Fairness in Regression Machine Learning Data Streams - Banking System

Juliano Cavalcante

Dissertation Report

Master in Modelling, Data Analysis and Decision Support Systems

Supervised by:

PhD Bruno Miguel Delindro Veloso

2024

Acknowledgments

I would like to thank first and foremost my dear wife Daia, who has been a true partner in all this journey. Without her, this adventure would not be possible.

Abstract

Perceived fairness is crucial to building positive customer relationships in financial services, where evaluating service quality can be challenging. Trust in financial institutions is often based on perceived fairness. Integrating fairness in machine learning with the broader concept of fairness in the service and banking industries presents an important synergy. Machine learning has become increasingly important in banking for fraud detection, personalized banking, customer recommendations, and risk management, and to prevent client churn.

This dissertation examines the concept of fairness in machine learning within the context of data streams in the banking industry, instead of the more common use of batch analysis, since there is always an influx of new data in any banking scenario. The research focuses on regression models rather than traditional classification models, reflecting the practical needs of the banking sector.

Two tests were run to find the most efficient algorithm in a credit approval banking dataset to achieve that goal, checking the protected columns linked to bias. By doing that, the hope was to find the most fair algorithm.

Preliminary results point that the most well-rounded algorithms (combining good results on the fairness metrics, all well as the model error itself and also taking into account computational time) are the EWA (Exponentially Weighted Average) Regressor and the OXT (Online Extra Trees) Regressor.

This project research shows the importance of controlling fairness in online learning models. The findings suggest that while machine learning can improve decision-making and risk assessment in the banking industry, it must be deployed transparently and ethically, to prevent discrimination and ensure equitable customer outcomes.

Keywords: Fairness; Machine Learning; Data Streams; Banking; Regression Models; Bias Mitigation; Online Learning; Equity; Transparency; Risk Assessment.

Resumo

A percepção de justiça é crucial na construção de relações positivas com os clientes nos serviços financeiros, onde avaliar a qualidade do serviço pode ser um desafio. A confiança nas instituições financeiras muitas vezes baseia-se na percepção de justiça. Integrar a justiça na aprendizagem de máquina com o conceito mais amplo de justiça nas indústrias de serviços e bancária apresenta uma sinergia imporante. A aprendizagem de máquina tornou-se cada vez mais importante na banca para deteção de fraudes, banca personalizada, recomendações aos clientes, gestão de risco e prevenção da rotatividade de clientes.

Esta dissertação examina o conceito de justiça na aprendizagem de máquina no contexto de fluxos de dados na indústria bancária, em vez do uso mais comum de análise em lotes, uma vez que há sempre um influxo de novos dados em qualquer cenário bancário. A pesquisa focase em modelos de regressão em vez dos tradicionais modelos de classificação, refletindo as necessidades práticas do setor bancário.

Para alcançar esse objetivo, foram realizados dois testes para encontrar o algoritmo mais eficiente num conjunto de dados de aprovação de crédito bancário, em relação às colunas protegidas ligadas ao viés. Com isso, esperava-se encontrar o algoritmo mais justo.

Os resultados preliminares apontam que os algoritmos mais equilibrados (combinando bons resultados nas métricas de justiça, bem como no próprio erro do modelo e também considerando o tempo de processamento computacional) são o Regressor EWA (Média Exponencialmente Ponderada) e o Regressor OXT (Online Extra Trees).

Este projeto de pesquisa destaca a importância de controlar a equidade em modelos de Online Learning. As conclusões sugerem que, embora o uso de modelos de ML possa melhorar a tomada de decisões e a avaliação de riscos na indústria bancária, ela deve ser implementada de forma transparente e ética, para evitar discriminação e garantir resultados equitativos para os clientes.

Palavras-chave: Justiça; Aprendizado de Máquina; Fluxos de Dados; Setor Bancário; Modelos de Regressão; Mitigação de Vieses; Aprendizado Online Equidade; Transparência; Avaliação de Risco.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Problem . . . 1
- 1.3 Dissertation Structure . . . 2

2 Problem Definition **3**
- 2.1 Fairness in the Service Business . . . 3
- 2.2 Machine Learning . . . 4
- 2.3 Fairness in Machine Learning . . . 5
  - 2.3.1 Definition and Importance . . . 5
  - 2.3.2 Individual and Group Fairness . . . 6
  - 2.3.3 Bias Mitigation . . . 7
- 2.4 Data Streams . . . 8
  - 2.4.1 Online Learning . . . 8
  - 2.4.2 Challenges of Data Streams . . . 10
  - 2.4.3 Summary . . . 11

3 Related Work **13**
- 3.1 Fairness in Banking . . . 13
- 3.2 Machine Learning in Banking . . . 14
- 3.3 Fairness Metrics for Regression on Data Streams . . . 15
  - 3.3.1 Correlation Coefficient of Residuals . . . 15
  - 3.3.2 Prediction Distribution Analysis . . . 16
- 3.4 Fairness in Banking Data Streams - Research Gap Identification . . . 18

4 Experiments and Results **19**
- 4.1 Data Description and Analysis . . . 19
  - 4.1.1 Selection of the Dataset . . . 19
  - 4.1.2 Characteristics of the Chosen Dataset . . . 20
  - 4.1.3 Change on the Target Column . . . 21
  - 4.1.4 Exploratory Data Analysis . . . 21
  - 4.1.5 Categorical Variables . . . 22
  - 4.1.6 Numerical Variables . . . 24
  - 4.1.7 Data Pre-Processing . . . 25
- 4.2 Forecast Model . . . 26
  - 4.2.1 The River Library . . . 26
  - 4.2.2 Architecture for the Project . . . 27

4.3 Analysis . . . 29
4.3.1 Correlation Coefficient of Residuals . . . 31
4.3.2 Prediction Distribution Analysis . . . 32
4.3.3 Answering the Research Questions . . . 33

5 Conclusions **35**

Bibliography **37**

Appendix **i**
A Description of Dataset Variables . . . ii
B Univariate Analysis . . . iii
C Bivariate Analysis . . . xi
C.1 Correlation Matrix . . . xi
C.2 Correlation between Pairs of Variables . . . xii
C.3 Correlation with Target Variable . . . xii
D Model Tests . . . xiii
D.1 MSE Tables . . . xiii
E Fairness Metrics . . . xv
E.1 Coefficient Correlation of Residuals Values . . . xv
E.2 Coefficient Correlation of Residuals Averages . . . xxvi

List of Figures

2.1 Online Learning Diagram. 9

4.1 ML Data Stream Architecture. 28
4.2 MSE Tests - First Group 29
4.3 MSE Tests - Second Group 30
4.4 Model Times 30

1 Distribution of Binary Variables. v
2 Distribution of Categorical Variables. vi
3 Distribution of Numeric Variables. vii
4 Boxplots of Numeric Variables. viii
5 Pearson Correlation Matrix of Numeric Columns. xi

List of Tables

4.1 Types and count of variables. 21
4.2 Percentage of Binary Variables. 22
4.3 Labels on the *payment_type* column. 22
4.4 Labels on the *housing_status* column. 23
4.5 Labels on the *employment_status* column. 23
4.6 Labels on the *source* column. 23
4.7 Labels on the *device_os* column. 23
4.8 Percentage of Lower Outliers. 24
4.9 Percentage of Upper Outliers. 24
4.10 Capping Limits. 25
4.11 Categorical Columns for One-Hot Encoding 26
4.12 MSE Evolution - Initial Tests 28
4.13 Avg. Models - Correlation Coefficient 31
4.14 Avg. Models - Prediction Distribution Analysis 33

A.1 Detailed description of the variables used. ii
B.1 Basic Stats for Numeric Columns. iv
B.2 Percentage of Lower Outliers - Complete Table. ix
B.3 Percentage of Upper Outliers - Complete Table. x
C.4 Correlations above 30 percent between pairs of variables. xii
C.5 Correlations with target variable *credit_risk_score*. xii
D.6 MSE Evolution - Linear Regression, PA Regressor and Bagging Regressor . xiii
D.7 MSE Evolution - EWA, Logistic, Perceptron xiii
D.8 MSE Evolution - OXT, MLP, SGT xiii
D.9 MSE Evolution - Hoeffding Tree, Hoeffding Adaptive Tree xiv
E.10 Linear Regression Model Evaluation. xv
E.11 PA Algorithm Model Evaluation. xvi
E.12 Bagging Algorithm Model Evaluation. xvii
E.13 EWA Algorithm Model Evaluation. xviii
E.14 Logistic Regression Model Evaluation. xix
E.15 Perceptron Model Evaluation. xx
E.16 OXT Regressor Model Evaluation. xxi
E.17 MLP Regressor Model Evaluation. xxii
E.18 SGT Regressor Model Evaluation. xxiii
E.19 Hoeffding Tree Regressor Model Evaluation. xxiv
E.20 Hoeffding Adaptive Tree Regressor Model Evaluation. xxv
E.21 Avg. - Linear Reg. xxvi

E.22 Avg. - PA xxvi
E.23 Avg. - Bagging xxvi
E.24 Avg. - EWA xxvi
E.25 Avg. - Logistic xxvi
26 Avg. - Perceptron xxvi
E.27 Avg. - OXT Regressor xxvii
E.28 Avg. - MLP Regressor xxvii
E.29 Avg. - SGT Regressor xxvii
E.30 Avg. - Hoeffding Tree Regressor xxvii
E.31 Avg. - Hoeffding Adaptve Tree Regressor xxvii

Chapter 1

Introduction

1.1 Motivation

Coming from a banking industry background, my goal is to write a dissertation on this topic, merging it with Business Intelligence and Machine Learning, areas I aim to explore further in my professional career. A significant influence in shaping my research focus has been Professor Bruno Veloso, a notable reference in this field at FEP, who suggested the theme of Fairness in Machine Learning online streams, his area of research.

Perceived fairness plays a crucial role in creating positive customer relationships, particularly in financial services, where assessing the quality of services before and even after purchase can be challenging. Fairness is a fundamental principle in business operations, although its precise definition is intricate due to its multifaceted nature. Customers tend to develop perceptions of trustworthiness towards financial service institutions based on their fairness perceptions, as highlighted by Roy, Devlin, and Sekhon (2015).

The convergence of fairness in machine learning with the concept of fairness inherent in the service industry in general and in the banking industry, in particular, resonated with me as a compelling synergy. Seeking to contribute to the originality of the research, I pivoted towards Regression Models instead of the more conventional Classification Models. This shift is particularly relevant given the practical applications in the banking sector, where questions related to credit allocation (*"How much credit should we grant for this client?"*) hold equal if not more importance than binary classification issues.

1.2 Problem

The use of machine learning in the banking sector has gained importance over the last few years. According to Donepudi (2017), some of its main uses include detecting money laundering/fraud patterns, personalized banking and automation, customer recommendation, and risk management. Other authors such as Rahman and Kumar (2020) and Lemos, Silva, and Tabak (2022) highlight the importance of machine learning in preventing client churn (retention), which is of great importance given the competitive banking market.

The concept of Online Learning refers to constantly learning from new data and making predictions. As new data becomes available, the model dynamically updates itself. Online learning algorithms should ideally deal with the streaming setting, unlike classical batch machine learning models that train once and then deploy in production. According to Barry,

Albert, Raja, Jacob, and Vinh-Thuy (2021) this is a very important notion in the banking industry since there are always new data incoming.

To address the challenge of limited availability of banking datasets due to confidentiality concerns, I was guided towards the BAF Dataset by Jesus et al. (2022), providing a viable solution for robust empirical analysis. The central goal of my thesis is to employ various Regression Algorithms in an Online Learning Environment, using the BAF Dataset. Using this, my goal is to identify key features in regression problems and measure algorithmic performance, particularly in scenarios that incorporate fairness considerations.

The central aim of this dissertation is to address critical questions related to the application of machine learning in the banking sector, with a focus on fairness and bias. This involves evaluating the current bias in ML models and creating effective metrics for their measurement. In addition, the dissertation examines how the choice of regression algorithm affects both the fairness and accuracy of predictions in an online learning setting for banking applications.

Understanding the real-world impact of these fair machine learning models is essential to assessing their viability and effectiveness in critical banking functions. In addition to that, the research explores how fairness-aware regression models influence customer trust and satisfaction, investigating the broader impact of fair machine learning practices on customer perceptions and relationships with financial institutions.

Finally, the dissertation develops a fairness metric that can measure and compare the bias levels of different algorithms, creating a standardized way to evaluate and select less biased models. This research seeks to contribute to fair and unbiased machine learning practices in the banking industry. This represents a unique intersection of machine learning, fairness, and regression models.

1.3 Dissertation Structure

This dissertation begins with the Problem Definition (a) section, which articulates the issue this research aims to address. Following this, the Related Work (b) chapter summarizes the most recent research on the subject, offering an overview of the existing literature on the topic. The next section, Experiments and Results (c) includes a detailed description of the dataset, an explanation of the methods employed, and an analysis of the findings. Finally, the dissertation concludes with a discussion of the results, their implications, and potential areas for future research.

Chapter 2

Problem Definition

2.1 Fairness in the Service Business

Artificial intelligence and machine learning have made significant advances in the banking industry recently, completely changing the way banks work and engage with their customers. In addition to increasing efficiency and precision, the employment of these technologies has raised important considerations of fairness and trust in the sector.

According to Roy et al. (2015), perceived fairness is a central component in maintaining satisfactory relationships with customers. The significance of perceived fairness in service relationships comes from the inherent difficulty in assessing services before and sometimes even after their purchase.

Fairness is an attractive underlying business principle even if it is inherently difficult to define because it is a nuanced, multidimensional construct. In the works of Castelnovo et al. (2020), the concept of fairness remains elusive and lacks a singular definition, mainly due to its contextual nature and intricate connections among various attributes. Two perspectives approach fairness: group fairness, which aims to safeguard marginalized groups, and individual fairness, which prioritizes equitable treatment on a personal level. However, it remains unclear to what extent these two concepts align with each other.

In Carr (2007), consumers value service fairness just as much as the service itself. It also recommends improving fairness perceptions to increase customer satisfaction and loyalty in the service provider-customer relationship. Fairness is at the heart of decision-making in business-customer settings, particularly in the service industry. Consumer attitudes toward fairness significantly influence their choices regarding service consumption. Although fairness is essential in the service industry, it is difficult to define and measure because the construct has a multidimensional characteristic.

There are three main types of fairness in the service industry, defined as follows:

1. **Distributive fairness** deals with how outcomes are perceived to be fair or unfair. It encompasses the cognitive, emotional, and behavioral responses individuals have toward the distribution of outcomes by a certain entity. When individuals perceive outcomes as unfair, it can evoke emotional reactions. These outcomes can originate from any entity or individual with the authority to distribute desired results unevenly, according to Adams and Freedman (1976).

2. **Procedural fairness** is associated with the fairness of processes used to develop an

outcome; thus, it is concerned with the action taken by providers of the service. Procedural fairness is contingent upon ensuring that all procedures and steps necessary for an exchange are deemed appropriate and aimed at mitigating any biases inherent in the situation. It is enhanced when customers perceive they have the opportunity to express their opinions and have them heard, or when their audience believes that the processes leading to the outcome are both consistent and precise. Research has demonstrated that procedural fairness has a notable and beneficial impact on customer satisfaction (Smith, Bolton, and Wagner (1999)) and trust (Roy et al. (2015)). Therefore, procedural fairness plays a critical role in maintaining enduring customer relationships, leading to recurring purchases. According to Wetsch (2006), procedural fairness is identified as the foremost among fairness metrics in fostering customer trust, which is essential for enhancing customer loyalty.

3. **Interactional fairness** involves the elements of communication between those delivering fairness and those receiving it, encompassing qualities like courtesy, integrity, and regard. When individuals perceive interpersonal unfairness, they are anticipated to respond unfavorably toward the individual whose actions caused the unfairness. (Masterson, Lewis, Goldman, and Taylor (2000)).

Research has confirmed that procedural, distributive, and interactional fairness serve as the most reliable indicators of overall fairness within the service sector, effectively measuring service quality. (Carr (2007)).

2.2 Machine Learning

Machine learning is, according to the definition given by (Van Klompenburg, Kassahun, and Catal (2020)), a subset of Artificial Intelligence (AI) focused on learning. It offers a practical approach to enhance yield prediction based on various features. Through the analysis of datasets, machine learning (ML) identifies patterns and correlations, extracting valuable insights. To build models, training is essential, using historical data to establish parameters based on past experiences. During the testing phase, the performance of the model is evaluated using a subset of historical data that is not involved in the training process.

ML models can be broadly categorized as descriptive or predictive, depending on the research objectives. Descriptive models aim to extract knowledge from the collected data, explaining past events, while predictive models focus on making future predictions (Alpaydin (2014)). The construction of a high-performance predictive model in ML poses several challenges, requiring the careful selection of algorithms tailored to the specific problem. In addition, the algorithms and underlying platforms chosen must efficiently handle large volumes of data.

Machine learning extends beyond solving database problems; it is an integral part of artificial intelligence. For a system to be considered intelligent, especially in dynamic environments, it should have the capability to learn and adapt to changes. This adaptability alleviates the need for system designers to anticipate and provide solutions for every possible situation.

Statistics underpin the construction of mathematical models in machine learning, as the primary task involves drawing inferences from a sample. In the realm of computer science, efficiency is crucial. Efficient algorithms are required for training, optimizing the model, and

handling the substantial data involved. Once a model is learned, its representation and algorithmic solutions for inference must also be efficient. In certain applications, the efficiency of learning or inference algorithms, which encompasses space and time complexity, is as crucial as predictive accuracy (Alpaydin (2014)).

Differences Between Classification and Regression Models

According to Alpaydin (2014), if a bank has a history of previous loans, that includes customer data and whether the loans were repaid. Using this specific data set from previous applications, the objective is to deduce a general rule that encodes the relationship between a customer's attributes and their associated risk. In essence, the machine learning system constructs a model based on historical data, enabling it to assess the risk of a new loan application and subsequently make acceptance or rejection decisions. Customers are categorized into two classes: low-risk and high-risk. The input to the classifier comprises customer information, and its role is to categorize the input into one of the two classes. Following training with historical data, the learned classification rule may take a form similar to the one described below.

$$\begin{aligned}&\text{IF income} > \theta_1 \text{ AND}\\ &\quad \text{savings} > \theta_2\\ &\quad \text{THEN low-risk}\\ &\quad \text{ELSE high-risk}\end{aligned}$$

Now, assuming that a client is classified as low-risk, the system could determine the value of the loan that will be granted. Let X be the attributes of the client and Y be the value of the loan. Again, by analyzing past transactions, we can collect training data, and the machine learning program fits a function to these data to learn Y as a function of X. The task is to learn the mapping from the input to the output. The approach in machine learning is that we assume a model defined up to a set of parameters: In a regression model, Y will be a number.

The machine learning program optimizes the parameters, θ, so that the approximation error is minimized. That is, the estimates are as close as possible to the correct values given in the training set. An example of a regression function could be the following:

$$\text{Y} = \beta_0 + \beta_1 \cdot \text{Income} + \beta_2 \cdot \text{Savings} + \epsilon$$

2.3 Fairness in Machine Learning

2.3.1 Definition and Importance

The notion of fairness in Machine Learning is still ambiguous and not uniquely defined, mainly because it is context-dependent with complex interdependencies among several attributes. Fairness is typically approached from two perspectives: group fairness and individual fairness. Group fairness aims to protect vulnerable groups, while individual fairness focuses on ensuring equal treatment on a personal level. However, it remains unclear whether these two concepts are fully compatible.

