UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DISTRIBUTED MACHINE LEARNING ALGORITHMS

Diploma Thesis

Michail Oikonomopoulos

Supervisor: Michael Vassilakopoulos

Volos 2019

Περίληψη

Καθώς τα Δεδομένα Μεγάλου Όγκου (Big Data) χρησιμοποιούνται ευρέως από τη βιομηχανία των χρηματοοικονομικών υπηρεσιών, πολλές τράπεζες και ιδρύματα δανεισμού αποφάσισαν να ανανεώσουν τις πολιτικές λήψης αποφάσεών τους. Η παρουσία των Δεδομένων Μεγάλου Όγκου καθόρισε επίσης επιτακτική την ανάγκη για μεγαλύτερη υπολογιστική ισχύ. Συνεπώς, οι κατανεμημένοι αλγόριθμοι μηχανικής μάθησης αποτελούν ένα βασικό εργαλείο για την αύξηση της ταχύτητας ανάπτυξης μοντέλων πρόβλεψης που είναι ακριβή, με το να παραλληλοποιούν τους υπολογισμούς των αλγορίθμων. Σκοπός αυτής της μελέτης είναι να εφαρμόσει τεχνικές μηχανικής μάθησης έτσι ώστε να κατασκευάσει κάποια μοντέλα πρόβλεψης πιστωτικού κινδύνου με τη χρήση της πλατφόρμας H2O. Αξιολογούμε τον πιθανό πιστωτικό κίνδυνο για εταιρίες με την αξιοποίηση αλγορίθμων του εργαλείου H2O που έχουν ως βάση την παραλληλοποίηση. Συγκεκριμένα, θα γίνει η προσπάθεια να αξιολογηθεί εάν ένας πελάτης μπορεί να θεωρηθεί αξιόπιστος για την εταιρεία δανεισμού.

Abstract

Due to the fact that Big Data have been widely adopted in the financial services industry, many banks and lending institutions have decided to renew their decision-making policies. The presence of Big Data has also established the vast need for more computing power. Thus, distributed machine learning constitutes a major tool to increase the speed of developing highly accurate predictive models, by parallelizing the computations of the algorithms. The purpose of this study is to apply machine learning techniques in order to construct several forecasting credit risk models with the use of H2O platform. We evaluate potential credit risk for businesses by exploiting the parallelization of the H2O algorithms. Specifically, we will try to assess if a particular client could be considered as trustworthy for the lending company.

Acknowledgements

First and foremost, I would like to express my sincere gratitude and appreciation to my thesis advisor and supervisor Professor Emeritus Elias N. Houstis of the Department of Electrical & Computer Engineering at University of Thessaly. His guidance and motivation had a prominent role to the fulfillment of my research work. He always showed me his trust during my university years, by also being very kind, supportive and always available when I needed him.

Besides my advisor, I would also like to thank the rest of my thesis committee: Associate Professor Michael Vassilakopoulos, Professor Emmanouil Vavalis and Associate Professor Dimitrios Bargiotas, as without their participation and approval of my work, I would have not fulfilled my thesis.

Finally, I am always grateful to my family and friends, for their continuous support and encouragement throughout all those years of study and my life in general. Especially, I would like to thank Eirini Tsitsi for all of her help during this year. Nothing would have been accomplished without all these people.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

P2P	Peer-to-Peer
AI	Artificial Intelligence
CPU	Central Processing Unit
GPU	Graphics Processing Unit
POJO	Plain Old Java Object
GUI	Graphical User Interface
RAM	Random Access Memory
LibSVM	Library for Support Vector Machines
pub_rec_bankruptcies	Number of public record bankruptcies
total_acc	Total number of credit lines currently in the borrower's credit file
installment	Monthly payment owed by the borrower if the loan originates
LC	Lending Club
grade	LC assigned loan grade
sub_grade	LC assigned loan subgrade
MSE	Mean Squared Error
AutoML	Automatic Machine Learning
DRF	Distributed Random Forest
GBM	Gradient Boosting Machine
DL	Deep Learning
MR	MapReduce
ANN	Artificial Neural Network
DNN	Deep Neural Network
XRT	Extremely Randomized Forest
GLM	Generalized Linear Model
AUC	Area Under The Curve
ROC	Receiver Operating Characteristics
SE	Stacked Ensemble

Chapter 1
Introduction

1.1 Motivation

As machine learning has become an important tool in a lot of modern computer applications and has made its impact in several fields of the modern world, the need for accelerating the training process of the models has become vast.

In financial services the competition is intense with firms looking for every advantage in marketing while trying to fight fraud, money laundering, as well as charged-off loans. Those companies who exploit the benefits of machine learning and Artificial Intelligence tend to increase both their revenues and also their customer satisfaction. Further, huge datasets are being used. So, machine learning tools which can automate and facilitate the modeling procedure are imperative.

Based on this background, the aim of this work is to select and implement a number of distributed based machine learning algorithms provided by H2O.ai and produce and evaluate the corresponding results.

1.2 Purpose and Research Question

In this research work, several machine learning techniques are used in order to predict a borrower's loan status behavior. The purpose is to give an insight on how distributed machine learning algorithms could be applied and the results they come up with. Therefore, the main research question that occurs is:

- How can a credit risk management problem be approached with a distributed in-memory machine learning tool such as H2O?

1.3 Outline

Chapter 2 addresses the theoretical background related to this work, focusing on the lending process in some financial services, basic concepts of credit risk management, distributed

machine learning and presentation of the H2O platform. After referring to related work in chapter 3, the whole methodology of the implementation is demonstrated, from data preprocessing to all the algorithms that were used. Chapter 5 discusses the evaluation of the models and results. Finally, chapter 6 summarizes the work that was made and gives a standpoint for possible future work and extension on this topic.

Chapter 2

Theoretical Background

2.1 Lending Process in Financial Services

This section describes two main categories of lending process. The one is the traditional bank lending process, while the second one, named peer-to-peer lending has flourished and gained attraction within the last decade, in the world of finance. Further the advantages of the latter will be examined.

2.1.1 Bank Lending Process

One of the major activities of banking companies is to grant loans. The term "bank lending policy" refers to the policy and guidelines adopted by a bank in order to make its lending process systematic and methodical. More specifically, lending policy is a set of criteria and standards developed and used by a lending institution or a bank to reach decision on a loan application. Banks lend the money which they themselves borrow from the depositors. This happens because they cannot keep the deposits idle or lend the deposits and not recollect. The bank lending process consists of 6 basic steps:

1. Firstly, banks need to find prospective loan customers, which can be either individuals or businesses.
2. The prospective customer's character and the purpose of a loan application have to be evaluated. This is achieved through an interview with the potential client. The customer has the opportunity to explain their credits needs which is particularly essential. This interview will enable the bank to decide about the approval on the customer's loan request.
3. The prospective customer's credit record has to be evaluated. A previous payment record should be taken into consideration because it reveals much about the customer's character and their sense of responsibility. The loan officer may contact other creditors who have previously loaned money to this customer so as to examine what their previous experience has been.

4. The potential customer's financial condition has to be examined. If everything seems to be favorable at this point the customer is asked to submit several essential documents the lender needs in order to fully evaluate the loan request. Some of these documents may be customer's complete financial statements. These documents are then examined to determine whether the customer has sufficient cash flow and backup assets to repay the bank's loan.

5. The customer's possible loan collateral has to be assessed. When the client's loan request is approved the loan officer usually checks on the property or other assets to be pledged as collateral to ensure that the bank has immediate access to the collateral or can acquire title to the property involved if the loan agreement has defaulted. When both the loan and the proposed collateral are approved, all the necessary documents about the loan agreement are signed by all parties.

6. The customer's compliance with the loan agreement has to be monitored continuously in order to ensure that the terms of the loan are being followed and that all required payments of principal and interest being made as promised.

2.1.2 Peer-to-peer (P2P) Lending

Peer-to-peer lending (P2P) is the action of lending money to individuals or businesses through online services and platforms that match lenders with borrowers. More specifically, P2P is direct and online lending outside traditional financial intermediaries like banks or other financial institutions. Peer-to-peer (P2P) lending platforms are websites where borrowers can solicit funds from investors. There are several definitions of P2P lending platforms. P2P can be considered as an example of financial disintermediation, as another technological disruption provoked by Internet, as a case of collaborative economy or even as a platform to give loans to financially excluded people.

The first commercial online P2P lending platform started in 2005 and was given the name Zopa. It was founded in February 2005 and was the first peer-to-peer lending company in the United Kingdom and the world.

P2P lending has attracted considerable attention in recent years and is very popular for both borrowers and lenders. It is worth noting that P2P is person-to-person, so investments go to real people and boost the societal value of the sharing economy in finance.

2.1.3 Advantages of Peer-to-peer (P2P) Lending

P2P lending platforms offer a multitude of advantages to borrowers who decide to make a loan application. Some of them are the following:

1. P2P lending platforms are less restrictive than traditional banks. P2P platforms serve borrowers regardless of the purpose of the loan.
2. The procedure of taking a loan is much faster with P2P platforms because these platforms have an automated online experience where customers can enter all their information and personal data and upload all the essential documents in order to receive an instant decision. This process is particularly time consuming when someone wants to borrow from a traditional bank as no interviews have to be done.
3. Many P2P platforms offer their customers interests rates that are even lower than those offered by the banks. Mainly, this applies to the platforms offering longer durations on their loans.
4. P2P lending platforms offer more favorable conditions for borrowers and lenders because these platforms are able to provide loans with lower intermediation costs than traditional banks because of their online functioning. Moreover, such lending platforms like P2P do not have to pay for costly branches.
5. P2P platforms give borrowers a quick decision for a loan application and investors have the opportunity to invest in loans with different credit ratings based on their own risk appetite.

2.1.4 The Role of Artificial Intelligence and Machine Learning in the Lending Process

It is apparent that the lending process has dramatically changed over time and due to the advanced technology associated with Big Data most banks or lending institutions are renewing their business models [5]. Back in the days where there was no data recording, bankers used to rely on the reputation of clients in order to agree to a loan. In recent years, as data have penetrated in the financial world, new ranking systems were developed which helped the banking industry to innovate and evolve. Thus, the entrance of machine learning and Artificial

Intelligence did not take a long time until making its presence and having a big impact in the field of finance.

To begin with, the process of creating a customer portfolio can always be risky for businesses. Therefore, AI and machine learning can optimize that process by assessing the goals of each customer and the risk tolerance so as to develop an individualized portfolio, based on factors including customer age, current income, employment opportunity, recent credit history and current assets.

