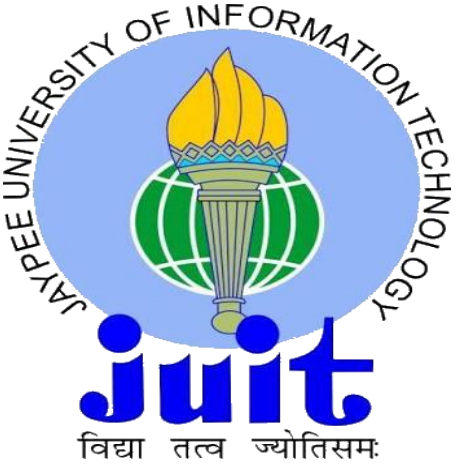


DATA SCIENCE AND VISUALISATION PROJECT REPORT

Submitted To-

Dr. Geetanjali

Submitted By-

Rashi Singh

181228

CS 51

Apoorv Vats

181308

CS 54

INTRODUCTION

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

This project is about **predicting the likelihood of e-signing a loan based on financial history**.

ABOUT THE DATASET

The dataset contains **17908** entries and parameters of financial history like 'entry_id', 'age', 'pay_schedule', 'home_owner', 'income', 'months_employed', 'years_employed', 'current_address_year', 'personal_account_m', 'personal_account_y', 'has_debt', 'amount_requested', 'risk_score', 'risk_score_2', 'risk_score_3', 'risk_score_4', 'risk_score_5', 'ext_quality_score', 'ext_quality_score_2', 'inquiries_last_month', 'e_signed'.

Link to the dataset - (original source: Kaggle)
https://drive.google.com/file/d/1k6oygMV8e12UoKU8yLkGit46QRTOFDE8/view?usp=sharing

PROBLEM STATEMENT

To visualise the financial data using various techniques and predict the chances of people e-signing a loan based on financial history.

The procedure followed is-

EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical

representations.

1. `df.head()`
2. `df.columns`
3. `df.describe()`

```
df.head()
```

	entry_id	age	pay_schedule	home_owner	income	months_employed	years_employed	current_address_year	personal_account_m	personal_account_
0	7629673	40	bi-weekly	1	3135	0	3	3	6	
1	3560428	61	weekly	0	3180	0	6	3	2	
2	6934997	23	weekly	0	1540	6	0	0	7	
3	5682812	40	bi-weekly	0	5230	0	6	1	2	
4	5335819	33	semi-monthly	0	3590	0	5	2	2	