There are many advantages in using algorithmic decision making, as computers can efficiently analyze large amounts of data with high accuracy. However, despite these advantages, there is ample evidence of the discriminatory impact of ML-based decision-making on individuals and groups based on protected attributes like gender or race. For example, US courts identified racial prejudice in Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), a software that assesses the likelihood of repeat offenses. Specifically, it was discovered that black defendants were classified with a higher risk of recidivism compared to their actual likelihood in contrast to white defendants. The search algorithms used on job-seeking platforms are another example. Investigations have revealed gender bias within these algorithms, as they tend to present higher-paying job opportunities more frequently to male candidates than to female candidates (Le Quy, Arjun, Vasileios, Wenbin, and Eirini (2022)).

According to Castelnovo et al. (2020), policy makers, companies and academic institutions are making efforts to establish guidelines and recommendations for ethics in AI, which includes fairness in ML. Diversity, non-discrimination, and fairness are considered key requirements for AI systems according to these initiatives. The principle of non-discrimination aims to allow all individuals an equal and fair prospect of accessing the opportunities available in a society. People facing similar circumstances should be treated equally and should not experience discrimination based solely on certain "protected" characteristics they possess, such as gender, sexual orientation, disability, age, race, ethnicity, nationality, or religion. Indirect discrimination occurs when certain characteristics or factors occur more frequently in the population groups against which it is unlawful to discriminate. As algorithmic decision-making systems often rely on correlations, there exists a potential to perpetuate or worsen indirect discrimination through stereotyping, particularly when differential treatment lacks justification.

Although fairness in a machine learning algorithm has some common ground that can be associated with Distributive and Interactional Fairness, it can be said that it has much more common ground with the Procedural Fairness association.

2.3.2 Individual and Group Fairness

Individual Fairness

Fairness definitions are categorized into two main groups based on the intended purpose: individual fairness and group fairness. The concept of individual fairness adheres to the idea that comparable individuals should be treated similarly (Binns (2020)). Thus, this notion focuses on the comparison of individuals, which consists of ensuring that any two similar individuals receive equal or similar outcomes.

Dwork, Moritz, Toniann, Omer, and Richard (2012) presents a simple definition: Fairness Through Awareness. By treating similar individuals similarly, it is possible to capture fairness by the principle that any two individuals who are similar concerning a particular task should be classified similarly. To achieve individual-based fairness, it is important to use a distance metric that determines how closely individuals resemble each other.

Group Fairness

On the other hand, group fairness is focused on requiring that people belonging to protected groups receive on average the same treatment as the whole population and is usu-

ally expressed as the equality of some statistical measure between groups (Verma and Rubin (2018)). Therefore, group fairness aims to provide equality of treatment for groups instead of specific individuals.

According to Mitchell, Potash, Barocas, D'Amour, and Lum (2018), the most common definitions of fairness in respect of groups identified by some protected attribute(s):

- Statistical Parity or Demographic Parity (DP) is achieved when groups have the same probability of being assigned to the positive predicted class, i.e. when the decision is independent of the sensitive feature value.

- Conditional Demographic Parity (CDP) modifies DP by requiring the parity of outcomes to hold not unconditionally, but within groups given by the level of other variables (e.g. credit risk level).

- Predictive Parity (PP), Equal Opportunity (EOpp), and Equality of Odds (EO) not only consider the prediction of the model but also the ground truth target. In particular, PP measures the precision or probability of a positive prediction to be in a positive class (positive predictive value) across groups. EOpp takes into account the likelihood of a subject belonging to a positive class receiving a positive prediction (true positive rate or recall). It is important to note that EO requires both true positive rate and false positive rate to be equal across groups.

2.3.3 Bias Mitigation

Bias mitigation Bias mitigation refers to the process of addressing specific aspects of the ML pipeline to remove the effect of unfair bias. Many techniques within the literature can be roughly classified into the following categories;

- *Pre-processing* methods are based on the idea of removing potential unfair biases directly from the training dataset. Then, a "standard" classifier is learned on this cleaned dataset.

- *In-processing* approach involves ensuring that a model generates fair results by incorporating constraints or penalties into the optimization process, thereby enforcing fairness during training.

- The focus of *post-processing* strategies is to mitigate unfair outcomes of an already trained ML model. The fundamental concept involves creating a fresh classifier based on the biased outcomes of the original model and optimizing a cost function that addresses false positives and false negatives while adhering to certain fairness constraints (Castelnovo et al. (2020)).

The initial stage of eliminating bias in ML models involves evaluating bias. Fairness depends on context; thus, there is a large variety of fairness measures. Only the computer science research area has introduced more than 20 measures of fairness to date. Because of that, there is no fairness measure that is universally suitable (Verma and Rubin (2018)).

Bias and discrimination are common problems in almost all domains. The concept of fairness varies between different fields and contexts. Assessing the effectiveness of fairness-aware algorithms proves to be difficult because they rely on specific fairness concepts. Hence,

it is essential to carefully choose or define the relevant fairness criteria for each problem within each domain, as there is no one-size-fits-all fairness standard. This remains a major challenge for researchers. (Le Quy et al. (2022)

Corbett-Davies and Goel (2018) suggest the adoption of an outcome-oriented approach represented by the following characteristics:

- Balancing inherent trade-offs in decision problems;
- Assessing calibration;
- Choosing the inputs and prediction targets;
- Creating strategies for data collection. While algorithms can mitigate numerous biases present in human decision-making, they can also amplify existing historical disparities if not crafted with caution.

However, these popular measures of fairness suffer from significant statistical limitations. Fair machine learning still has much to do, and several types of research could benefit from new statistical and computational insights. When carefully designed and evaluated, statistical algorithms have the potential to dramatically improve both the efficacy and equity of consequential decisions. With the growing popularity of these algorithms, it becomes increasingly crucial to guarantee their fairness.

2.4 Data Streams

The dynamic and evolving nature of data streams further complicates machine learning applications in banking. Data scientists and banks need to address numerous challenges while leveraging the opportunities provided by machine learning in data streams.

Traditional data management software relies on persistent datasets stored securely in stable storage. These datasets are queried and updated multiple times during their useful life. However, in various emerging application areas, data arrive continuously and require ongoing processing without the luxury of multiple passes over a static dataset. Large telecom and Internet service providers commonly install continuous data streams in their networks, collecting and analyzing detailed usage information from various network components for notable trends. Other applications that generate rapid, continuous, and large volumes of stream data include transactions in retail chains, ATM and credit card operations in banks, web server log records, etc. In many of these applications, the data stream is collected and stored in a data warehouse database management system (perhaps outside of the site), which often makes access to archived data prohibitively expensive (Garofalakis, Gehrke, and Rastogi (2016)).

2.4.1 Online Learning

The conventional approach to machine learning often operates in a batch-learning manner, typical of supervised learning tasks. This setup provides a comprehensive set of training data in advance to train a model using a specific learning algorithm. However, this paradigm requires that the entire training dataset be available before the learning task begins. Furthermore, the training process typically occurs offline due to the high cost associated with training. Batch learning methods have notable drawbacks, including inefficiency in terms of time and

space costs. Additionally, scalability issues arise in large-scale applications due to the need to re-train the model entirely when new training data emerges.

Online Learning represents a different paradigm in machine learning. It is designed for data that arrive sequentially, allowing the learner to continually learn and update the best predictor of future data at each step. Online learning effectively addresses the limitations of batch learning. Specifically, any new instance of data can immediately update the predictive model. Consequently, online learning algorithms demonstrate higher efficiency and scalability, especially in real-world data analytics applications dealing with large data and high-speed data streams (Hoi, Sahoo, Lu, and Zhao (2021)).

New data continuously refreshes the model. Online learning algorithms should ideally deal with the streaming setting: one pass on the data, constant or sublinear complexity, efficient use of resources (CPU and memory), and robustness to concept drift (Barry et al. (2021)).

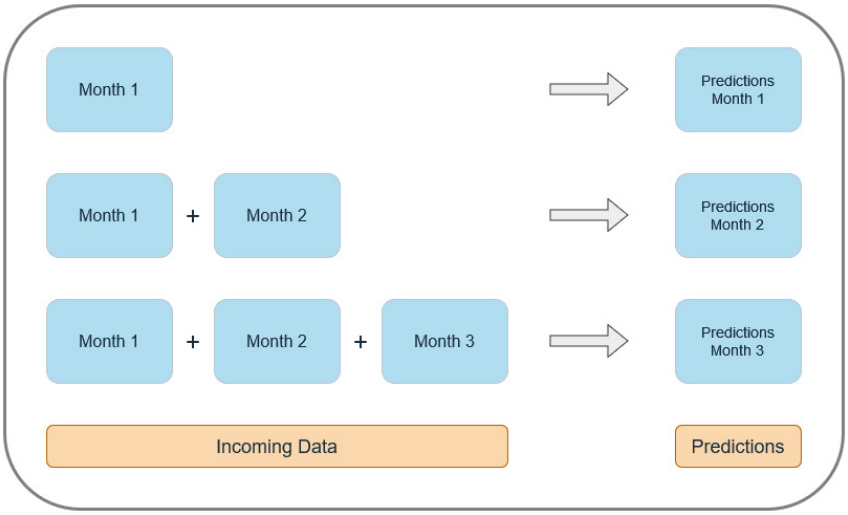

Figure 2.1: Online Learning Diagram.

Application of Online Learning

Online Learning algorithms apply to various types of tasks. In supervised learning, a prevalent objective is classification, where the goal is to predict the categories to which a new data instance belongs. This prediction is based on observations of past training data instance es with known category labels. An example of a well-studied task in online learning is online binary classification, such as spam email filtering, which involves distinguishing between two categories: ”spam” and ”benign” emails. Other types of supervised classification tasks include multiclass classification, multilabel classification, and multiple instance classification(Hoi et al. (2021)).

Bandit learning tasks, also known as multi-armed bandits (MAB), are another category. These algorithms find extensive use in various on-line recommendation systems, including applications such as on-line advertising for internet monetization, product recommendations in e-Commerce, movie recommendations for entertainment, and other personalized systems.

Furthermore, online learning algorithms can be deployed for unsupervised learning tasks, with clustering being a notable example. Clustering, or cluster analysis, involves grouping

objects in a way that items within the same cluster exhibit greater similarity to each other than to those in other clusters. In the context of online learning, the focus is on performing incremental cluster analysis in a sequence of instances, a common approach when dealing with data streams (Hoi et al. (2021)).

Big Data Streams refer to massive amounts of data that are generated continuously, exceeding the resources available for processing and storage. Big Data stream learning is the ability to extract relevant information and patterns from big data arriving as streams. Online learning has emerged as a favorable method for acquiring knowledge from continuous or changing data streams, especially in scenarios involving big data, where streams must be processed in real-time and disposed of promptly (Barry et al. (2021)).

The analysis of huge volumes of data has recently been the focus of intense research because such methods could give a competitive advantage to a given company. In today's business landscape, the ability to derive valuable insights from stored data is pivotal for making informed decisions and achieving success. This interest in uncovering insights from different types of data is mirrored across various domains of human endeavor. Additionally, it is important to note that data often arrive continuously in the form of data streams in many of these applications. Illustrative examples include network examination, forecasting financial data, managing traffic, processing sensor measurements, pervasive computing, tracking GPS and mobile devices, analyzing sentiment, and various additional applications (Krawczyk, Leandro, Gama, Stefanowski, and Woźniak (2017)).

2.4.2 Challenges of Data Streams

According to Gama (2010) and Stefanowski (2015), the emergence of data streams presents novel obstacles for machine learning and data mining, as conventional techniques are tailored for static datasets and struggle to efficiently handle rapidly expanding data volumes while also accounting for attributes like the following:

- data items within the stream are received sequentially over time;
- the order in which data items arrive is uncontrollable, necessitating the processing system to remain responsive at all times;
- the size of the data stream may be extensive, potentially infinite, making it impractical to store all data in memory;
- typically, only a single pass over the data stream is feasible, with processed items either discarded or occasionally stored, or aggregated statistics/synopses computed;
- data streams exhibit a rapid arrival rate relative to the processing capabilities of the system;
- data streams are susceptible to changes, with distributions generating examples potentially shifting dynamically;
- labeling data within streams may be costly or even infeasible in some instances, and labeling may not be immediate.

Two fundamental models of data streams are recognized: stationary, where examples are drawn from a fixed but unknown probability distribution, and nonstationary, where data can evolve. In the second case, target concepts (classes of examples) and/or attribute distributions change.

In simpler terms, the underlying concept that generates the data stream changes after a certain period of stability. This phenomenon is known as concept drift, also referred to as covariant shift. Concept drifts are mirrored in incoming instances and diminish the accuracy of classifiers/regression models trained using previous instances. Typical real-life streams affected by concept drift could include Other examples of real-life concept drifts including spam categorization, object positioning, industrial monitoring systems, financial fraud detection, and robotics (Krawczyk et al. (2017)).

Lately, data sources have experienced an unprecedented increase in data generation rates. This rapid generation of continuous streams of information has challenged the storage, computation, and communication capabilities of computing systems. Systems, models, and techniques have been proposed and developed over the past few years to address these challenges (Domingos and Hulten (2000)).

Research problems and challenges that have arisen in mining data streams have solutions using well-established statistical and computational approaches. We can categorize these solutions into data-based and task-based ones. In data-centric approaches, the objective is to analyze only a portion of the entire dataset or to reshape the data either vertically or horizontally into a more compact representation. In contrast, in solutions centered on specific tasks, computational theory methods have been embraced to achieve efficient use of time and space. Data stream mining has attracted considerable interest from the data mining community in recent years. Numerous algorithms have been suggested to extract insights from streaming data, employing techniques such as clustering, classification, frequency counting, and time series analysis (Gaber, Arkady, and Shonali (2005)).

Many online machine learning models, which are continually updated by processing new data, face challenges in direct application to certain banking applications. In cases like network anomaly detection, data instances are generated independently from various devices or sensors. This requires preprocessing and reordering of the streams by sensors before feeding them into online models.

A good example is Half-Space Trees for fast online anomaly detection. These trees update the mass value (count of observations traversing a node during training) without altering the splitting conditions or the features of the node itself. However, in some banking scenarios, the number and types of features are not fixed (independent and identically distributed - IID) and may change over time. Successive transactions might not share the same features. Consequently, ensemble methods based on a fixed-dimensional size would be ineffective for such dynamic banking applications (Barry et al. (2021)).

2.4.3 Summary

The use of machine learning (ML) in the banking sector has improved efficiency and accuracy, playing a crucial role in various applications such as fraud detection, customer recommendation, and risk management. However, this technological advancement brings critical considerations related to fairness and trust, as biased models can lead to financial losses and erode customer trust. Fairness in banking services and ML is vital for maintaining customer

satisfaction.

Fairness in machine learning is a complex and context-dependent construct. Fairness metrics should be used to address these issues. Bias mitigation strategies include pre-processing (cleaning biased data), in-processing (incorporating fairness during model training), and post-processing (adjusting outcomes of trained models). Evaluating bias is crucial for achieving fairness, although the dynamic nature of data streams in banking adds complexity to ML applications.

The next chapter will dive into methodologies for developing and implementing fairness metrics in machine learning, focusing on measuring and mitigating bias. It will emphasize the importance of fairness in dynamic and evolving data environments, particularly in the banking sector. Practical applications and implications of fairness-aware models in areas like credit allocation and risk management will be explored.

Chapter 3

Related Work

This chapter focuses on the use of machine learning and fairness within the banking sector, highlighting the importance of unbiased algorithms in financial services. The chapter sets the stage for understanding the practical and ethical implications of employing machine learning models, in an online learning environment. It underscores the necessity of developing fairness metrics to evaluate these models and ensure equitable treatment of all customers. The ultimate aim is to assess algorithmic performance and fairness, contributing academically and practically to the field.

It also proposes and presents two fairness metrics for regression models applied to data streams: the Correlation Coefficient of Residuals and Prediction Distribution Analysis. The chapter emphasizes the necessity of these metrics to ensure fairness in machine learning models used in banking, given the dynamic nature of data streams and the critical importance of unbiased decision-making in financial services.

3.1 Fairness in Banking

Banking is an environment where trust is considered important for customer engagement (Kuneva (2009)) Given that many products supplied by banks also involve significant risk, the presence of trust is important if customers are to engage with providers with confidence. In the context of buyer-seller relationships in banking, customers are uncertain about the outcomes of their exchanges with financial service institutions. As a result, customers tend to develop trustworthiness perceptions about the financial service institution based on their perceptions of fairness. (Roy et al. (2015)).

At the heart of the customer-bank relationship lies the concept of fairness, a key factor that influences customer trust and satisfaction. A study by Gokmenoglu and Amir (2021) highlights the importance of perceived fairness and trustworthiness in fostering customer trust within the banking sector. Fairness is essential for traditional and online banking services, with a focus on procedural, distributive, and interactional fairness as critical indicators of general fairness. Customer attitudes toward fairness significantly influence their choices within the service industry.

The notion of fairness in the service industry is of significant importance in shaping the dynamics between customers and service providers. It is widely acknowledged as a crucial element in fostering and strengthening the bond between those who trust (trustor) and those who trust (trustee) according to Gokmenoglu and Amir (2021). It is considered a key fun-

damental within satisfactory customer relations both for traditional banking, where fairness has a significant effect on customer loyalty (Kaura, Durga Prasad, and Sharma (2015)) and in online banking services, where the importance of fairness is still not to be overlooked (Zhu and Chen (2012)).

The importance of procedural fairness cannot be overstated, especially in elevating customer contentment and confidence, which in turn fosters lasting connections and encourages repeated transactions. To ensure fairness, financial institutions must interact with customers in a manner that is respectful, unwavering, and precise. This approach not only positively influences the financial success of the banking sector, but also strengthens the trust customers place in the industry (Carr (2007)).

Both distributive and interactional fairness contribute positively to the establishment of trust. This suggests that improving perceived fairness, as perceived by customers, will lead to greater trust within the banking sector. In turn, this benefits the financial service sector as greater perceived trustworthiness strengthens customer trust, expanding the volume of customer interactions between the trustor and the trustee. Banks must ensure that their customers feel as if they have been treated with the utmost respect and receive the same regard as other customers throughout the social exchange process. Procedural fairness also shows a significant positive relationship with customer trust directly (Roy et al. (2015)).

Thus, customer trust within the banking sector can be enhanced by providers making sure they communicate and resolve any disputes with care, also taking into account their customers' comments and concerns about the services offered. Thus, it is evident that improvements in the fairness of the banking sector will provide positive ramifications with respect to the commercial aspects of bank profits, as well as conclusively improve welfare by improving customer trust (Carr (2007)).

3.2 Machine Learning in Banking

Machine Learning (ML) and Artificial Intelligence (AI) have revolutionized the global banking and financial industries. The banking sector has accelerated changes in processes through their use. Machine learning examines past data and behaviors to forecast patterns and assist decision-making in this domain. With financial institutions increasingly adopting AI and machine learning, we witness the expanding range of applications for these technologies every day. However, the potential dangers associated with these two factors are also increasing. Some of its main uses are in (a) Anti-Money Laundering and Fraud Pattern Detection, (b) Personalized Banking and Automation, (c) Customer Recommendation, (d) Risk Management (Donepudi (2017)), and (e) Prevention of Churn Detection (Rahman and Kumar (2020)).

Banks encounter primarily risks such as credit, market, and operational risks, alongside other forms of risk such as liquidity, business, and reputational risks. Banks are actively involved in risk management to oversee, control, and assess these risks. Machine learning is a versatile tool that can be applied to various challenges, particularly in domains based on data interpretation and actionable insights. It provides the ability to identify significant patterns within datasets. It has become a common tool for almost any task faced with the requirement of extracting meaningful information from data sets. When tasked with extracting valuable insights from data, along with the intricate nature of the patterns to be analyzed, a programmer might struggle to offer explicit and comprehensive instructions for the execution process

(Leo, Suneel, and K. (2019)).

