PREDICTIONS OF LOAN E-SIGNING BASED ON FINANCIAL STATS OF APPLICANTS USING MACHINE LEARNING

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Apoorv Vats (181308)

Under the supervision of

Dr. Himanshu Jindal

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat,

Solan-173234, Himachal Pradesh

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **"Predictions of Loan E-Signing Based on Financial Stats of Applicants Using Machine Learning"** is in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to May 2021 under the supervision of **Dr. Himanshu Jindal** (Assistant Professor (SG), Dept. CSE & IT).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Apoorv Vats, 181308

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Himanshu Jindal
Assistant Professor (SG)
Department CSE & IT

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for his divine blessing that makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Himanshu Jindal**, **Assistant Professor(SG)**, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of **"Artificial Intelligence"** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Himanshu Jindal**, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Apoorv Vats (181308)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

1. ML: Machine learning
2. ANN: Artificial neural network
3. SVM: Support Vector Machine

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

A lot of intermediaries have been operating between the bank and the applicant so that the applicant can fill the form but since these intermediaries receive a lot of applications it has become tough to analyze them manually. Therefore, it has become necessary to create a Machine Learning model so as to make the task simple. Various researchers on e-signing have published a number of research papers, however, the algorithms used in previous research papers achieved lesser accuracy on the given dataset. The aim of this project is to improve the efficiency or accuracy of e-signing using various machine learning algorithms.

CHAPTER 1: INTRODUCTION

1.1 Introduction

Representing information and data in the form of graphics is called Data Visualization and their tools are used to see trends, patterns, etc. in data. It has been used in this paper to better understand the Financial History of the applicant. Financial history is the record of a person's current loans, their dates, amount, etc. Since, the Financial History contains the data related to a person's finances, it is easy to find out whether a person has that many assets using which he would be able to pay back the loan. This way we use Financial History to determine the e-signing of loans. Several techniques are there for e-signing of loans such as digitized loans. We can make use of Financial History to determine the e-signing of the loan which leads us to the introduction of Digitized loan.

1.2 Problem Statement

Digitized loans are used to maintain financial transactions. Digital lending/digitized loan is basically making use of web-based computing so as to originate and renew loans for speedy and efficient decisions. In Digital lending, an online loan application is offered by a bank on its website, and it can be completely automated. It helps financial institutions by providing a lot of opportunities to improve productivity, etc. with cheaper, faster, and automated services. Lending companies basically work by analyzing the financial history of the loan applicant and then determining whether the applicant is too risky or not. An optimized model is required to predict whether the applicant using the website will be risky or not for the loan by using the e-signing process.

1.3 Objectives

The main objectives of our project are understanding the financial data of a person and using an optimized model to predict whether the applicant using the website will be risky or not for the loan by using the e-signing process. Manually doing this task is very difficult due to the number of loan applications and also, it is more time-consuming.

1.4 Methodology

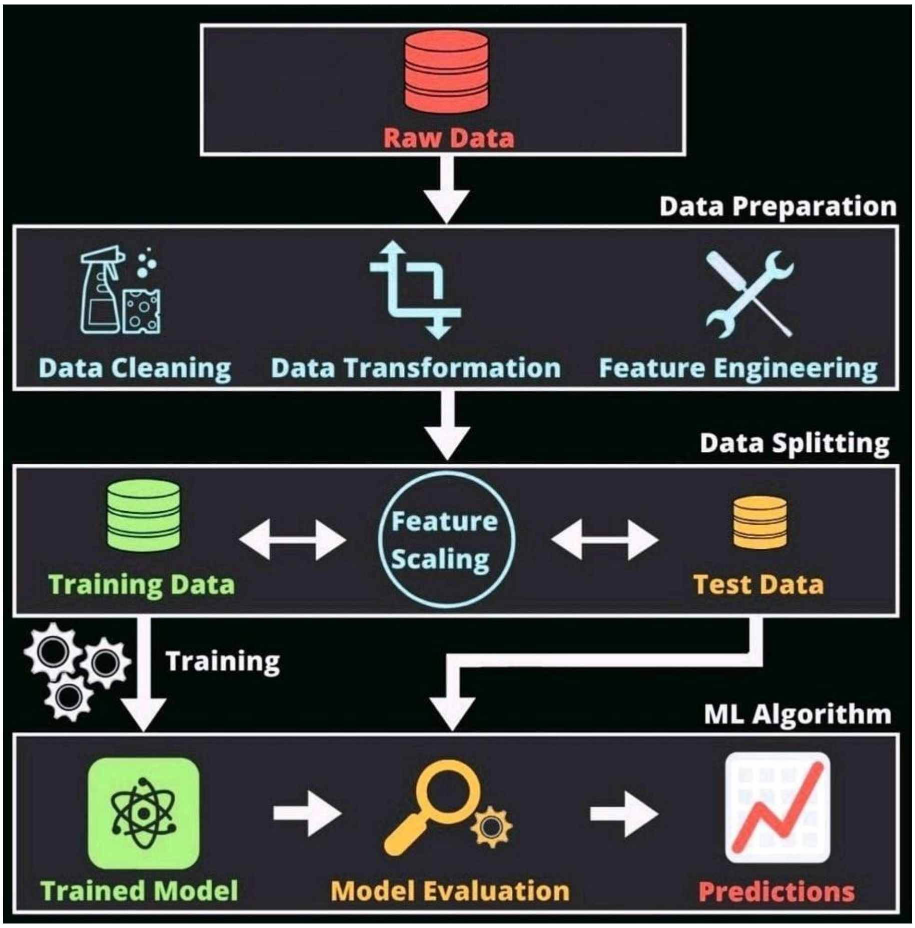


Fig.1 Project Design/ Plan

1.5 Organization

The report is organized as follows: In Chapter 2, literature survey has been presented which depicts the various approaches used by different authors. Chapter 3 highlights the methodology and system development of the project. It represents various computational, experimental and mathematical concepts of the project. Also, we have focused on the software and hardware platforms needed for implementing the model. In chapter 4 we have presented the performance analysis of the project which specifies the accuracy of the project. Also, we have shown the required dataset and its related information. Chapter 5 presents the conclusions of the project and the observations seen in the results. It also provides the applications of the project and the future scope of the same.

