Rafael Rahal, Marcelo Dias

Development of a green venture classification method using Natural Language Processing

São Paulo, SP

2023

Rafael Rahal, Marcelo Dias

Development of a green venture classification method using Natural Language Processing

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Supervisor: Prof. Dr. Pedro Luiz Pizzigatti Correa

São Paulo, SP

2023

Catalogação-na-publicação

Rahal, Rafael
Development of a green venture classification method using Natural Language Processing / R. Rahal, M. Dias -- São Paulo, 2023.
74 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Machine Learning 2.Empresas Verdes 3.Data Visualization I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t. III.Dias, Marcelo

Resumo

O empreendedorismo sustentável tornou-se um tópico cada vez mais importante ao longo dos anos. No entanto, classificar e definir empreendimentos como sustentáveis tem se mostrado uma tarefa não trivial. Este artigo tem como objetivo criar um modelo de aprendizado de máquina baseado em Processamento de Linguagem Natural para classificar empreendimentos empreendedores como sustentáveis ou não. Utilizando tecnologia de ponta para modelos de linguagem, foi empregado um modelo BERT pré-treinado com ajuste fino para classificar os empreendimentos sustentáveis. Lidando com questões típicas da classificação de empreendimentos verdes, como o viés incluído na definição de sustentabilidade, o modelo alcança uma surpreendente precisão de 96%. Os resultados deste artigo auxiliam na classificação de empreendimentos sustentáveis para uma variedade de funções, como investidores de risco ou funcionários com metas ecologicamente corretas. Além disso, a rotulagem ecológica poderia ser detectada de uma perspectiva de aprendizado de máquina, e também foi possível reunir mais recursos para a classificação de empreendimentos verdes para futuros pesquisadores.

Palavras-chave: Sustentabilidade, Processamento de Linguagem Natural (NLP), Classificação, Empreendedorismo Ambiental.

Abstract

Sustainable entrepreneurship has become an increasingly important topic over the years. However, classifying and defining ventures as sustainable has proven to be not trivial. This paper aims to create a machine learning model based on Natural Language Processing to classify entrepreneurial ventures into sustainable or not. Using state-of-the-art technology for language models, a pre-trained BERT model with fine-tuning to classify the sustainable ventures was used to perform the task. Dealing with typical issues of classifying green ventures, such as the bias included in the definition of Sustainability, the model reaches astounding 96% accuracy. The outcome of this paper helps in the classification of sustainable ventures for a multitude of roles, such as venture capitalists or employees with environmentally friendly goals. Apart from this, eco-labeling could be detected from a machine learning perspective and also it was able to gather more resources for the classification of green ventures for future researchers.

Keywords: Sustainability, NLP, Classification, Environmental Entrepreneurship.

List of Figures

List of Tables

Contents

1 Introduction

Through this introduction, first it will be deeply explained the concepts and importance of Entrepreneurship and Green Entrepreneurship. After that, a broad view on Machine Learning topics, and the required theoretical knowledge for the creation of a language based model to classify green and non green ventures. Ending with the Methodology of this research.

1.1 Entrepreneurship Basics

Entrepreneurship can be understood as the art of making things happen with creativity and motivation. It consists of the pleasure of accomplishing with innovation any personal or organizational project, in a permanent risk situation. It is to assume proactive behavior when facing issues that need to be resolved. Entrepreneurship is the awakening of the individual to take full advantage of his rational and intuitive potential. It is the search for self-knowledge in a process of permanent learning, in an attitude of openness to new experiences and new paradigms. Entrepreneurial behavior drives the individual and transforms contexts. In this sense, entrepreneurship results in the destruction of old concepts, which, because they are old, no longer have the capacity to surprise and delight. The essence of entrepreneurship is in change, one of the few certainties in life. Therefore, the entrepreneur sees the world with new eyes, with new concepts, with new attitudes and purposes. The entrepreneur is an innovator of contexts. The entrepreneur's attitudes are constructive. They have enthusiasm and good humor. For him there are not only problems, but problems and solutions. Entrepreneurship, according to (SCHUMPETER, 2021), is a process of "creative destruction," whereby existing products or methods of production are destroyed and replaced by new ones. Already for (DOLABELA, 2006) corresponds to a process of transforming dreams into reality and wealth.

For (BARRETO, 1998) "entrepreneurship is the ability to create and build something from very little or almost nothing". It is the development of an organization as opposed to observing, analyzing, or describing it.

According to (DORNELAS J. C., 2008) an entrepreneur is one who detects an opportunity and creates a business to capitalize on it, taking calculated risks. In any definition of entrepreneurship we find, at least, the following following aspects concerning the entrepreneur:

- has the initiative to create a new business and passion for what he does uses available resources in a creative way,

- transforming the social and economic environment where he lives
- accepts to take calculated risks and the possibility of failure.

For (CHIAVENATO, 2004) entrepreneurship is the energy of the economy, the lever of resources, the impulse of talents, the dynamics of ideas. Even more: he is the one who sniffs out the opportunities and needs to be very quick, taking advantage of fortuitous opportunities, before other adventurers do so. The entrepreneur is the person who starts and/or operates a business to realize an idea or personal project taking risks and responsibilities and continuously innovating.

"Entrepreneurs can be said to fall equally into two teams: those for whom success is defined by society and those who have an internal notion of success" (DOLABELA, 2006, p. 44).

To be an entrepreneur means to possess, above all, the impulse to materialize new things, realize one's own ideas and dreams, and to experience personality and behavioral characteristics that are not very common in people. In my view, the common components in all the definitions of an entrepreneur: has the initiative to create a new business and passion for what uses available resources in a creative way, transforming the social and economic environment in which they live; they accept to take risks and the possibility of failure.

"The entrepreneur is someone who dreams and seeks to transform his dream into reality" (DOLABELA, 2006, p. 25). A person of any age can be an entrepreneur.

Economists realize that the entrepreneur is essential to the process of economic development, and in their models are taking into account the value systems of society, in which the individual behaviors of its members are fundamental. In other words, there will be no economic development without entrepreneurial leaders at its base.

It is no longer of any use to accumulate a stock of knowledge. It is necessary that we know how to learn. Alone and always. As an entrepreneur does in real life: by doing, making mistakes, learning (DORNELAS J. C., 2008).

The good entrepreneur, when adding value to products and services, is permanently concerned with resource management and the concepts of efficiency and effectiveness. (DRUCKER, 2002) does not see entrepreneurs causing change, but sees entrepreneurs exploiting the opportunities that change creates (in technology, consumer preference, social norms, etc.).

This defines entrepreneur and entrepreneurship: the entrepreneur seeks change, and responds to and exploits change as an opportunity. "The role of entrepreneurship in economic development involves more than just increase in production and per capita

income; it involves initiating and constituting changes in the structure of business and society" (HISRICH; PETER, 2004, p.33).

Entrepreneurship is a specific domain. It is not an academic discipline with the sense that is usually attributed to sociology, psychology, physics, or any other well-established discipline. We refer to entrepreneurship as being, first of all, a field of study. This is because there is no absolute paradigm, or a scientific consensus.

We know that entrepreneurship translates into a set of practices capable of guaranteeing the generation of wealth and a better performance to those societies that support and practice it, but that there is no absolute theory in this respect. It is worth emphasizing that it is of fundamental importance to understand this basic premise in order to correctly be able to interpret what is written and published on this theme.

Although entrepreneurship has been a subject for centuries, it was in the 1980s that it became the subject of studies in almost all areas of knowledge in most of the nations. Entrepreneurship, in all its aspects, has been assuming a prominent place in the economic policies of developed and developing countries and developing countries.

1.2 Objectives

As the intersection of technology and environmental responsibility becomes increasingly vital in the modern business landscape, this work endeavors to explore the synergy between green entrepreneurship and machine learning. By embarking on this journey, we aim to contribute valuable insights and tools that transcend traditional boundaries, fostering a deeper understanding of sustainable business practices.

1.2.1 General Objectives

The overarching aim of this research is to establish a connection between green entrepreneurship and machine learning. This involves the development of a predictive model designed to discern, based on textual descriptions, whether a company adheres to environmentally sustainable practices. This broader objective seeks to contribute significantly to the identification and promotion of environmentally conscious business initiatives.

1.2.2 Specific Objectives

1. **Model Development:** Develop a machine learning model proficient in analyzing and interpreting textual data from company descriptions, effectively classifying them as either green or non-green entities.

2. **Web Interface Implementation:** Create an intuitive and user-friendly web interface enabling real-time interaction with the developed model.

3. **Data Visualization Integration:** Incorporate data visualization components into the web interface to enhance user comprehension of model predictions. This includes graphical representations of the training dataset, providing insights of the given dataset.

1.3 Green Entrepreneurship

Climate change is a historic global problem that has now reached critical sizes. Under the influence of industrialization in the 20th century, the cause of climate change changed from natural to anthropogenic. Each industrial revolution has increased the environmental costs of economic growth, from exhaustible natural resources to production and consumption waste. The modern world is undergoing the Fourth Industrial Revolution, the main characteristic of which is total automation (United Nations, 2016). These processes have a direct consequence of growth of the energy intensity of economies.

In 2015 the international community adopted the Paris Agreement on Combating Climate Change as a collective response of humanity to the global challenges of climate change. This agreement, which came into force less than a year later and was signed by 196 countries, aims to significantly reduce global greenhouse gas emissions and limit the global temperature increase to 2 degrees Celsius within this century. As well as this, it aims to find means to further limit this increase to 1.5 degrees (United Nations, 2022). The agreement calls for all countries to commit themselves to reducing their emissions and work together to adapt to the impacts of climate change, and encourages countries to strengthen their commitments over time. The Agreement paves the way for developed countries to assist developing countries in their efforts to mitigate and adapt to climate change, while providing a framework for transparent monitoring and reporting on countries' achievement of climate goals.

Much attention has been paid to the issues of climate change in the existing literature. The relevance of combating climate change is emphasized in numerous works by scientists analyzing diverse aspects of this phenomena. One of the fundamental works is Huber Lamb's "Climate, History and the Modern World", that has inspired many academics to touch upon this topic: it overviewed the history of climate change and the record of human's reactions to it in the past, as well as an outline of main consequences caused by changes in the environment (LAMB, 2002). Many scientists worked on expanding the research. For example, Neville Brown, in "History and Climate Change", gave a European perspective of the issue, examining multiple socio-political and economical aspects while

trying to understand whether environmental changes can also serve as triggers for human history (BROWN, 2001).

Wolfgang Behringer, in his work "A Cultural History of Climate" underscored the importance of understanding cultural development for correct interpretation of climate change. He based his work on the studies of historians such as Emmanuel Le Roy Ladurie, Hubert Lamb, Brian Fagan and Lucien Boia to show the historical changes of the social and political meanings of climate change (BEHRINGER, 2010). In "Why We Disagree About Climate Change", M. Hulme expanded this concept to show why the narrative of climate change is malleable enough to be exploited in different ways and given new meaning to (HULME, 2009). S. Bronniman also analyzed the ambivalence of the concept and the use of visual representations of climate, that are sometimes given new meaning to and often in controversial ways (BRÖNNIMANN, 2002). In later works, a drastic change of tones can often be seen with scientists not questioning the nature of environmental issues anymore. For example, S. Fawzy, A. Osman, J. Doran and D. Rooney give a general overview on the decidedly negative consequences of climate change and move on to assess different strategies on mitigating it (FAWZY et al., 2020). B. O'Neill and others, in "Achievements and needs for the climate change scenario framework", also tend to focus on the strategies against climate change rather than the process itself. They find that the research community has recently worked out a unified approach towards the studies, having created a scenario framework on which it builds policy planning (O'NEILL et al., 2020).

Given the high degree of elaboration of the general issues of combating climate change, the role of entrepreneurship in this process has also been widely analyzed in literature: J.K. Hall and G.A. Daneke (HALL; DANEKE; LENOX, 2010), I. Okumus (OKUMUS, 2013), G. Berle (BERLE, 1993). The importance of studying green entrepreneurship is proved by the trends in academia: researchers often point out the negative effects on climate caused by businesses. For example, B. Mrkajic, S. Murtinu and V. Scalera underlined that business activities are usually unsustainable and thus have to be widely reviewed to eliminate their negative consequences (MRKAJIC; MURTINU; SCALERA, 2019). D. Pachecho, T. Dean and D. Payne examined how international organizations, such as the United Nations, worked on raising awareness of unsustainable business practices and changing business management to alleviate its detrimental environmental effects (PACHECO; DEAN; PAYNE, 2010).

Indeed, the activity of businesses can be seen to affect climate change. K. O'Neill and D. Gibbs emphasize the importance of green entrepreneurship by examining it through the aspect of green marketing, or tactical and strategic marketing orientation. They come to a conclusion that the inability of some businesses to transition to more environment-friendly practices only signify their blindness to the array of opportunities offered in the

field of green entrepreneurship, because old-style unsustainable business strategies hinder both the development of companies and the global improvement of climate change (GIBBS; O'NEILL, 2016). That is why D. Etsy and A. Winston, in "Green to Gold: How Smart Companies Use Environmental Strategy to Innovate, Create Value, and Build Competitive Advantage", underline the need to show the benefits of green practices to entrepreneurs, because so many opportunities in the field allow businesses to benefit from them and take advantage of failing old-style markets (ETSY; WINSTON, 2006).

In recent decades, there has been an ever closer relationship between economic development and changes in the environment, and the mutual influence of both ecology on economic development and the results of economic activity of the world community on the state of the natural environment is increasing. As a result of the current growth in the scale of economic activity of people, there is a catastrophic destructive effect on the ecosystem, which leads to an increase in the global environmental crisis. The destruction of elements of the environment irreversibly leads to a shortage of resources and, accordingly, to the emergence of new economic problems, and also endangers the life and development of future generations. With the growth of postindustrial societies, huge enterprises have more opportunities to become sustainable, and can play a big role in supporting global ecology. Not only does academia see green practices as valuable for companies, some even argue that sustainability is a question of 'survival and prosperity' of every enterprise (BAKER; SINKULA, 2005).

In recent decades, there has been an ever closer relationship between economic development and changes in the environment, and the mutual influence of both ecology on economic development and the results of economic activity of the world community on the state of the natural environment is increasing. As a result of the current growth in the scale of economic activity of people, there is a catastrophic destructive effect on the ecosystem, which leads to an increase in the global environmental crisis. The destruction of elements of the environment irreversibly leads to a shortage of resources and, accordingly, to the emergence of new economic problems, and also endangers the life and development of future generations. With the growth of postindustrial societies, huge enterprises have more opportunities to become sustainable, and can play a big role in supporting global ecology. Not only does academia see green practices as valuable for companies, some even argue that sustainability is a question of 'survival and prosperity' of every enterprise (BAKER; SINKULA, 2005).

It has to be pointed out that classifying green ventures has its specificities and unsolved issues. The problem of determining a green enterprise lies in the term 'green entrepreneurship' being ambiguous and its definition varying from research to research. There is no single approach, the meaning tends to go from being narrowed down to just ecological practices to broadly encompassing the societal impact of a company. The

United Nations Environment Programme takes on a wider approach and emphasizes the importance of the green economy. In that way, it encourages transitions to economies that are "low carbon, resource efficient and socially inclusive".

L. Cekanavicius et al., in their careful analysis of existing literature and documents on the topic, come to define green business as "an organization that is committed to the principles of environmental sustainability in its operations, strives to use renewable resources, and tries to minimize the negative environmental impact of its activities." (ČEKANAVIČIUS; BAZYTĖ; DIČMONAITĖ, 2014), thus encompassing both environmental and social practices of a company. They see green enterprises as sustainable in the sense of minimizing their harm environmentally, socially and economically, both in the short- and the long-term.

The common practices that companies can deploy can be summarized as the 4 'R's that stand for recovery, reuse, recycling, reduction, and which are often used by large international companies. The 'R's can signify multiple phenomena; for example, reduction can denote reducing both consumption and emissions (KASSAYE, 2001).

Moreover, to signify their sustainability, companies engage in ecolabeling to increase customers' interest. The bias created by greenwashing brings even more complexity to the classification task. There are two ways of defining the businesses: through self-labeling or through becoming part of existing labeling schemes. Self-labeling means that enterprises subjectively define themselves as green or ecological, and have to make sure their clients understand the labels and value them. On the other hand, companies can apply to be recognized by developed eco labeling schemes, which requires more specific changes in the enterprises and can impose requirements that are harder to meet.

L. Cekanavicius et al. conduct an empirical analysis to determine what companies perceive as relevant practices when they identify themselves as green. Among others, the following were often mentioned: manufacturing ecological products, organizing "a day without a car", funding external environmentally-friendly projects, organizing seminars on ecological issues. Even more prominent were practices like switching off lights and computers, maintaining non-smoking offices or "offices without paper" (ČEKANAVIČIUS; BAZYTĖ; DIČMONAITĖ, 2014). All these can be taken as examples of what subjective self-labeling could be based on when companies have the freedom to define their sustainability without specific criteria.

Moreover, another way of labeling would be through industry classification. Defining an industry as sustainable can bring other problems to the table. Taking into consideration L. Cekanavicius et al. analysis, defining a whole industry as green is not granular enough. Solar panel creators might have a clear green impact but might be on a day-to-day basis not sustainable. Which also needs to be taken into account if the production of said solar panel would be made in an environmentally friendly way. Therefore, the identification of

green as a whole industry lacks the granularity of the company's behavior and outcome.

