Loan Approval System Using Classification Algorithms In Python

Dissertation submitted in fulfillment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

With Specialization in Data Science (AI & ML)

By

Poothi Chandrasekhar Reddy

Reg.No.: 12100403

School of Computer Science and Engineering

Lovely Professional
University Phagwara,
Punjab (India)
Month: April,
Year: 2024

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled “Loan Approval System Using Classification Algorithms In Python” in partial fulfillment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering with a specialization in Data Science (AI & ML) at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Ajay Sharma. I have not submitted this work elsewhere for any degree or diploma. I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents an authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Name of candidate: Poothi Chandrasekhar Reddy

Registration Number: 12100403

Abstract:

Technologies have enhanced the existence of the human species and the quality of life they live. Every day they plan to create something new and different. We have a solution for every other problem we have machines that support our lives and make us somewhat complete in the banking sector candidates will receive proofs/advances before sanction of loan amount. The request was approved by disapproved depending on the historical data of the candidate by the system. A lot every day people apply for a loan in the banking sector, but the banks have limited funds. In this case, a correct prediction would be very beneficial using some class function algorithm.

For example, logistic regression, random forest classifier, support vector machine classifier, etc. The profit and loss of the bank depend on the amount of loans, whether it is a client or a client repaying the loan. Debt collection is the most important thing for the banking sector. The process improvement plays an important role in the banking sector.

Candidate historical data was used to build a machine-learning model using different classification algorithms. The main goal of this post is to predict whether a new applicant has been granted a loan or not using machine learning models trained on a historical data set

Introduction:

Aim: To determine the loan approval system using machine learning algorithms.

Synopsis: Loan approval is a very important process for banking organizations. The systems approved or reject the loan applications. Recovery of loans is a major contributing parameter in the financial statements of a bank. It is very difficult to predict the possibility of payment of loan by the customer. In recent years many researchers worked on loan approval prediction systems. Machine Learning (ML) techniques are very useful in predicting outcomes for large amount of data. In this paper different machine learning algorithms are applied to predict the loan approval of customers. In this paper, various machine learning algorithms that have been used in past are discussed and their accuracy is evaluated. The main focus of this paper is to determine whether the loan given to a particular person or an organization shall be approved or not.

Problem statement:

Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. Here they have provided a partial data set.

Theoretical Background

Machine learning basics :

Machine learning is a subset of artificial intelligence (AI) that involves the development of algorithms and models that allow computers to learn and make predictions or decisions without being explicitly programmed.

Data: ML relies on data for training and learning patterns.
Types: There are different types of ML, including supervised, unsupervised, and reinforcement learning.
Training: In supervised learning, models are trained on labeled data. In unsupervised learning, models find patterns in unlabeled data. Reinforcement learning uses a rewardbased system for decision-making.
Algorithms: ML algorithms include decision trees, neural networks, and support vector machines, among others.
Applications: ML is used in various fields like image recognition, natural language processing, recommendation systems, and autonomous vehicles.
Evaluation: Model performance is assessed through metrics like accuracy, precision, recall, and F1-score.
Iterative Process: ML often involves iterative model training, testing, and refinement. Bias and Ethical Considerations: It's crucial to address biases in data and models and consider ethical implications.
Scalability: ML can scale from small data analysis to big data applications with appropriate tools and infrastructure. **Continuous Learning:** ML models can adapt and improve over time with new data.

Supervised and Unsupervised Learning:

Supervised Learning:

In supervised learning, the algorithm is trained on a labeled dataset. It learns to make predictions or classify data based on input-output pairs.
Common tasks include classification (assigning labels) and regression (predicting numerical values).
Examples: image classification, spam email detection, and predicting house prices.

Unsupervised Learning:

Unsupervised learning deals with unlabeled data. Algorithms find patterns, structures, or relationships in the data without predefined categories.Common tasks include clustering (grouping similar data) and dimensionality reduction.
Examples: customer segmentation, topic modeling, and principal component analysis (PCA).