The use of new technology and methods to support risk assessment tasks is a growing trend in this sector. Lately, machine learning (ML) techniques and, to a certain degree, deep learning (DL) have been employed to evaluate credit risk and, on a broader scale, forecast potential bank collapses. Currently, conventional statistical techniques remain widely employed for this objective. However, machine learning techniques are surpassing traditional methods by allowing practitioners to model previous decisions, take advantage of them for alternative scenarios, and anticipate future unpredictable events (Guerra and Mauro (2021)).

Banking regulators have echoed many of the fairness concerns regarding machine learning expressed in the media, which partly underpin this conference. Specifically, regulators are concerned about the insufficient level of explanation provided by machine learning processes for their decisions. In contrast, regulators are also intrigued by these models because of their increased accuracy. An objective of regulators is to ensure that the appropriate products are provided to the suitable customers, given the credit profile of the customer.

The existence of discrimination requires not only statistical significance but also consideration of 'practical significance.' Practical significance is a threshold that is set by a court or regulator to determine whether the difference is meaningful. The idea is that nearly any difference can be statistically significant when using thousands or millions of observations. However, that difference may not reflect an outcome that is appreciably different for the protected class (Schmidt, Bernard, and Syeed (2018)).

3.3 Fairness Metrics for Regression on Data Streams

In this chapter are presented two possible Fairness Metrics to be used in the context of regression models. Fairness metrics are studied more widely in classification models. Its use on regression is more recent and the two solutions presented are adaptations of metrics used on classification problems.

3.3.1 Correlation Coefficient of Residuals

The correlation coefficient of residuals with sensitive features measures the relationship between the errors (residuals) of a model's predictions and certain sensitive attributes (such as age, income, and employment status). This can help identify whether the model is biased against specific groups.

Theoretical Explanation

- **Residuals:** In regression, residuals are the differences between the observed values (true values) and the predicted values. Mathematically, for a data point i, the residual e_i is:

  $$e_i = y_i - \hat{y}_i$$

  where y_i is the true value and $\hat{y}_i$ is the predicted value.

- **Sensitive Features:** attributes that could be the basis for unfair treatment, such as age, race, gender, income, and employment status. The goal is to ensure that the model does not systematically favor or disadvantage any particular group based on these features.

- **Correlation Coefficient:** The Pearson correlation coefficient is a measure of the linear relationship between two variables. It ranges from -1 to 1, where: 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship and 0 indicates no linear relationship.

  The formula for the Pearson correlation coefficient r between two variables X and Y is:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}}$$

  where $\bar{X}$ and $\bar{Y}$ are the means of X and Y, respectively.

A significant correlation between the residuals and a sensitive variable indicates that the model errors are not independent of that feature. A strong positive correlation between residuals and income may indicate that the model overestimates or underestimates credit risk scores for people in a given income range. A low or zero correlation suggests that the performance is not skewed towards any specific group described by the sensitive feature, as it shows that the residuals (and hence the model errors) are independent of the feature.

Using the correlation coefficient of residuals as a fairness metric helps to identify indirect biases. Although the model might not directly use sensitive features for predictions (e.g., age, race), the presence of correlation in residuals could reveal hidden biases due to indirect relationships. This is particularly useful in understanding and mitigating disparate impact, ensuring that model errors do not systematically favor or disadvantage any particular group.
.

3.3.2 Prediction Distribution Analysis

It is a fairness metric that evaluates how model predictions are split among various groups based on sensitive features (such as age, income, and employment status). The objective is to determine whether the model predicts responses that are biased in favor of certain groups, as this may be an indicator of unfair treatment.

Key Concepts

- **Sensitive Features**: Attributes that can define groups for which fairness needs to be ensured, such as age, income, gender, race, and employment status.
- **Prediction Distribution**: The set of predictions made by the model for a specific group. By analyzing the distribution of predictions, one can identify potential disparities between different groups.
- **Group Definition**:
  - **Continuous Features**: For a continuous feature X (e.g., age, income), groups can be defined by a threshold t. For example:

    Group 0: $X < t$

    Group 1: $X \geq t$

  - **Binary or Categorical Features**: For a binary feature Z (e.g., employment status), groups are defined by the feature values:

$$\text{Group 0: } Z = 0$$

$$\text{Group 1: } Z = 1$$

- **Statistical Measures**: To compare distributions, statistical measures such as mean and median predictions within each group are used.

Methodology

- **Group Segmentation**:
  - For continuous features, define groups based on a threshold. For example, for age, you might define two groups: those below a certain age and those above.
  - For binary features, define groups directly by the feature values. For example, for employment status, one group could be employed, and the other could be unemployed.
- **Calculate Predictions**:
  - Make predictions $\hat{y}_i$ for each data point i.
  - Separate the predictions into groups based on the sensitive feature values.
- **Analyze Distribution**:
  - Calculate statistical measures such as the mean and median of predictions for each group.

Metrics

- **Mean Prediction**: The average predicted value within each group. For a group G with n members, the mean prediction μ_G is calculated as:

$$\mu_G = \frac{1}{n} \sum_{i \in G} \hat{y}_i$$

- **Median Prediction**: The median predicted value within each group.

These metrics help identify discrepancies in the model's predictions for different groups. If the mean or median predictions differ significantly between the groups, it suggests that the model might be biased.

Example

Consider a model that predicts credit risk scores and we want to ensure fairness concerning age and income. For age, we might use a threshold of 50 years to define two groups. For income, we use a threshold to define low and high-income groups.

Prediction Distribution Analysis for Age

- **Group 0**: Customers under 50 years of age.
  - Mean Prediction:
$$\mu_{G_0} = \frac{1}{n_0} \sum_{i \in G_0} \hat{y}_i$$
  - Median Prediction:
$$\text{median}_{G_0} = \text{median}(\{\hat{y}_i \mid i \in G_0\})$$
- **Group 1**: Customers 50 years or older.
  - Mean Prediction:
$$\mu_{G_1} = \frac{1}{n_1} \sum_{i \in G_1} \hat{y}_i$$
  - Median Prediction:
$$\text{median}_{G_1} = \text{median}(\{\hat{y}_i \mid i \in G_1\})$$

By comparing these metrics, we can observe whether the model systematically predicts higher or lower credit risk scores for one age group compared to the other. A similar analysis is performed for income and employment status.

3.4 Fairness in Banking Data Streams - Research Gap Identification

Fairness in banking data streams is an intricate challenge. To preserve fairness in the banking system, novel methods are required in light of the constantly changing nature of data and the need for rapid alternatives. Achieving fairness and avoiding discrimination becomes crucial as financial institutions apply machine learning for customer service, predictive analytics, and risk assessment.

This chapter offers an analysis of fairness and trust in the banking industry from a technical and customer-focused standpoint. It also highlights the importance of addressing fairness in machine learning applications in data streams.

However, there is not sufficient literature on the matter of how to achieve fairness in a banking data stream environment, especially when utilizing regression models. This dissertation aims to elucidate this.

Chapter 4

Experiments and Results

Chapter 4 presents machine learning analysis using authentic datasets, including fairness checks. It emphasizes the importance of using large and domain-specific tabular datasets that reflect real-world and decision-making scenarios. The chapter also discusses the need for current and diverse data to explore newer solutions.

The chapter describes the characteristics of the dataset chosen for the study. The exploratory data analysis phase examines the structure of the dataset. Next, we present the Python library used for the analysis. Finally, we conduct proper ML analysis in parallel with the fairness metrics tests, thus verifying the algorithms with the least bias.

4.1 Data Description and Analysis

4.1.1 Selection of the Dataset

As presented in Jesus et al. (2022), the evaluation of new ML methods using authentic data sets is crucial to the advancement of ML research. The recognition of fairness evaluation as a fundamental aspect of ML practice is on the rise. According to Friedler et al. (2019), the scarcity of large-scale domain-specific tabular data sets is illustrated, which are crucial for many critical decision-making applications where fairness testing has significant value.

Usually, good datasets for machine learning benchmarks accurately show how a certain target population spreads out and changes over time. They are also very helpful for training machine learning models that are specifically designed for a task. Sizeable datasets derived from real-world applications serve both purposes well, encompassing a diverse range of observations. Consequently, insights gained from benchmarks conducted on such datasets are widely acknowledged for their applicability to real-world production environments(Paullada, Raji, Bender, Denton, and Hanna (2021)).

In the context of Fair Machine Learning, an additional crucial dimension of a dataset is the task's context, particularly in high-impact domains. These are domains where the outcomes of machine learning systems significantly impact the lives of the subjects involved. Examples of impactful domains include criminal justice, hiring processes, and financial services.

Another vital consideration for the community is the faithfulness of the dataset settings. In other words, datasets originating from real-world scenarios are highly preferred, especially when machine learning methods are deployed in these contexts. In such cases, the effects of a new method can be evaluated and compared with alternative approaches or even with

the original decision-making policy. These evaluations can then be turned into real-world solutions, such as improving the fairness of models, especially when it concerns historically discriminated groups. Another essential component of the datasets is the presence of protected attributes, as well as considerations about privacy, representation, scale, and recency (Friedler et al. (2019)).

A noticeable trend within the machine learning community is the limited use of datasets for the consistent validation and benchmarking of fairness methods. This trend points to a concentration on a smaller set of datasets for experimental observations. One prevalent challenge is the advancing age of a majority of datasets employed in Fair Machine Learning, coupled with the extensive testing performed on them. This leads to a stagnation of observed results and poses technical considerations for deprecating these datasets (Luccioni et al. (2022)), thereby restricting the exploration of new solutions.

The BAF suite, created by Jesus et al. (2022), includes six datasets derived from an authentic online bank account opening fraud detection dataset, offering a pertinent context for Fair ML applications. In this scenario, model predictions determine the grant or denial of financial services to individuals, potentially exacerbating existing social inequities. Persistent denial of credit access to a particular group may contribute to or widen wealth inequality gaps. In multiple time steps, each variant in the dataset suite introduces predetermined and controlled types of data bias.

The datasets were created using Generative Adversarial Network (GAN) models. They contain a million records (rows) of individual applications. Each record includes thirty different features/columns of the applicants, such as employment status, phone numbers, and some aggregated data (such as how often applications come from certain zip codes). The data spans eight months (indicated in the "month" column). Some of these columns are protected attributes, including the applicant's age, income, and employment status.

4.1.2 Characteristics of the Chosen Dataset

Originally, the datasets were related to the identification of fraudulent online bank account opening applications from a sizable consumer bank.

Each instance (row) in the dataset corresponds to an individual application for an account opening, all of which were submitted through an online platform. The applicants gave their explicit consent to store and process the collected data. The dataset covers eight months of information that spans from February to September.

The *fraud_bool* column (original target column) stores the label for each instance, where a positive label denotes a fraudulent attempt and a negative label denotes a legitimate application. The target column used for this work is the *credit_risk_score*, that the bank gives to the client, according to its internal model.

Jesus et al. (2022) describes in detail the measures taken to ensure privacy on the datasets, including Splitting Strategy, Protected Attributes, Performance Metric, and Fairness Metric. The data also features a careful consideration of bias pattern disparities such as Group Size, Prevalence, and Separability, as explained by Pombal et al. (2022).

The BAF Suite contains a base dataset and five variants, with different characteristics, and the goal is to offer a diverse set of additional algorithmic fairness challenges. Variant V will be the chosen variant for this work. To Jesus et al. (2022), this variant features changes in data bias patterns over time; but with a separability bias component in the first six months that is

removed after that period, where the features that change are related to both the protected attribute and the target. Most models in the real world operate in highly dynamic environments, making them highly susceptible to temporal distribution shifts.

4.1.3 Change on the Target Column

Originally, the Target Variable is the *fraud_bool* column, meaning that the dataset has a classification nature. But given the nature of this work, the selected target column will be *credit_risk_score*, since the analysis will be done using Regression.

Because of that, the rows when fraud_bool is true (meaning that this set of rows features a fraud transaction) will be suppressed from the analysis dataset.

Since these rows are only 11,030 of the total of 1 million (or 1.1%) and the correlation between this and the other numerical and binary columns varies between -0.03 and 0.07 (a non-existent correlation), the suppression of these rows will not interfere with the experiment.

When it comes to loan approvals, predicting each client's risk score has a huge impact on their financial future. A fair regression model makes sure that factors like race, gender, or socioeconomic status don't unfairly influence the predicted score, which can determine the value of the loan a person may borrow.

By using fair machine learning techniques in these regression models, financial institutions can keep fairness and equality while protecting their own interests. These models aim to find the right balance between being accurate and being fair, knowing that even small biases can have significant effects on individuals and society as a whole. Adopting fair ML methods not only supports ethical decisions, but also reduces the risk of damaging a company's reputation and helps build trust with both customers and regulators.

4.1.4 Exploratory Data Analysis

Variables Type

The dataset have a 1-month variable column, 30 predictor variable columns describing each of the transactions, and 1 target variable (*credit_risk_score*). Appendix A provides a detailed description of each variable and its corresponding units. The variables are mostly numerical, as presented in Table A2, with some specific characteristics discussed in the following sections.

Type	Count
Numerical (including target)	20
Categorical	5
Binary	6
Total	31

Table 4.1: Types and count of variables.

Missing Values

There are no missing data values in any of the columns.

Binary Variables

Column	True - Percentage	False - Percentage
email_is_free	48.18	**51.83**
phone_home_valid	**50.62**	49.37
phone_mobile_valid	14.24	**85.76**
*has_other_card*s	24.89	**75.11**
foreign_request	**97.60**	2.40
keep_alive_session	44.37	**55.93**
Total	100.00	100.00

Table 4.2: Percentage of Binary Variables.

According to the table in this section, as well as the histogram featured in the appendix, we can see that there is a somewhat even distribution on the columns *email_is_free*, *phone_home_valid* and *keep_alive_session*.

But there is a majority of *phone_mobile_valid*, *has_other_cards* and *foreign_request* on the false option.

4.1.5 Categorical Variables

On the next three variables analyzed, we can not access the true labels, so the comments will be regarding the distributions.

payment_type and housing_status

Label	Percentage
AB	39.91
AA	24.96
AC	24.65
AD	10.45
AE	0.03
Total	100.00

Table 4.3: Labels on the *payment_type* column.

On the *payment_type* variable, the labels AB, AA, and AC compose 89% of the rows.

In the housing_status variable, we have four different labels with more than 10% each. The sum for BC, BB, BA, and BE have a total of more than 95% of the entries.

employment_status

In opposition to the previous variables, this one features a single label with the majority of the rows (CA), with over two thirds of the entries.

Label	Percentage
BC	33.61
BB	30.20
BA	21.49
BE	11.86
BD	2.65
BF	0.17
BG	0.02
Total	100.00

Table 4.4: Labels on the *housing_status* column.

Label	Percentage
CA	68.51
Others	31.49
Total	100.00

Table 4.5: Labels on the *employment_status* column.

Label	Percentage
Internet	99.23
TELEAPP	0.77
Total	100.00

Table 4.6: Labels on the *source* column.

source and device_os

In these two columns, we have access to the true labels.

In the *source* column, almost 99% used the Internet (using a browser on a computer). Less than 1% used the TELEAPP of the banking institution.

Label	Percentage
linux	33.72
windows	30.42
other	30.00
macintosh	5.02
x11	0.82
Total	100.00

Table 4.7: Labels on the *device_os* column.

In the *device_os* column variable, there is a close distribution with the three bigger options ('linux', 'windows' and 'other'), accounting for more than 94%.

4.1.6 Numerical Variables

Univariate Analysis

In the Appendix section (Table B.1, Figure 4, Figure 5), it is possible to check some of the Univariate Analysis for the Numerical Variables. Histograms and boxplots present the basic stats and distributions.

Regarding Outliers, none of the 20 numerical variables features more than 1% of Lower Outliers.

For upper outliers, we have 10 columns with less than 1% of the total entries, 7 columns between 1% and 10%, and 3 columns with more than 10%.

Column	Lower Outliers
device_distinct_emails_8w	0.70
credit_risk_score (target)	0.31
intended_balcon_amount	0.14
Other 17 Columns	0.00

Table 4.8: Percentage of Lower Outliers.

Column	Upper Outliers
prev_address_months_count	23.83
intended_balcon_amount	24.64
bank_branch_count_8w	19.17
zip_count_4w	6.68
days_since_request	9.13
session_length_in_minutes	7.38
current_address_months_count	3.27
device_distinct_emails_8w	2.96
velocity_6h	1.25
date_of_birth_distinct_emails_4w	1.21
velocity_24h	0.76
credit_risk_score (target)	0.34
customer_age	0.02
Other 8 Columns	0.00

Table 4.9: Percentage of Upper Outliers.

Bivariate Analysis

The core of the bivariate analysis will be the correlation matrix, depicted in the appendix (Figure 6 and Tables C.4 and C.5), calculated only for the numerical variables using the appropriate Pearson correlation coefficient to measure the relationship between two variables. This is still a useful tool to explain how the different pairs of variables relate.

Of the 380 possible pair combinations of numeric variables, only eight of them have a higher correlation than 0.3 (positive or negative).

In addition to the analysis between pairs of variables, it is also worth taking a detailed look at the correlation between predictors and target variables. In analyzing these values, extracted from the correlation matrix, only the column proposed_credit_limit correlates bigger than 0.6.

All the other 18 pars correlate less than 0.2, meaning that none of them is a decent predictor of the target.

4.1.7 Data Pre-Processing

Missing Values

Initially, no specific treatment is required for missing values in the data itself. But some of the columns ('prev_address_months_count', 'current_address_months_count', 'intended_balcon_amount', 'bank_months_count', 'session_length_in_minutes') list MVs as '-1'. So a column was created to flag when a given row contains an MV.

Capping

Some of the numerical columns feature extreme outliers. We decided to cap these values (i.e., defining them as the value on the edge of that calculation). The capped values are those that:

- Lower Cap: Lower whisker of the box-plot or the 5th percentile (whichever is lower);
- Upper Cap: Upper whisker of the box-plot or the 95th percentile (whichever is higher);

Here is a table with the list of columns and the values used to cap:

Column	Lower Cap	Upper Cap
prev_address_months_count	5	215
customer_age	10	80
bank_branch_count_8w	0	1500
zip_count_4w	0	3500
days_since_request	0	4
session_length_in_minutes	0	25
current_address_months_count	0	350
velocity_6h	0	14000
date_of_birth_distinct_emails_4w	0	22
velocity_24h	1297	8500
intended_balcon_amount	0	100
bank_branch_count_8w	0	1500

Table 4.10: Capping Limits.

Scaling

After capping, we scale all numerical values that are still not using the 0-to-1 scale, to avoid distortions in the models.

One Hot Encoding

All categorical columns with unique labels were converted into Boolean columns with two possible values: 1 (if the row label is the same as the original columns) or 0 (if not the same). The columns 'device_distinct_emails_8w', although is numeric, feature only four different unique values: ('-1', '0', '1' and '2'), so we used the same treatment as the categorical ones:

Columns where this process was done:

Column	No of Labels/New Columns
payment_type	5
employment_status	7
source	2
housing_status	7
device_os	5
device_distinct_emails_8w	4

Table 4.11: Categorical Columns for One-Hot Encoding

Type conversions and Rounding

Following the previous treatments, we listed some columns with unique values consisting of '0' and '1' as floats. We converted them all to Boolean. In addition, some of the numerical columns featured 5 or 6 decimals. In some preliminary tests, this caused some models not to work. So, they were all rounded to only three decimals.

4.2 Forecast Model

4.2.1 The River Library

According to Montiel et al. (2021), in machine learning, the conventional method involves processing data in batches or fragments. Batch learning models operate assuming all the data are accessible simultaneously. When a fresh set of data is accessible, these models need to undergo full retraining.