CHAPTER 2: LITERATURE SURVEY

Patil *et al.* in 2019, worked on creating a model which had an accuracy of around 64% and has an algorithm that helps in predicting whether a person will be able to complete the e-signing of a loan [1]. An advantage of this model is to target those predicted to not reach the e-sign with customized onboarding. The abstract and idea was taken from above mentioned work and in this paper the accuracy has been improved on a different dataset and have reached an accuracy of *65.04% by implementing the dataset using Stacking Classifier.*

Goyal *et al.* in 2016, worked on a similar paper in which it was evaluated to predict the finance status for an organization in R language [2]. The experiment was done five times on the same data set and Tree Model for Genetic Algorithm was found to be the best model for predicting the finance for customers. Outcome of the paper is to understand the methodology of experimenting several times with the same dataset but with different algorithms.

Chen *et al.* in 2018, worked on increasing the accuracy, lowering the false positive rate, and increasing the speed for SDN controllers, and solving some cloud-related problems like private cloud isolation of users and network flow control using XGBoost [4]. Through the above-mentioned paper, this paper has implemented the working of XGBoost and learnt some keywords which have been used in this paper.

Prokhorenkova *et al.* in 2019, presented a paper that had the key algorithmic techniques behind CatBoost [5]. The methods used in the paper are designed to combat predicting changes caused by a special type of target leakage present in all current applications of gradient reinforcement algorithms. The outcome of the paper was the working of CatBoost and some keywords which have been used in this work.

Guolin Ke *et al.* in 2017, presented a paper that solved the problem of the efficiency and scalability which are inadequate when the feature dimension is high, and the data size is large for the LightGBM [6]. To address this issue, two novels were proposed: One-Side Sampling (GOSS) Gradient-Based and Exclusive Feature Bundling (EFB). Through this paper, LightGBM's working and some keywords were established which have been used in this work.

Daoud *et al.* in 2019, presented a paper on the comparison of XGBoost, CatBoost, and LightGBM [7]. The purpose of this research was to compare effectiveness of the three gradient methods. New features were generated, and various techniques were used for ranking and selection of the best features. The implementation showed that LightGBM when compared to CatBoost and XGBoost is accurate and faster. The outcomes of this paper helped in increasing the accuracy of the model in this paper by implementing all three algorithms together using Voting and Stacking classifier.

Kumar *et al.* in 2017, presented a paper on Breast Cancer prediction with the help of Voting Classifier [3]. The aim of the study was to compare the results of supervised learning classification models and a combination of these algorithms with Voting Classifier technique. The dataset was taken from the Wisconsin University database. This paper helped in understanding the working and usage of Voting Classifier, in learning about a few keywords related to the same and in creating a model having better accuracy than the rest of the algorithms.

Malmasi *et al.* in 2018, presented a paper to test Stacking Classifier on Native Language Identification as it was a new approach and was not previously implemented [8]. This work presented a set of tests using three individual-based models and was tested individually with multiple configurations and algorithms. The paper helped in understanding the working and usage of Stacking Classifier, in learning about a few keywords related to the same and in creating a model having the best accuracy among all the algorithms.

Although several scientists/researchers have proposed various machine learning-based e-signing methods or schemes including classifiers, however, the authors have not tested on various confusion matrix parameters through Voting and Stacking Classifier for e-signing loan applications.

CHAPTER 3: SYSTEM DEVELOPMENT

Computational

For this project, Jupyter Notebook was used to write and execute code in python language as it is well suited to machine learning, deep learning, and data analysis.

Python is a widely used programming language. It is garbage-collected as well as dynamically-typed. Multiple programming paradigms are supported, that includes object oriented, structured, and functional programming.

Due to its comprehensive standard library, it is often described as a "batteries included" language. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It uses dynamic name resolution (late binding), which binds method and variable names during program execution.

JupyterLab is the a web-based interactive development environment for notebooks, code, and data. It has a flexible interface that allows its users to configure and arrange workflows. It is widely used for data science, scientific computing, computational journalism, and machine learning.

Its modular design helps to expand and enrich functionality. It is the original web app for sharing and creating computational documents. It offers a simple and streamlined document-centric experience.

The system used had i7-9750H @2.6GHz and 16GB RAM.

Mathematical

Baye’s Theorem(For Naive Bayes)

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}$$

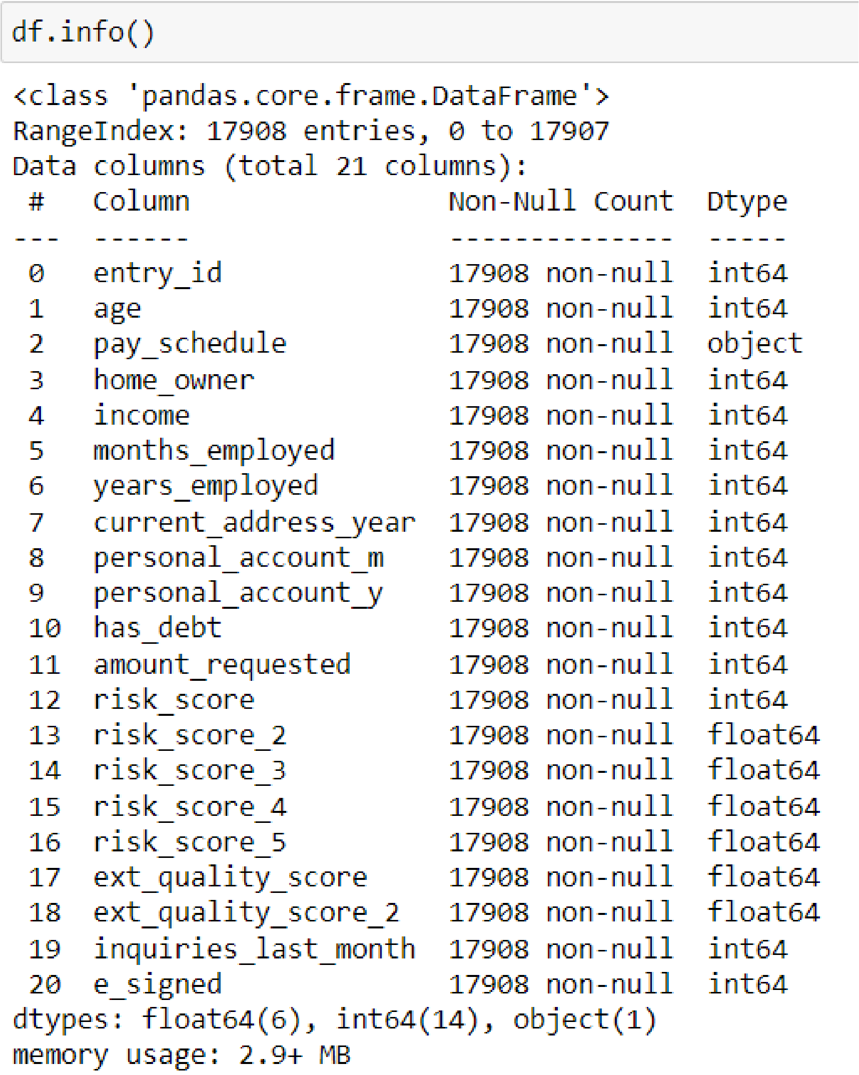
Sigmoid Function(For Logistic Regression)

