CAIO DE PRÓSPERO IGLESIAS

FORECASTING DEFORESTATION IN THE PANTANAL BIOME USING A MACHINE LEARNING MODEL

São Paulo
2022

CAIO DE PRÓSPERO IGLESIAS

FORECASTING DEFORESTATION IN THE PANTANAL BIOME USING A MACHINE LEARNING MODEL

Graduation Work presented to the Polytechnic School – University of São Paulo for attaining the Production Engineer's Degree

São Paulo
2022

CAIO DE PRÓSPERO IGLESIAS

FORECASTING DEFORESTATION IN THE PANTANAL BIOME USING A MACHINE LEARNING MODEL

Graduation Work presented to the Polytechnic School – University of São Paulo for attaining the Production Engineer's Degree

Advisor:

Prof. Dr. Celma de Oliveira Ribeiro

Co-advisor:

Dr. Pedro Gerber Machado

São Paulo
2022

Catalogação-na-publicação

Iglesias, Caio
Forecasting Deforestation in the Pantanal Biome using a machine learning model / C. Iglesias -- São Paulo, 2022.
124 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Machine Learning 2.Desmatamento 3.Pantanal 4.Análise Preditiva 5.Sustentabilidade I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.

To my family and friends, who helped me get this far

ACKNOWLEDGMENTS

I want to thank all those who somehow supported me and believed in my capacity, encouraging me always to dream higher.

Mainly, I would like to thank my family for all their love and support and for always encouraging me in everything I have done. To my mother and father, I thank you for being excellent examples and for encouraging me to study, instigating my curiosity. To my brother, I thank you for your companionship and endless moments of partnership. Finally, to my uncles, cousins, and grandparents, I thank you for all the good moments and the incentive and support during the difficult moments.

I would also like to thank Dr. Pedro Gerber Machado and Prof. Dr. Celma de Oliveira Ribeiro for all their help and commitment during the execution of this work. You taught me a lot; this work would not have been possible without you.

I also thank all my friends for making the whole experience fruitful and helping me whenever needed. Moreover, I would like to thank everyone I have met during my internships so far: you have been essential in my learning and training.

Finally, I would like to thank all the teachers and mentors I had at school, at the University of São Paulo, and at the École CentraleSupélec, who contributed immensely to my academic, professional, and personal development. To them, all my gratitude for having shown me the pleasure of learning and the power of knowledge.

"Learning is the only thing the mind never exhausts, never fears, and never regrets."

-- Leonardo da Vinci

ABSTRACT

To mitigate climate change, halting deforestation is considered a critical part of one of the United Nations Sustainable Development Goals, which aims to protect life in terrestrial ecosystems. Unfortunately, however, the Pantanal, an area of significant socio-ecological relevance for Brazil and the world, is experiencing increasingly high deforestation rates. If the pattern continues, about 86% of the natural vegetation will be lost or irreversibly modified.

To achieve the goal of ending deforestation by 2030, innovative techniques are needed. However, while modern statistical techniques have already been used to understand and predict deforestation in South Asia or the Amazon, studies in the Pantanal region are scarce. In this context, this work aims to simulate different scenarios of land use decisions and burned areas to understand deforestation better and predict the amount of natural area lost by 2030 in each scenario.

To do this, a state-of-the-art Machine Learning model called XGBoost was used, which included features such as agricultural production, cattle head production, a burned area, and deforestation in previous years, among others, to predict deforestation in the Pantanal. After several iterations testing different features and hyper-parameters, the model reached a median R2 value of 75.28%, demonstrating a good predictive ability.

Thus, the forecast in a realistic scenario is that about 2.65 million hectares will be deforested between 2021 and 2030, considering all the municipalities surrounding the Pantanal. In addition, it has been shown that decreasing cattle production by about 14% (with, for example, Brazil joining the Meat Free Monday) and controlling the number of burned areas can prevent 10% of the natural area from being lost, compared with the pessimistic scenario. Finally, the Pantanal is going through one of its most problematic phases, and immediate action is needed to guarantee the quality of life for future generations.

Keywords – Pantanal, deforestation, sustainability, forecasting, Machine Learning, XGBoost, land use, fires.

RESUMO

Para mitigar as mudanças climáticas, acabar com o desmatamento é considerado como uma parte crítica de um dos Objetivos de Desenvolvimento Sustentável da Organização das Nações Unidas (ONU), que visa proteger a vida em ecossistemas terrestres. Porém, o Pantanal, área de fundamental relevância sócio-ecológica para o Brazil e para o mundo, apresenta taxas de desmatamento cada vez mais elevadas. Se o padrão continuar, cerca de 86% da vegetação natural será perdida ou irreversivelmente modificada.

Para atingirmos o objetivo de acabar o desmatamento até 2030, são necessárias técnicas inovadoras. Porém, enquanto técnicas estatísticas modernas já foram utilizadas para entender e prever o desmatamento no sul da Ásia ou na Amazônia, estudos na região do Pantanal são muito raros. Nesse contexto, esse trabalho tem como objetivo simular diferentes cenários de decisões de uso da terra e de áreas queimadas para entender melhor o desmatamento e prever qual o tamanho da área natural que será perdida até 2030 em cada cenário.

Para isso, foi utilizado um modelo state-of-the-art de Machine Learning chamado XGBoost, que usou features como produção agrícola, produção de cabeça de gado, área queimada, desmatamentos nos anos anteriores, entre outras, para prever o desmatamento no Pantanal. Depois de diversas iterações testando diferentes features e hiper-parâmetros, o modelo atingiu como mediana um valor de R^2 de 75.28%, demonstrando uma boa capacidade preditiva.

Assim, a previsão em um cenário realista é que cerca de 2.65 milhões de hectares sejam desmatados entre 2021 e 2030, considerando todos os municípios que envolvem o Pantanal. Além disso, foi demonstrado que diminuir a produção de gado em cerca de 14% (tendo por exemplo o Brasil participando da iniciativa Meat Free Monday) e controlando o número de áreas queimadas pode evitar que 10% da área natural seja perdida, em relação ao cenário pessimista. Finalmente, o Pantanal passa por uma de suas fases mais problemáticas e agir imediatamente é necessário para garantir a qualidade de vida das futuras gerações.

Palavras-Chave – Pantanal, desmatamento, sustentabilidade, análise preditiva, Aprendizado de Máquina, XGBoost, uso da terra, incêndios.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

PART I

INTRODUCTION

"Living is worthwhile if one can contribute in some small way to this endless chain of progression"

-- Paul Dirac

Over the last 20 years, the world has observed relevant gains in economic activity, which cost a great deal in terms of equitability and sustainability (DOMINGUEZ et al., 2022). Hence, with the growth of the population and consumption per capita, sustainability is one of our main challenges as a society today.

To try to mitigate climate change, diminishing deforestation has been promoted as a critical part of one of the United Nations' Sustainable Development Goals (SDGs), which aims to protect, restore and promote sustainable usage of terrestrial ecosystems (NATIONS, 2016). As deforestation is responsible for around 10% of global warming (WWF, 2022), it permeates a series of discussions worldwide involving several stakeholders, from managers to environmentalists (SILVA; ABDON, 1998).

Substantial work has been done to estimate, track, and comprehend deforestation patterns and drivers in South Asia, South-Central Africa, and the Amazon region in South America (SILVA LEILA M.G. FONSECA, 2019). However, some very relevant areas from a socio-ecological point of view, such as the Pantanal, have hardly been analyzed, even more through modern modeling techniques. With more than 180,000 km^2, the Pantanal was declared a National Patrimony by the Federal Constitution of 1988 and a World Biosphere Reserve and a Natural Patrimony of Humanity by UNESCO in 2000(TORTATO, 2018).

However, deforestation has already compromised a large part of the highlands surrounding the Pantanal (ROQUE et al., 2016). If the current pattern continues, more than 86% of the natural vegetation of the Brazilian Pantanal wetlands will be soon lost or irreversibly modified, leading to severe consequences, such as changes in the cycles of floods and droughts, compromise of the biodiversity, and deregulation of the climatic-hydrologic dynamics(MIRANDA C. S., 2018). Moreover, it was estimated that, from the Brazilian government's point of view, the cost of deforestation is more than US$7300 per hectare (ha) - considering all types of factors that will be lost, from water regulation to ecotourism - while the benefit of implanting cultivated pastures in the deforested area is only around US$28/ha (MORAES, 2008).

Several threats to the Pantanal, of different natures, have been identified and classified into two groups: from outside the region, in the form of activities that adversely impact the

rivers that feed it, and from inside the Pantanal itself, in the form of activities inside the biome that impact it directly, its ecosystem (MITTERMEIER et al., 1990). Deforestation was a crucial threat in both categories, and the primary motivations for the deforestation were cattle pasture, and intensive agriculture (MITTERMEIER et al., 1990).

To achieve the Sustainable Development Goal of halting deforestation by 2030, innovative approaches to deforestation forecasting are urgently needed (BALL et al., 2021) because of its complexity and interaction of numerous socioeconomic, political, and environmental factors at different spatial and temporal scales (JAFFE et al., 2021). In this context, statistical techniques that can handle multiple data types can be used to forecast and simulate different scenarios (DOMINGUEZ et al., 2022). One such technique is Machine Learning, which is the science of programming computers so they can learn from data (GERON, 2017).

Given the aforementioned considerations, this research aims at predicting Pantanal's deforestation until 2030 using modern Machine Learning techniques under different scenarios that vary in terms of agricultural production, the number of heads of cattle produced, and the area burned during the years. The model chosen for the study is Extreme Gradient Boosting (XGBoost), an algorithm that combines many simple models sequentially in a clever manner and yields state-of-the-art results.

Thereby, our main objective is to simulate different scenarios of agricultural/cattle decisions and burned areas (a realistic one, an optimistic one, and a pessimistic one) that can aid in decision-making for strategic land use decisions in Brazil. As a secondary objective, we want to build a robust and scalable code base so others can contribute to the project, increasing its reach and impact. Finally, hopefully, this project helps to ensure the integrity of this critical biome and to guarantee the well-being of future generations.

PART II

LITERATURE REVIEW

1 PANTANAL AND ITS IMPORTANCE

Pantanal is peculiar due to its importance in preserving biodiversity. That is why it has been studied by many authors, who carefully described its importance and the main threats it suffers. The goal of this chapter is thus to make a deep dive into the literature and motivate our study.

Figure 1: Map of Brazilian biomes.

Source: Extracted from (IBGE, 2014)

Located in the heart of South America, it constitutes a geological depression in the upper Paraguay river with an area of about 180,000 km^2. It is considered the world's most significant tropical wetland area, located in the central portion of South America, with its territory distributed 80% in Brazil, 15% in Bolivia, and 5% in Paraguay (SWARTS, 2000). Table 1 shows a more granular view of how Pantanal is distributed in Brazil, divided by its most relevant municipalities. For instance, one can see that 44,74% of the Pantanal is located in a single municipality, called Corumbá, in Mato Grosso do Sul. This will be

useful in our study since it will focus exclusively on the Brazilian Pantanal; thus, only those municipalities will be considered.

Table 1: Participation of the municipalities in the Pantanal physiographic area (km^2).

Municipalities	Plateau	Pantanal (A)	Total (B)	Total IBGE	A/B (%)	A/C (%)
Mato Grosso	31.170	48.865	80.035	81.955,89	61,0	35,36
Barão de Melgaço	83	10.782	10.865	11.611,78	99,2	7,80
Cáceres	11.051	14.103	25.154	25.321,14	56,1	10,21
Itiquira	6.751	1.731	8.482	8.836,98	20,4	1,25
Lambari D'Oeste	1.439	272	1.711	1.719,1	15,9	0,20
Nsa Sra. Livramento	4.019	1.115	5.134	5.331,57	21,7	0,81
Poconé	3.434	13.972	17.406	17.126,38	80,3	10,11
Sto. Ant. Leverger	4.393	6.890	11.283	12.008,94	61,1	4,99
Mato Grosso do Sul	37.193	89.318	126.511	131.417,50	70,6	64,64
Aquidauana	3.936	12.929	16.865	17.008,00	76,7	9,36
Bodoquena	2.500	46	2.546	2.514,30	1,8	0,03
Corumbá	2.858	61.819	64.677	65.165,80	95,6	44,74
Coxim	4.351	2.132	6.483	10.844,40	32,9	1,54
Ladário	311	66	377	341,40	17,5	0,05
Miranda	3.421	2.106	5.527	5.494,50	38,1	1,52
Sonora	3.598	719	4.317	4.088,90	16,7	0,52
Porto Murtinho	12.739	4.717	17.456	17.782,90	27,0	3,41
Rio Verde de MT	3.479	4.784	8.263	8.177,30	57,9	3,46
Total (C)	68.363	138.183	206.546	213.373,39	66,9	100,00

Source: Extracted from (SILVA; ABDON, 1998)

The Pantanal is bordered by the Cerrado, by the semi-deciduous forest of the transition zone between Amazonia and cerrado, and by the dry Chaco formations of Bolivia and Paraguay and the vegetation includes elements from all these formations, constituting thus a diverse mosaic, as can be seen on Figure 1. Furthermore, the biome is recognized globally by its fauna abundance (MITTERMEIER et al., 1990). Therefore, as mentioned in Part I, it has been considered a National Patrimony by UNESCO since 2000. (TORTATO, 2018)

In more detail, Pantanal's flora has approximately 2000 species of plants. As for the fauna, about 582 species of birds are identified in the Pantanal, 132 species of mammals, 265 species of fish, 113 species of reptiles, and 41 species of amphibians. In addition, the Pantanal has several endangered species, such as the hyacinth macaw (Anodorhynchus hyacinthinus) and the jaguar (Panthera onca) (TORTATO, 2018).

The vegetation cover has been significantly altered in the Brazilian Pantanal wetland. Deforestation has already compromised a large part of the highlands surrounding the

Pantanal (ROQUE et al., 2016). As mentioned in Part I, if the current pattern continues, more than 86% of the natural vegetation of the Brazilian Pantanal wetland will be lost, or modified (MIRANDA C. S., 2018). Moreover, this could lead to severe consequences, such as changes in the cycles of floods and droughts, compromise of the biodiversity, deregulation of the climatic-hydrologic dynamics, etc. (MIRANDA C. S., 2018)

Furthermore, approximately 95% of the Pantanal is made up of private properties, and only 4.6% are protected by conservation units (UCs, *Unidade de Conservação* in Portuguese), of which only 2.9% are full-protection and 1.7% sustainable use UCs, such as Private Natural Heritage Reserves (HARRIS et al., 2005) (TORTATO, 2018). Therefore, it becomes evident that there is a need for stricter public policies aiming at the sustainable use of these lands and their conservation (MIRANDA C. S., 2018).

It is also essential to understand the threats to the Pantanal and its different natures. These threats can be divided into two distinct categories: from outside the region (in the form of activities that adversely impact the rivers that feed it), and threats from inside the Pantanal itself (in the form of activities inside the biome that impact it directly) (MITTERMEIER et al., 1990). The primary threats from outside the region are deforestation and burning of forests at the headwaters of rivers feeding the river, pollution (mainly due to toxins used in agricultural projects outside of Pantanal), gold mining, and the international animal and skin trade. On the other hand, the main threats from within the region are: drainage (for agriculture and cattle ranching), commercial overfishing, hunting, destructive tourism, and deforestation for cattle pasture and intensive agriculture (MITTERMEIER et al., 1990). Therefore, one may notice that deforestation is a critical threat in both categories, mainly motivated by cattle pasture and intensive agriculture.

In addition, some cost-benefit analyses were done from the Brazilian government's point of view (MORAES, 2008), quantifying the different types of benefits lost by deforesting 1 hectare (ha) of Pantanal and the benefits gained from deforestation. The results can be seen in Table 2. The costs of deforestation are about US$7383/ha, while the benefit is about US$28.2/ha (MORAES, 2008) if it is considered that the entire area will be used for cultivated pasture. In other words, the opportunity cost of preserving the areas is derisory compared to the potential benefits of keeping the area preserved. It is also worth highlighting that the public benefits lost represent more than 96% of the total cost, which indicates the need for public policies (MORAES, 2008).

Table 2: Cost-benefit analysis from the Brazilian government point of view (in US$/ha, in 2007 prices).

Costs and Benefits	Value (US$/ha)
Costs of deforestation	**7387**
Local private benefits lost	260
Local public benefits lost	7127
Benefits of deforestation	**28**
Implantation of cultivated pastures	28

Source: Adapted from (MORAES, 2008)

According to the author, (MORAES, 2008), the leading local private benefits that would be lost with deforestation are: timber products, non-timber forest products, ecotourism, and ranching on native pastures. The local public benefits lost include disturbance regulation, water regulation, water supply, waste treatment, erosion control, and culture. Thus, the main conclusion from the author's point of view is that the cost-benefit of deforestation does not favor deforestation. Moreover, the author emphasizes that from a global point of view, the external benefits of the Pantanal reach a value of about US$10,0062/ha, considering factors such as the regulation of the emission of greenhouse gases, climate regulation, genetic resources, existence value, among others. Thus, the author even argues that a global planner should offer benefits for Brazil to preserve the Pantanal (MORAES, 2008).

Another critical point worth mentioning is the relevance of the fires in the deforestation of the Pantanal biome. Pantanal is the biome that most burned in Brazil in 36 years, having a total of $84.403km^2$, which represents a total of 57.5% of its area. For contrast purposes, the second most burned biome was the Cerrado, with around 36% of its area burned (MAPBIOMAS, 2021). Moreover, in Brazil, around 65% of all the burned areas occurred in native vegetation, i.e., in natural areas. In addition, Corumbá, one of the principal municipalities of Pantanal, was the municipality with the largest burned area in km^2 until 2020 (MAPBIOMAS, 2021). This indicates the relevance of considering the fires in natural areas in our analysis and modeling in Chapter 6.

In conclusion, considering the local and global importance of the Pantanal, its main threats, and the potential financial risk, it is clear that it is vital to understand the different scenarios for deforestation of this biome in the coming years, to assist public

decision-making and avoid significant natural and socioeconomic losses.

2 FORECASTING DEFORESTATION

As seen in the introduction, deforestation is an important topic to understand nowadays. As Brazil has vast areas of vegetation rich in fauna and flora, such as the Amazon and the Pantanal, it is the target of many studies that deal with this topic. Here, the main interest is in studies on predictive analysis of deforestation, which seeks, based on historical data, to create models that can predict the deforestation of the coming years as accurately as possible. Also, the aim is to understand which mathematical models are being used and to see if these studies manage to study the impact of different scenarios/decisions on deforestation since this is our final goal in this study.

While searching for the primary references in deforestation forecasting, some interesting facts could be noticed. Firstly, most of the key references were studies done in the Amazon rainforest since it contains half of the planet's remaining tropical forests and holds 10% of the global carbon reserves (JAFFE et al., 2021). Therefore, the number of papers using Statistical/Machine Learning models to predict deforestation in Pantanal was small, as will be shown next.

Moreover, researchers are using various models to solve this type of problem. For instance, articles use Deep Convolutional Neural Networks, Hybrid Deep Learning Models with Long Short Term Memory neural networks (LSTM), Bayesian hierarchical spatial models, etc. As the papers are being presented, the idea behind each method will be quickly discussed. Anyway, bear in mind that those are different modeling strategies and can serve different general purposes. Also, one may note that while many authors try to understand the drivers of deforestation, i.e., the essential features for the models, they usually do not explore different scenarios, which could help in public policy decisions. Lastly, most articles were published after 2021, reinforcing the idea that this is a hot topic.

(BALL et al., 2021) explores Computer Vision models with the use of different architectures of Deep Convolutional Neural Networks to predict deforestation. Computer vision is a field of Artificial Intelligence (AI) that enables computers to extract relevant

information from images (IBM, 2022). Neural Networks, on the other hand, are a type of Machine Learning model inspired by the visual cortex of animals that have reached incredible success in the last few years in different domains, such as translation, image recognition, and predictive analysis (CHOLLET, 2017). Deep Convolutional Neural Network (CNN) is a type of Neural Network that can extract patterns (such as detecting edges and color patterns) from images, thus being one of the most successful models today for Computer Vision (GERON, 2017). It is called "convolutional" since the main operation performed by the model is a mathematical operation called convolution (usually a two-dimensional convolution).

In this article, the main idea is that the Deep CNN will be able to learn spatiotemporal patterns from the pixels of the satellite imagery. The models explored by the authors go from 3D Convolutional Neural Networks (a type of Deep CNN, where the convolution operation is done in three dimensions instead of two) to Convolutional Long Short Term Memory (a type of Neural Networks that is specialized in training time series, by being able to retain information about the data it has seen in the past). These models aimed to predict spatial maps that indicate each pixel's deforestation risk in the next year. The best performing model was the 3DCNN network, with a pixel-wise accuracy of between 80% and 90%, i.e., the authors successfully identified which pixels would be deforested with this accuracy. One interesting driver noticed by the authors was that pixels around new access routes, such as roads, usually had a high risk of deforestation. At the same time, the model did not assign a high risk to places that recently suffered natural loss events.

Another interesting paper concerning deforestation in the Amazon rainforest is by (SILVA LEILA M.G. FONSECA, 2019). The study focused on a specific region in the southwest of Pará state, a deforestation expansion frontier. The model used here was a stepwise spatiotemporal Bayesian Network approach. In a nutshell, a Bayesian Network can be defined as a probabilistic graphical model representing information about uncertainty in a graph, with each variable in a node and the conditional probability on the edges (YANG, 2019). For that, they used several static and temporal variables, such as distance from degraded areas, distance from hot spots, distance from pasture areas, protected areas, distance from roads, etc. As addressed by the article, Bayesian Networks (BNs) are suitable approaches to inferring about static processes. Still, they have been used to model the spatially explicit process by linking them to software that processes geographical information data. Using the observed values (i.e., evidence), the BN model can then predict a time series of probability images, where the pixel values represent the

probability of deforestation in that area. The most important features identified by the authors to predict deforestation are the distance from a hot spot (i.e., fires) and the distance from degraded areas. In contrast, the area of protected forests was the feature that mitigated deforestation the most.

The paper by (JAFFE et al., 2021) is another interesting one that aims to predict deforestation in the Amazon rainforest using Bayesian methods. Here, the authors coupled high-resolution land-cover maps with Bayesian hierarchical spatial models to predict the areas more likely to be deforested in the next three years. Moreover, the authors wanted to identify the area's main drivers of recent deforestation. They concluded that the recent deforestation was positively associated with forest edge density (the concentration of edges between forest and other land use classes), the length of roads and waterways, elevation, and terrain slope. Moreover, it was negatively associated with distance to urban areas, roads, indigenous lands, areas designated as protected or indigenous territory, and municipality GDP per capita. The main drivers between these were the forest edge density and the distance to roads, which demonstrated more considerable predictive power. The authors also sustain the idea that short-term predictions can be used by the authorities to prevent the deforestation of these areas.

Until now, the methods to predict deforestation in the Amazon rainforest included the Bayesian and CNN methods. The paper (COSTA et al., 2021), however, uses a different methodology to predict deforestation in the Legal Amazon. The authors use Box-Jenkins, a time series methodology that applies autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) models to fit the time series. Therefore, while other papers tried to use a notion of space to predict deforestation in a specific location, this paper aimed to predict the forestation in the Amazon Legal area more generally, i.e., the area that will be deforested in the following years, based on the time series. The conclusion was that if there were no significant interventions in the series, the deforestation rates were expected to remain from 7559.97 km^2 to 7730.88 km^2. However, in the case of an intervention (external factors that can be physical, economic, political, and so on), they concluded that these numbers would go as high as 10429.28km^2 to 28669.75 km^2, representing an increase of almost 120%. Therefore, the authors reinforce the importance of maintaining and expanding the environmental governance structure for the Brazilian Legal Amazon, mainly based on the instruments of Command and Control.

The last article related to forecasting deforestation in the Amazon rainforest is the (DOMINGUEZ et al., 2022), which uses yet another type of modeling and data structure. In this paper, the authors chose to use a Hybrid Deep Learning Model that combines

a Dense Neural Network (i.e., the traditional neural network that mimics the cortex of animals) with a recurrent LSTM, defined above. The Dense network received as inputs the static data, namely geographical, forest and watershed, and the LSTM received a time series with the annual deforestation of the last 20 years (1999-2019). After many iterations, trying different setups and hyper-parameters, and using augmented data, the authors obtained an R-squared score of 87.82%, which is the proportion of variability in the data explained by the model. They predicted that in 2030 deforestation will reach around 1 million km^2, and they reinforce that the result will help us understand the impact of man's footprint on the Amazon rainforest.

Besides this great variety of work done with the Amazon rainforest, some work has also been done in the Pantanal biome. For instance, the paper (SILVA et al., 2011) studied the evolution of deforestation in the Brazilian Pantanal and its surroundings from 1976 to 2008. The authors concluded that the Plateau was the most affected area, with around 58.90% deforested in the period studied. Moreover, the authors used a simple statistical method that used the average geometric growth rate to predict each subdivision's deforestation until 2050. The authors concluded that if the present conditions remain, the tendency is to suppress the natural vegetation from the Plateau until 2029. Finally, the authors recommend that an installation of a monitoring system should be done to enhance effective deforestation control.

Another group of researchers from Brazil, Colombia, and the U.K. used a spatially explicit model to identify drivers of vegetation loss in Pantanal, and the surrounding area, i.e., the Upper Paraguay River basin (here called UPRB) (GUERRA et al., 2020). They used a probabilistic model that considers vegetation loss as contagious but also considers the drivers identified in those locations. More precisely, the model uses Monte Carlo Markov Chains (MCMC) to obtain a posterior probability that a "native vegetation" cell x is converted into "anthropogenic use" within a defined time interval t. The authors concluded that the drivers of vegetation loss for the lowland are distance to roads and rivers and elevation, while the drivers for the Plateau are distance to cities. Moreover, they concluded the cumulative rate of native vegetation loss projected for 2050 was around 3% for the lowland and 10% for the Plateau. Finally, they indicated that identifying an arc of vegetation loss requires urgent conservation policies and new perspectives for management.

2.1 Forecasting Deforestation Literature Review Summary

Finally, a summary of the studies reviewed is shown in Table 3. It shows the diversity in the modeling used, the specific goals, and the location they explored. The methods go from simple statistical methods, like Average Geometric Growth Rate, to Bayesian spatial models and complex deep learning structures, such as 3DCNN or Hybrid Deep Learning models with Dense Neural Networks and LSTM.

However, one may notice an absence of articles addressing deforestation prediction in the Pantanal, especially with modern Machine Learning techniques, which can be extremely useful in providing insights to decision-makers. Moreover, most papers looked for a spatial relationship to predict the probability of deforestation in a particular pixel. Still, in general, the papers did not explore how the main threats of deforestation (agriculture and cattle ranching, as exposed in chapter 1) affect vegetation loss in this region.

Therefore, our goal is to propose a modern approach to predicting deforestation in the Pantanal, with the possibility of exploring how land-use decisions will affect the output. Our primary reference will be the article (DOMINGUEZ et al., 2022). Still, instead of using Dense Neural Networks with LSTM, another model called XGBoost will be used, with the methodology described in the chapter III. However, before moving there, let us first explore the literature on Supervised Learning and Tree-based Machine Learning so that the key concepts to develop our model are explored.

Table 3: Summary of articles reviewed on deforestation forecasting.

