Loan Application Status Prediction using Machine Learning

This Article discuss about whether the loan of the applicant will be approved or not on the basis of the details provided by the dataset. Using different machine learning algorithm, we can predict which applicant will get approved or not, this will be very useful for the banks or those who are approving the loan.

Author:

SAFIK(Internship-33)

►Problem Definition

The goal of this project is to build a model that can predict the status of the loan application on the basis of the details provided in the dataset. The data set includes the details like credit history, loan amount, their income, dependents, etc.

►Data Analysis

In the given dataset, which has the details of the applicant like education, credit history, loan amount, etc.

The dataset has total 614 rows and 13 columns. The column names are Loan_ID, Gender, Married, Dependents, Education, Self_Employed, ApplicantIncome, CoapplicantIncome, Loan_Amount, Loan_Amount_Term, Credit History, Property_Area and Loan_Status.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

Information of the dataset

The column Loan_Status is our dependent/target, all other columns are independent/features. The target is categorical therefore the problem is a classification problem.

There are some features which contains null values. The variables Gender, Married, Dependents, Self_Employed, Loan_Amount, Loan_Amount_Term and Credit History contains the null values.

```
Loan_ID              0
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount          22
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
dtype: int64
```

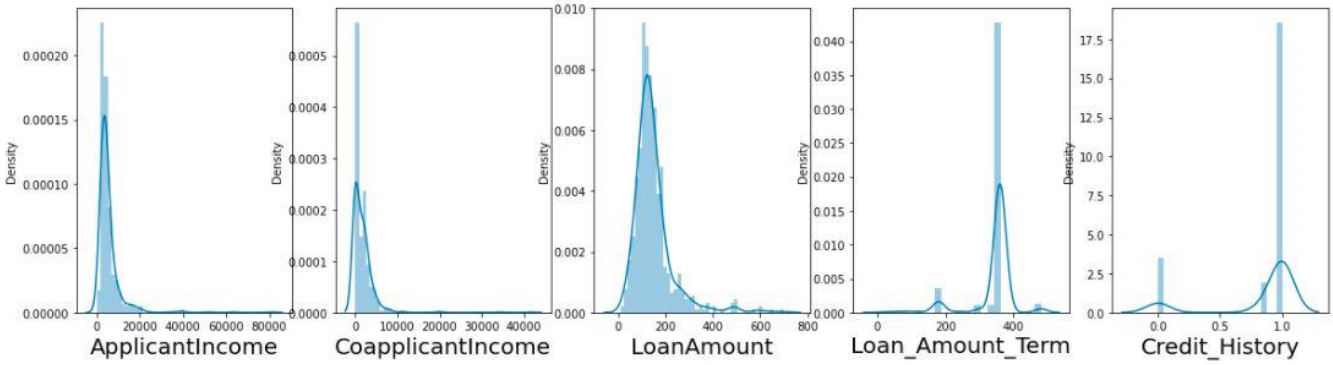
Null values

►EDA Concluding Remarks

- Heatmap was plotted for the correlation among the variables. The ApplicantIncome and Loan Amount has a correlation of 0.57.

Correlation between variables

- The Distribution plot shows that ApplicantIncome and CoapplicantIncome is not normally distributed.



Distribution of the data

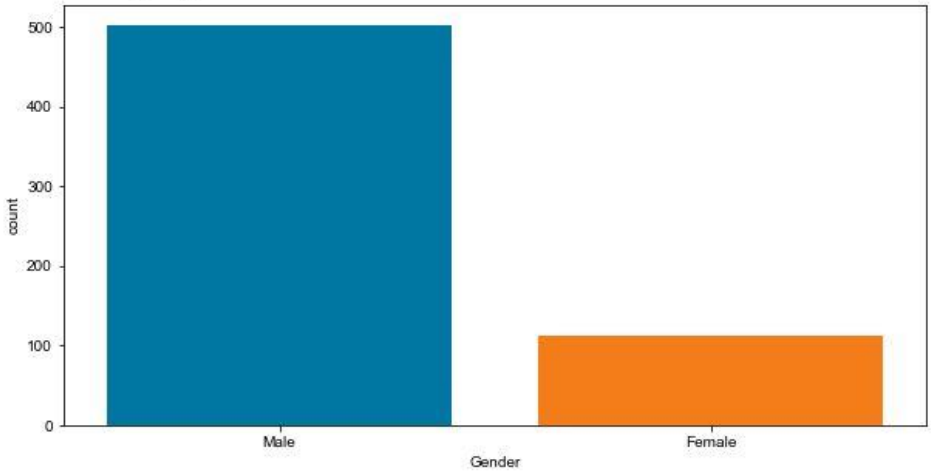
- From the statistical summary, mean is greater than median, large difference between min and 25th percentile, large difference between 75th percentile and max in ApplicantIncome, CoapplicantIncome and LoanAmount. So, outliers are present.

