DATA SCIENCE PROJECT:-

Credit Risk Analysis of P2P Lending Platform

(Technocolabs Software)

TEAM –

1.Abhishek Sharma

2.Marrapu Santosh

3.Eaga Jaswanth Krishna

4.Sandip Shivdas Dighore

5.Kalluri Yaswanth Kumar

6.Vijay K

7.Suvinraj.M

8.Orooj Fatma

INTRODUCTION

Bondora is a peer-to-peer lending platform from Estonia.Bondora serves as an online community from investing in personal debt starting at €1, and enables borrowers to receive funding directly from invstorss.The investment are dominated in EUR and are available from retail investors. The company behind the platform is Bondora Capital OU which has founded in 2015 with its headquarters in Tallin,Estonia.

At Bondora, you can invest in projects denominated in EUR starting from €1. You can also set up an auto-invest feature and buy and sell loans on a secondary market.

Bondora separates investors' funds from the platform company assets and has a wind-down plan in place to ensure minimal damage in case the platform goes out of business. It discloses statistics about its funded loans on a dedicated public page and shares its monthly funding volumes with P2PMarketData.

Problem Statement

To determine an individual’s loan eligibility and calculate the appropriate loan amount based on specific parameters, enabling company to make informed lending decisions while ensuring fair practices and maximizing profitability.

ABSTRACT

This dataset contains information on 134,530 loan applications, with each application having 114 different features. The dataset includes both numerical and categorical features that provide various details about the borrowers, loan amounts, repayment schedules, borrower demographics, credit scores, employment details, and more.

Some of the important features in the dataset are:

ActiveLateCategory: Categorizes loans in Principal Debt based on the number of days in debt.

ActiveLateLastPaymentCategory: Categorizes loans based on the number of days since the last payment and determines if it is overdue.

ActiveScheduleFirstPaymentReached: Indicates whether the borrower has reached the first payment date according to the active schedule.

Age: The age of the borrower when they applied for the loan.

Amount: The amount the borrower received on the primary market or the principal balance from the secondary market.

AmountOfPreviousLoansBeforeLoan: The value of previous loans taken by the borrower.

AppliedAmount: The original amount applied for by the borrower.

AuctionBidNumber: Unique bid number associated with an auction.

AuctionId: A unique number assigned to each auction.

AuctionName: Name of the auction, often defined by the purpose of the loan.

AuctionNumber: Unique number associated with an auction.

BidPrincipal: The amount bid on the primary market or the purchase price on the secondary market.

BidsApi: The amount of investment offers made via API.

BidsManual: The amount of investment offers made manually.

BidsPortfolioManager: The amount of investment offers made by Portfolio Managers.

BoughtFromResale_Date: The date when the investment was purchased from the secondary market.

City: The city of the borrower.

ContractEndDate: The date when the loan contract ended.

Country: The residency of the borrower.

County: The county of the borrower.

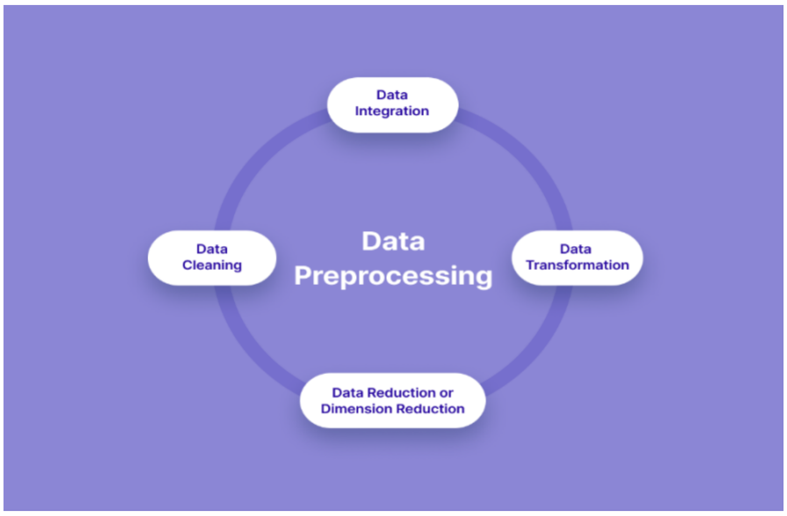
To extract important features from this dataset, various techniques such as feature selection, feature importance analysis, correlation analysis, or machine learning algorithms. These methods will help identify the features that have a significant impact on the target variable or contribute the most to the overall predictive power of the model.

By analyzing and understanding these features, we have gain insights into the loan applications, borrower characteristics, and factors that influence the loan repayment process.

DATA PREPROCESSING:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we used data preprocessing task.



Some of the key steps we have followed in data pre-processing :

1)Understanding the data

2)Missing value treatment

3)Handling outliers

4)Creation of target variable

5)Transforming some variables into appropriate format

Missing value treatment:

In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data.

In this process of missing value treatment, any features having more than 40% null values are eliminated. Subsequently, the remaining columns are filled with their respective central tendency values, taking into account the type of variable they represent. After dropping necessary columns, I am left with around 40 features out of 113.

Handling outliers

One of the most important steps as part of data preprocessing is detecting and treating the outliers as they can negatively affect the statistical analysis and the training process of a machine learning algorithm resulting in lower accuracy.

Outliers refer to exceptionally large or small values in numerical features, and they can significantly impact data analysis. Therefore, in the case of numerical columns, these outliers are removed as part of the data processing step.