As it will be explained further, in order to perform a classification task, the machine needs input data and a label. The machine will learn how the data maps to the label and if the label is not well-defined, then so will be the classification. All these difficulties raise the task to classify ventures to a problematic position.

Nevertheless, the task is worth tackling for several reasons. As green entrepreneurship becomes more valued, more funds are being created for it. Grants for green ventures end up facing the same problem that this paper tries to achieve, classifying what a green venture is. For the funds, it needs to define what a green enterprise is, so that the grant can be given correctly. Since companies could be self-labeling and applying for such funds for the extra money without really having the benefit to the environment, which would be the original intent of the financial support.

Also, many venture capitalists (VC) try to steer their funds into environmentally friendly ventures. As some of them have interest in the green objective, in order to invest in companies that share this objective, they also fall under the same situation, classifying green ventures. To better define a green venture, would be to better steer the money of the VC and therefore bring more success to its investments.

Finally, as more people give importance to green entrepreneurship, it becomes a reason for employees to find employers that follow such practices. Therefore, once again, another case where classifying green ventures becomes the problem. As employees would prefer to work on such ventures that would accomplish their goals that go beyond financially, but ideologically.

1.4 Machine Learning

The definition of machine learning is "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." (MITCHELL, 1997).

That is, machine learning is a subfield of artificial intelligence dedicated to the development of algorithms and techniques that allow the computer to learn, that is, that allow the computer to improve its performance in some task.

There are several situations in which machine learning is desirable. In general, computer systems capable of learning are used to solve problems that cannot be solved by traditional programming methods, such as imperative, functional or object-oriented methods (PISTORI, 2003; PRATI, 2006). In principle (PRATI, 2006), there is still no known algorithm implemented by traditional programming methods that is capable of recognizing,

for example, handwritten characters. However, using machine learning techniques it is possible to design a computational system that learns to recognize handwritten characters, through the observation of a large amount of manuscripts.

We can say that in traditional programming paradigms, the designer system or developer is exclusively in charge of finding the implementable computational representation of the solution to the problem to be solved (MITCHELL, 1997; PISTORI, 2003). On the other hand, machine learning methods offer the designer system resources to create a computational system capable of obtaining an automatic (or semi-automatic) solution, achieved from particular examples of the problem (MITCHELL, 1997).

The same understanding can be superimposed on other cases, in which a computer system must observe a set of facts and be able to distinguish features of interest in those facts (eg, observing a string of characters and discriminating the DNA sequences). According to (PISTORI, 2003) the same learning environment can be used to solve problems that are different from the one originally proposed.

In order to apply machine learning, it is necessary to have a mass data of training and testing, with several attributes that we deem significant in addition to the results we expect for each of the reported elements. In other words, so that the computer can learn it needs the data and the answers.

A practical example already widely used is the optical recognition of characters (OLIVEIRA et al., 2006). It presents a matrix with (n x m) elements, where each element represents a painted or unpainted pixel on an input device. After training initial start, the computer may be able to recognize characters that are similar to those presented previously.

Another example would be the prediction based on climatic factors (CASSOLA; BURLANDO, 2012). In this case, there is a database containing the conditions of weather (visibility, temperature, relative humidity and wind speed) of previous rains and the result of whether or not there was rain that day. In possession of this previous mass of data, we can apply this base to a training algorithm, so that, after training, he can predict whether, with the current weather conditions, there will be rain tomorrow or not.

One factor that influences the result is the amount of information (elements) contained in the test mass, since, in many cases, the existing information is not wide enough for adequate training. To circumvent this problem, there are techniques such as “cross validation” that, for example, promotes through iterations, an increase in comparisons between the few elements, promoting an increase in training and leading to a more satisfactory result (WITTEN; FRANK, 2005).

1.4.1 Neural Networks

The basic Machine Learning architecture would be an Artificial Neural Network (ANN). An ANN is composed of a set of computational elements called neurons, which relate the output values and input by the equation:

$$y^{ij} = f(\sum_{i'}^{n} y^{i'(j-1)} w_{i'}^{ij} + b^{ij}) \tag{1.1}$$

Neuron Output

where y^{ij} = output value of neuron i of layer j; n = number of neurons in the previous layer; $y^{i'(j-1)}$ = output value of neuron i' of the previous layer; wi'^{ij} = value of synaptic weight of neuron i of layer j, activated by the neuron i' of the previous layer; b^{ij} = value compensation of neuron i of layer j; f = function of activation of the neuron i.

The development of an ANN consists of determining its architecture, that is, the numbers of layers and of neurons in each layer, as well as adjusting its free parameters w's and b's, this phase known as training. The architecture varies according to the complexity of the problem and cannot be defined before training, constituting a search based on trial and error (HAGAN; DEMUTH; BEALE, 1997).

Furthermore, ANN can be used as tools of interpolation (SáRKöZY, 1999), and its ability to learn to differentiate input parameters makes them capable of solving very complex problems in several areas of knowledge. Not even, its effectiveness has been proven to be able to approximate any continuous function with only 2 layers of depth (FUNAHASHI, 1989), also known as the Universal Approximation Theorem.

When it comes to training, the algorithm that allows this adjustment of weights in an ANN is called Backpropagation (CUN et al., 1990). It works from partial derivatives of the function that calculates the error to be reduced (also called cost function). From the calculation of these derivatives, a gradient is found, which indicates in which direction the weights must be adjusted (up or down). In the backpropagation algorithm this calculation is performed from the last layer to the first layer. It means that the error in the last layer is calculated and then the partial derivatives of this error are propagated from the last layer to the first.

When it comes to optimizing the network to perform a certain task, the Backpropagation is used with the Gradient Descent algorithm. This is an optimization algorithm that tries to find the minimum in a function. Considering that a function that calculates the error of the network it's used, it tries to minimize the error. Backpropagation is a way of calculating the gradient for all the layers in the ANN, and Gradient Descent uses this gradient to find the minimum (GOODFELLOW; BENGIO; COURVILLE, 2016). Figure 1 shows visually the Gradient Descent method, with the gradient it can be known the

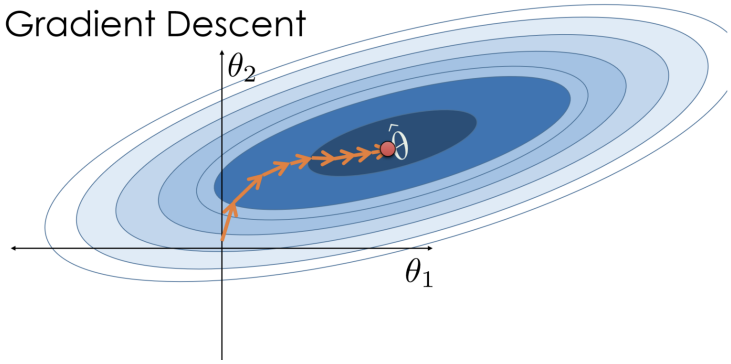


Figure 1 – Gradient Descent visualized (GOODFELLOW; BENGIO; COURVILLE, 2016)

direction where the function decreases the most, and following this path it eventually leads to the point of minima.

1.4.2 Recurrent Neural Networks

Recurrent Neural Network models (RNR) (GOODFELLOW; BENGIO; COURVILLE, 2016) have an architecture that favors the storage of the history of precedent terms, thus being very useful for creating models of language because it does not need a window limitation. The central idea of this architecture consists of the feedback of the sequential elements, so that the input of each one of them serves, not only for the prediction of the next item in the sequence, but also for the formation of an intermediate component, a state. These states, represented in Figure 1 as h's, are in practice matrices, and function as a kind of condensed memory of the preceding elements. They also enter as input to later states. This is a way of relaying at each moment the effects of previous inputs to the rest of the sequence (GOODFELLOW; BENGIO; COURVILLE, 2016).

In the graph of Figure 2, consider x and y to be vectors of any dimension and the state h as an array. The graph's arrows represent the weight matrices. The states are calculated from the recurring equation:

$$h^i = f(h^{i-1}, x^i) \tag{1.2}$$

RNN output

State $h(0)$ is normally initialized randomly and enters, together with the first input (the first term of the sequence), in state $h(1)$. The target of this first step is the second term of the sequence (y^i). Then the second term in the sequence targets the next term, and so on. After training the network and applying the back-propagation algorithm, it is expected that the last state has incorporated a certain memory of all previous states and captured the dependency relationships between the terms.

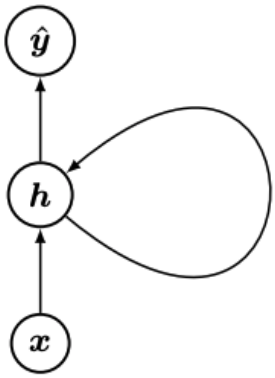


Figure 2 – RNN Computational Graph (GOODFELLOW; BENGIO; COURVILLE, 2016)

This architecture of this network was widely used in language applications, since in any of these cases, it was needed to have memory of past inputs. A classic example of its use is on translators (NGUYEN et al., 2018) . It is also one of the first solutions for Natural Language Processing problems.

1.4.3 Natural Language Processing

Over the last decades, Machine Learning has been greatly developed and became a very useful tool. Among its enormous plethora of applications, there is Natural Language Processing (NLP). NLP is the intersection of artificial intelligence and linguistics (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), applying statistical methods to text understanding. Created in the 1950s during the cold war, one of the first applications was a translator from Russian to English (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), but with time, this method developed and grew into a big contemporary research topic. With it, many challenges in many different areas can be tackled. E.g. in the Entrepreneurship area, a text-based innovation metric can be used to evaluate ventures (BELLSTAM; BHAGAT; COOKSON, 2020).

NLP computationally handles the various aspects of human communication, such as sounds, words, sentences and speech, considering formats and references, structures and meanings, contexts and uses. In a very broad sense, it can be said that NLP aims to make the computer communicate in human language, not always necessarily at all levels of understanding and/or generation of sounds, words, sentences and speeches (ALLEN, 1995). These levels are:

- Phonetic and Phonological: the relationship of words with the sounds that produce;
- Morphological: from the construction of words from units of meaning primitives and how to classify them in morphological categories;

- Syntactic: from the relationship of words to each other, each one assuming its structural role in sentences, and how sentences can be parts of others, constituting sentences;
- Semantic: the relationship of words with their meanings and how they are combined to form the meanings of sentences; and
- Pragmatic: from the use of phrases and sentences in different contexts, affecting the meaning.

When handling the text to be used in a NLP model, some steps need to be taken in order to prepare the data accordingly. These steps are: lexical analysis, elimination of stopwords and stemming (BAEZA-YATES; RIBEIRO-NETO, 1999).

In the lexical analysis step, also known as tokenization, it is done the conversion of the whole text into an array of words or tokens, splitting the full text, basically trimming the spaces. Thus, the main objective of this step is to identify the words that constitute the text. In the next phase, articles, prepositions and conjunctions are eliminated by removing the stopwords list. This is a list defined in each language of words that are not semantically useful in a phrase. Then, lexical normalization can be performed through stemming, since often the user specifies a word in the text, but only variations of this word are present in a relevant document. E.g. the word “go” and the word “went” become both “go” after this process.

Other methods of text-based understanding are such as the Computer-Aided-Text-Analysis (CATA) methodology. This method consists of the use of computations to analyze text, such as counting the frequency of words, counting the space between words and more (Short, McKenny and Reid, 2018). With this information, much can be accomplished, e.g. it can be used to understand literary text, giving insights on how the words are used and even understanding how their adoption changed throughout the years (Palmquist, Carley and Dale, 2020). Also among the uses of CATA methodology, a dictionary of words can be constructed. A dictionary would be a group of words that represent a theme and can be used to classify how much a text is connected to a certain topic. Dictionaries for human rights, sustainability, and others, were created to aid researchers on their text-based analysis around these topics (PENCLE; MăLăESCU, 2016) , as it can be seen on Table 1.1.

However, with this kind of methodology, we are bound to classify a venture based only on basic metrics extracted from the words. What this paper tries to develop is a new and better approach to text-based sustainability metrics, by using Machine Learning, more specifically, NLP to reach better results. With the implementation of artificial intelligence, we can find connections between words that a simple text analysis method cannot do. Boosting the understanding of text to more than just the use of certain words connected

Dimension	Example of words	Number of words
Employee	Adopted Child, Health Benefits, Educate, Employed, Discriminatory, etc...	319
Human Rights	Aboriginals, Fairness, Oppressive Regime, Same Sex, Religious Diversities, etc...	297
Social and Community	Transparent, Foodbank, Indiginous People, Social Issue, etc...	174
Environment	Acid Rain, Conservation, Fossils, Green Engineering, Renewable Energy, etc...	451

Table 1 – Dictionary dimensions, example of their content and their size (PENCLE; MăLăESCU, 2016)

to the topic, but also extract information from their meaning and use that for a better classification.

1.5 Methodology

The task proposed to be achieved with the development of this study is a text classification (YANG, 1999). We aim to assign the text description of a company into the categories of "green" or "not green". As explained, our approach of doing that is through a NLP model. The model that will be used is the fine-tuned state-of-the-art model, BERT (DEVLIN et al., 2018), which will be deeply explained in further sections. For its fine tuning, it will be added an Artificial Neural Network with one output neuron that will represent the classification probability, "1" for 100% probability of being green and "0" for the opposite. The dataset used, which is also going to be better explained later on, is given by the Crunchbase database. The language used will be python and the development environment, Jupyter Notebook. Furthermore, the results will then be compared with the CATA counting words classification method.

2 The Machine Learning Model

2.1 Model

The model chosen for this project is a fine-tuned version of BERT (DEVLIN et al., 2018). This is a state-of-the-art model that uses the Transformer architecture and newest techniques, providing impressive results. It was open-sourced by the creators allowing easy access from anyone in the community, as well as having great documentation. Making it a great option for a project like the one this document tries to achieve. Furthermore, in order to achieve the objective proposed the fine-tuning part is essential when it comes to a pre-trained model like BERT. A feed forward neural network layer was added in the end to perform the classification, where it connects the output from BERT to a 2 neurons layer, representing the outcomes "not green" and "green".

The network architecture is represented in Figure 2.1. As it can be seen, before the text is fed into the model, there is a pre-processing part stated as Tokenization. As explained in the NLP section, this step processes the text data into a way that the model can understand. In the case of BERT, it is a bit different than the usual tokenization process, since the language model needs a specific type of input, as it will be further explained. And in the end of the diagram, it can be seen both of the neurons that represent the classification in "green" and "not green" classes.

2.1.1 Transformers and the Attention Mechanism

Transformer is an architectural concept for Neural Networks (VASWANI et al., 2017), which revolutionized the language models by adding the concept of attention. Attention mechanisms are a family of computational methods that are inspired by the human concept of attention to provide algorithms with the ability to learn to focus their resources and assign different degrees of importance to different aspects of inputs, which usually leads to an improvement in the results obtained in several applications.

More than that, neural attention is a correlation detection mechanism. In the context of natural language processing applications, this mechanism allows the detection of correlations between components of a text, such as words, in order to map the surface structure of language and its context and ordering relationships.

An example using the case applied for this work: we receive a small text in natural language that represents the description of a company and we wish to classify how sustainable the company is.

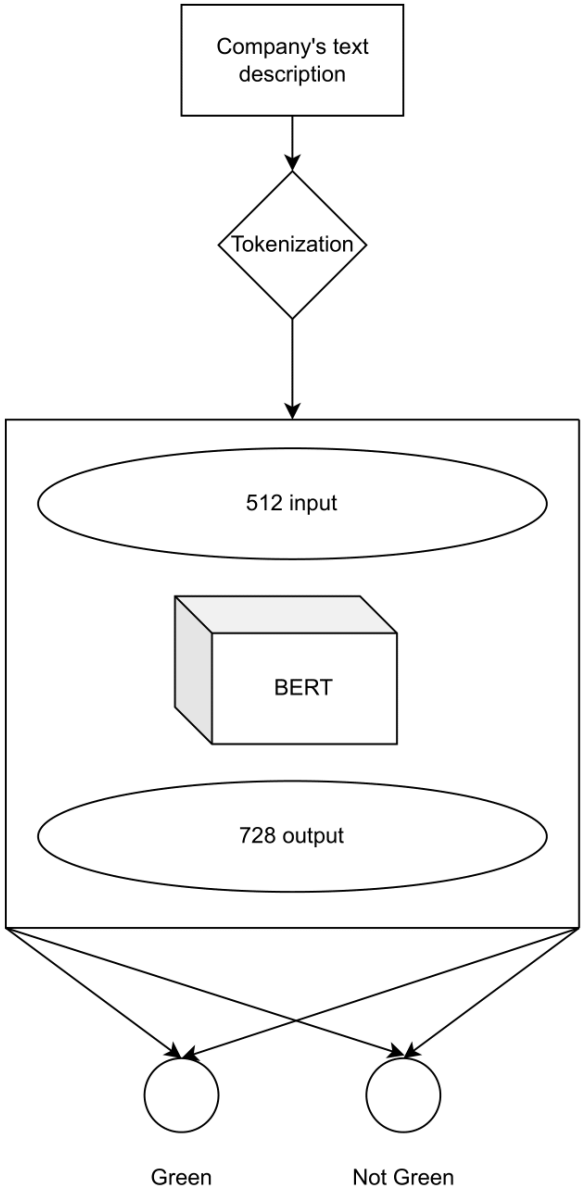


Figure 3 – Network Architecture

To do so, we receive the text in the form of a sequence of representations vectors $V = \{v_0, \cdots, v_n\} : V \in \Re^{n \times d}$, one for each word that makes up the text. Our proposal is to generate a vector representation c for the entire text, of the same dimension d of vectors v_i, which captures the full context of the description, using a large amount of representations c to train a small neural network to perform the classification of sustainability.