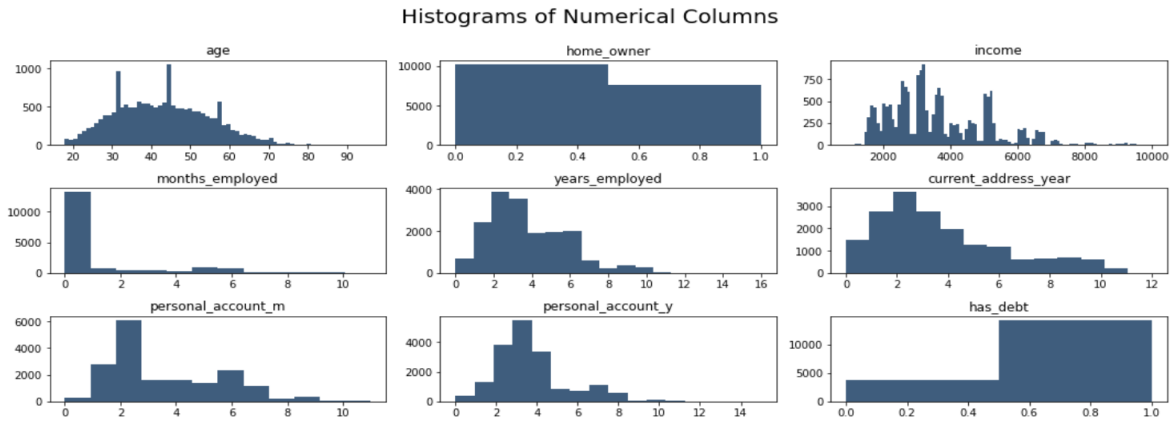
PRE - PROCESSING

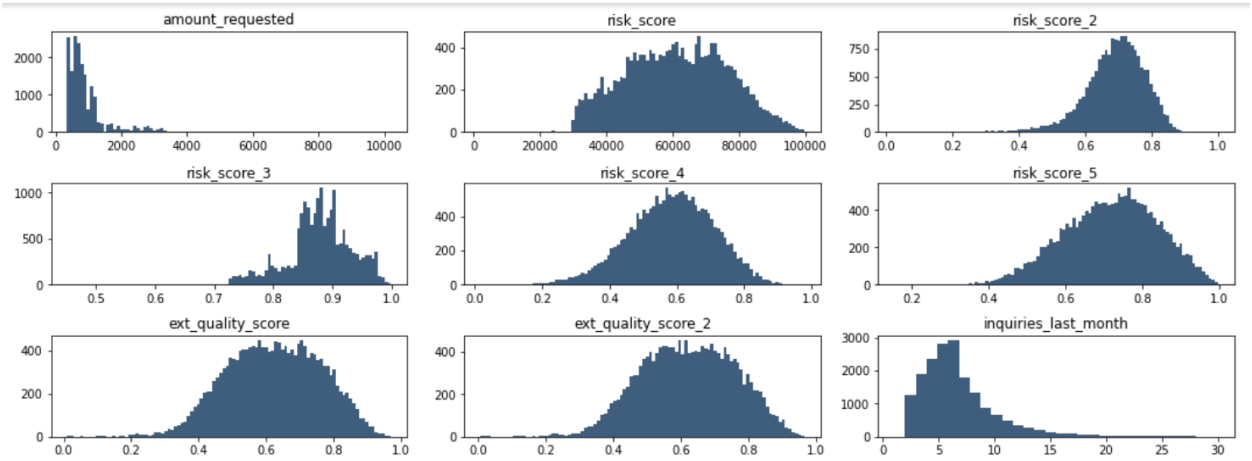
Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

1. Check for NaN values: `df.isna().any()`
2. Removing columns that are not needed: `df2 = df.drop(columns = ['entry_id', 'pay_schedule', 'e_signed'])`