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	614.000000	614.000000	614.000000
mean	5403.459283	1621.245798	146.412162	342.000000	0.842199
std	6109.041673	2926.248369	84.037468	64.372489	0.349681
min	150.000000	0.000000	9.000000	12.000000	0.000000
25%	2877.500000	0.000000	100.250000	360.000000	1.000000
50%	3812.500000	1188.500000	129.000000	360.000000	1.000000
75%	5795.000000	2297.250000	164.750000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

Statistical summary

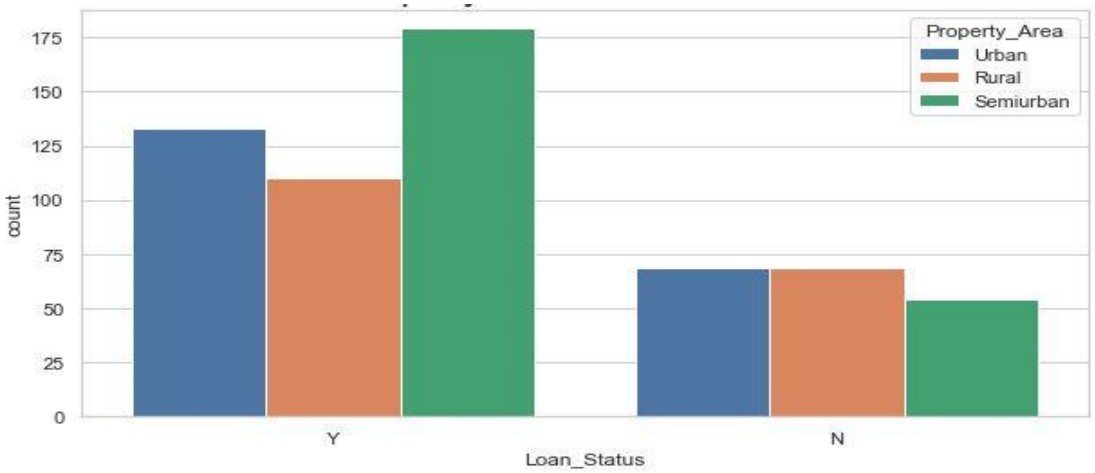
- The column Gender contains 502 Males and 112 Females. Loan approval is not biased on gender.

```
Male      502
Female    112
Name: Gender, dtype: int64
```



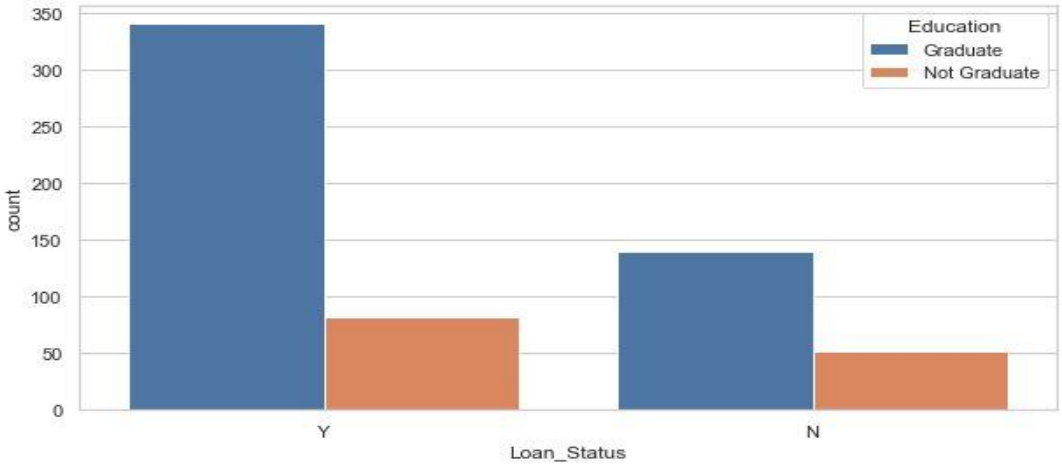
Gender Count

- The applicants from Semi-urban area getting more approval than others from urban and rural.