A trivial way to generate the c representation of the description would be adding the v_i representations of the words that compose it. However, there could be a group of words, in specific, which can be more important than others when it comes to establishing whether the company is green or not: the adverbs, for example, explaining how things are done.

Knowing this, a simple attention mechanism can be used to make the algorithm pay more attention to adverbs than other words, giving greater importance to them when generating the representation of the text description c.

To do so, it is sufficient to define the level of attention given to each word as a function of the similarity between that word and the pattern vector u, using attention levels generated as weights to balance the sum of vectors in the generation of c. Not coincidentally, it is precisely this procedure that this simple mechanism of attention, explained in Equations below, performs:

$$e_i = (u, v_i) \tag{2.1}$$

$$a_i = softmax(e_i) = \frac{e_i}{\sum e_i} \tag{2.2}$$

$$c = \sum_i a_i v_i \tag{2.3}$$

Basic attention mechanism equations

First, it is calculated a similarity coefficient e_i between each representation vector v_i of the input sequence and the pattern vector u, using a function of similarity : d d . There are many options in the literature for similarity (LUONG; PHAM; MANNING, 2015).

Then it normalizes the similarity coefficients e_i, using the softmax normalization function and thus generating the so-called attention coefficients a_i. The algorithm then generates the vector c by making a sum of the input vectors vi, balanced by attention coefficients a_i, so that the vector c is nothing more than a combined measure of linear correlation between the standard vector and the representation vectors.

The representation of the company description will therefore have much more influence from the words that have high correlation with the pattern vector u, that is, of words that have adverbial function.

In general, what the Transformer does is find new representations for the sentences, by adding iteratively, through the connections of leak, their previous representations with autocorrelation measures of those same representations; the algorithm learns, through its attention functions, which parts of these measures should be intensified and which should be mitigated, generating thus representations, for each word of the sentence, that encapsulate their relations of context with the other words in the same sentence. The result of this process is illustrated in Figure 4.

In Figure 4, it is possible to see a color map representing the attention score, that is, the measure of autocorrelation, given by the Transformer each word of a sentence to measure its relationship to the word “it”. It’s possible to note that the values assigned

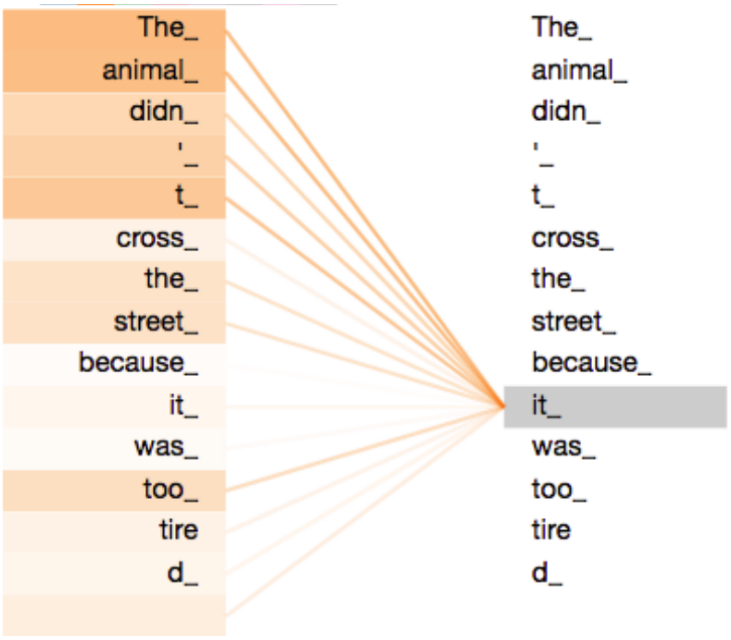


Figure 4 – Transformer attention engineer output (Alammar, 2018)

to "The" and "animal", which represent the referent of the word "it", are prominent, indicating that this relationship was learned by the algorithm through its representation of autocorrelation.

The architecture of Transformers resembles the Encoder-Decoder model. An Encoder-Decoder mapping model (Bahdanau, 2014) is a system composed of two RNN's whose main function is to map one sequence to the other. Encoder-Decoder models have been widely used in linguistic tasks, especially in the development of dialog systems and in machine translation.

In a translation context, for example, the model receives as input a sequence from a source language and produces as output a sequence in a target language. The generated sequence must, in addition to preserving the semantic content of the source sequence, present a syntax accepted by speakers of the target language.

It can also be used in image applications, in the example of Figure 5 specifically for image segmentation. Instead of RNNs, in this case the model consists of two Convolutional Neural Networks (CNN). The architecture basically consists in getting the input data and shrinking to a lower data representation, so it can then grow into the same format as the input but applying some kind of transformation. That transformation being a translation of a phrase from one language to another, or a segmentation of an image.

Finally, the model learning method is similar to an ANN: the model outputs are compared with the targets, the error is propagated back through the backpropagation algorithm, the network weights are updated in order to decrease this error and the process is repeated until the completion of all epochs.

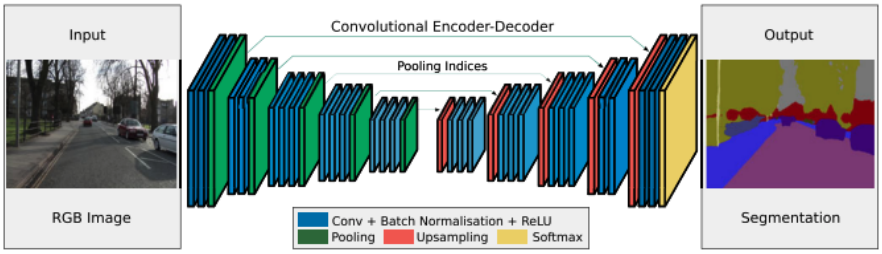


Figure 5 – Encoder-Decoder architecture for image segmentation (Badrinarayanan, 2017)

2.1.2 BERT

As stated earlier, Transformer achieves state-of-the-art results for automatic text translation, however it is not trivially adaptable to other tasks. After the publication of Transformer, a series of research papers in NLP focused precisely on the challenge of creating architectures based on it that could be equally successful in performing other language tasks such as automatic classification of texts (PETERS et al., 2018).

These efforts culminated in the proposition of BERT, the Bidirectional Encoder Representations from Transformers (DEVLIN et al., 2018). BERT was able not only to intelligently adapt the Transformer structure, but also to link this structure to the concepts of transfer learning and semi-supervised learning. Such association allows BERT to perform different linguistic tasks with practically the same process of training, obtaining state-of-the-art results for many of them, including those that have little data for training.

In the light of the explanation given about the Transformer's architecture, the architecture of BERT, illustrated in Figure 6, is very simple to understand. BERT consists of the Transformer's encoding component, that is, a composite of Encoders of Transformer, which receive a matrix of dense vector representations of a sentence and return an attention coding array. Where it is fed into a small ANN to perform the classification, in the case of the project in this paper.

BERT models are slightly larger than Transformer in terms of parameters: `BERTBASE` has $N = 12$, $d = 768$ and $h = 12$ and `BERTLARGE` has $N = 24$, $d = 1024$ and $h = 16$, while the original Transformer had $N = 6$, $d = 512$, and $h = 8$. While N is the number of encoders, d is the size of the hidden layer and h is the size of the attention vectors. The architectural differences are superficial; the contrast between BERT and the original Transformer mainly in the ways in which the data is provided and how the training is accomplished.

Instead of a single input sentence S_0, BERT receives as input a sequence = [CLS], S1, [SEP], S2, [SEP], consisting of a special symbol of agglutination [CLS], a sentence S_1 and a sentence S_2, both finalized by a special separation symbol [SEP]. The representations of sentences are generated analogously to the Transformer, with segmentation into sub-words,

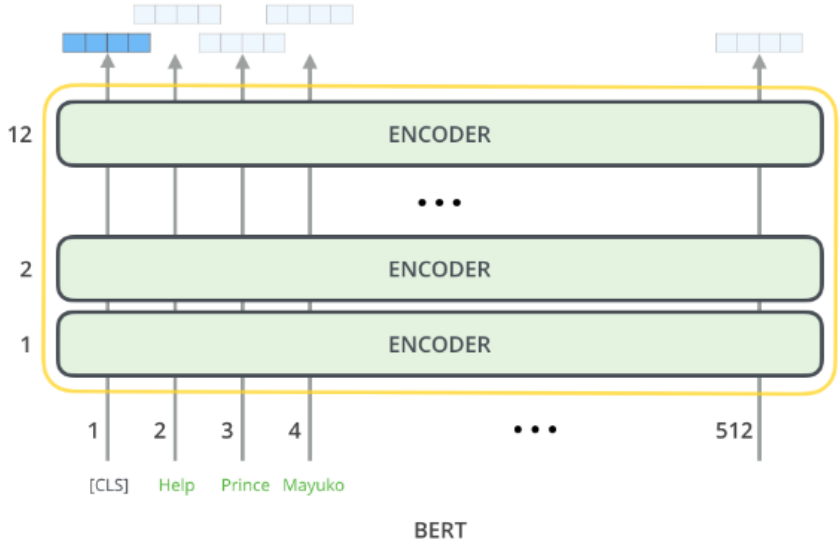


Figure 6 – BERT Architecture (Alammar, 2018)

using in this case only the WordPiece algorithm, but in addition to position embedding, used in Transformer, BERT adds to each vector representation also a learned sentence embedding, which differentiates whether the token belongs to S_1 or S_2. From this input, BERT then returns an array that contains a representation of attention for each of the words of sentences, including the special agglutination word [CLS].

Both the input and the output of BERT are defined according to the generic structure that it has to take on in order to solve the number of distinct linguistic tasks that it proposes to solve. The input consists of two sentences because many of the linguistic tasks it solves are characterized by two sentences, as is the case with question answering, characterized by a question and an answer sentence, or the inference task, characterized by a premise and a conclusion sentence. The output is generic because, for each of these tasks, a layer will be connected extra to BERT, which will receive the attention representation matrix and use it to generate an appropriate output for that task.

BERT Training is divided into two stages: pre-training and fine-tuning. Pre-training is an unsupervised training process that applies to all target tasks. It consists of two unsupervised tasks: masked language and next sentence prediction. In masked language modeling, 15% of the words in the input sentences of the training set are replaced by other symbols, as indicated in Table 2. The task is to find out what the substituted word was.

In predicting the next sentence, 50% of the sentence pairs contained in the set of training have the sentence S_2 replaced by another random sentence from the set. The task is to predict whether or not S_2 is the next legitimate sentence, given S_1, using for this, only the attention coding c returned by BERT for the special symbol of agglutination [CLS] of S_1

Substitution Type	Frequency
Substitution for the special symbol [mask]	80%
Substitution for a random word	10%
No substitution	10%

Table 2 – Word substitution protocol for masked language model task from BERT (DEVLIN et al., 2018)

The two tasks are performed concurrently during pre-training, where their combined cost is minimized. This process allows BERT to learn values for the parameters of your Encoders that are coded with the linguistic relationships between different words and their contexts.

For each distinct linguistic task, as mentioned before, BERT is coupled with an extra layer, geared towards that specific task, and supervised training is performed using data for the target task, with the encoder parameters starting from values obtained in pre-training. This is the process called fine-tuning.

In the case of classifying texts in k distinct classes, the improvement consists of provide the attention coding c returned for the agglutination symbol [CLS] to a simple ANN, defined by a matrix WC dk, followed by a softmax, using the result as a probability distribution over the possible classes to perform the forecast. A common classification cost function is then used to adjust both the parameters of the ANN and the Encoder stack.

Still in the case of classification, it is used degenerate sequences of the form = [CLS], S1, [SEP], null, [SEP] during refinement, as the rating is a task that uses a single sentence and not two.

For reasons of computational efficiency, the sequences S, used as input to the BERT, have their size, in words, usually limited by a maximum of |S|. At the original article, during training the values |S| = 128 for 90% of cases and |S| = 512 for the remaining 10% in order to maximize training efficiency without inhibiting learning of position embeddings for longer sentences (DEVLIN et al., 2018). It is also possible, depending on the implementation, to complete sentences with words null until they reach the standard size |S|, in order to optimize efficiency.

The motivation for this architecture is the idea that pre-training could learn general linguistic knowledge in a robust way, allowing training periods much shorter to adapt this knowledge for specific tasks. The hypotheses behind the model proved to be true: not only the improvement time is, in practice, actually much less than pre-training, one hour on one TPU versus four days on 16 TPUs, but also as a single pre-training was really capable of being used to perform several different linguistic tasks, obtaining state-of-the-art results (DEVLIN et al., 2018).

In addition, the authors of BERT made available not only the source code, but also

a pre-trained version of the multilingual model. This version has similar dimensions to the BERTBASE model and has already been trained in 104 different languages. For a language with fewer resources of training it is possible to use a monolingual version of BERT in a language with more features, such as English or Chinese, automatically translating training data into those languages, or using pure multilingual version, performing the improvement training with original data in the target language, being able to or not to take extra pre-training steps using data if it is required. Experiments carried out by the BERT developers themselves indicate that the latter option tends to present better results in this case (DEVLIN et al., 2018). Recent works even indicate that the multi-lingual model is capable of generalizing linguistic knowledge between different languages in performing tasks (PIRES; SCHLINGER; GARRETTE, 2019).

These reasons make BERT one of the main, if not the main candidate algorithms currently for performing text classification tasks and several others linguistic tasks, as well as Transformer and its based models are the main candidates for translation assignments. This is without taking into account the additional factor that, because they are recent, a lot of scientific knowledge about the algorithms themselves and their and science in different applications can still be produced and aggregated.

2.2 Dataset

When it comes to any machine learning model, the input data is one of the most important steps of the process. For this specific case, we are trying to create a model that can classify sustainable and non sustainable ventures based on their text descriptions. Therefore, we need a dataset that contains the text description of the venture, and the label, showing if the company is sustainable or not.

For the text description, a dataset from the website Crunchbase was used (RAHAL; DIAS, 2023). This is a company that provides a database with an extensive amount of information about each venture. Its business allows companies to analyze their own data and then come up with strategies to increase their revenue, sales, growth, etc. With that, Crunchbase allows users to have information on funding and how to empirically increase the probability of securing an investment round.

However, we use the gigantic plethora of companies in Crunchbases database to feed our model with diverse text descriptions. Varying from many types of ventures, it allows us to train our model optimally, avoiding overfitting to certain types of companies. These text descriptions are bound to be slightly biased, since they are set by the own company.

There are two text descriptions available in the database, a smaller and a more complete description. When training the model, as it will be further explained, it was

trained in the both descriptions to compare their outcome. In the next section, Training, there will be a deeper dive into each hyperparameter and their comparisons. For the same company, Spotify, for example, the longer description and the shorter description would be respectively:

"spotify is a commercial music streaming service that provides restricted digital content from a range of record labels and artists users can browse through the interface by artist album genre playlist record label and direct searches it also enables individuals to create share and edit playlists with other users if users want recommendations they can integrate their system with lastfm an application that provides music recommendations based on listening history the radio feature installed in spotify creates random playlists for its users that are related to preferred artists social media integration is a popular feature that enables users to connect their spotify accounts to their facebook and twitter profiles this enables them to access their friends favorite music and playlists and share their choices with others as well spotify users can purchase their preferred content through partner retailers spotify is available for mobile device platforms such as android blackberry boxee ios linux meego squeezebox windows mobile and more"

"Spotify is a commercial music streaming service that provides restricted digital content from a range of record labels and artists."

Tokens, as explained before, are the final outcome of the tokenization process. After applying stemming, and removing the suffixes of the word, and also not counting stopwords. Given that, the short description with approximately 35 tokens in average was compared to the long description, with about 300 tokens in average. As it can be seen in Figure 7, the histogram of the amount of tokens for each dataset, first using the shorter version and the second one with the original longer version.

As it can be seen in the two histograms, they represent quite different curves. The histogram is cut out at 512, since that is the biggest input size that the model BERT can receive. In the first histogram, the curve peaks at around 30 tokens and has its maximum at 67 tokens. The second histogram represents a more skewed normal distribution and has a clear cut off at 512, showing that there are about 200 companies with more than 512 tokens.

Having more tokens does not necessarily mean having a better result, because there is a trade off between size and efficiency. Having more tokens means having more information, therefore better precision, but also means bigger complexity. That increases drastically the time in training.