Moreover, potential customers who struggled with establishing a credit can also be benefited with the use of machine learning. Due to the fact that machine learning models can evaluate potential risk with highly precision, credit scores for underserved costumers could accurately be predicted. Lenders are also benefited in comparison to other institutions which use traditional credit score methods, as those customers who were unidentified can now be evaluated and assigned a credit score.

In overall, AI and machine learning are widely being used in many financial institutions in order to enhance their decision making. Due to the narrow character of modern AI models, researchers and enterprises have adopted the practice of combining multiple models to cover more ground [2]. Large-scale datasets which contain client information are fed into machine learning algorithms in order to assess credit quality and thus result to a profitable loan agreement.

2.2 Credit Risk Management

It is a common knowledge that every business has to take several types of risks [18]. In the world of finance, many companies typically provide credit sales to their customers. They provide them with the opportunity to pay in order to reserve goods and services. Although, company takes credit risk by lending money to customers. Credit risk is defined as the probability of non-payment by the customer [6]. The awareness of a business's partner creditworthiness is a prerequisite for the accurate decision of providing credit sales. Hence, credit risk management stands for the management of credit sales to minimize the credit risk for the company [17].

2.2.1 Credit Scoring

One of the most important processes in banks credit management decisions is credit evaluation.

Credit evaluation includes collecting, analyzing and classifying different credit variables in order to assess the credit decisions. One of the most significant tools, to classify a bank's customers, is credit scoring. Credit scoring is the process of modelling creditworthiness, and a technique that helps organizations decide whether or not to grant credit to consumers who apply to them [1][29].

There are also many other definitions of credit scoring. Anderson (2007) proposed that the term credit scoring should be divided into two parts, credit and scoring. The first word "credit" comes from the Latin word "credo" which means "I trust in" or "I believe" in Latin. Currently, the word "credit" means "buy now and pay later". The second word "scoring" declares "the use of all numerical tools that help us to rank order cases to differentiate between their qualities". Therefore, combining the meaning of these two words, Anderson (2007) mentioned that credit scoring is "the use of statistical models to transform relevant data into numerical measures that guide credit decisions" [30]. Moreover, Gup and Kolari (2005) stated that "credit scoring is the use of statistical models to determine the likelihood that a prospective borrower will default on a loan" [31]. In addition, credit scoring is the set of decision models and their underlying techniques that help lenders in the granting of consumer credit. These techniques will decide who will get credit, how much credit they should get, and what operational strategies will enhance the profitability of the borrowers to lenders [32].

2.2.2 FICO Score

FICO has long been involved with using AI as part of the analytic approach [16]. FICO Score is a type of credit score which corresponds to a borrower and lenders make use of it along with other metrics and details in order to asses credit risk and determine whether to extend credit of a specific borrower [11]. Length of credit history is considered to be a general rule of thumb. In other words, the score tends to be higher when an individual has a long credit. Although, having favorable scores in the other categories, even someone with a short credit history can have a good score.

2.3 Parallel Machine Learning

In the modern age, most computing systems from personal laptops to cloud computing systems are available for parallel and distributing computing. Distributed computing performs an increasingly critical role in modern data processing. Particularly, applying AI in

distributed environments is becoming an element of high added value and economic potential [10]. So, efforts to create effective parallel machine learnings algorithms have become more intensive and more successful over the years, as the need for processing big amounts of data has become essential. An overview of parallel hardware architectures that are used in the machine learning process would be given.
They can practically be classified into single-machine (often shared memory) and multi-machine (often distributed memory) systems [9].

2.3.1 Single-Machine Parallelism

Parallelism can be found everywhere in today's computer architecture, internally on the chip in the form of pipelining and out-of-order execution, as well as exposed to the programmer in the form of multi-core or multi-socket systems. Multi-core systems have a long tradition and can be programmed with either multiple processes (different memory domains), multiple threads (shared memory domains), or a combination of both. The main difference is that multiprocess parallel programming forces the programmer to consider the distribution of the data as a first-class concern while multi-threaded programming allows the programmer to only reason about the parallelism, leaving the data shuffling to the hardware system (often through hardware cache-coherence protocols).

2.3.2 Multi-Machine Parallelism

It is a fact, that the training of large-scale models consists a very compute-intensive procedure. Hence, single machines cannot cope with the intensity of those tasks and are often not capable to finish the process in a desired time-frame. So, in order to accelerate the computation further, it can be distributed across multiple machines which are connected by a network. Parameters of a significant importance in this case are latency, bandwidth and message rate. Different network technologies provide different performance for the training of models.

2.4 H2O.ai

In this section, we present the H2O project, as well as its open-source user interface H2O Flow, which are both produced by the company H2O.ai.

2.4.1 What is H2O

Figure 2.1 : H2O.ai logo

H2O is an open source, in-memory, distributed, fast, and scalable machine learning and predictive analytics platform that allows you to build machine learning models on big data and provides easy way to put into production of those models in an enterprise environment. As H2O is using in-memory compression techniques, it can handle large scale datasets in-memory, even with a fairly small cluster [12][14]. This is the reason why H2O is considered to be a "fast" platform as data are being distributed across the cluster and stored in-memory in a compressed columnar format, allowing the parallelization of data.

The core code of H2O is written in JAVA and a distributed Key/Value store is used to access and reference data, models, objects, etc., across all nodes and machines. The algorithms are implemented on top of H2O's distributed Map/Reduce framework and utilize the Java Fork/Join framework for multi-threading, with the goal to allow simple horizontal scaling to a given problem in order to produce a solution to the highest speed. So, H2O constitutes a software that can be used for data modelling and general computing, with the primary purpose of a distributed (many machines), parallel (many CPUs) and in memory processing engine. The parallelism of H2O is divided into two categories. The first one is to launch a single node (local machine), which is the one that was used in our implementation. The second one is to launch multiple nodes in a cluster. The data parser of H2O has the ability to guess the schema of the dataset that is imported and read from multiple sources in various formats.

H2O offers good quality along with speed, ease-of-use and model-deployment for the multiple Supervised and Unsupervised algorithms. Thus, machine learning models that are built with R or Python can be easily converted to POJO format and be deployed on any Java environment. Furthermore, hardware needs are being decreased in comparison with other similar frameworks such as Spark or TensorFlow. At last, the experiments can be simply done with H2O by just starting the framework and make experiments with Python or R on a browser notebook [24].

2.4.2 H2O Flow

H2O Flow is an open-source user interface for H2O. It is a web-based interactive environment that provides the user with the opportunity to combine code execution, text, mathematics, plots, and rich media in a single document.

Flow not only provides the ability to use H2O interactively, but also has mechanisms for capturing, replaying, annotating, sharing and presenting the analysis workflow. H2O Flow allows the user to import files, build models, iteratively improve them, make predictions and finally add rich text to build up vignettes of their work for sharing and presentation.

The hybrid user interface of H2O Flow seamlessly blends command-line computing with a modern graphical user interface. However, rather than displaying output as plain text, Flow provides a point-and-click user interface for every H2O operation. It allows to access any H2O object in the form of well-organized tabular data.

In this research work, we used H2O Flow only for the evaluation of our models as the interface makes the results and plots more comprehensible. Below, we can see the main menu of H2O Flow.

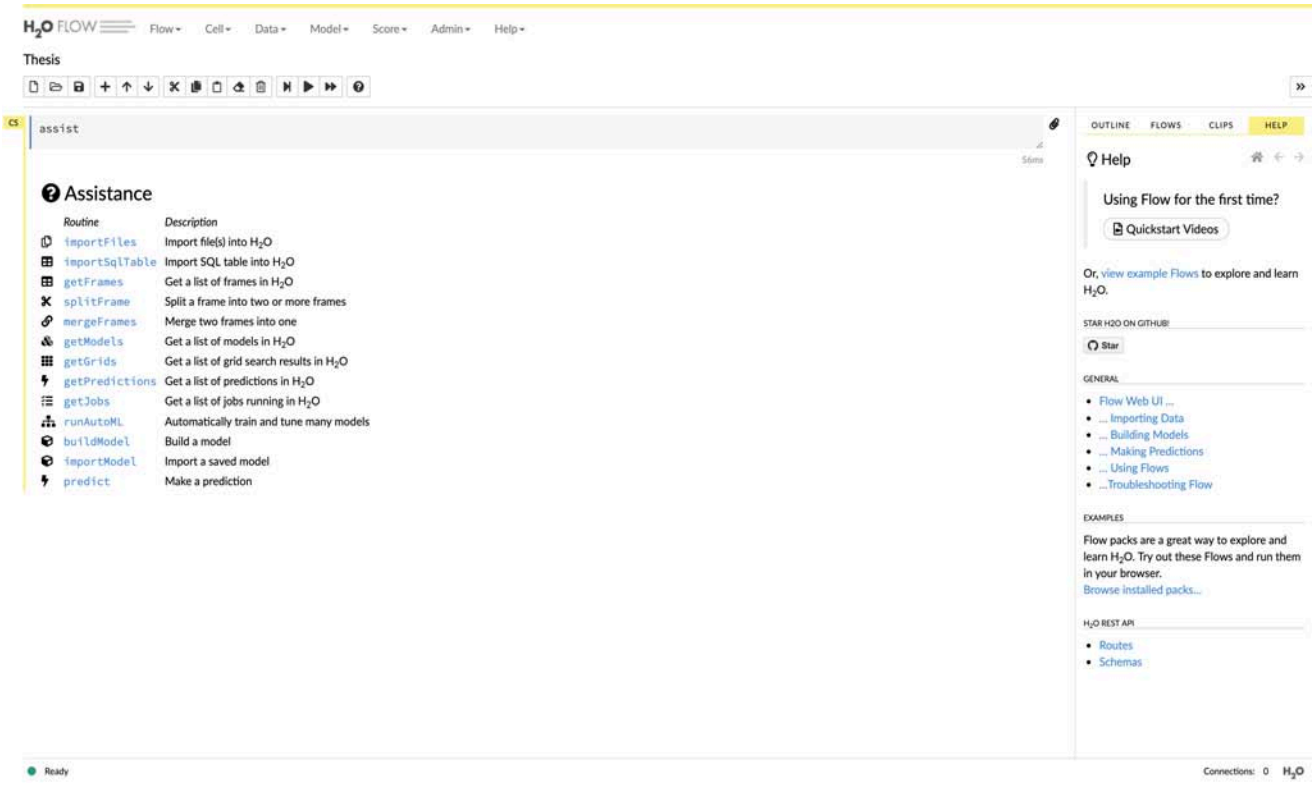

Figure 2.2 : H2O Flow main menu

2.5 Software Tools

In this part, we describe some more software tools that were used for the implementation part of this research work beyond H2O.

