0.94	1.00	0.89	0.68	1.00	0.80	0.96	1.00	**0.96**	0.66	1.00	0.78	0.91	1.00	0.95
33	0.75	0.99	**0.76**	0.34	1.00	0.50	0.38	1.00	0.52	0.25	1.00	0.38	0.52	0.99	0.62
34	0.69	0.99	**0.66**	0.54	1.00	0.60	0.49	0.99	0.52	0.43	0.99	0.48	0.71	0.98	0.55
35	0.60	0.99	0.54	0.70	0.99	**0.69**	0.60	0.99	0.62	0.60	1.00	0.67	0.80	0.99	**0.69**
36	0.56	1.00	0.72	0.85	1.00	**0.91**	0.85	1.00	0.90	0.84	1.00	0.89	0.89	1.00	0.89
37	0.71	1.00	0.83	0.83	1.00	**0.88**	0.76	1.00	0.84	0.83	1.00	**0.88**	0.81	0.99	0.74
38	0.55	1.00	0.67	0.57	1.00	0.68	0.59	1.00	0.68	0.54	1.00	0.65	0.70	0.99	**0.72**
39	0.94	1.00	0.90	0.63	1.00	0.77	0.94	1.00	**0.97**	0.86	1.00	0.92	0.97	1.00	0.92
40	0.58	0.99	**0.64**	0.38	1.00	0.55	0.36	1.00	0.51	0.37	1.00	0.54	0.41	0.99	0.53

