Data Mining and Business Intelligence (ITA5007)

Project

Loan Sanction Prediction

Team Members

Kishore (21MCA0093)

Manibharathi (2MCA0094)

SathiyaNarayanan (21MCA0222)

Introduction

Banks are making a major part of profits through loans. Though a lot of people are applying for loans. It's hard to select the genuine applicant, who will repay the loan. While doing the process manually, a lot of misconception may happen to select the genuine applicant. Therefore we are developing a loan prediction system using machine learning, so the system automatically selects the eligible candidates. This is helpful to both bank staff and applicants. The time period for the sanction of loan will be drastically reduced. In this project we are predicting the loan data by using some machine learning algorithms like Decision Tree, Regression, Regression.

Dataset

The dataset used in this project is taken from Kaggle.

Total No of rows in train Dataset are 615

Total No of rows in test Dataset are 368

Total number of attributes present in the dataset are 13

Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N

Data Summarization

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

Algorithms

Logistic Regression

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.

```
model = LogisticRegression()
model.fit(train_x, train_y)

pred_cv = model.predict(test_x)
print("Accuray" ,accuracy_score(test_y,pred_cv) * 100)

filename = 'logistic_regression.h5'
joblib.dump(model, filename)
```

Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

```
model = DecisionTreeClassifier()
model.fit(train_x, train_y)

pred_cv = model.predict(test_x)
print("Accuray" ,accuracy_score(test_y,pred_cv) * 100)

filename = 'decision_tree.h5'
joblib.dump(model, filename)
```

Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in 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.

As the name suggests, "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." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

```
model = RandomForestClassifier()
model.fit(train_x, train_y)

pred_cv = model.predict(test_x)
print("Accuray" ,accuracy_score(test_y,pred_cv) * 100)

filename = 'random_forest.h5'
joblib.dump(model, filename)
```