Reference	**Title**	**Location**	**Model**	**Main conclusions**
(BALL et al., 2021)	Using deep CNNs to forecast spatial patterns of Amazonian deforestation	Amazon rainforest	2DCNN, 3DCNN, ConvLSTM	- Pixel-wise accuracy of around 80-90%; - Pixel around new access routes have high risk; - Pixel around recent natural loss event have low risk.
(SILVA LEILA M.G. FONSECA, 2019)	A Spatio-temporal Bayesian Network approach for deforestation prediction in an Amazon rainforest expansion frontier	Amazon rainforest (expansion frontier in the southwest of Pará)	Stepwise Spatio-temporal Bayesian Network	Highest contribution variables: - Distance from hot spot; - Distance from degraded areas. Highest mitigation variables: - Protected area.
(JAFFE et al., 2021)	Forecasting deforestation in the Brazilian Amazon to prioritize conservation efforts	Amazon rainforest	Bayesian hierarchical spatial models	- Accuracy is better for smaller time windows; - Most deforestation in North-Easter of Amazon (5 million ha); Main drivers: - distance to road (negatively related); - forest edge density (positively related).
(COSTA et al., 2021)	Deforestation forecasts in the Legal Amazon using intervention models	Amazon rainforest (Legal Amazon)	Box-Jenkins	No intervention: - Deforestation around 7559.97 - 7730.88 km2 Intervention: - Deforestation around 10429.28 - 28669.75 km2 Hence, need for maintenance and expansion of the environmental governance structure for the Brazilian Legal Amazon
(DOMINGUEZ et al., 2022)	Forecasting Amazon Rain-Forest Deforestation Using a Hybrid Machine Learning Model	Amazon rainforest	Hybrid Deep Learning Model (LSTM + Dense Neural Networks)	- R-squared: 87.82%; - Deforestation will reach around 1 million km2 by 2030
(SILVA; ABDON, 1998)	Evolution of Deforestation in the Brazilian Pantanal and surroundings in the timeframe 1976 - 2008	Pantanal (Upper Paraguay river basin)	Average Geometric Growth Rate	- Plateau most affected with around 59% deforested, while the floodplain had around 12% deforested. If conditions remain: - Suppression of natural vegetation from plateau until 2029 - Suppression of natural vegetation from floodplain until 2045 Hence, there should be an installation of a monitoring system.
(GUERRA et al., 2020)	Drivers and projections of vegetation loss in the Pantanal and surrounding ecosystems	Pantanal (Upper Paraguay river basin)	Monte Carlo Markov Chains (MCMC)	Cumulative rate of native vegetation loss by 2050: - 10% for the plateau; - 3% for the lowland. Identification of an arc of vegetation loss, which requires urgent conservation policies and new perspectives for management.

Source: The author.

3 AN INTRODUCTION TO XGBOOST

This chapter aims to present the state-of-the-art model, called XGBoost, that will be used in this study. However, XGBoost is quite a complex model. Therefore, some of its building blocks are going to be presented first. Hence, the sections of this chapter are: (i) an introduction to what is Supervised Learning (the most know area in Machine Learning); (ii) the concept of Regression Trees, the simplest tree-based regression model; the concept of Gradient Boosting that will combine the Regression Trees in a more complex model; (iv) the XGBoost and its improvements over Gradient Boosting.

In a nutshell, the XGBoost is a scalable ML system for tree boosting that has gained a lot of prominence in Machine Learning competitions, such as those in Kaggle. For instance, among the 29 challenge-winning solutions in 2015, 17 used XGBoost, either solely or as part of an ensemble (when a group of machine learning models is used jointly). The second most popular method was neural networks, used in about 11 of the solutions (CHEN; GUESTRIN, 2016b).

3.1 The principles of Supervised Learning

3.1.1 A general view and notation

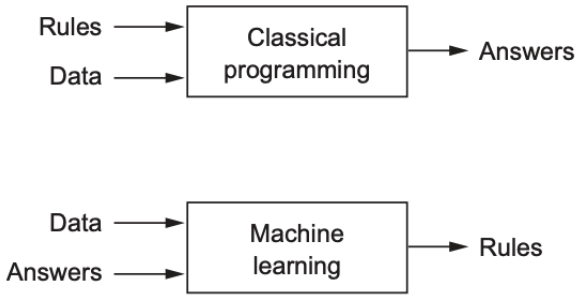
Supervised Learning is a topic studied in the context of Statistical Learning (or Machine Learning). Generally speaking, it concerns statistical models that try to predict or estimate the output based on a set of inputs (JAMES et al., 2013). Some examples of problems that fall in this category are: predicting if an email is spam or not based on its text, predicting the price of a house based on its characteristics, or predicting the deforestation of the Pantanal Forest in Brazil based on historical data and land use decisions. The inputs are the model's features, and the output is called the target variable.

In our case, for example, the target variable is the deforestation in Pantanal for a specific year in a particular municipality, and the features are variables that could

help the model in the forecasting, such as the number of fires, agricultural and livestock production, last year's deforestation, and so on.

The first step is to train the model to learn how to predict the outputs. This means some examples of inputs and outputs will be given, and our model will try to learn the underlying rules to best predict the outputs based on this set of inputs. This approach differs from the classical programming one, as shown in Figure 2 (CHOLLET, 2017).

Figure 2: Machine Learning: a new programming paradigm.



Source: Extracted from (CHOLLET, 2017).

Note here that the nature of the outputs can vary, being quantitative (e.g., the price of a house) or qualitative (e.g., whether an email is a spam or not). When the output is quantitative, the problem is called a regression problem; when it is qualitative, it is called a classification problem. Let us discuss the regression problem here since our target variable (deforestation in a particular year and municipality) will be quantitative.

Let us then use the following notation:

- Input: A $n \times p$ matrix $\boldsymbol{X}$, where n is the number of examples the model will be trained with, and p is the number of features.
- Output: A $n \times 1$ vector $\boldsymbol{Y}$, which represents our output variable for each of the n examples.

Then, if one lets $X \in \mathbb{R}^p$ be a random input vector (it would be equivalent to one example of our matrix $\boldsymbol{X}$), and $Y \in \mathbb{R}$ a real-valued random output variable (the corresponding output of this example in $\boldsymbol{Y}$), one can write:

$$Y = f(X) + \epsilon \tag{3.1}$$

where $f(X)$ is some fixed unknown function and ϵ is a random error term, independent from X and with mean 0. Then, let us try to find $\hat{Y} = \hat{f}(X)$, that approximates well Y. In a linear regression, for example, finding the function $\hat{f}(X)$ is equivalent to finding the coefficients, i.e., the constant values that multiply the features. Learning these coefficients is the step called training the model. As seen in the section 3.2, for regression trees, the training step will help us find the best splits to build the tree that best fits our data.

3.1.2 Measuring the quality of our fit

Our prediction error will have as components a reducible error that depends on how well $f(X)$ can be estimated and an irreducible error that depends on ϵ. This irreducible error exists because even with the best model, the features selected (X) may not be enough to predict Y perfectly. One way of assessing how well our model fits the data (or the quality of the fit to our data) is the Root Mean Squared Error (RMSE), which can be defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2} \tag{3.2}$$

where $\hat{f}(x_i)$ is the prediction for the ith observation (deforestation predicted for a municipality and year) and y_i the value observed (the actual deforestation for the municipality in that year).

Essentially, this is a way of computing the distance between two vectors (the observed values vector, y and the prediction vector, $\hat{f}(x)$) using the Euclidean distance (i.e., the ℓ_2 norm). However, since one takes the square of the errors before the average, it gives a relatively high weight for significant errors, which might be undesirable if one has many deforestation values that are outliers (GERON, 2017). Therefore, another more robust metric is the MAE, which can be defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.3}$$

In this case, it is the ℓ_1 norm (GERON, 2017). Finally, the last validation metric worth defining for this study is the R^2. It can be defined as:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} \tag{3.4}$$

where RSS represents the residual sum of squares, defined by $RSS = \sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$ and TSS represents the total sum of squares, defined by $TSS = \sum_{i=1}^{n}(y_i - \bar{y})$ (JAMES et al., 2013). Recall here that $\bar{y}$ represents the mean of the target variables. Thus, the TSS measures the total variance in the output variable, and RSS measures the variability not explained by the model. Therefore, the R^2 measures the percentage of variability in Y that our model could explain. It is worth mentioning that, besides being called R-squared, outside the context of linear regression, the R^2 can be a negative value if the model used is worst than using the horizontal line with the average as a prediction. Therefore, generally, $R^2 \in (-\infty, 1]$.

However, there is an important subtlety. Measuring these quantities on the training data or unseen data is entirely different. This is going to be addressed in the next section.

3.1.3 Bias-Variance trade-off

The metrics defined can be computed in two different ways: (i) for the training data; (ii) for unseen data (or test data). The most natural approach seems to be (i), i.e., train the model for all the observations (for example, the deforestation in multiple years and the features selected for the same years) and then see how well the model predicts the deforestation for these observations. This might seem like a good approach at first, but there is no guarantee that the training $RMSE$ (i.e., calculating it over the training dataset) will be close to the Test $RMSE$ (i.e., calculating it on new data).

Figure 3 exemplifies this problem. Suppose our Y variable is the deforestation area in a particular year X. The black points represent the actual deforestation value for these years, i.e., these are the y_i values, each point a different observation (a different i). Moreover, three different models ($\hat{f}$) can be chosen to forecast deforestation.

One may think that the green curve is the best option since it fits the black points pretty accurately. However, Figure 3 (b) shows otherwise. In this chart, the Test Mean Squared Error (MSE, the square of the $RMSE$) is represented in red, and the Training MSE in gray. They are both represented as a function of the Flexibility of the model, and the colored rectangles correspond to one of the $\hat{f}$ in Figure 3 (a). For example, the green curve is the rightmost rectangle, the more flexible/complex model.

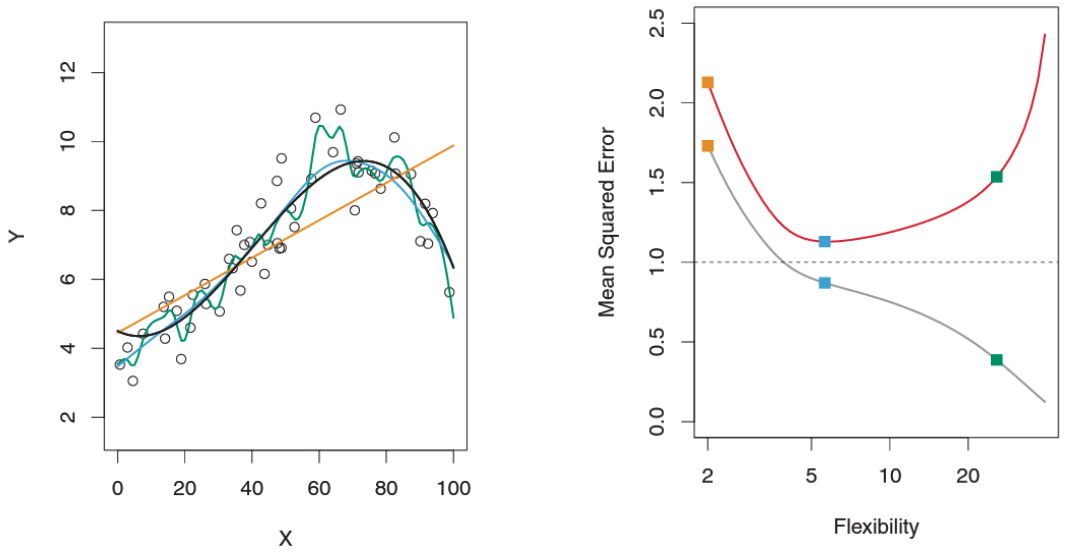
One may see that this model indeed has a good Training MSE, as it fits well the black points. However, it performed poorly on unseen years, represented by the red $U - shaped$ curve. This phenomenon is usually called overfitting, i.e., when your model

fits the training data well but does not generalize well. The problem is that the model fits all the noise in the data. When this happens, one may say that the model has high variance, meaning that if another training data were selected, the $\hat{f}$ would be completely different (high variability of the prediction function) since it will try to fit the new training data perfectly.

Figure 3: Example of the importance of the bias-variance trade-off.

(a) Three different $\hat{f}$, with different complexities

(b) The Test MSE (Red) and the Training MSE (Gray)



Source: Extracted from (JAMES et al., 2013)

On the other hand, the linear regression in orange is a really simple model for the problem. In this case, the model would have low variance (the curve is likely to be similar to a new training set) but high bias. It is said the model is biased because it is expected that the model is always off by a little since it is a simpler model than needed. Generally, there is a trade-off between the bias and the variance. A model that balances both well (as the blue curve in our example) is ideal, with just enough complexity for predicting the unseen data well.

The two main conclusions are: (i) always evaluate the metrics on unseen data; (ii) when building a model, always balance bias and variance to obtain the smaller Test Error. Point (i) will be particularly relevant in section 6.3, where the model's performance will be evaluated, and point (ii) in section 6.2, where the hyper-parameter tuning will be made to balance bias and variance automatically. Moreover, a more technical discussion can be

found on A.3.

3.1.4 Hyper-parametrs

The goal of this subsection is to define what a hyper-parameter is and its role in Machine Learning. There are two types of parameters in Machine Learning Models: (i) one that can be initialized and updated during the training phase (e.g., the weights β in a Linear Regression); (ii) the hyper-parameters that define the ML model architecture and that need to be determined before the training phase (YANG; SHAMI, 2020). As a simple example of a hyper-parameter, imagine in the context of Figure 3 the functions used were polynomials with different degrees. In this case, the weights of each polynomial term would be a parameter, but the degree of the polynomial is a hyper-parameter that needs to be chosen before training.

More complex models tend to have a lot more hyperparameters. XGBoost, for example, presents dozens of them that can be tweaked in the best manner to avoid overfitting but also ensure that the model is complex enough to fit the data. In XGBoost, as discussed in section 3.4, what is learned during the training phase is the features that need to be used in each split and the split points, but things like the learning rate, the shrinkage parameter or how the rows and columns are sub-sampled need to be determined beforehand.

3.2 Regression Trees

3.2.1 Definition

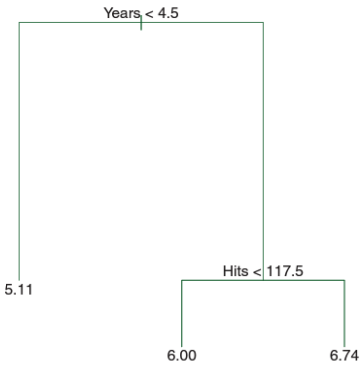
A Regression Tree is one of the simplest Tree-based methods. Let us first describe a simple example of how a Regression tree works. Suppose one wants to predict the salary of baseball players with two features: the experience of the player (the number of years he has played in the league) and the number of hits he made in the previous year. As a quick note, the output is actually the logarithm (log) salary since the data was log-transformed as a data-preprocessing step (JAMES et al., 2013).

One would imagine that the more experienced the player and the more hits he made last year, the higher his salary. Indeed, that is seen in Figure 4 (a). The tree is a sequence of splitting rules that start at the top of the tree. For example, the top split separates players with less than 4.5 years of experience to the left branch and players with more

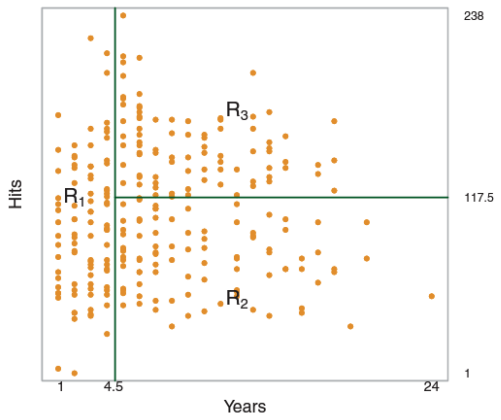
than 4.5 years of experience to the right. Then, the prediction for those on the left branch is simply the mean log salary (5.11) for the players with less than 4.5 years of experience since there is no other split on the left. However, for more experienced players, there is a second split. If the player had more than 117.5 hits in the last season, the log salary prediction would be 6.74, higher than the 6.00 predicted for the less-performing players.

Figure 4: Simple example of Regression Tree for the Baseball Salary Prediction Problem.

(a) Regression Tree for the Baseball Salary Prediction Problem



(b) The partition of the feature space made by the regression tree in (a)



Source: Extracted from (JAMES et al., 2013).

The structure formed on the left is called a tree because of its analogy with a real tree, which has branches and leaves. In the Regression Tree, one calls node the points where the splits are made, separating the data into two branches. Finally, the leaves are the terminal nodes or the nodes that don't have any more splitting after them (HASTIE; TIBSHIRANI; FRIEDMAN, 2001)(the points with the predictions 5.11, 6.00, and 6.74 here). Finally, one may note that all the splits made are "binary" splits, i.e., only two branches come out of each node.

Finally, Figure 4 (b) shows a representation of how the splits divide the feature space (the vector space with two variables, the number of years and the number of hits). The vertical line corresponds to the first split, separating the space in one region R_1 (players with less than 4.5 years) and the right region, which will then be separated in R_2 and R_3 by the second split on the right branch. It is worth noticing that all the orange points (that represent the different players) that fall into the same region R_m will then have the same exact prediction that is denoted c_m. The value of c_m will typically be the average salary of all the players in that region (e.g., 5.11 for all players in R_1).

The main idea is thus to divide the feature space into non-overlapping regions R_ms, and, for each region, the same prediction c_ms will be made. Mathematically, this can be represented by (HASTIE; TIBSHIRANI; FRIEDMAN, 2001):

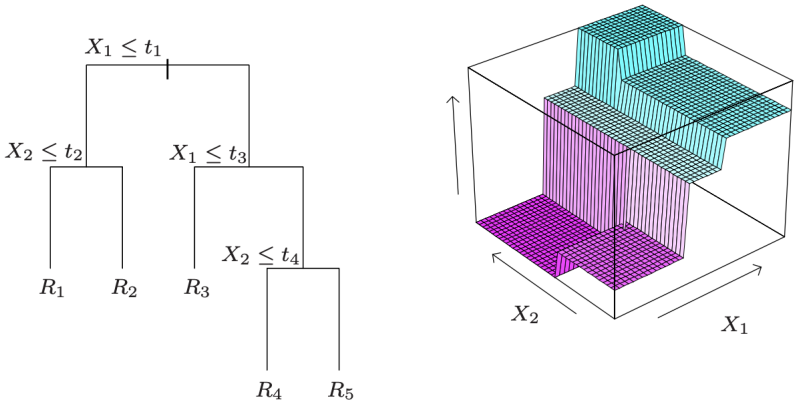
$$\hat{f}(x) = \sum_{m=1}^{M} c_m \, I\{x \in R_m\} \tag{3.5}$$

where I represents the indicator function, i.e. it is 1 when $x \in R_m$ and 0 otherwise. Then, our prediction will be:

$$\hat{c}_m = Ave(y_i \mid x_i \in R_m) \tag{3.6}$$

As one can notice, the prediction here follows the idea of estimating the regression function ($\mathbb{E}(Y \mid X = x)$, c.f. A.2). One can see a graphic representation of a generalized tree with four different splits and five different regions in Figure 5, where the X_i represent each of the model's features (e.g., number of hits made last year and years of experience). The $\hat{f}$ generated by the regression tree is represented on the right, and it is constant in all the regions R_m, each with the value c_m.

Figure 5: Example a partition of a two-dimensional space, in a Regression Tree.



Source: Extracted from (HASTIE; TIBSHIRANI; FRIEDMAN, 2001)

3.2.2 Finding the best split

However, computing the best binary partition (a non-overlapping division of the feature space) between all the possible choices is generally computationally infeasible because there is an enormous number of possibilities. Thus, it is necessary to solve this problem in a greedy approach, i.e., the heuristic of making the locally optimal choice at each iteration is used (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). At each step, one will try to find the best splitting variable j and the best split point s to make our split, i.e., which binary split would result in the best prediction. Mathematically, one wants to split into two regions, R_1 and R_2, defined as

$$R_1(j, s) = \{X \mid X_j \leq s\} \quad \text{and} \quad R_2(j, s) = \{X \mid X_j > s\} \tag{3.7}$$

To determine the best splitting variable and best split point, one wants to solve the following optimization problem (JAMES et al., 2013)

$$\min_{j,s} \sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{3.8}$$

where $\hat{y}_{R_1} = \hat{c}_1$ and $\hat{y}_{R_2} = \hat{c}_2$, i.e., the average of the training observations that in the respective regions. Therefore, at each step, this optimization problem will be solved in a greedy manner (i.e., optimizing locally), which is much simpler than minimizing the mean squared error considering all the possible partitions of the feature space.

3.2.3 Determining the size of the tree via penalization

However, suppose one continues splitting the feature space until the end. In that case, one will get to a point where each training observation will be in one region, and one will have zero training error (if there are not the same features with different predictions). For example, imagine that in Figure 4, one would divide the feature space so that each player (each orange point) fell in a different region. It would be necessary to create a vast, complex tree. Then, in this case, the data would be overfitted, having a model that will have a considerable variance and will not perform well on the test data set (c.f. subsection 3.1.3). Therefore, the growth of the tree must be limited, i.e., the number of splits that will be performed.

One technique used to limit the tree's growth is called Tree Pruning, which consists

of growing a huge tree T_0 and then pruning it back to obtain a subtree. A subtree would be a smaller tree with fewer splits than the original tree. For example, in our Figure 4, a subtree would be the tree with only one split (Years < 4.5).

To select a smaller subset of trees to consider, one can use what is called cost complexity pruning (JAMES et al., 2013), which will help us find a subsequence of subtrees indexed by a non-negative parameter α. For each α, one can find a subtree $T_\alpha \subset T_0$ such that

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T| \tag{3.9}$$

is minimized. Recall that $|T|$ is the number of terminal nodes or leaves in a tree. For example in Figure 5, $|T| = 5$. The term $\alpha|T|$ is extremely relevant here since it represents a key concept in Machine Learning called Regularization. The main idea here is that α controls the trade-off between the training error and the tree complexity. If $\alpha = 0$, one will choose the tree that best fits the training data, i.e., T_0, the (complete) most complex tree. However, if α is big, one will tend to pick a simpler tree (with fewer leaves) since a lot of weight is put in the term $|T|$ and (equation 3.9) is trying to be minimized (JAMES et al., 2013).

After having a sequence of subtrees (obtained by using the (equation 3.9), one could estimate the test error using some technique (e.g., validation dataset or cross-validation, c.f. 6.3) and find the best value for α, i.e., the tree indexed by α that will get the smaller test error, after obtaining the best α, one can go back to the complete data and get the subtree T_α that minimized (equation 3.9) (JAMES et al., 2013). This completes our mission of obtaining a tree with a good balance between bias and variance that should thus perform relatively well in a test data set.

However, while trees are useful because of their interpretability, they generally perform poorly. Nonetheless, by cleverly aggregating many regression trees, it is possible to get outstanding results. Here, let us focus on a particular way of combining these trees, called Boosting, the basis of XGBoost, the model used in this study.

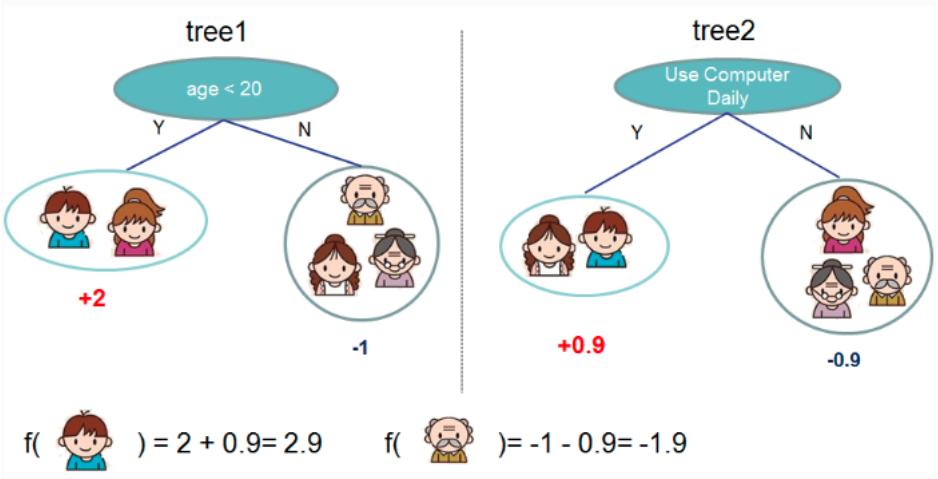
3.3 Boosting and Gradient Boosting

3.3.1 Definition of Boosting

As mentioned in the previous section, Boosting is a general approach that can improve our predictions. It is one of the most important and influential ideas in Machine Learning in the last decades, which is based on the idea of building a model using many "weak" predictors combined to form a powerful "committee" (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). Although it is a method that can be applied to many different models, let us focus on its applications to Regression Trees.

The main idea of boosting is to grow trees sequentially, and each tree will be grown with the information from the previous trees (JAMES et al., 2013). In the end, one wants to fit an additive model, i.e., an ensemble of trees of the form $\sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$, where $\mathcal{F}$ is the space of regression trees (CHEN; GUESTRIN, 2016b). A simple example of how these trees work can be found in Figure 6, where one can see an ensemble of two trees that try to predict whether someone will like a computer game. One can see that having an age below 20 years and using a daily computer increase the probability of someone liking a computer game. However, in contrast with Figure 4, here there are two trees, and the way to combine them is to sum the predictions. Indeed, one may notice that the prediction for both the boy and the older man is the sum of each tree's predictions.

Figure 6: Tree Ensemble Model: the final prediction is the sum of predictions from each tree.



Source: Extracted from (CHEN; GUESTRIN, 2016b)

3.3.2 Intuition: fitting in Boosting

Moreover, one will fit this model in a stage-wise manner, introducing in each stage weak learners that compensate for the shortcomings of the existing ensemble, which are identified using gradients (LI, 2022).Therefore, at each step t, the prediction would be (CHEN; GUESTRIN, 2016a):

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned} \tag{3.10}$$

As one can see, at each step, the model gains one more tree f_k that contributes to the final model. At the last iteration, the prediction will be the sum of every tree generated in each of the t iterations. Hence, at each step t, one will choose the regression tree $f_t(x_i)$ that will improve the quality of our prediction at most. Let us understand the idea intuitively, and a more formal approach for gradient boosting can be found in B. Let us denote $F(x) = \sum_{k=1}^{t-1} f_k(x)$, i.e., the model at step t, and $h(x) = f_t(x)$, i.e., the regression tree to be added to the ensemble in this step. Intuitively, let us improve the model such that (LI, 2022):

$$F(x_i) + h(x_i) = y_i, \quad \forall i \in \{1, 2, ..., n\} \tag{3.11}$$

since, $F(x_i) + h(x_i)$ will be the new model after our ith iteration, and it should hopefully accurately predict y_i. Equivalently, one could write

$$h(x_i) = y_i - F(x_i), \quad \forall i \in \{1, 2, ..., n\} \tag{3.12}$$

In order to obtain this for $h(x)$, one may train the regression tree to fit the data $(x_1, y_1 - F(x_1))$, $(x_2, y_2 - F(x_2))$, ... , $(x_n, y_2 - F(x_n))$, since one wants $h(x_i)$ to output exactly $y_i - F(x_i)$. This means one can fit the model to the residuals, i.e., to $y_i - F(x_i)$, seeking to compensate for the shortcomings of the existing model at t (LI, 2022). Then, the ensemble will be grown a total of M times, being M a parameter that needs to be tuned by balancing bias and variance (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

3.3.3 Intuition: Gradient Boosting

The idea of the gradient is directly related to this very intuitive idea of fitting the residuals. Let us consider the loss $L(y, F(x)) = (y - F(x))^2/2$ and the objective to minimize $J = \sum_i L(y_i, F(x_i))$. Since the goal is to reduce the loss, one may take a step in the direction of the gradient, i.e., $\frac{\partial J}{\partial F(x_i)}$. This is a widely known technique called gradient descent, where the idea is to minimize a function by moving in the opposite direction of the gradient, using a learning rate ρ, such as $\theta_i \coloneqq \theta_i - \rho\frac{\partial J}{\partial \theta_i}$. However, when one computes the gradient, it is possible to notice that it is equal to the negative residual (LI, 2022)

$$\frac{\partial J}{\partial F(x_i)} = F(x_i) - y_i = -residual_i \tag{3.13}$$

Therefore,

$$\begin{aligned} F(x_i) &\coloneqq F(x_i) + h(x_i) \\ F(x_i) &\coloneqq F(x_i) + residual_i \\ F(x_i) &\coloneqq F(x_i) - 1\frac{\partial J}{\partial F(x_i)} \end{aligned} \tag{3.14}$$

Hence, updating F based on residual is equivalent to updating F based on the negative gradient. The benefit of using this definition is that other losses can be chosen, such as the L_1 loss or the Huber loss, which is quadratic for small values, but linear for large values, which makes it more robust when dealing with outliers (LI, 2022).