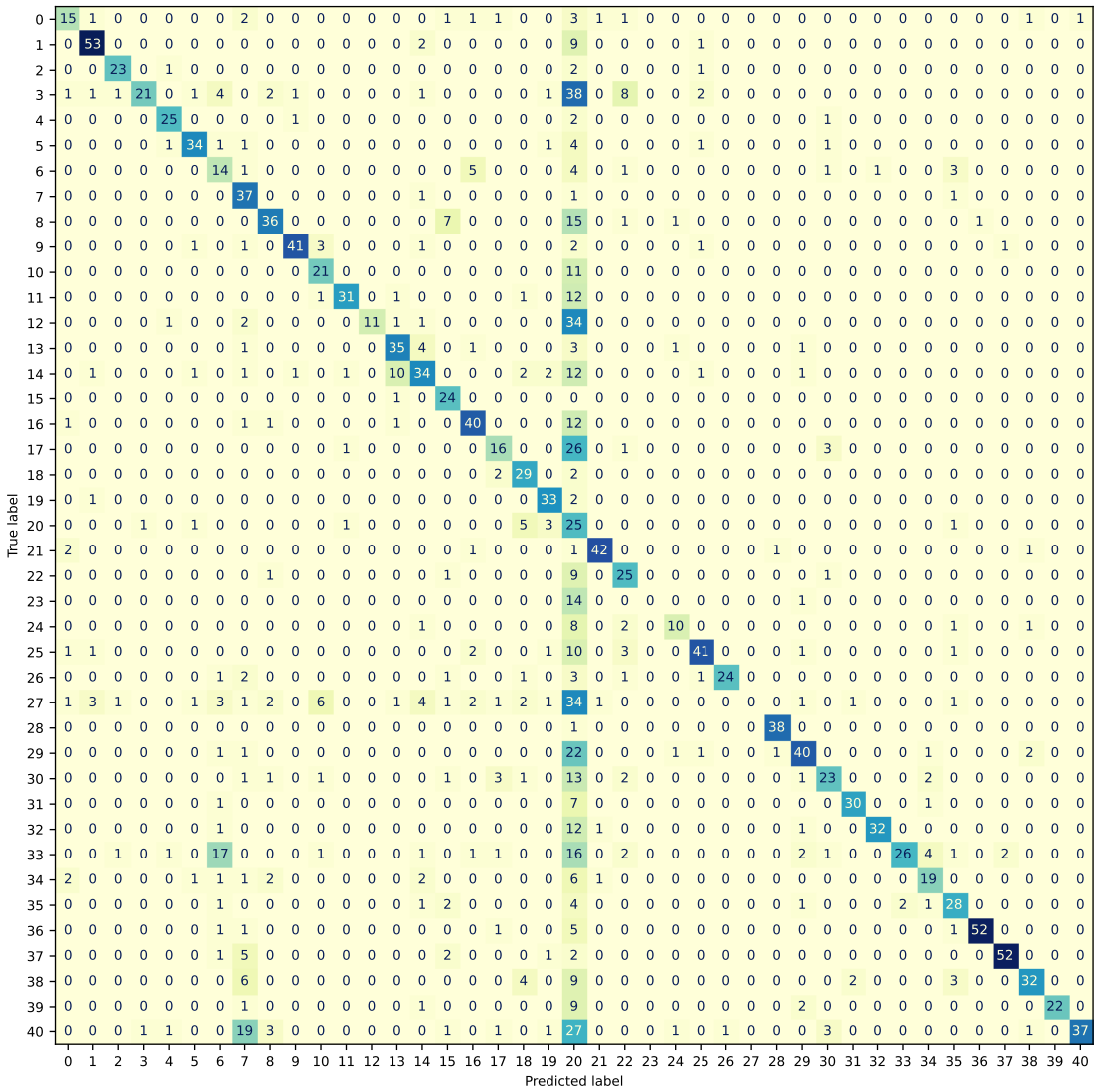
Source: The Author

Figure 6.1 – Confusion matrix for Snorkel's predictions on $\mathcal{U}'$