Machine Learning Algorithm Types That I Used In This Project:

Random Forest:

A Random Forest is an ensemble machine learning algorithm that is primarily used for classification and regression tasks. It is based on the concept of decision trees and combines the predictions from multiple decision trees to make more accurate and robust predictions. Random Forests are widely used in various fields, including data science, finance, healthcare, and more. Here are the key characteristics and components of Random Forest:

1. **Ensemble Learning**: Random Forest is an ensemble method, meaning it combines the results of multiple individual models to make a final prediction. In the case of Random Forest, the base models are decision trees.
2. **Decision Trees**: Each tree in the Random Forest is a decision tree, a simple predictive model that recursively splits the data into subsets based on the values of the input features.
3. **Bootstrapping**: Random Forest uses a technique called bootstrapping, which involves creating multiple random subsets (with replacement) from the original training data. Each tree in the forest is trained on a different bootstrap sample.
4. **Feature Randomness**: In addition to using bootstrapped samples, Random Forest introduces randomness by considering only a random subset of features at each split point when constructing decision trees. This helps in reducing overfitting and increasing diversity among the trees.
5. **Voting or Averaging**: For classification tasks, each tree in the Random Forest predicts a class, and the final prediction is determined by a majority vote (mode) of the individual tree predictions. For regression tasks, the final prediction is typically the average of the individual tree predictions.
6. **Out-of-Bag (OOB) Error**: Since each tree is trained on a bootstrap sample, some data points are left out (out-of-bag) in each iteration. These out-of-bag data points can be used to estimate the model's accuracy without the need for a separate validation set.
7. **Feature Importance**: Random Forest provides a measure of feature importance, indicating how much each feature contributes to the model's performance. This can be useful for feature selection and understanding the importance of different variables.

8. **Robustness**: Random Forests are less prone to overfitting compared to individual decision trees because the combination of multiple trees tends to smooth out the noise and generalize better.
9. **Parallelization**: Training the trees in a Random Forest can be easily parallelized, making it suitable for large datasets.
10. **Hyperparameter Tuning**: Random Forests have several hyperparameters that can be tuned, such as the number of trees in the forest, the maximum depth of the trees, and the number of features considered at each split.

Random Forest is a versatile and powerful machine learning algorithm that is effective in a wide range of applications. It is known for its high accuracy, robustness, and resistance to overfitting. It is often considered one of the top choices for both classification and regression tasks, especially when you want a model that is easy to use "out of the box" and doesn't require extensive hyperparameter tuning.

K-Nearest Neighbors (KNN):

KNN is a simple and intuitive machine learning algorithm used for classification and regression tasks.

- **Classification:** To classify a data point, KNN looks at its 'k' nearest neighbors (data points with similar features) in the training dataset and assigns the class that's most common among those neighbors.

- **Regression:** For regression, KNN predicts a numerical value based on the average or weighted average of the 'k' nearest neighbors' values.

  - 'k' is a user-defined parameter, and the choice of 'k' affects the algorithm's performance. A smaller 'k' value makes the model sensitive to noise, while a larger 'k' value makes it more robust but might lead to over-smoothing.

  - KNN is non-parametric, meaning it doesn't make strong assumptions about the underlying data distribution.

  - It's a lazy learner because it doesn't build an explicit model during training. Instead, it memorizes the entire training dataset for prediction.

  - KNN's performance can be affected by the distance metric used (e.g., Euclidean distance) and the feature scaling.

- It's suitable for small to medium-sized datasets and can be computationally expensive for large datasets since it requires calculating distances between data points.

Support Vector Machine (SVM):

SVM is a powerful machine learning algorithm used for classification and regression tasks.