The requirement of having the data readily available is a challenge for real-world applications where data is continuously generated. Besides this, keeping historical data often requires dedicated storage and processing, which can be impractical in some situations, such as saving extensive network logs from a data center.

A different approach is to treat data as a stream, an infinite sequence of items where data is not stored, and to learn from one data sample at a time.

River, a Python library developed as a merging of two previous libraries (*Creme* and *scikit-multiflow*), includes various enhancements over its predecessors, including support for mini-batches, improved processing time, an expanded array of metrics for classification, regression, and clustering, as well as the inclusion of additional clustering methods.

Architecture

River consolidates continuous development efforts into a unified project. It incorporates some core elements written in *Cython* for enhanced performance. Many applications are

accommodated, spanning classification, regression, clustering, representation learning, multilabel and multioutput learning, forecasting, and anomaly detection.

River machine learning models are mixtures that serve as extended classes, mirroring learning tasks such as classification, regression, and clustering. This type of design creates compatibility across the library and makes it easier to make changes to existing models. In addition, the API can be easily integrated into the creation of new models.

The River predictive models perform two functions: learning (or training) using the *learn-one* method, which updates the model's internal state, and predicting, achieved via methods like *predict-one* (for classification, regression, and clustering), *predict-proba-one* (for classification), and *score-one* (for anomaly detection). River also has transformers, which are objects that transform input through the *transform-one* method, with the *"*-one"* suffix denoting that the input is a single data sample.

Pipelines

Pipelines play a crucial role in the River framework, offering a convenient and elegant means to sequence a series of operations, ensuring reproducibility. A pipeline essentially constitutes a list of estimators applied sequentially.

The only requirement is that the initial n-1 steps must be transformers, while the final step can be a regressor, classifier, clusterer, transformer, etc. For example, models such as logistic regression may be sensitive to data scale, making it a best practice to scale the data before feeding them into a linear model.

River supports both incremental and batch learning methods. Instance-incremental methods update their internal state with one sample at a time, whereas batch-incremental learning involves using mini-batches of data.

4.2.2 Architecture for the Project

Setup

Below we have a diagram of the Architecture that will be used to perform the analysis, which will be fully detailed in the final version of the dissertation, according to Bifet, Read, Abdessalem, and Holmes (2017).

Description of the Model in Production - Initial Tests

The River library is used for linear regression analysis, following a sequential approach to model training and evaluation. A concise breakdown of the methodology is as follows.

Initially, a pipeline is established using the create_pipeline() function. With that pipeline, a linear regression model instance is started along with the mean squared error (MSE) metric, encapsulating the predictive performance evaluation.

After that, data pre-processing is performed, including the removal of incomplete records from the data set to ensure data integrity. Subsequent operations are performed iteratively for each month (0 to 7 in the case of this dataframe).

Training unfolds iteratively for each month's dataset. A new pipeline and metric instance are instantiated for each iteration, facilitating independent model refinement and performance evaluation. Before model training, the dataset is partitioned into feature vectors (X) and corresponding target variables (y).

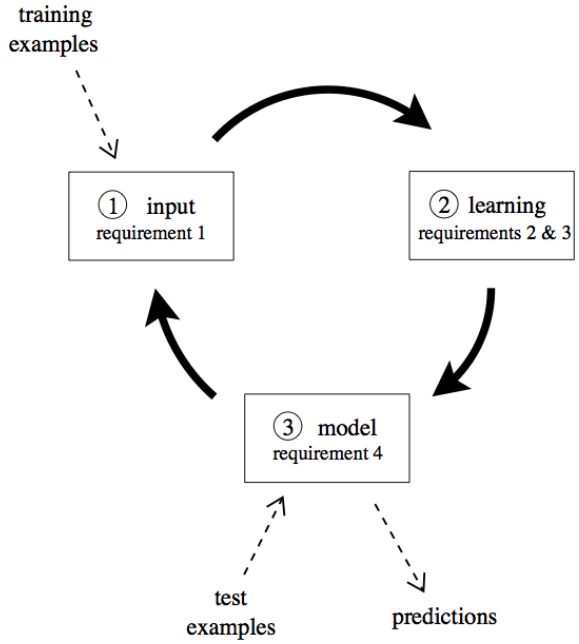

Figure 4.1: ML Data Stream Architecture.

Model refinement is achieved through an incremental learning paradigm in which the model is sequentially updated with each data instance from the current month. This involves predicting the target variables for each instance, followed by the model's adaptation to the observed target value (yi) through the pipeline.learn_one(xi, yi) function call. At the same time, the MSE metric is updated to reflect the model's performance on the current data instance via metric.update(yi, y_pred).

After training the model for each month, the mean squared error (MSE) is calculated to serve as a quantitative measure of predictive accuracy. This evaluation metric is instrumental in assessing the model's efficacy across varying temporal data distributions.

Here is a table with the evolution of the MSE metric on the duration of the analysis:

Month	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
MSE	0.0116	0.0117	0.0106	0.0092	0.0087	0.0089	0.0084	0.0080

Table 4.12: MSE Evolution - Initial Tests

Under the assumption that an Online Learning algorithm improves the model as new data arrive at each subsequent period, this initial model seems to be working fine, according to Hoi et al. (2021) or Barry et al. (2021).

Testing Different Models

River contains a wide range of different models that can be applied. The next study implemented a systematic evaluation process that involved several key components to achieve this. First, a collection of linear and ensemble regression models was selected, including Linear Regression, Passive Aggressive Regressor, Bagging Regressor, Exponentially Weighted Average Regressor, Logistic Regression, Perceptron, OXT Regressor, MLP Regressor, SGT Regressor, Hoeffding Tree and Hoeffding Adaptive Tree.

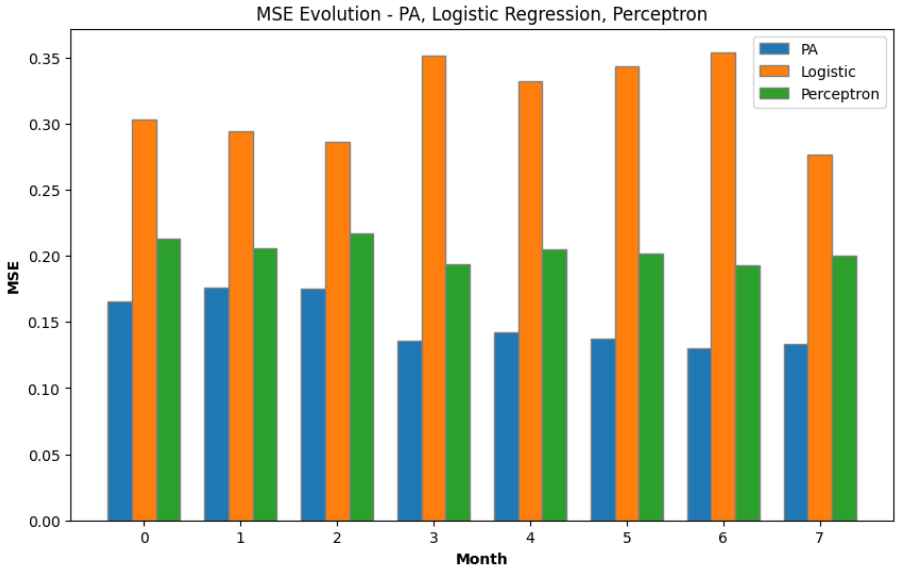

Figure 4.2: MSE Tests - First Group

The process iterated over each model and performed assessments for each monthly interval. The study filtered the dataset within each iteration to isolate the data corresponding to the respective month under evaluation. At this initial stage, the goal was to run the different models using the default parameters whenever possible.

The study concluded with an analysis of the model performance, including MSE values at monthly intervals. In addition, the computational efficiency of each model was evaluated by recording the total time taken for model training and evaluation.

In the next two graphs, a summary of the MSE results of the initial tests is provided using several algorithms. A third graph is provided with the total times for each algorithm.

More detailed tables containing all values are featured in the Appendix section.

A discernible trend emerges in the MSE values, with certain models consistently yielding lower errors compared to others. In particular, linear regression and the bagging regressor exhibit superior predictive performance relative to the PA regressor. Likewise, the Perceptron model tends to demonstrate higher MSE values in comparison to the Exponentially Weighted Average (EWA) and Logistic Regression models.

The initial results also elucidate the computational demands of each model. It becomes evident that certain models require substantially more time for both training and evaluation. Noteworthy examples include the Multi-Layer Perceptron (MLP) and Sparse Group Testing (SGT) models, which consistently manifest longer processing times across all intervals.

4.3 Analysis

Several Fairness Metrics can be used to compare the fairness of the different algorithms. According to Jesus et al. (2022), the sensitive features of this database are 'customer_age', 'income', and 'employment_status'. So, different methods of fairness were used.

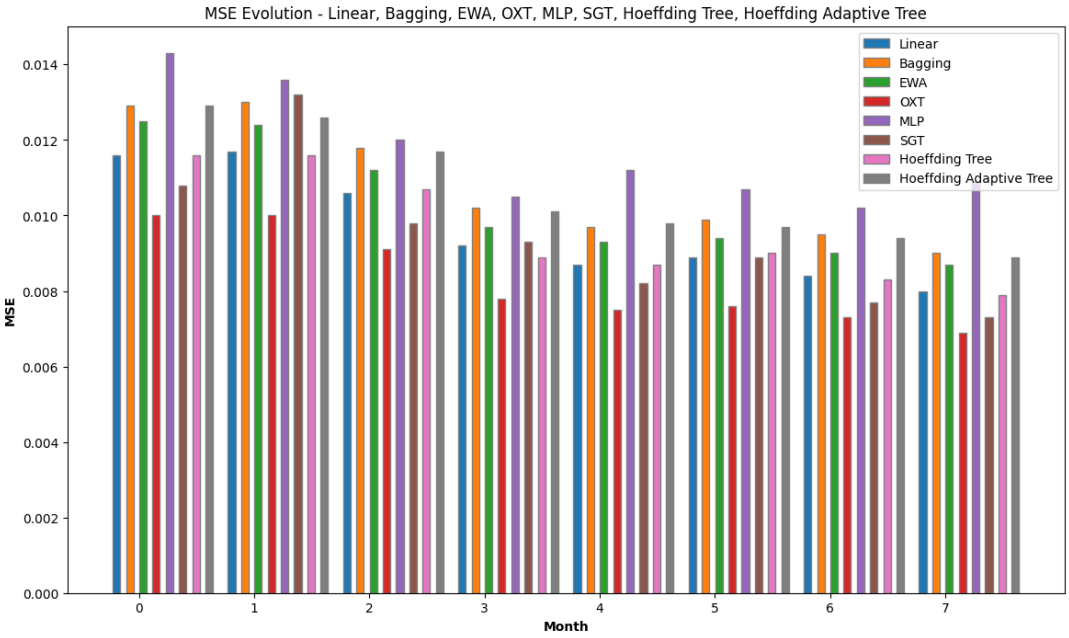

Figure 4.3: MSE Tests - Second Group

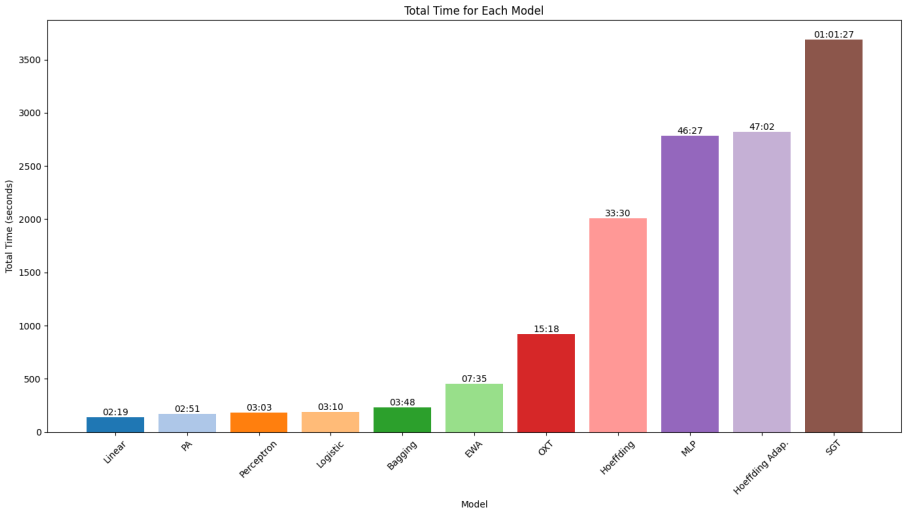

Figure 4.4: Model Times

4.3.1 Correlation Coefficient of Residuals

The first of them was the calculation of the correlation coefficient between the residuals (the differences between true target values and predicted values) on the sensitive features for each month.

Upon training the model incrementally over successive months, the script systematically computes the residuals, representing the disparities between the actual target values and the model's predictions. Subsequently, it calculates the correlation coefficients between these residuals and various sensitive features. These coefficients show how much the model's predictive errors correlate with specific demographic attributes.

By scrutinizing the correlation coefficients across different sensitive features for each month, the script offers insights into the presence and magnitude of biases within the model's predictions. A higher correlation coefficient suggests a stronger association between predictive errors and the corresponding demographic attribute, implying a potential bias. In contrast, a lower correlation coefficient indicates a lesser correlation, suggesting a more fair distribution of predictive errors between different demographic groups.

Detailed results for the tests are presented in the Appendix. In the following table, we can see the values from the fairness metrics correlation coefficient:

Algorithm	Average Coef	Error M7	Time
Hoeffding	0,0126 (1)	0.0079 (3)	0:44:23.636 (8)
EWA	0.0187 (2)	0.0087 (4)	0:06:26.479 (6)
OXT	0.0206 (3)	0.0069 (1)	0:15:15.769 (7)
Bagging	0.0231 (4)	0.0090 (6)	0:05:13.220 (5)
Linear	0.0235 (5)	0.1333 (8)	0:01:28.688 (1)
Hoeffding Adaptive	0,0254 (6)	0.0087 (4)	6:13:04.506 (10)
MLP	0.0297 (7)	0.0109 (7)	1:32:25.784 (10)
SGT	0.0455 (8)	0.0073 (2)	1:03:21.223 (9)
Logistic	0.0771 (9)	0.2766 (11)	0:02:02.006 (2)
Perceptron	0.0903 (10)	0.2000 (10)	0:02:07.243 (3)
PA	0.0996 (11)	0.1333 (8)	0:03:00.729 (4)

Table 4.13: Avg. Models - Correlation Coefficient

The Average Coefficient represents the correlation between the residuals of the differences between the true target values and the predicted values for the sensitive features over each month. A lower coefficient indicates reduced bias and suggests fairer predictions.

Regarding each of the algorithms:

- **EWA:** demonstrates a good balance with a low Average Coefficient (0.0187), low Error M7 (0.0087), and moderate computation time (0:06:26.479). Thus, EWA emerges as a well-rounded option, excelling in fairness, accuracy, and efficiency.
- **OXT:** exhibits excellent accuracy (Error M7 = 0.0069) and fairness (Average Coefficient = 0.0206), although it is less efficient computationally (0:15:15.769).
- **Bagging:** it performs well in terms of fairness (0.0231) and efficiency (0:05:13.220), with a slightly higher error rate (0.0090) compared to EWA and OXT.
- **SGT:** provides good accuracy (Error M7 = 0.0073) but falls short in fairness (0.0455) and efficiency (1:03:21.223).

- **MLP:** shows a relatively high error (0.0109) and significantly longer computation time (1:32:25.784), making it a less favorable option despite its reasonable fairness (0.0297).
- **Linear and Logistic:** While Linear is notably efficient (0:01:28.688), it performs poorly in accuracy (Error M7 = 0.1333) and fairness (0.0235). Logistic regression underperforms across all metrics.
- **Perceptron and PA:** exhibit high errors and coefficients, with PA having a considerable computation time (0:03:00.729).
- **Hoeffding and Hoeffding Adaptive:** Hoeffding shows the smallest coefficient of residuals (0,0126), and both show small errors at the end of the period (0.0079 and 0.0087). The downside is the high processing time (0:44:23.636 for Hoeffding and 6:13:04.506 for Hoeffding Adaptive).

The analysis reveals that EWA and OXT are the most balanced algorithms, providing a favorable combination of fairness, accuracy, and computational efficiency. EWA, in particular, is slightly more advantageous due to its shorter computation time. Bagging also presents a strong case due to its efficient and fair performance, albeit with a marginally higher error rate. In contrast, despite having certain strengths, SGT, MLP, and PA generally lag when all performance metrics are considered. So, EWA and OXT are the recommended algorithms for applications that require balanced performance.

4.3.2 Prediction Distribution Analysis

Another possible fairness metric is the Prediction Distribution Analysis, which compares the distribution of predictions for different groups. This can help to see whether one group systematically receives higher or lower predictions. That analysis evaluates how a machine learning model's predictions differ across groups defined by a sensitive feature. It helps identify potential biases in the model's predictions.

It uses a threshold, a parameter that is used to split the continuous sensitive feature into two groups. For each group defined in the previous step, the function calculates the mean and median of the model's predictions:

- mean_pred_0 and median_pred_0 are the mean and median predictions for group_0.
- mean_pred_1 and median_pred_1 are the mean and median predictions for group_1.

Then, the function returns a dictionary containing both groups' calculated mean and median predictions. This output can be used to compare the prediction distributions and assess whether the model treats different groups fairly. These statistics provide insight into how the model's predictions vary between the two groups, which is important for identifying disparities. In the following table, it is possible to see the summary of the results:

According to this analysis:

- **EWA:** a well-rounded option, presenting one of the smallest errors but tolerable mean and median on the Prediction Distribution Analysis (PDA). Computational time is on the middle section.

Algorithm	Mean	Median	Error M7	Time
Logistic	0.0026 (1)	0.0139 (2)	0.2766 (11)	0:04:27.852 (4)
SGT	0.0179 (2)	0.0028 (1)	0.0073 (2)	0:58:50.992 (10)
EWA	0.0320 (3)	0.0282 (8)	0.0087 (6)	0:09:14.056 (6)
MLP	0.0240 (4)	0.0234 (5)	0.0109 (8)	0:50:40.538 (9)
OXT	0.0247 (5)	0.0229 (3)	0.0069 (1)	0:18:05.997 (7)
Hoeffding	0,0294 (6)	0,0261 (6)	0.0079 (3)	0:44:39.841 (8)
Hoeffding Adaptive	0,0287 (7)	0,0229 (3)	0.0084 (5)	4:31:02.560 (11)
Bagging	0.0309 (8)	0.0274 (7)	0.0090 (7)	0:04:17.811 (3)
Linear	0.0336 (9)	0.0282 (8)	0.0080 (4)	0:02:01.450 (1)
PA	0.0360 (10)	0.0383 (10)	0.1333 (9)	0:02:28.892 (2)
Perceptron	0.1104 (11)	0.2361 (11)	0.2000 (10)	0:04:28.787 (5)

Table 4.14: Avg. Models - Prediction Distribution Analysis

- **OXT:** The best one regarding the Error of the model, with mid results on the PDA. Computation time is one of the worst (but less than half of the worst options).
- **Bagging:** The model results and the PDA performance are on the middle part of the table, but it has one of the best computation times.
- **SGT:** The second-best model performance and the PDA numbers are near the top (the best one on Mean and the second best on median). But its computation time is the worst.
- **MLP:** middle of the table numbers for model performance and the PDA results. Computation values are the second worst.
- **Linear:** The error of the model is one of the best, but the PDA results are close to the bottom.
- **Logistic:** The model error is the worst of all algorithms, but the PDA values are the best and the second best respectively. Computation time is good (fourth best one).
- **Perceptron and PA:** Although the computation time for these models is very good, the results for both the model performance and the PDA fairness metric are very close to the bottom of the table.
- **Hoeffding and Hoeffding Adaptive:** Both present good results for the median/mean differences and low values for the error on the seventh month. But the processing times are among the worst (Hoeffding Adaptive processing time is 4x bigger than the next-to-last algorithm).