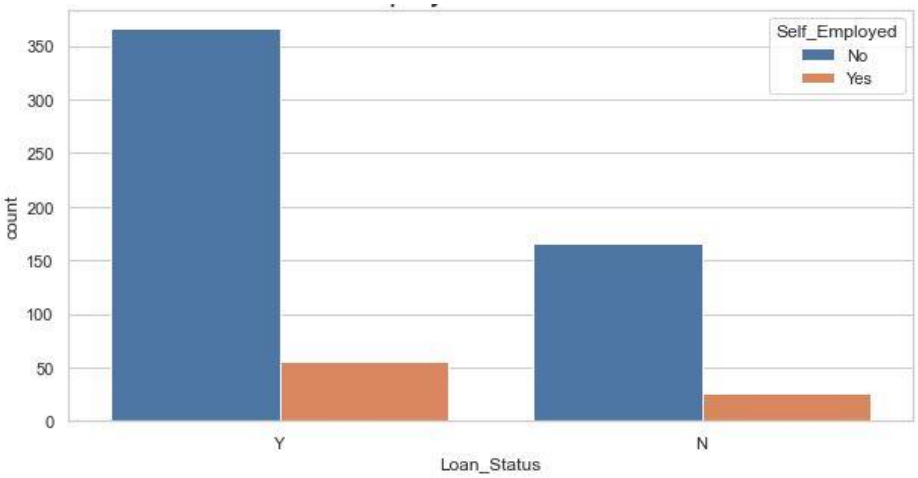
Loan approval with respect to property area

- The chance of getting approval for graduate applicants are higher than not graduate applicant.



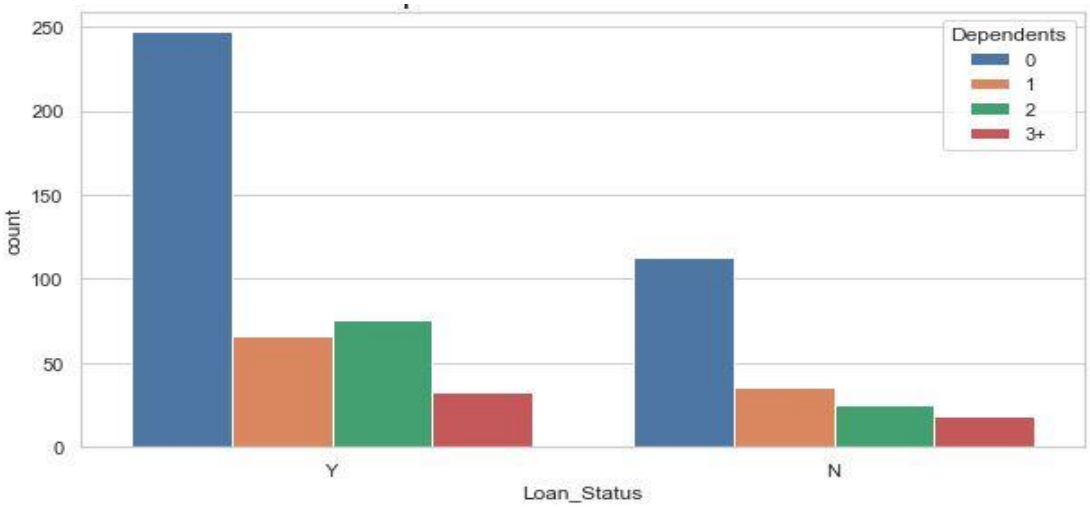
Loan approval with respect to education

- Self-employed applicants are getting less approval.



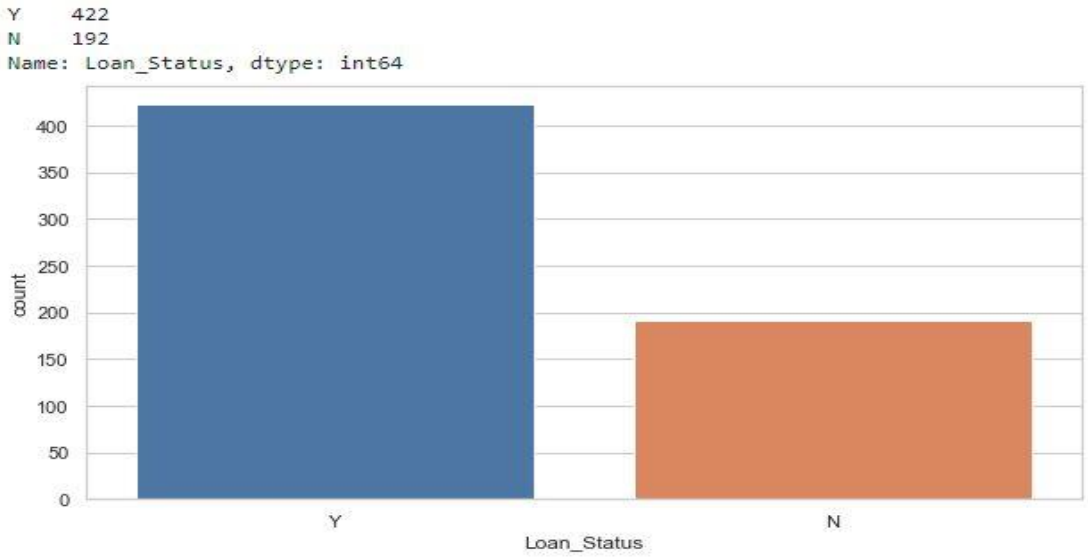
Loan approval with respect to self-employed