- **Objective**: SVM finds the optimal hyperplane that best separates data into different classes (for classification) or predicts values (for regression).
- **Margin**: It maximizes the margin, the distance between the hyperplane and the nearest data points from each class.
- **Kernel Trick**: SVM can handle non-linear data by transforming it into a higherdimensional space using kernel functions.
- **Support Vectors**: The data points closest to the hyperplane are called support vectors and crucial for defining the decision boundary.
- **Regularization**: SVM has a regularization parameter (C) that balances maximizing the margin and minimizing classification errors.
- **Well-suited for**: SVM is effective for small to medium-sized datasets and tasks with clear class separation.
- **Strengths**: It's robust, works well with high-dimensional data, and can handle both linear and non-linear problems.
- **Weaknesses**: SVM can be computationally intensive for large datasets and may require careful parameter tuning.
- **Applications**: Used in image classification, text classification, and anomaly detection, among others.

SVM is a versatile algorithm widely used in various domains for its ability to handle complex classification problems.

Decision Tree:

A decision tree is a machine learning model that resembles a flowchart. It's used for classification and regression tasks.

- **Structure**: The tree consists of nodes, with the top node called the root. Nodes split into branches based on a feature's value.
- **Nodes**:
- **Root Node**: Represents the initial decision point.
- **Internal Nodes**: Split data based on specific features.
- **Leaf Nodes**: End points that indicate a final class or regression value.
- **Splitting**: Decision tree splits data at each node into subsets to separate classes or predict values, based on the feature that provides the best discrimination.

- **Objective**: It aims to create a tree that minimizes impurity or error, making accurate predictions for unseen data.
- **Overfitting**: Decision trees can overfit by creating overly complex, deep trees that don't generalize well. Pruning or limiting tree depth can help.
- **Interpretability**: Decision trees are easy to understand and interpret, making them valuable for explaining decisions.
- **Ensemble Methods**: Decision trees can be used as building blocks for ensemble methods like Random Forest.
- **Applications**: Decision trees are used in various fields, including medical diagnosis, finance, and customer churn prediction.

Decision trees are a fundamental and intuitive tool in machine learning, suitable for both beginners and experts, with applications in many domains due to their interpretability and effectiveness.

Logistic Regression:

Logistic regression is a statistical method used for analyzing a dataset in which there are one or more independent variables that determine an outcome. It is most commonly used for binary classification problems, where the outcome variable has two possible values, such as 0/1, Yes/No, True/False, or Pass/Fail. Logistic regression models the probability of the binary outcome as a function of the independent variables. Here are some key points about logistic regression:

1. **Sigmoid Function**: Logistic regression uses the sigmoid (logistic) function to model the probability of the dependent variable taking a particular value. The sigmoid function maps any real-valued number to a value between 0 and 1.

2. **Hypothesis Function**: The logistic regression model's hypothesis function looks like this:
   $P(Y=1)=1/(1+e\text{^}-(b0+b1X1+b2X2+\ldots+bnXn))$

Where:

- $P(Y=1)$ is the probability of the outcome being 1.
- $b0,b1,b2,\ldots,bn$ are the coefficients of the model. • $X1,X2,\ldots,Xn$ are the independent variables.

3. **Coefficient Estimation**: The coefficients ($b0,b1,b2,\ldots,bn$) are estimated using techniques like maximum likelihood estimation. These coefficients describe the relationship between the independent variables and the log-odds of the binary outcome.