2.5.1 Anaconda & Anaconda Navigator

Anaconda is a free and open source distribution of the Python and R programming languages which is mainly used for scientific computing, large-scale data processing and predictive analytics, that intents to simplify package management and deployment.

Anaconda Navigator is a desktop graphical interface (GUI) included in Anaconda distribution that allows to launch applications such as Jupyter Notebook and also manage conda packages, environments and channels without using command-line commands.

2.5.2 Python

The Python programming language is an open source, cross-platform, high level, dynamic, interpreted language and of the most popular general-purpose programming languages nowadays. It constitutes one of the fastest growing programming languages and is used in several scientific field. In recent years, Python has extended its reach to the data science community due to the vast number of tools and libraries which made it easy and quick to use. The main Python libraries used were Pandas, NumPy and Matplotlib. All the three of them were pre-installed in the Anaconda installation package. Pandas is an open source Python package, used for fast and flexible data manipulation and analysis, as it offers data structures and operations for manipulating numerical tables and time series data. NumPy is also a Python based library, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. At last, Matplotlib is a plotting library that was used for the visualizations in the section of data analysis.

2.5.3 Jupyter Notebook & Google Colaboratory

Jupyter Notebook is a web-based interactive computational environment for creating Jupyter notebook documents. It allows to capture the whole computational process, by developing, documenting, and executing code, as well as communicating the results.

Google Colaboratory is a free Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive. It supports free virtual machines to use with about 12GB RAM and 50GB hard drive space and also free access to GPU resources in order to speed up the machine learning processing. It was used in order to boost up procedures such as the H2O's AutoML which require significant time, with the use of GPU.

The whole part of the implementation was completed using Python and more specifically in Jupyter Notebook, into the Anaconda and Google Colaboratory environment.

Chapter 3

Related Work

The idea of automating the lending process using machine learning methods does not consist a new field of research. Related machine learning processes have been implemented in the field of credit risk management. We present several of those works in a chronological order.

In the work of Tsai et al. (2014), the Lending Club dataset has been used to optimize peer lending risk. The method they adopted to approach the problem, was trying to minimize the probability of classifying a bad loan as good. Four machine learning algorithms were employed and more specifically Naïve Bayes, Random Forest, Logistic Regression and LibSVM. Logistic Regression provided the best results among them [22]. Another relevant study was conducted by Lopes et al. (2017), which aimed at identifying default clients with credit recovery potential, with the use of distributed machine learning algorithms implemented on H2O platform. The dataset was obtained by the extraction of information from legacy systems and customers relationship data marts and the implemented algorithms used were Generalized Linear Model, Distributed Random Forest, Deep Learning and Gradient Boosting Machine. The study was carried out in three segments of a bank's operations and achieved great results. from the predictive models seem to be of a great value [27]. At last Addo et al. (2018), were also involved in building binary classifiers based on machine learning and deep learning models on real data, in order to predict the default loan probability. In particular, the top ten important features from these models were selected and then used in the modeling process to test the performance of binary classifiers. The conclusion drawn from that research work, was that the tree-based models behave a more stable manner than the models based on multilayer artificial networks [5].

Chapter 4

Methodology

In this chapter, we describe the process that was followed from the collection of data, to analysis and preparation, and finally the machine learning algorithms that were used.

Figure 4.1: Predictive Analytics

4.1 Data Preparation

4.1.1. Data Collection

This work required data in order to construct the prediction models and draw conclusions. The data for this research work were freely collected from the Lending Club Statistics web page and have been downloaded without registration. Specifically, the datasets that were used were from years 2007-2011 and 2012-2013. The two datasets were merged into a single one. Our merged dataset, which combines the years from 2007 to 2013 has 230,721 observations and 145 initial variables. The description for all the 145 variables can be found in the Lending Club website.

4.1.2. Data Description

In this subsection, we describe the data that were used in this work. More profoundly, we indicate the loan data that were included in this research work, as also some more relevant information according to them.

Lending Club company matches investors with potential borrowers through its online platform. The personal loan amount can range from $1,000 to $40,000 and the loan offer that a borrower receives is based on the information provided on the application and credit report.

Loan grade is the result of a formula created from Lending Club that takes into account a combination of several credit risk indicators from the credit report and loan application of the borrower. All loans have either a 36- or 60-month term, with fixed interest rate and equal payments.

According to Lending Club, each Note in a borrower's account corresponds to a specific loan status. Below, we can see the description of every loan status used in the dataset:

- **Current:** Loan is up to date on all outstanding payments.

- **Fully paid:** Loan has been fully repaid, either at the expiration of the 3- or 5-year year term or as a result of a prepayment.

- **Charged Off:** Loan for which there is no longer a reasonable expectation of further payments. Upon Charge Off, the remaining principal balance of the Note is deducted from the account balance.

- **Default:** Loan has not been current for an extended period of time.

- **Issued:** New loan that has passed all Lending Club reviews, received full funding, and has been issued.

- **In Grace Period:** Loan is past due but within the 15-day grace period.

- **Late (16-30):** Loan has not been current for 16 to 30 days.

- **Late (31-120):** Loan has not been current for 31 to 120 days.

Out of those eight loan status categories we only kept the observations which belong to either *Fully Paid* or *Charged Off* loan status as we want to construct a binary classification problem of whether a potential borrower will default their loan or not.

4.1.3 Data Preprocessing & Cleaning

Data preprocessing consists an integral part in any machine learning project. As data in the real world tend to be incomplete, that step affects directly the ability of a machine learning model to learn, with the quality of data and the useful information that can be derived from it. It is of a key importance to preprocess our data before feeding it into our model. The basic concepts that were covered in this work were the following:

1. Handling Missing Values
2. Handling Categorical Values
3. Feature Scaling

The initial dataset which consists of 145 initial variables, had a lot of missing values. The first strategy that was used to handle those missing values, was to remove those features if they have more than 25% of missing values. After performing that step, the number of remaining variables decreased to 61. At a later stage, we applied an imputation process to substitute the remaining missing values of the dataset. We chose to deal with these values in each column, depending on the variable type. In particular, we used the most frequent value for object type columns, while for the rest of them we used the median value.

Moreover, we resolve that many variables that have been left in the dataset should be excluded for two main reasons. The first reason is that many variables lack information value. For instance, *url* variable, which represents a web link to the loan listing, or the *member_id* variable, which matches each borrower to a unique number, cannot provide our prediction model with useful information in order to assess if a loan will default or not. The last reason

for leaving out several variables, is that from the definition of the features of the dataset, there were variables which reveal information that practically is not available before the applicant receives the loan. Two of them are *funded_amnt*, which indicates the total amount committed to that loan at that point in time and *total_pymnt*, which shows the payments received to date for the total amount that was funded.

Since the remaining categorical variables in our dataset are nominal, we used One-Hot Encoding to handle those variables. These variables are *home_ownership*, *verification_status*, *purpose*, *addr_state* and *application_type*. So, with One-Hot Encoding *n* columns are created, where *n* is the number of unique values that the nominal variables can take. So out of those *n* columns for each one of those five variables, only one can have a value equal to 1 while the rest of them will have a value equal to 0.

Feature scaling is a method used to normalize the range of independent variables or features of data, since the range of values among the data varies widely. In our case, we used the standardize option of H20.ai, which is an available hyperparameter in Deep Learning, Generalized Linear Model and K-means algorithms. This option specifies whether to standarize numeric columns to have zero mean and unit variance.

4.2 Data Analysis & Statistics

This section focuses on a very basic step in any data science project, which is the statistical analysis of the data. In general, statistics is a branch of science that deals with the collection, organization, analysis of data and drawing of inferences from the samples to the whole population. The results and inferences are precise if proper statistical tests are used [25]. Descriptive statistics are used to give an insight to the possible relationship that could happen between several variables in a sample or population. Therefore, they enable us to depict the data in a more meaningful way, which allows a simpler apprehension of the data.

4.2.1 Correlation Analysis

Correlation analysis is a statistical method used to study and evaluate the strength of the relationship between two quantitative variables. We are going to use correlation analysis in this research work, in order to comprehend if there are possible relations between variables. Contrary to a common belief, correlation analysis does not reveal any final results about the research, as other variables may have had a significant impact on the results.

A correlation between two variables indicates that when there is an orderly change in one variable, there is also an orderly change in the other. It can be said that those variables alter together in a certain span of time. If a correlation is found, depending upon the numerical values measured, it can be either positive or negative. A positive correlation between two variables, means that if one variable increases the other one increases at the same time. Whereas, a negative correlation means that if one variable decreases the other one also decreases [21].

In this work, the Pearson product-moment correlation coefficient was used, which is a measurement of linear association and is denoted by r. Specifically, Pearson product-moment correlation tries to draw a line of best fit through the data of two variables, while the coefficient r, shows the distance of all these data points from this line.
The Pearson correlation coefficient, r, can take a range of values from +1 to -1.

- 0 indicates that there is no association between the variables
- > 0 indicates a positive association
- < 0 indicates a negative association

The strongest positive correlation possible is +1, whereas the strongest negative correlation possible is -1. Therefore, the closer the coefficient to either of those numbers, the stronger the correlation or inverse correlation of the data it serves is.

At this stage of our descriptive statistics section, a visualization of correlation matrix would be presented with a heatmap for an easier extraction and interpretation of results. In the heatmap below the dark red values represent strong correlations, which can be either positive or negative. As the red color tends to be lighter the correlation between those variables becomes weaker. Evidently, each variable has the strongest correlation with itself and it can be seen on the diagonal of the heatmap.