The most well-rounded results for reasonable computation time, model performance, and PDA fairness metric are EWA and OXT. MLP and SGT present good algorithm performance and low PDA mean and median, but high computation time.

4.3.3 Answering the Research Questions

The experiments were performed on the chosen dataset, derived from real-world banking data, have demonstrated that existing machine learning models exhibit varying degrees of bias.

Through the application of fairness metrics such as disparate impact and equal opportunity, it can be observed that certain demographic groups were systematically disadvantaged in the predictions. This bias was most pronounced in scenarios where the data distribution shifted over time, highlighting the susceptibility of the models to temporal bias patterns.

Two fairness metrics were developed to measure this bias. They integrate traditional fairness measures with a temporal component to account for changes in the data distribution. These metrics, when applied over different periods, provided a more comprehensive understanding of how bias evolves and its impact on model predictions.

Comparison of various regression algorithms revealed that the choice of algorithm significantly affects both fairness and accuracy. Although neural networks demonstrated higher predictive accuracy, they were also more prone to bias, particularly in scenarios with imbalanced data distributions. However, simpler models, such as linear regression, showed better fairness outcomes but at the cost of reduced accuracy.

The trade-off between fairness and accuracy highlights the significance of algorithm selection in banking applications, requiring the consideration of both predictive performance and ethical considerations. No single algorithm outperformed others in all aspects, highlighting the need for context-dependent regression model selection that takes into account the task's specific fairness and accuracy requirements.

Implementing fairness-aware machine learning models in the banking sector, particularly for credit allocation and risk management, has significant practical implications. The experiments demonstrated that fairness-sensitive models could reduce bias in credit scoring, leading to more equitable outcomes for marginalized groups. However, this came with a slight decrease in overall predictive accuracy, which could impact financial performance.

Despite this, the adoption of fairness-based models is crucial to maintaining customer trust and complying with regulatory requirements. The experiments suggest that incorporating fairness into model development processes not only enhances the ethical standards of banking operations but also fosters long-term customer relationships by ensuring more transparent and fair decision-making practices.

A key outcome of this research was the development of fairness metrics that allow the comparison of bias between different algorithms. These metrics, which combine static and dynamic aspects of fairness, provide a nuanced view of how models perform in real-world settings where data distributions are subject to change. By applying this metric, banks could better evaluate and select models that minimize bias, thus promoting more equitable outcomes in their decision-making processes.

Chapter 5

Conclusions

The dynamic context of the banking sector presents new challenges and opportunities for using machine learning in data streams. The nature of data streams in banking, where new transactions, client information, and other data continually flow, makes it a prime use case for machine learning and online learning, resulting in the importance of adapting machine learning models to evolving data and the potential benefits of real-time decision-making in the sector.

The intersection of fairness considerations and machine learning applications in the banking sector presents a complex landscape with significant implications. Fairness, particularly in the procedural, distributive, and interactional dimensions, emerges as a critical factor influencing customer satisfaction, trust, and loyalty within the service industry. The importance of fairness is evident not only in traditional banking, but also in the rapidly evolving landscape of online banking services.

Integrating machine learning into banking introduces a transformative dimension, offering opportunities for yield prediction, risk assessment, and decision-making based on vast datasets. Classification models, exemplified by the distinction between low- and high-risk customers, provide a foundation for credit risk assessment. Regression models extend the application of machine learning by predicting values such as the customer risk score based on customer attributes.

This study aimed to find the algorithms that offered the best balance between regression performance (less error), fairness based on some protected characteristics, and reasonable computation time in a streaming environment, similar to what could be found in a real scenario. The ones with better overall performance are the EWA (Exponentially Weighted Average) Regressor and the OXT (Online Extra Trees) Regressor.

Some of the main limitations of this work may be the limited number of algorithms tested (although the ones used were the ones with the best computation performance on this setup). Future works could use a different type of setup (instead of the Google Colab free tier used for this one) and different datasets (which are always a concern, especially if it is a banking dataset, as previously discussed). Future endeavors could include proposals for newer fairness metrics.

Despite these limitations, to the best of our knowledge, this is one of the first studies related to regression in a streaming setting with fairness measurement, which is an important fact and enriches the study in this area. It also opens the possibility of similar studies in the future that can use this dissertation as a starting point.

As the banking sector continues to embrace machine learning and artificial intelligence, there is a simultaneous need for vigilance in addressing fairness concerns. Regulatory bodies express both enthusiasm for the increased accuracy of machine learning models and concerns about the lack of explanations for their decisions. Striking a balance between efficiency, equity, and fairness remains a priority for the industry.

Weighing the challenges of ensuring fairness, preventing discrimination, and addressing the dynamic nature of data streams against the potential benefits of improved prediction, risk assessment, and decision-making is necessary.

Based on the findings and limitations identified in this study, several avenues for future research emerge that could advance the understanding and application of fairness-aware machine learning in the banking sector. One significant area for future exploration is the development and validation of new fairness metrics tailored specifically for regression models in streaming environments. Although the metrics used in this dissertation provide valuable insights, they may lack the granularity needed to fully capture the complexities of bias in continuous data streams. Research that focuses on creating more nuanced, context-sensitive fairness metrics could lead to more precise and actionable fairness assessments in real-time applications.

Another promising direction for future work lies in expanding the variety of machine learning algorithms tested in streaming scenarios, particularly in the context of large-scale real-world banking data. Although this study focused on a select group of algorithms, the exploration of more diverse model architectures, including deep learning and ensemble methods, could provide deeper insights into their trade-offs between accuracy, fairness, and computational efficiency. Using more advanced computing resources could also make it possible to test more complex models and larger datasets, leading to results that are more useful for large-scale applications.

Finally, there is a critical need for longitudinal studies that examine the long-term impacts of implementing fairness-aware machine learning models in the banking sector. Future research could investigate how these models affect not only customer outcomes and satisfaction over time but also their influence on financial institutions' risk management strategies and overall performance. Such studies could also explore how regulatory frameworks and industry practices evolve in response to the increasing integration of fairness considerations into machine learning models.

Bibliography

Adams, J. S., & Freedman, S. (1976). Equity theory revisited: Comments and annotated bibliography. *Advances in experimental social psychology*, *9*, 43–90.

Alpaydin, E. (2014). *Introduction to machine learning, third edition*.

Barry, M., Albert, B., Raja, C., Jacob, M., & Vinh-Thuy, T. (2021). *Challenges of machine learning for data streams in the banking industry , publisher = springer international publishing* [Generic].

Bifet, A., Read, J., Abdessalem, T., & Holmes, G. (2017). *Moa: Massive online analysis. kdd 2017 hands-on tutorial.* Retrieved from https://moa.cms.waikato.ac.nz/kdd-2017-hands-on-tutorial/ (Accessed: 2024-07-05)

Binns, R. (2020). On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 514–524).

Carr, C. L. (2007). The fairserv model: Consumer reactions to services based on a multidimensional evaluation of service fairness [Journal Article]. *Decision Sciences*, *38*(1), 107-130. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5915.2007.00150.x,eprint=https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-5915.2007.00150.x doi: https://doi.org/10.1111/j.1540-5915.2007.00150.x

Castelnovo, A., Riccardo, C., Del, G. G., Greta, G., Aisha, N., Daniele, R., & San, M. G. B. (2020). *Befair: Addressing fairness in the banking sector* [Conference Paper]. doi: 10.1109/BigData50022.2020.9377894

Corbett-Davies, S., & Goel, S. (2018). The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the sixth acm sigkdd international conference on knowledge discovery and data mining* (pp. 71–80).

Donepudi, P. (2017). Machine learning and artificial intelligence in banking [Journal Article]. *Engineering International*, *5*, 83-86. doi: 10.18034/ei.v5i2.490

Dwork, C., Moritz, H., Toniann, P., Omer, R., & Richard, Z. (2012). *Fairness through awareness* [Conference Paper].

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 329–338).

Gaber, M. M., Arkady, Z., & Shonali, K. (2005). Mining data streams: a review [Journal Article]. *SIGMOD Rec.*, *34*(2), 18–26 , ISSN = 0163-5808 , DOI = 10.1145/1083784.1083789. Retrieved from https://doi.org/10.1145/1083784.1083789

Gama, J. (2010). *Knowledge discovery from data streams*. CRC Press.

Garofalakis, M., Gehrke, J., & Rastogi, R. (2016). *Data stream management: Processing high-speed data streams* [Book]. Cham: Springer. doi: 10.1007/978-3-540-28608-0

Gokmenoglu, K. K., & Amir, A. (2021). The impact of perceived fairness and trustworthiness on customer trust within the banking sector. *Journal of Relationship Marketing*, *20*(3), 241–260.

Guerra, P., & Mauro, C. (2021). Machine learning applied to banking supervision a literature review [Journal Article]. *Risks*, *9*(7), 136 , ISSN = 2227-9091. Retrieved from https://www.mdpi.com/2227-9091/9/7/136

Hoi, S. C., Sahoo, D., Lu, J., & Zhao, P. (2021). Online learning: A comprehensive survey. *Neurocomputing*, *459*, 249–289.

Jesus, S., José, P., Duarte, A., André, C., Pedro, S., Rita, R., … Pedro, B. (2022). Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation [Journal Article]. *Advances in Neural Information Processing Systems*, *35*, 33563-33575.

Kaura, V., Durga Prasad, C. S., & Sharma, S. (2015). Service quality, service convenience, price and fairness, customer loyalty, and the mediating role of customer satisfaction. *International journal of bank marketing*, *33*(4), 404–422.

Krawczyk, B., Leandro, L. M., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey [Journal Article]. *Information Fusion*, *37*, 132-156. Retrieved from https://www.sciencedirect.com/science/article/pii/S1566253516302329 doi: https://doi.org/10.1016/j.inffus.2017.02.004

Kuneva, M. (2009). Restoring consumer trust in retail financial services. In *Esbg conference retail banking in europe–the way forward, lessons from the crisis and priorities for the future, european consumer commissioner, speech/09/403, brüssel* (Vol. 22, p. 2009).

Lemos, R. A. d. L., Silva, T. C., & Tabak, B. M. (2022). Propension to customer churn in a financial institution: a machine learning approach [Journal Article]. *Neural Computing and Applications*, *34*(14), 11751-11768. Retrieved from https://doi.org/10.1007/s00521-022-07067-x

Leo, M., Suneel, S., & K., M. (2019). Machine learning in banking risk management: A literature review [Journal Article]. *Risks*, *7*(1), 29 , ISSN = 2227-9091. Retrieved from https://www.mdpi.com/2227-9091/7/1/29

Le Quy, T., Arjun, R., Vasileios, I., Wenbin, Z., & Eirini, N. (2022). A survey on datasets for fairness-aware machine learning [Journal Article]. *WIREs Data Mining and Knowledge Discovery*, *12*(3), e1452. Retrieved from https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1452,eprint=https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1452 doi: https://doi.org/10.1002/widm.1452

Luccioni, A. S., Corry, F., Sridharan, H., Ananny, M., Schultz, J., & Crawford, K. (2022). A framework for deprecating datasets: Standardizing documentation, identification, and communication. In *Proceedings of the 2022 acm conference on fairness, accountability, and transparency* (pp. 199–212).

Masterson, S. S., Lewis, K., Goldman, B. M., & Taylor, M. S. (2000). Integrating justice and social exchange: The differing effects of fair procedures and treatment on work relationships. *Academy of Management journal*, *43*(4), 738–748.

Mitchell, S., Potash, E., Barocas, S., D'Amour, A., & Lum, K. (2018). Prediction-based decisions and fairness: A catalogue of choices, assumptions, and definitions. *arXiv preprint arXiv:1811.07867*.

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., … Bifet, A. (2021, jan). River: machine learning for streaming data in python. *J. Mach. Learn. Res.*, *22*(1).

Paullada, A., Raji, I. D., Bender, E. M., Denton, E., & Hanna, A. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, *2*(11).

Pombal, J., Cruz, A. F., Bravo, J., Saleiro, P., Figueiredo, M. A., & Bizarro, P. (2022). Understanding unfairness in fraud detection through model and data bias interactions. *arXiv preprint arXiv:2207.06273*.

Rahman, M., & Kumar, V. (2020). Machine learning based customer churn prediction in banking. In *2020 4th international conference on electronics, communication and aerospace technology (iceca)* (pp. 1196–1201).

Roy, S. K., Devlin, J. F., & Sekhon, H. (2015). The impact of fairness on trustworthiness and trust in banking [Journal Article]. *Journal of Marketing Management*, *31*(9-10), 996-1017 , DOI = 10.1080/0267257X.2015.1036101. Retrieved from https://doi.org/10.1080/0267257X.2015.1036101,eprint=https://doi.org/10.1080/0267257X.2015.1036101

Schmidt, N., Bernard, S., & Syeed, M. (2018). *How data scientists help regulators and banks ensure fairness when implementing machine learning and artificial intelligence models* (Vol. 19) [Conference Paper].

Smith, A. K., Bolton, R. N., & Wagner, J. (1999). A model of customer satisfaction with service encounters involving failure and recovery. *Journal of marketing research*, *36*(3), 356–372.

Stefanowski, J. (2015). Adaptive ensembles for evolving data streams–combining block-based and online solutions. In *International workshop on new frontiers in mining complex patterns* (pp. 3–16).

Van Klompenburg, T., Kassahun, A., & Catal, C. (2020). Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, *177*, 105709.

Verma, S., & Rubin, J. (2018). Fairness definitions explained. In *Proceedings of the international workshop on software fairness* (pp. 1–7).

Wetsch, L. R. (2006). Trust, satisfaction and loyalty in customer relationship management. *Journal of Relationship Marketing*, *4*(3-4), 29-42. Retrieved from https://doi.org/10.1300/J366v04n03_03 doi: 10.1300/J366v04n03_03

Zhu, Y.-Q., & Chen, H.-G. (2012). Service fairness and customer satisfaction in internet banking: Exploring the mediating effects of trust and customer value. *Internet Research*, *22*(4), 482–498.

Appendix

A Description of Dataset Variables

Variable	Description	Range	Type
credit_risk_score (new target)	Internal score of application risk	[−191, 389]	numeric
fraud_bool (original target)	If the application is fraudulent or not	[0,1]	binary
income	Annual income of the applicant	[0.1, 0.9]	numeric
name_email_similarity	Similarity between email and name	[−1, 380]	numeric
prev_address_months_count	Months in previous registered address	[−1, 380]	numeric
current_address_months_count	Months in currently registered address	[−1, 429]	numeric
customer_age	Age in years, rounded to the decade	[10, 90]	numeric
days_since_request	Days passed since application	[0, 79]	numeric
intended_balcon_amount	Initial transferred amount	[−16, 114]	numeric
velocity_6h	Avg applications in last 6 hours	[−175, 16818]	numeric
velocity_24h	Avg applications in last 24 hours	[1297, 9586]	numeric
velocity_4w	Avg applications in last 4 weeks	[2825, 7020]	numeric
bank_branch_count_8w	Applications in bank branch last 8 weeks	[0, 2404]	numeric
date_of_birth_distinct_emails_4w	Emails with same date of birth last 4 weeks	[0, 39]	numeric
bank_months_count	How old is prev account (if held)	[−1, 32]	numeric
proposed_credit_limit	Proposed credit limit	[200, 2000]	numeric
session_length_in_minutes	Length of user session	[−1, 107]	numeric
device_distinct_emails	Emails in website (last 8 weeks)	[−1, 2]	numeric
device_fraud_count	Fraudulent applications with device	[0,1]	numeric
month	Month where the application was made	[0, 7]	numeric
payment_type	Credit payment plan type	–	categorical
employment_status	Employment status of applicant	–	categorical
housing_status	Current residential status for applicant	–	categorical
source	Online source (Internet of Teleapp)	–	categorical
device_os	Operative system of device of the request	–	categorical
email_is_free	Domain of application email (free or paid)	[0,1]	binary
phone_home_valid	Validity of home phone	[0,1]	binary
phone_mobile_valid	Validity of mobile phone	[0,1]	binary
has_other_cards	If has other cards	[0,1]	binary
foreign_request	If origin country of request is different	[0,1]	binary
keep_alive_session	User option on session logout	[0,1]	binary

Table A.1: Detailed description of the variables used.

B Univariate Analysis

	count	mean	std	min	25%	50%	75%	max
income	1000000.0	0.578958	0.288226	1.000000e-01	0.300000	0.600000	0.800000	0.900000
name_email_similarity	1000000.0	0.487527	0.291367	7.898994e-07	0.214529	0.485893	0.754531	1.000000
prev_address_months_count	1000000.0	14.744036	43.134138	-1.000000e+00	-1.000000	-1.000000	-1.000000	384.000000
current_address_months_count	1000000.0	99.187295	94.070293	-1.000000e+00	27.000000	64.000000	154.000000	426.000000
customer_age	1000000.0	41.349480	13.751920	1.000000e+01	30.000000	50.000000	50.000000	90.000000
days_since_request	1000000.0	0.916476	5.068976	1.414624e-07	0.007451	0.015673	0.026988	77.845545
intended_balcon_amount	1000000.0	8.571482	20.544640	-1.571046e+01	-1.178938	-0.833821	-0.052483	113.086228
zip_count_4w	1000000.0	1517.471615	965.945989	1.000000e+00	885.000000	1208.000000	1846.000000	6830.000000
velocity_6h	1000000.0	5490.939853	2940.124992	-1.436497e+02	3332.981127	5190.722819	7371.559084	16802.052304
velocity_24h	1000000.0	4661.530105	1450.442084	1.297721e+03	3505.061781	4641.570668	5593.342744	9585.100782
velocity_4w	1000000.0	4733.511963	870.556097	2.858746e+03	4238.377282	4813.951362	5331.503654	7019.201030
bank_branch_count_8w	1000000.0	201.084771	473.738222	0.000000e+00	1.000000	10.000000	31.000000	2404.000000
date_of_birth_distinct_emails_4w	1000000.0	7.772255	4.815409	0.000000e+00	4.000000	7.000000	11.000000	37.000000
credit_risk_score	1000000.0	139.295998	71.428176	-1.770000e+02	90.000000	130.000000	188.000000	388.000000
bank_months_count	1000000.0	11.139313	12.124117	-1.000000e+00	1.000000	6.000000	25.000000	32.000000
proposed_credit_limit	1000000.0	551.037350	506.530093	1.900000e+02	200.000000	200.000000	1000.000000	2100.000000
session_length_in_minutes	1000000.0	7.817692	8.259055	-1.000000e+00	3.151601	5.246432	9.362126	83.213536
device_distinct_emails_8w	1000000.0	1.022276	0.192862	-1.000000e+00	1.000000	1.000000	1.000000	2.000000
device_fraud_count	1000000.0	0.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000
month	1000000.0	3.658708	2.116726	0.000000e+00	2.000000	4.000000	5.000000	7.000000

Table B.1: Basic Stats for Numeric Columns.