Creation of target variable:

As per the problem statement it is a classification task. So we have to create a target variable for binary classification. So We chosen status as target variable and transformed to binary form with the help of default date feature.

Transforming some variables into appropriate format:

Many features in data are not in appropriate format such as employ status, occupation, etc so we have converted the features into categorical columns. Later we have applied encoding techniques for model preparation.

EDA:

Exploratory Data Analysis (EDA) is an approach that is used to analyze the data and discover trends, patterns, or check assumptions in data with the help of statistical summaries and graphical representations.

Depending on the number of columns we are analyzing we can divide EDA into two types:

1)Univariate analysis

2)Bivariate analysis

Firstly We rechecked the missing values to make sure for effective eda then jumped into univariate analysis

Univariate analysis:

- Univariate analysis is analyzing a single feature at a time.
- Presented individual visualizations, such as histograms, bar charts, and box plots, for each relevant feature in the dataset.
- Described the distribution patterns and key insights obtained from the univariate analysis.

Correlation is an indication about the changes between two variables. So We plotted correlation matrix to show which variable is having a high or low correlation in respect to another variable.

Bivariate analysis:

- The bivariate analysis aims to determine if there is a statistical link between the two variables and, if so, how strong and in which direction that link is.
- Present visualizations that explore the relationship between the target variable and other features.
- Discuss any significant trends, correlations, or patterns observed during the bivariate analysis.

So EDA helped us to emphasize any important insights or relationships between variables that could be relevant for further analysis or modeling.

Bondara loan Eligibility Prediction Feature Engineering

Definition:-

Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model usingmachine learning or statistical modeling.

. Feature engineering in ML contains mainly four processes:

1.Feature

Creation,

2.Transformations,

3. Feature Extraction,

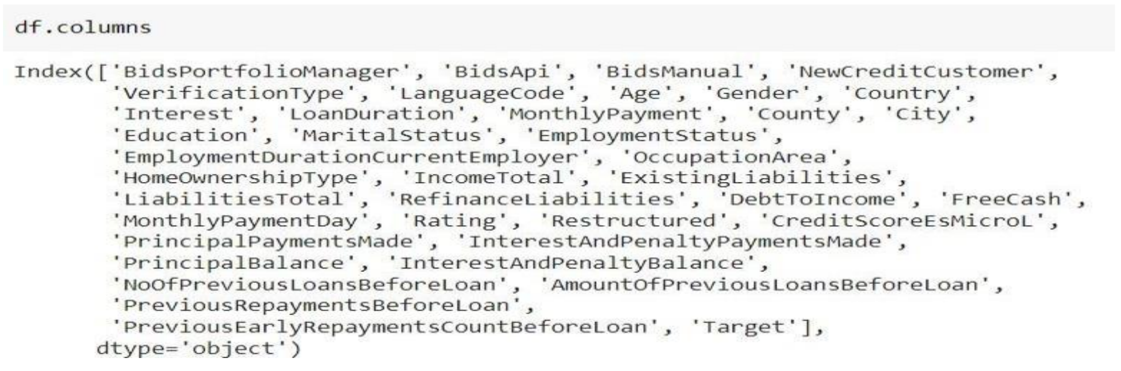
4.Feature Selection.

Feature Creation:

Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ration, and these new features have great flexibility.

After preprocess and Exporatory data analysis (EDA) task we got data contain 43 columns and out of those one of varibles as Target varible .

Column require for feature engineering are:-



```
df.columns
```

```
Index(['BidsPortfolioManager', 'BidsApi', 'BidsManual', 'NewCreditCustomer',
       'VerificationType', 'LanguageCode', 'Age', 'Gender', 'Country',
       'Interest', 'LoanDuration', 'MonthlyPayment', 'County', 'City',
       'Education', 'MaritalStatus', 'EmploymentStatus',
       'EmploymentDurationCurrentEmployer', 'OccupationArea',
       'HomeOwnershipType', 'IncomeTotal', 'ExistingLiabilities',
       'LiabilitiesTotal', 'RefinanceLiabilities', 'DebtToIncome', 'FreeCash',
       'MonthlyPaymentDay', 'Rating', 'Restructured', 'CreditScoreEsMicroL',
       'PrincipalPaymentsMade', 'InterestAndPenaltyPaymentsMade',
       'PrincipalBalance', 'InterestAndPenaltyBalance',
       'NoOfPreviousLoansBeforeLoan', 'AmountOfPreviousLoansBeforeLoan',
       'PreviousRepaymentsBeforeLoan',
       'PreviousEarlyRepaymentsCountBeforeLoan', 'Target'],
      dtype='object')
```

Transformations:

The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model. For example, it ensures that the model is flexible to take input of the variety of data; it ensures that all the variables are on the same scale, making the model easier to understand. It improves the model's accuracy and ensures that all the features are within the acceptable range to avoid any computational error.

Three are two types :

1.Handling missing values

Checking null values with isnull() functions it shows some columns contain has null values so also observe that there are two types of categories one is category another is numerical category so we have to handle separately,now I had observe the skew method of data distrubtion so thatwe can use the methods based on that.

Code:

```
df.skew(axis = 0, skipna = True)
```

Data is skew then for numerical data I had replaced with mean.

Data is not skew then for numerical data I had replaced with meadian.

Data is normal then I replace mode function values with categorical

values.

First of all I have to handle numerical missing values so I had to take those features are and replace the missing values

```
-Monthly Payment :
-Previous Repayments Before Loan
-Debt To Income
```

Secondly I had taken the categorical features and replace it with mode function.