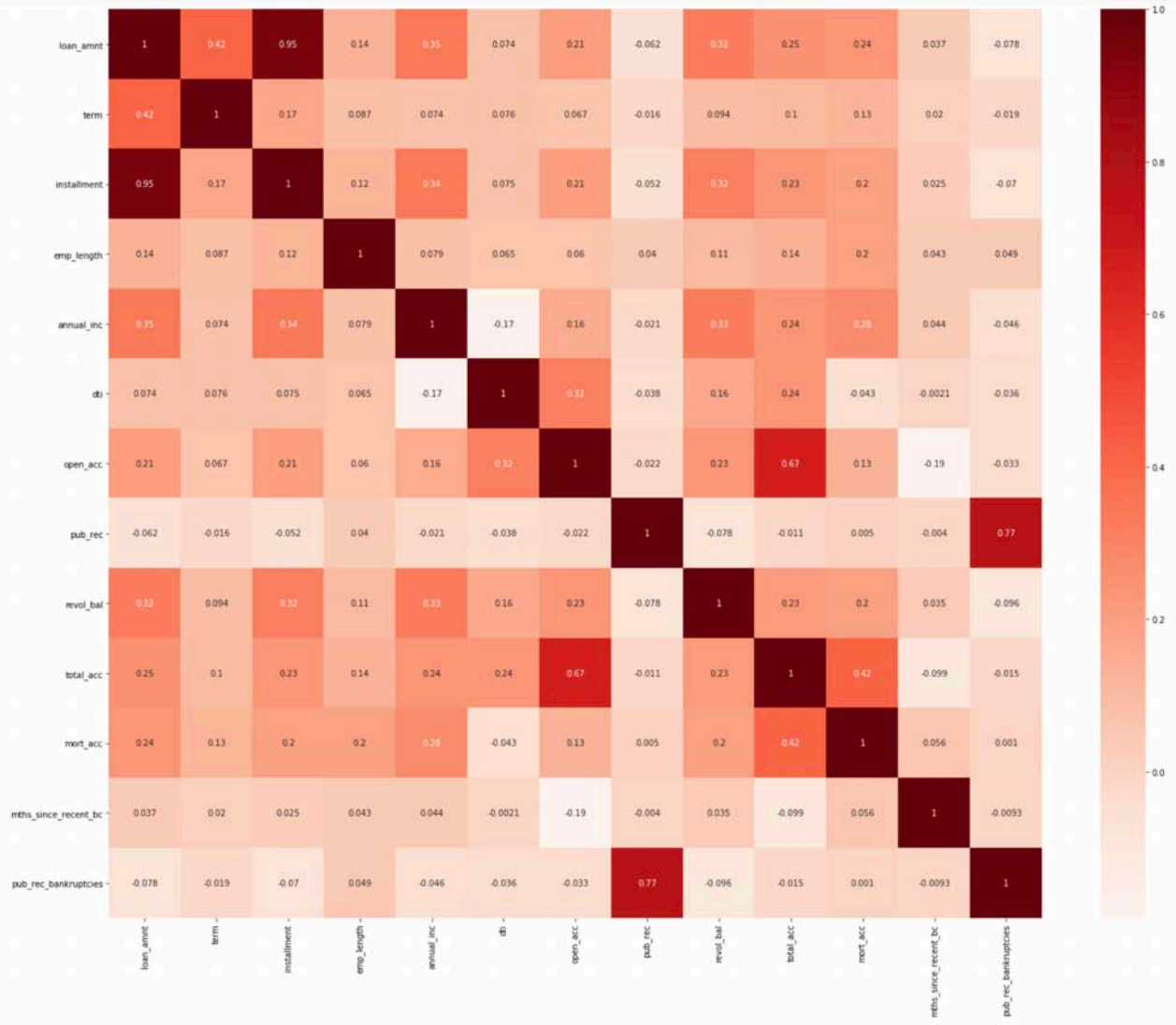

Figure 4.2: Correlation Matrix

With the correlation heatmap of the continuous variables in our dataset, we are interested to know which variables are correlated and how strong is the potential correlation. In our feature selection strategy, we decided to exclude features that have a higher intercorrelation value than 0.5 in absolute terms among the independent variables, as there is no need to keep variables with similar tendencies. Particularly, there are three cases in which the correlation is higher than 0.5 in absolute terms. The strongest correlation is between installment and *loan_amnt*, with a correlation value of 0.95. Furthermore, variable *pub_rec* is considerably correlated *pub_rec_bankruptcies* (r = 0.77), as well as *open_acc* with *total_acc* (r = 0.67). We decided to exclude the *installment*, *pub_rec_bankruptcies* and *total_acc* variables from our dataset.

So, after doing the necessary data preprocessing and excluding several variables, the format of our dataset that was used in the modeling section consisted from the variables shown in Table 4.1, along with their descriptions.

Abbreviated Name	**Description**
loan_status	Current status of the loan
term	The number of payments on the loan. Values are in months and can be either 36 or 60
int_rate	Interest Rate on the loan
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years
home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
annual_income	The self-reported annual income provided by the borrower during registration
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value
purpose	A category provided by the borrower for the loan request
addr_state	The state provided by the borrower in the loan application
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the

	requested LC loan, divided by the borrower's self-reported monthly income
earliest_cr_line	The month the borrower's earliest reported credit line was opened
open_acc	The number of open credit lines in the borrower's credit file
pub_rec	Number of derogatory public records
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
mort_acc	Number of mortgage accounts
mths_since_recent_bc	Months since most recent bankcard account opened

Table 4.1: Final dataset variables description

4.2.2 Distribution Analysis

Distributions (or generalized functions) are objects that generalize the classical notion of functions in mathematical analysis. Distributions make it possible to differentiate functions whose derivatives do not exist in the classical sense. In particular, any locally integrable function has a distributional derivative. Distribution plots visually assess the distribution of sample data by comparing the empirical distribution of the data with the theoretical values expected from a specified distribution. Information for the distribution of the numeric variables can be revealed. Several types of plots can be used in this analysis, such as histograms, probability plots and boxplots. Plots of some basic features of our dataset are being displayed.

Histograms

A histogram is a representation of the distribution of numerical data and consists an area diagram. It can be defined as a set of rectangles with bases along the intervals between class boundaries and with areas proportional to frequencies in the corresponding classes. In such representations, all the rectangles are bordering with each other since the base covers the intervals between class boundaries. The heights of rectangles are proportional to corresponding frequencies of similar classes and for different classes, the heights will be proportional to corresponding frequency densities.

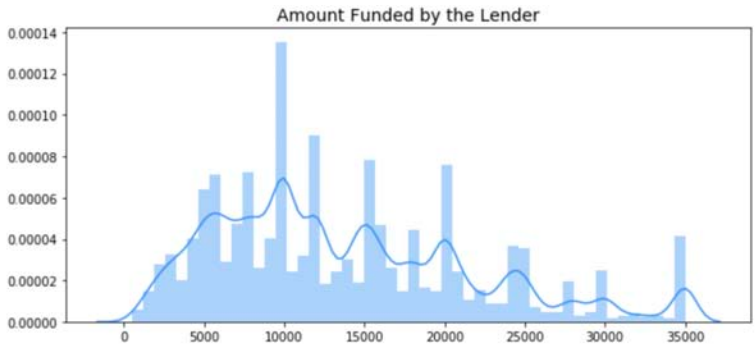

Figure 4.3: Distribution of loan amounts funded

On the histogram above we observe that most of the loans that were funded were between $5,000 and $20,000. Higher loans seem to have a much lower frequency.

Bar Charts

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

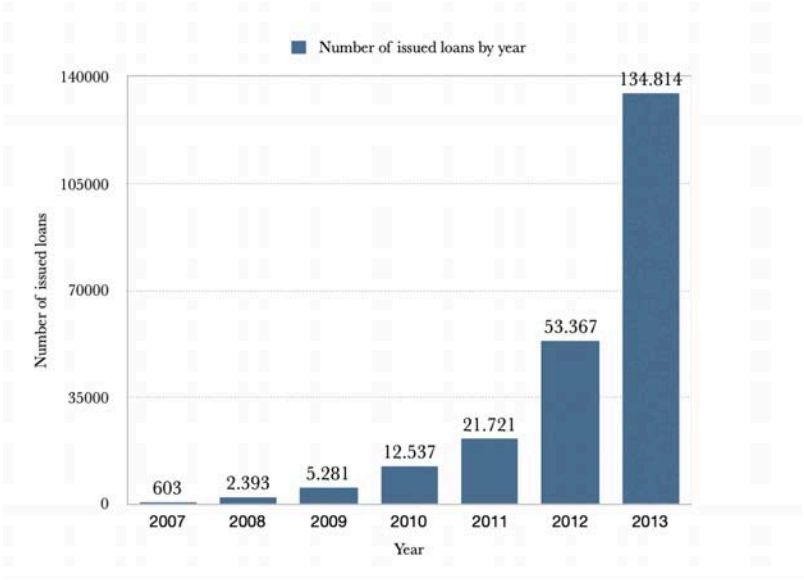

Figure 4.4: Issued loans per year

Figure 4.4 shows the number of issued loans for each year, during the time span of 2007 to 2013. It appears that there is a rapid growth of issued loans through those years.

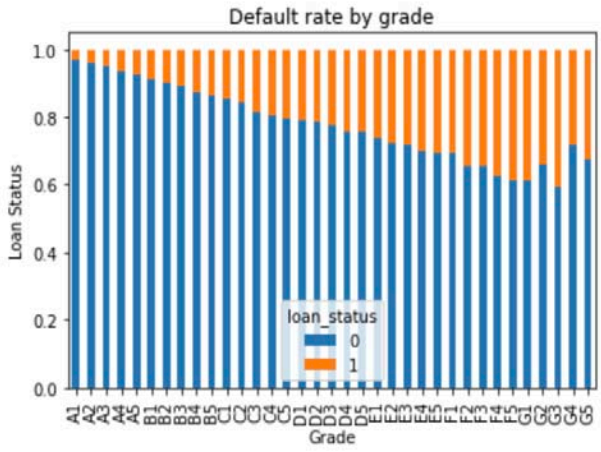

Figure 4.5: Grade distribution

Figure 4.5 depicts the relationship between loan status and grade. It can be seen that the higher the grade, the more probable for a borrower to pay a loan. However, *grade* and *sub_grade* features are based on Lending Club's (LC) assessment of a borrower's credit worth and thus we are not going to use them in our modeling process, as we want to build a more robust model.

4.3 Training, Validating and Testing Data

At this stage, the original dataset has to be divided in order to be able to evaluate our model's performance. As there are many different methods and variations of splitting the dataset, we are going to describe the three basic ways of generating our divided sets.

The first way of splitting the dataset, is to separate it into training and testing set, with the common split ratio to be around 70% or 80% for the training and the remaining for the testing data. In the second method, the original dataset split into three sets. One for the training, one for the validation and the last one for the testing data. Firstly, the model is trained and then the validation set is used to evaluate the performance of the model for different combinations of hyper parameter values and then keep the best trained model. With this method, problems like overfitting can be avoided, as even if a model has nearly perfect metrics on the testing data, it might perform poorly on testing. So, in order to identify those problems, we have to take into consideration our validation metrics. The last method is called k-fold cross validation which is often the preferred one because it provides the model with the opportunity to train on multiple train-test splits without having to sacrifice a validation split. Specifically, this technique involves randomly dividing the dataset into k groups or folds of approximately equal size.