$$p = \frac{1}{1 + e^{-y}}$$

Experimental

The project was implemented as follows-

- Exploratory Data Analysis is performed on the financial data.

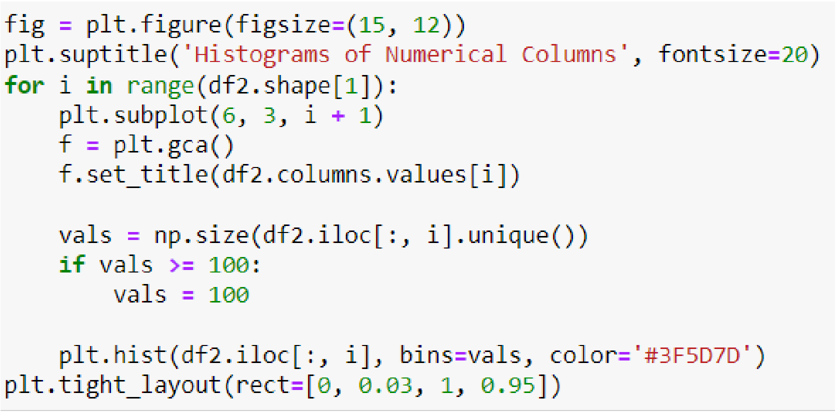


```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17908 entries, 0 to 17907
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   entry_id              17908 non-null  int64  
 1   age                   17908 non-null  int64  
 2   pay_schedule          17908 non-null  object 
 3   home_owner            17908 non-null  int64  
 4   income                17908 non-null  int64  
 5   months_employed       17908 non-null  int64  
 6   years_employed        17908 non-null  int64  
 7   current_address_year  17908 non-null  int64  
 8   personal_account_m    17908 non-null  int64  
 9   personal_account_y    17908 non-null  int64  
 10  has_debt              17908 non-null  int64  
 11  amount_requested      17908 non-null  int64  
 12  risk_score            17908 non-null  int64  
 13  risk_score_2          17908 non-null  float64
 14  risk_score_3          17908 non-null  float64
 15  risk_score_4          17908 non-null  float64
 16  risk_score_5          17908 non-null  float64
 17  ext_quality_score     17908 non-null  float64
 18  ext_quality_score_2   17908 non-null  float64
 19  inquiries_last_month  17908 non-null  int64  
 20  e_signed              17908 non-null  int64  
dtypes: float64(6), int64(14), object(1)
memory usage: 2.9+ MB
```