The second part of the dataset (RAHAL; DIAS, 2023) is the label, since that is what the model tries to achieve given a specific input. In the Crunchbase database, companies can also add in which category their ventures lie in. The category list goes

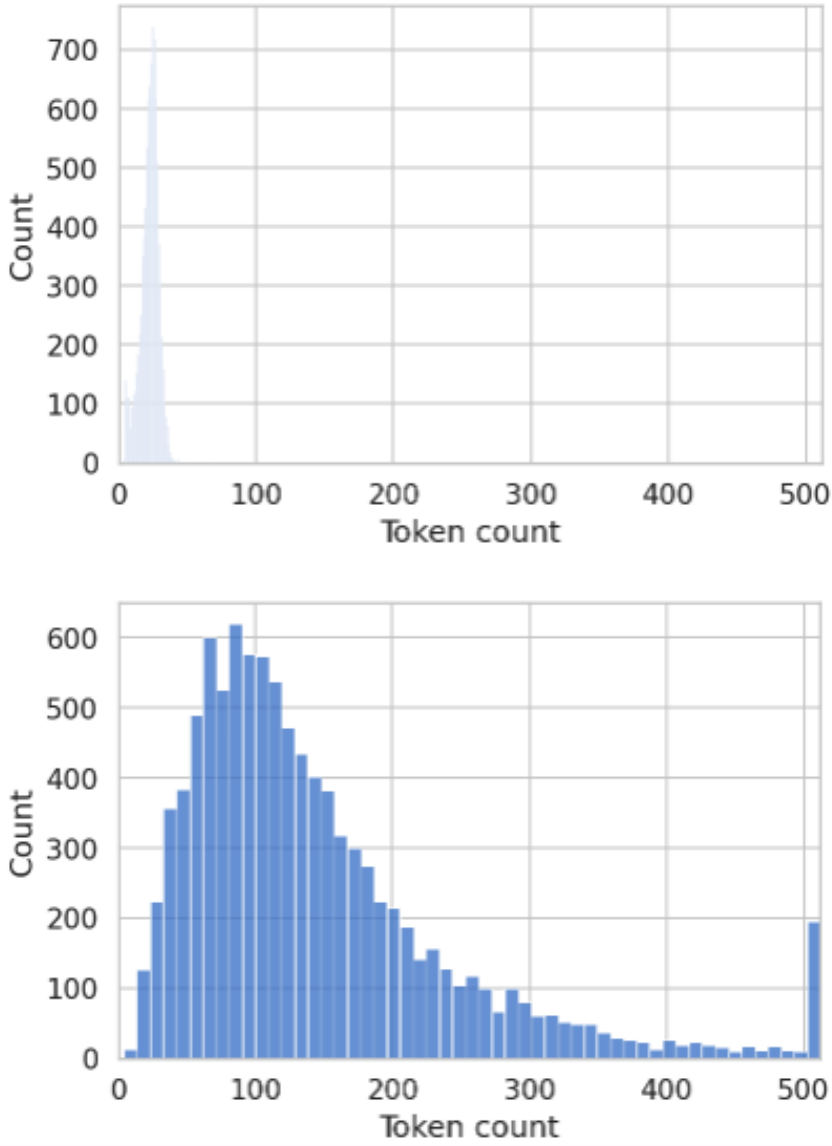


Figure 7 – Token count for each type of description

from Internet Services, Apps and Hardware to Real Estate, Government and Military and Lending and Investment. However, for our case, the category that is more valuable is Sustainability. For our dataset, it is used that if the company fits in the Sustainable Category, then it is considered green.

Once again this faces some bias, since the proper company chooses it. Many companies could try to get better investments or have better PR by admitting themselves as sustainable. However, to confirm or to deny this bias, in further sections, we compare the outcome of our model to the CATA methods. Therefore, we can then reach a conclusion on how well the machine classifies the sustainability and even if this labeling is suitable for such categorization.

Lastly, after building the dataset so far, it is faced with a problem when it comes

to training. In order to train a classification task properly, the machine needs a dataset not biased for any specific label. That is, when one label is more present in the dataset than another, the machine learns that it has more probability of being a specific label then the other. In situations where the outcome is not clear, the machine would then set the outcome as the most probable one. Therefore overfitting to the data.

To contest this problem, it is needed to slice the dataset in a way that the data does not get biased from the amount of samples for each label. The full data set of 550.616 samples, a huge amount of data, contained only 11.747 samples of sustainable companies, as shown in

Label	**Count**
Sustainable	11,747
Non Sustainable	538,869
Total	550,615

Table 3 – Dataset count, grouped by label

To counter this bias produced by the amount of data, it was selected randomly 11.747 samples of the non sustainable part of the dataset. Therefore leaving the dataset with a total count of 23.494 samples, equally divided between sustainable and non sustainable.

2.3 Training

BERT is a pre-trained model (DEVLIN et al., 2018). That means that the model was already trained on a prior dataset. The one used was the BooksCorpus (800M words) (ZHU et al., 2015) and the English Wikipedia (2,500M words). However the objective of the training is to make the model capable of interpreting language, so that it can then be later used and specialized into a specific task. BERTs authors trained the model in two different tasks, auto-completion and masking, as it was deeply explained in the BERT section. With this language understanding base, it is then available for anyone to fine-tune the model and use it as their will.

Fine-tuning is the step of training a pre-trained model for a specific task. That will be further explained in the next section. Not only, it will be also clarified what are Hyperparams and how to use them for fine-tuning a model.

Also, a big part of the training process is the choice of the Optimizer and the Loss Function. Both play an important role in achieving good performance. These two concepts will be also discussed further, in specific sections for each one.

Furthermore, it will be described all the issues that come with training a model. Training a Machine Learning model is the part where the machine properly learns. As

explained in the Backpropagation section, for a classification model, data is imputed to the model, computed and then compared the results with the target for that specific data input. After that the parameters are correct so that the next iteration can run a supposedly better classification. During this training process, many problems can come up. From the time usage, to the overfitting of the data, it will all be justified further.

Lastly, it will be compared all the models trained, varying their hyperparams and evaluating their results. The evaluation will be made through the use of charts and graphs, such as a Confusion Matrix and metrics such as F1-Score, which also will be explained then.

2.3.1 Fine Tuning

The term "fine tuning" refers to the act of perfecting something that was already made. For example, the same term can be used to adjust instruments to a specific tuning. That analogy applies to the machine learning models, which are pre-trained and then require a little more work to be adjusted to perform a specific task.

As (DAI; LE., 2015) explained, fine tuning can be extremely useful when training large models. The clear advantage from pre-training a model is the reduced training time. While training large language models to perform real world tasks would take days, if the training is based on a pre-trained model, it can drastically reduce this time. It then requires a dataset and label targeting the desired task to fine tune it, not a generic dataset, as it is usually used for pre-training.

Not only, another advantage is to overcome small datasets. Not a problem in our case, but it is quite common to have a small dataset when it comes to training classification models. Labeling the dataset can often be expensive, requiring hours of human work to label the data or even financially paying someone to do so. Therefore, fine tuning a model brings great advantages.

Like (BARONE et al., 2017) stated, there are many techniques when it comes to fine tuning a model:

- The first and most common one is truncating the last layer. That is, replacing the last layer and training the model again. That makes use of all the features learned in the pre-training, while changing the output of the model to fit our needs.
- Using smaller learning rates. That can make the machine slowly achieve the optimum through gradient descent for our specific task.
- Freezing the weights of the first few layers. Similar to truncating the last layer, we keep the weights, therefore the features, learned in pre-training. Changing only the outcome of the model.

BERT is a pre-trained language model, available for anyone who would like to fine tune it to a specific language task. To create the classification model of this paper, that seemed like a perfect solution. Since it not only brought with it a great state-of-the-art model that could interpret language, it allowed us to train the model to our needs in a feasible timespan. Also, the first technique of fine tuning was used, by adding a new layer at the end that would learn to use the features given by BERT to classify into green and non-green.

2.3.2 Hyperparams

An important part of building a machine learning model is the hyperparameters optimization (FEURER; HUTTER, 2019). Hyperparams address the model's design, not the weights. That is, it would be the size of a layer, the learning rate, the batch size, and so on. Therefore they cannot be trained, but chosen by the author of the model.

When developing a model, its architecture is often iterated over and over in the hyperparameters optimization process. This process can be automated, allowing the machine to train multiple models and then selecting the best set of hyperparams. However, that tends to be quite expensive in training time, since you are not training one model anymore, but many to be compared (FEURER; HUTTER, 2019). There are many techniques from which to use, from the use of a Random Forest approach.

Like (FEURER; HUTTER, 2019) explained in their paper "Hyperparameter Optimization", this process is extremely costly, but can bring great benefits. Especially in deep learning, where models have hundreds of layers, the combination of different sizes to be trained can reach absurd numbers. But the advantages are numerous, from reducing the human effort in the creation of the model to mainly increasing the models performance.

In our case, Sun, Qiu, Xu and Huang, in their work "How to Fine-Tune BERT for Text Classification?" (SUN et al., 2019) have already traced the path for the optimization of our model. Through many iterations and tests with the parameters, they managed to find the set that can bring the best performance. That drastically helped the development of this model, since the training time and human effort to optimize these parameters could be reduced. However, in this section it will be explained all the hyperparameters, their meaning and how they were chosen based on the work of (SUN et al., 2019).

2.3.2.1 Batch Size

After the wide use of deep ML models, the huge amount of training time became quite an issue. In order to deal with this problem, reducing the training time can be done by training our model in batches (GOODFELLOW; BENGIO; COURVILLE, 2016). What happens is that each batch is processed in sequence, however the elements within the

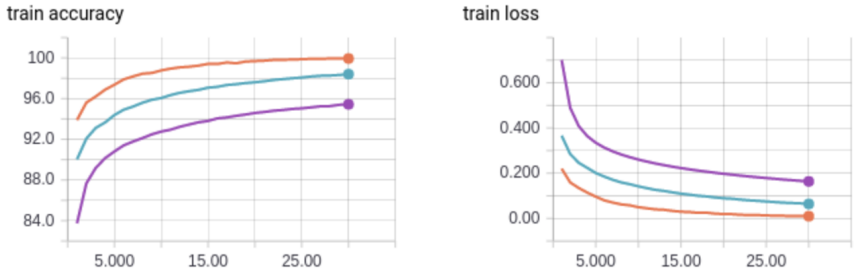


Figure 8 – Batch Size comparison for MNIST dataset (Shen, 2018)

batch can be parallelized during training. However, big batch sizes can make the model overfit the training data, that is, rely too much on the data and not generalize well. Also, the bigger the batch, the more space it is needed to compute it in parallel. Therefore the faster convergence in exchange for GPU space and performance.

The author can choose big batch sizes, to increase the parallelism, however smaller sizes can converge faster (DEVARAKONDA; NAUMOV; GARLAND, 2017). Or even as Devarakonda stated, choose an adaptive batch size that can bring the best of both worlds. This way, the batch size can be increased during training, encouraging the first batches to converge faster and then increasing so it can finish training faster.

On Figure 8, It can be seen the comparison between different batch sizes, orange representing 64, blue 256 and purple 1024. Clearly, the higher the batch size, the worse the performance. Even though it can reach convergence faster.

For our model, (SUN et al., 2019) recommended setting the batch size as 32. As explained before, the larger the number, the worse performance but faster convergence, however it can only reach that convergence because of the parallelism. Since the training of the model was done in a basic 12 GB GPU from Google Colabs, the increase in batch size would not increase the parallelism. The GPU could not afford a 32 batch size, given the not so big amount of RAM.

As (SUN et al., 2019) stated, the ideal would be a size that would fill the RAM of the machine where it is trained, bringing the most out of the machine's specs. Therefore the batch size hyperparam of our model was set to 16.

2.3.2.2 Epochs

The number of epochs is a hyperparameter that defines how many times the whole dataset is going to be fed to the model. Thus, giving the chance of each element in the dataset to influence our model many times.

This is used given the Gradient Descent method may not converge in one iteration of the dataset (RUDER, 2016). This method is the optimization method commonly

used to find the best optimal weights in the Backpropagation algorithm, as explained earlier. Therefore, after iterating the dataset multiple times, it can be found the point of convergence through the Gradient Descent method.

Retraining the algorithm in the same dataset multiple times can bring a smaller error rate, at the cost of higher overfitting. This problem and its solutions will be deeply explained further. However, in order to have the best results of the Epoch iteration, it is advisable to evaluate the model after each Epoch, saving the model that performed best (SUN et al., 2019).

For our model the best performance was already reached with four Epoch, used also by (SUN et al., 2019). In the Overfitting section, the Epoch was set to 10. This was done with a higher number, even though there is an increase in training time, because the learning curve (epoch x accuracy) is more detailed and it brings a better visualization to the model training. In the Overfitting section, those curves will be presented on the explanation of validation.

2.3.2.3 Maximum Length

As explained in the BERT section, the model can only input a maximum of 512 tokens. However, given training time issues, we couldn't afford to train the model using a large Maximum Length hyperparam. As it can be seen on Table 4, the training time grows when increasing the Maximum Length. The information on this table was done based on the approximately 20 thousand sized dataset with ten Epochs.

Maximum Length	Training Time
32	27 minutes
128	1h 25 minutes

Table 4 – Maximum length and time required to train the model

As it can be expected, the bigger the amount of information given to the model, the better it performs the classification but at the cost of training time. However, this will be further explained in the Comparing models section.

It is not the case of our model, but if there was the need of computing text longer than 512 tokens, we would need to apply some techniques to deal with this issue. Among the techniques possible, we have truncating and the hierarchical method (SUN et al., 2019).

The first one consists on removing part of the text to fit the 512 tokens, however there are different parts that could be removed:

- Head-Only - Using the first 512 tokens of the text

- Tail-Only - Using the last 512 tokens of the text
- Head-Tail - Diving empirically the tokens into first part of the text and the last

For the second technique, we divide the text into 512, so it can be fed into the model and aggregate those using three possible operators:

- Maximum - Getting the best of them all
- Mean - Getting the mean of all of them
- Self-Attention - Combining the attention vector of all of them

The result of this text can be seen on Table 5, where (SUN et al., 2019) trained the model with two different datasets (IMDb and Chinese Sogou News) and compared the error rate.

Method	IMDb	Sogou
Head-Only	5.63	2.58
Tail-Only	5.44	3.17
Head-Tail	5.42	2.43
Hierarchical Mean	5.89	2.83
Hierarchical Maximum	5.71	2.47
Hierarchical Self-Attention	5.49	2.65

Table 5 – Test error rates (%) on IMDb and Chinese Sogou News datasets (SUN et al., 2019)

The Head-Tail method is the one with the smallest error rate in both datasets. However, this is dependent on the dataset, possibly a different method might be more advantageous. For our model, after analyzing the dataset, the end of many descriptions were links to social media pages or websites of the company. Therefore, it was decided to follow with the Head-Only approach for this case.

2.3.2.4 Learning Rate

When applying the Gradient Descent optimization method, we introduce this new hyperparam that defines the step size of each iteration (ZEILER, 2012). Therefore, it basically defines how fast the model can learn.

As with the other hyperparameters, there is a trade-off between a high and a low learning rate. If set too low, the model may take too long to converge, causing higher training time, however if set too high, it may never converge, since the model cannot find the optimal spot as it gives too big of a step each time, jumping over the minimal point. As (ZEILER, 2012) said, a better approach can be an adaptive method that would decrease

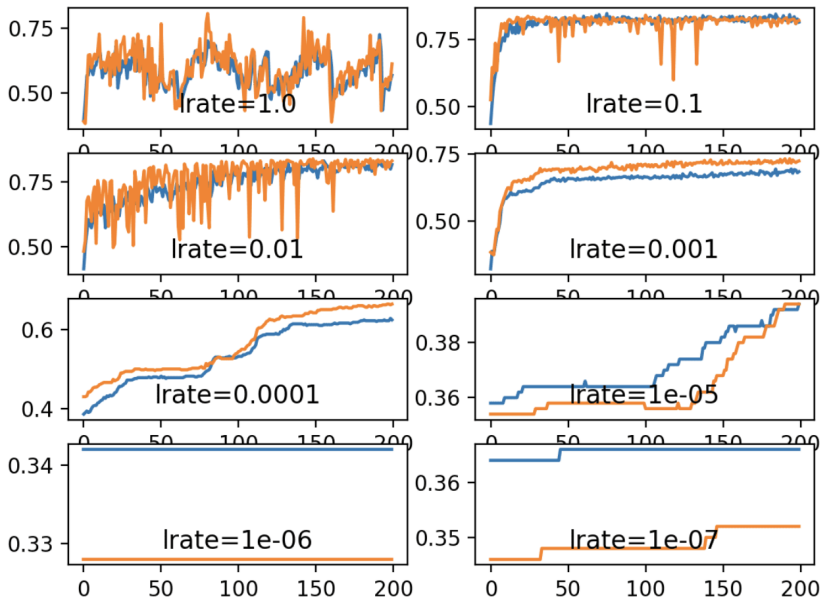


Figure 9 – Learning rate comparison - Plot of test and validation accuracy over epochs of training (Brownlee, 2019)

the learning rate while training happens. This allows the training process to be faster in the beginning and slows down when it needs to be more precise to find the optimal convergence point.

Learning Rates can be defined between zero and one, and in Figure 9 these learning rates and their respective accuracy are compared. In a Blob dataset used in Brownlee (2019) explanation, it is possible to deeply understand the effects of different learning rates.