The first fold is kept for testing and the model is trained on k-1 folds. This process is repeated k times and each time a different fold is used for validation. As that process is repeated k times, Mean Squared Error (MSE) is being computed k times and the Cross-Validation error is the average of the MSE over k folds. The cross-validation error formula is:

$$cv_k = \frac{1}{k} \sum_{i=1}^{k} MSE_i$$

Although there is higher computational intensity and more added time, k cross-validation reduces the bias as every data point gets to be tested exactly once and is used in training k-1 times.

In our thesis we used 80% of the data for training set and 20% for test set, applying also a 5-fold cross validation. That means that our original dataset has been randomly split into five subsets.

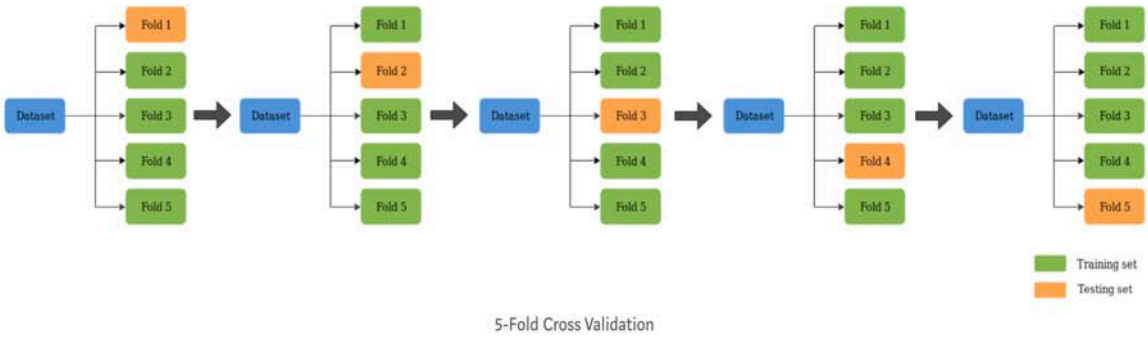

Figure 4.6: 5-Fold Cross Validation [19]

4.4 Algorithms & Parameter tuning

In context of the machine learning process, it is essential to choose the correct approach for tackling a specific task appropriately. In our case, we used machine learning methods to predict if a potential borrower would default their loan or not. So, this chapter describes theoretically all the classifiers as well as the AutoML interface, that were used for this task from H20.ai platform.

4.4.1 Distributed Random Forest (DRF)

Distributed Random Forest consists a powerful classification and regression tool. In general, random forest algorithm belongs to the decision tree family. However, it is an ensemble algorithm, meaning that more than one decision tree model is constructed, and then their results are used together to cope better with unseen situations [23]. The basic idea behind this, is that by constructing multiple decision trees, overfitting can be avoided.
More specifically, when given a set of data, Random Forest generates a forest of classification trees. Each one of those trees forms a weak learner, built on a subset of rows and columns. Thus, every weak learner is trained on a different data sample where sampling is done with replacement. The final prediction is made from the majority vote prediction from all the weak learners.
A step by step explanation of Random Forest algorithm is as follows:

1. Draw *ntree* bootstrap samples from the original data.

2. For each of the bootstrap samples, grow an unpruned classification tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample *m* of the predictors and choose the best split from among those variables.

3. Predict new data by aggregating the predictions of the tree trees (majority vote).

4.4.2 Gradient Boosting Machine (GBM)

Gradient Boosting Machine is another decision tree algorithm, just like random forest. It is a forward learning ensemble method, meaning that a combination of multiple decision tree outputs is made in order to make the final prediction. Gradient boosting combines two powerful tools: gradient-based optimization and boosting, which consist the central idea of the algorithm. Gradient-based optimization utilizes gradient computations to minimize the loss function of a model with respect to the training data [33]. The idea of boosting was perceived, by questioning whether a weak learner can be modified in order to become better. The classifiers are built sequentially, and each member of the ensemble constitutes an expert on the errors of the former members [26].

H2O's Gradient Boosting Machine

The Gradient Boosting Machine algorithm of H2O builds sequentially builds regression trees on all the features of the dataset in a fully distributed way, as each tree is built in parallel.

H2O's implementation of GBM uses distributed trees [13]. H2O overlays trees on the data by assigning a tree node to each row. The nodes are numbered, and the number of each node is stored in a temporary vector as "Node ID" for each row. H2O makes a pass over all the rows using the most efficient method, which may not necessarily be numerical order. A local histogram using only local data is created in parallel for each row on each node. The histograms are then assembled, and a split column is selected to make the decision. The rows are re-assigned to nodes and the entire process is repeated.

A MapReduce (MR) task computes the statistics and uses them to make an algorithmically-based decision, such as lowest mean squared error (MSE). H2O computes the stats for each new leaf in the tree, and each pass across all the rows builds the entire layer.

Each layer represents another MR task. For example, a tree that is five layers deep requires five passes. Each tree level is fully data-parallelized. Each pass is over one layer in the tree and builds a per-node histogram in the MR calls. As each pass analyzes a tree level, H2O then decides how to build the next level. H2O reassigns rows to new levels in another pass by merging the two passes and builds a histogram for each node. Each per-level histogram is done in parallel [13].

Even though the GBM algorithm builds each tree level at a time, H2O has the ability to quickly parallelize and distribute the run of an entire level.

4.4.3 Deep Learning (Neural Networks)

Neural networks or Artificial Neural Networks are an important class of tools for quantitative modeling [20]. Artificial Neural Networks are defined as a mathematical or computational model based on biological neural networks, and thus it consists an emulation of the biological neural system [28]. It is a hardly comprehensible algorithm due to its neutron mechanism with hidden layers. Though, ANN consists a very famous and powerful algorithm which can be applied a lot of complex machine leaning problems.
A typical Artificial Neural Network is organized in layers. Those layers are being made up of many interconnected nodes and represent the neurons. Each one of those neurons contains an activation function. A neural network may contain the following 3 layers:

- Input layer: Consists from the so-called passive neurons, as they do not change the data. It receives the information and transfers it to the neural network.

- Hidden layer: Apply given transformations to the input values inside the network. The values entering a hidden node are multiplied by certain weights. There may be one or more hidden nodes.

- Output layer: Receives connections from hidden layers or input layer and returns an output value that corresponds to the prediction of the response variable.

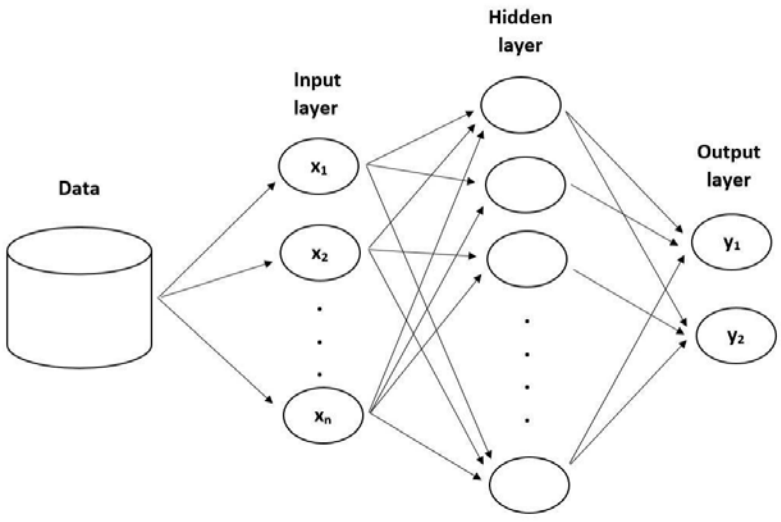

Figure 4.7: Artificial Neural Network [8]

The Deep Learning algorithm of H2O is based on a multi-layer feedforward artificial neural network (ANN) that is trained with stochastic gradient descent using back-propagation. The feedforward ANN, which is also known as Deep Neural Network (DNN) or multi-layer perceptron is the only supported type of Deep Neural Network in H2O.

Backpropagation

Backpropagation is a method to calculate the gradient of the loss function with respect to the weights in an ANN. More specifically, backpropagation distributes the error term back up through the layers, by repeatedly updating the weights at each node.
By using a stochastic gradient descent, the weight updates are calculated as following:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\vartheta C}{\vartheta w_{ij}} + \xi(t)$$

Where, η is the learning rate, C is the loss function and $\xi(t)$ a stochastic term. The choice of the loss function depends on factors such as the learning type and the activation function.

4.4.4 Stacked Ensembles

Ensemble machine learning methods use multiple learning algorithms so as to obtain better predictive performance than one that could be derived from any constituent learning algorithm. Some of the machine learning algorithms that were previously mentioned, such as Random Forest and Gradient Boosting Machine, are also ensemble learners.
The Stacked Ensemble method of H2O is a supervised ensemble machine learning algorithm that finds the optimal combination from a collection of prediction algorithms with the use of a process called stacking.

Stacking

The process of stacking, which is also named as Super Learning, is a class of algorithms which includes the training of a second-level "metalearner" in order to find the optimal combination of the base learners that were used.

The main discrimination between that Super Learner ensemble and some other ensemble methods that are also labeled as stacking, is the use of cross-validation so as to form the "level-one" data, which the "metalearner" is going to be trained on. That means, that the base models need to be trained and cross-validated before training the stacked ensemble.

In order to make the process of stacked ensemble algorithm more understandable a step by step clarification will be presented. Those steps are:

1. Set up the ensemble
    - Specify a list of L base algorithms
    - Specify a metalearning algorithm

2. Train the ensemble
    - Train each of the L base algorithms on the training set
    - Perform k-fold cross-validation on each of the learners and get the cross-validated predicted values from each of the L algorithms
    - A new N x L matrix is formed from the N cross-validated predicted values and the L algorithms, which along with the response variable form the "level-one" data
    - The metalearning algorithm is trained on the "level-one" data

3. Predict on new data
    - First generate predictions from the base learners
    - In order to generate the ensemble prediction, feed the base learner predictions into it

The Stacked Ensemble model we built was trained on the “level-one” data that were provided from a list of algorithms that included the Distributed Random Forest, Gradient Boosting Machine and Deep Learning, which we used as base models earlier.