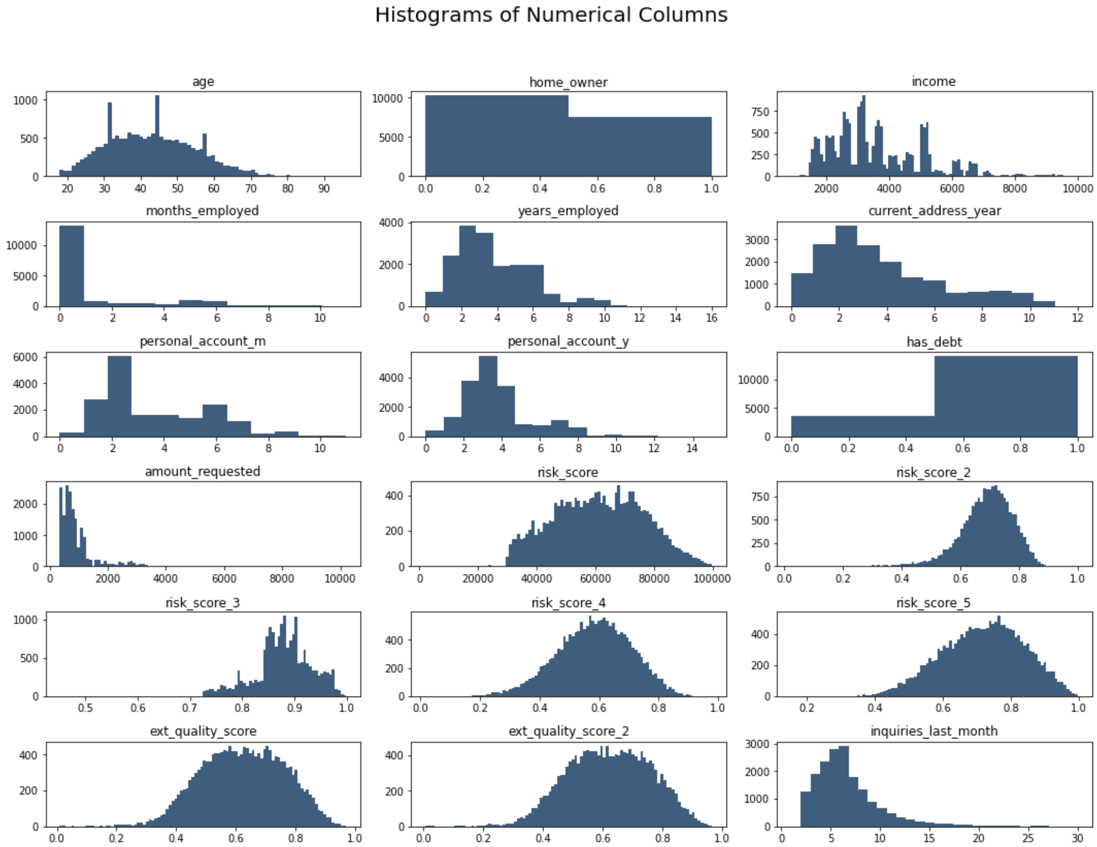
Fig.2 Exploratory Data Analysis

- Histogram for better understanding



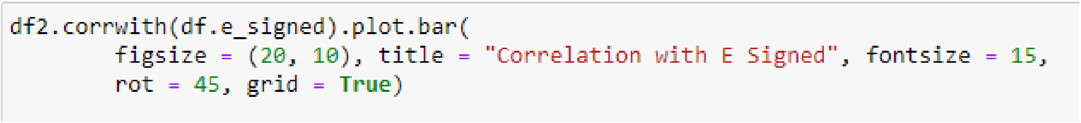
```
fig = plt.figure(figsize=(15, 12))
plt.suptitle('Histograms of Numerical Columns', fontsize=20)
for i in range(df2.shape[1]):
    plt.subplot(6, 3, i + 1)
    f = plt.gca()
    f.set_title(df2.columns.values[i])

    vals = np.size(df2.iloc[:, i].unique())
    if vals >= 100:
        vals = 100

    plt.hist(df2.iloc[:, i], bins=vals, color='#3F5D7D')
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

Fig.3 Code for Histogram



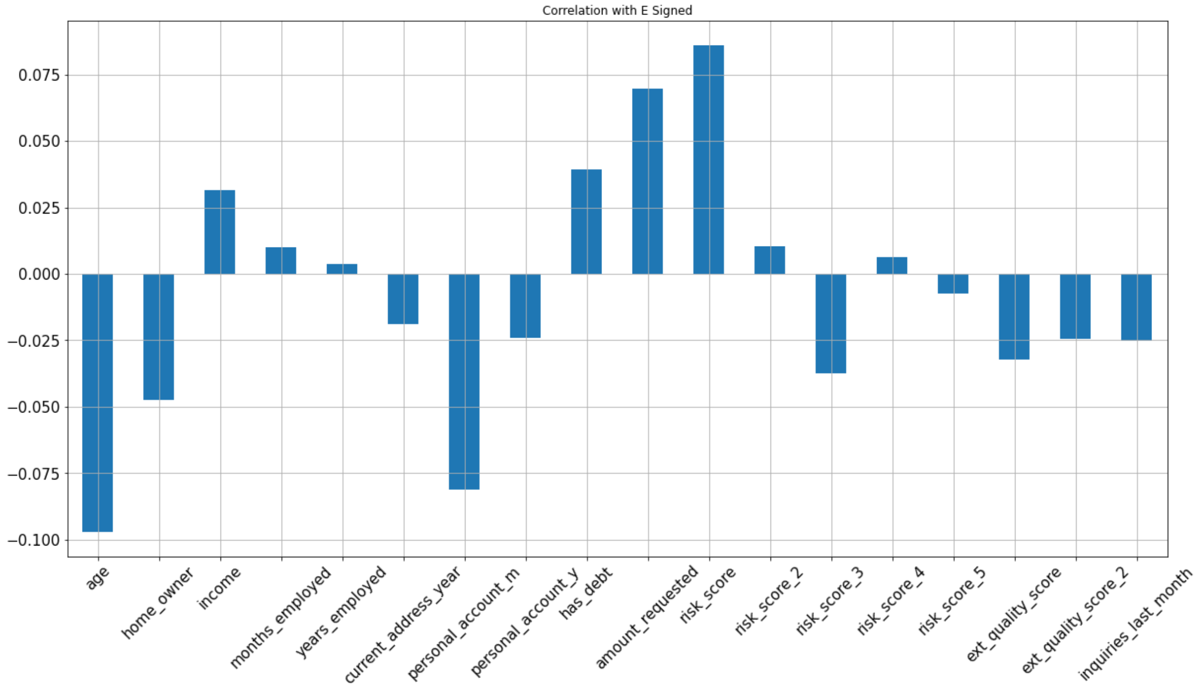
Graph 1: Histogram

- Correlation with response variable



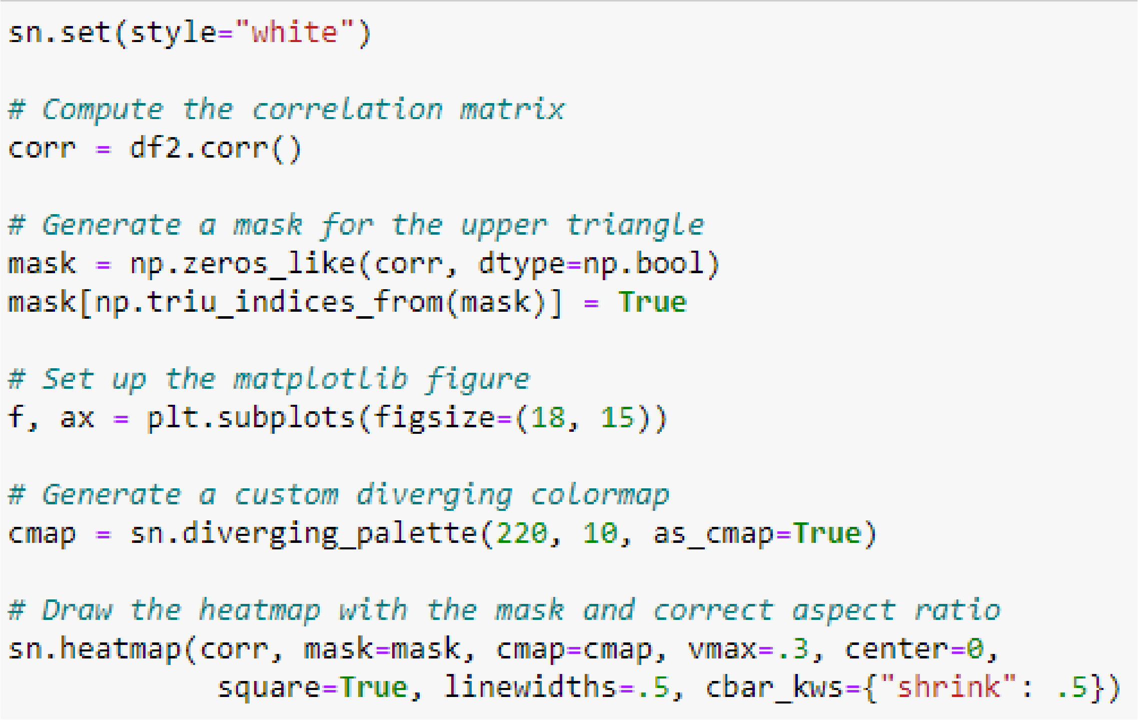
```
df2.corrwith(df.e_signed).plot.bar(
        figsize = (20, 10), title = "Correlation with E Signed", fontsize = 15,
        rot = 45, grid = True)
```

Fig.4 Code for Correlation with response variable



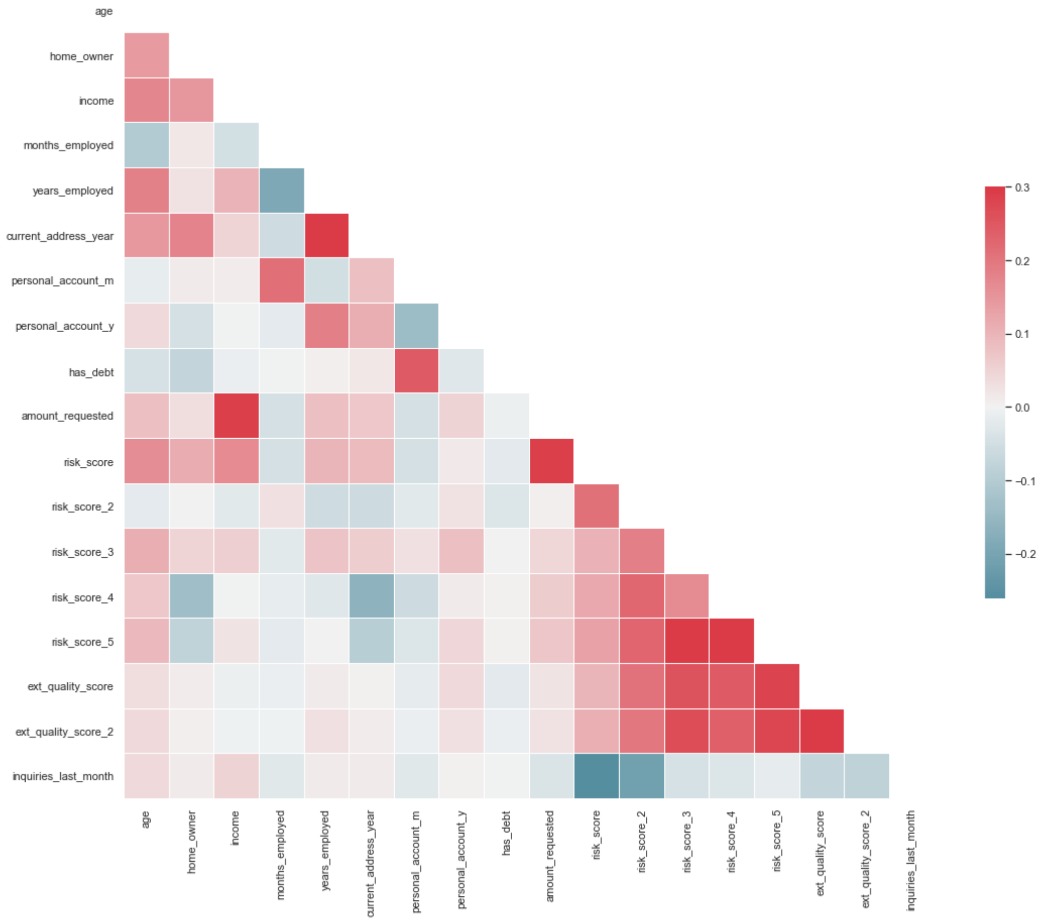
Graph 2: Correlation with response variable

- Correlation Matrix



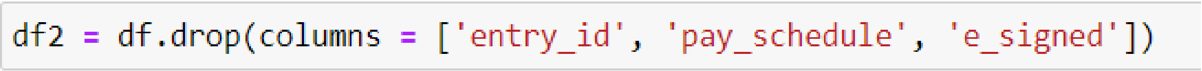
```
sn.set(style="white")