DATA VISUALISATION

1. **Histograms** : numerical columns



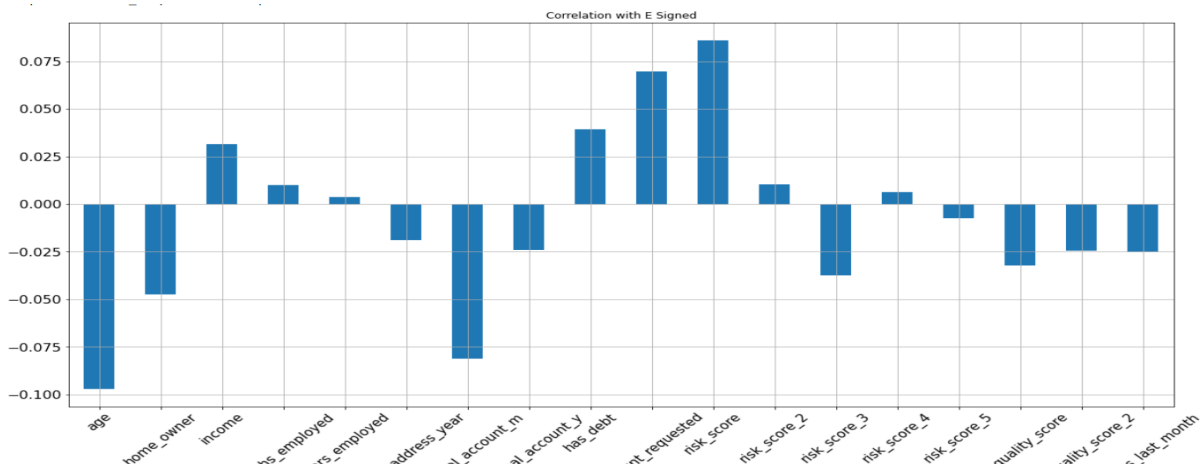


Conclusions drawn from the above plots-

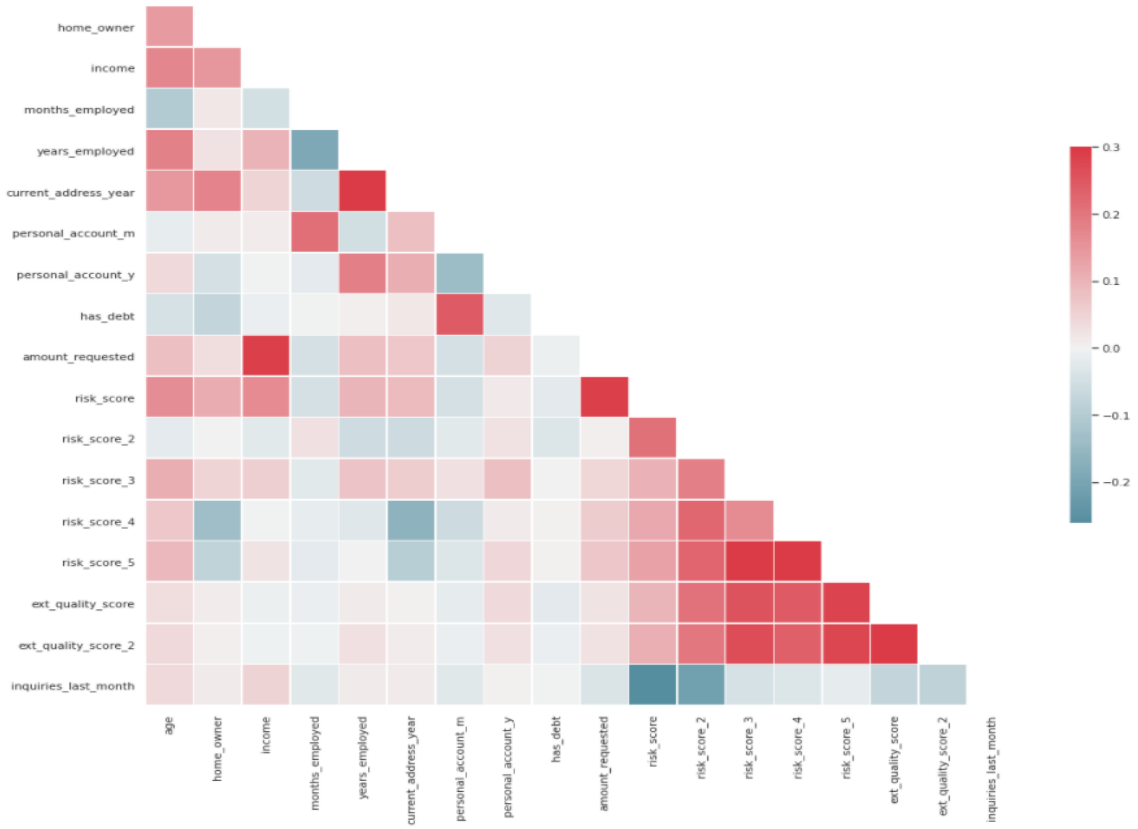
Maximum number of people have-

- Age: 44
- Income: between 2000 and 4000 dollars
- Months employed: 0
- Years employed: 2
- Personal account: 2-3

2. Correlation with Response Variable



3. **Correlation Matrix:** (using heatmap)



FEATURE ENGINEERING

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques.

It has mainly two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.

1. **One Hot encoding:** It allows the representation of categorical data to be more expressive. Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.

2. **Splitting Data:** The dataset was divided into *training dataset* to train the model and *testing data* to test the efficiency of the machine learning models.
3. **Feature Scaling**: Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

MODEL BUILDING

We have tried various algorithms on our model to test for the best accuracy we can get.

The algorithms are as follows-

Naive Bayes Classifier

It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

It is usually used for large datasets with less variables.

The accuracy was: 0.561

Logistic Regression

It is the go-to method for binary classification problems (problems with two class values).

The accuracy was: 0.562

SVM (Linear)

This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.

The accuracy was: 0.568

SVM (Kernel)

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of the kernel is to take data as input and transform it into the required form.

The accuracy was: 0.591

Linear SVM is a parametric model, an RBF kernel SVM isn't, and the complexity of the latter grows with the size of the training set.

Random Forests

It is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm.

The accuracy was: 0.621

This model gave us the best accuracy score.

K-FOLD CROSS VALIDATION

A K-Fold CV is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point.

```
Random Forest Classifier Accuracy: 0.63 (+/- 0.03)
```

APPLYING GRID SEARCH

It is an exhaustive search over a specified parameter value for an estimator. The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid.