4.4.5 AutoML (Automatic Machine Learning)

AutoML is an H2O interface that is used to automate the building process of many models, aiming to find the model with the best performance, without the need of any profound knowledge of the machine learning modeling process. AutoML can also be a helpful tool for advanced users, by providing a simple wrapper function that performs a large number of modeling-related tasks such as the parameterization, that would typically require a lot of code lines. Therefore, it provides the user with more time to focus on other tasks of the machine learning process such as data preprocessing and feature engineering.

AutoML trains and cross-validates the following algorithms (in the following order): three pre-specified XGBoost GBM (Gradient Boosting Machine) models, a fixed grid of GLMs, a default Random Forest (DRF), five pre-specified H2O GBMs, a near-default Deep Neural Net, an Extremely Randomized Forest (XRT), a random grid of XGBoost GBMs, a random grid of H2O GBMs, and a random grid of Deep Neural Nets. In some cases, there will not be enough time to complete all the algorithms, so some may be missing from the leaderboard. AutoML then trains two Stacked Ensemble models. In case there is a prior knowledge of what models would perform better in a specific dataset, exclude_algos argument can exclude several models, in order to accelerate the AutoML process.

Required Parameters

The design of H2O AutoML interface, requires as few parameters as possible. Specifically, it only requires the response column and the training set to run the models. Likewise, several optional parameters can be used, such as the predictor column names, the time limit for running the AutoML and the maximum number of models to build.
In our work, we set a maximum number of models at 30 and provided the function with the response and the independent variables.

AutoML Output

The AutoML object produces a “leaderboard” of models that were trained in the process. The models are ranked by a default metric based on the problem type we face. In our case, we have a binary classification problem, and the default metric is AUC.

4.4.6 Grid (Hyperparameter) Search

Grid Search is referred in this chapter as it was a part of the modeling procedure, even though it does not constitute a machine learning algorithm. As tuning the hyperparameters of model manually, until a great combination of values is found can be very tedious work, a more automatic process can be used instead [15]. Grid search is the process of performing hyper parameter tuning so as to define the optimal parameter values for the model we need to train. This procedure is of a significant importance as the performance of the entire model is based on the hyper parameter values specified.

H2O supports two types of grid search. Those are the Cartesian grid search and the random grid search. The one that was used in this research work is the Cartesian grid search. That means that a specific set of values has been selected, so as H2O would train a number of models with different combinations of those values and then compare the model performance to choose the best model. H2O would return a sorted list with the best model first, using the appropriate performance metric.

In this work, a grid search was used to tune the parameters for a gradient boosting machine model.

Chapter 5

Model Evaluation & Empirical Results

In this chapter, we demonstrate the performance metrics we used in our research work to evaluate the models, as well as the results we obtained from the models that were analyzed in the previous chapter. Before that, we first introduce the concept of the confusion matrix which is a prerequisite for the explanation of our performance metrics.

5.1 Performance Metrics

The concept of ROC and AUC builds upon the knowledge of confusion matrix, specificity and sensitivity. So, we are going to give an insight to those terms at first, and then we will proceed to the analysis of the performance metric we used for our model results.

5.1.1 Confusion Matrix

Confusion matrix is a performance measurement which is used for classification problems where the output consists of two or multiple classes. It is a table with four different combinations of predicted and actual values as shown in Figure 5.1 below.

		Predicted Class	
		Negatives	Positives
Actual Class	Negatives	TN	FP
	Positives	FN	TP

Table 5.1: Confusion Matrix

In the context of our research entrance to confusion matrix have the following meanings:

- TN is the number of correct predictions that instances are negative

- FP is the number of incorrect predictions that instances are positive
- FN is the number of incorrect predictions that instances are negative
- TP is the number of correct predictions that instances are positive

5.1.2 Specificity & Sensitivity

Specificity reveals the proportion of actual negatives that are correctly identified as such and is given by the formula:

$$specificity = \frac{TN}{TN + FP}$$

Sensitivity reveals the proportion of actual positives that are correctly identified as such and is given by the formula:

$$sensitivity = \frac{TP}{TP + FN}$$

5.1.3 AUC-ROC curve

The performance measurement we decided to use in order to evaluate the performance of the classifiers along with the leader model of the AutoML function is the Area Under the Curve (AUC). AUC - ROC curve consists one of the most important evaluation metrics when it comes to a classification problem. Receiver Operator Characteristic Curve (ROC) is a curve of probability, whereas AUC represents a degree or measure of the ability of a model to distinguish classes.

ROC can help in deciding the most suitable threshold value in order to get the best results. It is generated by plotting the True Positive Rate (TPR) on y-axis against the False Positive Rate on x-axis. Those two rates are calculated from the metrics of sensitivity and specificity, which were described earlier.

$$True\ Positive\ Rate = \ sensitivity\ = \frac{TP}{TP + FN}$$

$$False\ Positive\ Rate = (1 - \text{specificity})\ = \frac{FP}{FP + TN}$$

AUC measures the entire two-dimensional area underneath the entire ROC curve and ranges from (0,0) to (1,1). It provides an aggregate measure of performance across all possible classification thresholds. AUC values range from 0 to 1.

- 0 indicates a model whose predictions are 100% wrong
- 1 indicates a model whose predictions are 100% correct
- 0.5 indicates a model with no class separation capacity

5.2 Empirical Results

This section shows the results of the classifiers that were used. Specifically, we demonstrate the confusion matrix and AUC which were produced from each model on training and validating set. The graphs were obtained from H20 Flow interface.

5.2.1 Distributed Random Forest Performance

Training AUC & Confusion Matrix

The following figures show the best training error rate and AUC produced by the preprocessed data entered in the Distributed Random Forest model.

▾ TRAINING METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	0	1	Error	Rate	Precision
0	117001	37100	0.2408	37,100 / 154,101	0.89
1	14856	139154	0.0965	14,856 / 154,010	0.79
Total	131857	176254	0.1686	51,956 / 308,111	
Recall	0.76	0.90			

Table 5.2: DRF Training Correlation Matrix

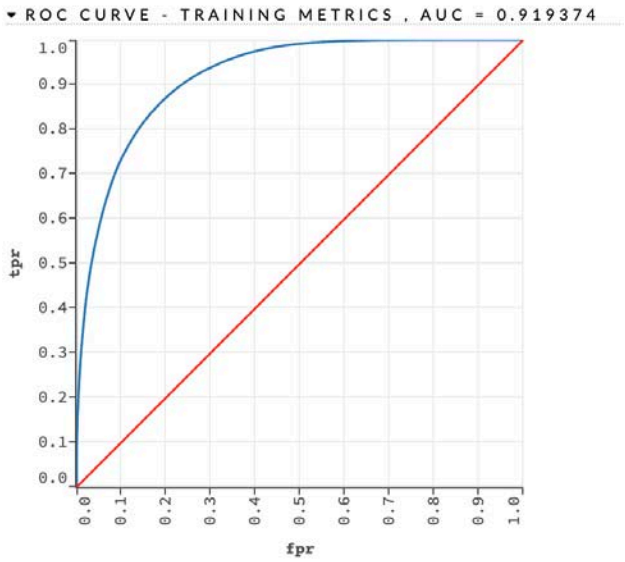

Figure 5.1: DRF Training AUC

Cross-Validation AUC & Confusion Matrix

The following figures show the best validation error rate and AUC produced by the preprocessed data entered in the Distributed Random Forest model.

▾ CROSS VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	0	1	Error	Rate	Precision
0	104551	49550	0.3215	49,550 / 154,101	0.90
1	11804	16500	0.4170	11,804 / 28,304	0.25
Total	116355	66050	0.3364	61,354 / 182,405	
Recall	0.68	0.58			

Table 5.3: DRF Validation Correlation Matrix

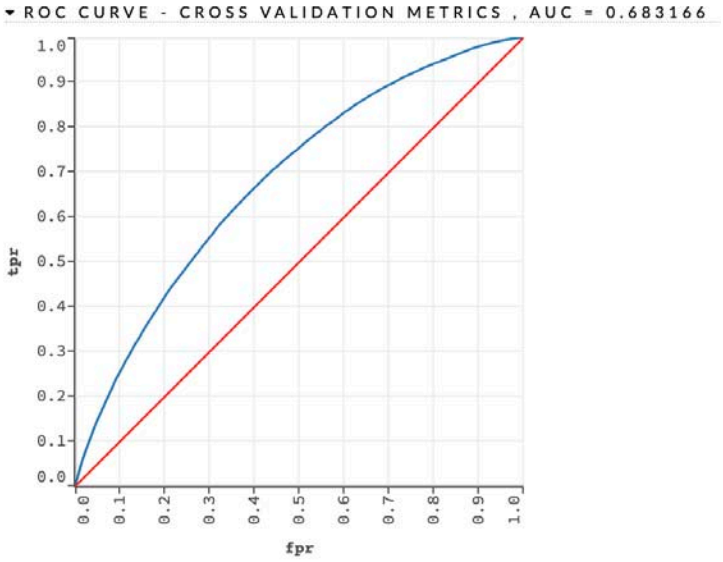

Figure 5.2: DRF Validation AUC

The graphs show a major decrease in the performance of the DRF model from training to validation, with a huge fall of AUC from 0.91 to 0.68.

5.2.2 Gradient Boosting Machine Performance

Training AUC & Confusion Matrix

The following figures show the best training error rate and AUC produced by the preprocessed data entered in the Gradient Boosting Machine model.

TRAINING METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	0	1	Error	Rate	Precision
0	47533	106568	0.6915	106,568 / 154,101	0.74
1	16810	137392	0.1090	16,810 / 154,202	0.56
Total	64343	243960	0.4002	123,378 / 308,303	
Recall	0.31	0.89			

Table 5.4: GBM Training Correlation Matrix

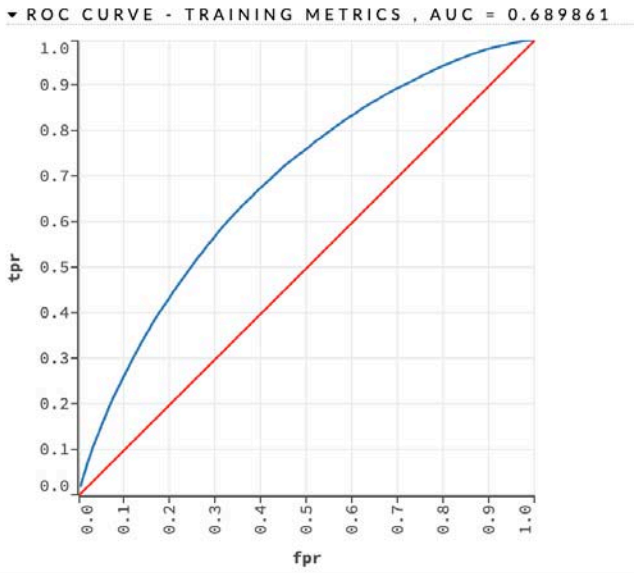

Figure 5.3: GBM Training AUC

Cross-Validation AUC & Confusion Matrix

The following figures show the best validation error rate and AUC produced by the preprocessed data entered in the Gradient Boosting Machine model.