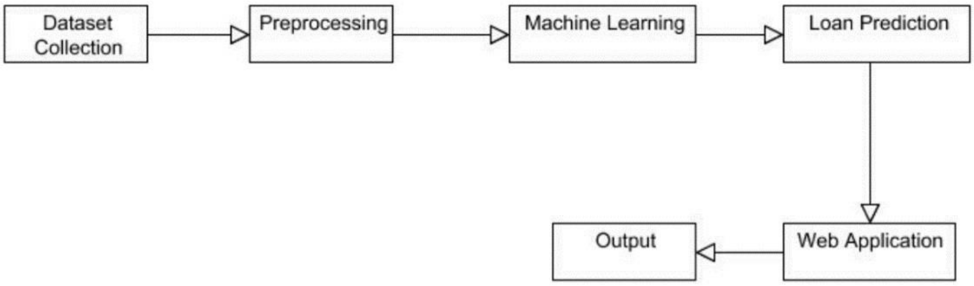
4. **Decision Boundary**: In a binary classification problem, a decision boundary is established, typically at 0.5 probability. If the calculated probability is greater than or equal to 0.5, the outcome is predicted as 1; otherwise, it's predicted as 0.

5. **Regularization**: Regularization techniques like L1 (Lasso) or L2 (Ridge) can be applied to prevent overfitting by penalizing large coefficient values.

6. **Evaluation Metrics**: Common evaluation metrics for logistic regression include accuracy, precision, recall, F1-score, and the receiver operating characteristic (ROC) curve. These metrics help assess the model's performance.

Logistic regression is widely used in various fields, such as medicine (for predicting disease occurrence), finance (for credit scoring), marketing (for customer churn prediction), and many other domains where binary classification is relevant. It's a straightforward and interpretable method for modeling the probability of a binary outcome based on one or more predictor variables.

Flowchart For Loan Approval System:



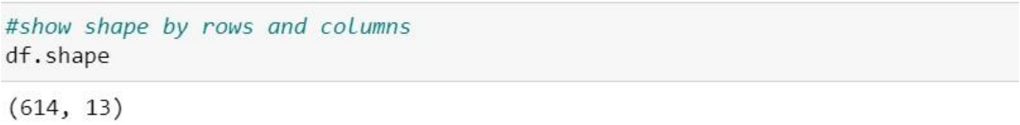
Dataset collection:

The dataset is collected from the kaggle.com. That dataset has some value like gender, marital status, self-employed or not, monthly income, etc. The dataset has the information, whether the previous loan is approved or not depends on the customer's information. That data will be pre-processed and proceed to the next step.

Dataset link:

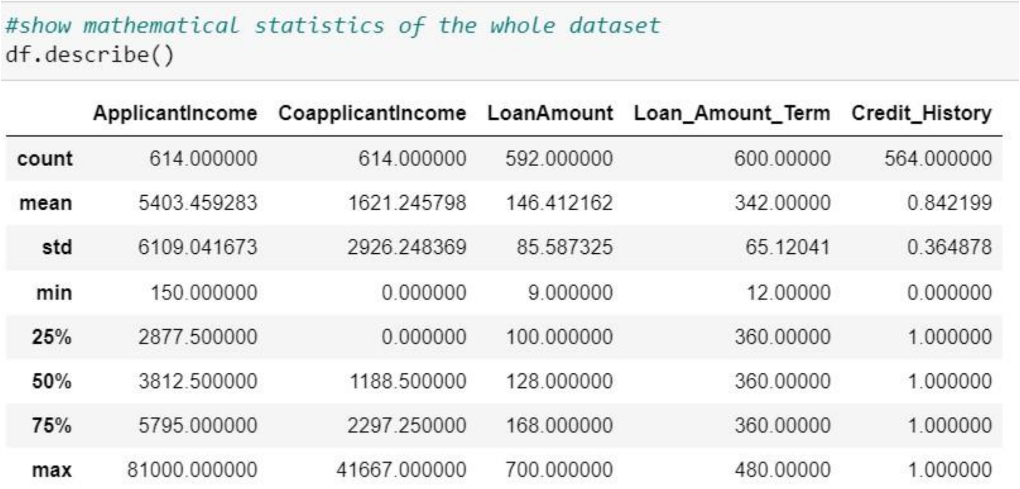
https://www.kaggle.com/datasets/ninzaami/loan-predication/data

Preprocessing:



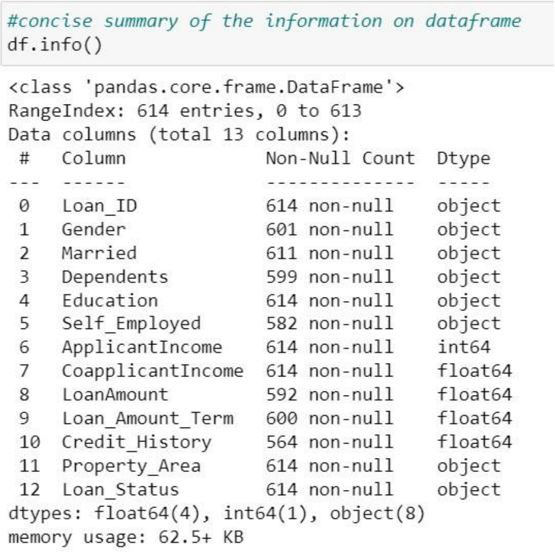
```
#show shape by rows and columns
df.shape
```

```
(614, 13)
```



```
#show mathematical statistics of the whole dataset
df.describe()
```

	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

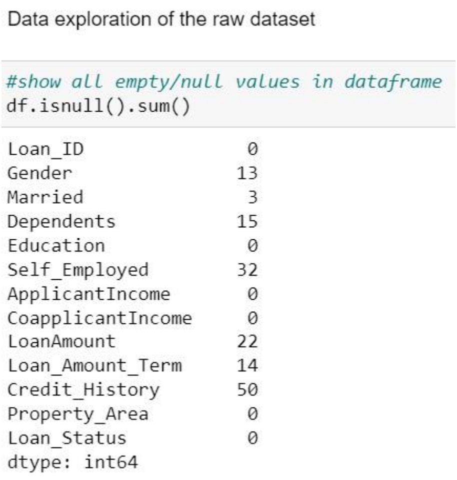


```
#concise summary of the information on dataframe
df.info()
```

```
<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
```

Dropping unnecessary columns:

```
df = df.drop(['Loan_ID'], axis = 1)
```



Data exploration of the raw dataset

```
#show all empty/null values in dataframe
df.isnull().sum()
```

```
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
```

Data Imputation:

Imputation is a technique for substituting an estimated value for missing values in a dataset. In this section, the imputation will be performed for variables that have missing values. Categorical Variables Imputation for categorical variables will be performed using mode.

```
df['Gender'].fillna(df['Gender'].mode()[0],inplace=True)

df['Married'].fillna(df['Married'].mode()[0],inplace=True)

df['Dependents'].fillna(df['Dependents'].mode()[0],inplace=True)

df['Self_Employed'].fillna(df['Self_Employed'].mode()[0],inplace=True)

df['Credit_History'].fillna(df['Credit_History'].mode()[0],inplace=True)

df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0],inplace=True)
```

Numerical Variables:

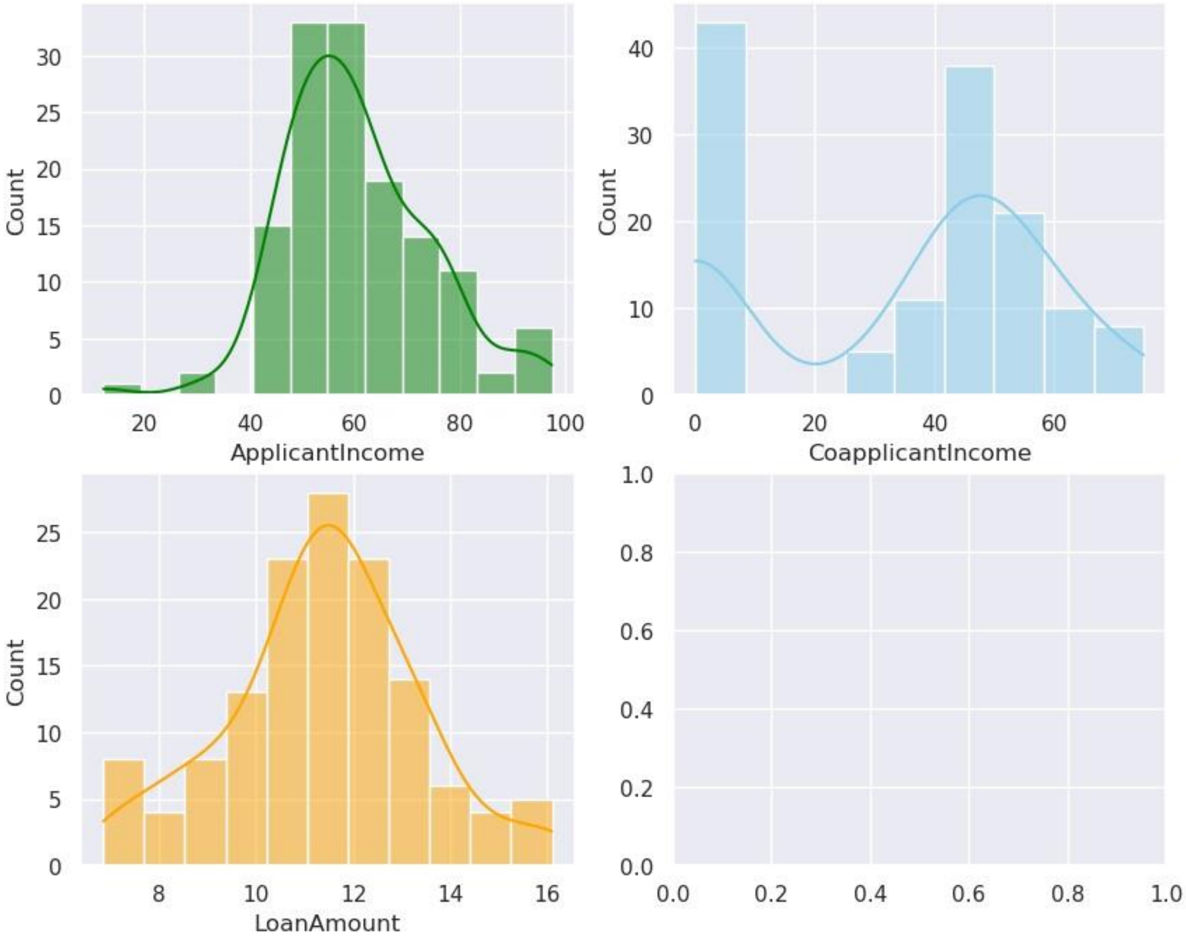
The imputation for numerical variables using mean.

```
df['LoanAmount'].fillna(df['LoanAmount'].mean(),inplace=True
```

Remove Outliers & Infinite values Since there are outliers, the outliers will be removed.

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

df = df[~((df < (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

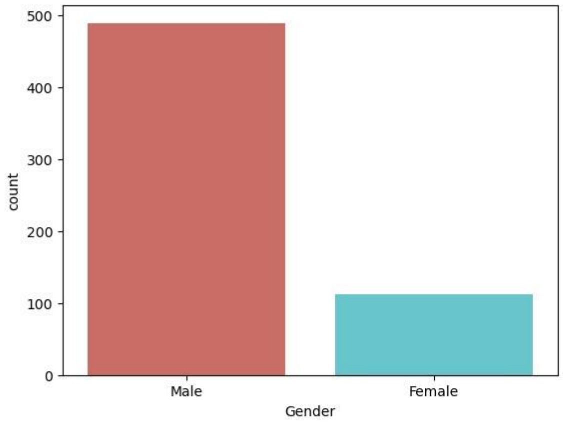
Skewed Distribution Treatment The distributions for ApplicantIncome, CoapplicantIncome, and LoanAmount are positively skewed. I will use square root transformation to normalize the distribution.