# Compute the correlation matrix
corr = df2.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(18, 15))

# Generate a custom diverging colormap
cmap = sn.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sn.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

Fig.5 Code for Correlation Matrix



Graph 3: Correlation Matrix

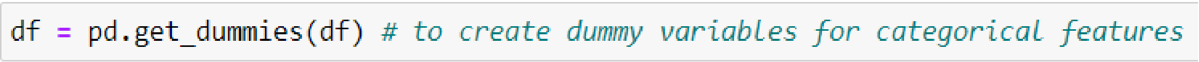
- Unnecessary columns are dropped



```
df2 = df.drop(columns = ['entry_id', 'pay_schedule', 'e_signed'])
```

Fig.6 Code for removing unnecessary columns

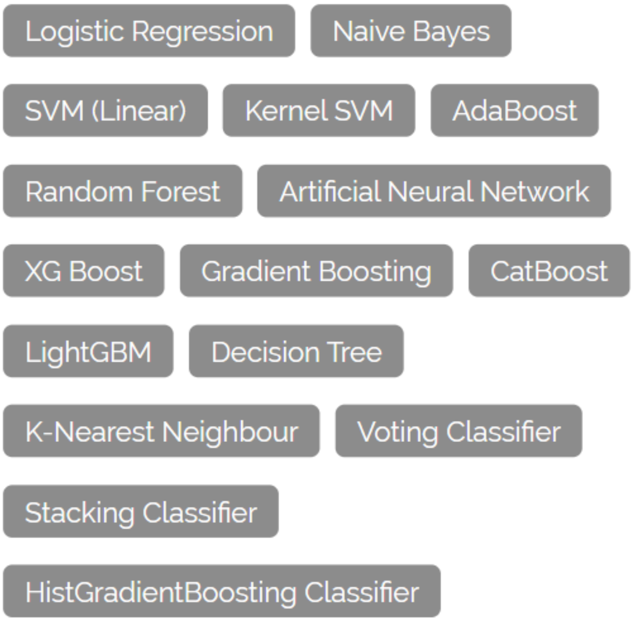
- One Hot Encoding is used for categorical variables



```
df = pd.get_dummies(df) # to create dummy variables for categorical features
```