CROSS VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	0	1	Error	Rate	Precision
0	104085	50016	0.3246	50,016 / 154,101	0.90
1	11629	16675	0.4109	11,629 / 28,304	0.25
Total	115714	66691	0.3380	61,645 / 182,405	
Recall	0.68	0.59			

Table 5.5: GBM Validation Correlation Matrix

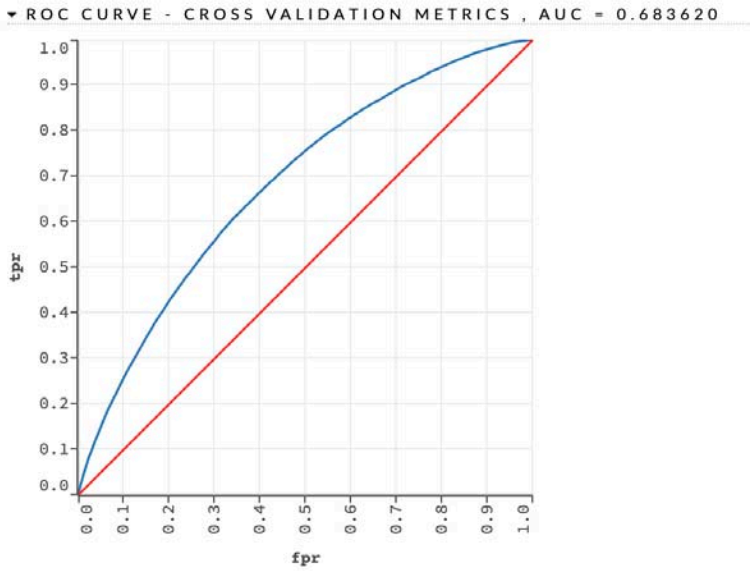

Figure 5.4: GBM Validation AUC

The graphs show a relatively similar behavior in the performance of the GBM model from training to validation, with the AUC falling from 0.69 to 0.68. Although, we can see that the error rate is augmented from 0.33 to 0.40, having a bad predictive ability in both classes.

5.2.3 Gradient Boosting Machine using Grid Search for Parameter tuning Performance

Training AUC & Confusion Matrix

The following figures show the best training error rate and AUC produced by the preprocessed data entered in the Gradient Boosting Machine model with a grid search being used for parameter tuning.

TRAINING METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	0	1	Error	Rate	Precision
0	91198	24478	0.2116	24,478 / 115,676	0.90
1	9780	11501	0.4596	9,780 / 21,281	0.32
Total	100978	35979	0.2501	34,258 / 136,957	
Recall	0.79	0.54			

Table 5.6: GBM-Grid Search Training Correlation Matrix

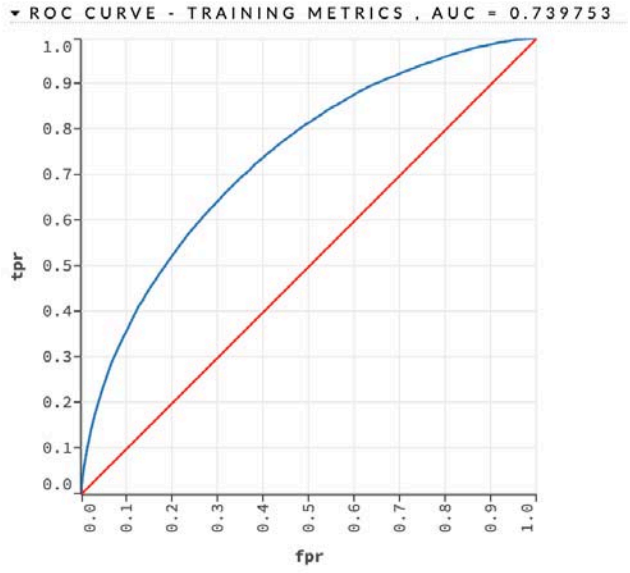

Figure 5.5: GBM-Grid Search Training AUC

Cross-Validation AUC & Confusion Matrix

The following figures show the best validation error rate and AUC produced by the preprocessed data entered in the Gradient Boosting Machine model with a grid search being used for parameter tuning.

CROSS VALIDATION METRICS - CONFUSION MATRIX ROW LABELS: ACTUAL CLASS; COLUMN LABELS: PREDICTED CLASS

	0	1	Error	Rate	Precision
0	79868	35808	0.3096	35,808 / 115,676	0.90
1	9160	12121	0.4304	9,160 / 21,281	0.25
Total	89028	47929	0.3283	44,968 / 136,957	
Recall	0.69	0.57			

Table 5.7: GBM-Grid Search Validation Confusion Matrix

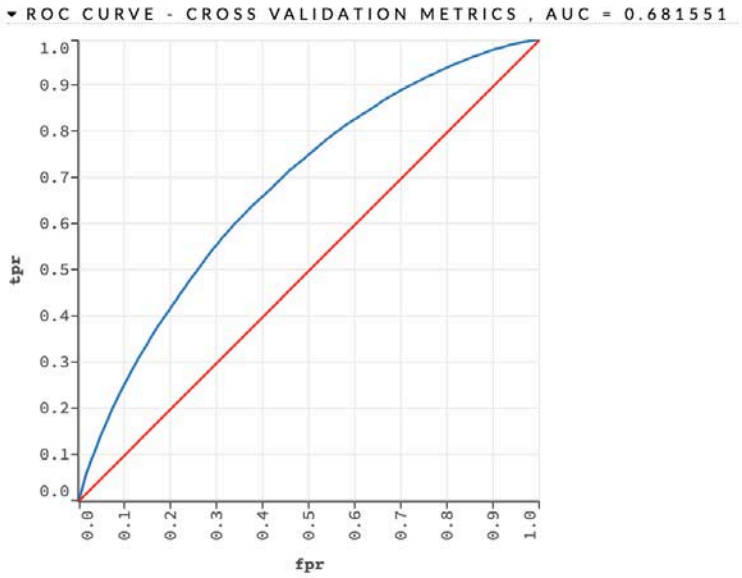

Figure 5.6: GBM-Grid Search Validation AUC

The graphs show a better performance of the GBM model tuned with a grid from the previous GBM model in training set, whereas it performed almost similar in validation set. A slight drop was observed in AUC, falling from 0.74 to 0.68.

5.2.4 Deep Learning Performance

Training AUC

The following figures show the best training AUC produced by the preprocessed data entered in the Deep Learning model.

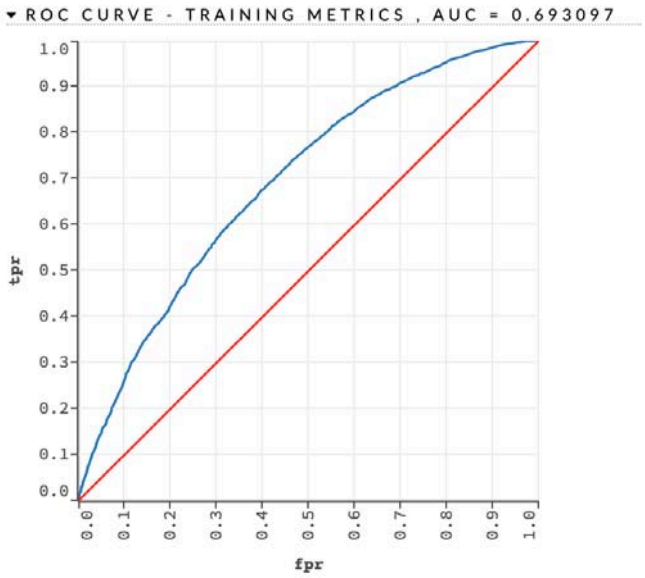

Figure 5.7: DL Training AUC

Cross-Validation AUC

The following figures show the best validation AUC produced by the preprocessed data entered in the Deep Learning model.

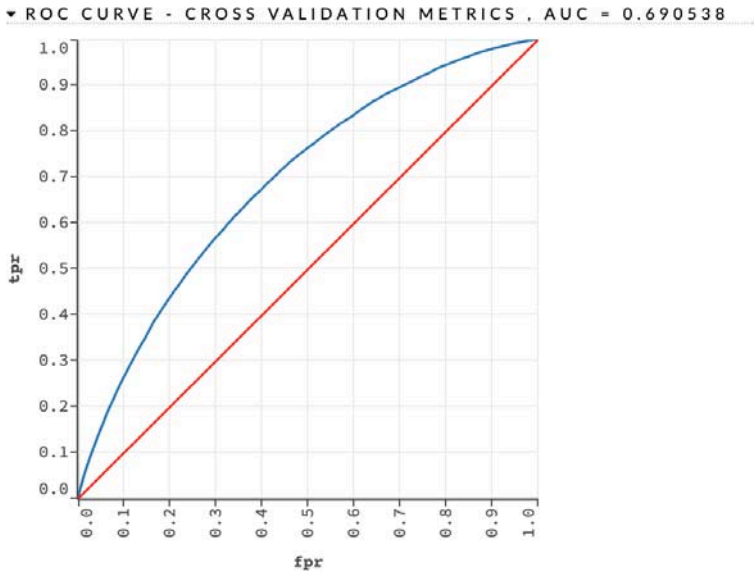

Figure 5.8: DL Validation AUC

The graphs show that the model behaves similarly in both training and validating set with an AUC at 0.69.

5.2.5 Stacked Ensemble Performance

The Stacked Ensemble we built used as base models the Distributed Random Forest model, Gradient Boosting Machine model with grid search and Deep Learning model. The achieved AUC at training set is 0.83. As there are no validation metrics, graphs for this model were not obtained.

5.2.6 Stacked Ensemble of AutoML Performance

Training AUC

The following figures show the best training AUC produced by the preprocessed data entered in the Stacked Ensemble model of AutoML.