1. Rating, 2.Free Cash, 3.Credit Score Microl, 4.County ,5.Gender ,6. Education, 7.Verification Type, 8.Employee Status ,9.Occuption Status,10. City
2. Now I again verify null values it shows zero because we already cleared those.

3. converting categorical to numeric:

After converting missing values we have to convert the numerical because we are mainly using numerical values in machine learning models . so for these I used label encoder and replace functions.

Then I check the datatypes of all features it shows int or float we completed.

Feature Selection:

While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant. If we input the dataset with all these redundant and irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the model. Hence it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less important features, which is done with the help of feature selection in machine learning. "Feature selection is a way of selecting the subset of the most relevant features from the original featuresset by removing the redundant, irrelevant, or noisy features.".

In this task first of I had calculate the mi scores (mutual information scores which is asme as information scores) by this I can observe the features importance .

```
from sklearn.feature_selection import mutual_info_classif as MIC
X = df.copy()

y = X.pop("Target")
mi_score = MIC(X,y)
```

MULTICOLLINEARITY:

Also I use correlation and heat map to understand the relation ship between varibles and removethe multi collinearity varibles.

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(22,22))
```

Next I use VIF method (Variance influence vector) which means it calculates the scores if those scores below threshold values(<5) which is acceptable otherwise we have to remove the highest VIF score varible again same process will be done until the all VIF scores of varibles is below 5.

```
# load statmodels functions
from statsmodels.stats.outliers_influence import
variance_inflation_factor
from statsmodels.tools.tools import add_constant

# compute the vif for all given features
def compute_vif(considered_features):

    X = df[considered_features]
    # the calculation of variance inflation requires a constant
    X['intercept'] = 1

    # create dataframe to store vif values
    vif = pd.DataFrame()
    vif["Variable"] = X.columns
```

Here I removed 3 features which have highest vif factor:

```
Amount
Use Of Loan
Applied Amount
```

After that I taken all features and applied **SelectKBest method** to get the most important features of this data. So I got most important 8 features

```
from sklearn.feature_selection import SelectKBest, mutual_info_classif
from sklearn.model_selection import train_test_split
k = 8
selector = SelectKBest(score_func=mutual_info_classif, k=k)
# Apply feature selection to the training data
X_train_selected = selector.fit_transform(X, y)
# Get the selected feature indices
selected_feature_indices = selector.get_support(indices=True)
# Get the selected feature names
```

```
print(selected_feature_names)
```

They are:

Applied Amount, Amount, Interest, Monthly Payment, City,Principal Payments

Made, Principal Balance

Interest And Penalty Balance

After that I verify the features with data I also observed that three two more features which isimportant so I include those also now features selection features is:

```
1.BidsPortfolioManager
2.Interest
3.MonthlyPayment
4.Rating
5.PrincipalPaymentsMade
 6.NoOfPreviousLoansBeforeLoan
7.PrincipalBalance
8.InterestAndPenaltyBalance
9.AmountOfPreviousLoansBeforeLoan
10.PreviousRepaymentsBeforeLoan
```

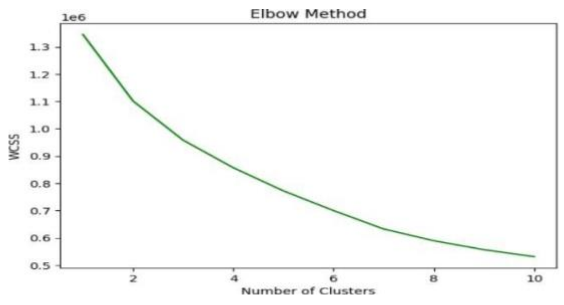
Feature Extraction:

Feature extraction is an automated feature engineering process that generates new variables byextracting them from the raw data. The main aim of this step is to reduce the volume of data sothat it can be easily used and managed for data modelling. Feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA).

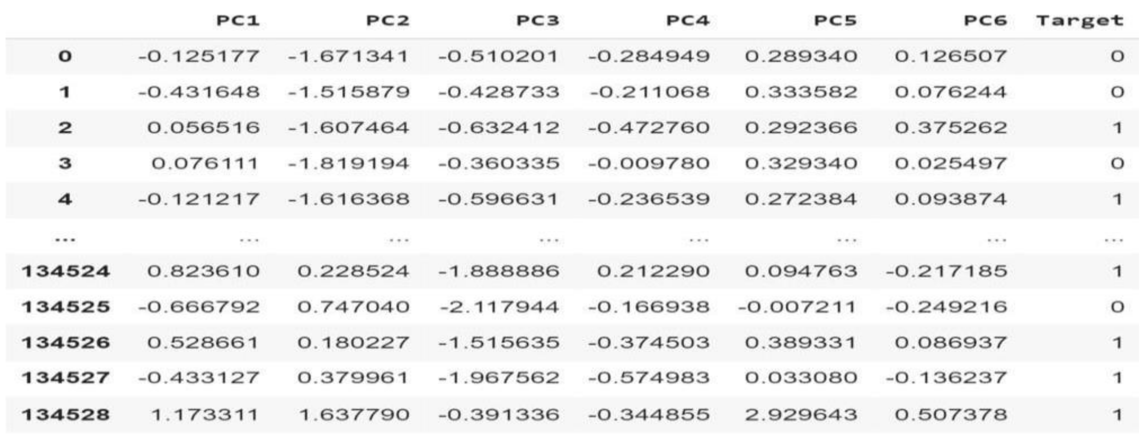
I had taken those 10 features for feature extraction and taken the target target varible (loan status default /not default).

I applied standardscaler to scale the data and also use fit and transform of this data

Elnow method:



Then I use elbow method to know the desired number of pca components so I observed that the graph is bend at 6 so I taken 6 PCA components. Then I perform PCA and feature extraction using the libraries I imported so it gives pca components and I also calculated the variance of those components it gives around 90 percent so I taken those pca 's.



	PC1	PC2	PC3	PC4	PC5	PC6	Target
0	-0.125177	-1.671341	-0.510201	-0.284949	0.289340	0.126507	0
1	-0.431648	-1.515879	-0.428733	-0.211068	0.333582	0.076244	0
2	0.056516	-1.607464	-0.632412	-0.472760	0.292366	0.375262	1
3	0.076111	-1.819194	-0.360335	-0.009780	0.329340	0.025497	0
4	-0.121217	-1.616368	-0.596631	-0.236539	0.272384	0.093874	1
...	...	...	...	...	...	...	...
134524	0.823610	0.228524	-1.888886	0.212290	0.094763	-0.217185	1
134525	-0.666792	0.747040	-2.117944	-0.166938	-0.007211	-0.249216	0
134526	0.528661	0.180227	-1.515635	-0.374503	0.389331	0.086937	1
134527	-0.433127	0.379961	-1.967562	-0.574983	0.033080	-0.136237	1
134528	1.173311	1.637790	-0.391336	-0.344855	2.929643	0.507378	1

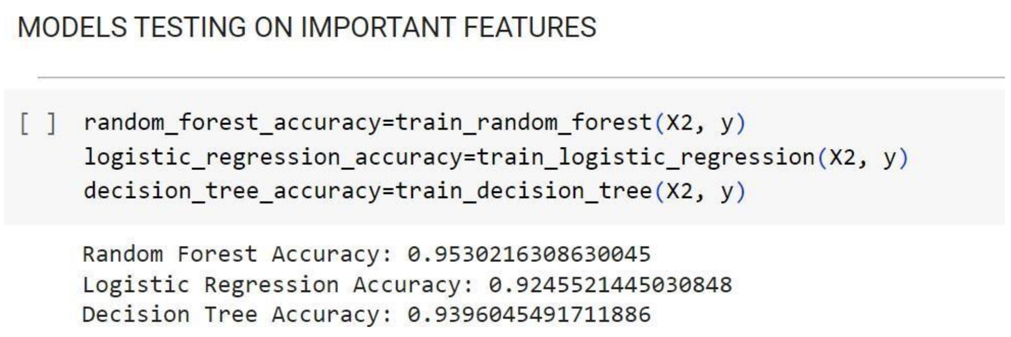
Benefits of feature engineering in machine learning:

- It helps in avoiding the curse of dimensionality.
- It helps in the simplification of the model so that the researchers can easily interpret it.
- It reduces the training time.
- It reduces overfitting hence enhancing the generalization.

Model selection and feature comparison

After selecting the features I calculated the accuries of all models with feature selectionand feature extraction output to know the better features .

Feature selected output accuracies-

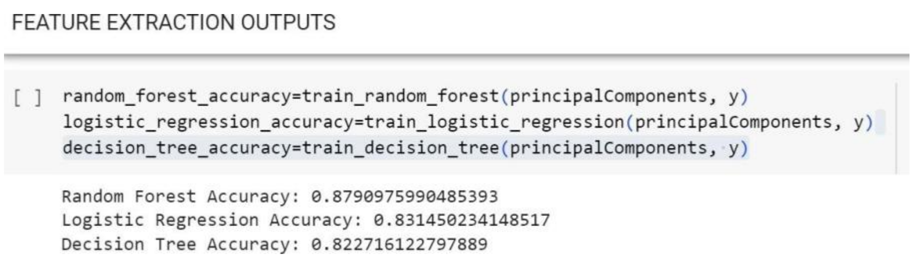


MODELS TESTING ON IMPORTANT FEATURES

```
[ ]  random_forest_accuracy=train_random_forest(X2, y)
     logistic_regression_accuracy=train_logistic_regression(X2, y)
     decision_tree_accuracy=train_decision_tree(X2, y)
```

```
Random Forest Accuracy: 0.9530216308630045
Logistic Regression Accuracy: 0.9245521445030848
Decision Tree Accuracy: 0.9396045491711886
```

Feature Extraction output accuracy:


FEATURE EXTRACTION OUTPUTS

```
[ ]  random_forest_accuracy=train_random_forest(principalComponents, y)
     logistic_regression_accuracy=train_logistic_regression(principalComponents, y)
     decision_tree_accuracy=train_decision_tree(principalComponents, y)
```