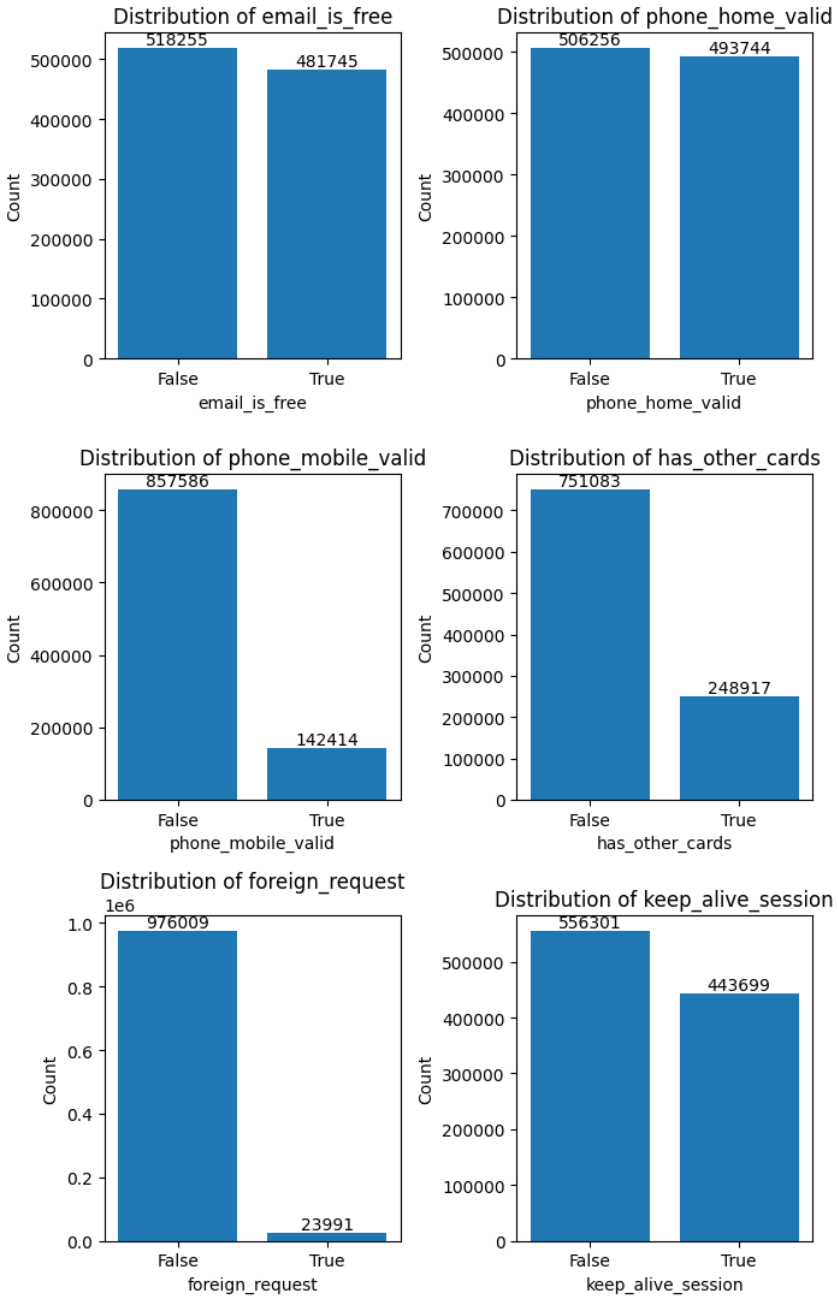

Figure 1: Distribution of Binary Variables.

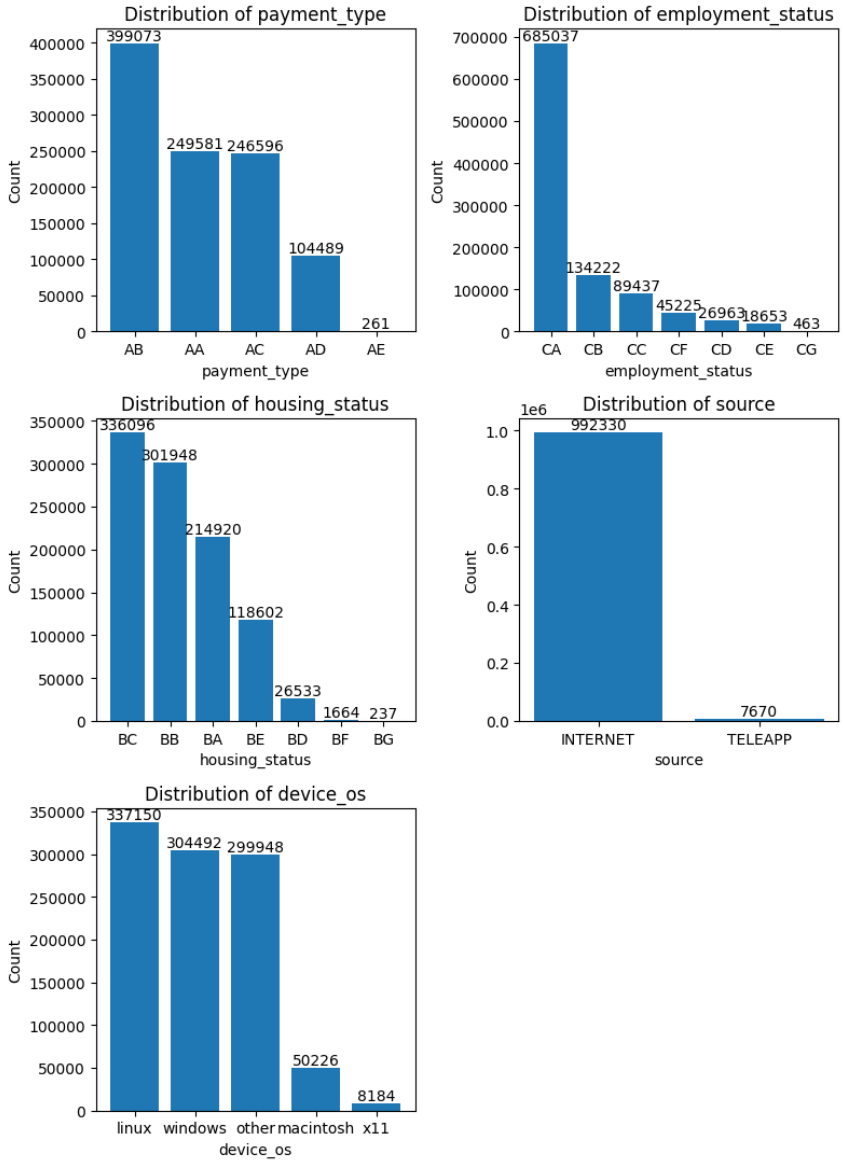

Figure 2: Distribution of Categorical Variables.

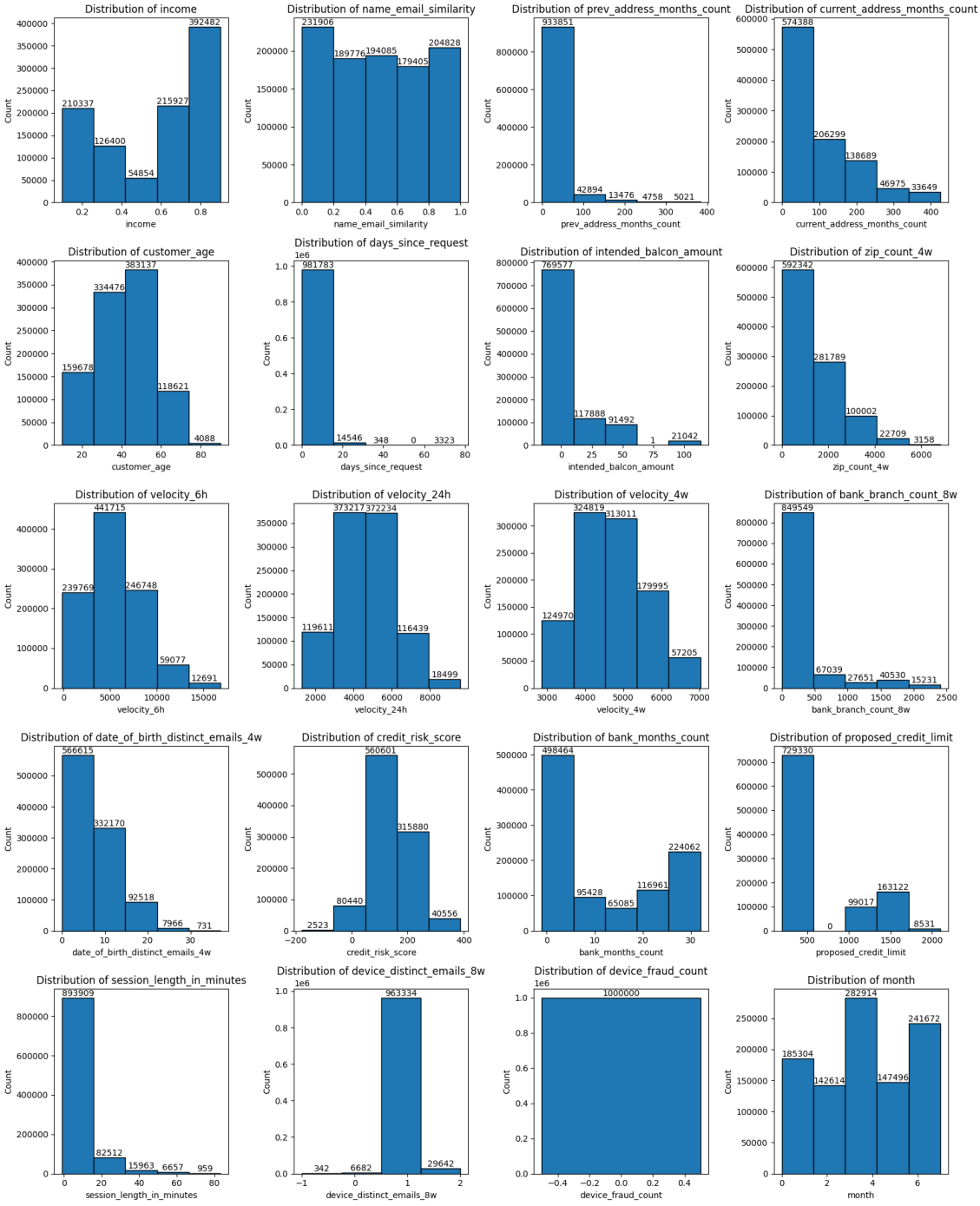

Figure 3: Distribution of Numeric Variables.

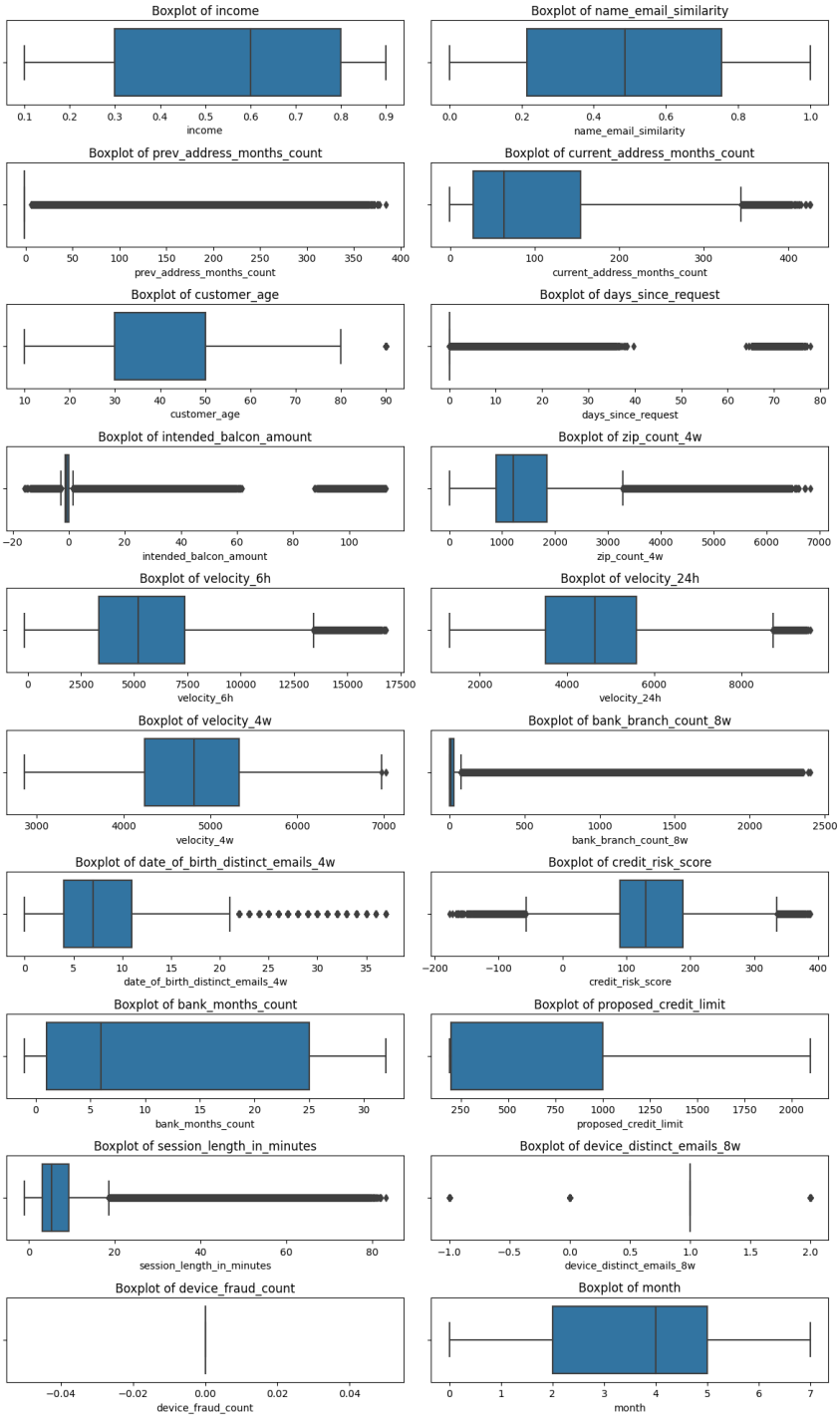

Figure 4: Boxplots of Numeric Variables.

Column	Lower Outliers
device_distinct_emails_8w	0.70
credit_risk_score (target)	0.31
intended_balcon_amount	0.14
income	0.00
name_email_similarity	0.00
prev_address_months_count	0.00
current_address_months_count	0.00
customer_age	0.00
days_since_request	0.00
zip_count_4w	0.00
velocity_6h	0.00
velocity_24h	0.00
velocity_4w	0.00
bank_branch_count_8w	0.00
date_of_birth_distinct_emails_4w	0.00
bank_months_count	0.00
proposed_credit_limit	0.00
session_length_in_minutes	0.00
device_fraud_count	0.00
month	0.00

Table B.2: Percentage of Lower Outliers - Complete Table.

Column	Upper Outliers
prev_address_months_count	23.83
intended_balcon_amount	24.64
bank_branch_count_8w	19.17
zip_count_4w	6.68
days_since_request	9.13
session_length_in_minutes	7.38
current_address_months_count	3.27
device_distinct_emails_8w	2.96
velocity_6h	1.25
date_of_birth_distinct_emails_4w	1.21
velocity_24h	0.76
credit_risk_score (target)	0.34
customer_age	0.02
month	0.00
income	0.00
name_email_similarity	0.00
prev_address_months_count	0.00
velocity_4w	0.00
bank_months_count	0.00
proposed_credit_limit	0.00
device_fraud_count	0.00

Table B.3: Percentage of Upper Outliers - Complete Table.

C Bivariate Analysis

C.1 Correlation Matrix

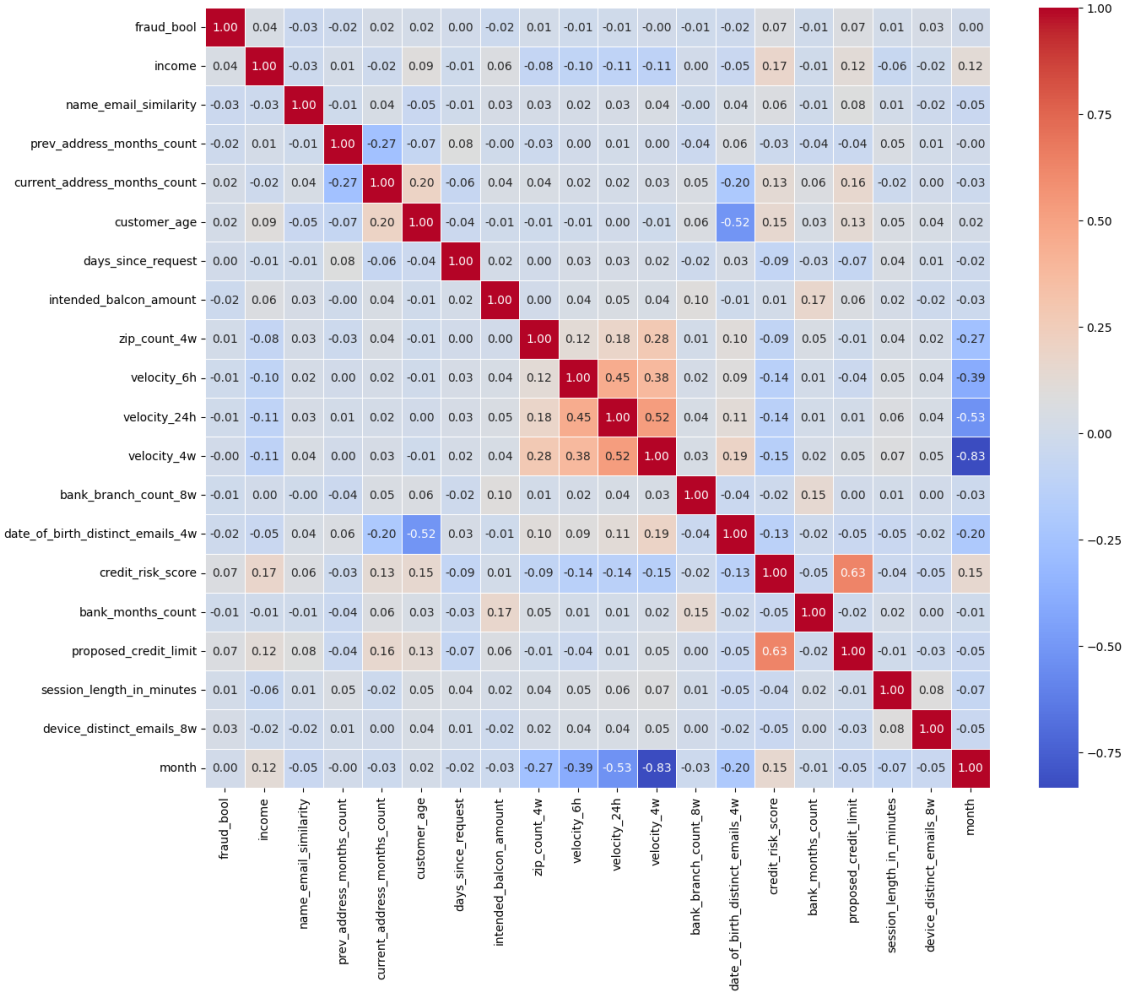

Figure 5: Pearson Correlation Matrix of Numeric Columns.

C.2 Correlation between Pairs of Variables

Variable 1	Variable 2	Correlation
credit_risk_score	*proposed_credit_limit*	0.634
velocity_24h	*velocity_4w*	0.521
velocity_6h	*velocity_24h*	0.450
velocity_6h	*velocity_4w*	0.382
velocity_6h	*month*	-0.393
customer_age	*date_of_birth_distinct_emails_4w*	-0.518
velocity_24h	*month*	-0.530
velocity_4w	*month*	-0.833

Table C.4: Correlations above 30 percent between pairs of variables.