A more formal approach to fitting Gradient Boosting may be found in B.

3.4 XGBoost

As mentioned in the introduction of this section, XGBoost is a state-of-the-art algorithm for Boosting and Supervised Machine Learning in general, typically the top choice for ML competitions. XGBoost stands for "Extreme Gradient Boosting" and, as the name suggests, is an improvement on the classical gradient Boosting algorithm, adding several new features that help increase the performance of the model, making it an extremely scalable solution and running ten times faster than existing popular solutions (CHEN; GUESTRIN, 2016b).

Let us now present some of its significant additions to the original Gradient Boosting Model, all described in (CHEN; GUESTRIN, 2016b). The improvements that are less relevant to this study and somehow more technical are described in C.

3.4.1 Regularized Objective Function

As may be seen in B, the goal is to minimize at each time a tree is being created a quantity $J^{(t)}$ that can be interpreted as minimizing the prediction error. However, it did not involve a regularization term (c.f. subsection 3.2.3) that controls the complexity of the tree is added. In the XGBoost, they have added these adjustments, regularizing the objective function with a term $\Omega(f_k)$. Using the same notation as before, where T is the number of leaves and c a vector with all the predictions c_m for the different regions, one may define the following penalization.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|c\|^2 = \gamma T + \frac{1}{2}\lambda \sum_{m=1}^{M} c_m^2 \tag{3.15}$$

where γ controls the strength of penalization of the size of the tree (the number of splits that are going to be made) and λ the strength of penalization of the size of the c_m, which helps smooth the final learned weights, avoid overfitting (CHEN; GUESTRIN, 2016b).

3.4.2 Shrinkage

Shrinkage is yet another type of regularization strategy. The idea is to control the boosting procedure's learning rate, scaling each tree's contribution by a factor of ν, between 0 and 1 (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). One would then have

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \nu f_t(x_i) \tag{3.16}$$

The goal of this technique is to leave space for future trees to improve the model by reducing the influence of each tree (CHEN; GUESTRIN, 2016b). Empirically, it has been found that smaller values (≈ 0.1) favor better test errors (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

3.4.3 Column and Row sub-sampling

Row sub-sampling means that one samples a fraction of the dataset at each iteration and grows the next tree only using this sample (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). It is a widely known technique in other tree-based ensemble methods, such as the Random Forest, a famous tree ensemble algorithm that uses bagging (trees grown in parallel, not sequentially) (CHEN; GUESTRIN, 2016b). Not only does it decrease the computing time, but it also helps reduce overfitting, producing a potentially more accurate model.

On the other hand, column sub-sampling means that at each iteration, the trees are grown using only a (random) subset of the columns (features), another technique also used in Random Forest. According to user feedback, this technique helps even more in preventing overfitting than the row sub-sampling, which is more traditional (CHEN; GUESTRIN, 2016b).

These two features are not necessarily implemented in the traditional gradient boosting but bring great results, enhancing the model's performance.

3.4.4 Sparsity-aware Split Finding

Another great feature XGBoost presents is its ability to handle sparse data. There are some different reasons why the data might be sparse: (i) missing values, (ii) frequent zero entries statistics, (iii) feature engineering techniques such as one-hot encoding (CHEN; GUESTRIN, 2016b).

XGBoost will handle all types of sparsity in a unified way. The approach here is to classify the instances in a default direction that will be chosen from data, with two options of default directions per branch (right and left). It is also worth mentioning that this sparsity-aware algorithm is much faster (up to 50 times) than the naive approaches (CHEN; GUESTRIN, 2016b).

PART III

METHODOLOGY

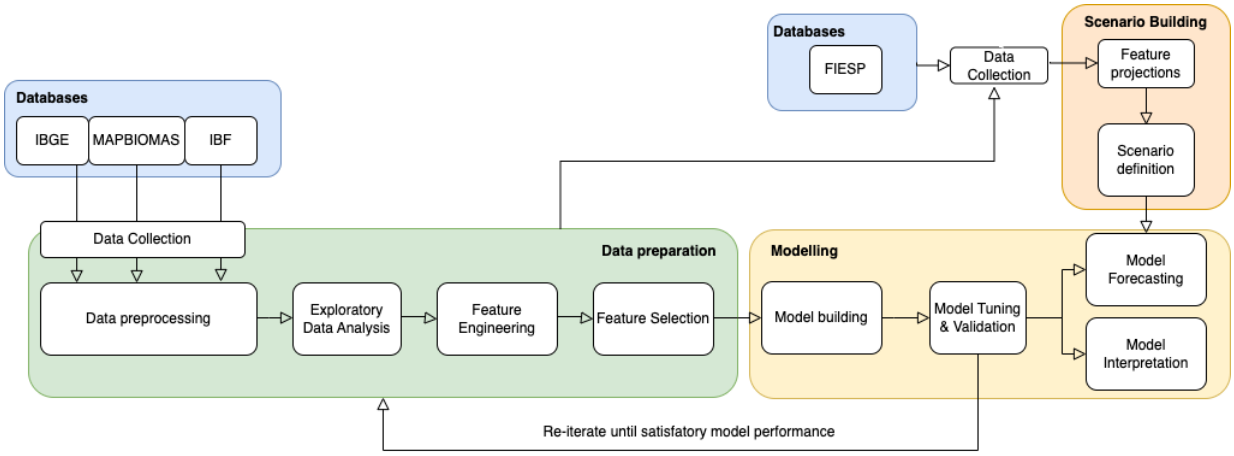
Having revisited the literature and having a good understanding of the concepts of Machine Learning and Tree-based statistical learning, as well as all the nuances of the Pantanal, it is time to structure the methodology to solve the problem.

The goal is to predict deforestation in different scenarios: a realistic one, an optimistic one, and a pessimistic one (from the environmental perspective). To make these different scenarios, different land-use decisions will be selected, i.e., the number of heads of cattle and tons of crops (the main ones being Corn, Sugarcane, and Soybeans) produced. This decision was based on articles in Chapter 1, which shows that one of the main drivers of deforestation in and around the Pantanal was agriculture and cattle ranching. Moreover, information about the Pantanal fires is needed since they are also responsible for a relevant part of the deforestation in Pantanal (MAPBIOMAS, 2021). On top of that, data from the most relevant environmental laws that were implemented and data from the Natural and Total Areas of the municipality have been collected, which will help our model predict deforestation more accurately.

Still, it is worth remembering that our focus is on a global view of deforestation and how it relates to agriculture, cattle ranching, and fires. So let us first explore the structure of our coding environment, collect the data used in the model, explore this data, create and select the features, and then move to the modeling section.

Figure 7 shows a global representation of the methodology. As one may see, there are three main blocks: (i) the Data Preparation, (ii) the Modelling, and (iii) the Scenario Building, all represented by different colors (green, yellow, and orange, respectively). Each of these building blocks of the methodology will be a different chapter of this Methodology part. Moreover, each step within the building blocks will be represented as different sections. Moreover, two additional sections have been added: (i) the code structure of the project that will give more clarity on how the project was built; (ii) the exploratory data analysis, which will help understand the problem.

Figure 7: Diagram with the main steps in the methodology.



Source: The author.

One may notice that the Data preparation part has more steps. Indeed, in Machine Learning projects, data preparation is one of the most important (and time-consuming) parts. Moreover, there is a loop between the modeling phase and the Data preparation. This loop is essential since building a predictive model relies heavily on testing and iterating over a simple model, which will then be complexified iteration after iteration.

However, before diving into each building block, let us first check how these building blocks were structured in our code base.

4 CODE STRUCTURE

It is essential to define how the code environment will be structured since the secondary objective for the project was to have a robust and scalable code base such that others can contribute to the project. In the long term, this could lead to increased reach and impact of the project.

To make our code scalable, organized, and quickly improvable, a modularized project structure was used, inspired by the project structure used during my internship in BCG GAMMA (part of BCG X), Boston Consulting Group's Data Science area. First, it is worth noting that all the code used in this project is available in ⟨https://github.com/iglesiascaio/pantanal_deforestation⟩.

Figure 8 shows a general view of the code structure. In general terms, the code is divided into three main folders: (i) data, where the data is stored in each step of the data preparation; (ii) runner, where the heavy primary data preprocessing and feature engineering work is done; (iii) notebooks, where most of the modeling and data analysis work is done. It is worth noting that a notebook (or Jupyter notebook) is a type of file frequently used in the Machine Learning field since it allows combining code, text, and data visualizations in the same document. In Figure 8, each of these folders is represented by a colored block.

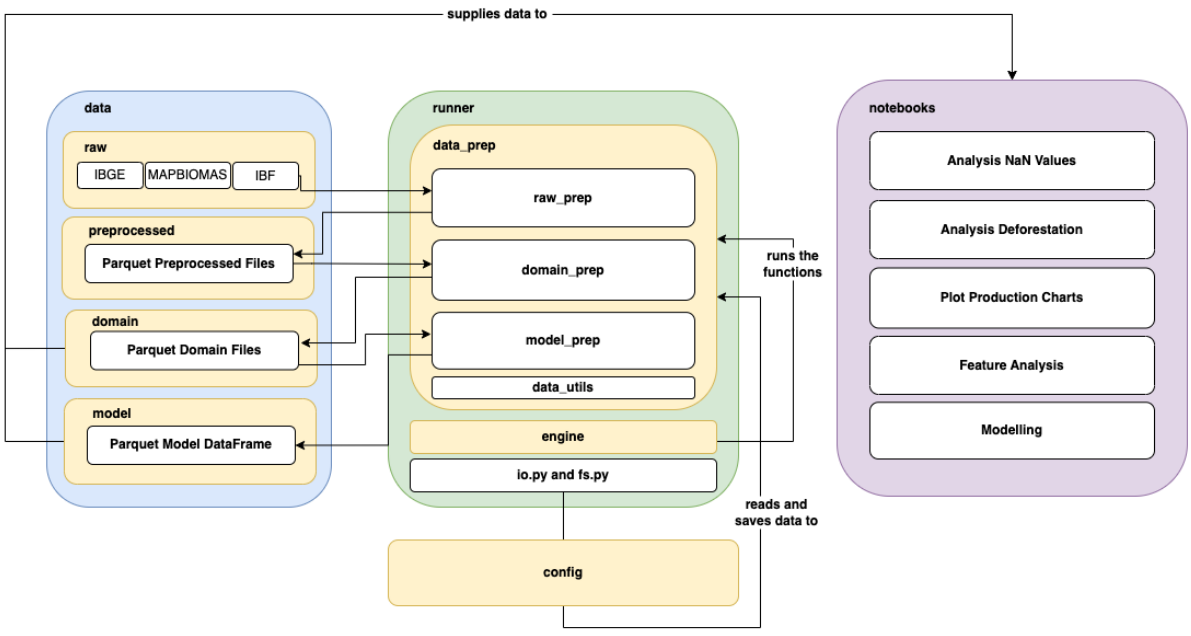
The idea of the pipeline is to divide each data processing step to isolate the modules, such that if there is a need to change the data or some preprocessing step, it does not affect the code globally. To do that the main idea is to divide the data processing steps into three: (i) the raw data preprocessing, (ii) the domain preparation, and (iii) the model data preparation.

The first step, shown as a folder called `raw_prep` in the diagram (Figure 8), is responsible for reading all the raw data (inside the `raw` folder in the data block) from the different Excel files of each source (MAPBIOMAS, IBGE, and IBF, as will be seen in Chapter 5). After that, the data is transformed into a standardized format, and the use-

less columns and rows for our problem are filtered out. Then, the preprocessed data is saved in the `preprocessed` folder in the data block in a standard format for tabular data called `parquet`, as shown in the diagram.

The second step, the domain preparation, is responsible for getting each preprocessed file from the `preprocessed` folder and transforming it in the correct format so it can be inputted in the modeling phase. Therefore, the data is transformed in a more significant manner, with operations that, for example, transform the data from a "wide" format (e.g., the value of the years on the columns) into a "long" format (e.g., a column named years and the values in the rows), that is more suitable to Machine Learning Models. Afterward, the domain data is saved into the `domain` folder in the standard `parquet` format. Usually, to plot charts and analyze the data in the notebooks, the `domain` version of the data is used since the data is already in an appropriate format, and they are still separated into individual files (e.g., one for deforestation, one for agricultural production, and so on).

Figure 8: Diagram with an overview of the code structure of the project.



Source: The author.

These first two steps (raw preprocessing and domain preprocessing) are both part of the Data preprocessing step, as shown in our methodology diagram (Figure 7). However, the third step, i.e., the model data preparation, falls in the Feature Engineering step since its main job is to create the final features that go into the modeling. Therefore, in this step, the `domain` data will be taken as input, and all the different files will be joined into

one table containing all the information. After that, the features are created, and the data is ready to enter the modeling phase. Finally, the modeling phase is entirely done inside the notebook called `model.ipynb` inside the `notebooks` folder.

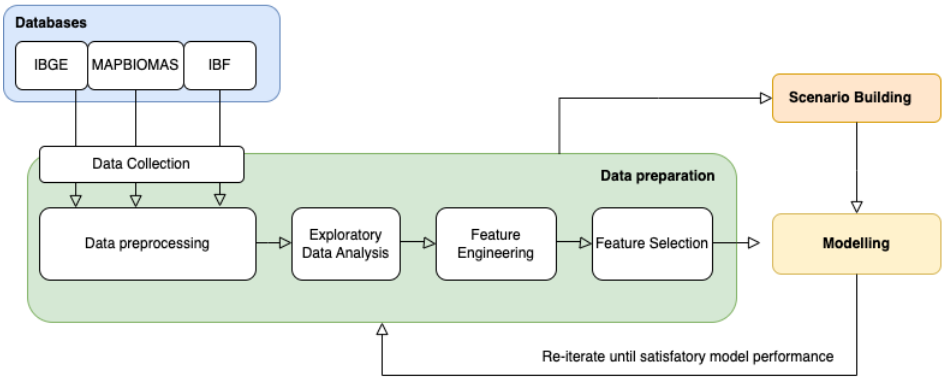
Finally, the role of other important files and folders in Figure 8 is worth noting. Firstly, the engine folder makes a simple code structure that allows running the files. For example, it creates a command that can be efficiently run in the terminal (a text-based interface to the computer) that makes all the data processing steps at once. On the other hand, the `io.py`, `fs.py` files and the `config` folder create a straightforward way of reading and saving files without the need to specify the all the time the path (where the data is in the computer), which facilitates the data processing step.

Note that this is a simplified view of the structure of the code, but it gives an overview of how it works. More detail can be found at D, where the structure with all the files is shown, as well as details of how to run some commands, which may be helpful for those who are interested in contributing to the project.

5 DATA PREPARATION

As mentioned above, Data preparation is often the most critical and time-consuming part of a Machine Learning Projects (GERON, 2017). As shown in Figure 9, the Data preparation chapter is divided into five parts: (i) Data Collection, where the sources of the data and the databases collected are going to be detailed; (ii) Data preprocessing; where the data will be treated and transformed in an appropriate format; (iii) Exploratory Data Analysis, where charts are plotted to understand the data; (iv) Feature Engineering, where the features elaborated are going to be shown; (v) where the techniques for selecting most relevant features and the features selected are going to be shown. Moreover, it is worth noting that the Data preparation part connects to both the modeling and scenario-building parts by providing data used in both.

Figure 9: Methodology Diagram with focus on the Data preparation.



Source: The author.

In a nutshell, the principal goal of this chapter is to describe the main steps from collecting the data to transforming it into valuable features that will help the model predict deforestation as accurately as possible. Moreover, referencing Chapter 4, the Data Preparation steps described here were mainly done inside the `runner` folder, while saving auxiliary files in `data`.

5.1 Data collection

To build the model, the data needs first to be collected. Since the model will predict the Pantanal's deforestation and its surroundings for different agricultural/cattle decision scenarios, data from deforestation, annual agricultural production, and annual livestock headcount was needed. Moreover, after our research on the Pantanal, some other important factors that could help us predict deforestation have emerged: the environmental laws and the fire data.

Therefore, the data was taken from three primary sources: the Brazilian Institute of Geography and Statistics (IBGE), MAPBIOMAS, and the Brazilian Forest Institute (IBF). The data obtained from each of the databases and the frequency collection of the data can be found in Table 4.

Table 4: Databases and data collected from each of them, with which frequency)

Databases	Frequency
IBGE	
Number of heads of cattle produced for each municipality	per year
Number of tons produced by crop for each municipality	per year
MAPBIOMAS	
Total area of each municipality	-
Natural area of each municipality	per year
Natural area burned for each municipality	per year
Deforestation area for each municipality	per year
IBF	
Main Environmental laws implemented in Brazil	-

Source: The author.

All these collected data will then be transformed into features for our model, as detailed in section 5.4.

5.2 Data preprocessing

5.2.1 IBGE data

As mentioned in Table 4, two main types of data were collected from the IBGE database: (i) the number of heads of cattle produced for each municipality (per year); (ii) the number of tons produced by crop for each municipality (per year)

There were six main preprocessing steps for the IBGE databases. The first step was to filter only the main municipalities that are part of Pantanal, as seen in Table 1. After that, since the data came in a "pivoted" format (i.e., with the years and crops in columns instead of rows), transforming this data from a "wide" to a "long" format was needed since the latter is more appropriate for modeling purposes. The third step was to treat undefined values, considering some as zeros and some as NaNs, depending on the type of undefined value. The fourth step was to filter only the most relevant crops using a Pareto Analysis done in (BATTI, 2020). Thus, the crops selected were Cotton, Rice, Coffee, Sugarcane, Beans, Oranges, Cassava, Corn, Soybeans, and Wheat. Then, since all the data came in different files, a joint DataFrame (i.e., table) was created, containing all the data collected and preprocessed.

Finally, the last step included assessing the data quality by understanding the patterns of the NaN values. Although a reasonable percentage of the variables - especially for the area produced by the crops - were missing (around 12.8% of the values), most were concentrated in the early years, and a few crops and municipalities. This means that one may want to use the quantity produced rather than the area for the modeling since it contains fewer NaN values and might be more insightful when interpreting the model. However, since the XGBoost can handle NaN values natively (c.f. section 3.4), there is no need to worry about how to treat these values. Furthermore, more details about the treatment of NaN values for the IBGE can be found in E.

5.2.2 MAPBIOMAS data

As mentioned in Table 4, four types of data from the MAPBIOMAS database were collected: (i) the Total area of each municipality, (ii) the Natural area of each municipality (per year), (iii) Natural area burned for each municipality (per year); (iv) Deforestation area for each municipality (per year).

To get this data, two different databases from MAPBIOMAS were collected. The

first is the Cover and Transitions database for Municipalities (Collection 7) with data from 1985-2021, and the second is the Fire Scars database (Collection 1) with data from 1985-2020. In contrast to the IBGE database, there is no need to worry about NaN values here since they do not appear.

The preprocessing for the MAPBIOMAS data was more straightforward, with only three steps. The first two steps were similar to steps one and two of the IBGE database, i.e., filter the municipalities of interest and transform the data from a "wide" into a "long" format. The third step involves the computation of the features (Total area, Natural Area, and Burned Area) and the computation of deforestation, the target variable of the problem. Thus, the third step (the computation) will be split in two, one for the features and one for the target variable.

5.2.2.1 Computing the Total, Natural, and Burned Area

The aforementioned Land Cover table is a table that shows the area for each different type of land use. The Total Area for each municipality by year is simply the sum of all the Land Covers for each City and Year. Likewise, the Natural Area for each municipality by year is the sum of all Natural Land Cover (level_0 = Natural) for each City and Year. Regarding the third piece of data collected, the Natural area burned for each municipality by year can be computed by summing all the burned area for each City and Year, filtering the Natural Area (level_0 = Natural).

The rationale behind getting the Total Area as a feature of the model is that a more extensive municipality tends to have more deforestation simply because it has more areas that could be deforested. Moreover, getting the Natural Area and its evolution in time might be helpful for the model to learn that as the Natural Area decreases, fewer potential areas will be deforested. Also, these variables can help the model differentiate one municipality from another. On the other hand, the Burned Natural Areas are helpful since it is a direct cause of deforestation, and it does not necessarily translate immediately into an increase in agricultural production.

5.2.2.2 Computing the deforestation

The deforestation area (in hectares) is our target variable, i.e., what the model will try to predict, which means it will be essential for constructing the model. To compute this variable, the Land Transitions data was used. This table shows the changes in Land Use that happen each year (e.g., forest area transformed into cattle ranching area).

Therefore, the approach was to take all transition data corresponding to a transition from a class in the Natural classes group into a level_0 Anthropic, since it means that some human activity has taken place in that area that transformed it from a Natural formation to an Anthropic one. The classes of type "Natural" can be found in Table 11.

5.2.3 IBF data

The last database used to collect features for the model is the IBF. As mentioned in Table 4, one type of data was collected from the IBF database: the Main Environmental laws implemented in Brazil.

In this case, the data did not come in a structured manner since it was presented in the text on their page on the Internet. Therefore, manually collecting the primary laws implemented during our study period, i.e., from 1985 until today, was necessary. The laws collected and a brief description of each can be found above, in order of their creation (IBF, 2020).

- **Agricultural Policy (1991)**: aims to protect the environment and establish the obligation to recover natural resources for companies exploiting the environment economically. It defines that the public authorities must supervise the rational use of the soil, water, fauna, and flora, develop environmental education programs, and encourage the production of seedlings of native species, among others.
- **National Water Resources Policy (1997)**: establishes the policy and the national system for water resources. It defines water as a limited natural resource with economic value that may have several uses, such as human consumption, energy production, transportation, sewage disposal, and others.
- **Environmental Crimes Law (1998)**: addresses criminal and administrative issues regarding actions that are harmful to the environment, granting environmental agencies mechanisms for punishing offenders, as in the case of environmental crimes committed by organizations.
- **National System of Nature Conservation Units (2000)**: aims the conservation of biological species varieties and genetic resources, preserve and restore the diversity of natural ecosystems, and promote sustainable development from natural resources.

- **New Brazilian Forest Code (2012)**: the preservation of native vegetation and revokes the Brazilian Forest Code of 1965, determining the responsibility of the owner of protected environments between the Permanent Preservation Area (APP) and the Legal Reserve (RL) to preserve and protect all ecosystems.

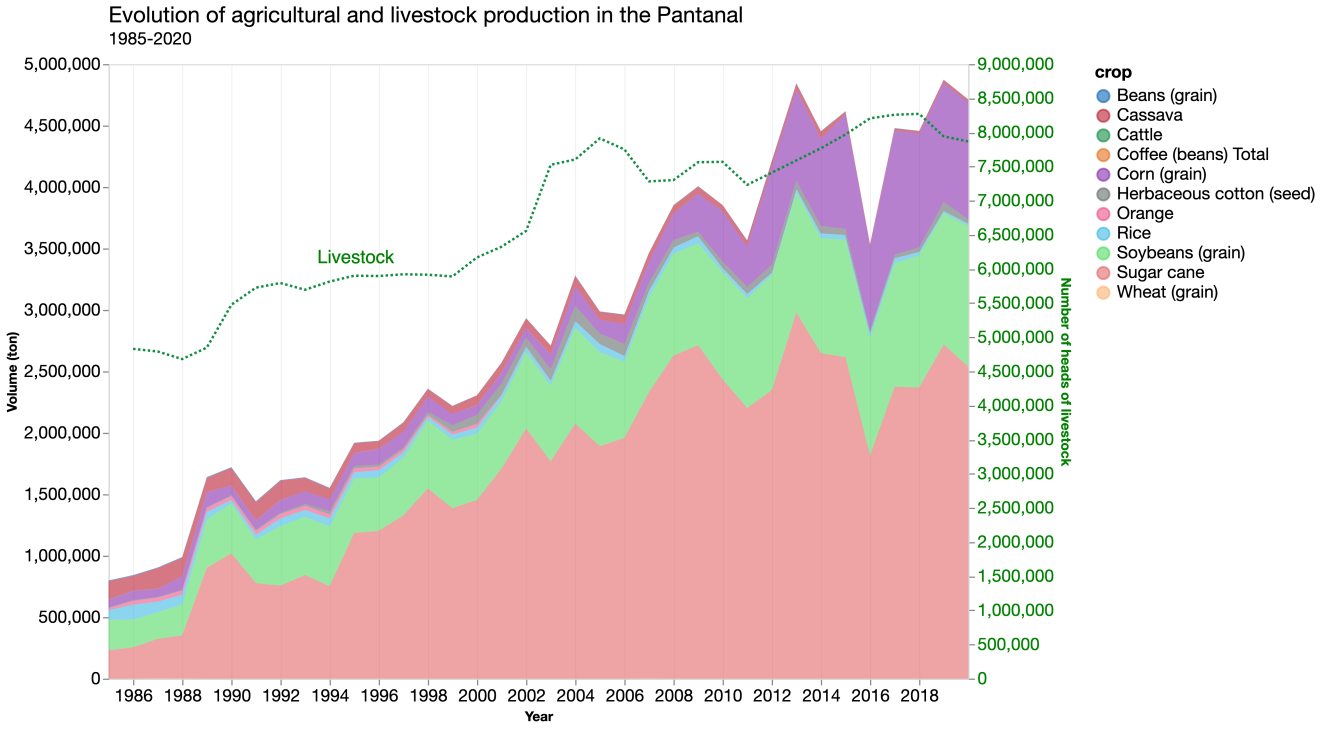
These five laws, according to IBF, were the primary environmental laws implemented between 1985 and 2022. Therefore, if they were effective, one would expect their implementation to reduce deforestation somehow. Let us explore how this information was converted into valuable data for the model in subsection 5.4.

5.3 Exploratory Data Analysis

After collecting and preprocessing our data, some Exploratory Data Analysis will be done to understand the data better.

Figure 10 shows the agricultural production of several crops in tons (left Y-axis) and the livestock headcount (right Y-axis) from 1985 to 2020 (X-axis).

Figure 10: Evolution of agricultural and livestock production in Pantanal (1985-2020).