If it is too low, like the two last tries with learning rate 1e-6 and 1e-7, the model suffers to learn. Through the epochs, the model seems to learn nothing or almost nothing, since it takes such a small step into the optimal direction. However, if the learning rate is set too high, like the first three tries with learning rate 1, 0.1 and 0.01, the training does not seem to converge. The gradient is taking too big of a step to find the minimal point, that is why the accuracy oscillates so much. Lastly, with a learning rate around 0.001, we can see that the oscillation is low and it gradually reaches a plateau, meaning that it has converged to the optimal point.

At (SUN et al., 2019) research, they have found the best learning rate for BERT that avoids Catastrophic Forgetting and converges to the optimal point as fast as possible. Catastrophic Forgetting happens when the weights of a pre-trained model are erased during the process of fine-tuning (MCCLOSKEY; COHEN, 1989). In order to avoid this issue, (SUN et al., 2019) found out that BERT needs lower learning rates, such as 2e-5, in order to converge, as shown in Figure 10. Therefore, for our model, we set that value as

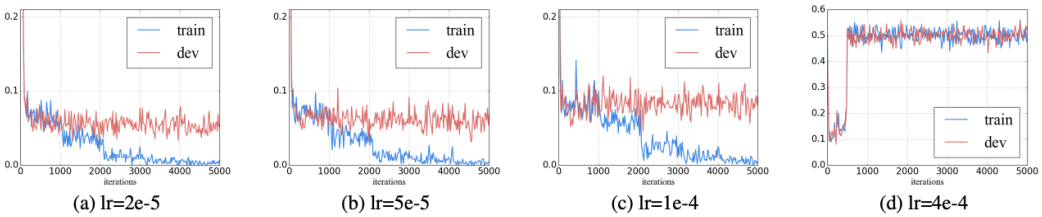


Figure 10 – Learning rate comparisons with BERT (SUN et al., 2019)

our learning rate.

2.3.3 Optimizer

In machine learning, optimizer refers to the algorithm responsible for finding the minima of an error function. As explained earlier, the Gradient Descent method is the most known and used for machine learning purposes, however, there are many algorithms that can apply this method and therefore reach different efficiencies. One of the most used ones and also the one recommended by (SUN et al., 2019) was Adam.

Adam's algorithm was first introduced in the paper "Adam: a method for stochastic optimization" by (KINGMA; BA., 2014). Adam is defined as an efficient method for stochastic optimization that requires only first order gradients with little memory requirement.

It speeds up the gradient descent algorithm by combining two techniques. First, by taking into account the 'exponentially weighted average' of the gradients. The use of averaging causes the algorithm to converge to minima at a faster rate. Second, using Root Mean Square Prop or RMSprop, an adaptive learning algorithm that instead of using the cumulative sum of squared gradients, it uses the 'exponential moving average'.

Mathematically speaking, the equations that compose the Adam algorithm are as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)(\frac{\delta L}{\delta w_t}) \tag{2.4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\frac{\delta L}{\delta w_t})^2 \tag{2.5}$$

Equation 2.2: Adam's algorithm

Once m_t and v_t have been initialized as 0, it is observed that they gain a tendency to be 'biased towards 0', both 1 and 2 $\approx$ 1. This Optimizer corrects this problem by computing m_t and v_t with 'bias correction'. This is also done to control the weights when reaching the global minimum to avoid high oscillations when close to it.

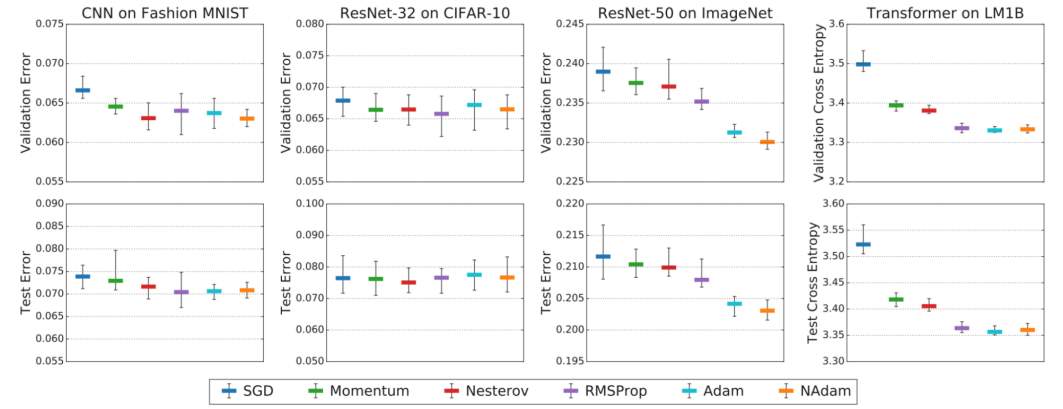


Figure 11 – Error comparison in different datasets and different optimization algorithms (Dami, 2019)

Intuitively, we are adapting to the gradient descent after each iteration so that it remains controlled and unbiased throughout, hence the name Adam. Now, instead of our normal weight parameters mt and v_t, we take the bias-corrected weight parameters m_t and v_t . Putting them together we get the new weight as Equation 2.3 shows:

$$w_{t+1} = w_t - \hat{m}_t(\frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}) \tag{2.6}$$

Equation 2.3: Adam's new weights

In Figure 11, it can be seen that from the algorithms we have available nowadays, Adam is one of the most efficient of them. Dima (2019) compared the algorithms through different datasets, but in most of them Adam gets the best results and if not, very small differences from the best one.

However, as much as Adam has accomplished, there are still upgrades to be done. As (LOSHCHILOV; HUTTER, 2017) explained, a simple alteration in the Adams algorithm can make effective results. Like almost all the optimization algorithms, they tend to use L2 regularization, that is a method to avoid overfitting by having a regularization in the training of each element. Nevertheless, (LOSHCHILOV; HUTTER, 2017) have found that using decoupled weight decay regularization can be more efficient. As this has been widely spread, most of the implementations of Adam in PyTorch, Keras and so on already come with this difference. Therefore, for our model we used the AdamW optimizer (Adam with Weight Decay).

2.3.4 Loss Function

As explained before, ANN basic training procedure consists of minimizing the error of a specific task by running an element through the network and correcting the

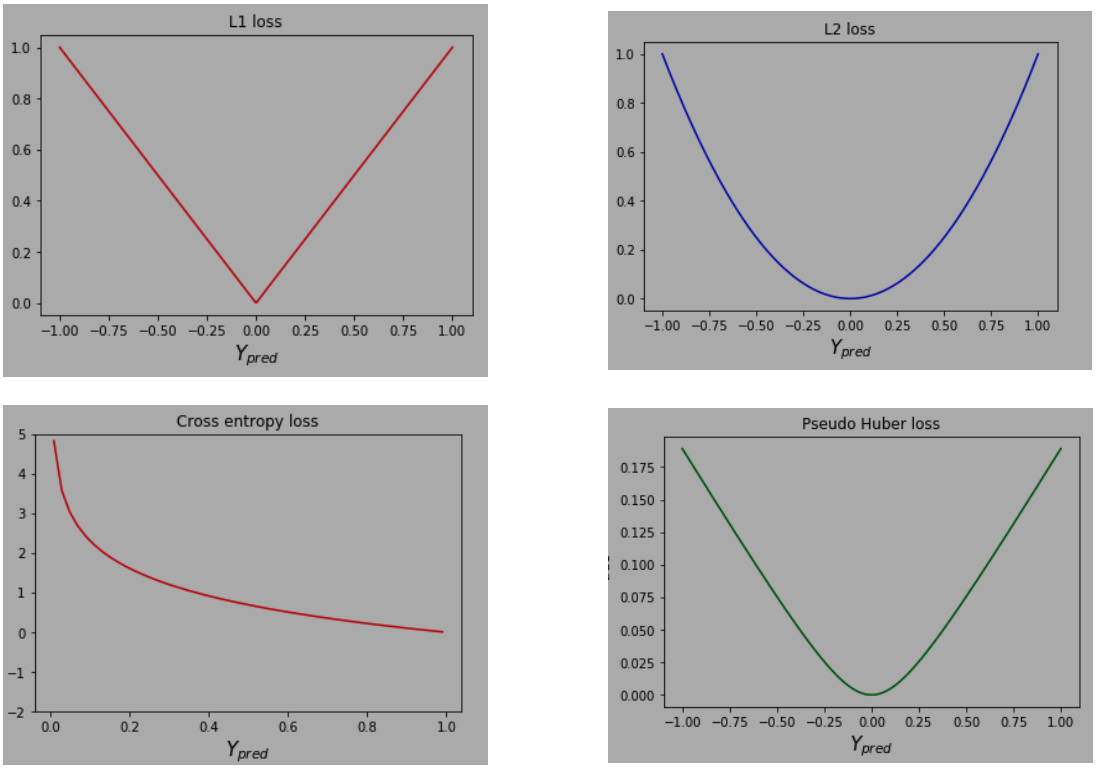


Figure 12 – Cross Entropy Error comparison for both labels (Odegua, 2018)

weights according to the label of that element. This whole process was deeply detailed in previous sections, with the exception of the definition of this error. It is nothing but a function. The gradient in the Gradient Descent method is the gradient of the loss function (GOODFELLOW; BENGIO; COURVILLE, 2016).

There are many Loss Functions that have their advantages and disadvantages depending on the output scenario they want to model. Figure 12 shows the most famous loss functions and their plot, being y the true value and $\hat{y}$ the predicted value. L_1 being the most basic of them and L_2 the squared version, which makes the function continuous and therefore derivative, allowing Gradient Descent. Pseudo-Huber loss is the combination of both L_1 and L_2 which has a parameter that changes the slope of the function. Finally the Cross Entropy loss function, also referred to as the logistic loss function.

$$L_1 = (y - \hat{y}) \tag{2.7}$$

$$L_2 = (y - \hat{y})^2 \tag{2.8}$$

$$\text{Cross Entropy} = -\sum_i p_i \log q_i = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \tag{2.9}$$

$$\text{Pseudo Huber } L(a) = \delta^2(\sqrt{1 + (\frac{a}{\delta})^2} - 1) \tag{2.10}$$

As Goodfellow (2016) described, the loss function is deeply connected to the output function. As we are doing a classification with Softmax activation function and also with

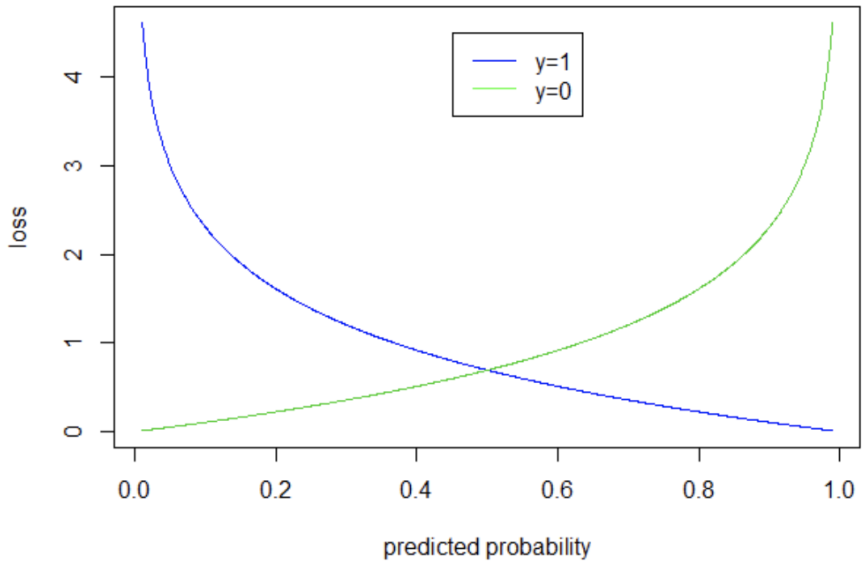


Figure 13 – Cross Entropy Error comparison for both labels

Gradient Descent optimization, the best option as defined also by Sun et al (2020) is the Cross Entropy function. The idea of the loss function is to measure the difference to the true value based on the output, since softmax will output a number from zero to one that represents the probability of that element being of a certain class. As Figure 13 shows, if the number is close to one, when the label should be one, then the loss is almost zero, but the opposite happens if the label should be zero, then the loss is huge. That is ideal to train our model, since it penalizes a lot for wrong predictions and almost nothing for correct ones. Not only, the gradient for the minima is pretty defined, which helps the model not only to learn, but to learn fast.

2.3.5 Training Issues

During training, many issues can come up. In this section let's have a look at the issues faced in the training of our model and how it was fixed.

2.3.5.1 Vanishing and Exploding Gradients

As seen before, the algorithm used for adapting the weights during training is called Backpropagation. In some deep neural networks where there are many layers, the gradient tends to decrease as we move backwards through the hidden layers. This means that neurons in the earlier layers learn much more slowly than neurons in the later layers. The phenomenon is known as The Vanishing Gradient Problem (Goodfellow, 2016). This is a very common problem, and is even more evident in Recurrent Neural Networks, used in Natural Language Processing applications.

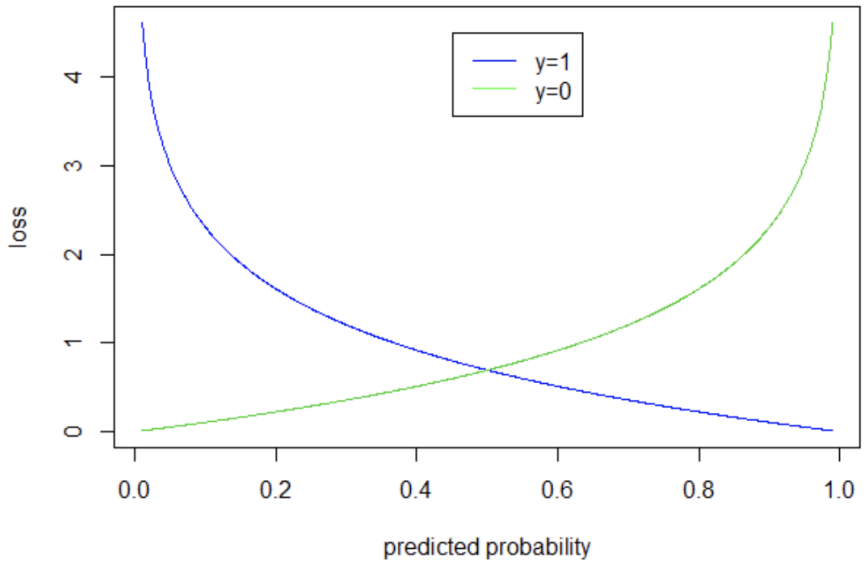


Figure 14 – ReLU plot and equation (Goodfellow, 2016)

On the other hand, sometimes the opposite can happen and the gradient gets much larger in the previous layers. This problem is called The Exploding Gradient Problem (Goodfellow, 2016), and it can be as problematic as the Vanishing Gradient Problem. Generally, the gradient in deep neural networks is found to be unstable, tending to explode or disappear in the earlier layers. This instability is a fundamental problem for gradient-based learning in deep neural networks.

This problem happens mostly because of the activation function. When the gradient is passed through the function, small gradients can become even smaller and big ones even bigger. On a small network that wouldn't be a problem, but having more robust networks, this problem is accentuated.

Each of the problems have their solutions. For The Vanishing Gradient Problem, what can be done is changing the activation function for one that does not propagate the small gradients. The ReLU activation function does that. In Figure 2.12 it can be seen the plot of the ReLU function, whose derivative would be 0 or 1, then preventing gradients from vanishing.

On the other hand, Exploding Gradients cannot be solved with that technique. Having big gradients can lead to unstable learning (Goodfellow, 2016) but what they mean is that there is a big change to be made on that weight. In the case of this model, we had to face The Exploding Gradient problem and a simple but effective way of fixing it is by simply clipping the gradients. Setting a threshold to the gradient would not affect the network decision when adapting that weight and it would allow the gradient to be safely passed through the early layers.

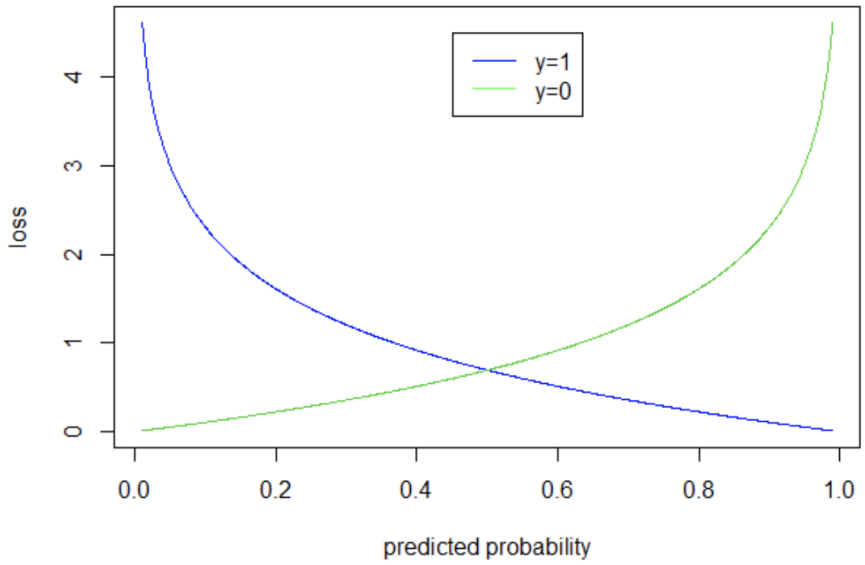


Figure 15 – Underfitting and Overfitting comparison (IBM Cloud Education, 2021)

2.3.5.2 Overfitting

Overfitting happens when the model fits so well the data received as input, that it describes an unrealistic reality. If a model agrees with the available data, this does not make it a good model. It can just mean that there is enough freedom in the model that it can describe almost any data set of a given size, without capturing any genuine insight into the phenomenon in question. When this happens, the model will work well for existing data, but will fail to generalize to new situations. The real test of a model is its ability to make predictions in situations that have not been exposed to it before.