C.3 Correlation with Target Variable

Variable	Correlation
proposed_credit_limit	0.634
income	0.175
month	0.155
customer_age	0.149
current_address_months_count	0.132
name_email_similarity	0.062
intended_balcon_amount	0.011
bank_branch_count_8w	-0.018
prev_address_months_count	-0.031
session_length_in_minutes	-0.037
device_distinct_emails_8w	-0.046
bank_months_count	-0.051
days_since_request	-0.088
zip_count_4w	-0.090
date_of_birth_distinct_emails_4w	-0.125
velocity_24h	-0.137
velocity_6h	-0.140
velocity_4w	-0.149

Table C.5: Correlations with target variable *credit_risk_score*.

D Model Tests

D.1 MSE Tables

Month	Linear	Time	PA	Time	Bagging	Time
0	0.0116	00:11.554	0.1655	00:17.062	0.0129	00:18.043
1	0.0117	00:09.207	0.1763	00:29.631	0.0130	00:24.725
2	0.0106	00:20.128	0.1749	00:24.588	0.0118	00:32.208
3	0.0092	00:19.917	0.1361	00:23.695	0.0102	00:36.721
4	0.0087	00:21.313	0.1419	00:23.120	0.0097	00:28.668
5	0.0089	00:21.261	0.1374	00:19.599	0.0099	00:32.939
6	0.0084	00:19.244	0.1298	00:18.777	0.0095	00:30.716
7	0.0080	00:17.144	0.1333	00:14.537	0.0090	00:24.584
Total		02:19.780		02:51.022		03:48.612

Table D.6: MSE Evolution - Linear Regression, PA Regressor and Bagging Regressor

Month	EWA	Time	Logistic	Time	Perceptron	Time
0	0.0125	00:36.407	0.3029	00:14.257	0.2134	00:14.403
1	0.0124	00:48.693	0.2943	00:19.591	0.2055	00:20.421
2	0.0112	01:05.758	0.2862	00:24.525	0.2169	:00:26.242
3	0.0097	01:10.485	0.3519	00:27.791	0.1940	00:28.313
4	0.0093	00:56.916	0.3321	00:23.243	0.2051	00:23.428
5	0.0094	01:07.004	0.3432	00:26.866	0.2015	00:26.939
6	0.0090	01:02.840	0.3542	00:34.291	0.1930	00:25.209
7	0.0087	00:47.860	0.2766	00:20.144	0.2000	00:18.547
Total		07:35.970		03:10.716		03:03.506

Table D.7: MSE Evolution - EWA, Logistic, Perceptron

Month	OXT	Time	MLP	Time	SGT	Time
0	0.0100	01:05.358	0.0143	03:40.316	0.0108	04:53.245
1	0.0100	01:29.093	0.0136	04:58.732	0.0132	02:21.674
2	0.0091	02:02.191	0.0120	06:38.853	0.0098	:06:25.269
3	0.0078	02:40.147	0.0105	07:14.977	0.0093	08:22.776
4	0.0075	01:46.088	0.0112	05:50.270	0.0082	09:01.423
5	0.0076	02:35.926	0.0107	06:50.292	0.0089	11:40.500
6	0.0073	01:54.388	0.0102	06:18.758	0.0077	10:08.926
7	0.0069	01:44.881	0.0109	04:55.629	0.0073	08:33.649
Total		15:18.080		46:27.832		1:01:27.472

Table D.8: MSE Evolution - OXT, MLP, SGT

Month	Hoeffding Tree	Time	Hoeffding Adaptive Tree	Time
0	0.0116	02:43.353	0.0129	03:45.145
1	0.0116	02:10.268	0.0126	03:07.492
2	0.0107	02:53.997	0.0117	04:43.107
3	0.0089	10:43.839	0.0101	12:39.306
4	0.0087	04:05.111	0.0098	04:50.225
5	0.0090	02:53.788	0.0097	05:04.629
6	0.0083	04:39.430	0.0094	07:32.742
7	0.0079	03:20.221	0.0089	05:20.120
Total		33:30.016		47:02.777

Table D.9: MSE Evolution - Hoeffding Tree, Hoeffding Adaptive Tree

E Fairness Metrics

E.1 Coefficient Correlation of Residuals Values

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0116	{ 'customer_age': -0.0113, 'income': 0.0453, 'employment_status_CA': 0.0399, 'employment_status_CB': -0.0274, 'employment_status_CC': -0.0158, 'employment_status_CD': -0.0249, 'employment_status_CE': -0.0053, 'employment_status_CF': 0.0049, 'employment_status_CG': 0.0025	1	0.0117	{ 'customer_age': 0.0264, 'income': -0.0004, 'employment_status_CA': -0.0501, 'employment_status_CB': -0.0145, 'employment_status_CC': 0.0493, 'employment_status_CD': 0.0326, 'employment_status_CE': 0.0241, 'employment_status_CF': 0.0294, 'employment_status_CG': 0.0046}
2	0.0106	{ 'customer_age': 0.0549, 'income': 0.0086, 'employment_status_CA': -0.0601, 'employment_status_CB': -0.0011, 'employment_status_CC': 0.0769, 'employment_status_CD': -0.0147, 'employment_status_CE': 0.0161, 'employment_status_CF': 0.0312, 'employment_status_CG': 0.0053	3	0.0092	{ 'customer_age': -0.0204, 'income': -0.0117, 'employment_status_CA': 0.0870, 'employment_status_CB': -0.0566, 'employment_status_CC': -0.0850, 'employment_status_CD': 0.0156, 'employment_status_CE': -0.0007, 'employment_status_CF': 0.0095, 'employment_status_CG': 0.0014}
4	0.0087	{ 'customer_age': 0.0019, 'income': 0.0019, 'employment_status_CA': -0.0345, 'employment_status_CB': 0.0238, 'employment_status_CC': -0.0109, 'employment_status_CD': 0.0236, 'employment_status_CE': -0.0150, 'employment_status_CF': 0.0447, 'employment_status_CG': 0.0052	5	0.0089	{ 'customer_age': -0.0077, 'income': -0.0124, 'employment_status_CA': 0.0357, 'employment_status_CB': -0.0381, 'employment_status_CC': -0.0091, 'employment_status_CD': 0.0235, 'employment_status_CE': -0.0100, 'employment_status_CF': -0.0158, 'employment_status_CG': -0.0016}
6	0.0084	{ 'customer_age': 0.0346, 'income': 0.0047, 'employment_status_CA': 0.0022, 'employment_status_CB': 0.0193, 'employment_status_CC': 0.0001, 'employment_status_CD': -0.0105, 'employment_status_CE': -0.0151, 'employment_status_CF': -0.0188, 'employment_status_CG': 0.0020	7	0.0080	{ 'customer_age': 0.0504, 'income': 0.0421, 'employment_status_CA': -0.0712, 'employment_status_CB': 0.0434, 'employment_status_CC': 0.0241, 'employment_status_CD': 0.0192, 'employment_status_CE': 0.0184, 'employment_status_CF': 0.0291, 'employment_status_CG': 0.0068}

Table E.10: Linear Regression Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.1655	{ 'customer_age': 0.0916, 'income': 0.0815, 'employment_status_CA': 0.3006, 'employment_status_CB': -0.2872, 'employment_status_CC': 0.1078, 'employment_status_CD': -0.0430, 'employment_status_CE': -0.1296, 'employment_status_CF': -0.1497, 'employment_status_CG': 0.0104	1	0.1763	{ 'customer_age': -0.0054, 'income': 0.0870, 'employment_status_CA': -0.0765, 'employment_status_CB': 0.0850, 'employment_status_CC': -0.0234, 'employment_status_CD': -0.0206, 'employment_status_CE': -0.0562, 'employment_status_CF': 0.1206, 'employment_status_CG': 0.0116}
2	0.1749	{ 'customer_age': -0.1330, 'income': -0.1979, 'employment_status_CA': -0.0085, 'employment_status_CB': -0.1299, 'employment_status_CC': 0.0164, 'employment_status_CD': 0.0212, 'employment_status_CE': 0.0001, 'employment_status_CF': 0.1963, 'employment_status_CG': -0.0069	3	0.1361	{ 'customer_age': -0.2445, 'income': -0.0782, 'employment_status_CA': 0.1422, 'employment_status_CB': -0.0267, 'employment_status_CC': -0.0624, 'employment_status_CD': -0.0984, 'employment_status_CE': 0.1036, 'employment_status_CF': -0.1565, 'employment_status_CG': 0.0081}
4	0.1419	{ 'customer_age': -0.0328, 'income': 0.0099, 'employment_status_CA': 0.1382, 'employment_status_CB': 0.0805, 'employment_status_CC': -0.2966, 'employment_status_CD': -0.1580, 'employment_status_CE': -0.0493, 'employment_status_CF': 0.1623, 'employment_status_CG': 0.0025	5	0.1374	{ 'customer_age': -0.0052, 'income': 0.0473, 'employment_status_CA': 0.4889, 'employment_status_CB': -0.2377, 'employment_status_CC': -0.3073, 'employment_status_CD': -0.0620, 'employment_status_CE': -0.1631, 'employment_status_CF': -0.1378, 'employment_status_CG': -0.0031}
6	0.1298	{ 'customer_age': -0.0581, 'income': -0.1001, 'employment_status_CA': 0.0610, 'employment_status_CB': -0.0447, 'employment_status_CC': -0.1054, 'employment_status_CD': 0.0162, 'employment_status_CE': 0.0436, 'employment_status_CF': 0.0389, 'employment_status_CG': 0.0255	7	0.1333	{ 'customer_age': -0.0274, 'income': -0.1186, 'employment_status_CA': -0.2610, 'employment_status_CB': 0.2014, 'employment_status_CC': 0.1263, 'employment_status_CD': 0.0695, 'employment_status_CE': -0.0712, 'employment_status_CF': 0.0844, 'employment_status_CG': 0.0139}

Table E.11: PA Algorithm Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0129	{ 'customer_age': 0.0137, 'income': 0.0448, 'employment_status_CA': 0.0247, 'employment_status_CB': -0.0230, 'employment_status_CC': -0.0050, 'employment_status_CD': -0.0224, 'employment_status_CE': -0.0044, 'employment_status_CF': 0.0154, 'employment_status_CG': 0.0003	1	0.0130	{ 'customer_age': 0.0214, 'income': 0.0037, 'employment_status_CA': 0.0010, 'employment_status_CB': -0.0236, 'employment_status_CC': 0.0150, 'employment_status_CD': 0.0108, 'employment_status_CE': 0.0187, 'employment_status_CF': -0.0036, 'employment_status_CG': -0.0007}
2	0.0118	{ 'customer_age': 0.0820, 'income': -0.0344, 'employment_status_CA': -0.0842, 'employment_status_CB': -0.0021, 'employment_status_CC': 0.0891, 'employment_status_CD': 0.0147, 'employment_status_CE': 0.0228, 'employment_status_CF': 0.0423, 'employment_status_CG': 0.0045	3	0.0102	{ 'customer_age': -0.0202, 'income': -0.0065, 'employment_status_CA': 0.0764, 'employment_status_CB': -0.0542, 'employment_status_CC': -0.0665, 'employment_status_CD': 0.0065, 'employment_status_CE': 0.0012, 'employment_status_CF': 0.0083, 'employment_status_CG': 0.0006}
4	0.0097	{ 'customer_age': 0.0167, 'income': 0.0025, 'employment_status_CA': -0.0242, 'employment_status_CB': 0.0257, 'employment_status_CC': -0.0121, 'employment_status_CD': 0.0132, 'employment_status_CE': -0.0094, 'employment_status_CF': 0.0265, 'employment_status_CG': 0.0041	5	0.0099	{ 'customer_age': -0.0048, 'income': 0.0166, 'employment_status_CA': 0.0398, 'employment_status_CB': -0.0254, 'employment_status_CC': -0.0050, 'employment_status_CD': 0.0060, 'employment_status_CE': -0.0217, 'employment_status_CF': -0.0276, 'employment_status_CG': -0.0033}
6	0.0095	{ 'customer_age': 0.0548, 'income': -0.0112, 'employment_status_CA': -0.0308, 'employment_status_CB': 0.0348, 'employment_status_CC': 0.0240, 'employment_status_CD': -0.0040, 'employment_status_CE': -0.0170, 'employment_status_CF': -0.0061, 'employment_status_CG': 0.0038	7	0.0090	{ 'customer_age': 0.0332, 'income': 0.0304, 'employment_status_CA': -0.0874, 'employment_status_CB': 0.0529, 'employment_status_CC': 0.0319, 'employment_status_CD': 0.0105, 'employment_status_CE': 0.0240, 'employment_status_CF': 0.0434, 'employment_status_CG': 0.0076}

Table E.12: Bagging Algorithm Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0125	{ 'customer_age': -0.0003, 'income': 0.0398, 'employment_status_CA': 0.0235, 'employment_status_CB': -0.0176, 'employment_status_CC': -0.0031, 'employment_status_CD': -0.0255, 'employment_status_CE': -0.0081, 'employment_status_CF': 0.0097, 'employment_status_CG': 0.0031	1	0.0124	{ 'customer_age': 0.0284, 'income': -0.0100, 'employment_status_CA': -0.0420, 'employment_status_CB': -0.0033, 'employment_status_CC': 0.0351, 'employment_status_CD': 0.0285, 'employment_status_CE': 0.0162, 'employment_status_CF': 0.0188, 'employment_status_CG': 0.0032}
2	0.0112	{ 'customer_age': 0.0409, 'income': 0.0104, 'employment_status_CA': -0.0474, 'employment_status_CB': -0.0047, 'employment_status_CC': 0.0587, 'employment_status_CD': -0.0084, 'employment_status_CE': 0.0137, 'employment_status_CF': 0.0311, 'employment_status_CG': -0.0017	3	0.0097	{ 'customer_age': -0.0361, 'income': 0.0004, 'employment_status_CA': 0.0620, 'employment_status_CB': -0.0364, 'employment_status_CC': -0.0718, 'employment_status_CD': 0.0156, 'employment_status_CE': 0.0015, 'employment_status_CF': 0.0117, 'employment_status_CG': 0.0031}
4	0.0093	{ 'customer_age': -0.0026, 'income': 0.0009, 'employment_status_CA': -0.0137, 'employment_status_CB': 0.0083, 'employment_status_CC': -0.0159, 'employment_status_CD': 0.0200, 'employment_status_CE': -0.0173, 'employment_status_CF': 0.0342, 'employment_status_CG': 0.0057	5	0.0094	{ 'customer_age': -0.0074, 'income': -0.0117, 'employment_status_CA': 0.0126, 'employment_status_CB': -0.0104, 'employment_status_CC': -0.0073, 'employment_status_CD': 0.0257, 'employment_status_CE': -0.0057, 'employment_status_CF': -0.0144, 'employment_status_CG': -0.0076}
6	0.0090	{ 'customer_age': 0.0177, 'income': 0.0078, 'employment_status_CA': 0.0146, 'employment_status_CB': 0.0225, 'employment_status_CC': -0.0184, 'employment_status_CD': -0.0141, 'employment_status_CE': -0.0152, 'employment_status_CF': -0.0232, 'employment_status_CG': 0.0010	7	0.0087	{ 'customer_age': 0.0355, 'income': 0.0398, 'employment_status_CA': -0.0462, 'employment_status_CB': 0.0190, 'employment_status_CC': 0.0194, 'employment_status_CD': 0.0141, 'employment_status_CE': 0.0157, 'employment_status_CF': 0.0250, 'employment_status_CG': 0.0075}

Table E.13: EWA Algorithm Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.3029	{ 'customer_age': 0.1837, 'income': 0.1658, 'employment_status_CA': 0.1108, 'employment_status_CB': -0.0420, 'employment_status_CC': 0.0042, 'employment_status_CD': -0.0846, 'employment_status_CE': -0.0497, 'employment_status_CF': -0.0789, 'employment_status_CG': 0.0030	1	0.2943	{ 'customer_age': 0.2047, 'income': 0.1439, 'employment_status_CA': 0.0830, 'employment_status_CB': -0.0186, 'employment_status_CC': 0.0110, 'employment_status_CD': -0.0739, 'employment_status_CE': -0.0564, 'employment_status_CF': -0.0710, 'employment_status_CG': -0.0002}
2	0.2862	{ 'customer_age': 0.1417, 'income': 0.1395, 'employment_status_CA': 0.0827, 'employment_status_CB': -0.0206, 'employment_status_CC': 0.0021, 'employment_status_CD': -0.0723, 'employment_status_CE': -0.0399, 'employment_status_CF': -0.0730, 'employment_status_CG': 0.0047	3	0.3519	{ 'customer_age': 0.1365, 'income': 0.1454, 'employment_status_CA': 0.1636, 'employment_status_CB': -0.0403, 'employment_status_CC': -0.0522, 'employment_status_CD': -0.0945, 'employment_status_CE': -0.0546, 'employment_status_CF': -0.1159, 'employment_status_CG': -0.0002}
4	0.3321	{ 'customer_age': 0.1415, 'income': 0.1708, 'employment_status_CA': 0.1397, 'employment_status_CB': -0.0333, 'employment_status_CC': -0.0324, 'employment_status_CD': -0.0895, 'employment_status_CE': -0.0547, 'employment_status_CF': -0.0991, 'employment_status_CG': 0.0004	5	0.3432	{ 'customer_age': 0.1169, 'income': 0.1601, 'employment_status_CA': 0.1336, 'employment_status_CB': -0.0330, 'employment_status_CC': -0.0423, 'employment_status_CD': -0.0750, 'employment_status_CE': -0.0568, 'employment_status_CF': -0.0873, 'employment_status_CG': 0.0005}
6	0.3542	{ 'customer_age': 0.1322, 'income': 0.1550, 'employment_status_CA': 0.1116, 'employment_status_CB': -0.0305, 'employment_status_CC': -0.0344, 'employment_status_CD': -0.0668, 'employment_status_CE': -0.0425, 'employment_status_CF': -0.0800, 'employment_status_CG': -0.0008	7	0.2766	{ 'customer_age': 0.1443, 'income': 0.1406, 'employment_status_CA': 0.1167, 'employment_status_CB': -0.0279, 'employment_status_CC': -0.0320, 'employment_status_CD': -0.0756, 'employment_status_CE': -0.0495, 'employment_status_CF': -0.0808, 'employment_status_CG': -0.0013}