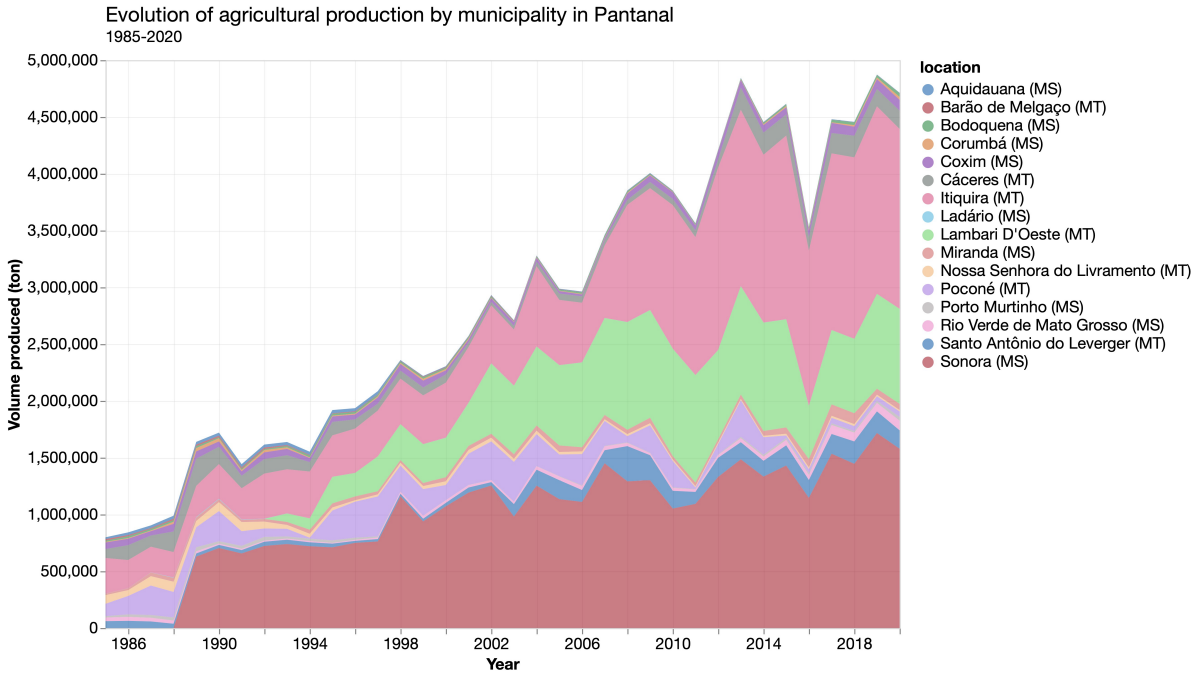


Source: The author.

Total agricultural production increased by more than 4.5 times over 35 years, reaching around 4.5 million tons in 2020. However, this growth seems quite linear, despite a slight decrease in overall production in 2016, mainly driven by a reduction in Sugar Cane production. Furthermore, it can be seen that the production is very concentrated in 3 main products: Corn, Soybeans, and Sugarcane. About cattle, it is possible to see that the livestock headcount has grown about 60%, now reaching about 8 million heads.

To understand how the total agricultural production is distributed among the different Pantanal municipalities, Figure 11 shows that the leading municipalities in terms of agricultural production are Sonora in Mato Grosso do Sul, Itiquira in Mato Grosso, and Lambari D'Oeste, also in Mato Grosso.

Figure 11: Evolution of agricultural production by municipality in Pantanal (livestock excluded) (1985-2020).

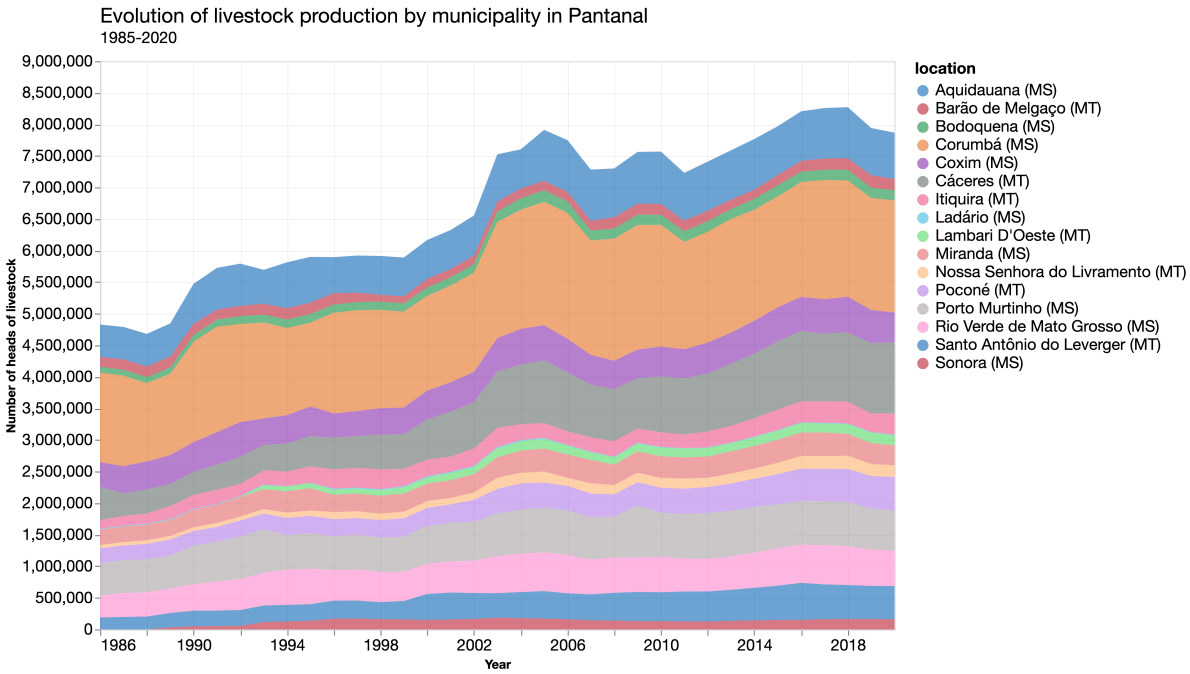


Source: The author.

Figure 12 shows how livestock production is distributed among the different Pantanal municipalities. Visually, it is possible to see that the municipalities with the most significant agricultural production are not the same as those with the most relevant livestock production. Moreover, notice that livestock production is more uniformly distributed between the municipalities than agricultural production and that most of the municipalities produce a relevant number of cattle heads, which was not the case for the production

of the different crops. Here, the leading municipalities in terms of cattle production are Aquidauana in Mato Grosso do Sul state, Corumbá, also in Mato Grosso do Sul state, and Cáceres in Mato Grosso state.

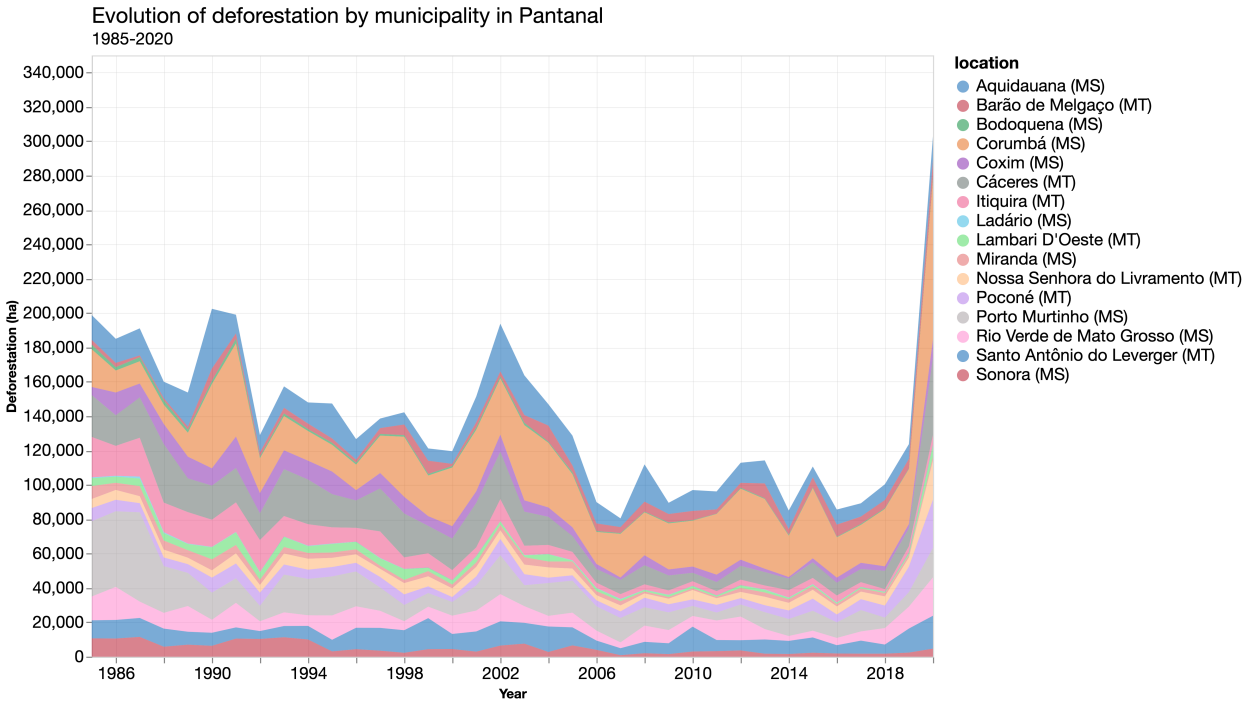
Figure 12: Evolution of livestock production by municipality in Pantanal (1985-2020).



Source: The author.

Having better understood how agricultural and cattle production growth in the Pantanal municipalities happened, let us now look at deforestation, which is what the model will predict. One can see in Figure 13 an evolution of deforestation by municipality. Firstly, one can see that the deforestation of the Pantanal municipalities is in the order of tens of thousands of hectares. Furthermore, deforestation decreased significantly until 2017, reaching less than 100000 ha, less than half of the deforestation in 1985. This is probably related to the municipalities having fewer and fewer areas to explore. Also, productivity by hectare gets more prominent with technological advancements.

Figure 13: Evolution of deforestation by municipality in Pantanal (1985-2020).

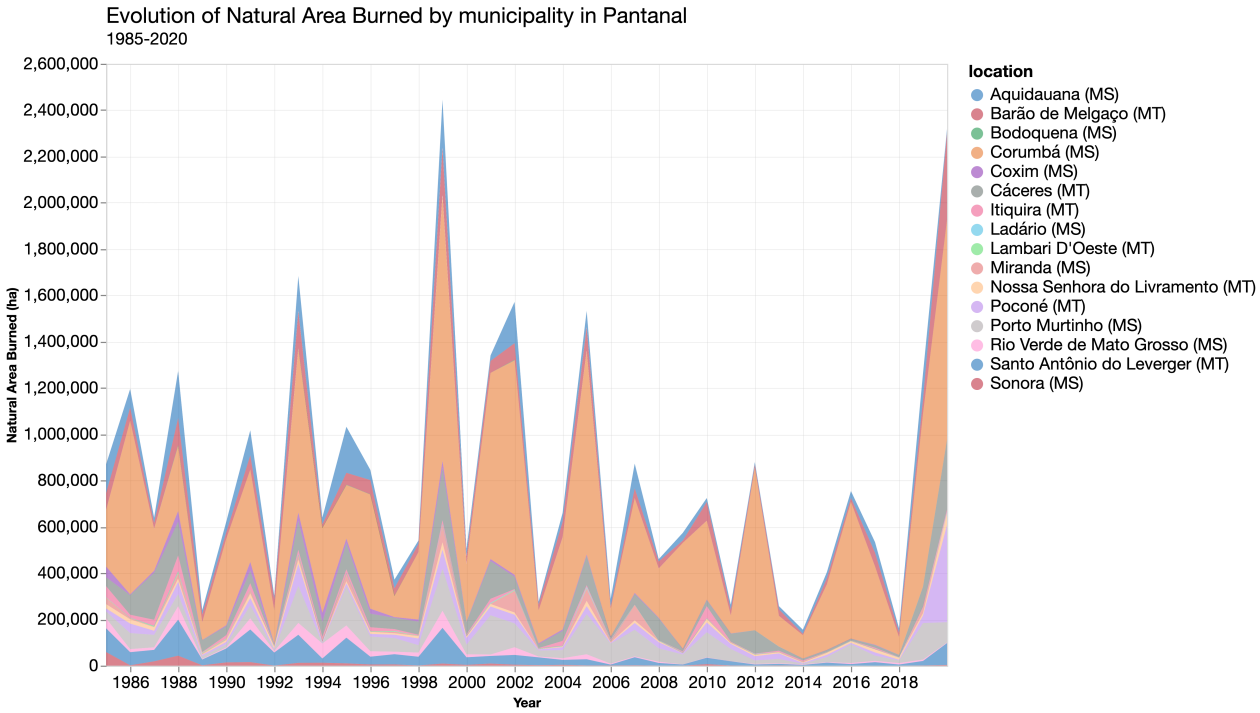


Source: The author.

However, in 2017 deforestation started to increase again, reaching almost 300000 ha in 2021, a level of deforestation never obtained before. As will be seen next, this is directly correlated to the fires in the region in the last few years. During those years, one can notice that the municipality of Corumbá, in Mato Grosso do Sul state was the one that oscillated the most in terms of deforestation, with some peak years. Fires most probably caused these peaks since, historically, the city is a leader in fires, and in 2010, for example, concentrated more than 40% of the fires in Mato Grosso do Sul (FERNANDES, 2010).

Moreover, Figure 14 suggests some correlation between deforestation and the area burned by the fires since some of the peaks in the graphs coincide, mainly the one after 2018. Furthermore, the predominance of the municipality of Corumbá in the burnings draws attention, which is one of the main ones in cattle head production. However, notice that the burned areas are even higher than the deforestation area, which suggests that not all of the fires are directly converted into some anthropic activity.

Figure 14: Evolution of Natural Area Burned by municipality in Pantanal (1985-2020).



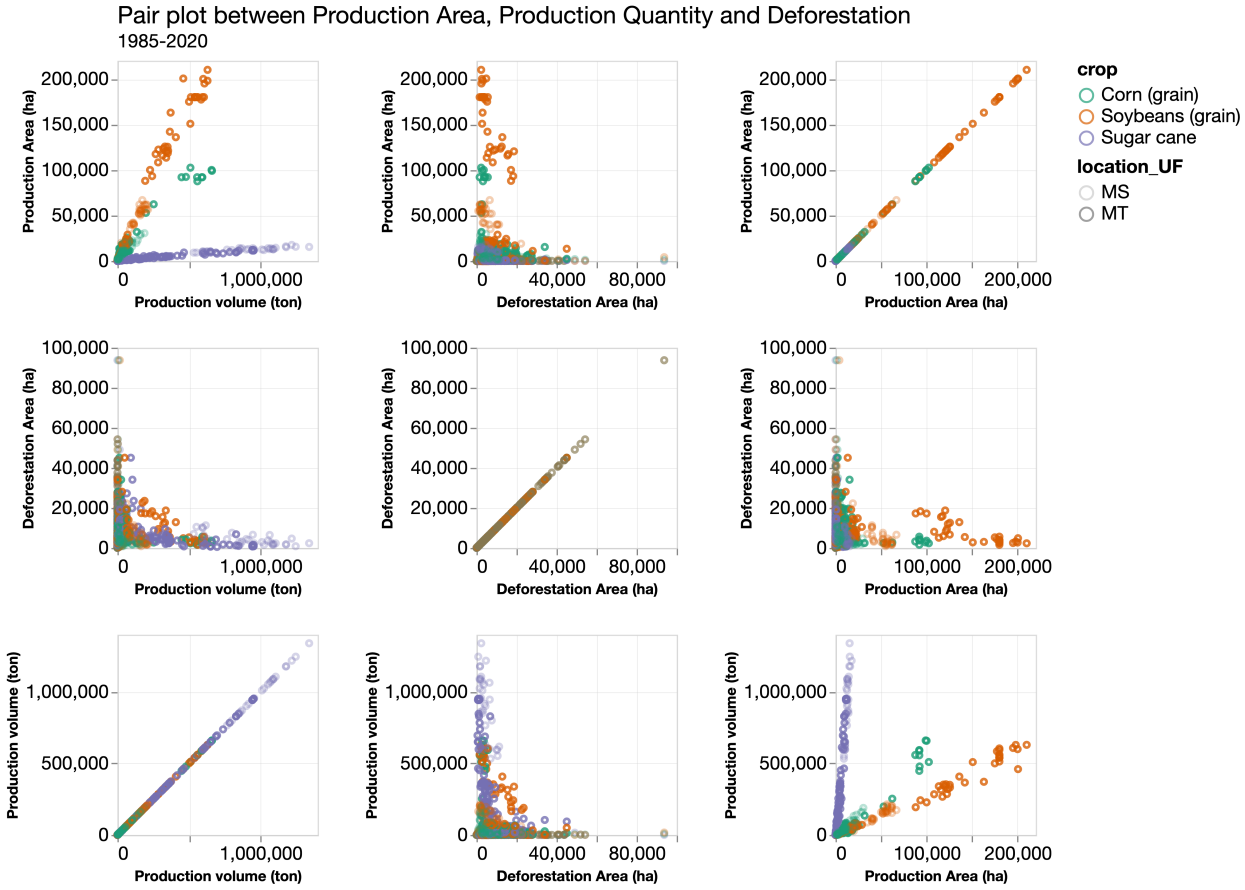
Source: The author.

After analyzing the data temporally, broken down by crops and municipalities, let us now understand the relationship between production, area, and deforestation variables, which are some of the main quantitative variables. Figure 15 shows a pair plot containing these three variables, focused on the three main crops identified in Figure 10. The first thing that stands out is the relationship between the production volume and quantity variables. As expected, for each crop, the relationship between quantity produced and area planted for each crop is linear. Furthermore, note that different crops have higher production areas than others by the angular coefficient of the line formed. For example, Sugarcane seems to be the crop with the highest yield per area, as it has the highest angular coefficient in the graph in the lower right corner. On the other hand, Soybeans have the lowest yield in terms of volume, as it needs much larger areas to produce the same volume.

However, the scatter plot between quantity produced/planted area and deforestation shows no clear relationship between the variables. This may seem counter-intuitive at first glance, but when looking at Figure 13, it starts making more sense: even though in 2019 deforestation escalated a lot, the amount produced hardly varied much in absolute terms from one year to the next. Therefore, agricultural production is not the primary

driver of deforestation in terms of area.

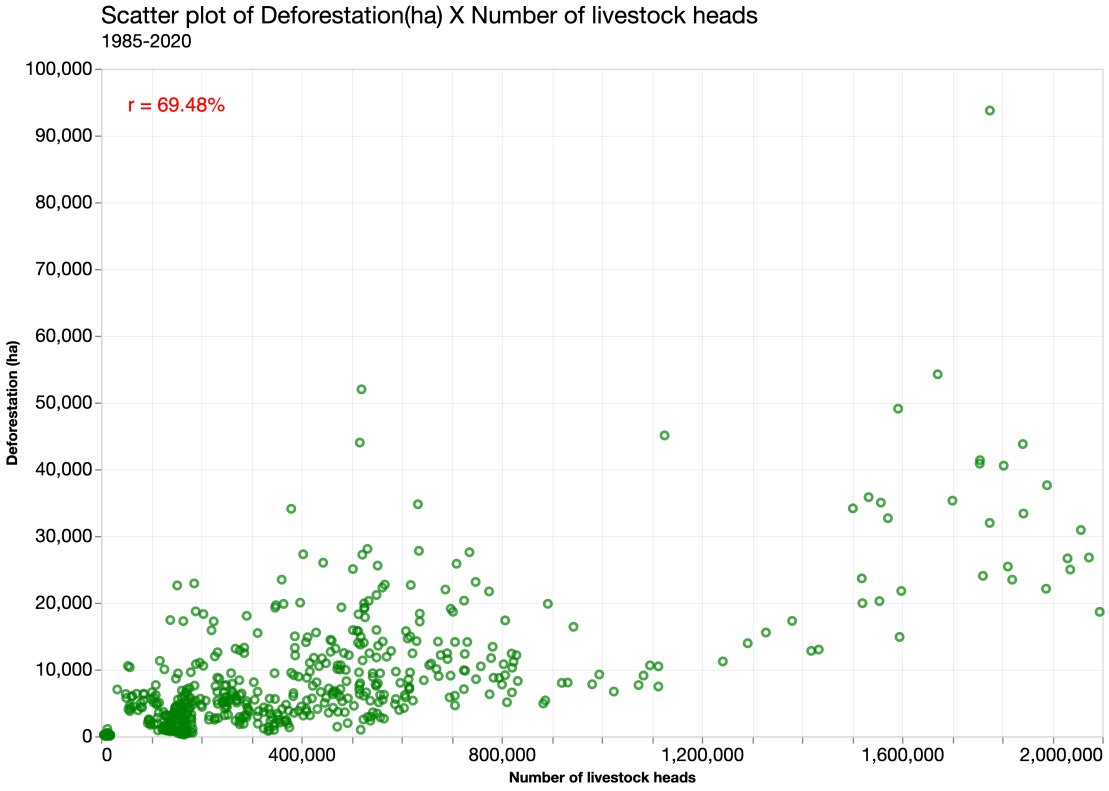
Figure 15: Pair plot between Production Area, Production Quantity, and Deforestation.



Source: The author.

On the other hand, Figure 16 indicates that the correlation between deforestation and livestock headcount is much stronger. Indeed, computing the correlation yields a value of 69.48%, indicating a strong positive correlation between the two variables. That indicates that cattle production will probably be a more relevant feature for our model than agricultural production, thus being a better predictor variable.

Figure 16: Scatter plot between Deforestation and Number of livestock heads.

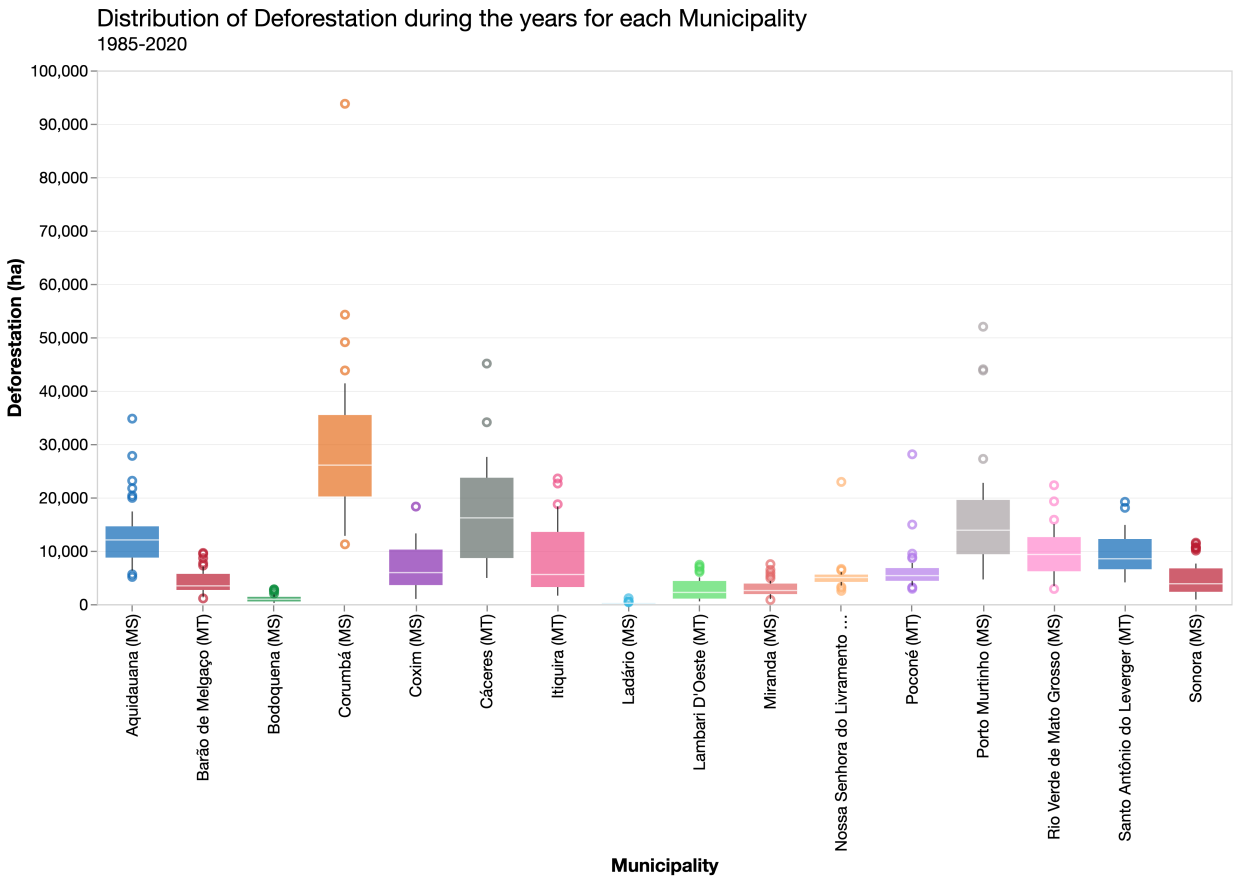


Source: The author.

After looking at the relationship between the variables, one may want to understand how deforestation was distributed over the years for each municipality. That is, one may like to know, for example, if the deforestation in the municipality of Corumbá stayed constant every year or if it varied a lot over the years. To do this, it is possible to use boxplots, which show, for each municipality, how deforestation was distributed. Remember here that a boxplot shows the median and the 25% and 75% quartiles (Q1 and Q3, respectively) and the outliers, which will be represented here by dots.

Then, Figure 17 shows that the distribution changes a lot from county to county. For example, the municipalities of Ladário and Bodoquena always have consistently low areas of deforestation. On the other hand, municipalities like Corumbá have the first quartile (25%) above 20000 hectares, which shows that there are constantly very high deforestation rates. Still, the municipality of Corumbá and Cáceres seem to be the ones with the highest deforestation and the most inconstant since they have the largest amplitude. The outliers on Corumbá and Porto Murtinho may be related to the fires, which are pretty standard in these municipalities (c.f. Figure 14).

Figure 17: Distribution of Deforestation during the years for each municipality.



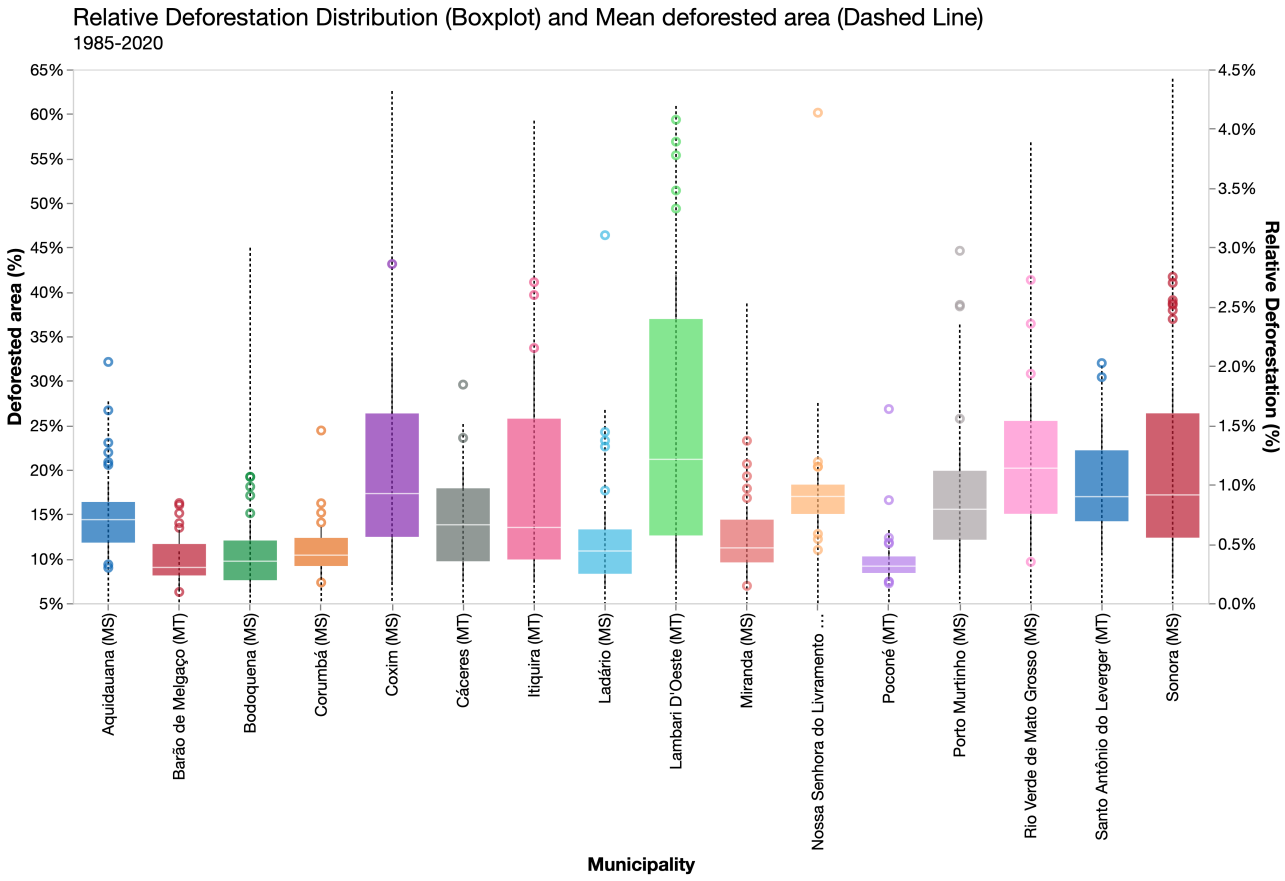
Source: The author.