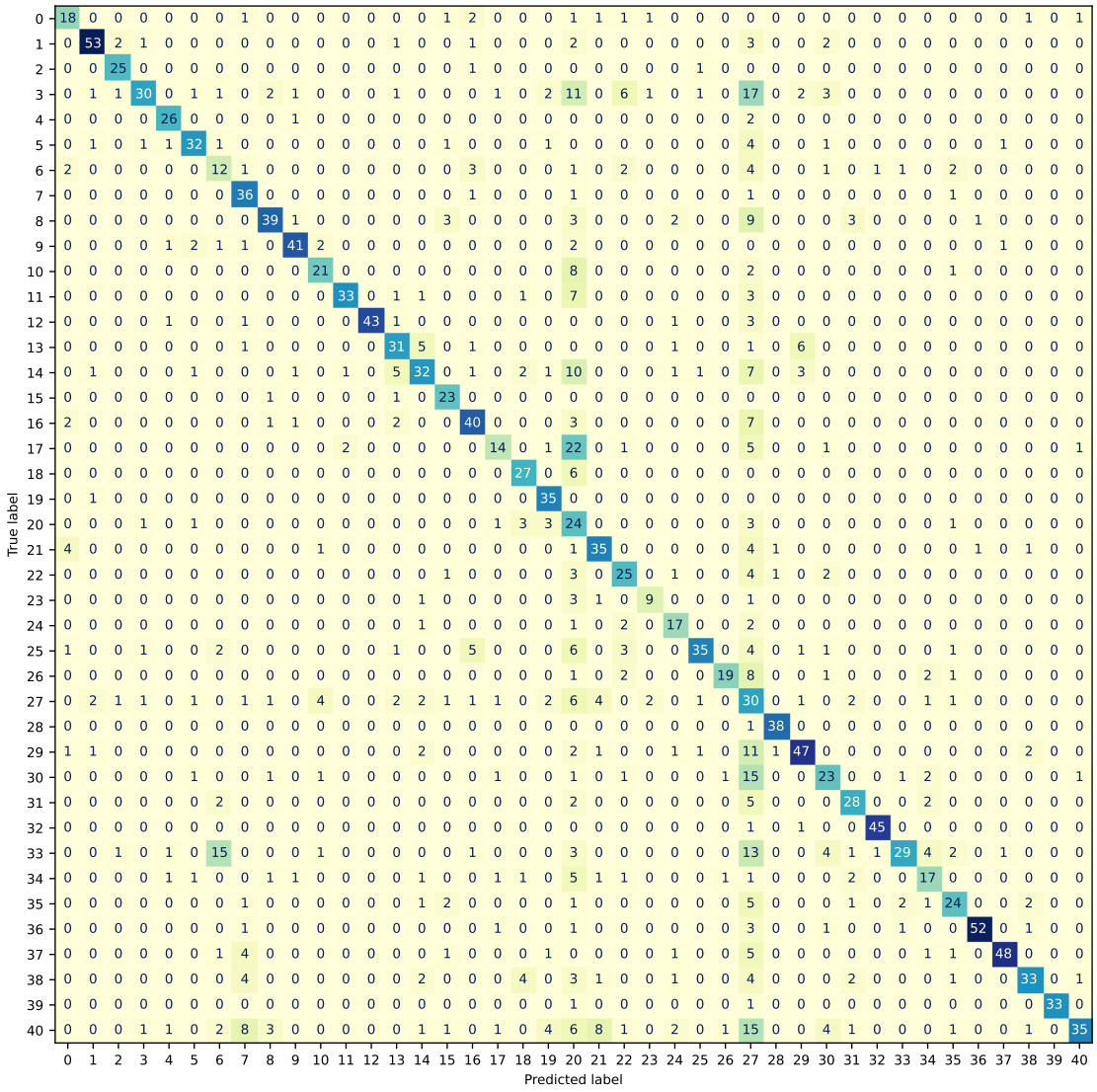
Source: The Author

Figure 6.2 – Confusion matrix of Logistic Regression's predictions on $\mathcal{U}'$