As Figure 15 shows, when the model overfits the data, it will perform incredibly for that specific dataset, but it will fail to represent different datasets. However, when it underfits, it also does not represent reality correctly. The ideal would be an optimum in the middle of these two situations.

Underfitting usually happens when there is lack of training, the more it trains, the more it fits the data. Therefore there isn't any technical solution for that problem, but for overfitting it does. There are many solutions to overfitting, such as adding a Dropout layer or comparing the accuracy with a validation dataset.

The dropout technique consists of disabling random nodes during batch training (Srivastava, 2014). As Figure 16 shows, it removes a certain amount of nodes making the network lose some of the capacities it had before, obliging it to generalize and therefore to avoid overfitting. So that this process does not become harmful to the accuracy of the model, the dropout nodes are randomly chosen every batch. As a disadvantage of this technique, the model then requires almost double the training time to converge (Srivastava,

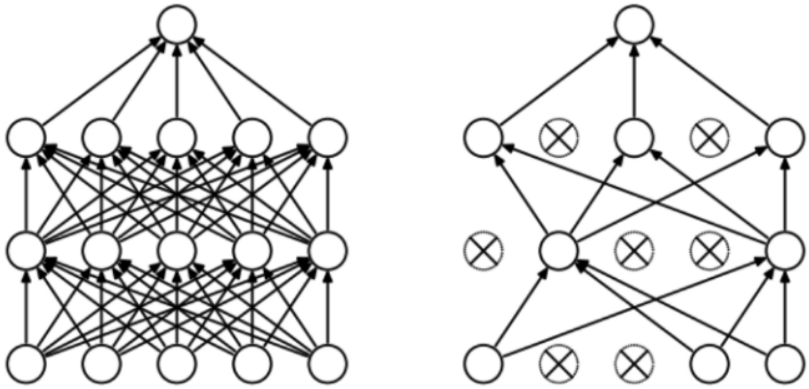


Figure 16 – Representation of the Dropout technique (Srivastava, 2014)

2014).

When training the data, it is expected that the model will overfit it at some point, however, there are ways to evaluate how well the model is doing in generalizing the data. It is common that before training the dataset would be split into three parts. It is usually divided into train, validation and test, usually with a 90%, 5% and 5% ratio, respectively. In this case, 80% of the dataset would train the model and the validation dataset would check how well the model is doing in generic data, since it has not yet seen that input. Afterwards, the test data is used for properly calculating the accuracy.

With a validation dataset, training can be better evaluated. The validation accuracy during training is the one that should be taken into account, as it can be seen in Figure 17. The chart clearly shows the overfitting issue. For each Epoch the training accuracy grows as the model overfits the data, however the validation oscillates around the same value, showing that our model is not overfitting the data, given the Dropout technique explained before.

2.3.5.3 Time Issues

When training a machine learning model, it usually requires a huge amount of computational power. That is why companies like Google offer services of Virtual Machines so that users can train their models in the powerful computers that the company offers (Google Cloud, 2022). When training a pre-trained model, as it was discussed before, it can make machine learning much more accessible, however it still has its cost.

The model proposed in this paper was trained without funds and therefore ran on an average and ordinary laptop. Therefore many constraints could have been discarded given some funding. However, as constraints bring creativity, a way of surrounding these problems was using Google Colab.

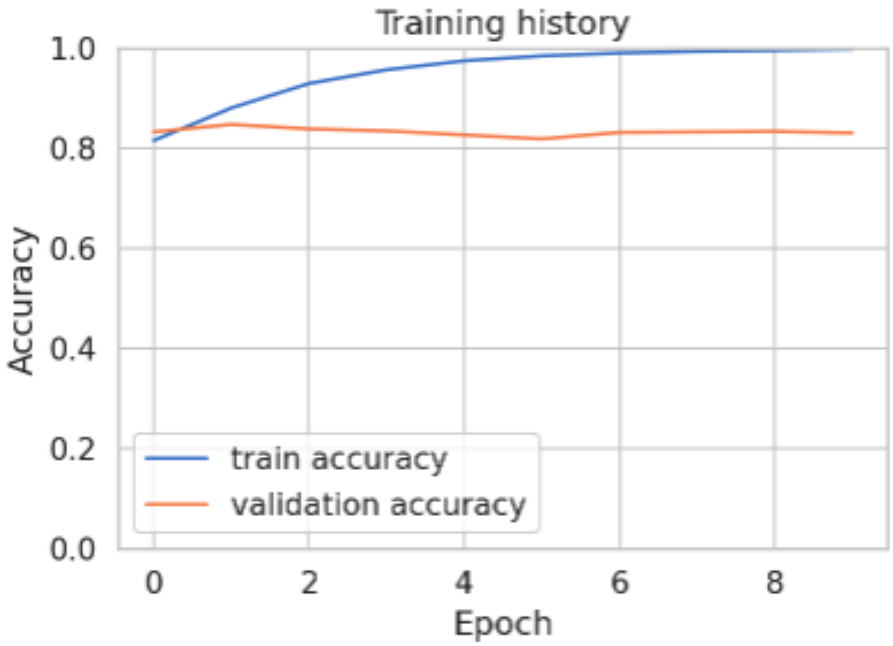


Figure 17 – Train and validation accuracy

Google Colab is a notebook application by Google that allows users to code a small section of code and run it separately from the rest. Widely used in the Data Science community, this tool also offers a certain amount of GPU time from the Google servers. As a way of boosting the training efficiency, this feature allows users to use a virtual machine from Google that is usually much more powerful than the average laptop.

This tool drastically improved the training time for the model. It was possible then to achieve great accuracy by setting the hyper params in a way that would increase the training time reasonably but also increase accuracy. This could have been even better if there was more computational power available. However, in a real-world scenario, this was a good solution to fix the problem with training time.

2.3.6 Comparing models

In this section models will be compared based on different hyperparameters and specs. In order to evaluate these models, the metrics Precision, Recall and Accuracy will be used. Precision stands for the sum of true positives divided by true positives and false positives, it shows how well the positive cases are classified from all classified positives. Recall is basically the opposite, the sum of all true positives divided by true positives and false negatives, showing how well the positives are classified from all actual positives. Accuracy, on the other hand, simply measures all correctly identified cases. For better understanding of each metric, Figure 2.16 shows each of them graphically.

These metrics are essential when evaluating a classification model, since it gives

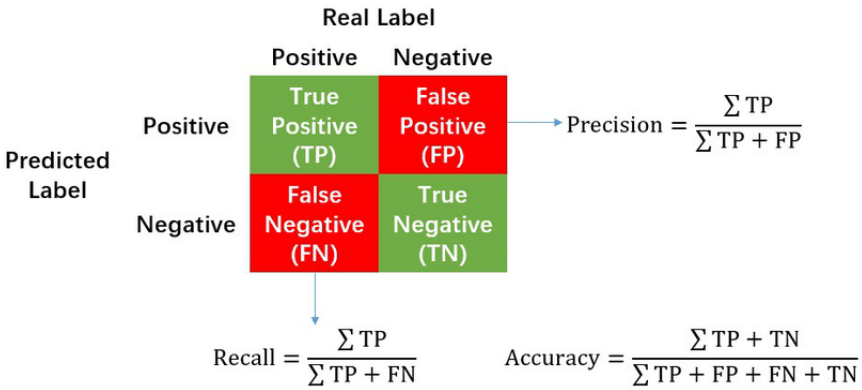


Figure 18 – Visualization of Confusion Matrix, Precision, Recall and Accuracy (Jun et al, 2019)

more insight into each small characteristic of the model. With these, it is possible to understand where the model fails and where it thrives.

Not only, to help visualize these, the Confusion Matrix will be used. This visualization shows the amount for each case, true positive, false positive, true negative and false negative. As it can be seen also in Figure 18, this is a good way to visualize all the metrics explained before, making it easier to comprehend them.

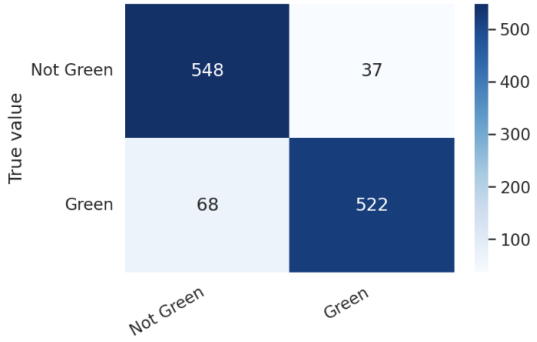
Not only, another important measure to keep track of is the Support. This metric states how many elements were tested. This is important to determine if any problems from the model could have been caused by the lack of elements in the dataset. As usually, the more the better, however it also increases training time. So ideally a middle ground would be found where the machine has enough input to generalize the task without spending too much time during training.

In the next comparisons, the Support value is always at 1175 in total. This happens since the total dataset of 23.494 is splitted into three parts, as explained before. The dataset used to evaluate the model is the test dataset, after the training and the validation.

2.3.6.1 Maximum Length

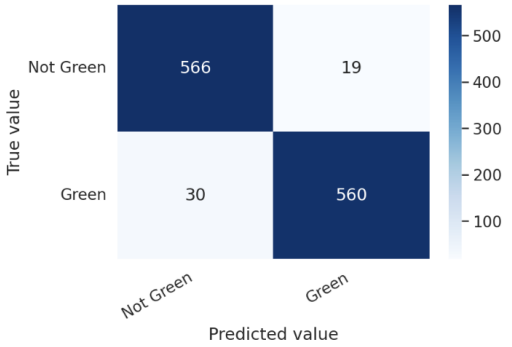
As explained in the Maximum Length section and represented in Table 4, we can see that having a bigger maximum length when training our model brings a bigger complexity to the model. The training time roses but so does the amount of information given to the model. To compare which Maximum Length would fit our model better, we compared two versions, one with 32 and another with 128. More than this would bring more issues with the training time. The models trained in this section used as hyperparams: Batch Size 16, 10 Epochs and 2e-5 for Learning Rate.

In Figure 19 it shows the classification report and the confusion matrix for a



	precision	recall	f1-score	support
0	0.89	0.94	0.91	585
1	0.93	0.88	0.91	590
accuracy			0.91	1175
macro avg	0.91	0.91	0.91	1175
weighted avg	0.91	0.91	0.91	1175

Figure 19 – Classification Report and Confusion Matrix for Maximum Length 32



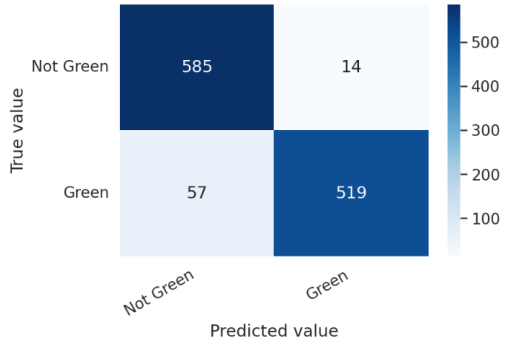
	precision	recall	f1-score	support
0	0.95	0.97	0.96	585
1	0.97	0.95	0.96	590
accuracy			0.96	1175
macro avg	0.96	0.96	0.96	1175
weighted avg	0.96	0.96	0.96	1175

Figure 20 – Classification Report and Confusion Matrix for Maximum Length 128

maximum length of 32. As it can be seen, the precision of “Not Green” is slightly behind the “Green” one and the opposite happens for recall. This means that the model is slightly skewed into saying a company is not sustainable when it is. It has a bigger number of false negatives. It shows that this model had difficulties defining what is green and in case of doubt, skewed into more “Not Green” than the opposite.

The accuracy for 32 is 91%, which already is a considerable result, we decided to try also with a maximum length of 128, which is shown in Figure 20. The version with 128 clearly performs better with an accuracy of 96%. However, it faces the same issue as the smaller version with the false negatives. Even though it is less than before, since now we have a precision for our negatives of 95% as before of 88

Both models end up facing the same issue, even though the bigger one clearly



	precision	recall	f1-score	support
0	0.91	0.98	0.94	599
1	0.97	0.90	0.94	576
accuracy			0.94	1175
macro avg	0.94	0.94	0.94	1175
weighted avg	0.94	0.94	0.94	1175

Figure 21 – Classification Report and Confusion Matrix for smaller description

performs better than the other. It also proves what was expected, that given more information to the model makes it perform better.

2.3.6.2 Short and Long Description

As explained in the Dataset section, two descriptions were possible when creating the dataset. A shorter description, with maximum length of 67, and a longer description, with maximum length of over 512. In order to determine which one was better for identifying sustainable ventures, we trained a model with the short description. Comparing it with the previous model with 128 maximum length, we can see that they still face the same problem with the false negatives. It also provides a slightly worse accuracy, given also by the smaller length.

The shorter description follows as expected, performs slightly worse than using the big description with a big maximum length. Following the same problems seen before, we could prove that the model with bigger descriptions is more advantageous.

3 Online Interface and Data Visualization

In this chapter, we will enhance our machine learning efforts by extending them to include user-facing applications, with the goal of increasing accessibility to the model for the general public. This expansion comprises two key components: an online interface that enables users to directly engage with the model, and a data visualization view that harnesses our extensive training dataset. This view not only offers valuable context to users but also provides essential storytelling tools for guiding future work.

3.1 Online Interface

The landscape of machine learning has evolved, shifting from a primarily theoretical focus to a more practical application, as seen with recent developments like ChatGPT. To enhance the comprehensiveness of our work and align with this contemporary trend, we have opted to implement a web-based interface, providing users with convenient access to our model.

Our approach involves integrating the model into a web server, leveraging well-established web protocols for communication with the client side. The client interface will enable users to input descriptions and determine whether the model identifies them as matching with environmentally friendly ventures. This integration aims to bridge the gap between theoretical advancements and practical utility, ensuring our work remains relevant in the evolving landscape of machine learning applications.

3.1.1 Choice of Technologies

To implement our online interface, we faced the task of selecting the appropriate technologies for our web-based product. With the advancements in modern web development, a plethora of technologies is available for facilitating client-server interactions. On the client side, we chose JavaScript with the React framework, leveraging its well-established status and abundant online resources. For the backend, we opted for Python, aligning with the language used in developing the model. To streamline the creation of a web server, we employed the Flask micro-framework.

Figure 3.1.1 delineates the high-level architecture of our system. It illustrates the client-side system, written in JavaScript with the React framework, comprising two essential modules: the user interface, detailed in the section, and the visualization module, expounded upon in Section 3.2. Additionally, Figure 3.1.1 portrays the server-side system, encompassing a Python-based web server using Flask, and the machine learning model,

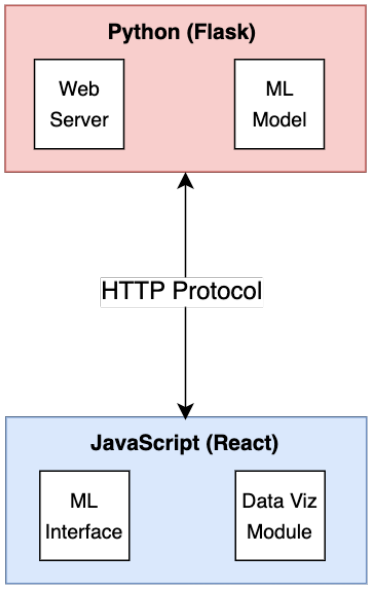


Figure 22 – High level architecture of our web based system

crafted with the PyTorch machine learning library. This cohesive selection of technologies forms the backbone of our system, ensuring seamless integration and functionality.

3.1.2 Integration with the Machine Learning Model

This subsection aims to elucidate the integration of a machine learning model into a web-based application, building upon the model developed in the previous chapter that discerns whether a given description corresponds to a green venture. With the model trained, we utilized PyTorch functionality to save its current state, serializing all parameters. This enables us to load the model seamlessly in any other environment using PyTorch.

Having the model in the desired state on our web server, the loading and initialization process occurs during the web server's initialization. Figure 3.1.2 illustrates the execution pipeline when a request reaches the web server. Let's delve into the details:

1. **Request Handling:** The initial step involves a HTTP POST request to the `/predict` endpoint. The request's body contains a `description` field with the string value representing the provided description.

2. **Preprocessing:** Subsequently, we preprocess the description string to transform it into a `data-loader`, a `Data Type` serving as the model's input.

3. **Model Inference:** The `data-loader` is then fed into our model, yielding an integer (0 or 1) indicating a non-green or green venture. We post-process this data to convert it into a boolean.