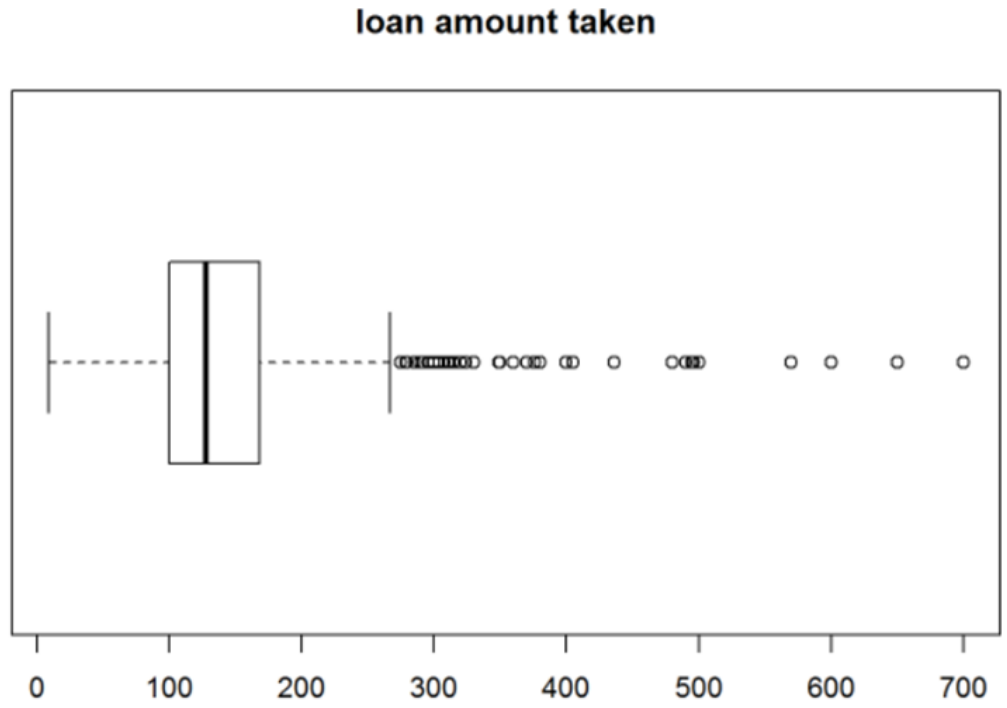
Attribute Selection

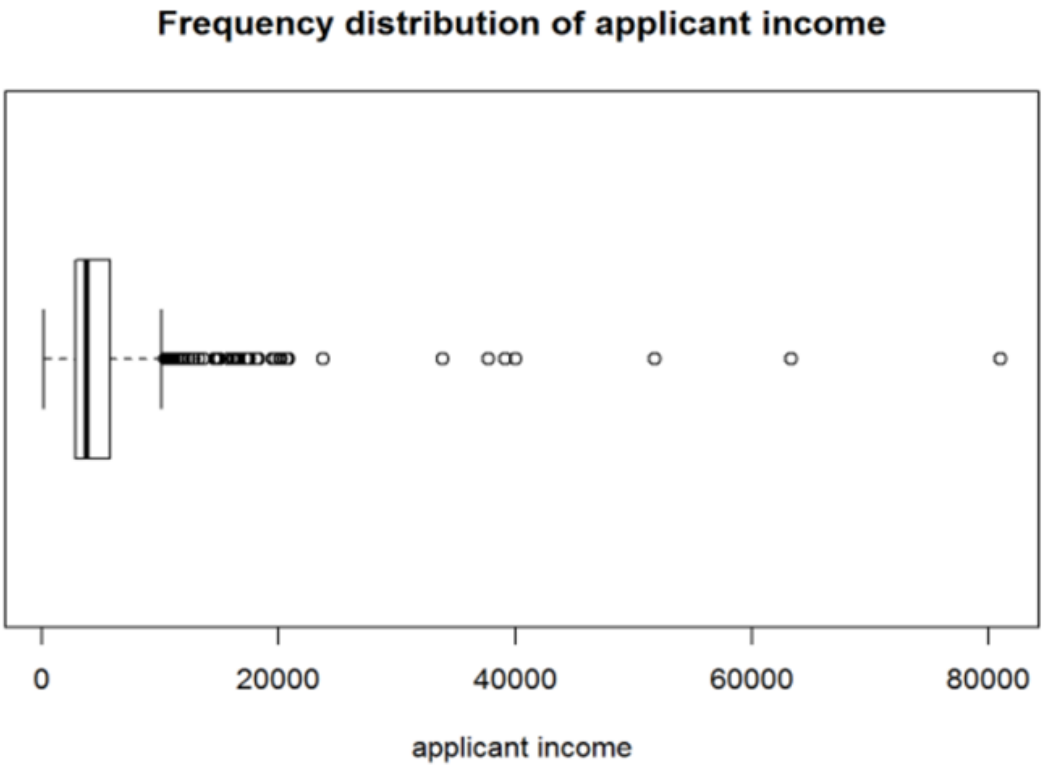
- Gender
- Married
- Dependents
- Education
- Self employed
- Applicant Income
- Coapplication Income
- LoanAmount
- Loan Amount Term
- Credit History
- Property Area

Result

Logistic Regression produces an accuracy of 82.70
Decision Tree produces an accuracy of 70.81
Random Forest produces an accuracy of 81.08