- Applicant with no dependents have high chance of getting approval.



Loan approval with respect to dependents

- Target variable has 422 approved application and 192 declined applications. From the dataset 68.72% getting approval and 31.27% not getting approval.



Loan approval count

►Preprocessing Pipeline

Data preprocessing is one of the most important steps in machine learning. The quality of the data is improved in preprocessing, this gives the best result. Data cleaning, Data integration, Data transformation and Data reduction are other stages of data preprocessing.

Treating null values :

Sometimes in the dataset there may be null values present in the columns, that can be missing value, unknown value or no value.

In our dataset null values are present in the columns Gender, Married, Dependents, Self_Employed, Loan_Amount, Loan_Amount_Term and Credit History.

The null values in the dataset can be replaced in different ways. Here we are replacing null values using fillna.

The null values in columns Gender, Married, Dependents, Self_Employed are replaced by using mode() and Loan_Amount, Loan_Amount_Term, Credit History are replaced by mean().

The data distribution in ApplicantIncome and CoapplicantIncome is not normally distributed, more data is on left side only less data on the right side. The both variables are right skewed. Skewness can be removed by using different methods. Here log transform and square root transform is used.

Encoding :

Encoding is used to convert the human readable categorical values into machine readable numeric values. Encoding is also an important step in preprocessing. Different type of encoder are One Hot Encoder, Ordinal Encoder and Label Encoder.

Here am using Label encoder to convert the categorical values into numeric values.

Outlier Removal :

Outliers are the data points which are distant from other data points. That may be due to variation in the measurement or due to error. Outliers need to be removed from the dataset for the better performance of the model.

There are different ways to remove outliers using Z-score and Interquartile range. Here we are using z-score method for the removal of outliers.

Balancing Imbalanced data :

For the classification problems we need to check the target variable is balanced or not. If the target variable is not balanced the output getting from the model will not be accurate.

There is different way to balanced the data, here we are resampling the minority data to the length of majority data, over sampling is done. The data is balanced with highest number of values present in it.

►Building Machine Learning Models

Our problem is a classification problem, we have to predict that the loan status as yes or no. There are several models for the classification problems.

Before fitting the dataset to the model, we need to split the feature variables and target variable. Then splitting the data for training and testing, here the test size is 25% and training size is 75%.

1.Support Vector Machine Classifier :

The support vector machine classifier model is a supervised machine learning model that uses classification algorithm for two group classification problem.

The Accuracy score of the support vector classifier model is 77%.

2.Logistic Regression :

Logistic regression is a supervised machine learning classification algorithm used to predict probability of a target. Only two classes will be there either 0 or 1.

The Accuracy score of logistic regression model is 69.5%.

3.K-Nearest Neighbors classifier :

K-Nearest neighbors classifier is a supervised machine learning algorithm used for classification and regression problem.

The Accuracy score of KNN model is 76%.

4.Decision Tree Classifier :

Decision tree classifier is a supervised machine learning algorithm for both classification and regression problem. But more accurate in classification problem.

The Accuracy score of Decision tree model is 86%.

5.Random Forest Classifier :

Random forest classifier is an ensemble method for both classification and regression problem. This classifier contains number of decision trees and average of their output is taken for the improved accuracy.

The Accuracy score of random forest classifier is 87%.

6.AdaBoost Classifier :

Adaboost classifier is a boosting ensemble technique. This will boost the performance of any machine learning algorithm.

The Accuracy score of adaboost classifier is 72%.

7.Gradient Boosting Classifier :

Gradient boosting classifier is also ensemble method, that works by combining several weak models to create new strong predictive model.