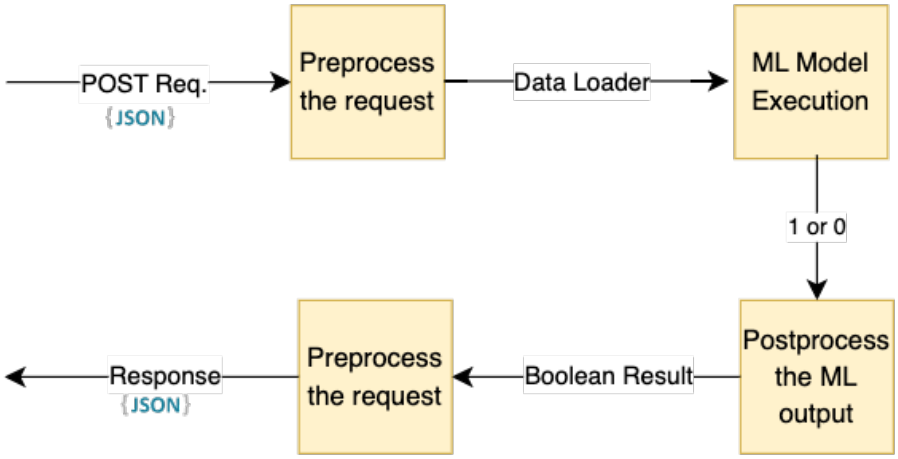


Figure 23 – A pipeline diagram representing a POST request to our back-end Server

4. **Response Generation:** The final step is the client response. If all stages execute successfully, a 200 code is returned with the result in the response body under the `is_green` field. Any error in the pipeline prompts a 500 code, indicating an error. This systematic approach ensures a robust and effective execution of the model in the web-based application.

For the client side, this segment of the application is relatively straightforward. A form featuring an input field allows users to enter a venture description. Upon selecting a button, the content of this input field is then transmitted via a POST request, serving as the entry point for Figure 3.1.2. Additionally, we ensure that the input field is not left empty.

Recognizing that generating a venture description is not the simplest task, we have enhanced the interface by incorporating four buttons. Three of these buttons provide predefined values for sample descriptions, allowing users to test the system without the need to formulate a description. The fourth button selects randomly from a pool of 40 examples within the application, enabling users to further test with a variety of randomized values. This user-friendly approach aims to facilitate testing while ensuring a seamless experience for individuals interacting with the application.

3.2 Data Visualization

To enhance our work's completeness, we've decided to further leverage the dataset used to train our machine learning model. As outlined in Section X, the dataset is rich in various features, but our model focused solely on sustainability and description. To explore more aspects of the dataset, we've created three data visualization charts. Analyzing these charts allows users to better understand the model's training dataset and gain insights

into potential avenues for further work.

This section describes the three visualizations we introduced in this work. We explain the reasoning behind choosing these visualizations, detail the pre-processing steps for creating them, and provide images of the visualizations. The results will be analyzed in Chapter Y, alongside the outcomes of the machine learning model.

3.2.1 Choropleth Map

The initial chart introduced is a choropleth visualization. This graphical representation delineates a world map, employing varying shades of blue to signify the prevalence of sustainable companies in each country. The intensity of the hue correlates with the density of such enterprises. This chart serves as a valuable tool for discerning the global distribution of sustainable companies, offering researchers and academicians a visual insight into the geographic landscape of environmentally conscious businesses. Moreover, this visualization not only facilitates an understanding of regional concentrations but also presents an opportunity for scholarly exploration of the socio-economic factors influencing the proliferation of sustainable practices in different parts of the world.

3.2.1.1 Pre Processing Data

The pre-processing phase for this chart was straightforward. The initial step involved creating a new column in the dataset to incorporate a Boolean variable representing the 'is_sustainable' feature, streamlining subsequent steps. Within the dataset, there exists a column named 'country' containing country codes. The process entailed counting all entries with the same country name and marked as sustainable. This procedure generated a JSON file that will be interpreted by our client-side application. The snippet of this JSON file is presented in Listing 3.1.

Listing 3.1 – Snippet of the generated JSON file

```
{
  "country": "BR",
  "count": 25,
  ...
}
```

3.2.1.2 Visualization

As described above, the chart is very straightforward. Figure 3.2.1.2 illustrates the chart, where countries with a darker green color represent a higher number of sustainable companies, while lighter-colored countries have fewer such companies. The color scale ranges from 0 to B, with 0 denoting the least and B indicating the highest number. It

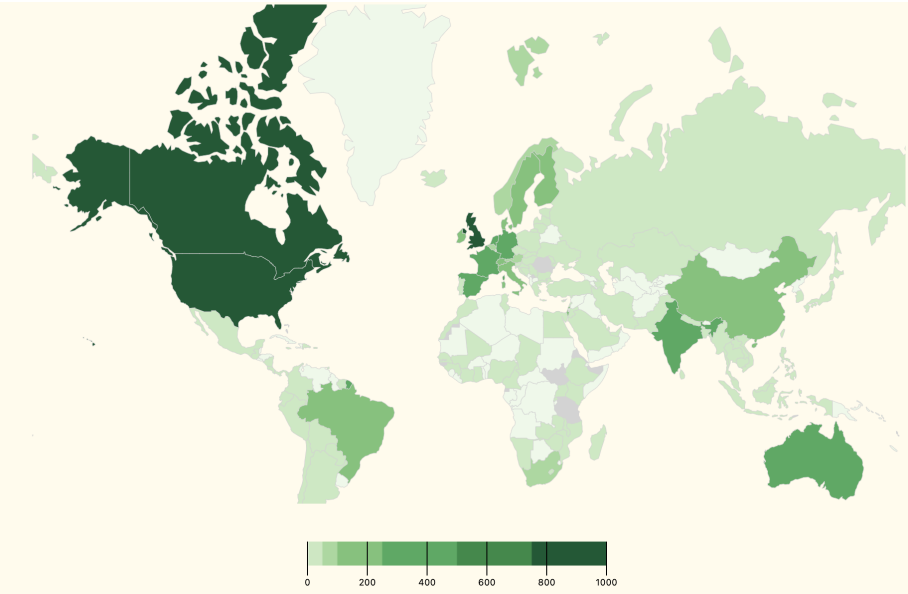


Figure 24 – Our Choropleth Map

is crucial to note that data for some countries is unavailable, and these countries are highlighted with a gray color.

3.2.2 Lollipop Chart

Initially, we considered creating a word cloud with the description words. However, as outlined by (RIVADENEIRA et al., 2007), word clouds may not be the most accurate visualization, often leading to confusion and misinterpretation. Instead, we opted for a lollipop chart to illustrate the distribution of words across all the descriptions in our dataset.

This visualization allows users to discern nuances within the companies present in the dataset. We excluded common English stopwords during the preprocessing step, ensuring that only relevant words remained. This refined approach provides a clearer understanding of the types of companies described.

3.2.2.1 Pre Processing Data

The preprocessing step for this chart was also straightforward. Initially, we systematically went through all the descriptions and selected words that were not stopwords, utilizing the NLTK (Natural Language Toolkit) package for assistance. Subsequently, we added each word to a Python dictionary and tallied the frequency of each encounter. A snippet of the generated JSON data is presented in Listing 3.2.

Listing 3.2 – Snippet of the generated JSON data

```
{
```

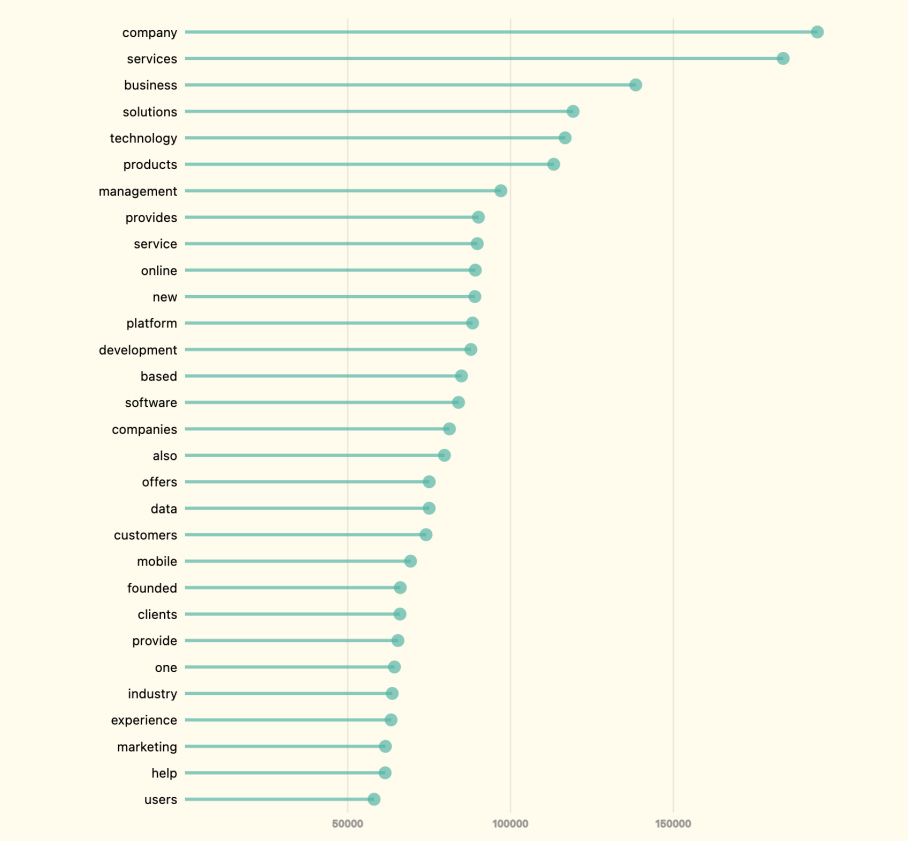


Figure 25 – Our Lollipop Chart

```
  "text": "example",
  "count": 15,
  ...
}
```

3.2.2.2 Visualization

We chose to employ a horizontal lollipop chart, as it facilitates the legibility of each label's name. Additionally, we made the decision to showcase only the top 30 ranking words. Figure 3.2.2.2 illustrates the conclusive outcome of this visualization.

3.2.3 Circular Bar Plot

Our final visualization takes the form of a circular bar chart. This chart serves the purpose of elucidating the categories in which the companies within the dataset are described, along with exploring the correlation between these categories and sustainability. Through an examination of this chart, users can gain additional insights into the sectors from which the companies originate and understand the sustainability aspects associated with these sectors.

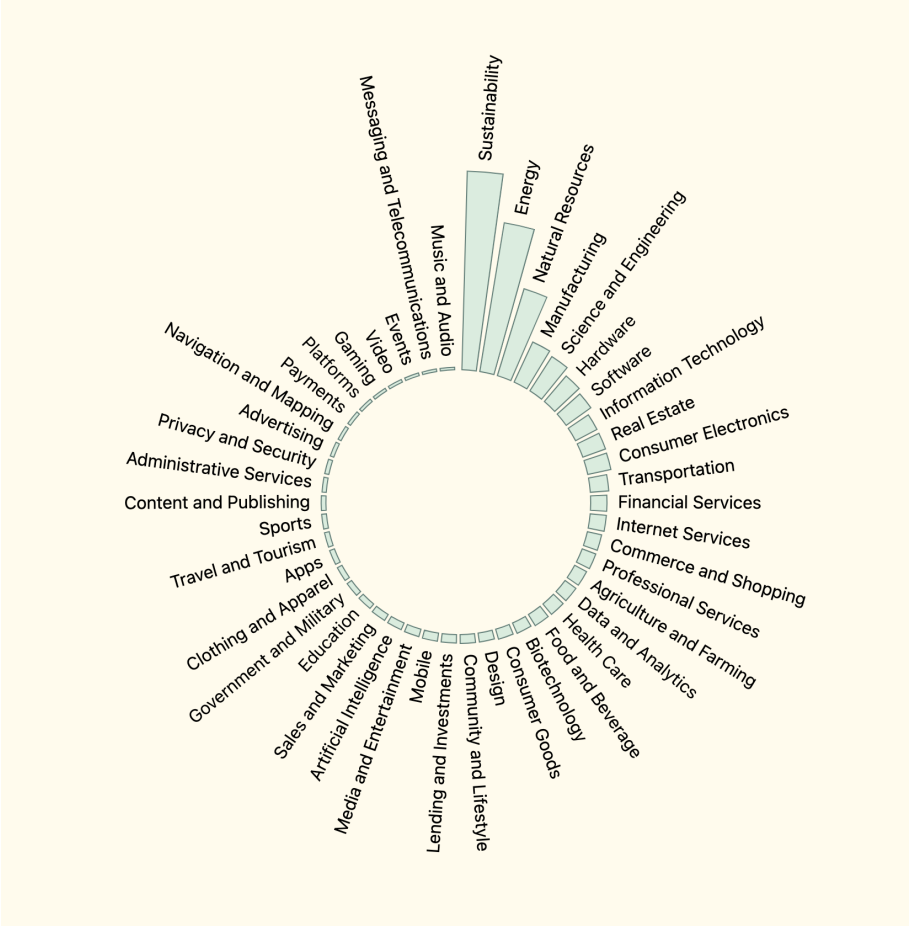


Figure 26 – Our Circular Bar Chart

3.2.3.1 Pre Processing Data

In the preprocessing phase for generating the circular, we focused on the 'category_groups_list' column, which was an array containing various categories associated with each company. To effectively analyze and visualize this data, we needed to expand this column. Utilizing the explode function, we separated each category into individual entries. Subsequently, we created a JSON file to capture the count of sustainable and non-sustainable companies for each category. Each category resulted in two entries in the JSON file, one representing the count of sustainable companies and the other representing the count of non-sustainable companies.

Listing 3.3 – Snippet of the generated JSON data for the circular chart

```
{
  "category": "Technology",
  "count": 45
},
{
  "category": "Software",
```

```
  "count": 180
},
...
```


3.2.3.2 Visualization

The circular barplot offers a clear visual representation of sustainability within different sectors in our dataset. Each sector is represented by a bar, showcasing the count of sustainable companies. The absence of a counterpart bar for non-sustainable entities emphasizes the exclusive focus on sustainable companies in this visualization. This deliberate simplicity ensures a straightforward and unambiguous portrayal of the distribution of sustainable companies across various industry sectors.

4 Final Considerations

4.1 Conclusions

In this chapter, we will delve into the accomplishments of our project. Our objective centered on crafting a machine learning model capable of predicting the environmental friendliness of a business based on its description. Simultaneously, we engineered a user-friendly website facilitating interaction with the model, incorporating visuals such as charts to enhance comprehension of the data. The outcomes will be segmented into three distinct parts of our work, with a primary emphasis on elements fully developed by our team.

4.1.1 Machine Learning Model

The result of the model must be carefully analyzed, since the machine is being taught on how to copy the label proposed. However, this label can be biased, since companies might define themselves as Sustainable to attract green focused employees and customers. Not only, it can also be used as a way of getting green venture funding. As L. Cekanavicius et al stated, green ventures can have more access to funds. This label problem brings much more complexity to the task of classifying green ventures.

In the end, the best model of all the comparisons used the following hyperparameters; Batch Size 16, 10 Epochs and 2e-5 for Learning Rate, 128 Max Length and with the bigger description. The final model reached 96% accuracy, therefore it is extremely successful in reaching the label we defined in the Dataset section.

In order to showcase some results, Figure 27 shows the description, the machine's outcome and the label as a source of truth for a case where the machine got the right "Green" label. The company works in clean energy, so it is clear that it would fit a Sustainable category. As a non biased selection, the machine then performs as expected and can reach the result wanted.

Figure 28 on the other hand shows the case where the machine gets the right "Not Green" label. This company does not state any green related processes in their description and indeed does not set themselves as Sustainable, therefore the machine also performs as expected setting the label as "Not Green".

In Figure 29 we can see an example of the machine failing to classify the venture. In this case, the source of truth is set as "Not Green" while the machine says otherwise. We can see that the model is drifted towards ecological ventures since the description discusses a lot about climate. However, by reading the description it clearly indicates that

```
Description:
 clean energy business network is a nonprofit
organization aimed at advancing the small business
voice of the clean energy economy the organization
is working to grow the clean energy economy
through policy public education and business
support for small and mediumsize companies it has
over 3000 across the us in the areas of the clean
energy economy including renewable energy energy
efficiency natural gas and other advanced energy
and transportation technologies cebn was started
in 2009 by the pew charitable trusts and is now an
independent initiative of the business council for
sustainable energy a coalition of companies and
trade associations from the energy efficiency
natural gas and renewable energy sectors

     True Value: Green
Machine Outcome: Green
```

Figure 27 – Description, true value and correct model outcome for green label.

```
Description:
 centennial ventures is a venture capital firm
investing in network companies and related
enabling software and technology enterprises with
the potential to be market leaders their
investment focus is on early and laterstage
opportunities centennial ventures vii a 341
million partnership is their most recent fund
being invested

     True Value: Not Green
Machine Outcome: Not Green
```

Figure 28 – Description, false value and correct model outcome for green label.

```
Description:
 world sports news is a 24 hours sports news
channel with information update all second this
website started october 2008 the site summarizes
all sports news in one place it brings together
the main news from espn bbc new york times goal
fox sports usa today among other sources

     True Value: Green
Machine Outcome: Not Green
```

Figure 29 – Description, true value and incorrect model outcome for not green label.

```
Description:
 the climate corporation formerly weatherbill aims
to help farmers around the world protect and
improve their farming operations with uniquely
powerful software hardware and insurance products
the companyâ€  s proprietary climate technology
platform combines hyperlocal weather monitoring
agronomic modeling and highresolution weather
simulations to deliver climate basic and climate
pro mobile saas solutions that helps farmers
improve profitability by making better informed
operating and financing decisions

     True Value: Not Green
Machine Outcome: Green
```

Figure 30 – Description, false value and incorrect model outcome for green label.

the company tries to improve profitability and it is not connected with environmentally friendly practices.