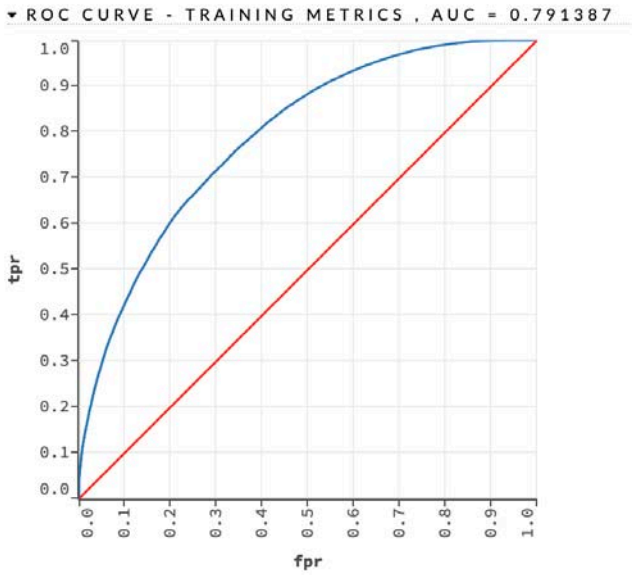

Figure 5.9: SE-AutoML Training AUC

Cross-Validation AUC

The following figures show the best validating AUC produced by the preprocessed data entered in the Stacked Ensemble model of AutoML.

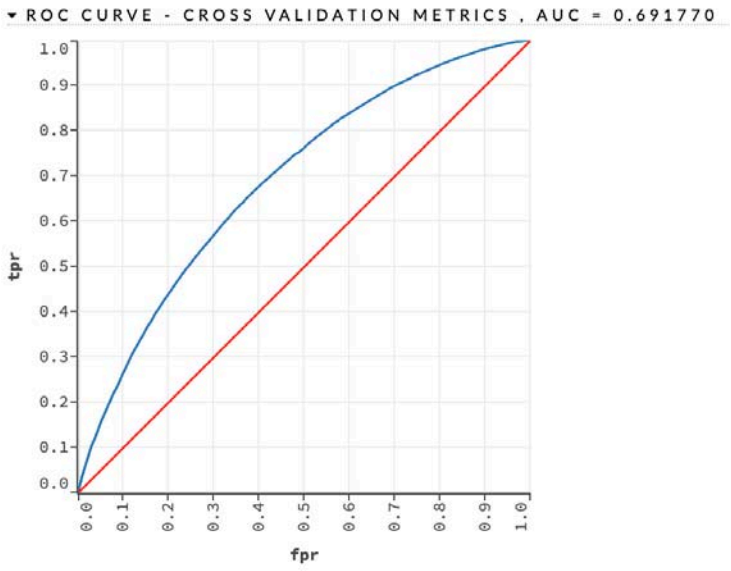

Figure 5.10: SE-AutoML Validation AUC

The graphs show that the Stacked Ensemble model trained with AutoML, decreased significantly its performance from training to validation set.

Chapter 6

Conclusion

6.1 Results Summary

In this part, aggregated results of the models used to predict the loan status parameter of a borrower are presented. The following table demonstrates a comparison of AUC in training, validation and test set. The best behavior is shown by the Stacked Ensemble model, which was the leader model of the AutoML function of H2O.

Classifiers	Train AUC	Validation AUC	Test AUC
DRF	0.92	0.68	0.68
GBM	0.69	0.68	0.68
GBM (Grid Search)	0.74	0.68	0.68
DL	0.70	0.69	0.69
SE	0.84	-	0.69
SE (AutoML)	0.79	0.69	0.70

Table 6.1: Results Summary

It is obvious, that the model with the best performance metrics in test data seems to be the Stacked Ensemble model of H2O's AutoML, achieving AUC equal to 0.70.

6.2 Conclusion

In this research, an approach of how distributed machine learning algorithms, and specifically algorithms provided by the H2O platform, could be used to a credit risk topic is shown, including also the relevant data analysis and modeling process. In the data analysis section, data charts were presented in order to show the distribution of several important features. Also, a correlation matrix was used to depict how each feature is associated with each other.

In particular, we focused on Lending Club dataset and the prediction of a potential borrower's loan status. As we faced a binary classification problem, the possible classification results could be charged off and fully paid. The specific dataset takes place over a time span of 5 years.

From the predictive models developed and tested on the above data, turned out to be the Stacked Ensemble constructed from H2O's AutoML, the one that achieved the highest AUC score on test data. We can conclude, that a tool such as H2O can be really beneficial for a data scientist as it provides the opportunity to reduce processing time with the parallelization of the algorithms, as well as automate the modeling process.

6.3 Future Work

Further research can be done in order to come up with more interesting and better results in the future. Extending the work on this topic, an attempt to use H2O on a multi-node cluster and in larger datasets could be done, so as to observe the speed up of the processing procedure. Moreover, sentence and sentiment analysis could be used on text features of the initial dataset in order to improve the overall performance of the classifiers.

References

[1] Thomas, L. C., *A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers*. International Journal of Forecasting, 2000,16 (2), 149–172.

[2] T. Volk, "ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR OPTIMIZING DEVOPS, IT OPERATIONS, AND BUSINESS," ENTERPRISE MANAGEMENT ASSOCIATES, 2018.

[3] S. Rahman, "Combining Machine Learning with Credit Risk Scorecards," *FICO*, 01-Nov-2017. [Online]. Available: https://www.fico.com/blogs/analytics-optimization/ai-combining-machine-learning-with-credit-risk-scorecards/. [Accessed: 04-Feb-2019].

[4] A. E. Khandani, A. J. Kim, and A. W. Lo, "Consumer Credit Risk Models via Machine-Learning Algorithms," p. 55, May 2010.

[5] P. Addo, D. Guegan, and B. Hassani, "Credit Risk Analysis Using Machine and Deep Learning Models," *Risks*, vol. 6, no. 2, p. 38, Apr. 2018.

[6] S. Goyal and V. Blog, "Credit Risk Prediction Using Artificial Neural Network Algorithm," 14-Mar-2018. [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/credit-risk-prediction-using-artificial-neural-network-algorithm?fbclid=IwAR05Wn_qMbxhl5oTSYAewPcr6GAWja0KrrHwxdeW4jf47iyc-CqUH4bpWos. [Accessed: 07-Feb-2019].

[7] E. Sharova, "Data Analytics Models in Quantitative Finance and Risk Management," *KDnuggets*. [Online]. Available: https://www.kdnuggets.com/2016/12/data-analytics-models-quantitative-finance-risk-management.html. [Accessed: 04-Feb-2019]. *****

[8] T. Wendler and S. Gröttrup, *Data Mining with SPSS Modeler: Theory, Exercises and Solutions*. Springer International Publishing, 2016.

[9] T. Ben-Nun and T. Hoefler, “Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis,” *arXiv:1802.09941 [cs]*, Feb. 2018.

[10] S. Omatu, S. Rodríguez, G. Villarrubia, P. Faria, P. Sitek, and J. Prieto, Eds., *Distributed computing and artificial intelligence, 14th International Conference*, vol. 620. New York, NY: Springer Berlin Heidelberg, 2017.

[11] J. Kagan, “FICO Score,” *Investopedia*, 23-Jan-2018. [Online]. Available: https://www.investopedia.com/terms/f/ficoscore.asp. [Accessed: 04-Feb-2019].

[12] T. Nykodym, T. Kraljevic, A. Wang, and W. Wong, *Generalized Linear Modeling with H2O*, 7th ed. 2019.

[13] C. Click, M. Malohlava, V. Parmar, H. Roark, and J. Lanford, “Gradient Boosted Machines with H2O.” H2O.ai, 2015.

[14] W. Zhou, “H2O vs Sparkling Water,” *My Big Data World*, 07-Nov-2017.

[15] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O’Reilly Media, Inc., 2017.

[16] S. Zoldi, “How to Build Credit Risk Models Using AI and Machine Learning,” *FICO*, 06-Apr-2017. [Online]. Available: https://www.fico.com/blogs/analytics-optimization/how-to-build-credit-risk-models-using-ai-and-machine-learning/. [Accessed: 04-Feb-2019].

[17] S. Anna, K. Boris, and W. Ivana, “Impact of Credit Risk Management,” *Procedia Economics and Finance*, vol. 26, pp. 325–331, Jan. 2015.

[18] J. Robinson, “Introduction to H2O.ai,” *Jamal Robinson*, 02-Mar-2018.

[19] K. Hewa, “K-Fold Cross Validation,” *Data Driven Investor*, 16-Dec-2018.

[20] G. P. Zhang, “Neural Networks For Data Mining” in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2009, pp. 419–444.

[21] “Pearson Product-Moment Correlation - When you should run this test, the range of values the coefficient can take and how to measure strength of association.” [Online]. Available: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php. [Accessed: 05-Jul-2019].

[22] K. Tsai, S. Ramiah, and S. Singh, “Peer Lending Risk Predictor,” p. 5, 2014.

[23] D. Cook, *Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI*. O’Reilly Media, Inc., 2016.

[24] Y. Yenigün, “Spark ML to H2O Migration for Machine Learning in iyzico,” *iyzico.engineering*, 01-Nov-2017. [Online]. Available: https://iyzico.engineering/spark-ml-to-h2o-migration-for-machine-learning-in-iyzico-dcba86b8eab2. [Accessed: 07-Feb-2019].

[25] R. Winters, A. Winters, and R. G. Amedee, “Statistics: A Brief Overview,” vol. 10, no. 3, p. 5, 2010.

[26] T. Hastie, S. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2017.

[27] R. G. Lopes, M. Ladeira, and R. N. Carvalho, “Use of machine learning techniques in the prediction of credit recovery,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 3, pp. 1432–1442, Aug. 2017.

[28] Nashaat El-Khamisy Mohamed, Ahmed Shawky Morsi El-Bhrawy, “Artificial Neural Networks in Data Mining”, IOSR Journal of Computer Engineering (IOSR-JCE) eISSN: 2278-06,p-ISSN: 2278-8727, Volume 18, Issue 6, Ver. III (Nov.-Dec. 2016).

[29] O. Amat, R. Manini, and M. Antón Renart, “Credit concession through credit scoring: Analysis and application proposal,” *Intangible Capital*, vol. 13, no. 1, p. 51, Jan. 2017.

[30] R. Anderson, *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation*. Oxford: Oxford University Press, 2007.

[31] Gup, B. E., & Kolari, J. W. (2005). Commercial Banking: The management of risk. Alabama: John Wiley & Sons, Inc

[32] L. C. Thomas, D. B. Edelman, and J. N. Crook, *Credit Scoring and Its Applications*. SIAM, 2002.

[33] James Finance, “Machine Learning in Credit Risk Modeling.” CrowdProcess Inc., Jul-2017.