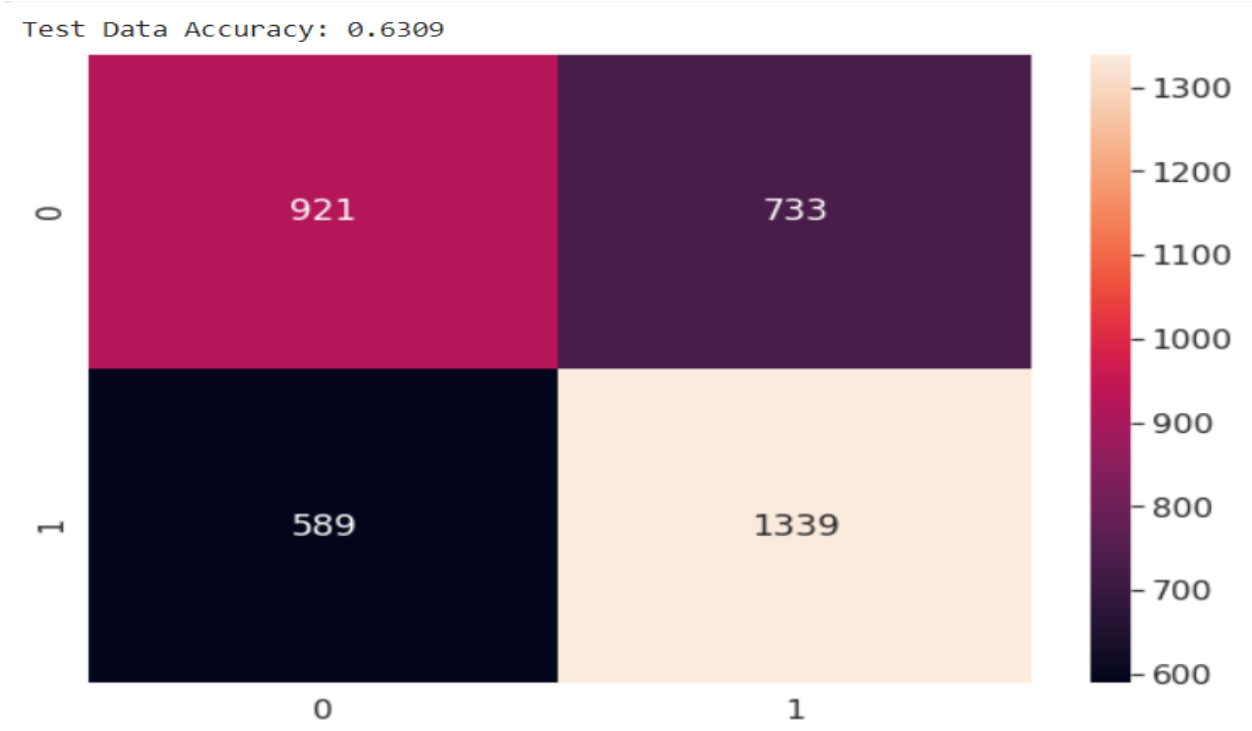
```
results
```

	Model	Accuracy	Precision	Recall	F1 Score
0	Naive Bayes	0.561697	0.578924	0.681017	0.625834
1	Linear Regression (Lasso)	0.562535	0.576386	0.706432	0.634817
2	SVM (Linear)	0.568398	0.577769	0.735996	0.647354
3	Kernel SVM	0.591569	0.605730	0.690871	0.645505
4	Random Forest (n=100)	0.621720	0.640098	0.678942	0.658948
5	Random Forest (n=100, GSx2 + Entropy)	0.625070	0.640828	0.690353	0.664669
6	Random Forest (n=100, GSx2 + Gini)	0.630932	0.646236	0.694502	0.669500

From the above table we can see that the accuracy has increased.

CONFUSION MATRIX

A **confusion matrix** is a tabular summary of the number of correct and incorrect predictions made by a classifier. It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score.



CHALLENGES

1. **Hyper-parameter Tuning**: Since it is very difficult to find hyper-parameters manually, grid search method was used for hyper-parameter tuning.
2. **Error:** Since, there was a possibility of error, we used three techniques - MAE, MSE, RMSE to detect them.
3. **Total Months Employed:** Since the time of employment was divided into months employed and years employed it did not give any relevant results, therefore total months employed column was calculated and used so it could give meaningful results .

RESULTS

The table below shows the predictions that we have made about the e-signing of loans using the model. Also for comparison there is another column e_signed which tells whether the loan was signed or not in reality.

	entry_id	e_signed	predictions
8	6493191	1.0	0
9	8908605	1.0	0
12	6889184	1.0	1
16	9375601	0.0	1
18	8515555	1.0	1
...	...	...	...
17881	5028251	1.0	1
17888	8958068	0.0	0
17890	3605941	0.0	1
17901	1807355	0.0	1
17907	1498559	1.0	1

3582 rows × 3 columns

REFERENCES

1. Data Visualization
2. Exploratory Data Analysis
3. Feature Scaling