```
     Random Forest Accuracy: 0.8790975990485393
     Logistic Regression Accuracy: 0.831450234148517
     Decision Tree Accuracy: 0.822716122797889
```

So out of those feature selection is gives the best accuracy and out of all modelsrandom forest is best model it gives the higher accuries.

CLASSIFICATION MODELS

Defintion:-

Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:

1. Binary Classifier:

If the classification problem has only two possible outcomes, then it is called as Binary Classifier.

Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

2. Multi-class Classifier:

If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.

Example: Classifications of types of crops, Classification of types of music.

Types of ML Classification models:

Classification Algorithms can be further divided into the Mainly two category:

Linear Models:-

1. Logistic Regression
2. Support Vector

   Machines

2. Non-linear Models:-

1.K-Nearest Neighbours

2Kernel SVM

3.Naïve Bayes

4.Decision Tree Classification

5.RandomForest Classification

We are doing the classification with three models both linear and non linear .logitsiv regression (linear model) and decision tree and random forest (non linear model.

RANDOM FOREST

Definition:-

ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple* classifiers to solve a complex problem and to improve the performance of the model.

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Assumptions:

1. There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

2. The predictions from each tree must have very low correlations.

Working:

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

1. Select random K data points from the training set.

2. Build the decision trees associated with the selected data points (Subsets).

3. Choose the number N for decision trees that you want to build.

4. For new data points, find the predictions of each decision tree, and assign the newdata points to the category that wins the majority votes.

STEPS OF IMPLEMENT ATION:

1. Data Pre-processing step:-

in this step we are taking the features and checking for null values and replace null values and taking the data into X as independent and Y as dependent features and next we have to slitdata into train and test using train_test_split function into 80 and 20 perecnt ration.

```
# Splitting the data into train and test sets

IF = ['BidsPortfolioManager', 'Interest', 'MonthlyPayment', 'Rating',
       'PrincipalPaymentsMade', 'NoOfPreviousLoansBeforeLoan',
       'PrincipalBalance', 'InterestAndPenaltyBalance',
       'AmountOfPreviousLoansBeforeLoan',

'PreviousRepaymentsBeforeLoan']
X = df[IF]
```

```
# Split the data into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y,
```

2. Fitting model to the Training set:-

After splitting data then we have to import model and we have to fit it with x_train and y_train then model is trained and ready for prediction. .

```
# Modelling

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
```

3. Predicting the test result:-

After develop the model we have to test with test data to identify the model working.it givesthe output.

```
y_pred = model.predict(X_test)
print(y_pred)

accuracy = accuracy_score(y_test, y_pred)
```

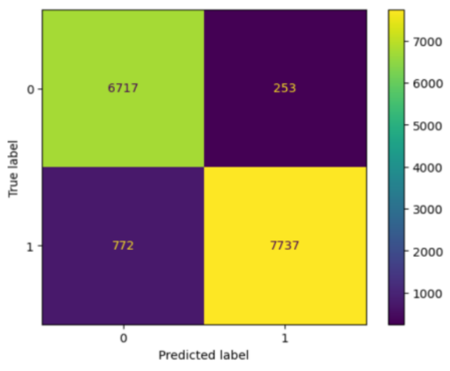
it gives the accuracy as 94 percenatge.

4. Test accuracy of the result(Creation of Confusion matrix):-

Now we have compare the predicted values with actual data and printed the classificationreport and it gives accuracy and score also.

Confusion matric:

```
from sklearn.metrics import confusion_matrix, accuracy_score,
roc_auc_score

cm = confusion_matrix(y_test, y_pred)
```



classification report:

```
import sklearn.metrics as metrics
print(metrics.classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.96      0.93      6970
           1       0.97      0.91      0.94      8509

    accuracy                           0.93     15479
   macro avg       0.93      0.94      0.93     15479
weighted avg       0.94      0.93      0.93     15479
```

5. creating the pipeline for model:

After completing all details I created a pipeline for model to store it in .pkl file. For this requirement we had to import pickle module and pipeline library with that pipeline library we have to created pipeline and and store as classification.pkl format with pickle dump and when we have requirement we have to import that classification.pkl with pickle.;load function. And also we can test data it will also give better accurracy

```
from sklearn.pipeline import Pipeline
import pickle
```

fit the pipeline:-

```
pipeline_randomforest.fit(X_train, y_train)
```

save it with pickle.dump:-

```
# Save the model to a pickle file
import pickle

with open('Classification_task.pkl', 'wb') as file:
```

import model with pickle.load:-

```
# Open the model from the pickle file

with open('Classification_task.pkl', 'rb') as file:
```

predict with pipeline:-

```
y_data_pred = Classify_model.predict(X_test)
```

```
print(y_data_pred)
```

accuracy of model with pipeline:

```
accuracy_pipeline_randomforest = accuracy_score(y_test, y_data_pred)
```

Advantage of model:

1. It takes less training time as compared to other algorithms.

2. It predicts output with high accuracy, even for the large dataset it runs efficiently.3.It

can also maintain accuracy when a large proportion of data is missing.

Credit Risk Analysis of P2P Lending Platform

Regression Analysis

For Regression Model we have created 3 Target variables

1- Equated Monthly Instalments (EMI)
2- Eligible Loan Amount (ELA)
3- Preferred Return on Investment (PROI)

I. Loan Tenure

Loan Tenure, or the number of months starting from the loan issue date until its maturity date, is a key factor in calculating all 3 target variables,

Calculation Procedure:

LoanTenure = $(MaturityDat\_Original_{year} - LoanDate_{year})\ x\ 12 -$

$(MaturityDate\_Original_{month} - LoanDate_{month})$

II. Equated Monthly Instalments (EMI)

• EMI refers to the fixed amount of money the borrower pays to a lender as part of the repayment towards an outstanding loan within a special period.

$$\text{EMI} = \text{P x r x} \frac{(1+r)^n}{((1+r)^n-1)}$$

P: “Amount” in a month
R: “Interest”
N: “LoanTenure”

III. Eligible Loan Amount (ELA)

• ELA refers to the amount of loan that a borrower is eligible for after reviewing his data.
• ELA should reflect the amount a lender is willing to grant the borrower.
• Components of ELA:
A : “Applied Amount”
r : “Interest”
n : “LoanTenure”
I : “IncomeTotal”
L : “LiabilitiesTotal”

Calculation Procedure:
For each row in the dataset:
1- Calculate: Total Payment Due = (A + (A*r) * n
2- Calculate: **Max allowable amount = (I – L) * 30%**
3- If (Total Payment Due <= Max allowable amount)
Then ELA = AppliedAmount
Else ELA = Max allowable amount

IV. Preferred Return on Investment (PROI)

Return on Investment or ROI is a popular profitability metric used to evaluate how well an investment has performed, It tries to directly measure the amount of return on a particular investment.

• ROI is calculated using many formulas depending on the case in hand, one of them is

$$\text{ROI} = \frac{\textit{Interest Amount}}{\textit{Loan Amount}} \, x\, 100$$

In order to account for the risk of investing, we used a feature importance technique, followed by the weight of evidence technique to select the highly effective independent variables, and which interval inside each variable has much wait in classifying a default or non-default loan, The method is as follows:

Feature Importance Selection:

1- Using RandomForestRegressor Model from scikit-learn library in Python, we Identified these independent variables as the most important while predicting ROI,

a- Interest

b- AppliedAmount

c- LoanTenure

d- IncomeTotal

e- LiabilitiesTotal

f- DebtToIncome

2- Since Interest is used in the calculation of ROI, then we'll exclude it, and Since

DebtToIncome is covering LiabilitiesTotal information especially when we keep IncomeTotal, then we have excluded LiabilitiesTotal as well.

Weight of Evidence and Information value:

Using the remaining 4 independent variables, we have determine the

effectiveness of specific intervals inside each variable on the LoanStatus, whether the borrower defaulted or not,

For each one of the 4 variables,

1- Divided the range of the variables in 50 equal ranges starting of min value and ending at max.

2- Calculated WoE for each interval using this equation

$$\boldsymbol{WoE_i = \ln \left(\frac{\%(LoanStatus = 1)_i}{\%(LoanStatus = 0)_i}\right)}$$

And then we have Calculated the Information Value on the independent variable using this equation,

$$\text{IV} = \sum_{i=1}^{50}(P(LoanStatus = 1) - P(LoanStatus = 0))\ x\ WoE_i$$

3- Redefine the intervals of each variable depending on the results of WoE.

4- Choose which intervals reduces the risk of default, which would increase it, and which ones are relatively neutral, I came up with these results,

Independent Variable	Interval	Effect on risk of default
LoanTenure	≤ 19 months	decrease risk
LoanTenure	> 25 months	Increase risk
AppliedAmount	850 ≤ amnt ≤ 1175	decrease risk
AppliedAmount	amnt ≥ 2000	increase risk
IncomeTotal	amnt ≤ 1000	decrease risk
DebtToIncome	rate = 0	decrease risk
DebtToIncome	rate ≠ 0	increase risk

All other intervals have neutral effect on a Loan being defaulted or not.

Preferred ROI (PROI) Calculation Procedure:

After defining the intervals that has effect on risk of default of the loan, I used the following

procedure to calculate the risk of value,

```
1- Calculate ROI for each row in the dataset,
2- Set PROI = median (ROI) of the all values in the dataset (anchor value)
3- Check AppliedAmount,
If 850 ≤ AppliedAmount ≤ 1175:
        PROI = PROI – 5 (decrease ROI with 5%)
If AppliedAmount ≥ 2000:
        PROI = PROI + 5 (increase ROI with 5%)
4- Check for LoanTenure,
If LoanTenure ≤ 19:
        PROI = PROI – 5 (decrease ROI with 5%)
If LoanTenure > 25:
        PROI = PROI + 5 (increase ROI with 5%)
5- Check IncomeTotal:
If IncomeTotal ≤ 1000:
        PROI = PROI – 5 (decrease ROI with 5%)
6- Check for DebtToIncome:
If DebtToIncome == 0:
        PROI = PROI – 5 (decrease ROI with 5%)
Else:
        PROI = PROI + (Increase ROI with 5%)
7- Return final PROI.
```

Model Building: -

For the regression task we have selected 10 Independent variables which affects the target variables and 3 dependent variables

Independent Variables: -

‘Age’, ‘AppliedAmount’, ‘DebtToIncome’, ‘LoanTenure’, 'UseOfLoan', 'IncomeTotal', 'Gender', 'MaritalStatus', 'InterestAmount', 'OccupationArea'

Dependent Variables: -

'EMI', 'ELA', 'PROI'

Steps: -

1) Filling Missing values by using Simple Imputer for both categorical features and numerical features

2) Converting Categorical Features into numerical by using One Hot Encoding

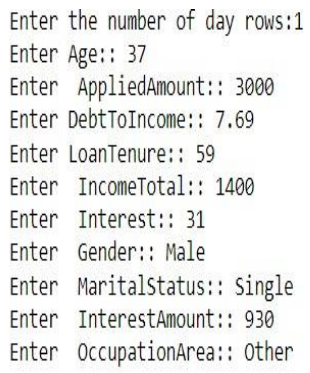
3) Standardization of numerical features by using Standard Scaler

4) Applying Linear Regression and Ridge Regression for prediction

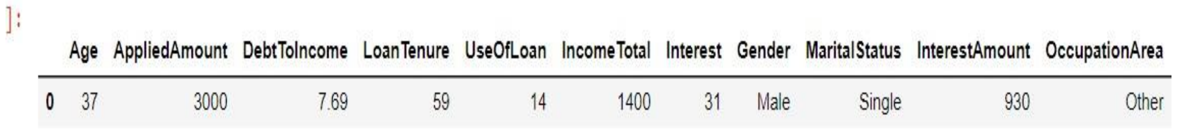
Results: -

The overall R-squared value for Linear Regression is 79.89 % and for Ridge Regression it is 79.87 %.

So, we have chosen Linear Regression for final modelling.

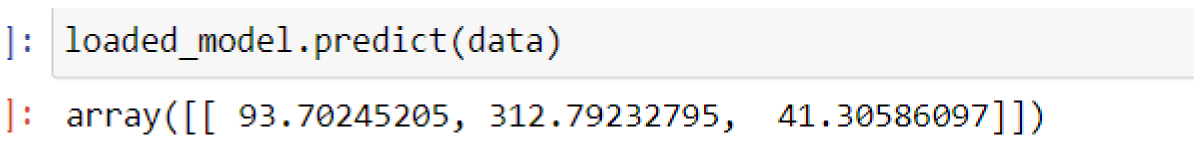