However, while very useful, this graph can be tricky in two main ways: (i) the municipalities have different sizes, and therefore, it can make sense to look at relative deforestation, and; (ii) some municipalities are almost entirely deforested already, and this can make deforestation rates low, but not because it is a municipality that takes excellent care of preservation, but just because there is little left to deforest. With this in mind, Figure 18 was made, which seeks to address these two complications and give a more transparent view of what is happening.

Here, there are two independent axes: the one on the left represents the percentage that has already been deforested in that municipality, and the one on the right represents the distribution of relative deforestation over the years (represented by boxplots). Thus, one may see that although the municipality of Corumbá (MS) is the municipality with the most considerable deforestation, it has one of the lowest relative deforestation rates overall (with the third quartile just above 0.5%), also being one of the municipalities with the most natural forest of all, with less than 10% deforested. Furthermore, one can see for

example, that the municipality Lambari D'Oeste (MT), Sonora(MS), Coxim (MS), and Itiquira (MT) are municipalities in a less than favorable situation since it already has a large part of their territory deforested (nearly 60%) and continues to have high relative deforestation rates. Moreover, the municipality of Barão de Melgaço (MT) and Poconé (MT) may be one of the Pantanal municipalities in a better situation since they have low relative deforestation rates while also being two of the least deforested overall and also maintaining absolute deforestation low (c.f. Figure 17).

Figure 18: Relative Deforestation Distribution (Boxplot) and Mean deforested area (Dashed Line) during the years for each municipality.



Source: The author.

5.4 Feature Engineering

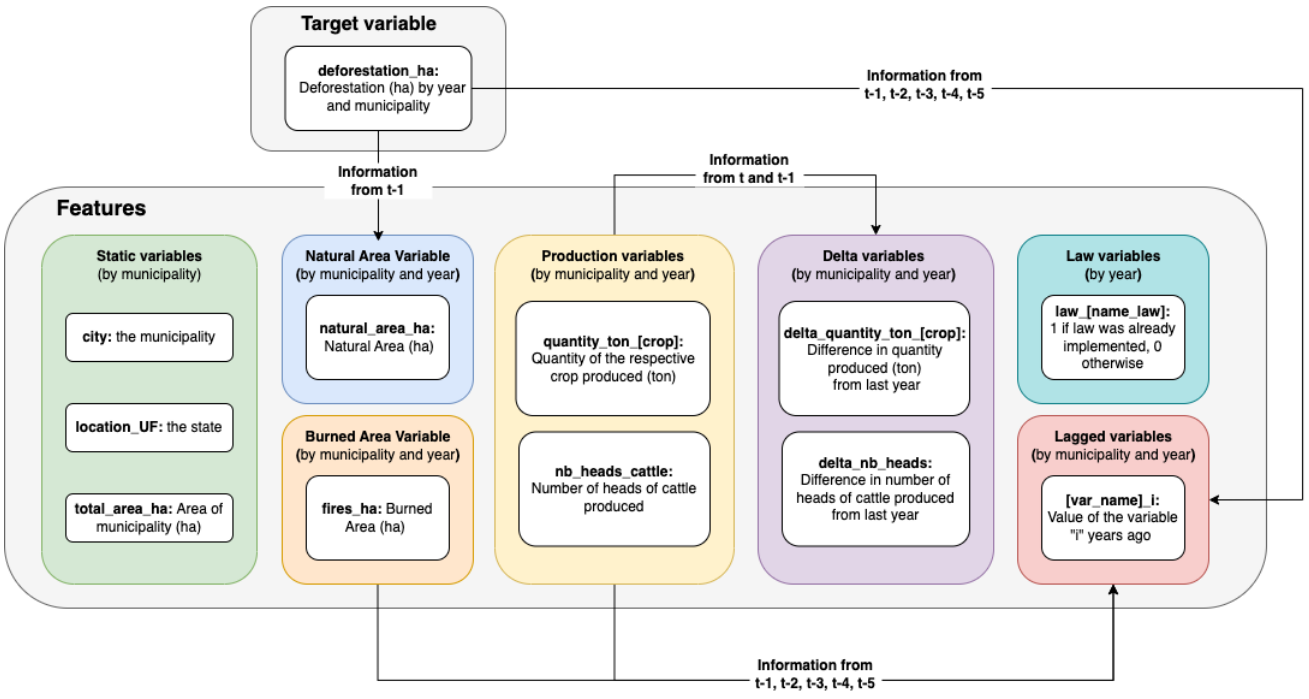
Feature engineering can be defined as transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. In short, it is manually designing the input X to get the best

model possible (BROWNLEE, 2014). Thus, one can consider that the feature engineering part actually started with the studies about Pantanal, followed by the selection of the data to collect in section 5.1.

It is of note that all of the features were idealized after the discoveries of Chapter 1, highlighting the relevance of agricultural cultures and livestock production fires in the Pantanal's deforestation. Moreover, the relevance of some of the environmental law's in the Brazilian scenario is known and could have a somewhat relevant impact on illegal deforestation in Mato Grosso and Mato Grosso do Sul states (IBF, 2020).

In total, during the Feature Engineering phase, a total of 38 features were created, which are described in Table 12, within its type (numerical, binary, categorical), if it is static or temporal and if it was collected as is or created somehow. However, Figure 19 gives a summarized view of the different categories of features created and how they relate.

Figure 19: Diagram of types of features created during the Feature Engineering.



Source: The author.

Let us now explore each category, together with the motivation behind each of them:

1. **Static municipality variable**: includes all the static variables represented in Figure 19, i.e, `city`, `location_UF` and `total_area_ha`. The idea of adding these vari-

ables to the model is that it must somehow know how to differentiate between counties since the same model will be used for all of them. Moreover, having a variable such as the `total_area_ha` is useful because the model can learn that bigger municipalities will naturally tend to have bigger deforestation, as discussed in Chapter 5.

2. **Natural Area Variable**: As discussed in Chapter 5, the rationale behind getting the evolution of the Natural Area in time is that the model might learn that as the Natural Area reduces, the potential area that can be deforested also reduces, which may lead to a decrease in deforestation.

3. **Burned area variable**: Another relevant factor that directly affects the conversion of Natural Areas into Anthropic Areas, such as defined in the Land Use Transition database from MAPBIOMAS, is the fires. As discussed in section 1, Pantanal was the biome with the most burned area in the last 36 years. This must be considered to predict deforestation in this biome's municipalities accurately.

4. **Production variables**: As discussed in (MITTERMEIER et al., 1990), some main threats to Pantanal are intensive agriculture and cattle pasture. Therefore, it becomes clear the importance to adding these features to our model. Here, the main cultures (Corn, Soybeans, and Sugarcane) were left as individual variables and the other cultures were aggregated in `quantity_ton_others_permanent` for the other permanent cultures and `quantity_ton_others_temporary` for the other temporary cultures.

5. **Delta Variables**: The delta variables are all the variables that start with the `delta` keyword in Table 12. These variables represent the difference of the truncated variable difference from the actual *year* to the $year-1$, which can be mathematically represented by $max(0, x_i^t - x_i^{t-1})$, where i represents the type of variable, and t represents the year. The rationale for creating these variables was that a significant difference in livestock production could indicate that more area is being allocated to this activity, which could come from deforestation. On the other hand, if the livestock production did not change from one year to the other, one would imagine that no significant deforestation took place from year $t-1$ to t.

6. **Law Variables**: The law variables are all the variables that start with the keyword `law` in Table 12. These laws, in general, have as their primary objective to protect the environment and reduce to a minimum the consequences of devastating actions (IBF, 2020). Therefore, the primary environmental laws implemented since 1985

were transformed in features for the model in the following manner: binary variables that have the value one if the law was already implemented and 0 if it was not yet implemented. Therefore, it could be represented mathematically by the indicator variable $\mathbb{I}\{t >= implementation\_date_{law}\}$.

7. **Lagged Variables**: Finally, the lagged variables are all the variables that finish with _i, for i $\in \{1, 2, 3, 4, 5\}$. The lagged variables are simply the value of the variable in the year $t-1$, i.e., x_i^{t-1}, where i represents the variable type and t represents the year. The rationale behind creating these features is that time series tend to have some autocorrelation. That is, the value of a variable in a specific time is correlated to a value of the same variable (or other) in another period. Moreover, the model could even learn the delta variables only having the lagged features since it only depends on x_i^t and x_i^{t-1}. One key lagged feature created here is deforestation, since knowing the deforestation in the last years at least helps the model differentiate the different municipalities, but also may help learn some deforestation patterns over the years.

However, there is still a need to select which of the manually designed features is helpful for our model, which is the goal of the following subsection.

5.5 Feature Selection

The feature selection phase is a critical step in the modeling phase. In short, one needs to input only valuable data for our model. Otherwise, it will be misguided by the useless piece of data provided. To find the very best subset selection, one would need to perform an algorithm that iterates in all the possible variable sizes (for example, $k \in \{1, 2, ..., p\}$) and fit all the $\binom{p}{k}$ models that contain exactly k predictors and select the best model for k predictors using some validation metric. Then, finally, one would want to select the best k that minimizes the prediction error. However, unfortunately, the number of models to be tested in this method grows exponentially (2^p, which can be interpreted as a binary decision of putting a feature or not for all p features), which makes it computationally expensive to test all the possibilities (JAMES et al., 2013).

Hence, it was decided to use the Backward stepwise selection. In a nutshell, the idea is to start with the full model $\mathcal{M}_p$ and then choose the least important variable to drop, using some validation metric. After that,one defines the model $\mathcal{M}_{p-1}$ without the dropped variable and find the next variable to drop. This process continues until $\mathcal{M}_1$, with only

one variable. Then one would compare all the models from $\mathcal{M}_p$ to $\mathcal{M}_1$ and select the one with lower cross validation (c.f. subsection 6.3) prediction error (JAMES et al., 2013).

However, since 38 features were created, computing all this would require a lot of computation time, so a kind of heuristic of the Backward stepwise selection was used, with the help of the Shapley Additive Explanations (SHAP) values, a method that helps in evaluating the feature importance that will be more detailed in Chapter 9. First, a model was fitted with all the 38 features and a selection of the top $\frac{2}{3}$ of features was made, already filtering a great part of the variables. After that, an attempt was made to take more variables out from the model, verifying if it enhanced the model's performance or not. Finally, the features selected are shown in Table 5.

There are some things worth noticing about the features that were selected. Firstly, one notices see that only one of the Static municipality variables was kept. This makes sense since the `total_area_ha` is already useful to differentiate between the municipalities and it is a quantitative variable, which means there is no need to use something like One hot encoding, which largely increases the number of columns. Secondly, one can see that both the `natural_area_ha` and the `fires_ha` variable were selected, which proved our intuition that both would indeed be useful for helping predict deforestation.

However, something interesting happens with the production and delta variables. Only half of them were selected, leaving the "other permanent" and "other temporary" variables out of the cut, as well as the sugarcane variable. Hence, the first thing that can be inferred is that even aggregated, the other cultures are not quite relevant to our study. Furthermore, it is interesting to notice how the main culture that did not make the cut was sugarcane, which is exactly the culture that presented the smaller area by volume produced, in Figure 15.

Another interesting fact was that none of the variables related to the implementation of environmental laws were selected. There are some hypotheses that can justify this, which should be tested in future studies. First, they could be already reflected in other variables. Secondly, it is possible that these laws are not necessarily the main ones for the Pantanal region or, finally, these laws may not be efficient in halting deforestation.

Lastly, one can notice that 100% of the Lagged variables were selected, which means that indeed, information from the past from deforestation, livestock production, and fires helps us predict future deforestation.

Table 5: Selected features after heuristic of Backward stepwise selection.

Selected Features	**Percentage selected**
1.Static municipality variables	33.33%
total_area_ha	
2. Natural Area Variable	100.00%
natural_area_ha	
3. Burned Area Variable	100.00%
fires_ha	
4. Production Variables	50.00%
nb_heads_cattle	
quantity_ton_corn_(grain)	
quantity_ton_soybeans_(grain)	
5. Delta Variables	50.00%
delta_nb_heads	
delta_quantity_ton_corn_(grain)	
delta_quantity_ton_soybeans_(grain)	
6. Law Variables	0.00%
-	
7. Lagged Variables	100.00%
deforestation_ha_1	
deforestation_ha_2	
deforestation_ha_3	
deforestation_ha_4	
deforestation_ha_5	
nb_heads_cattle_1	
nb_heads_cattle_2	
nb_heads_cattle_3	
nb_heads_cattle_4	
nb_heads_cattle_5	
fires_ha_1	
fires_ha_2	
fires_ha_3	
fires_ha_4	
fires_ha_5	

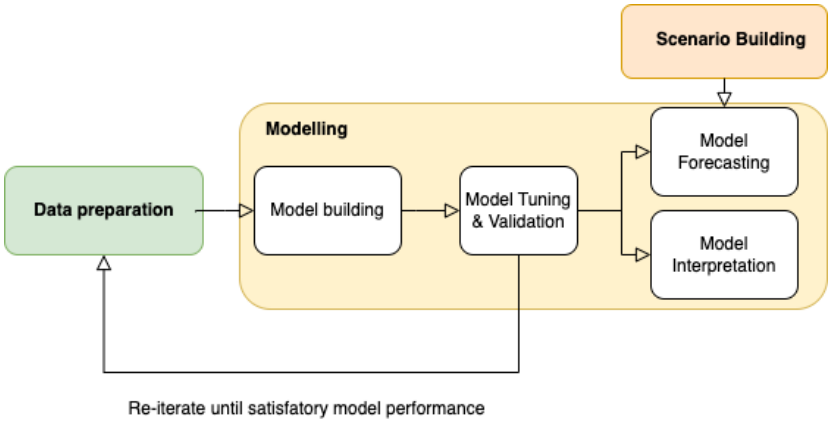
Source: The author.

Having a good idea of how the data behaves, let us move on to the modeling phase, where everything seen so far will be combined to build the final model.

6 MODELLING

This Chapter is divided into three sections: (i) building the model's Pipeline, (ii) Hyper-parameter tuning, and (iii) model validation with the time series cross-validation. These three steps are presented in Figure 20, in the first two blocks (Model Building and Model Tuning Validation). While the model forecasting and model interpretation are theoretically part of the modeling phase, they were put in the Results' part (Part IV) since they represented concrete outcomes of the model.

Figure 20: Methodology Diagram with focus on the Modelling.



Source: The author.

As a first remark, it is worth mentioning that since the complete dataset produced by the data preparation phase has a size $n = 560$, considering all the municipalities, using a different model for each municipality is not feasible since each model would be trained in a tiny dataset and statistical models tend to perform better with large datasets. One possible approach would be to use data augmentation, but it has been chosen to use a single model to model all the municipalities, as in (DOMINGUEZ et al., 2022).

6.1 Model Pipeline

As detailed in section 3.4, the model chosen for our forecasting is Extreme Gradient Boosting (XGBoost). As explained before, it is a Gradient Boosting Tree-Based Supervised Learning Model, a state-of-the-art algorithm with many improvements from the classical Gradient Boosting Model. The implementation of the model was done in Python in one of the Jupyter Notebooks (as aforementioned, notebooks are files that can combine coding, texts, and visualizations) inside the folder `notebooks`, as seen in Chapter 4. Moreover, it is worth mentioning that the main Python packages used for its implementation are `pandas`, `xgboost`, and `sklearn`.

At first, since there were categorical variables such as the municipality and the state of the municipality, there was a need to structure a more complex Pipeline that would preprocess the categorical and the numerical/binary variables differently. However, after the Feature Selection step described in the subsection 5.5, the categorical variables were not chosen for the final model, so the more complex Pipeline structure is not needed.

Moreover, XGBoost turns out to be a handy model since not a lot of preprocessing steps are needed. For instance, being a Tree-Based model, XGBoost is not sensitive to monotonic transformations of its features, such as scaling, since it only needs to choose the split points (CHEN; GUESTRIN, 2016b). That is, a split in one scale corresponds directly to a split on the other scale, which is why an operation such as scaling is not needed.

Furthermore, as mentioned in subsection 3.4, the XGBoost algorithm has a sparsity-aware split finding, which means it can deal with NaN values as a default. However, this does not necessarily mean it will perform better than some inputting methods. Thus, some options were explored: (i) leave the NaN values; (ii) attribute with the mean; (iii) input with the median, and (iv) input with x_i^{t-1}. Anyhow, the best performing one was the (i), i.e., do not input any value at all and let XGBoost deals with the NaN values.

Hence, because of what was mentioned above, the final Pipeline is simply the XGBoost encapsulated by a `RandomizedSearchCV` that will be essential in the hyper-parameter tuning, described in the following section.

6.2 Hyper-parameter Tuning

Tuning hyper-parameters is one of the most critical steps when creating a Machine Learning model, especially in the case of Tree-based ML models or Deep Learning, where there are a lot of hyper-parameters. Recall that, as mentioned in 3.1.4, hyper-parameters define the model architecture and must be determined before the training phase. Moreover, the main goal of these parameters is to play with the bias-variance trade-off described in the subsection 3.1.3, i.e., they can increase or decrease the complexity of the model, making it more or less likely to overfit or underfit.

The hyper-parameter optimization process has four main components (YANG; SHAMI, 2020): (i) the Estimator (e.g., XGBoost); (ii) the Search Space (all the possible values for the hyper-parameters); (iii) the Search Method (e.g., Randomized Search, described later); (iv) Evaluation method (e.g., R^2 in k-fold cross-validation).

Let us now explore in more detail the Search Space and the Search method used since the Estimator was already well discussed in subsection 3.4, and the Evaluation method will be explained in subsection 6.3.

XGBoost contains dozens of hyper-parameters that could be explored, but some of the main ones have been chosen to be optimized based on (CHEN; GUESTRIN, 2016b). Let us now detail what each of them represents (CHEN; GUESTRIN, 2016a):

- `max_depth` (range: $[0, +\infty)$): maximum depth of each tree. The bigger this value is, the more complex the model will be, and it will be more likely to overfit.
- `learning_rate` (range: [0,1]): step size shrinkage parameter ν explored in subsection 3.4, used to prevent overfitting.
- `subsample` (range: $(0, 1]$): row sub-sample, as discussed in subsection 3.4. Represents the sub-sample ratio of the available training examples used for training a tree at each boosting iteration.
- `colsample_bytree` (range: $(0, 1]$): type of column sub-sample, as discussed in subsection 3.4. Represents the sub-sample ratio considered when constructing each tree, i.e., the percentage of the features that will be used for each tree's construction.
- `colsample_bylevel` (range: $(0, 1]$): type of column sub-sample, as discussed in subsection 3.4. Represents the sub-sample ratio considered when constructing every

new depth level of the tree, i.e., the percentage of the features available to that tree that will be used for constructing each depth level.

- `n_estimators` (range: $[10, +\infty)$): the number of trees that the boosting will consider. Decreasing this hyper-parameter reduces the likelihood of overfitting.

While these are the hyper-parameters selected to be optimized, the search space is defined with the parameter values the model will explore. Thus, the Search Space explicitly used in our modeling is shown in Figure 38.

The Search Method chosen, on the other hand, was the `RandomizedSearchCV`. The idea of this method, in contrast with the `GridSearchCV`, is that not every single combination of the hyper-parameters will be tried (which in our case would be computationally expensive), but a `n_iter = 10` samples of the parameter distribution (c.f. Figure 38) (PEDREGOSA et al., 2011). Here, any probability distribution could have been used for the parameters. Still, since there was no prior knowledge available about which parameters would suit our problem better, a simple uniform probability distribution was used, i.e., all the parameters in the list will have the same probability of being chosen.

The selected hyper-parameters are displayed in Table 6. One can see that some parameters were chosen to complexify the model more than the `default` setting, such as `max_depth` and `n_estimators`, that represented larger values than the default ones (6 and 100). However, the `learning_rate` and the `subsample` were chosen at shallow values, which highly prevents overfitting. Finally, the sub-sampling of columns, both for the `colsample_bytree` and `colsample_bylevel`, were chosen for values smaller than the default (1, i.e., without any subsampling). Still, these are not small numbers, which may indicate that most of the features chosen are pretty helpful for the training phase.

Table 6: Hyper-parameters optimized and best value found in `RandomSearchCV`

Hyper-parameter	**Best value found**
`max_depth`	15
`learning_rate`	0.005
`subsample`	0.1
`colsample_bytree`	0.9
`colsample_bylevel`	0.8
`n_estimators`	1000

Source: The author.

In the following subsection, some details will be given about evaluating the hyper-parameters for each sample extracted using `RandomSearchCV`, with a technique called Time series Cross-Validation.

6.3 Time Series Cross Validation

Generally, as seen in 3.1.3, the training error tends to get lower and lower when the model's flexibility is increased. However, the test error usually has a *U-shape*, since if one increases too much the flexibility of the model, it will overfit the data (c.f. Figure 3) (JAMES et al., 2013). Furthermore, the training error is usually smaller than the test error since the algorithms are optimized to minimize the training error. Anyhow, it must be kept in mind that what matters to us is how the model performs on unseen data since the objective is to forecast deforestation, something that has not yet happened.

A typical approach for assessing how the model performs is to use a validation set, i.e., split the data into two randomly (the train and validation datasets), train the model on the training dataset, and then use the model to predict the responses for the observations in the validation set and compare the predictions with the actual values. However, this approach relies on a specific training and validation set selection, which could result in misleading conclusions (JAMES et al., 2013).

Therefore, one could use an alternative method called k-fold Cross Validation, which consists of splitting the data in k parts randomly and then, training the model with $k-1$ groups and validate on the remaining part. This would be done for k iterations, each time having one of the parts as the validation set. Finally, the final evaluation of the model's performance would be obtained by averaging out the validation metric of choice. Moreover, it is worth mentioning that some of the usual values normally used for k in this method are $k = 5$ or $k = 10$ (JAMES et al., 2013).

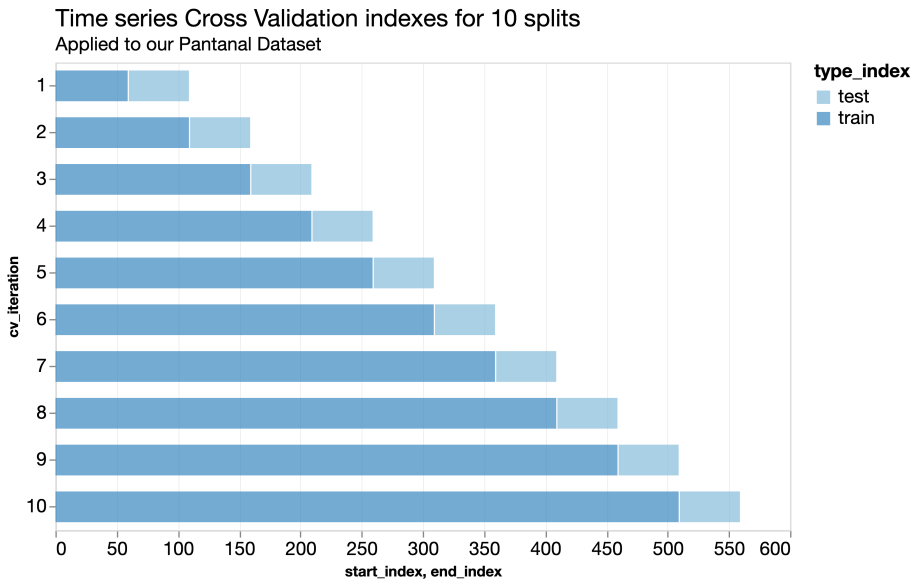
However, for our specific problem, it is not possible to use precisely this approach because of one particularity: our data represents a time series, i.e., they are indexed in time order (the years). Hence, using a usual k-fold Cross Validation approach would be a huge mistake since data from the future would be used to predict data from the past in our validation.

Therefore, it is possible to use a variation of the k-fold method that is called the Time Series Cross-Validation. In this approach, in the split k, the first k folds will be received as a train set and the fold $k+1$ as the test set. Unlike the usual k-fold cross-validation

approach, in this case, the successive training sets are supersets of the training sets that come before them (PEDREGOSA et al., 2011). Furthermore, for a typical value $k = 10$, for example, it is actually needed to split the data into $k+1$ parts, which becomes evident in Figure 21.

Figure 21 represents the training and test set indexes for our specific dataset, where dark blue represents the training sets and light blue the test set. Here, $k = 10$ was chosen, since it is one of the most popular ones, which presents a good trade-off between the error estimation and computational efficiency (JAMES et al., 2013). Nevertheless, one can see that the first splits are trained on a really small part of the data, which may harm their performance.

Figure 21: Representation of training and test data sets in each iteration of the Time Series Cross Validation.



Source: The author.

Thus, to evaluate the performance of a certain model (e.g., the final model or a model during the hyper-parameter tuning phase), one can compute the following value:

$$TSCV_k = \frac{1}{k}\sum_{i=1}^{k} M_i \tag{6.1}$$

where M_i represents a metric of choice computed using the ith test set and $TSCV_k$ represent the Time Series Cross Validation score for k iterations. However, since, in our case, the first CV iterations have a small portion of the dataset, it would be reasonable

to use some more robust metrics, such as

$$TSCV_k^{robust} = \text{median}_i(M_i) \tag{6.2}$$

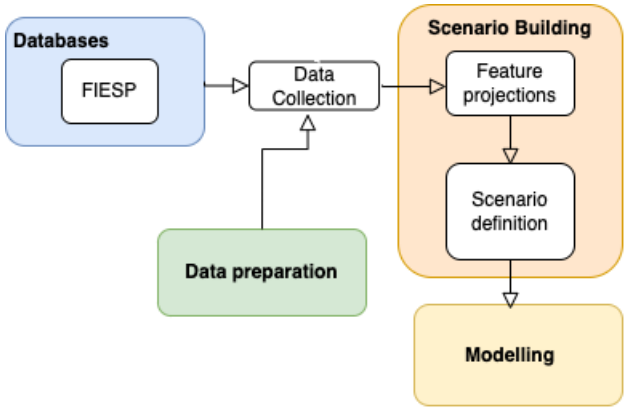
or even evaluate the distribution of the results with, for example, a box plot. Note that the metric M_i can be any validation metric. The ones used in our validation are the R^2, $RMSE$, and MAE, detailed in section 3.1.2.

The results using everything established so far concerning the cross-validation will be presented in Part IV. Finally, let us now move to the definition of the scenarios proposed for the study of future years.

7 SCENARIO BUILDING

To have some sensibility of what may happen in the following years, three scenarios were be established: a pessimistic scenario, a realistic one, and an optimistic one (considering the environmental view, where the optimistic version accounts for less deforestation). To do that, one needs to project all the features selected for the model for the following years. Therefore, let us first make a realistic projection of what may happen and then adapt the growth rates to obtain an optimistic forecasting and a pessimistic one.