Fig.7 Code for One Hot Encoding

- Various Machine Learning models are run to find the most efficient solution



Logistic Regression
Naive Bayes
SVM (Linear)
Kernel SVM
AdaBoost
Random Forest
Artificial Neural Network
XG Boost
Gradient Boosting
CatBoost
LightGBM
Decision Tree
K-Nearest Neighbour
Voting Classifier
Stacking Classifier
HistGradientBoosting Classifier

Fig.8 ML algorithms used

- Accuracy, precision, recall, and F1-Score are calculated for each algorithm

- Algorithms Used

Naive Bayes

It is an ML algorithm that is based on probability. It uses the Bayes theorem and is used for classification tasks. Bayes Theorem Naïve Bayes is mostly used for larger datasets with fewer variables. [9]

Logistic Regression

It is usually used when dependent variable or target is categorical. Therefore, it can be said that it is a method used for problems with two class values. It makes use of the Sigmoid function and is used to map real values(predictions) to values between 0 and 1(probabilities). [9]

Artificial Neural Network

It is similar to the human brain which has neurons interconnected to one another. ANN is usually used in places where what has happened in the past is repeated in almost the same way. [9]

K-nearest neighbor

It can be used for both classification as well as regression and stores all the available cases, and the new cases are classified based on similarity. [9]

SVM Linear

SVM is a linear model. It is used for classification and regression problems. SVM can be used for linear as well as non-linear problems. In SVM algorithm, a hyperplane is used to separate the data into classes, and it supports both dense and sparse input. [9]

Kernel SVM

The Kernel is a set of mathematical functions. It takes the data as input, are used by SVM, and transforms it into the required form. [9]

Decision Tree

It can be represented as a flowchart-like tree structure. In it, each internal node is a test on an attribute. Each branch is a result of the test. Each leaf node holds a class label. They are used in operations research, specifically in decision analysis, to help to identify a plan to reach its goal. [9]

Random Forest

Random forest is a supervised ML algorithm. Each tree in a random forest lets out a class prediction and the class having the maximum votes become our prediction. In the end, it merges them together and a stable and accurate prediction is obtained. Both classification and regression problems make use of Random Forests. [9]

XGBoost

It is a decision-tree-based group Machine Learning algorithm. These are used in small to medium structured or tabular data. It pushes the limits of computing power for boosted tree algorithms thus, improving the performance and computational speed. [9]

AdaBoost

It is one of the earliest boosting ML algorithms used for solving problems and it helps to combine many weaker classification models into a single strong classifier and is used for both the classification and regression problems. Its working involves putting more weight on instances that are difficult to classify and less weight on the ones that have already been handled well. [9]

Gradient Boosting

It is also used for regression and classification problems, which can produce a group of weak prediction models, generally decision trees having a fixed size as base learners. [9]

CatBoost

It is an algorithm that is used for gradient boosting and on decision trees. It can work with various data types to solve many problems that businesses face today. It builds one of the most accurate models on whatever dataset it is fed with requiring minimal data prep. [9]

LightGBM

It is a gradient boosting framework and is based on the concept of decision trees. It is used when the requirement is to increase the efficiency of the model and it focuses on the accuracy of results. LightGBM makes use of two novel techniques: Exclusive Feature Bundling and Gradient-based One Side Sampling. [9]

Voting Classifier

The Voting Classifier is a Machine Learning algorithm that predicts the output by taking the results of each classifier based on majority voting. It can be seen in Figure 9.1, the Classification models (C_1, C_2, C_3) and their predictions (P_1, P_2, P_3) which have been calculated using various algorithms, and then through majority voting, final output P_f is predicted. Instead of creating separate models for each algorithm, this algorithm helps in creating a single model which forecasts output on the basis their combined majority of voting for each output class. [9]

Voting Classifier are of two types -

Hard Voting: supports majority voting

Soft Voting: supports average voting

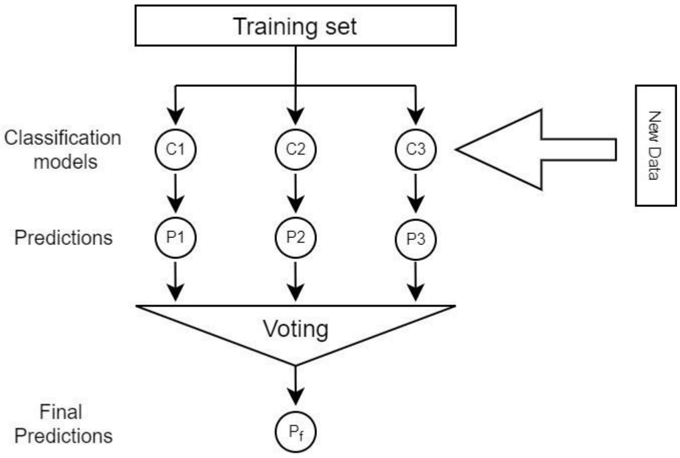


Fig.9 Voting Classifier

Stacking Classifier

Stacking is an ensemble learning algorithm that uses meta classifier to combine multiple classification models. The models are trained separately on the basis of the complete training set and then the meta-classifier fits on the basis of the output of the first level classifier. It can be seen in Figure 9.2, the Classification models (C_1, C_2, C_3) and their predictions (P_1, P_2, P_3) have been calculated using various algorithms, and then through the meta classifier, the final output P_f is predicted. Stacking uses the strength of each individual estimator by using their output as input of a final estimator. In the model presented in this paper for Voting Classifier, three algorithms have been used for voting namely, XGBoost, CatBoost, and LightGBM having the weights 1, 1, and 1 respectively in hard voting and 1, 3, and 5 respectively in soft voting. The reason for using these three algorithms was that after testing for various combinations of algorithms the best accuracy was seen in the above-mentioned ones. For Stacking Classifier, the following three algorithms for the first level classification have been used, these are - XGBoost, CatBoost, and LightGBM, and the final estimator or the meta classifier for the model was Logistic Regression.