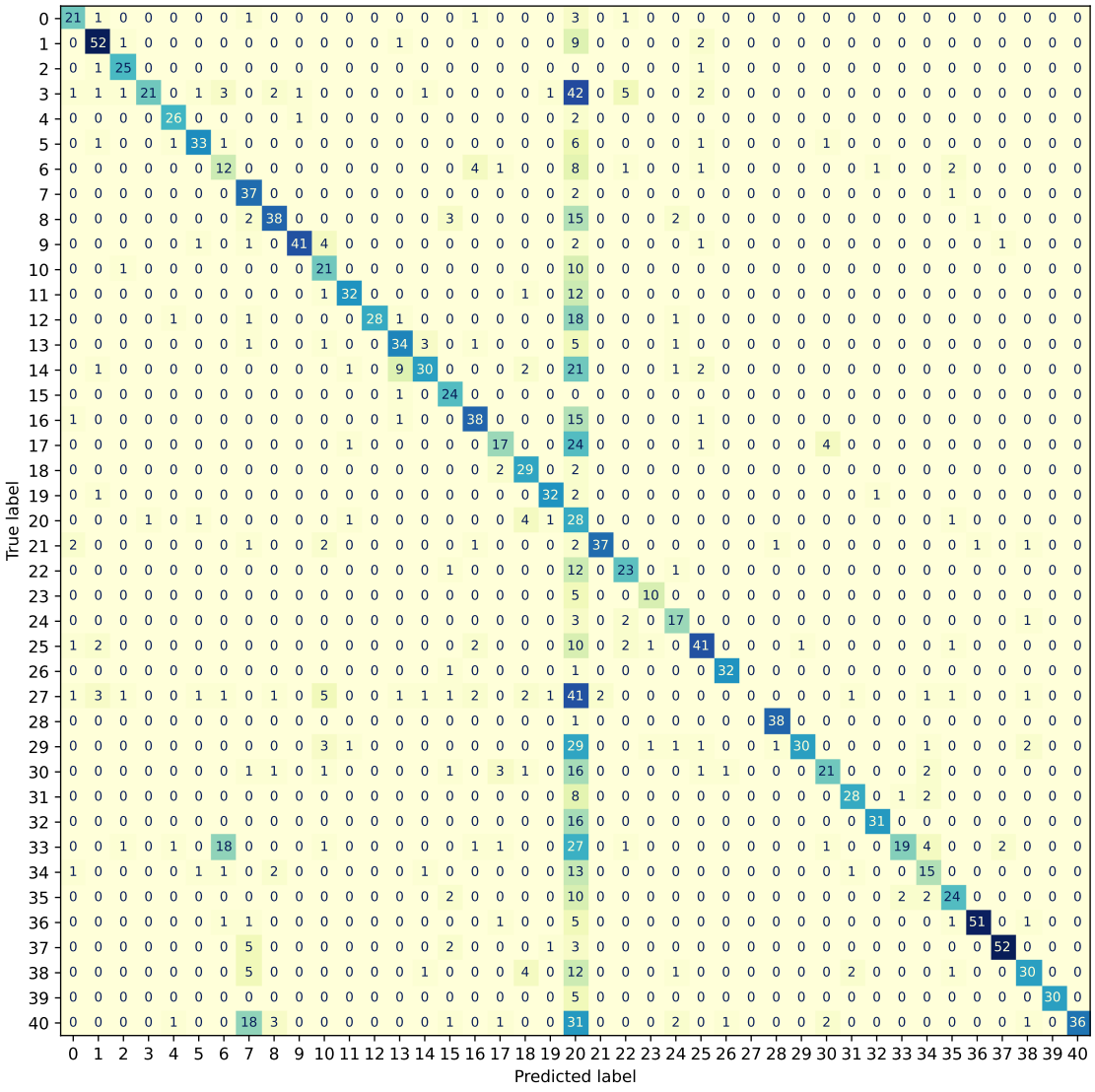
Source: The Author

Figure 6.3 – Confusion matrix for Random Forest's predictions on $\mathcal{U}'$



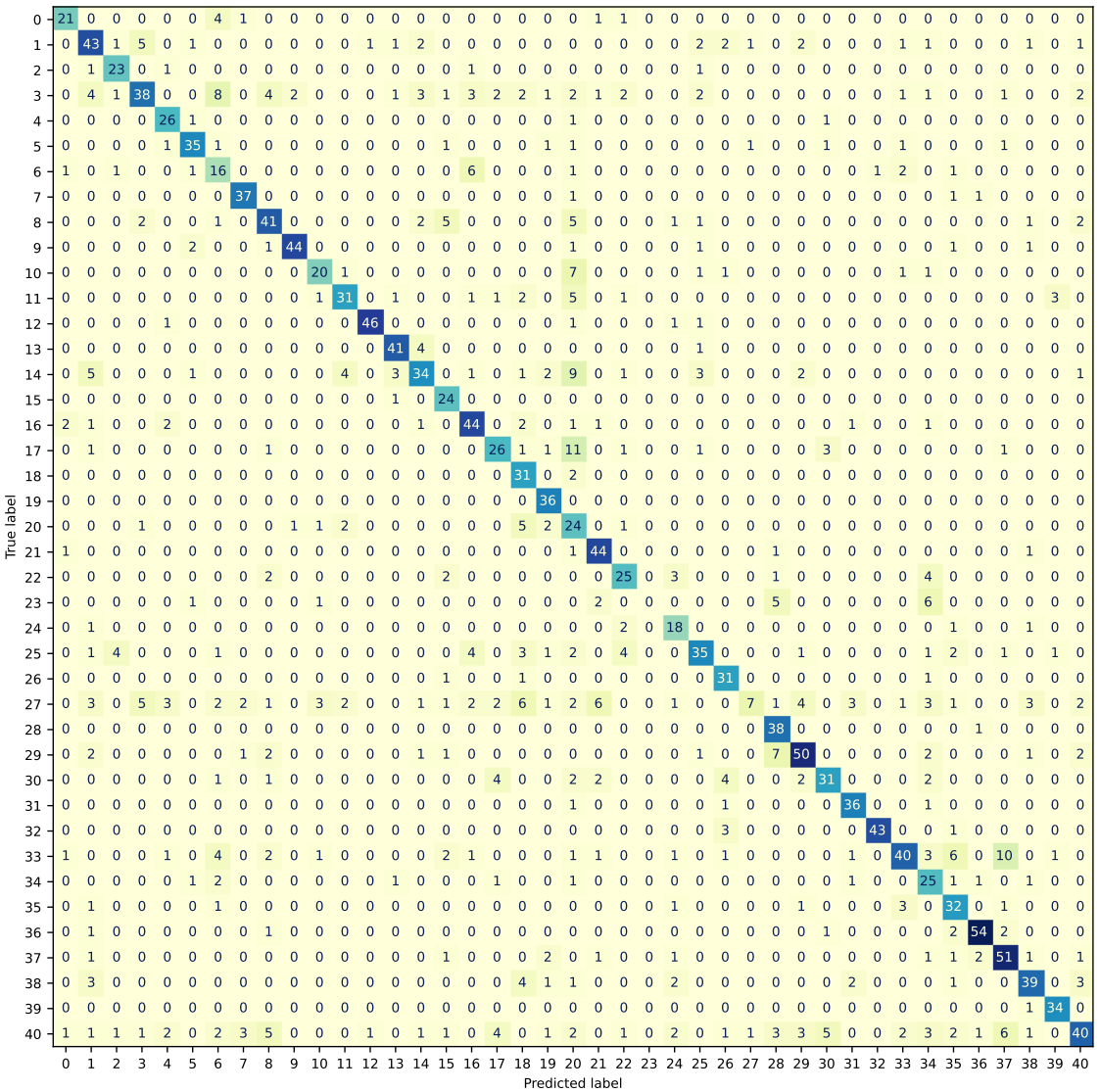
Source: The Author

Figure 6.4 – Confusion matrix of SVM classifier's predictions on $\mathcal{U}'$



Source: The Author

Figure 6.5 – Confusion matrix for BERT's predictions on $\mathcal{U}'$



Source: The Author

7 CONCLUSION

In this work, we presented different methods to automatically classify records from CNEFE data based on their land use ids and descriptions. The descriptions are written in natural language, and can be vague or contain misspellings, making the task harder. First, we labeled 44k instances manually and assembled lexicons with words that are distinctive of each class. The lexicons were employed to create 122 labeling functions, from which we selected 85 to classify the entire dataset. From the annotated instances, we trained four classification models based on different algorithms, namely Logistic Regression, Random Forest, SVM, and BERT. The trained classifiers were employed to predict the labels of completely unseen instances. The results show F1-macro ranging from 0.65 to 0.71 and F1-micro from 0.61 to 0.70.

The top-three performances were from BERT, the rule-based classifier (implemented with snorkel), and Random Forests.

Even though BERT had the best scores, it was the only classifier that had lower macro-F1 (0.69) than micro-F1 (0.70). This implies that it suffered more than the other classifiers with the small number of instances in some classes. Other classifiers also suffered from the class imbalance of the training instances, but we can see they generally did better on recognizing some patterns for all classes than being able to have precise predictions for only some of them on their performance.

It was discussed that many instances had more than one economic activity label associated with their descriptions. In future works, a multi-label approach could be explored. In addition, having found better performances with the BERT and Random Forest classifiers, we can investigate how much impact different configurations or strategies like oversampling can have on their performances.

REFERENCES