Exploratory data analysis:

Analyzing the complete data and all the columns using exploratory data analysis

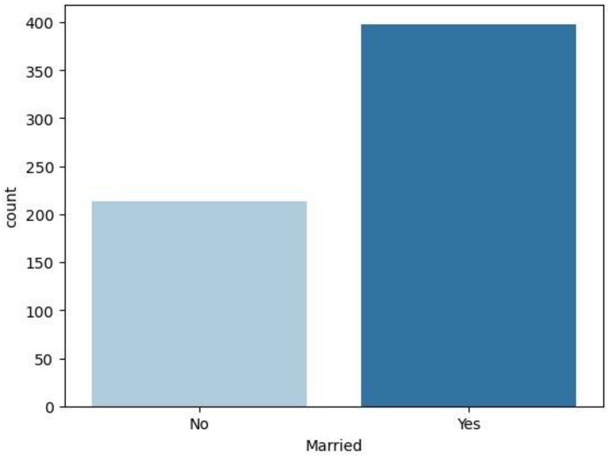
Gender:



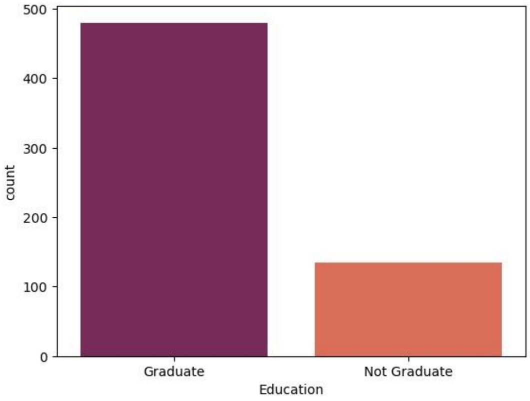
From the above results, the number of male applicants is higher than the one of female applicants and there are some missing values.

Marriage:

The number of married is higher than not married and there are some missing values there.

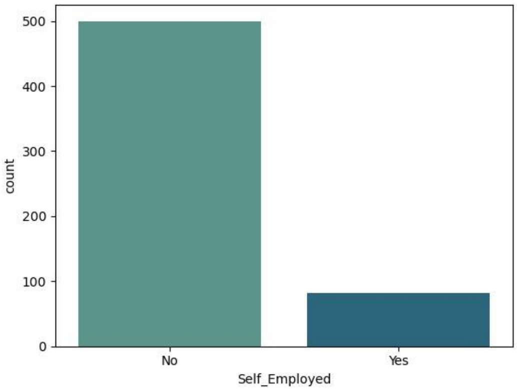


Education:



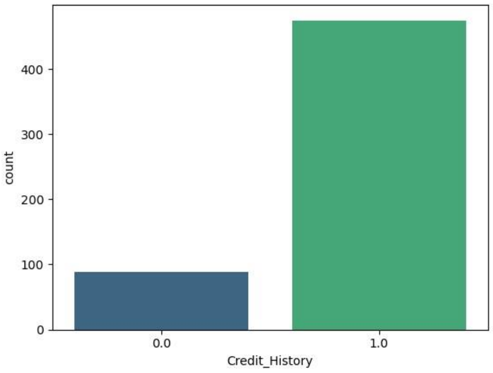
The number of applicants that have graduated is higher than non-graduates.

Self Employed:



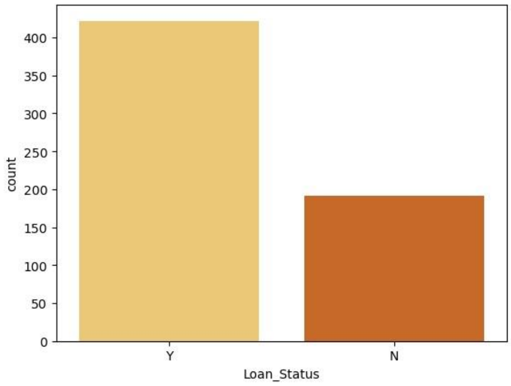
There are more not self employed than self employed with some missing values.