Figure 22: Methodology Diagram with focus on the Scenario Building.



Source: The author.

Figure 22 shows the methodology diagram focusing on Scenario Building. One may see that it receives data from the Federation of Industries of the State of São Paulo (FIESP) and other data from the data preparation step (information on the growth rate of the fires). Finally, the modeling block receives these scenarios for making the predictions. Thus, this chapter will be divided into two sections representing the chapter's building blocks: the feature projections and the scenario definition.

7.0.1 Projecting the features

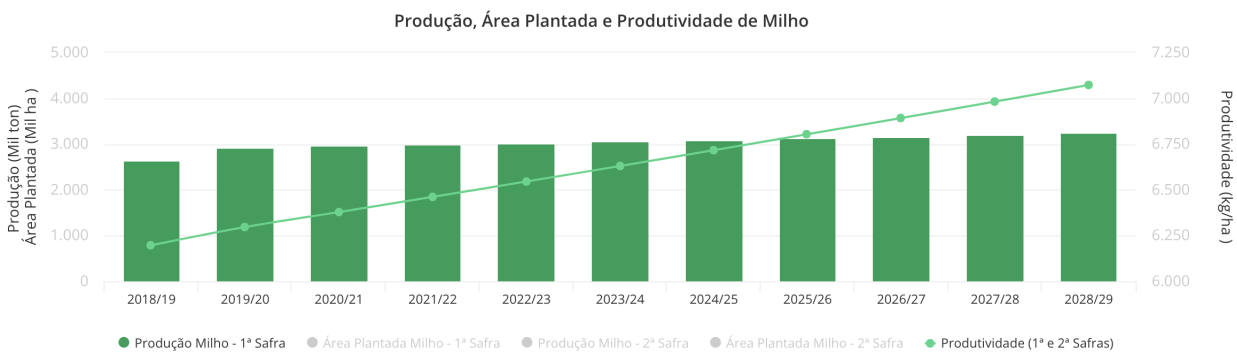
7.0.1.1 Projection of the production variables

For the Projection of the production variables, the leading source was the Federation of Industries of the State of São Paulo (FIESP), which provided estimates of the expected growth of livestock and agricultural production (FIESP, 2020).

However, one can see that the predictions made by FIESP are only made in the granularity of the regions. Thus, let us consider that the Central-West region (that contains both MS and MT states that are in the scope of this study) can be used as an estimation for the growth of production for the municipalities considered here.

Let us exemplify our methodology for getting growth for the following years with the production of corn. Figure 23 shows the type of prediction that may be found in (FIESP, 2020).

Figure 23: FIESP Projection for the Corn Production (thousand of tons) in the Central-West region until 2029.



Source: Extracted from (FIESP, 2020).

As one can see, the estimated production of corn for each of the years in thousands of tons is shown in the green bars. Therefore, it is possible to estimate the year-on-year growth for the municipalities by computing:

$$growth\_rate_t^i = \frac{forecast_t^i}{forecast_{t-1}^i} \tag{7.1}$$

for every type of product i and year t. Since, for the production variables, there is only interest in livestock, corn, and soybean production, one can use this method to estimate

its growth. The growth results obtained with this method are presented in Table 7.

Table 7: Projection of Production features until 2030

Year (t)	$growth\_rate_t^{livestock}$	$growth\_rate_t^{sugarcane}$	$growth\_rate_t^{corn}$
2021	+2.41%	+2.97%	+3.63%
2022	+2.89%	+7.17%	+3.63%
2023	+2.83%	+2.10%	+3.64%
2024	+2.82%	+2.45%	+3.64%
2025	+2.77%	+3.56%	+3.64%
2026	+2.04%	+3.09%	+3.64%
2027	+2.02%	+2.94%	+3.64%
2028	+1.98%	+2.79%	+3.64%
2029	+1.92%	+3.40%	+3.64%
2030	+1.92%	+3.35%	+3.64%

Source: The author.

7.0.1.2 Projection of the burned area variable

The other variable that needs to be projected is the area burned for each year. To predict this, different approaches were tried, such as ARIMA, a famous statistical model for time series, and even more advanced state-of-the-art models for predicting time series, like Neural Prophet, from Meta. Still, neither performed exceptionally well for the time series of the fires. This probably happens because of two main reasons: (i) the series are volatile (high variance), being hard to find patterns such as auto-correlations; (ii) the series contains a minimal number of data points (only 35 per series), since each municipality needed to be treated separately.

Therefore, a more straightforward approach was chosen. First, for each of the municipalities, the median growth rate over the last five years was computed, and then this value was used for every year of that municipality. Note that the median was used instead of the mean. The idea here is that the growth rates for the burned areas have a considerable variance and tend to have a lot of outliers, so a more robust metric was selected. The values obtained for the median growth are then:

Table 8: Median growth rate of the Burned area of the last five years by municipality.

Municipality	**Median growth rate (last 5 years)**
Aquidauana	+13.67%
Barão de Melgaço	+210.17%
Bodoquena	-5.60%
Corumbá	+27.16%
Coxim	-45.09%
Cáceres	+38.54%
Itiquira	-15.88%
Ladário	+132.82%
Lambari D'Oeste	+0.92%
Miranda	-6.83%
Nossa Senhora do Livramento	+17.06%
Poconé	+142.84%
Porto Murtinho	-44.23%
Rio Verde de Mato Grosso	+11.21%
Santo Antônio do Leverger	+128.84%
Sonora	+234.54%

Source: The author.

In general, the median growth rate, which is going to be used for the next years, is positive, which reflects what was seen in Figure 14.

7.0.1.3 Projection of the other variables

Besides the production variables and the burned area variables, there is no need to project any of the other variables. Let us understand why that is the case.

- **Static municipality variables**: `total_area_ha` is a static variable that remains the same for the next years.
- **Natural Area Variable**: `natural_area_ha` can be computed by $natural\_area_t = natural\_area_{t-1} - deforestation_{t-1}$
- **Delta variables**: can be computed by $delta\_quantity_t = quantity_t - quantity_{t-1}$

- **Lagged variables**: no need to be computed; one may get the last five variables for each.

Now that the Projection of the features is defined, let us move on to the definition of the optimistic, realistic, and pessimistic scenarios.

7.0.2 Defining the Optimistic, Realistic and Pessimistic Scenario

In subsection 7.0.1, it was explored how the different features could be projected into the future in the most realistic possible by using the FIESP predictions and a predicted burned area growth based on the last years. This will thus be our realistic scenario.

For the pessimistic scenario, let us consider that every prediction will be 20% higher than previously imagined, i.e., every growth rate will be multiplied by 1.2. This could thus represent a scenario where the exportation and local consumption of corn, soybeans, and livestock highly increased. Moreover, in this scenario, the fires are not controlled and continue increasing to levels bigger than ever (also 20% higher than the realistic scenario), as has been happening in the last couple of years.

On the other hand, for the optimistic scenario, it is supposed that it is possible to reduce around 14.3% (or $\frac{1}{7}$) the consumption of red meat by, for example, having Brazil join the Meat Free Monday, which is an international campaign that encourages countries not to eat meat on Mondays. France, for example, participates actively in this campaign (*Lundi sans viande*, in french) by not serving meat in university restaurants around the country. Moreover, in 2019 more than 500 french personalities signed on behalf of the initiative (MONDE, 2019). Still, one may consider that the growth for the fires will come back to the median growth rate of 2010-2015, represented in Table 9. Notably, the growth rates were way lower, most negative. The same growth as the realistic setting will be considered for the other production variables.

Table 9: Median growth rate of the Burned area of the years 2010-2015 by municipality

Municipality	**Median growth rate (2010-2015)**
Aquidauana	27.82%
Barão de Melgaço	-48.38%
Bodoquena	152.98%
Corumbá	-23.66%
Coxim	19.02%
Cáceres	12.19%
Itiquira	-37.21%
Ladário	-41.82%
Lambari D'Oeste	-20.12%
Miranda	-47.82%
Nossa Senhora do Livramento	-28.28%
Poconé	-16.93%
Porto Murtinho	-50.96%
Rio Verde de Mato Grosso	-6.39%
Santo Antônio do Leverger	-35.31%
Sonora	-89.44%

Source: The author.

Finally, having defined all three scenarios, let us move to the results.

PART IV

RESULTS

This part aims to show the results obtained by our model, using the methodology described in Part III. This part is composed of 3 main chapters: (i) model validation, where validation metrics and charts are shown; (ii) model interpretation, where the SHAP values indicate how the model is functioning; (iii) future forecasting for the three scenarios until 2030.

8 MODEL VALIDATION

To validate our model, let us use the Time Series Cross Validation, as explained in the subsection 6.3. In each iteration of the Cross Validation, the model will be trained using the "training" subset of the data (highlighted in dark blue in Figure 21) and then validated using the "test" subset (highlighted in light blue in Figure 21), with each of the three metrics presented in the subsection 6.3: R^2, the Root Mean Squared Error ($RMSE$) and the Mean Absolute Error (MAE).

Let us start by plotting the predictions versus the actual test data for each set of each iteration. One can see the results in Figure 24. Each chart corresponds to one iteration of the Time Series Cross Validation (c.f. Figure 21) and represents a comparison between the model predictions and the test data set. Moreover, note that in Figure 24, all the municipalities are mixed in the data since the test set contains data from all of them. This justifies the high variance of deforestation, going from zero to approximately 40,000 ha.

The model was inspected in terms of R^2 for each iteration. Mainly on iterations 5, 7, 8, and 9, where it has a R^2 above 82.8% for all of them, the model accurately predicted most of the future deforestation. The worst iteration is the first one, which makes sense since it has the smallest training set (c.f. Figure 21).

Moreover, the last iteration ($CV = 10$) was the second worst result, which seems counterintuitive at first sight since it has the most training data. However, Figure 13 and 14 show that deforestation and fires had an unusually big value for the last few years, way above the expected values. Therefore, it makes sense that the model could not accurately predict these years. Furthermore, the model predicted well peaks in deforestation but missed their magnitude. Another possible explanation is that the model still did not know the harm the fires could cause at such high levels as the ones presented in the last year of Figure 14 since there were no occurrences of this level of natural burning in the past.

Figure 24: Comparison between $\hat{y}_i$ and y_i for each Time Series CV iteration.

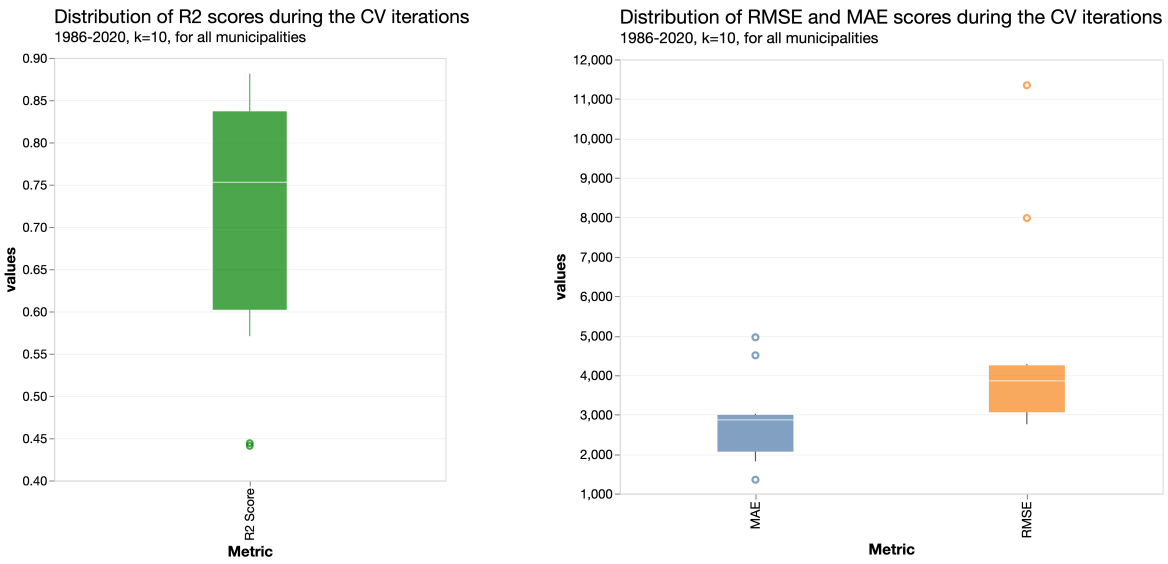
Source: The author.

Figure 25 shows the distribution of the three metrics previously discussed: R^2, the Root Mean Squared Error ($RMSE$), and the Mean Absolute Error (MAE).

Figure 25: Distribution of the 3 metrics (R^2, $RMSE$, MAE) using boxplots.

(a) R2 Score Boxplot for the CV iterations

(b) RMSE and MAE Boxplots for the CV iterations



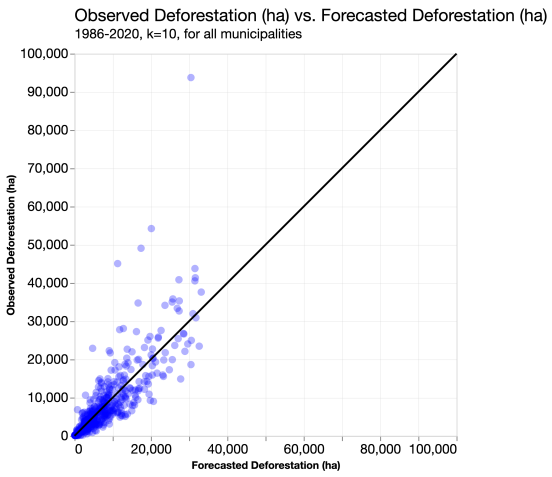
Source: The author.

As seen in Figure 25 (a), the median R^2 is just above 75% at 75.28%. Moreover, two data points are considered outliers, which are precisely the first and last iterations of the Cross Validation. The boxplot also shows that the distribution is asymmetric and has a long left tail, which means that the mean is probably lower than the median. Indeed, computing the mean of the R^2 yields 70.70%, which is lower than the median.

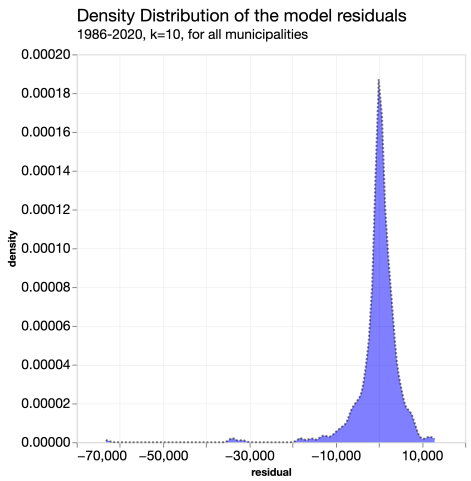
Figure 25 (b) shows the distribution of the Mean Absolute Error, the ℓ_1 norm as discussed in subsection 3.1.2, as well as the distribution of the Root Mean Squared Error, the ℓ_2 norm, during the different iterations of the Cross Validation. In this case, RMSE is higher than the MAE, which is expected since $MAE \leq RMSE$ is always true. Furthermore, the median of the MAE is just below 3,000 ha at 2,866 and the mean of the MAE is around 2,882 ha, representing the expected absolute error for an unseen example. The RMSE, on the other hand presents a higher variance, having outliers that correspond to the first and last iteration of the CV. Nonetheless, this metric disproportionately penalizes higher errors, leading to a maximum RMSE of more than 11,000 ha, more than twice as high as the maximum MAE. Precisely because of the presence of these outliers, the preferred metric, in this case, would be the MAE (GERON, 2017).

Figure 26: Scatter plot observed vs. predicted deforestation (a) and model residuals distribution (b).

(a) Observed vs. Predicted deforestation (ha)



(b) Density Distribution of the Model Residuals

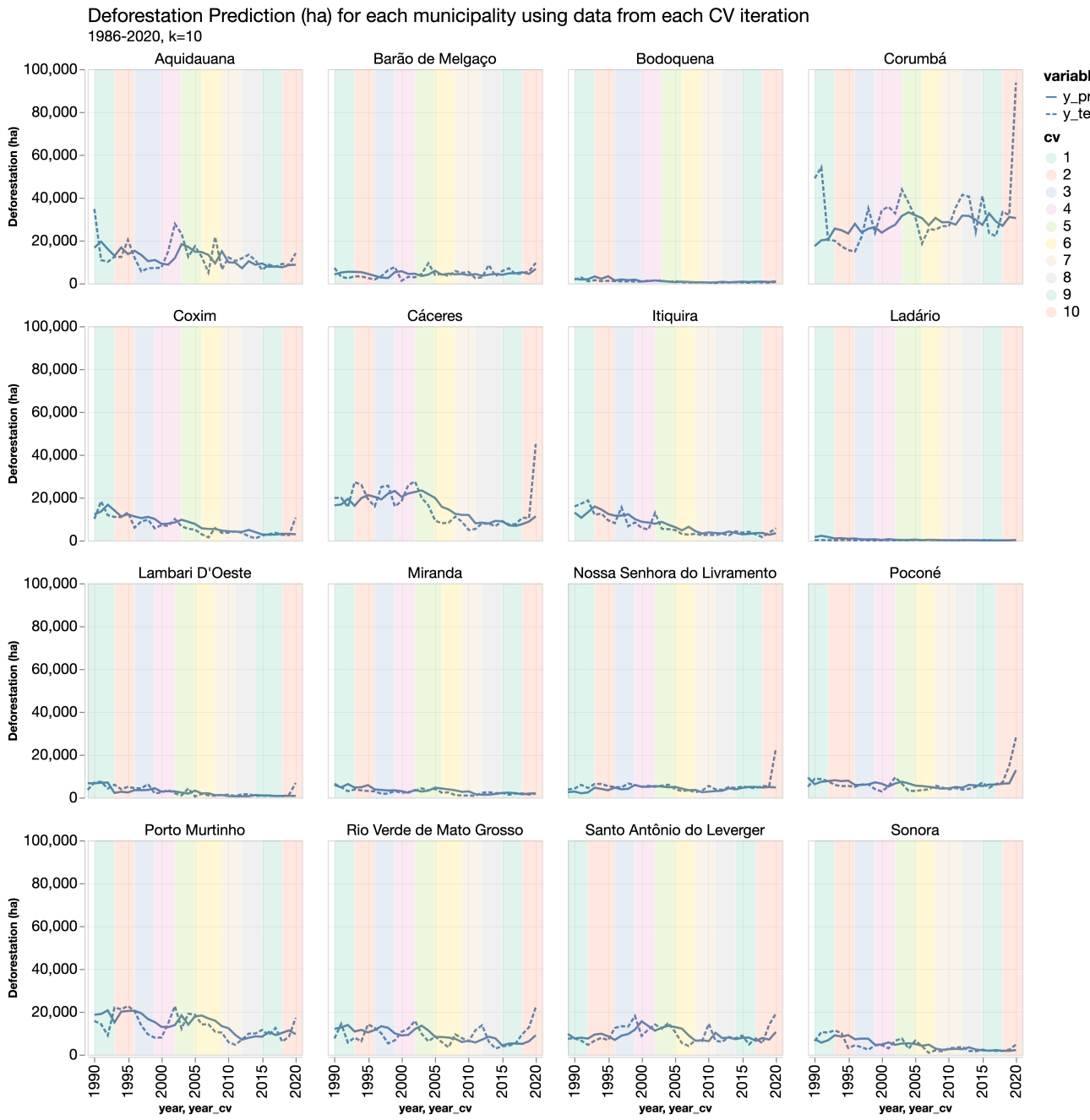


Source: The author.

Figure 26 takes a different approach to look at the results, inspired by (DOMINGUEZ et al., 2022). Here, all the values from all the CV iterations are mixed, obtaining a list of all the y_i observed values and all the $\hat{y}_i$ predicted values. Thus, all the prediction points here corresponded to predictions of unseen data when these points were part of the test data set in the CV. Figure 26 (a) shows the scatter plot with the observed deforestation (ha) on the y-axis and the predicted deforestation (ha) on the x-axis. The black line represents a perfect model with no residual at all. In general, the model can predict deforestation accurately. However, some observations have been highly underestimated, such as the blue point furthest away from the black line, which had a deforestation of over 90,000 ha and a prediction of around 30,000.

Figure 26 (b), on the other hand, shows a density plot of the residuals, i.e., $residual_i = \hat{y}_i - y_i$ for each observation i. As can be seen, the mode of the distribution of the residuals is indeed zero, which makes sense if compared to Figure 26 (a). However, a long left tail is identified, meaning the model underestimates the actual value. On the other hand, the long tail originates from the abnormally high deforestation values in the last year, which the model could not accurately predict, underestimating it.

Figure 27: Comparison between $\hat{y}_i$ and y_i for each municipalities.



Source: The author.

Finally, Figure 27 shows the deforestation forecasting in hectares by municipality. It is essential to understand that all the predicted values (the unbroken line in the charts) were the model outputs when these specific data points were on the test dataset during the Cross Validation. The different colored regions highlight which data points were estimated in which iteration of the Cross Validation.

Here the results show that the model follows the general trends quite well but is less

accurate when predicting deforestation for the municipalities with the most considerable deforestation, such as Corumbá, which relates to what is shown in Figure 26 (a). Moreover, one can see how the iterations of the cross-validations with higher R^2, such as the 8th one (represented in the gray region in the chart), correspond to a better prediction since the residuals in this region tend to be lower than in other regions.

9 MODEL INTERPRETATION: SHAPLEY ADDITIVE EXPLANATIONS

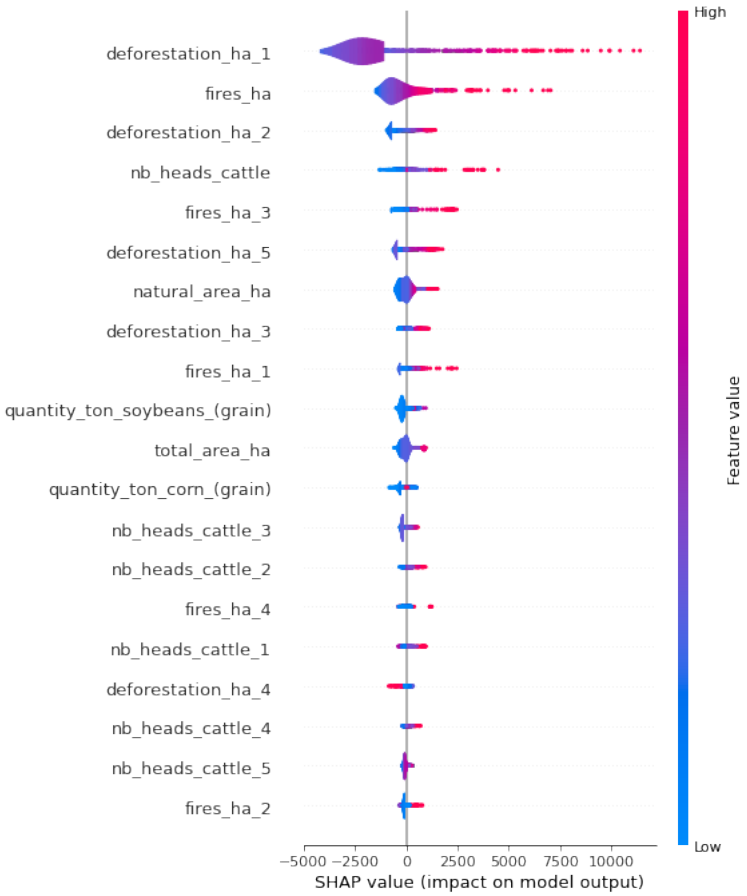
This chapter aims to grasp what the model is doing, avoiding the idea of considering it as a black box. To do this, let us use the SHAP (Shapley Additive Explanations) values, which consist of a unified approach for interpreting model predictions based on game theory. The mathematical details of how it works and its theory are out of this study's scope, but more details can be found in (LUNDBERG; LEE, 2017).

In short, the idea of the SHAP values is the following: the bigger the SHAP value, the more it impacts the model in the sense of predicting higher deforestation. Conversely, a negative SHAP value contributes negatively to the deforestation prediction, making it smaller.

Figure 28 is the SHAP Summary Plot, with the violin type of plot. This chart combines the feature importance with the feature effects. Firstly, the features are sorted in terms of the feature importance (MOLNAR, 2022). That is, the most relevant variable for the model is the lagged deforestation with $t = 1$, followed by the burned area, the lagged deforestation with $t = 2$, and the livestock production.

Moreover, each point presented in this chart represents a Shapley value (x-axis) for a particular feature (y-axis) and observation. The points' color indicates the features' value from low to high, with red being the highest value and blue being the lowest value (MOLNAR, 2022). Therefore, by taking, for example, the `deforestation_ha_1`, one can see that most of the data have a small lagged deforestation, represented by a thick region on the chart of blue points. Moreover, since these points are on the negative side of the x-axis, having a small lagged deforestation will contribute negatively to the prediction of deforestation, which makes sense. Moreover, the reddest points contribute the most to the SHAP values (of the order of 10,000).

Figure 28: SHAP Summary Plot with violin shape.



Source: The author.

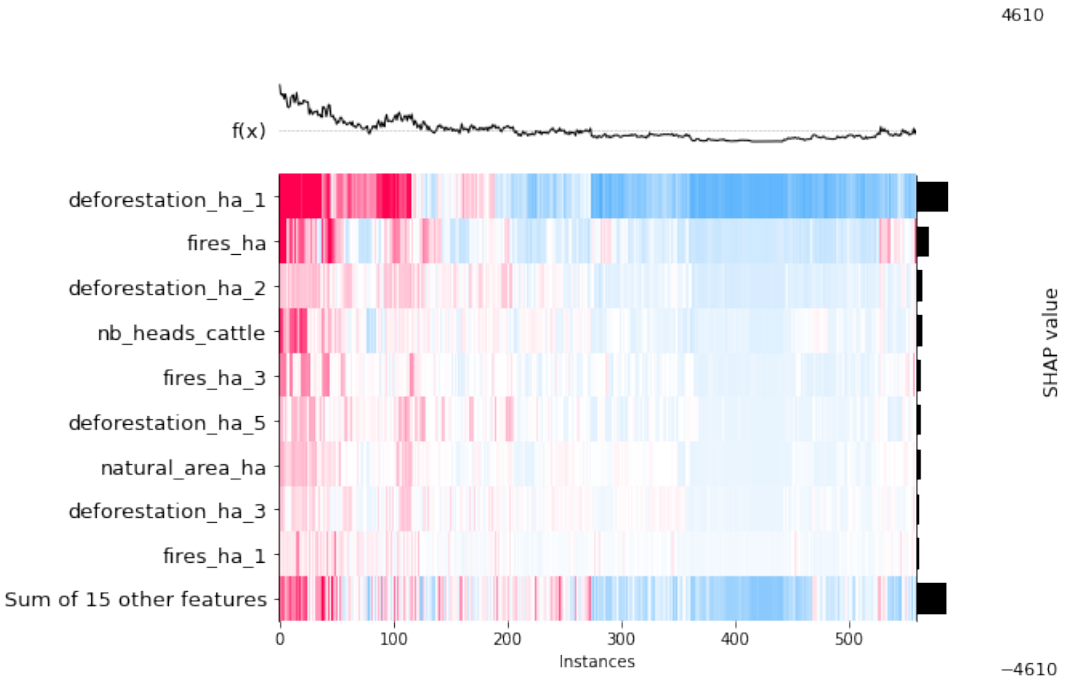
As aforementioned, the most important features are the lagged deforestation with $t = 1$, the burned area, the lagged deforestation with $t = 2$, and the livestock production. Moreover, not only are they the most critical features, but they contribute to the SHAP value in the expected sense, i.e., more fires, lagged deforestation, or livestock production implies more deforestation. This assures us that the model is doing what it was expected to do. It is also worth noting that the lagged features of the `fires_ha` were also considered essential features (especially `fires_ha_3` and `fires_ha_4`), which may indicate that the burned areas may take some time to start being used for some human activity.