```
Enter the number of day rows:1
Enter Age:: 37
Enter  AppliedAmount:: 3000
Enter DebtToIncome:: 7.69
Enter LoanTenure:: 59
Enter  IncomeTotal:: 1400
Enter  Interest:: 31
Enter  Gender:: Male
Enter  MaritalStatus:: Single
Enter  InterestAmount:: 930
Enter  OccupationArea:: Other
```



]:

	Age	AppliedAmount	DebtToIncome	LoanTenure	UseOfLoan	IncomeTotal	Interest	Gender	MaritalStatus	InterestAmount	OccupationArea
0	37	3000	7.69	59	14	1400	31	Male	Single	930	Other



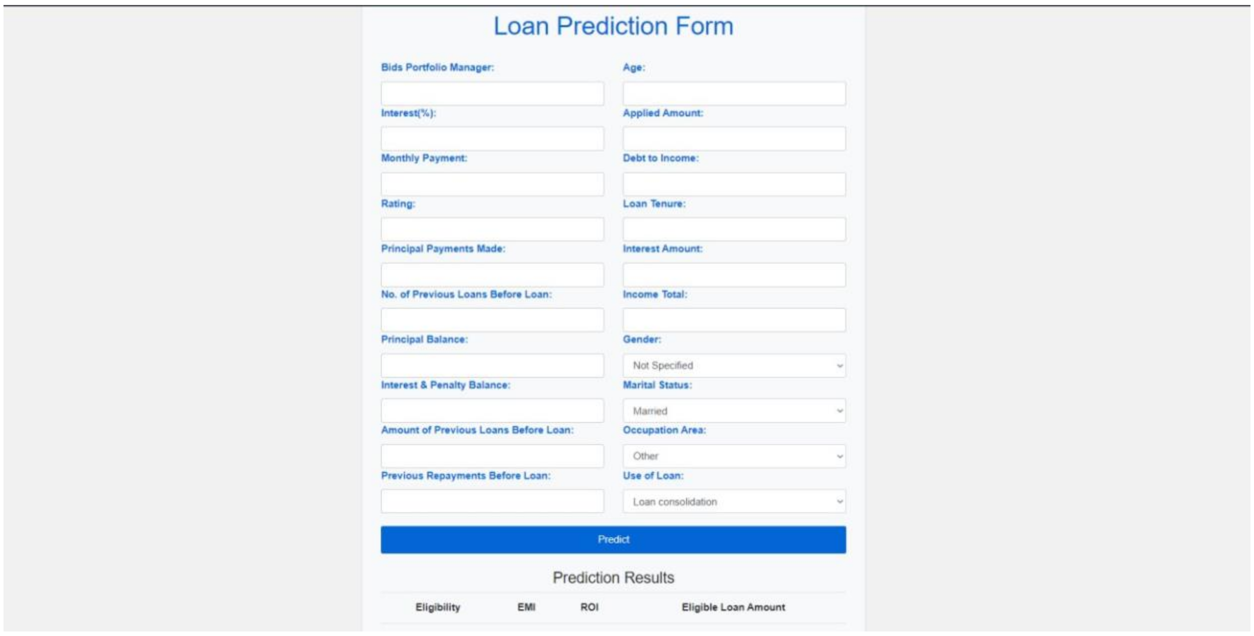
```
]: loaded_model.predict(data)