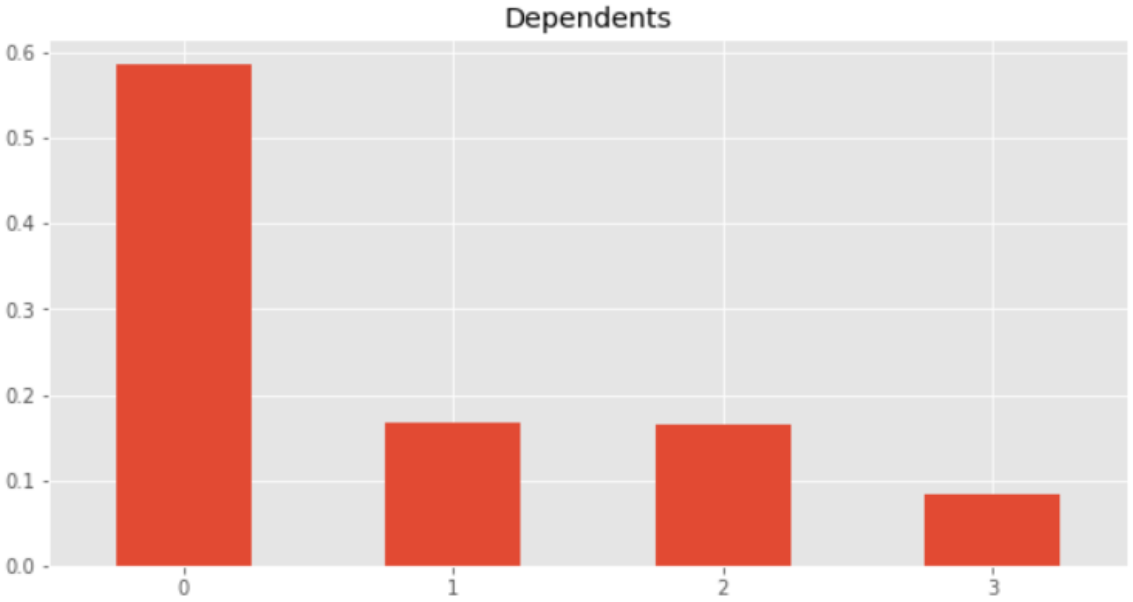
Visualization

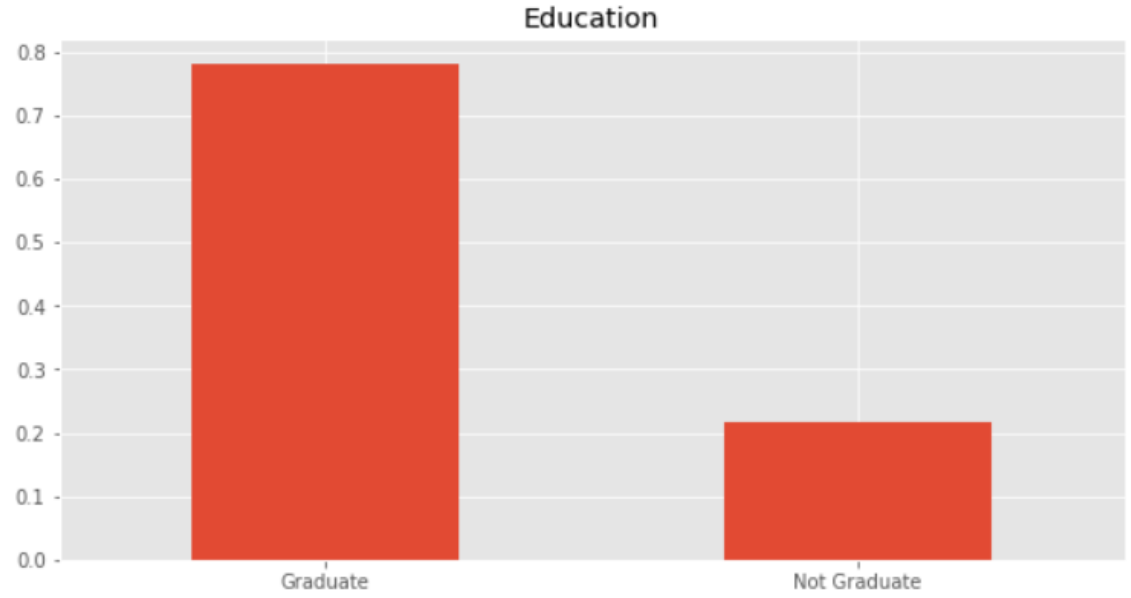





Loan Status
400
350
300
250
200
150
100
50
0
Y
N


ApplicantIncome
CoapplicantIncome
LoanAmount
Loan_Amount_Term
Credit_History
Loan_Status
1 -0.12 0.57 -0.047 -0.019 -0.0047
-0.12 1 0.19 -0.059 0.011 -0.059
0.57 0.19 1 0.037 -0.00061 -0.033
-0.047 -0.059 0.037 1 -0.0047 -0.023
-0.019 0.011 -0.00061 -0.0047 1 0.54
-0.0047 -0.059 -0.033 -0.023 0.54 1
0.8
0.6
0.4
0.2
0.0





Comparison of algorithms

Logistic Regression produces an accuracy of 82.70
Decision Tree produces an accuracy of 70.81
Random Forest produces an accuracy of 81.08

Logistic Regression performs better than the other two algorithms, because logistic regression is best suited for prediction where the outcome is binary (like true/false, yes/no).

Random Forest produces better results than Decision tree as random forest uses multiple decision trees under the hood to do the process.

Conclusion

We tested prediction by creating 3 different models and found out that the Logistic Regression model gives higher accuracy.