BATTISTAM, C. K. **Procedimentos de pesquisa em Geografia do Comércio e do Consumo: delimitação, intensidade e especialização de áreas centrais. Análises a partir de Marília/SP, São Carlos/SP e São José do Rio Preto/SP**. 2015. Available from Internet: <http://hdl.handle.net/11449/124140>.

BREIMAN, L. Random forests. In: . [S.l.]: Springer, 2001. v. 45, n. 1, p. 5–32.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv, 2018. Available from Internet: <https://arxiv.org/abs/1810.04805>.

IBGE, C. S. **Cadastro Nacional de Endereços para Fins Estatísticos auxiliará na produção de pesquisas domiciliares**. 2011. <https://censo2010.ibge.gov.br/noticias-censo.html?busca=1&id=1&idnoticia=2028&t=cadastro-nacional-enderecos-fins-estatisticos-auxiliara-producao-pesquisas-domiciliares&view=noticia>. Access on August 29th 2022.

JURAFSKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. Pearson Prentice Hall, 2009. (Prentice Hall series in artificial intelligence). ISBN 9780131873216. Available from Internet: <https://books.google.com.br/books?id=fZmj5UNK8AQC>.

LEVENSHTEIN, V. I. et al. Binary codes capable of correcting deletions, insertions, and reversals. In: SOVIET UNION. **Soviet physics doklady**. [S.l.], 1966. v. 10, n. 8, p. 707–710.

LORENA, A. et al. **Inteligência Artificial: uma abordagem de Aprendizado de Máquina (2a edição)**. [S.l.]: LTC, 2011. ISBN 9788521637493.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. **arXiv preprint arXiv:1711.05101**, 2017.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. [S.l.]: Cambridge Univ. Press, 2009. ISBN 0521865719.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

NAVARRO, G. A guided tour to approximate string matching. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 33, n. 1, p. 31–88, 2001.

PADURARIU, C.; BREABAN, M. E. Dealing with data imbalance in text classification. **Procedia Computer Science**, v. 159, p. 736–745, 2019. ISSN 1877-0509. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1877050919314152>.

PETERS, M. E. et al. **Deep contextualized word representations**. arXiv, 2018. Available from Internet: <https://arxiv.org/abs/1802.05365>.

PHILIPS, L. Hanging on the metaphone. **Computer Language**, v. 7, n. 12, p. 39–43, 1990.