]: array([[ 93.70245205, 312.79232795,  41.30586097]])
```

Deployment:

The loan prediction system uses a random forest classification model and a multiple linear regression model to predict whether a borrower will be eligible for a P2P loan. The inputs to the system are the borrower's age, debt to income ratio, loan tenure, interest amount, income total, gender, marital status, occupation area, use of loan, and bids portfolio manager. The outputs of the system are the borrower's eligibility, EMI, ROI, and eligible loan amount.

The random forest classification model is used to predict the borrower's eligibility. The model is trained on a dataset of historical loan applications. The multiple linear regression model is used to predict the borrower's EMI, ROI, and eligible loan amount. The model is trained on a dataset of historical loan repayments, and it uses a technique called "linear regression" to predict the borrower's repayment behavior.

The loan prediction system is used by P2P lenders to assess the risk of lending money to borrowers. The system helps lenders to make informed decisions about whether to approve a loan application, and it helps to ensure that borrowers are able to repay their loans.



Loan Prediction Form

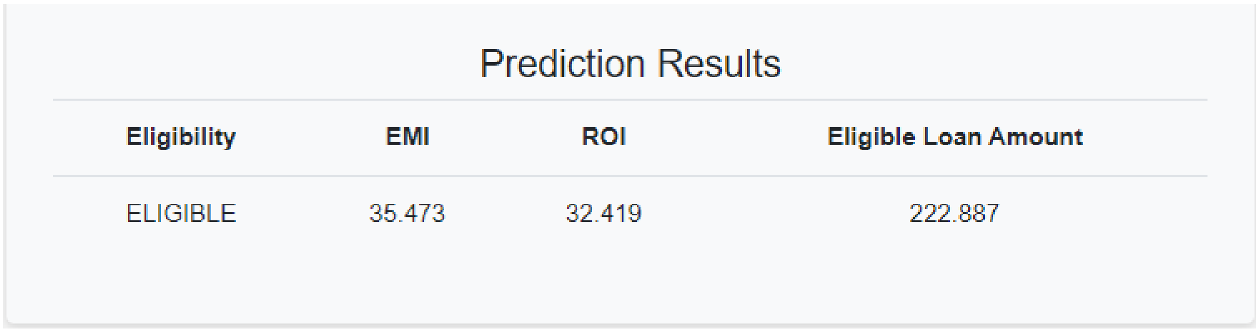
Bids Portfolio Manager:	Age:
Interest(%):	Applied Amount:
Monthly Payment:	Debt to Income:
Rating:	Loan Tenure:
Principal Payments Made:	Interest Amount:
No. of Previous Loans Before Loan:	Income Total:
Principal Balance:	Gender: Not Specified
Interest & Penalty Balance:	Marital Status: Married
Amount of Previous Loans Before Loan:	Occupation Area: Other
Previous Repayments Before Loan:	Use of Loan: Loan consolidation

Predict

Prediction Results

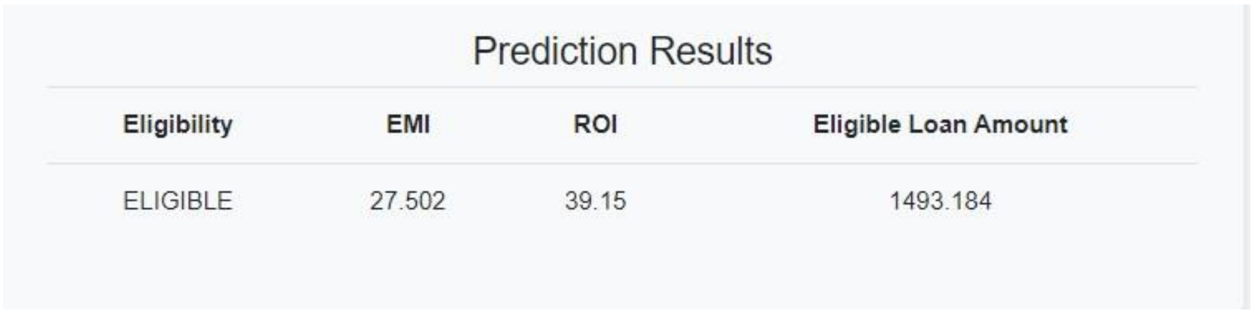
Eligibility	EMI	ROI	Eligible Loan Amount

Sample Outputs:



Prediction Results

Eligibility	EMI	ROI	Eligible Loan Amount
ELIGIBLE	35.473	32.419	222.887



Prediction Results

Eligibility	EMI	ROI	Eligible Loan Amount
ELIGIBLE	27.502	39.15	1493.184

The app is made using the Flask framework, which is a popular Python framework for creating web applications. The app first loads the two models from pickle files. The first model is a classification model that predicts whether a borrower is eligible for a loan. The second model is a regression model that predicts the monthly payment, return on investment, and eligible loan amount for a borrower who is eligible for a loan.

Here is a step-by-step explanation of how the app works:

1. The app loads the two models from pickle files.
2. The app defines two routes: / and /predict.

3. The user enters the borrower's information on the /predict route.
4. The app gets the borrower's information from the request.
5. The app creates two NumPy arrays, one for the classification model and one for theregression model.
6. The NumPy arrays are then passed to the respective models to get the predictions.
7. The predictions are then extracted and returned to the user.

link of the website : https://analysing-financial-statistics-of-p2p.onrender.com

CONCLUSION

In conclusion, the regression and random forest models performed well and exhibited strong predictivecapabilities in the given dataset containing 134,530 rows and 114 columns. Both models were able to effectively predict the target variable(s) based on the available features.

The regression model, being a traditional statistical modeling technique, provided accurate predictions by analyzing the relationships between the target variable and the independent variables. It consideredvarious factors such as borrower demographics, loan characteristics, credit scores, and employment details to make predictions. The regression model demonstrated its ability to capture patterns and make accurate predictions based on these factors.

On the other hand, the random forest model, which is an ensemble learning method, utilized a collection of decision trees to form predictions. By leveraging the power of multiple trees, the random forest model considered different combinations of features and performed well in capturing complex patterns and interactions within the data. This resulted in accurate predictions and robust performance.

Both models proved their effectiveness in predicting the target variable(s) based on the available dataset. Their ability to accurately forecast outcomes showcases their potential to assist in decision- making processes related to loan applications, risk assessment, and financial planning. However, it is crucial to validate and evaluate the models on unseen data and assess their performance metrics, suchas accuracy, precision, recall, or mean squared error, to ensure their reliability in real-world scenarios.

Further analysis and model evaluation, including cross-validation, testing on external datasets, and comparison with other models, can provide additional insights and strengthen the overall conclusionregarding the performance and reliability of the regression and random forest models.