Figure 30 now shows a case where the machine also failed to classify the venture but with the source of truth as "Green". This is a clear case where the text description does not state anything related to sustainability, but the label is set as Sustainable. The company is a sports news channel that could have set themselves as environmentally friendly to receive higher funds or to attract more customers in a public relationship marketing play.

An interesting outcome of this project is that the machine could be used to find out false eco-labelling companies. In cases where the machine states as "Not Green" and the company states the opposite, it could be worth investigating.

4.1.2 Online Interface

The online interface served as the link between our model and users (RAHAL; DIAS, b; RAHAL; DIAS, a). As explained in Section 3.1, we designed the interface as a web application. In this app, users can input any description and get feedback on whether it matches a green venture.

Though straightforward, the interface can be seen as a success. The model functions just like in the command line; we tested it with every input in Section 4.1.1, and the response was consistent. The interface also features a user-friendly UI, making it easy for users. Refer to Figure 31 for a screenshot of the interface.

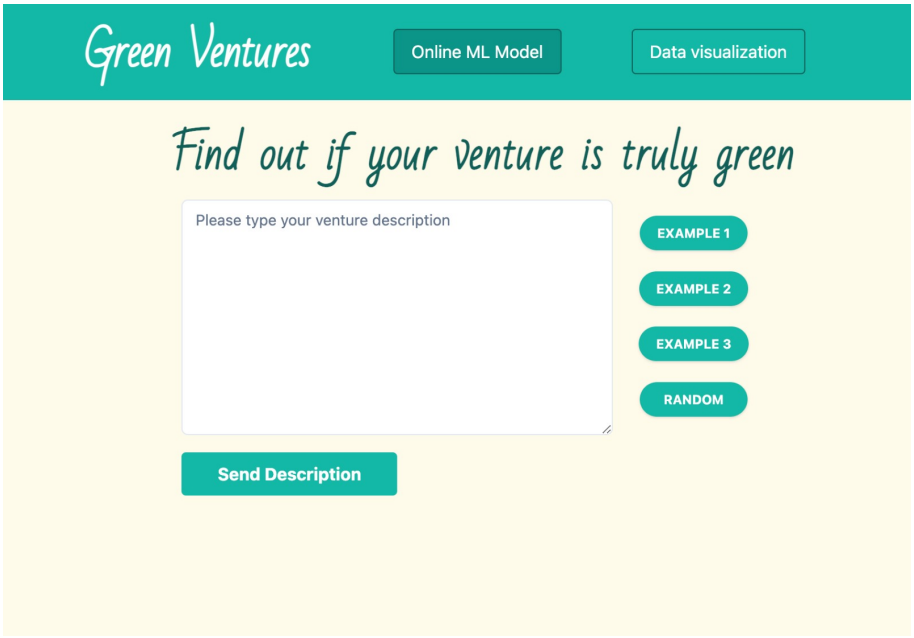


Figure 31 – A screenshot of the online interface for the Machine Learning Model

4.1.3 Data Visualization

The data visualization part of this task aimed to get more insights into the training data for our Machine Learning model (RAHAL; DIAS, b). We used three visualizations to analyze features of the dataset. In this section, we'll explain what kind of insights can be seen in each of the visualizations.

4.1.3.1 Choropleth Map

Examining the global distribution of sustainable ventures through a choropleth map offers valuable insights into the dynamics of environmental entrepreneurship. The dominance of green companies in the USA and Europe, comprising a substantial 74% of the total sustainable ventures worldwide, is driven by a confluence of factors. Robust environmental policies, advanced technological infrastructure, and a heightened awareness of sustainability contribute to this significant presence. Meanwhile, the emergence of green enterprises in Latin America, China, and India, collectively contributing 21%, reflects a promising global shift toward eco-friendly practices, underlining a shared commitment to environmental responsibility.

However, the noticeable scarcity of green ventures in Africa, representing only 5% of the total, demands a nuanced approach. While acknowledging the lower overall representation of African countries in the dataset, proactive strategies are essential. Crafting tailored initiatives that align with local contexts and address specific challenges can unlock the vast potential for positive environmental impact and foster economic growth on the continent. By bridging this gap, the global community can strive for a more equitable and

comprehensive integration of sustainability practices across diverse regions, emphasizing the need for inclusive strategies that encompass the unique socio-economic and environmental landscapes of different continents.

4.1.3.2 Lollipop Chart

The lollipop chart provides a clear visualization of word frequencies in the venture descriptions dataset. Notably, terms like "company" (194,218) and "services" (183,712) stand out with the highest frequencies, highlighting a predominant focus on organizational and service-related aspects within the dataset. These words paint a picture of ventures that are heavily engaged in providing services and shaping their identity as companies.

Furthermore, words like "technology" (116,756) and "software" (84,014) indicate a substantial emphasis on the technological dimension of these ventures. This finding suggests that the dataset is characterized by a strong presence of tech-centric ventures. On the other hand, the low frequencies of words associated with environmental consciousness, such as "green" or "environment," imply that these terms are not frequently employed in the descriptions. This could signify a distinctive linguistic pattern among the companies, where eco-friendly aspects are not prominently featured in their textual representations.

4.1.3.3 Circular Bar Chart

Examining the circular bar plot illustrating sustainable companies in various sectors reveals intriguing patterns. Sectors like 'Energy' and 'Manufacturing' stand out with exceptionally high numbers of sustainable companies, indicating a strong commitment to sustainability in industries directly tied to environmental impact. In contrast, sectors like 'Music and Audio' and 'Messaging and Telecommunications' have notably fewer sustainable companies, suggesting potential areas for increased focus on sustainability practices.

The data underscores the significant impact of certain industries on overall sustainability efforts, with 'Energy' and 'Manufacturing' contributing substantially to the total count. Conversely, smaller counts in sectors like 'Music and Audio' and 'Messaging and Telecommunications' could prompt further exploration into the challenges or opportunities for sustainability in these areas. Additionally, the collective count of sustainable companies across all sectors emphasizes the overall commitment to sustainability across diverse industries. In conclusion, the circular bar plot serves as a useful tool for identifying trends and areas that may require more attention to foster sustainability across various business sectors.

4.2 Future Work

Expanding on the existing natural language processing groundwork with attention methods, future work can focus on refining the model's interpretability. Enhancing how easily users can understand the reasons behind the green venture assessments is vital. Techniques such as model-agnostic interpretability could be explored to highlight specific text features influencing the model's decisions, making the classification process more transparent.

Moreover, the user interface of the web application could be improved through iterative testing. Making the application more user-friendly and intuitive will enhance its usability. Additionally, integrating features for collaborative decision-making or knowledge sharing among users interested in sustainable ventures can foster a sense of community. These future directions aim to enhance both the technical aspects of the model and the overall user experience with the web application, ensuring practical utility and societal impact.

Bibliography

ALLEN, J. *Natural language understanding*. [S.l.]: Benjamin-Cummings Publishing Co, 1995. Citado na página 24.

BAEZA-YATES, R.; RIBEIRO-NETO, B. Modern information retrieval. v. 463, 1999. Citado na página 25.

BAKER, E. W.; SINKULA, M. J. Environmental marketing strategy and firm performance: Effects on new product performance and market share. *Journal of the academy of marketing science*, v. 33, p. 461–475, 2005. Citado na página 18.

BARONE, A. V. M. et al. *Regularization techniques for fine-tuning in neural machine translation*. [S.l.], 2017. Citado na página 38.

BARRETO, L. P. Educação para o empreendedorismo. *Educação Brasileira*, v. 20, n. 41, p. 189–197, 1998. Citado na página 13.

BEHRINGER, W. *A cultural history of climate*. [S.l.]: Polity, 2010. Citado na página 17.

BELLSTAM, G.; BHAGAT, S.; COOKSON, J. A. A text-based analysis of corporate innovation. *Management Science*, v. 67, n. 7, p. 4004–4031, 2020. Citado na página 24.

BERLE, G. The green entrepreneur: Business opportunities that can save the earth make you money. 1993. Citado na página 17.

BRÖNNIMANN, S. Picturing climate change. *Climate Research*, v. 22, n. 1, p. 87–95, 2002. Citado na página 17.

BROWN, N. History and climate change. 2001. Citado na página 17.

CASSOLA, F.; BURLANDO, M. Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output. *Applied energy*, v. 99, p. 154–166, 2012. Citado na página 21.

ČEKANAVIČIUS, L.; BAZYTĖ, R.; DIČMONAITĖ, A. Green business: challenges and practices. *Ekonomika*, v. 93, n. 1, p. 74–88, 2014. Citado na página 19.

CHIAVENATO, I. *Empreendedorismo: dando asas ao espírito empreendedor*. São Paulo: Saraiva, 2004. Citado na página 14.

CUN, Y. L. et al. Handwritten digit recognition with a back-propagation network. In: _____. *Advances in Neural Information Processing Systems 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990. p. 396–404. ISBN 1558601007. Citado na página 22.

DAI, A. M.; LE., Q. V. Semi-supervised sequence learning. *Advances in neural information processing systems*, v. 28, 2015. Citado na página 38.

DEVARAKONDA, A.; NAUMOV, M.; GARLAND, M. *Adabatch: Adaptive batch sizes for training deep neural networks*. [S.l.], 2017. Citado na página 40.

DEVLIN, J. et al. *Bert: Pre-training of deep bidirectional transformers for language understanding*. [S.l.], 2018. Citado 7 vezes nas páginas 9, 26, 27, 31, 33, 34, and 37.

DOLABELA, F. Luisa's secret. 2006. Citado 2 vezes nas páginas 13 and 14.

DORNELAS J. C., A. *Empreendedorismo: transformando ideias em negócios*. Rio de Janeiro: Elsevier, 2008. Citado 2 vezes nas páginas 13 and 14.

DRUCKER, P. F. The discipline of innovation. *Harvard business review*, v. 80, n. 8, p. 95–102, 2002. Citado na página 14.

ETSY, D.; WINSTON, A. Green to gold: How smart companies use environmental strategy to innovate. *Create Value, and Build Competitive Advantage*, p. 1–366, 2006. Citado na página 18.

FAWZY, S. et al. Strategies for mitigation of climate change: a review. *Environmental Chemistry Letters*, v. 18, p. 2069–2094, 2020. Citado na página 17.

FEURER, M.; HUTTER, F. Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, p. 3–33, 2019. Citado na página 39.

FUNAHASHI, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural networks*, v. 2, n. 3, p. 183–192, 1989. Citado na página 22.

GIBBS, D.; O'NEILL, K. Future green economies and regional development: A research agenda. *Regional Studies*, v. 50, p. 161–173, 2016. Citado na página 18.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. Citado 6 vezes nas páginas 7, 22, 23, 24, 39, and 46.

HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. *Neural network design*. [S.l.]: PWS Publishing Co, 1997. Citado na página 22.

HALL, J. K.; DANEKE, G. A.; LENOX, M. J. Sustainable development and entrepreneurship: Past contributions and future directions. *Journal of business venturing*, v. 25, n. 5, p. 439–448, 2010. Citado na página 17.

HISRICH, R. D.; PETER, M. P. *Entrepreneurship*. [S.l.]: McGraw-Hill/Irwin, 2002, 2004. Citado na página 15.

HULME, M. *Why We Disagree About Climate Change: Understanding Controversy Inaction and Opportunity*. Cambridge: [s.n.], 2009. Citado na página 17.

KASSAYE, W. W. Green dilemma. *Marketing Intelligence & Planning*, v. 19, p. 444–455, 2001. Citado na página 19.

KINGMA, D. P.; BA., J. Adam: A method for stochastic optimization. ArXiv preprint. 2014. Citado na página 44.

LAMB, H. H. *Climate, history and the modern world*. [S.l.]: Routledge, 2002. Citado na página 16.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. ArXiv preprint. 2017. Citado na página 45.

LUONG, M.-T.; PHAM, H.; MANNING, C. D. *Effective approaches to attention-based neural machine translation*. [S.l.], 2015. Citado na página 29.

MCCLOSKEY, M.; COHEN, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, v. 24, p. 109–165, 1989. Elsevier. Citado na página 43.

MITCHELL, T. M. 1ª edition. ed. [S.l.]: McGraw-Hill ISBN, 1997. Citado 2 vezes nas páginas 20 and 21.

MRKAJIC, B.; MURTINU, S.; SCALERA, V. G. Is green the new gold? *Venture capital and green entrepreneurship, Small Business Economics*, v. 52, n. 4, p. 929–950, 2019. Disponível em: <https://EconPapers.repec.org/RePEc:kap:sbusec:v:52:y:2019:i:4:d:10.1007_s11187-017-9943-x>. Citado na página 17.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, v. 18, n. 5, p. 544–551, 2011. Citado na página 24.

NGUYEN, A. et al. Translating videos to commands for robotic manipulation with deep recurrent neural networks. *IEEE International Conference on Robotics and Automation (ICRA)*, p. 3782–3788, 2018. Citado na página 24.

OKUMUS, I. *Turkey's Sustainable Development Performance In Terms Of Green Economy Indicators*. Dissertação (Mestrado) — Master's Thesis, Gaziantep, 2013. Citado na página 17.

OLIVEIRA, A. L. et al. Optical digit recognition for images of handwritten historical documents. *Ninth Brazilian Symposium on Neural Networks*, p. 166–171, 2006. Citado na página 21.

O'NEILL, B. C. et al. Achievements and needs for the climate change scenario framework. *Nature climate change*, Nature Publishing Group UK London, v. 10, n. 12, p. 1074–1084, 2020. Citado na página 17.

PACHECO, D. F.; DEAN, T. J.; PAYNE, D. S. Escaping the green prison: Entrepreneurship and the creation of opportunities for sustainable development. In: *Journal of Business Venturing, , vol. 25(5), , September*. [S.l.: s.n.], 2010. p. 464–480. Citado na página 17.

PENCLE, N.; MăLăESCU, I. What's in the words? development and validation of a multidimensional dictionary for csr and application using prospectuses. *Journal of Emerging Technologies in Accounting*, v. 13, n. 2, p. 109–127, 2016. Citado 3 vezes nas páginas 9, 25, and 26.

PETERS, M. et al. Deep contextualized word representations. *NAAC*, 2018. Citado na página 31.

PIRES, T.; SCHLINGER, E.; GARRETTE, D. How multilingual is multilingual bert? *arXiv preprint*, 2019. Citado na página 34.

PISTORI, H. *Tecnologia Adaptativa em Engenharia de Computação: estado da arte e aplicações*. Tese (Doutorado) — Escola Politécnica da USP, 2003. Citado 2 vezes nas páginas 20 and 21.

PRATI, R. C. *Novas abordagens em aprendizado de máquina para a geração de regras, classes desbalanceadas e ordenação de casos*. Tese (Doutorado) — ICMC-USP, São Carlos, 2006. Citado na página 20.

RAHAL, R.; DIAS, M. *Development of a green venture classification method using Natural Language Processing - BackEnd*. Disponível em: <https://github.com/rrrahal/tcc-modelo>. Citado na página 65.

RAHAL, R.; DIAS, M. *Development of a green venture classification method using Natural Language Processing - FrontEnd*. Disponível em: <https://github.com/rrrahal/tcc-frontend>. Citado 2 vezes nas páginas 65 and 66.

RAHAL, R.; DIAS, M. *General Company Dataset - Including Companies Description*. Zenodo, 2023. Data set. Disponível em: <https://zenodo.org/records/10362094>. Citado 2 vezes nas páginas 34 and 35.

RIVADENEIRA, A. W. et al. Getting our head in the clouds: Toward evaluation studies of tagclouds. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2007. (CHI '07), p. 995–998. ISBN 9781595935939. Disponível em: <https://doi.org/10.1145/1240624.1240775>. Citado na página 59.

RUDER, S. An overview of gradient descent optimization algorithms. ArXiv preprint. 2016. Citado na página 40.

SCHUMPETER, J. A. *The Theory of Economic Development (1st ed.)*. Routledge, 2021. Disponível em: <https://doi.org/10.4324/9781003146766>. Citado na página 13.

SUN, C. et al. How to fine-tune bert for text classification? *China national conference on Chinese computational linguistics*, 2019. Citado 8 vezes nas páginas 7, 9, 39, 40, 41, 42, 43, and 44.

SáRKöZY, F. Gis functions-interpolation. *Periodica Polytechnica Civil Engineering*, v. 43, n. 1, p. 63–87, 1999. Citado na página 22.

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Citado na página 27.

WITTEN, I. H.; FRANK, E. *Data Mining: Practical machine learning tools and techniques*. 2nd. ed. San Francisco: Morgan Kaufmann, 2005. Citado na página 21.

YANG, Y. An evaluation of statistical approaches to text categorization. *Information retrieval*, v. 1, n. 1-2, p. 69–90, 1999. Citado na página 26.

ZEILER, M. D. Adadelta: an adaptive learning rate method. ArXiv preprint. 2012. Citado na página 42.

ZHU, Y. et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Proceedings of the IEEE international conference on computer vision*, 2015. Citado na página 37.