The Accuracy score of gradient boosting classifier is 79%.

Hyper Parameter Tuning

Hyper parameter tuning is used to find out the best parameter which gives the algorithm better performance. Hyper parameter tuning is applied for every model, K-nearest neighbors, Random Forest, Adaboost and Gradient boosting model gives better result.

K-nearest neighbor classifier gives accuracy score 85%. This model predicts 89 True positive cases out of 102 cases and 81 True negative cases out of 98 cases. It predicts 13 False positive cases out of 102 cases and 17 False negative cases out of 98 cases.

```
knn=KNeighborsClassifier(algorithm='auto',n_neighbors=18,weights='distance',leaf_size=1)
knn.fit(x_train,y_train)
predknn=knn.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predknn))
print('Confusion matrix :',confusion_matrix(y_test,predknn))
print('Classification report :',classification_report(y_test,predknn))
```

```
Accuracy score : 0.85
Confusion matrix : [[89 13]
 [17 81]]
Classification report :              precision    recall  f1-score   support

           0       0.84      0.87      0.86       102
           1       0.86      0.83      0.84        98

    accuracy                           0.85       200
   macro avg       0.85      0.85      0.85       200
weighted avg       0.85      0.85      0.85       200
```

Random forest model gives the accuracy score of 88.5%. This model predicts 88 Ture positive cases out of 102 cases and 89 True negative cases out of 98 cases. It predicts 14 False positive cases out of 102 cases and 9 False negative cases out of 98.

```
rfc=RandomForestClassifier(criterion='entropy',min_samples_split=3,bootstrap=True,max_features='auto')
rfc.fit(x_train,y_train)
predrfc=rfc.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predrfc))
print('Confusion matrix :',confusion_matrix(y_test,predrfc))
print('Classification report :',classification_report(y_test,predrfc))
```

```
Accuracy score : 0.885
Confusion matrix : [[88 14]
 [ 9 89]]
Classification report :               precision    recall  f1-score   support

           0       0.91      0.86      0.88       102
           1       0.86      0.91      0.89        98

    accuracy                           0.89       200
   macro avg       0.89      0.89      0.88       200
weighted avg       0.89      0.89      0.88       200
```

Adaboost model gives the accuracy score of 88.5%. This model predicts 90 True positive cases out of 102 cases and 87 True negative cases out of 98 cases. It predicts 12 False positive cases out of 102 cases and 11 False negative cases out of 98 cases.

```
ad=AdaBoostClassifier(algorithm='SAMME.R',base_estimator=rfc,learning_rate=1.0,n_estimators=70)
ad.fit(x_train,y_train)
predad=ad.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predad))
print('Confusion matrix :',confusion_matrix(y_test,predad))
print('Classification report :',classification_report(y_test,predad))
```

```
Accuracy score : 0.885
Confusion matrix : [[90 12]
 [11 87]]
Classification report :               precision    recall  f1-score   support

           0       0.89      0.88      0.89       102
           1       0.88      0.89      0.88        98

    accuracy                           0.89       200
   macro avg       0.88      0.89      0.88       200
weighted avg       0.89      0.89      0.89       200
```

Gradient boosting model gives the accuracy score of 89.5%. This model predicts 90 True positive cases out of 102 cases and 89 True negative cases out of 98 cases. It predicts 12 False positive cases out of 102 cases and 9 False negative cases out of 98 cases.

```
gb=GradientBoostingClassifier(criterion='friedman_mse',learning_rate=1.0,loss='exponential',n_estimators=70)
gb.fit(x_train,y_train)
predgb=gb.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predgb))
print('Confusion matrix :',confusion_matrix(y_test,predgb))
print('Classification report :',classification_report(y_test,predgb))
```

```
Accuracy score : 0.895
Confusion matrix : [[90 12]
 [ 9 89]]
Classification report :              precision    recall  f1-score   support

           0       0.91      0.88      0.90       102
           1       0.88      0.91      0.89        98

    accuracy                           0.90       200
   macro avg       0.90      0.90      0.89       200
weighted avg       0.90      0.90      0.90       200
```

►Concluding Remarks

This project has built a model that can predict the status of loan application. Seven different classifiers are used in this project i.e., Support vector, Logistic regression, K-nearest neighbor, Decision tree, Random Forest, Adaboost and Gradient boosting.

After hyper parameter tuning, the best fitted model is the Random Forest model with accuracy score of 88.5% and ROC-AUC score of 88.54%. This is the best model compared to other models.