Moreover, features such as the `natural_area_ha` and `total_area_ha` were also relevant, contributing positively to the SHAP values with the growth of features. The only feature that seems counter-intuitive is the `deforestation_ha_4`, which seems to decrease the SHAP value as it grows. This could be a way of modeling capturing some cyclical behavior of deforestation, or it can be just a coincidence on the specific dataset used since the impact is not that relevant.

Figure 29 shows yet another quite interesting representation of the SHAP values. This Figure represents the heatmap plot, which, as seen, has the instances on the x-axis and the features on the y-axis, with the colors representing the SHAP values (red for high, blue for low). The instances are ordered using a hierarchical clustering technique that uses the explanation similarities of the instances to bring them together. Note that there is a $f(x)$ in the top, which represents the output of the model for each instance x. Moreover, the black bars on the right show each feature's importance.

First, it can be observed that the left region of the chart has high SHAP values for the most relevant features and represents the instances with the highest output overall (c.f. function at the top of the graph). Furthermore, it is interesting to note how close most of the predictions are to each other, with only a few deforestation predictions being significantly higher (precisely those concentrated in the left corner of the graph). Another interesting point that can be noticed is that, despite not appearing visually, the other 15 features are pretty relevant since the sum of their importance is greater than the importance of the most prominent feature, which is the deforestation of the previous year.

Figure 29: SHAP heatmap plot.



Source: The author.

It is also possible to look at some of the main features in detail to see how they relate

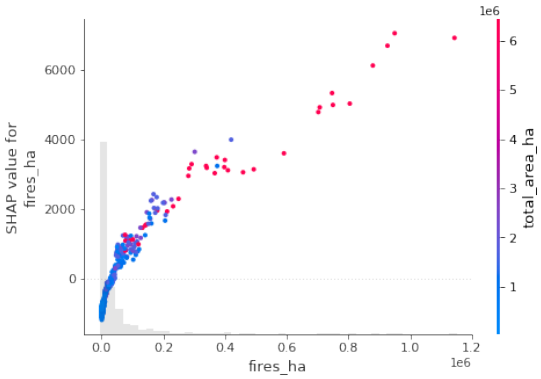
to the SHAP values. This can be evaluated in the Dependence Plot, represented in Figure 30 for the variables `fires_ha` and `nb_heads_cattle`, which are the most actionable of the main features, from a public policy perspective. Each chart point represents one row of our dataset; the x-axis represents the feature value, and the y-value stands for the SHAP value. Moreover, the distribution of the features can be seen in the gray histogram. On the other hand, the color of the points represents the value of a chosen feature (represented in the right), which denotes the features that have the most interaction with the feature of interest (MOLNAR, 2022).

One may notice some interesting facts in Figure 30. Firstly, Figure 30 (a) shows that the relation between the `fires_ha` and SHAP is almost linear; however, at the beginning of the curve, the derivative is higher, meaning that having no fire or having a small amount of fire is entirely different for the model. In addition, the gray histogram shows that most examples have a really small or even null fire since cases of intense fire are sporadic. This might explain why the model had a hard time predicting extreme deforestation in the last years. Furthermore, the color of the points shows that the most extensive fires tend to occur in municipalities with larger total areas.

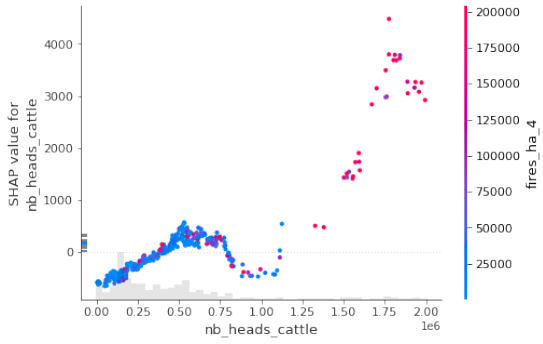
As for Figure 30 (b), there are two clusters of data points: (i) the ones with less than a million heads produced by year; (ii) the ones with more than a million. Moreover, it seems like the way the SHAP values vary with the `nb_heads_cattle` depends on the cluster, since the slope for the two parts of the chart seems significantly different, being quite higher for points in (ii). This seems to indicate that in order to have a really big livestock production, deforestation starts becoming far more intense. In addition, it seems that somehow high values of the variable `fires_ha_4` are a lot related to the points from (ii). This could indicate that livestock production starts only a certain number of years after the fire, or it could only be a coincidence and that all the fire variables interact intensively with livestock production.

Figure 30: Scatter plot observed vs. predicted and distribution for the model residuals.

(a) SHAP Dependence Plot for `fires_ha`



(b) SHAP Dependence Plot for `nb_heads_cattle`



Source: The author.

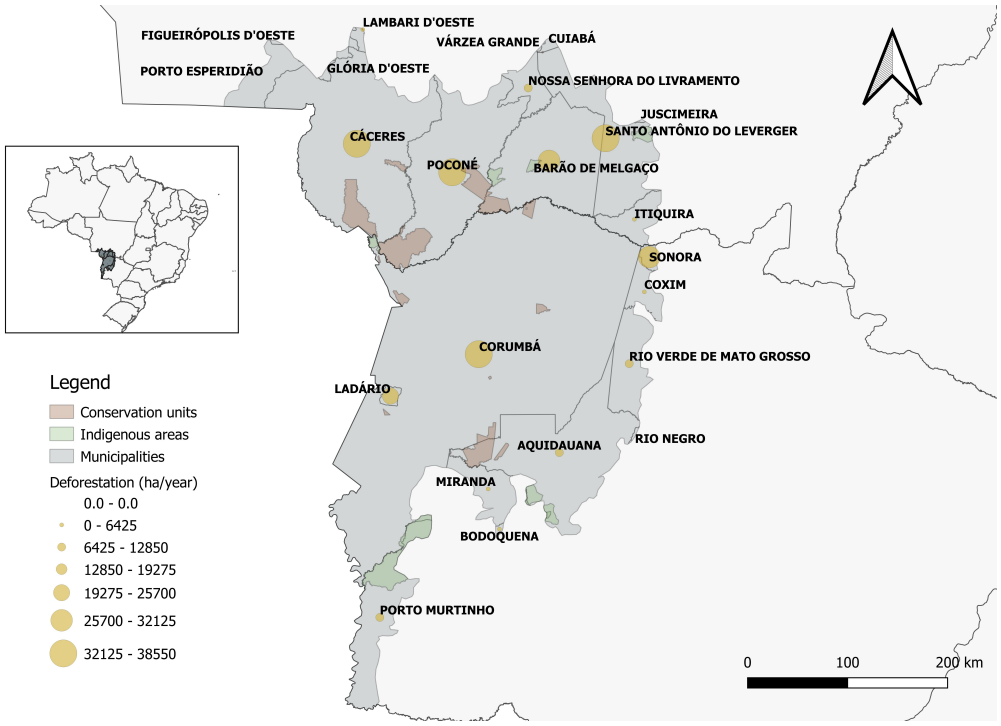
10 FORECASTING FOR DIFFERENT SCENARIOS

The goal of this Chapter is to predict the deforestation for each municipality under each scenario: (i) the realistic one, estimated using the FIESP data and the median of the five last year's growth rates for the forest fires; (ii) the pessimistic one, from the environmental viewpoint, where every prediction is 20% worst than previously imagined and (iii) the optimistic one, where it is possible to reduce the livestock production by 14.3% by making Brazil join the Meat Free Monday and where it is feasible to reduce the growth of fires to the levels obtained between 2010-2015, which are a lot better than the ones from the last five years.

Since statistical methods tend to perform worse when trained on fewer observations, the final model should use the whole data set for its training. (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). Thus, the first step is to retrain the model on all 560 observations in our dataset. After that, it will be needed to predict deforestation sequentially, year after year, for each of the scenarios. This is necessary since the features from $t-1$ are needed to construct the delta, lagged variables and the natural area variable.

Figure 31 shows a map with the expected realistic deforestation in 2030. As can be seen, the most deforested municipalities are Corumbá, Cacéres, and Santo Antônio do Leverger, all in the range of 30,000 to 40,000 ha deforested by year. However, Cáceres and Santo Antônio do Leverger seems to be more worrisome since the total area of the municipalities is smaller (around three times smaller for Cáceres and around six times smaller for Santo Antônio do Leverger). Furthermore, the conservation units in Santo Antônio do Leverger also draw attention because of its conservation units, which need to be preserved even with this high deforestation rate, which may be a challenge.

Figure 31: Map of the realistic predictions for the deforestation (ha)/year by municipality in 2030.



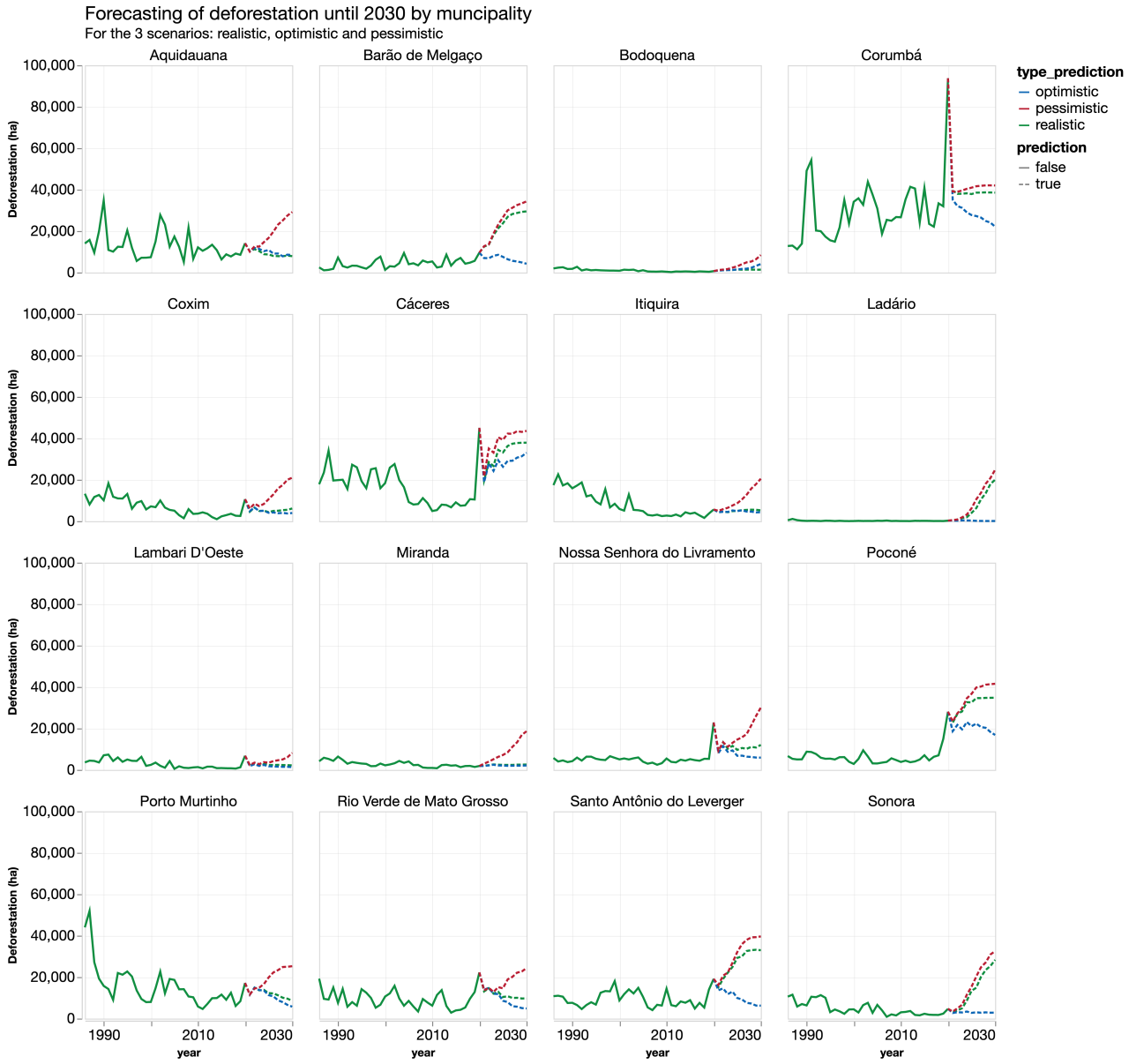
Source: The author.

Figure 32 shows the forecasting for deforestation for each of the municipalities in the Pantanal and each scenario until 2030. The predictions for the scenarios follow a befitting pattern, with the optimistic scenario having the lowest deforestation, the realistic in the middle, and the pessimistic with the highest deforestation.

Attention should be given to Corumbá. As can be seen, even after retraining the model on the complete data, the model continues to consider the last available observed value (2020) for Corumbá an outlier since the prediction for 2021 drops even with similar feature values. This may relate to the fact that in the hyper-parameter tuning, several parameters were selected to reduce overfitting. This prevents the model from quickly concluding that deforestation will increase to this level with just one data point.

Moreover, Ladário is another example deserving of attention since the model predicted that deforestation will increase highly in the following years. However, this is directly related to our assumption that the fires will continue growing at the median rate from the last five years, which was pretty high (+132,82%). Although this is a reasonable assumption, the model interprets that if this rate is maintained, the municipality will reach unprecedented levels of deforestation in 10 years.

Figure 32: Forecasting of Pantanal's Deforestation by the municipality until 2030 for each scenario.

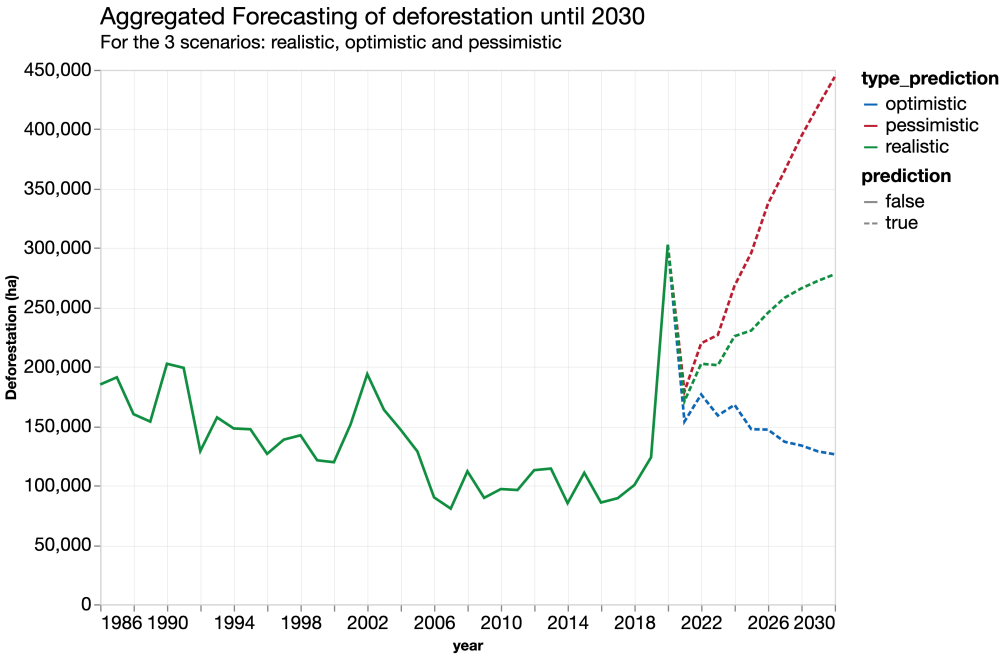


Source: The author.

Finally, let us analyze the data in aggregate form, which makes it easier to have a global view of what will happen to the Pantanal. As shown in Figure 33, the behavior of the general graph is strongly related to the Corumbá graph since it represents the most significant part of deforestation among all the municipalities. Still, it is pretty clear here the difference between the scenarios and how much actions to reduce meat consumption and fires can decrease deforestation in the realistic scenario. It is also clear how critical it can be to maintain the pace of production and burning of recent years and how impactful

it is to increase this production pace even more since the pessimistic curve would imply a 50% increase over the data for 2020, which already showed the highest deforestation until then.

Figure 33: Aggregated forecasting of Pantanal's Deforestation until 2030 for each scenario.



Source: The author.

At last, Table 10 shows a summarized view of each scenario, aggregating all the areas that would be deforested in the following ten years and how much of the remaining natural area of the Pantanal municipalities it represents. Thus, it becomes even clearer the impact that the public policies proposed in the optimistic scenarios could cause, preventing the Pantanal municipalities lose around an extra 5.7% of their natural area in comparison to the realistic scenario and more than 10% in comparison to the pessimistic one.

Table 10: Analysis of the total impact of each scenario in the remaining natural area

Scenario	**Total deforestation (2021-2030)**	**% of Natural Area deforested**
Pessimistic	3.45 million ha	22.61%
Realistic	2.65 million ha	17.38%
Optimistic	1.78 million ha	11.66%

Source: The author.

PART V

CONCLUSION

This study presented a modern approach to forecast deforestation in Patanal's municipalities for the next ten years for three different scenarios. The model used was the Extreme Gradient Boosting (XGBoost), a state-of-the-art model that, after several steps of feature engineering, feature selection, and hyper-parameter tuning, proved helpful in forecast Pantanal's deforestation in the coming years, with a median R^2 of 75.28%.

It was also possible to understand the model's decisions and quantify the impact of each variable on deforestation by using the SHAP Values, that has shown the relevance of controlling the growth of the fires and livestock production, the main drivers of Patanal's deforestation. Furthermore, the study highlighted non-obvious patterns such that the area burned three years ago is a good predictor of deforestation. Since deforestation was considered the act of transforming a natural area into an anthropic area, this could mean that burned areas only start being used for human activities after some time.

Moreover, the study succeeded in its primary objective of simulating different scenarios that could aid decision-making in land use decisions and public policies. For instance, it has been shown that decreasing cattle production by around 14% (e.g., by having Brazil join the Meat Free Monday) and controlling the area burned can prevent Pantanal from losing more than 10% of its natural area in comparison to the pessimistic scenario. While forecasting deforestation is a specially complex task because of the interaction of numerous socioeconomic, political, and environmental factors, the simulated scenarios have a firm ground with reality. Therefore, they can be used as a basis for designing solutions to the deforestation problem.

Nonetheless, about the second objective that had been determined, a robust and scalable code base has been built such that others can collaborate with the project and increase the reach and impact of these results and techniques. Furthermore, knowing the lack of application of advanced statistical techniques for environmental prevention, especially in the Pantanal, it is necessary to expand the dissemination of these studies as much as possible so that chances of preserving the Brazilian biomes are maximized.

Therefore, looking forward, this model can be improved by adding new features, such as the number of new buildings, which politician was in the government, information about municipal and state laws, among others. Also, it would be possible to use different techniques, such as data augmentation, to make a separate model for each municipality, which could increase its performance. Still, this same model could be applied to other regions of Brazil that have great socioeconomic importance and suffer from high deforestation rates.

Finally, the Pantanal is going through one of its most problematic phases. As seen in the data analyses, never before has there been such high levels of deforestation and fires, with so much natural vegetation land being transferred to human use. Given the world-wide state of alert regarding the preservation of the environment and the environmental impacts that deforestation in the Pantanal biome may cause, it is necessary to act so that the quality of life of future generations can be maintained.

REFERENCES

BALL, J. et al. Using deep convolutional neural networks to forecast spatial patterns of amazonian deforestation. 2021. Disponível em: ⟨https://doi.org/10.1101/2021.12.14.472442⟩.

BATTI, M. M. E. The effect of land use policies on brazil's farming production. 2020.

BELLMAN, R. *Adaptive control processes: A guided tour. (A RAND Corporation Research Study)*. 1961. Princeton, N. J.: Princeton University Press, XVI, 255 p. (1961).

BROWNLEE, J. Discover feature engineering, how to engineer features and how to get good at it. Machine Learning Mastery, 2014. Disponível em: ⟨https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/⟩.

CHEN, T.; GUESTRIN, C. Introduction to boosted trees. 2016. Disponível em: ⟨https://xgboost.readthedocs.io/en/stable/tutorials/model.html⟩.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. Disponível em: ⟨http://arxiv.org/abs/1603.02754⟩.

CHOLLET, F. *Deep Learning with Python*. [S.l.]: Manning, 2017. ISBN 9781617294433.

COSTA, A. d. S. et al. Deforestation forecasts in the legal amazon using intervention models. *Research, Society and Development*, v. 10, n. 4, p. e8710413787, Apr. 2021. Disponível em: ⟨https://rsdjournal.org/index.php/rsd/article/view/13787⟩.

DOMINGUEZ, D. et al. Forecasting amazon rain-forest deforestation using a hybrid machine learning model. *Sustainability*, v. 14, n. 2, 2022. ISSN 2071-1050. Disponível em: ⟨https://www.mdpi.com/2071-1050/14/2/691⟩.

FERNANDES, M. Corumbá concentra quase 40% das queimadas de ms. *Diário Online*, 2010. Disponível em: ⟨https://www.diarionline.com.br/?s=noticia&id=18247⟩.

FIESP. Outlook fiesp: Projeções para o agronegócio brasileiro 2029. 2020. Disponível em: ⟨https://outlookdeagro.azurewebsites.net/OutLookDeagro/pt-BR⟩.

GERON, A. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media, 2017. ISBN 978-1491962299.

GUERRA, A. et al. Drivers and projections of vegetation loss in the pantanal and surrounding ecosystems. *Land Use Policy*, v. 91, p. 104388, 2020. ISSN 0264-8377. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0264837719315595⟩.

HARRIS, M. B. et al. Safeguarding the pantanal wetlands: Threats and conservation initiatives. *Conservation Biology*, v. 19, n. 3, p. 714–720, 2005. Disponível em: ⟨https://conbio.onlinelibrary.wiley.com/doi/abs/10.1111/j.1523-1739.2005.00708.x⟩.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics).

IBF. As principais leis ambientais no brasil. 2020. Disponível em: ⟨https://www.ibflorestas.org.br/conteudo/leis-ambientais⟩.

IBGE. Ibge lança o mapa de biomas do brasil. 2014. Disponível em: ⟨https://agenciadenoticias.ibge.gov.br/pt/agencia-home.html⟩.

IBM. What is computer vision? 2022. Disponível em: ⟨https://www.ibm.com/topics/computer-vision⟩.

JAFFE, R. et al. Forecasting deforestation in the brazilian amazon to prioritize conservation efforts. *Environmental Research Letters*, IOP Publishing, v. 16, n. 8, p. 084034, jul 2021. Disponível em: ⟨https://dx.doi.org/10.1088/1748-9326/ac146a⟩.

JAMES, G. et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. Disponível em: ⟨https://faculty.marshall.usc.edu/gareth-james/ISL/⟩.

LI, C. A gentle introduction to gradient boosting. 2022. Disponível em: ⟨https://www.chengli.io/tutorials/gradient_boosting.pdf⟩.

LUNDBERG, S.; LEE, S.-I. *A Unified Approach to Interpreting Model Predictions*. arXiv, 2017. Disponível em: ⟨https://arxiv.org/abs/1705.07874⟩.

MAPBIOMAS. As cicatrizes deixadas pelo fogo no território brasileiro. MAPBIOMAS, 2021. Disponível em: ⟨https://mapbiomas-br-site.s3.amazonaws.com/Fact_Sheet.pdf⟩.

MIRANDA C. S., P. F. A. C. P. A. Changes in vegetation cover of the pantanal wetland detected by vegetation index: a strategy for conservation. 2018. Disponível em: ⟨https://www.scielo.br/j/bn/a/S9YBCDcn8dhZ8JD3gvPnB5r/abstract/?lang=en⟩.

MITTERMEIER, R. A. et al. Conservation in the pantanal of brazil. Cambridge University Press, v. 24, n. 2, p. 103–112, 1990.

MOLNAR, C. *Interpretable Machine Learning*: A guide for making black box models explainable. 2. ed. [s.n.], 2022. Disponível em: ⟨https://christophm.github.io/interpretable-ml-book⟩.

MONDE, L. L'appel des 500 pour un « lundi vert » : « nous nous engageons à remplacer la viande et le poisson chaque lundi ». 2019. Disponível em: ⟨https://outlookdeagro.azurewebsites.net/OutLookDeagro/pt-BR⟩.

MORAES, A. S. Pecuária e conservação do pantanal: Análise econômica de alternativas sustentáveis - o dilema entre benefícios privados e sociais. 2008. Disponível em: ⟨https://repositorio.ufpe.br/bitstream/123456789/3702/1/arquivo3426_1.pdf⟩.

NATIONS, U. Sustainable development goals. 2016. Disponível em: ⟨https://sdgs.un.org/goals⟩.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

ROQUE, F. O. et al. Upland habitat loss as a threat to pantanal wetlands. *Conservation Biology*, [Wiley, Society for Conservation Biology], v. 30, n. 5, p. 1131–1134, 2016. ISSN 08888892, 15231739. Disponível em: ⟨http://www.jstor.org/stable/24760914⟩.

SILVA, J. d. et al. Evolution of deforestation in the brazilian pantanal and surroundings in the timeframe 1976-2008. 2011. Disponível em: ⟨https://www.geopantanal.cnptia.embrapa.br/publicacoes/3geo/artigo-3.pdf⟩.

SILVA, J. da; ABDON, M. Delimitação do pantanal brasileiro e suas sub-regiões. 1998. Disponível em: ⟨http://mtc-m12.sid.inpe.br/col/sid.inpe.br/iris@1912/2005/07.19.20.30.13/doc/santos.pdf⟩.

SILVA LEILA M.G. FONSECA, T. S. K. M. I. S. E. A. C. A spatio-temporal bayesian network approach for deforestation prediction in an amazon rainforest expansion frontier. 2019. Disponível em: ⟨https://doi.org/10.1016/j.spasta.2019.100393⟩.

SWARTS, F. *The Pantanal*. [S.l.: s.n.], 2000.

TORTATO, F. Resumo executivo da proposta de criação do mosaico de unidades de conservação do pantanal norte. jul. 2018.

WASSERMAN, L. *All of statistics : a concise course in statistical inference*. New York: Springer, 2010. ISBN 9781441923226 1441923225. Disponível em: ⟨http://www.amazon.de/All-Statistics-Statistical-Inference-Springer/dp/1441923225/ref=sr_1_2?ie=UTF8&qid=1356099149&sr=8-2⟩.

WWF. The effects of deforestation. 2022. Disponível em: ⟨https://www.wwf.org.uk/⟩.

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, v. 415, p. 295–316, 2020. ISSN 0925-2312. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0925231220311693⟩.

YANG, X.-S. Preface. In: YANG, X.-S. (Ed.). *Introduction to Algorithms for Data Mining and Machine Learning*. [s.n.], 2019. ISBN 978-0-12-817216-2. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/B9780128172162000065⟩.

APPENDIX A – TECHNICAL DETAILS ON SUPERVISED LEARNING

A.1 Statistical Decision Theory

Let us now discuss some statistical theory in more detail that serves as a basis for developing the supervised learning models used. This section is entirely based on (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). Denote that the X and Y follow a joint distribution $\mathbb{P}(X, Y)$. Recall that the Probability Density Function (PDF) for two random variables (U, V) satisfy (WASSERMAN, 2010):

1. $f(u, v) \geq 0$ for all (u, v)
2. $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) du dv = 1$ and,
3. for any set $A \subset \mathbb{R} \times \mathbb{R}$, $\mathbb{P}((U, V) \in A) = \int \int_A f(u, v) du dv$.