To improve the accuracy, various splits were tried. The splits tried were 80-20, 70-30, and 60-40. Correlation Matrix has been used to visualize the relationship between two variables and Confusion Matrix to evaluate the model and calculate True Positive and True Negative.

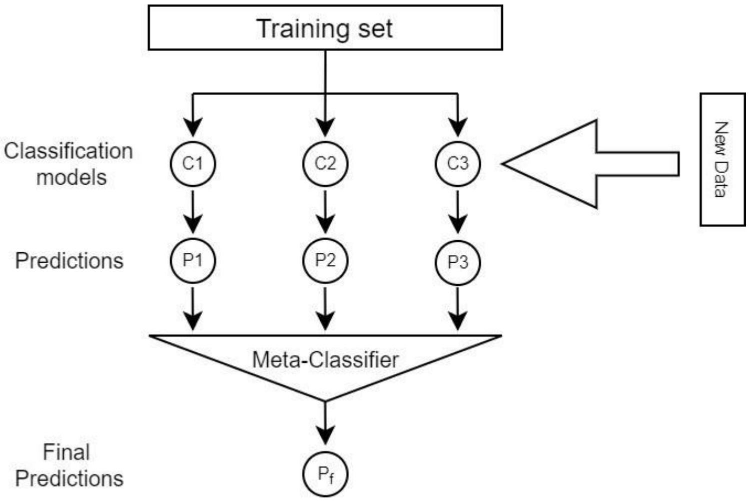


Fig.10 Stacking Classifier

CHAPTER 4: PERFORMANCE ANALYSIS

Dataset

In this project, the dataset used was downloaded from Kaggle. The dataset contains 17908 entries and parameters of financial history

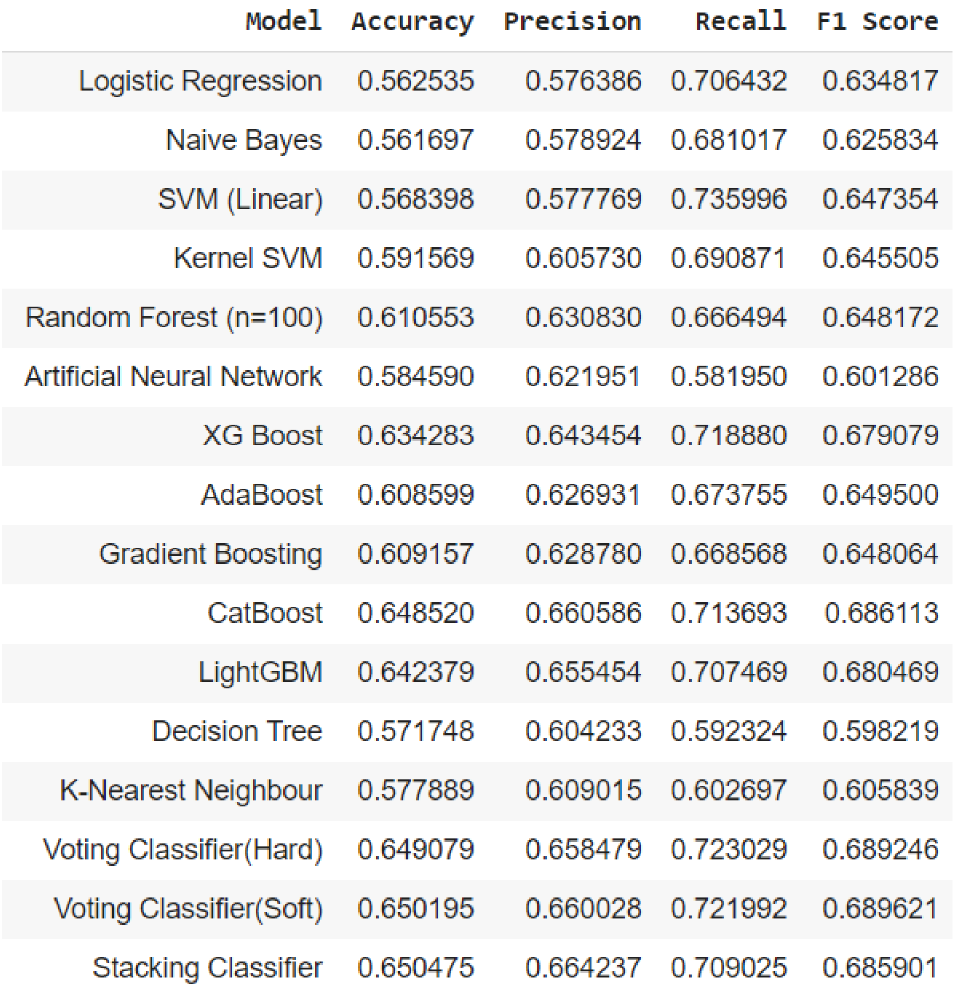
S. no	Features	Information
1	entry_id	Unique identity of the applicant
2	age	age of the applicant
3	pay_schedule	how often applicant gets paid
4	home_owner	whether the applicant has a home or not
5	income	monthly income of the applicant
6	months_employed	no. of months the applicant has been employed
7	years_employed	no. of years the applicant has been employed
8	current_address_year	how many years has the applicant stayed at his current address

9	personal_account_m	no. of months the applicant has his personal account
10	personal_account_y	no. of years does the applicant has his personal account
11	has_debt	whether the applicant has debt or not
12	amount_requested	amount the applicant has applied for
13	risk_score	given to applicant by the finance/engineering teams
14	risk_score_2	given to applicant by the finance/engineering teams
15	risk_score_3	given to applicant by the finance/engineering teams
16	risk_score_4	given to applicant by the finance/engineering teams
17	risk_score_5	given to applicant by the finance/engineering teams
18	ext_quality_score	given to applicant by the finance/engineering teams
19	ext_quality_score_2	given to applicant by the finance/engineering teams
20	inquiries_last_month	no. of inquiries made by applicant in last month
21	e_signed	e-signing completed or not

Table 1: Dataset

Implementation

- The data was split in 80-20 for training and testing respectively
- The highest accuracies obtained were 65.04% and 65.01% for Stacking Classifier and Voting Classifier(Soft) respectively.



Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.562535	0.576386	0.706432	0.634817
Naive Bayes	0.561697	0.578924	0.681017	0.625834
SVM (Linear)	0.568398	0.577769	0.735996	0.647354
Kernel SVM	0.591569	0.605730	0.690871	0.645505
Random Forest (n=100)	0.610553	0.630830	0.666494	0.648172
Artificial Neural Network	0.584590	0.621951	0.581950	0.601286
XG Boost	0.634283	0.643454	0.718880	0.679079
AdaBoost	0.608599	0.626931	0.673755	0.649500
Gradient Boosting	0.609157	0.628780	0.668568	0.648064
CatBoost	0.648520	0.660586	0.713693	0.686113
LightGBM	0.642379	0.655454	0.707469	0.680469
Decision Tree	0.571748	0.604233	0.592324	0.598219
K-Nearest Neighbour	0.577889	0.609015	0.602697	0.605839
Voting Classifier(Hard)	0.649079	0.658479	0.723029	0.689246
Voting Classifier(Soft)	0.650195	0.660028	0.721992	0.689621
Stacking Classifier	0.650475	0.664237	0.709025	0.685901

Fig.11 Results

Comparison

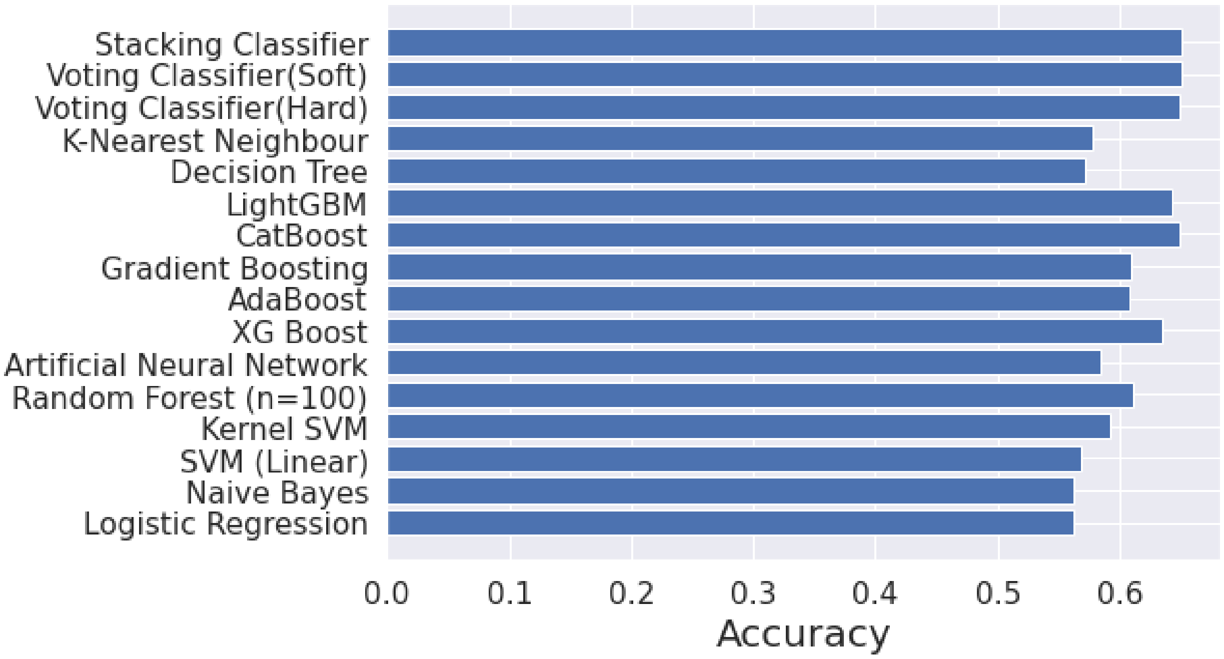
The best technique that had been used for implementing this was Random Forest which has 64% accuracy [1].

CHAPTER 5: CONCLUSION

5.1 Conclusions

The conclusion that can be drawn from the project is-

- Stacking Classifier and Voting Classifier(Soft) gave the most accurate results.



Graph 4: Accuracy Bar Graph

5.2 Future Scope

The future scope of the project is as follows-

- The accuracy of the model can be increased by increasing the number of epochs in ANN model.
- This accuracy of ML models can be further improved by better data pre-processing and hyperparameter tuning.

5.3 Applications Contributions

The main implication of this project is automating the job of people who spend hours and go through tons of loan applications and financial history of applicants to see if the loan should be passed or not.

REFERENCES

[1] Patil, S. B., Chougale, A. S., Chougule, R. P., Havaldar, A. A., & Belagali, S. S. (2019). Speculating the Likeliness of e-Validating a Loan based on Financial Transactions. *Journal of Advances in Computational Intelligence Theory*, *1*(2, 3).

[2] Goyal, A., & Kaur, R. (2016). Accuracy Prediction for Loan Risk Using Machine Learning Models. *Int. J. Comput. Sci. Trends Technol*, *4*(1), 52-57.

[3] Kumar, U. K., Nikhil, M. S., & Sumangali, K. (2017, August). Prediction of breast cancer using voting classifier technique. In *2017 IEEE international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM)* (pp. 108-114). IEEE.

[4] Chen, Z., Jiang, F., Cheng, Y., Gu, X., Liu, W., & Peng, J. (2018, January). XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. In *2018 IEEE international conference on big data and smart computing (bigcomp)* (pp. 251-256). IEEE.

[5] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). CatBoost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*.

[6] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*, 3146-3154.

[7] Al Daoud, E. (2019). Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset. *International Journal of Computer and Information Engineering*, *13*(1), 6-10.

[8] Malmasi, S., & Dras, M. (2018). Native language identification with classifier stacking and ensembles. *Computational Linguistics*, *44*(3), 403-446.

[9] Vats, A., Singh, R., Rathee, G., & Saini, H. (2021). Predictions of Loan E-Signing Based on Financial Status of Applicants Using Machine Learning. In Assistive Technology Intervention in Healthcare (pp. 121-135). CRC Press.