Table E.14: Logistic Regression Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.2134	{ 'customer_age': -0.2236, 'income': -0.1335, 'employment_status_CA': -0.1331, 'employment_status_CB': 0.1221, 'employment_status_CC': -0.0922, 'employment_status_CD': 0.0625, 'employment_status_CE': 0.0553, 'employment_status_CF': 0.0977, 'employment_status_CG': -0.0071	1	0.2055	{ 'customer_age': -0.1430, 'income': -0.1279, 'employment_status_CA': -0.0701, 'employment_status_CB': -0.0618, 'employment_status_CC': 0.0665, 'employment_status_CD': 0.0822, 'employment_status_CE': 0.0583, 'employment_status_CF': 0.0718, 'employment_status_CG': 5.73e-05}
2	0.2169	{ 'customer_age': 0.0651, 'income': -0.1383, 'employment_status_CA': -0.3061, 'employment_status_CB': 0.1284, 'employment_status_CC': 0.1407, 'employment_status_CD': 0.1034, 'employment_status_CE': 0.0459, 'employment_status_CF': 0.1684, 'employment_status_CG': -0.0013	3	0.1940	{ 'customer_age': -0.1685, 'income': -0.1474, 'employment_status_CA': 0.0178, 'employment_status_CB': -0.0920, 'employment_status_CC': -0.1333, 'employment_status_CD': 0.1389, 'employment_status_CE': 0.0134, 'employment_status_CF': 0.1822, 'employment_status_CG': -0.0068}
4	0.2051	{ 'customer_age': -0.1982, 'income': -0.1685, 'employment_status_CA': -0.1386, 'employment_status_CB': 0.1082, 'employment_status_CC': -0.1073, 'employment_status_CD': 0.1135, 'employment_status_CE': 0.0656, 'employment_status_CF': 0.1583, 'employment_status_CG': 0.0068	5	0.2015	{ 'customer_age': -0.0830, 'income': -0.1753, 'employment_status_CA': -0.0781, 'employment_status_CB': -0.0487, 'employment_status_CC': 0.0854, 'employment_status_CD': 0.0636, 'employment_status_CE': 0.0526, 'employment_status_CF': 0.0499, 'employment_status_CG': -0.0045}
6	0.1930	{ 'customer_age': -0.0341, 'income': -0.2034, 'employment_status_CA': -0.0744, 'employment_status_CB': 0.0849, 'employment_status_CC': -0.0176, 'employment_status_CD': 0.0468, 'employment_status_CE': 0.0180, 'employment_status_CF': 0.0126, 'employment_status_CG': 0.0055	7	0.2000	{ 'customer_age': -0.0475, 'income': -0.1660, 'employment_status_CA': -0.1377, 'employment_status_CB': 0.0319, 'employment_status_CC': -0.0028, 'employment_status_CD': 0.0681, 'employment_status_CE': 0.0647, 'employment_status_CF': 0.1664, 'employment_status_CG': -0.0037}

Table E.15: Perceptron Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0100	{ 'customer_age': 0.0406, 'income': 0.0572, 'employment_status_CA': 0.0355, 'employment_status_CB': -0.0166, 'employment_status_CC': 0.0048, 'employment_status_CD': -0.0292, 'employment_status_CE': -0.0104, 'employment_status_CF': -0.0250, 'employment_status_CG': 0.0010	1	0.0102	{ 'customer_age': 0.0218, 'income': 0.0502, 'employment_status_CA': 0.0292, 'employment_status_CB': -0.0147, 'employment_status_CC': 0.0014, 'employment_status_CD': -0.0210, 'employment_status_CE': -0.0033, 'employment_status_CF': -0.0232, 'employment_status_CG': -0.0021}
2	0.0091	{ 'customer_age': 0.0364, 'income': 0.0414, 'employment_status_CA': 0.0299, 'employment_status_CB': -0.0173, 'employment_status_CC': 0.0048, 'employment_status_CD': -0.0131, 'employment_status_CE': -0.0081, 'employment_status_CF': -0.0304, 'employment_status_CG': 0.0059	3	0.0078	{ 'customer_age': 0.0256, 'income': 0.0441, 'employment_status_CA': 0.0333, 'employment_status_CB': -0.0148, 'employment_status_CC': -0.0090, 'employment_status_CD': -0.0243, 'employment_status_CE': -0.0159, 'employment_status_CF': -0.0098, 'employment_status_CG': 0.0010}
4	0.0075	{ 'customer_age': 0.0173, 'income': 0.0540, 'employment_status_CA': 0.0347, 'employment_status_CB': -0.0031, 'employment_status_CC': -0.0031, 'employment_status_CD': -0.0313, 'employment_status_CE': -0.0128, 'employment_status_CF': -0.0325, 'employment_status_CG': 0.0022	5	0.0076	{ 'customer_age': 0.0197, 'income': 0.0295, 'employment_status_CA': 0.0333, 'employment_status_CB': -0.0095, 'employment_status_CC': -0.0067, 'employment_status_CD': -0.0235, 'employment_status_CE': -0.0090, 'employment_status_CF': -0.0251, 'employment_status_CG': 0.0018}
6	0.0073	{ 'customer_age': 0.0212, 'income': 0.0479, 'employment_status_CA': 0.0517, 'employment_status_CB': -0.0178, 'employment_status_CC': -0.0255, 'employment_status_CD': -0.0272, 'employment_status_CE': -0.0085, 'employment_status_CF': -0.0279, 'employment_status_CG': 0.0013	7	0.0069	{ 'customer_age': 0.0238, 'income': 0.0359, 'employment_status_CA': 0.0308, 'employment_status_CB': -0.0163, 'employment_status_CC': -0.0072, 'employment_status_CD': -0.0201, 'employment_status_CE': -0.0095, 'employment_status_CF': -0.0105, 'employment_status_CG': 0.0016}

Table E.16: OXT Regressor Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0143	{ 'customer_age': 0.0516, 'income': 0.0168, 'employment_status_CA': 0.0005, 'employment_status_CB': 0.0212, 'employment_status_CC': 0.0241, 'employment_status_CD': -0.1057, 'employment_status_CE': -0.0398, 'employment_status_CF': 0.0375, 'employment_status_CG': 0.0058	1	0.0136	{ 'customer_age': 0.0346, 'income': 0.0116, 'employment_status_CA': -0.0080, 'employment_status_CB': 0.0219, 'employment_status_CC': 0.0114, 'employment_status_CD': -0.0714, 'employment_status_CE': -0.0271, 'employment_status_CF': 0.0479, 'employment_status_CG': 0.0013}
2	0.0120	{ 'customer_age': 0.0296, 'income': 0.0059, 'employment_status_CA': -0.0138, 'employment_status_CB': 0.0207, 'employment_status_CC': 0.0124, 'employment_status_CD': -0.0656, 'employment_status_CE': -0.0252, 'employment_status_CF': 0.0474, 'employment_status_CG': 0.0086	3	0.0105	{ 'customer_age': 0.0225, 'income': 0.0082, 'employment_status_CA': 0.0028, 'employment_status_CB': 0.0101, 'employment_status_CC': 0.0009, 'employment_status_CD': -0.0741, 'employment_status_CE': -0.0378, 'employment_status_CF': 0.0517, 'employment_status_CG': 0.0058}
4	0.0112	{ 'customer_age': 0.0331, 'income': 0.0273, 'employment_status_CA': -0.0008, 'employment_status_CB': 0.0371, 'employment_status_CC': 0.0089, 'employment_status_CD': -0.0949, 'employment_status_CE': -0.0399, 'employment_status_CF': 0.0357, 'employment_status_CG': 0.0058	5	0.0107	{ 'customer_age': 0.0299, 'income': 0.0136, 'employment_status_CA': -0.0011, 'employment_status_CB': 0.0217, 'employment_status_CC': 0.0057, 'employment_status_CD': -0.0933, 'employment_status_CE': -0.0355, 'employment_status_CF': 0.0504, 'employment_status_CG': 0.0042}
6	0.0102	{ 'customer_age': 0.0328, 'income': 0.0168, 'employment_status_CA': -0.0008, 'employment_status_CB': 0.0271, 'employment_status_CC': 0.0033, 'employment_status_CD': -0.1038, 'employment_status_CE': -0.0372, 'employment_status_CF': 0.0502, 'employment_status_CG': 0.0044	7	0.0109	{ 'customer_age': 0.0368, 'income': 0.0203, 'employment_status_CA': -0.0137, 'employment_status_CB': 0.0513, 'employment_status_CC': 0.0162, 'employment_status_CD': -0.1096, 'employment_status_CE': -0.0412, 'employment_status_CF': 0.0454, 'employment_status_CG': 0.0073}

Table E.17: MLP Regressor Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0108	{ 'customer_age': 0.1057, 'income': 0.0802, 'employment_status_CA': 0.0595, 'employment_status_CB': -0.0382, 'employment_status_CC': 0.0187, 'employment_status_CD': -0.0370, 'employment_status_CE': -0.0231, 'employment_status_CF': -0.0411, 'employment_status_CG': 0.0008	1	0.0132	{ 'customer_age': 0.1377, 'income': 0.1061, 'employment_status_CA': 0.0528, 'employment_status_CB': -0.0204, 'employment_status_CC': 0.0158, 'employment_status_CD': -0.0453, 'employment_status_CE': -0.0330, 'employment_status_CF': -0.0453, 'employment_status_CG': -0.0025}
2	0.0098	{ 'customer_age': 0.0758, 'income': 0.0758, 'employment_status_CA': 0.0422, 'employment_status_CB': -0.0216, 'employment_status_CC': 0.0121, 'employment_status_CD': -0.0356, 'employment_status_CE': -0.0175, 'employment_status_CF': -0.0374, 'employment_status_CG': 0.0057	3	0.0093	{ 'customer_age': 0.0672, 'income': 0.1055, 'employment_status_CA': 0.0991, 'employment_status_CB': -0.0282, 'employment_status_CC': -0.0225, 'employment_status_CD': -0.0636, 'employment_status_CE': -0.0369, 'employment_status_CF': -0.0708, 'employment_status_CG': 0.0004}
4	0.0082	{ 'customer_age': 0.0776, 'income': 0.0910, 'employment_status_CA': 0.0734, 'employment_status_CB': -0.0221, 'employment_status_CC': -0.0101, 'employment_status_CD': -0.0486, 'employment_status_CE': -0.0312, 'employment_status_CF': -0.0524, 'employment_status_CG': -0.0003	5	0.0089	{ 'customer_age': 0.0647, 'income': 0.1238, 'employment_status_CA': 0.0852, 'employment_status_CB': -0.0221, 'employment_status_CC': -0.0178, 'employment_status_CD': -0.0576, 'employment_status_CE': -0.0350, 'employment_status_CF': -0.0597, 'employment_status_CG': 0.0001}
6	0.0077	{ 'customer_age': 0.0520, 'income': 0.0924, 'employment_status_CA': 0.0696, 'employment_status_CB': -0.0246, 'employment_status_CC': -0.0230, 'employment_status_CD': -0.0417, 'employment_status_CE': -0.0208, 'employment_status_CF': -0.0427, 'employment_status_CG': -0.0019	7	0.0073	{ 'customer_age': 0.0598, 'income': 0.0845, 'employment_status_CA': 0.0667, 'employment_status_CB': -0.0285, 'employment_status_CC': -0.0138, 'employment_status_CD': -0.0375, 'employment_status_CE': -0.0253, 'employment_status_CF': -0.0378, 'employment_status_CG': -0.0002}

Table E.18: SGT Regressor Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0116	{ 'customer_age': 0,0225 , 'income': 0,0226 , 'employment_status_CA': 0,0284 , 'employment_status_CB': -0,0123 , 'employment_status_CC': -0,0180 , 'employment_status_CD': -0,0065, 'employment_status_CE': -0,0062, 'employment_status_CF': -0,0099, 'employment_status_CG': 0,0008	1	0.0116	{ 'customer_age': 0,0091, 'income': 0,0026, 'employment_status_CA': 0,0177, 'employment_status_CB': -0,0101, 'employment_status_CC': 0,0037, 'employment_status_CD': -0,0129, 'employment_status_CE': 0,0002, 'employment_status_CF': -0,0176, 'employment_status_CG': 0,0011}
2	0.0107	{ 'customer_age': 0,0140, 'income': 0,0021 , 'employment_status_CA': 0,0025, 'employment_status_CB': 0,0015, 'employment_status_CC': 0,0007, 'employment_status_CD': -0,0065, 'employment_status_CE': 0,0010, 'employment_status_CF': -0,0051, 'employment_status_CG': 0,0073	3	0.0089	{ 'customer_age': -0,0363, 'income': -0,0014, 'employment_status_CA': 0,0460, 'employment_status_CB': -0,0319, 'employment_status_CC': -0,0340, 'employment_status_CD': 0,0039, 'employment_status_CE': -0,0073, 'employment_status_CF': -0,0008, 'employment_status_CG': 0,0012}
4	0.0087	{ 'customer_age': 0,0143, 'income': 0,0753, 'employment_status_CA': -0,0054, 'employment_status_CB': 0,0148, 'employment_status_CC': 0,0028, 'employment_status_CD': -0,0016, 'employment_status_CE': -0,0103, 'employment_status_CF': -0,0067, 'employment_status_CG': 0,0078	5	0.0090	{ 'customer_age': 0,0084, 'income': 0,0069, 'employment_status_CA': 0,0180, 'employment_status_CB': -0,0074, 'employment_status_CC': 0,0044, 'employment_status_CD': -0,0181, 'employment_status_CE': -0,0090, 'employment_status_CF': -0,0136, 'employment_status_CG': 0,0026}
6	0.0083	{ 'customer_age': 0,0247, 'income': 0,0325, 'employment_status_CA': 0,0314, 'employment_status_CB': -0,0158, 'employment_status_CC': -0,0122, 'employment_status_CD': -0,0118, 'employment_status_CE': -0,0156, 'employment_status_CF': -0,0102, 'employment_status_CG': 0,0027	7	0.0079	{ 'customer_age': 0,0364, 'income': 0,0110, 'employment_status_CA': 0,0134, 'employment_status_CB': -0,0084, 'employment_status_CC': 0,0081, 'employment_status_CD': -0,0075, 'employment_status_CE': -0,0096, 'employment_status_CF': -0,0151, 'employment_status_CG': 0,0046}

Table E.19: Hoeffding Tree Regressor Model Evaluation.

Month	MSE	Fairness Metrics	Month	MSE	Fairness Metrics
0	0.0121	{ 'customer_age': 0,1135 , 'income': 0,0848 , 'employment_status_CA': 0,0164 , 'employment_status_CB': 0,0020, 'employment_status_CC': 0,0062, 'employment_status_CD': -0,0247, 'employment_status_CE': -0,0178, 'employment_status_CF': -0,0214, 'employment_status_CG': 0,0057	1	0.0115	{ 'customer_age': 0,0413, 'income': 0,0163 , 'employment_status_CA': 0,0108, 'employment_status_CB': -0,0002, 'employment_status_CC': 0,0117, 'employment_status_CD': -0,0181, 'employment_status_CE': -0,0062, 'employment_status_CF': -0,0212, 'employment_status_CG': -0,0025}
2	0.0111	{ 'customer_age': 0,0834, 'income': -0,0097, 'employment_status_CA':-0,0216, 'employment_status_CB': 0,0286, 'employment_status_CC': 0,0376, 'employment_status_CD': -0.0356, 'employment_status_CE': -0.0175, 'employment_status_CF': -0.0374, 'employment_status_CG': 0.0057	3	0.0095	{ 'customer_age': 0,0236, 'income': -0,0100, 'employment_status_CA': 0,0018, 'employment_status_CB': 0,0076, 'employment_status_CC': -0,0134, 'employment_status_CD': 0,0032, 'employment_status_CE': -0,0085, 'employment_status_CF': 0,0053, 'employment_status_CG': 0,0004}
4	0.0093	{ 'customer_age':0,0171 , 'income': -0,0097 , 'employment_status_CA': 0,0240, 'employment_status_CB': 0,0056, 'employment_status_CC': -0,0123, 'employment_status_CD': -0,0100, 'employment_status_CE': 0,0012, 'employment_status_CF': -0,0356, 'employment_status_CG': 0,0066	5	0.0093	{ 'customer_age': 0,0186, 'income': -0,0160, 'employment_status_CA': 0,0166, 'employment_status_CB': 0,0000, 'employment_status_CC': 0,0120, 'employment_status_CD': -0,0260, 'employment_status_CE': -0,0112 'employment_status_CF': -0,0237, 'employment_status_CG': 0,0025}
6	0.0091	{ 'customer_age': 0,0853, 'income': 0,1235 , 'employment_status_CA': 0,0803 'employment_status_CB': -0,0160, 'employment_status_CC': -0,0240 , 'employment_status_CD': -0,0571, 'employment_status_CE': -0,0281, 'employment_status_CF': -0,0635, 'employment_status_CG': -0,0010	7	0.0087	{ 'customer_age': 0,0523, 'income': -0,0171, 'employment_status_CA': 0,0431, 'employment_status_CB': -0,0337, 'employment_status_CC': -0,0071, 'employment_status_CD': -0,0077, 'employment_status_CE': -0,0070, 'employment_status_CF': -0,0223, 'employment_status_CG': 0,0038}

Table E.20: Hoeffding Adaptive Tree Regressor Model Evaluation.

E.2 Coefficient Correlation of Residuals Averages

Attribute	Average Value
Customer Age	0,0259
Income	0,015
Employment Status CA	0,0476
Employment Status CB	0,0280
Employment Status CC	0,0339
Employment Status CD	0,0206
Employment Status CE	0,0131
Employment Status CF	0,0229
Employment Status CG	0,0037

Table E.21: Avg. - Linear Reg.

Attribute	Average Value
Customer Age	0,0748
Income	0,0901
Employment Status CA	0,1846
Employment Status CB	0,1366
Employment Status CC	0,1307
Employment Status CD	0,0611
Employment Status CE	0,0771
Employment Status CF	0,1308
Employment Status CG	0,0103

Table E.22: Avg. - PA

Attribute	Average Value
Customer Age	0,0309
Income	0,0188
Employment Status CA	0,0460
Employment Status CB	0,0302
Employment Status CC	0,0311
Employment Status CD	0,0110
Employment Status CE	0,0149
Employment Status CF	0,0216
Employment Status CG	0,0031

Table E.23: Avg. - Bagging

Attribute	Average Value
Customer Age	0,0211
Income	0,0151
Employment Status CA	0,0328
Employment Status CB	0,0153
Employment Status CC	0,0287
Employment Status CD	0,0190
Employment Status CE	0,0117
Employment Status CF	0,0210
Employment Status CG	0,0041

Table E.24: Avg. - EWA

Attribute	Average Value
Customer Age	0,1502
Income	0,1526
Employment Status CA	0,1177
Employment Status CB	0,0308
Employment Status CC	0,0263
Employment Status CD	0,0790
Employment Status CE	0,0505
Employment Status CF	0,0858
Employment Status CG	0,0014

Table E.25: Avg. - Logistic

Attribute	Average Value
Customer Age	0,1204
Income	0,1575
Employment Status CA	0,1195
Employment Status CB	0,0847
Employment Status CC	0,0807
Employment Status CD	0,0849
Employment Status CE	0,0467
Employment Status CF	0,1134
Employment Status CG	0,0045

Table 26: Avg. - Perceptron

Attribute	Average Value
Customer Age	0,0258
Income	0,0450
Employment Status CA	0,0348
Employment Status CB	0,0138
Employment Status CC	0,0078
Employment Status CD	0,0237
Employment Status CE	0,0097
Employment Status CF	0,0231
Employment Status CG	0,0021

Table E.27: Avg. - OXT Regressor

Attribute	Average Value
Customer Age	0,0339
Income	0,0151
Employment Status CA	0,0052
Employment Status CB	0,0264
Employment Status CC	0,0104
Employment Status CD	0,0898
Employment Status CE	0,0355
Employment Status CF	0,0458
Employment Status CG	0,0054

Table E.28: Avg. - MLP Regressor

Attribute	Average Value
Customer Age	0,0801
Income	0,0949
Employment Status CA	0,0686
Employment Status CB	0,0257
Employment Status CC	0,0167
Employment Status CD	0,0458
Employment Status CE	0,0278
Employment Status CF	0,0484
Employment Status CG	0,0015

Table E.29: Avg. - SGT Regressor

Attribute	Average Value
Customer Age	0,0207
Income	0,0193
Employment Status CA	0,0204
Employment Status CB	0,0128
Employment Status CC	0,0105
Employment Status CD	0,0086
Employment Status CE	0,0074
Employment Status CF	0,0099
Employment Status CG	0,0035

Table E.30: Avg. - Hoeffding Tree Regressor

Attribute	Average Value
Customer Age	0,0544
Income	0,0540
Employment Status CA	0,0268
Employment Status CB	0,0117
Employment Status CC	0,0155
Employment Status CD	0,0200
Employment Status CE	0,0127
Employment Status CF	0,0273
Employment Status CG	0,0060

Table E.31: Avg. - Hoeffding Adaptve Tree Regressor