UN, D. **68% of the world population projected to live in urban areas by 2050, says UN**. 2018. <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html>. Access on September 9th 2022.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WEI, F. et al. Empirical study of deep learning for text classification in legal document review. In: IEEE. **2018 IEEE International Conference on Big Data (Big Data)**. [S.l.], 2018. p. 3317–3320.

APPENDIX A — CNAE HIERARCHY USED TO LABEL CNEFE

Table A.1 – CNAE reduced hierarchy scheme

Section	*Division*	*Group*	*Class*	*Denomination*
A				**AGRICULTURA, PECUÁRIA, PRODUÇÃO FLORESTAL, PESCA E AQUICULTURA**
B				**INDÚSTRIAS EXTRATIVAS**
C				**INDÚSTRIAS DE TRANSFORMAÇÃO**
D				**ELETRICIDADE E GÁS**
E				**ÁGUA, ESGOTO, ATIVIDADES DE GESTÃO DE RESÍDUOS E DESCONTAMINAÇÃO**
F				**CONSTRUÇÃO**
G				**COMÉRCIO; REPARAÇÃO DE VEÍCULOS AUTOMOTORES E MOTOCICLETAS**
	45			COMÉRCIO E REPARAÇÃO DE VEÍCULOS AUTOMOTORES E MOTOCICLETAS
	46			COMÉRCIO POR ATACADO, EXCETO VEÍCULOS AUTOMOTORES E MOTOCICLETAS
	47			COMÉRCIO VAREJISTA
		47.1		Comércio varejista não especializado
			47.11-3	Comércio varejista de mercadorias em geral, com predominância de produtos alimentícios - hipermercados e supermercados
			47.12-1	Comércio varejista de mercadorias em geral, com predominância de produtos alimentícios - minimercados, mercearias e armazéns
			47.13-0	Comércio varejista de mercadorias em geral, sem predominância de produtos alimentícios
		47.2		Comércio varejista de produtos alimentícios, bebidas e fumo
		47.3		Comércio varejista de combustíveis para veículos automotores
		47.4		Comércio varejista de material de construção
		47.5		Comércio varejista de equipamentos de informática e comunicação; equipamentos e artigos de uso doméstico
		47.6		Comércio varejista de artigos culturais, recreativos e esportivos
		47.7		Comércio varejista de produtos farmacêuticos, perfumaria e cosméticos e artigos médicos, ópticos e ortopédicos
		47.8		Comércio varejista de produtos novos não especificados anteriormente e de produtos usados
H				**TRANSPORTE, ARMAZENAGEM E CORREIO**

Table A.1 – Continued on next page

Table A.1 – Continued from previous page

Section	*Division*	*Group*	*Class*	*Denomination*
	49			TRANSPORTE TERRESTRE
	50			TRANSPORTE AQUAVIÁRIO
	51			TRANSPORTE AÉREO
	52			ARMAZENAMENTO E ATIVIDADES AUXILIARES DOS TRANSPORTES
	53			CORREIO E OUTRAS ATIVIDADES DE ENTREGA
I				**ALOJAMENTO E ALIMENTAÇÃO**
	55			ALOJAMENTO
	56			ALIMENTAÇÃO
J				**INFORMAÇÃO E COMUNICAÇÃO**
K				**ATIVIDADES FINANCEIRAS, DE SEGUROS E SERVIÇOS RELACIONADOS**
L				**ATIVIDADES IMOBILIÁRIAS**
M				**ATIVIDADES PROFISSIONAIS, CIENTÍFICAS E TÉCNICAS**
N				**ATIVIDADES ADMINISTRATIVAS E SERVIÇOS COMPLEMENTARES**
O				**ADMINISTRAÇÃO PÚBLICA, DEFESA E SEGURIDADE SOCIAL**
P				**EDUCAÇÃO**
Q				**SAÚDE HUMANA E SERVIÇOS SOCIAIS**
R				**ARTES, CULTURA, ESPORTE E RECREAÇÃO**
S				**OUTRAS ATIVIDADES DE SERVIÇOS**
			94.91-0	Atividades de organizações religiosas
U				**ORGANISMOS INTERNACIONAIS E OUTRAS INSTITUIÇÕES EXTRATERRITORIAIS**

Source: The Author