Let us search for a function $f(X)$ that will try to predict Y in the "best way possible", given the input X. The "best way possible" is defined by minimizing some loss function, being the most common and convenient the squared error loss, which is defined as $L(Y, f(X)) = (Y - f(X))^2$. Therefore, let us choose as a criterion the minimization of the expected (squared) prediction error (EPE(f)). Let us first recall the definition of expected value for a continuous random variable X (WASSERMAN, 2010):

$$\mathbb{E}(X) = \int x f(x) dx \tag{A.1}$$

One can thus define EPE(f):

$$\text{EPE}(f) = \mathbb{E}(Y - f(X))^2 = \int [y - f(x)]^2 f(x, y) dx dy \tag{A.2}$$

Since $\mathbb{P}(X, Y) = \mathbb{P}(Y \mid X)\mathbb{P}(X)$, one can condition on X and write:

$$\text{EPE}(f) = \mathbb{E}_X \mathbb{E}_{Y|X}([Y - f(X)]^2 \mid X) \tag{A.3}$$

Then, it is sufficient to do a minimization on the EPE(f), point-wisely:

$$f(x) = \text{argmin}_c \, \mathbb{E}_{Y|X}([Y - c]^2] \mid X = x) \tag{A.4}$$

Finally, one finds the regression function

$$f(x) = \mathbb{E}(Y \mid X = x) \tag{A.5}$$

which is the function that minimizes the expected prediction error. This means that if one uses the $L_2^{loss} = (Y - f(X))^2$, the best prediction for each x is the conditional mean. An interesting fact is that if one changes the loss to $L_1^{loss} = |Y - f(X)|$, for instance, the optimal solution would be the conditional median. Let us now explore a simple supervised learning model that tries to estimate the regression function directly: the KNN.

A.2 K-nearest-neighbors and its limitations

The K-nearest-neighbors is probably one of the simplest supervised learning algorithms. The KNN algorithm, as it is commonly called, is a non-parametric method, which means it does not assume a parametric form for $f(X)$, which makes it a lot more flexible (JAMES et al., 2013). Moreover, it virtually has a zero training time since the predictions are directly computed using the training data.

The prediction for $\hat{f}(x)$ using the KNN is given by (JAMES et al., 2013)

$$\hat{f}(x) = \frac{1}{K} \sum_{x_i \in N_K(x)} y_i = Ave(y_i \mid x_i \in N_K(x)) \tag{A.6}$$

where $N_k(x)$ is the neighborhood of x, defined by the k closest points x_i to x in the training sample. Therefore, it is an average of the predictions for the k closest points of x. The KNN directly approximates the optimal regression function computed in A.1. More precisely, there are two approximations (HASTIE; TIBSHIRANI; FRIEDMAN, 2001):

- expectation is approximated by the average;

- conditioning at a point is relaxed to conditioning on some region somehow close to the point

Moreover, notice that the approximation gets better as $N, k \to \infty$ and $K/N \to 0$, when $\hat{f}(x) \to \mathbb{E}(Y \mid X = x)$. However, while it seems that a somewhat general estimator of the regression function was found, KNN has some serious limitations:

- does not work well for small data samples;
- as p (the number of features and the dimension of x_i) gets large; the metric size also grows, which makes the convergence rate decreases drastically

The second point is intrinsically related to the curse of dimensionality, first introduced by Bellman (BELLMAN, 1961), which shows that our intuition breaks down in high dimensions. Let us consider the KNN model for a training data set uniformly distributed in a 10-dimensional unit hypercube and consider one wants to do an estimation based on 10% of the data. To get 10% of the data, one needs to cover a range of 0.8, since $0.8^{10} \approx 0.1$. Therefore, one needs to get 80% of the range of each coordinate to get a small proportion of the data, which is entirely counterintuitive. Moreover, such neighbors are no longer "local," so the KNN approach stops working correctly.

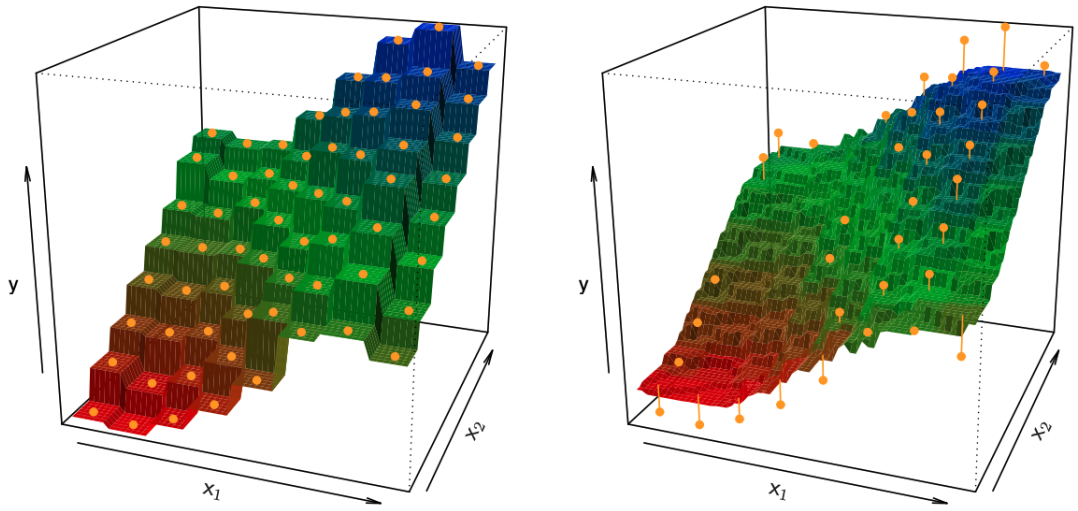
Therefore, finding other models that may rely on more rigid assumptions to obtain better predictions is needed. In the next section, a parametric model (linear regression) will be explored, as well as a key concept in statistical learning: the bias-variance trade-off.

A.3 KNN vs. Linear regression and the Bias-Variance trade-off

As seen in the last section, the KNN is a non-parametric approach, which means it can be pretty flexible. Moreover, note that the most flexible case of a KNN is when K=1. This becomes clear in Figure 34 (JAMES et al., 2013), where one sees that K=1 results in a rough step function, while for K=9, it is a smoother fit.

Moreover, for K=1, one can see a classic example of overfitting, where there is basically a zero training error. This means that for every input on the training set, the model will perfectly predict Y. However, for new data (the test set), one will probably have a significant error, since our model fits all the noise in the training set (the ϵ discussed in 3.1.1).

Figure 34: Plots of $\hat{f}$ using KNN regression on a two-dimensional dataset for K=1 (left) and K=9 (right).



Source: Extracted from (JAMES et al., 2013).

By changing the K, one can observe one of the most essential concepts in statistical learning: the bias-variance trade-off. For example, one can prove that, for a given x_0, the expected test Mean Squared Error (MSE) can be decomposed in three quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$ and the variance of the random error ϵ (JAMES et al., 2013):

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon) \tag{A.7}$$

This means that if one repeatedly estimated f with a series of training sets, the average squared error in x_0 would be decomposed by these three factors. Moreover, if one wanted to compute the overall test MSE error, one would have to do an average in all possible values of x_0 in the test set.

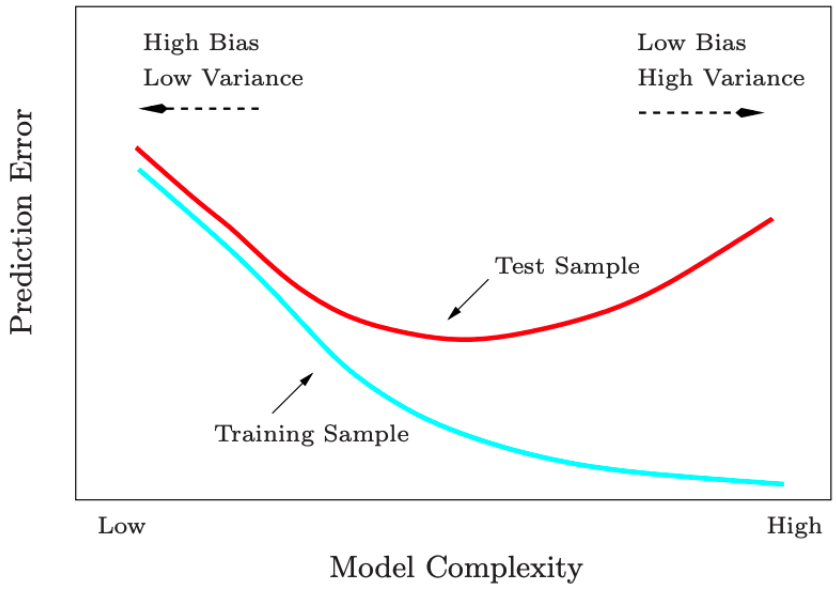
Let us now detail what each of the terms means (JAMES et al., 2013):

- $Var(\hat{f}(x_0))$: the amount by which $\hat{f}$ would change if estimated by using a different data set. More flexible models (e.g, 1-NN) have a large variance since they highly depend on the training set.
- $[Bias(\hat{f}(x_0))]^2$: error introduced by approximating a complicated model by a simpler function. In general, less flexible models (e.g. Linear Regression, discussed later in this section) tend to have a higher bias.

- $Var(\epsilon)$: the variance of our random error, that represents important features that are not included in the training set and other unmeasured variations.

Generally, as the complexity of the model is increased, the variance increases and the bias decreases, as shown in Figure 35. However, notice that the optimal Test Error happens somewhat in the middle, where one balances bias and variance. Therefore, if one considers the KNN case for different Ks, one has a flexible case when K=1 and a less flexible one when K is large. Therefore, the best K (that minimizes the test error) is somewhat in the middle, and the process of finding this K is commonly called hyperparameter tuning.

Figure 35: Test and training error as a function of model complexity.



Source: Extracted from (JAMES et al., 2013).

Therefore, to finish this section, let us compare two extreme models, a really flexible one and a really rigid one: 1-NN and Linear Regression. Recall that a Linear Regression is a parametric model with the form $f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$, where the β_j are estimated during the training phase.

- 1-NN: Extremely flexible model that will perfectly fit the training set, with zero training error. Rely on the assumption that $f(x)$ can be approximated by a locally constant function and have extremely high variance and low bias.
- Linear Regression: assumes that the regression function $\mathbb{E}(Y \mid X)$ is linear or that the linear model is a good approximation. It has thus strong assumptions and

generally has small variance but high bias. Generally, it will yield good results only if the linear assumption is not too far from the truth.

Therefore, one can conclude the section 3.1 with some conclusions:

- The ultimate goal is to estimate Y and the best theoretical approximation is the regression function ($f(x) = \mathbb{E}(Y \mid X)$). However, even the best estimation will have some error related to the ϵ that represents variables that are not included in the model and other variations.
- KNN is a simple non-parametric model that tries to directly approximate the regression function. However, it is limited, mainly because of the curse of dimensionality.
- The test MSE can be decomposed into three terms: variance, bias, and random error. 1-NN and Linear Regression are examples of models that are on the two extremes (extremely flexible vs. really rigid). Ideally, one wants to balance bias and variance to have a small test MSE.

Therefore, one wants to search for models that are in the middle of the spectrum and are somewhere in between the extremely rigid and flexible models. This is where other models, such as Tree-based models, come into place.

APPENDIX B – FORMALIZATION: FITTING IN GRADIENT BOOSTING

Formally, to find a good ensemble of trees, one cannot use traditional optimization methods, since the search happens in the space of functions. Hence, let us again use a greedy approach, as did in the construction of each Regression Tree (c.f. section 3.2). Then, at each step t one will add the tree f_t that minimizes (CHEN; GUESTRIN, 2016b)

$$J^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) \tag{B.1}$$

using the notation presented in Equation 3.10. Here, a second-order Taylor approximation may be used to help us optimize the objective function (CHEN; GUESTRIN, 2016b)

$$J^{(t)} \approx \sum_{i=1}^{n} [L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \tag{B.2}$$

where $g_i = \partial_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)})$. By removing the constant terms, one obtains a simplified version equivalent to minimizing B.2. Then, one seeks to minimize (CHEN; GUESTRIN, 2016b)

$$J^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \tag{B.3}$$

Notice that this new objective function only depends on g_i and h_i, which makes it possible to use custom loss functions, as mentioned above. Also, the exact solver can be used to perform different tasks only changing the loss from, for example, logistic regression to pairwise ranking (CHEN; GUESTRIN, 2016a).

The equation B.3, with some more few manipulations, gives us a way of evaluating the quality of a tree structure, which then results in a way of evaluating the splits. This makes it possible to optimize one step of the tree at a time and build the tree gradually in a greedy manner, each time adding the best split considering all the ensemble model until step t.

APPENDIX C – MORE TECHNICAL IMPROVEMENTS MADE BY XGBOOST

This section shows some XGBoost improvements that are not directly used by the study and are somewhat more technical.

C.1 System Design improvements

XGBoost also implements several System Design improvements which make the model more efficient and scalable. As mentioned by the authors, these changes make the model run more than 10 times faster than other popular solutions, also scaling to billion examples in distributed settings.

Some of these features are (i) Column block for parallel learning, in order to sort the data; (ii) Cache-aware access, which makes the fetching of the gradient statistics more efficient; (iii) Blocks for Out-of-core Computation, by using block compression and block sharding, which ensures the machine's resources is used fully. All of these features help the model become more scalable, which is its main objective. More details can be found on (CHEN; GUESTRIN, 2016b).

C.2 Weighted Quantile Sketch

Typically, in order to choose candidates to split one may use percentiles, which makes the candidates evenly distributed in the data. However, things get more complicated when one has weighted data points. For instance, one can rewrite the equation B.3, also now considering the regularized objective function introduced by XGBoost by (CHEN; GUESTRIN, 2016b)

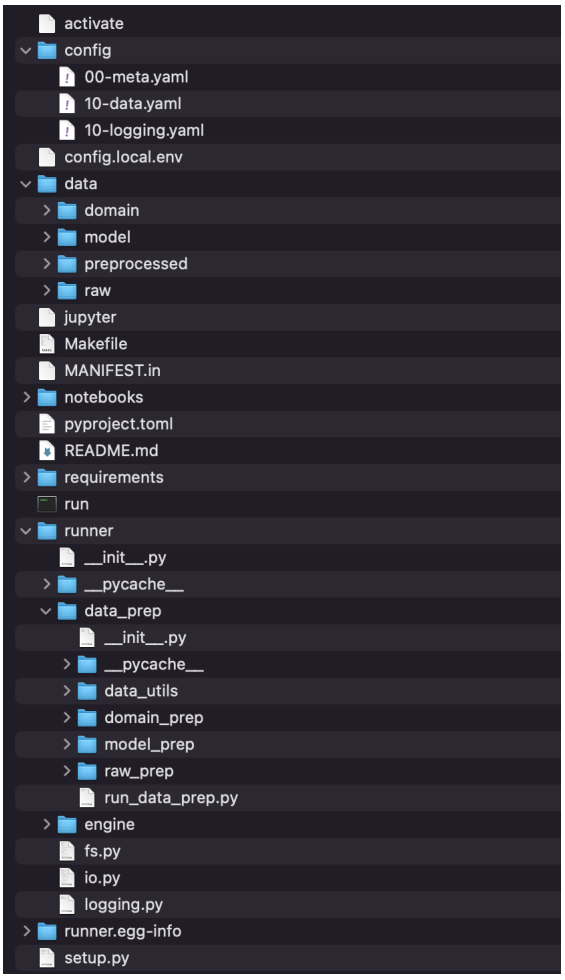
$$\sum_{i=1}^{n} \frac{1}{2} h_i (f_t(x_i) - \frac{g_i}{h_i}) + \Omega(f_t) + constant \tag{C.1}$$

This equation makes it clear that it is in fact a weighted squared loss with label $\frac{g_i}{h_i}$ and weights h_i. Since there were no theoretical approaches to finding the candidates so that they satisfy the necessary criteria for the weighted dataset case, the authors of XGBoost developed a new distributed weighted quantile sketch algorithm that has a theoretical guarantee. The details of this approach are out of the scope of this study but can be found in (CHEN; GUESTRIN, 2016b).

APPENDIX D – DETAILS ON THE CODE STRUCTURE

This appendix aims to go through the main files and folders in our project and their functions. It is important to recall that all the code is available in ⟨https://github.com/iglesiascaio/pantanal_deforestation⟩ and open to access and collaboration.

Figure 36: Code Structure of the Project.



Source: The author.

- **activate**: Document file that is useful to activate the `pantanal` virtual environment, install the dependencies that are missing from the requirements, and also define some path variables, both for Python and for gamma_config, a package used in the project that is going to be detailed in the next items. It is used once the code opens and can be easily run in the terminal with the command: `. ./activate`.

- **config**: Folder that contains the config YAML files that are used to determine, for example, some configurations of the logger used, the paths of the data files, etc. To use this folder properly, one needs the public package created by BCG GAMMA (part of BCG X), Boston Consulting Group's Data Science area. The idea here is to, jointly with the `io.py`, provide a straightforward way to read and save files. For example, to read the Land Use Transitions MAPBIOMA database, after correctly filling the config `10-data.yaml`, one can easily read the file by using `io.load_table(\raw", \land_use_transitions")`. This makes it a lot easier to read all types of data files since one can use the same command for all of them. Moreover, if one gets a new version of the database, one could just change the config file and all the code would be updated, which is really useful.

- **data**: Folder used to store all our data, in all the different preprocessing phases. The raw folder contains the Excel files used in the project, the preprocessed contains the data after the raw data preprocessing, the domain folder the data after the domain data preparation, and, finally, the model folder contains our `model_df.parquet` which is the DataFrame that will be directly be used in the modeling phase.

- **notebooks**: Folder containing Jupyter Notebooks that were used during the Exploratory Data Analysis part and also in the modeling part. In order to do the Exploratory Data Analysis one typically uses the domain data and for the modeling part, one uses the model data.

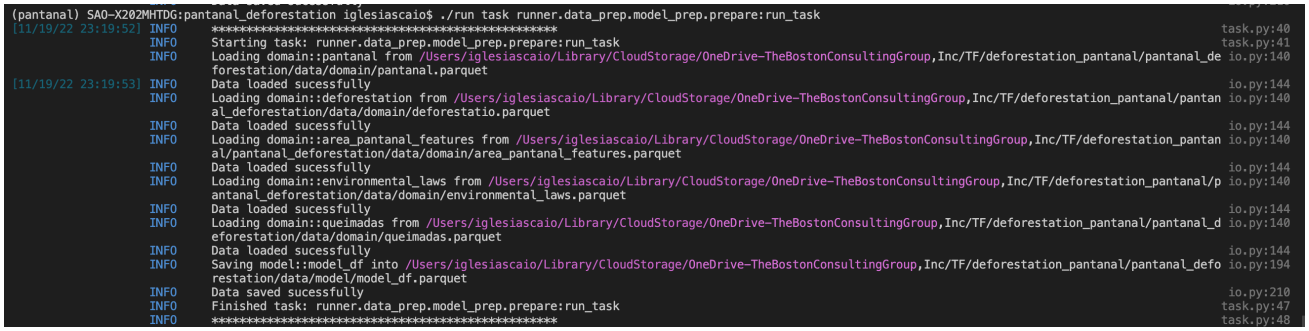
- **requirements**: Folder containing the library requirements in order to run the project. An extensive list of packages was used, but some of the main ones are `pandas`, `numpy`, `altair`, `seaborn`, `sklearn`, `gamma-config`, `click` and `black`.

- **run**: Unix Executable file that calls the `main()` function. Jointly with the `click` package, is used to facilitate running different pieces of code in the terminal. For instance, if one wants to run all the raw data preprocessing, one could simply run the command `./run raw_prep build-all`. Moreover, one can easily run a single task, i.e. a single predetermined script from a file using `./run task path_file:func`, which will then run the function `func` in the `path_file`.

- **runner**: Folder containing the main data processing files. This folder is divided mainly into two parts: (i) `data_prep`, (ii) `engine`, and the (iii) auxiliary files.
  - (i) The `data_prep` is divided as explained above, i.e. there is the `raw_prep` folder, containing all the files responsible for creating the preprocessed versions of the files, the `domain_prep` responsible for creating the domain version of the files and finally, the `model_prep` which creates the final `model_df` table, used in the modeling. Moreover, there is the `data_utils` folder that contains some files with auxiliary functions that are useful throughout the code, such as some string processing functions, and the `run_data_prep.py` file that is used to create a pipeline that runs all the preprocessing of the data.
  - (ii) On the other hand, the `engine` is used to define the `main` function and all the code structure that allows us to use the `./run` commands presented above.
  - (iii) Finally, there are three auxiliary files in the folder: the `fs.py` is used mainly to deal with paths in a clever manner, `io.py` is used to abstract read files jointly with the `config` as presented above and, finally, the `logging.py` configures the logger with the configurations inside the `10-logging.yaml` in the config.
- **setup.py**: Python file used to define the runner module, which makes it a lot easier to use its functions inside the notebooks, without the need of being in a certain path. For instance, inside any notebook file, importing the `io` module is as easy as `from runner import io` even if the notebooks are not in the root of the project.

Figure 37 shows a quick example of how this data pipeline can be used in practice and how the logs are displayed in our Visual Studio Code terminal. In this specific example, a specific task is being run (with the function `run_task()`) in the `prepare.py` file inside the `model_prep` that aims at saving the `model_df` in a `parquet` file that will be used by our `model.ipynb` notebook in the modeling phase. Note how the logger registers all the files that are being loaded and saved and the paths to the files, while also registering the run times, which are pretty useful for optimizing the code.

Figure 37: Example of a `./run` command used in the VS Code terminal.



Source: The author.

APPENDIX E – DETAILS ON THE NAN VALUES ANALYSIS

This appendix shows details of the analysis made to understand the frequency and patterns of NaN values in our databases.

E.1 IBGE data

In the case of IBGE, there were 3 different types of undefined data: "-", "..." and "...", the first being an absolute zero, the second an indication that the value does not apply and the third an indication that the data was not available. For the purpose of the model, the first case was considered as 0, and the other two cases were as NaN.

After this treatment, it was necessary to assess the frequency of the NaN values, which could potentially impair the performance of our model. For the agricultural data, about 12.81% of the area produced (ha) was missing, and about 4.4% for the quantity produced (ton) data. For livestock, only about 1.7% of the data was missing.

Furthermore, for the missing data for quantity of crops produced (which represent in total about 4.4% of the data), there were some patterns: (i) almost 40% of the missing data was concentrated in the years 1985-1988 and about 83% between 1985-1997; (ii) almost 75% of the missing data was present in 3 municipalities (Lambari D'Oeste, Sonora and Porto Murtinho); (iii) about 62% of the missing data was present in two crops (Coffee and Orange).

As for the missing data on the area produced by crops (which represents in total about 12.8% of the data), similar patterns were found: (i) about 70% of the missing data were concentrated in the years 1985-1988 and about 93% between 1985-1997; (ii) almost 33% of the missing data present in 3 municipalities (Lambari D'Oeste, Sonora and Porto Murtinho); (iii) almost 50% of the missing data in three crops (Coffee, Orange, and Cassava).

ANNEX A – TABLE NATURAL AREA IDS

Table 11: Natural Classes from MAPBIOMAS with its IDs

Class	Class ID
1.Forest	
1.1. Forest Formation	3
1.2. Savanna Formation	4
1.3. Mangrove	5
1.5. Wooded Sandbank Vegetation	49
2. Non Forest Natural Formation	
2.1. Wetland	11
2.2. Grassland	12
2.3. Hypersaline Tidal Flat	32
2.4. Rocky Outcrop	29
2.5. Herbaceous Sandbank Vegetation	50
2.6. Other non Forest Formations	13

Source: The author.

ANNEX B – TABLE ALL FEATURES

Table 12: Features collected and created for the model during the Feature Engineering

Feature name	Feature meaning	Temporal/Static	Type of Data	Collected/Created
year	Year	Temporal	Numerical	Collected
city	Municipality	Static	Categorical	Collected
location_UF	State	Static	Categorical	Collected
total_area_ha	Area (ha)	Static	Numerical	Collected
natural_area_ha	Natural Area (ha)	Temporal	Numerical	Collected
fires_ha	Burned Area (ha)	Temporal	Numerical	Collected
nb_heads_cattle	Livestock production	Temporal	Numerical	Collected
quantity_ton_corn_(grain)	Corn production (ton)	Temporal	Numerical	Collected
quantity_ton_sugar_cane	Sugar Cane production (ton)	Temporal	Numerical	Collected
quantity_ton_soybeans_(grain)	Soybeans production (ton)	Temporal	Numerical	Collected
quantity_ton_others_permanent	Other permanent production (ton)	Temporal	Numerical	Created
quantity_ton_others_temporary	Other temporary production (ton)	Temporal	Numerical	Created
delta_nb_heads	Δ Livestock production	Temporal	Numerical	Created
delta_quantity_ton_corn_(grain)	Δ Corn production (ton)	Temporal	Numerical	Created
delta_quantity_ton_sugar_cane	Δ Sugarcane production (ton)	Temporal	Numerical	Created
delta_quantity_ton_soybeans_(grain)	Δ Soybeans production (ton)	Temporal	Numerical	Created
delta_quantity_ton_others_permanent	Δ Other permanent production (ton)	Temporal	Numerical	Created
delta_quantity_ton_others_temporary	Δ Other temporary production (ton)	Temporal	Numerical	Created
law_agricultural_policy	Has Agricultural Policy Law	Temporal	Binary	Created
law_environmental_crimes_law	Has Env. Crimes Law	Temporal	Binary	Created
law_nature_conservation_units	Has Conservation Unit Law	Temporal	Binary	Created
law_water_resources_policy	Has Water Resources Policy Law	Temporal	Binary	Created
law_new_brazilian_forest_code	Has Water Resources Policy Law	Temporal	Binary	Created
deforestation_ha_1	Lagged Deforestation ($t = 1$)	Temporal	Numerical	Created
deforestation_ha_2	Lagged Deforestation ($t = 2$)	Temporal	Numerical	Created
deforestation_ha_3	Lagged Deforestation ($t = 3$)	Temporal	Numerical	Created
deforestation_ha_4	Lagged Deforestation ($t = 4$)	Temporal	Numerical	Created
deforestation_ha_5	Lagged Deforestation ($t = 5$)	Temporal	Numerical	Created
fires_ha_1	Lagged Burned Area ($t = 1$)	Temporal	Numerical	Created
fires_ha_2	Lagged Burned Area ($t = 2$)	Temporal	Numerical	Created
fires_ha_3	Lagged Burned Area ($t = 3$)	Temporal	Numerical	Created
fires_ha_4	Lagged Burned Area ($t = 4$)	Temporal	Numerical	Created
fires_ha_5	Lagged Burned Area ($t = 5$)	Temporal	Numerical	Created
nb_heads_cattle_1	Lagged Livestock prod. ($t = 1$)	Temporal	Numerical	Created
nb_heads_cattle_2	Lagged Livestock prod. ($t = 2$)	Temporal	Numerical	Created
nb_heads_cattle_3	Lagged Livestock prod. ($t = 3$)	Temporal	Numerical	Created
nb_heads_cattle_4	Lagged Livestock prod. ($t = 4$)	Temporal	Numerical	Created
nb_heads_cattle_5	Lagged Livestock prod. ($t = 5$)	Temporal	Numerical	Created

Source: The author.

ANNEX C – SEARCH SPACE FOR HYPER-PARAMETER TUNING

Figure 38: Dictionary of Parameters Distribution used for the `RandomizedSearchCV`.

```python
import numpy as np

# Parameters dict for RandomizedSearchCV
params_xgb = {
        "max_depth": [3, 4, 5, 6, 10, 15, 20],
        "learning_rate": [0.005, 0.01, 0.05, 0.1, 0.2, 0.3],
        "subsample": np.arange(0.1, 1.0, 0.1),
        "colsample_bytree": np.arange(0.4, 1, 0.1),
        "colsample_bylevel": np.arange(0.4, 1, 0.1),
        "n_estimators": [100, 400, 500, 600, 1000],
    }
```

Source: The author.