Credit history:



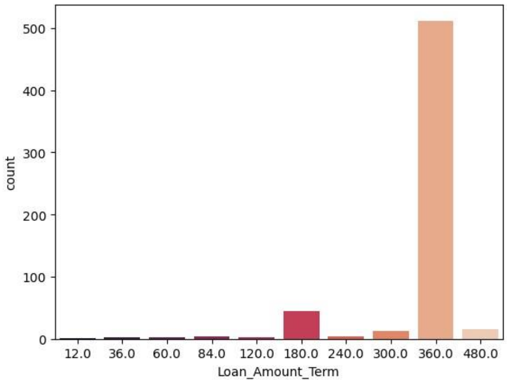
The number of applicants have good credit history is higher compared to applicants that have bad credit history. It also can be seen, there are missing values.

Loan status:



The number of approved loans is higher compared to rejected loans and there is no missing values in this column.

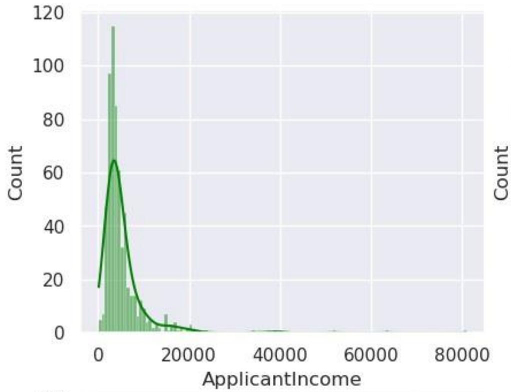
Loan amount term:

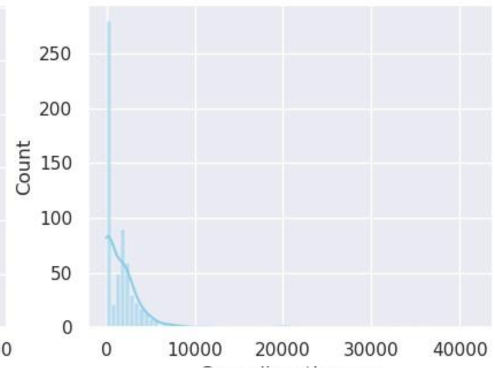


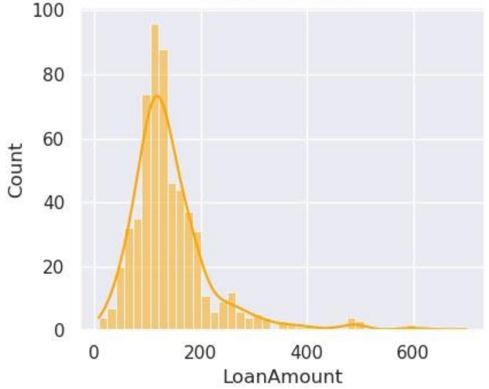
From the results, the 360 days loan duration is the most popular compared to others.

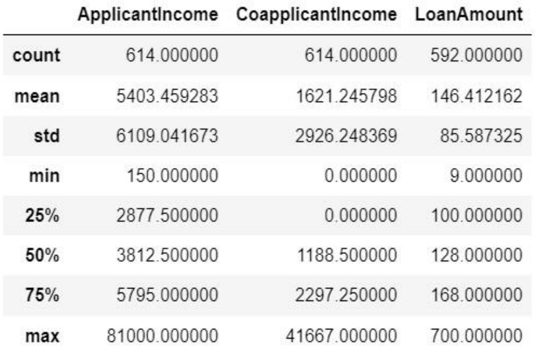
Numerical Variable

I will explore the numerical variables in the dataset





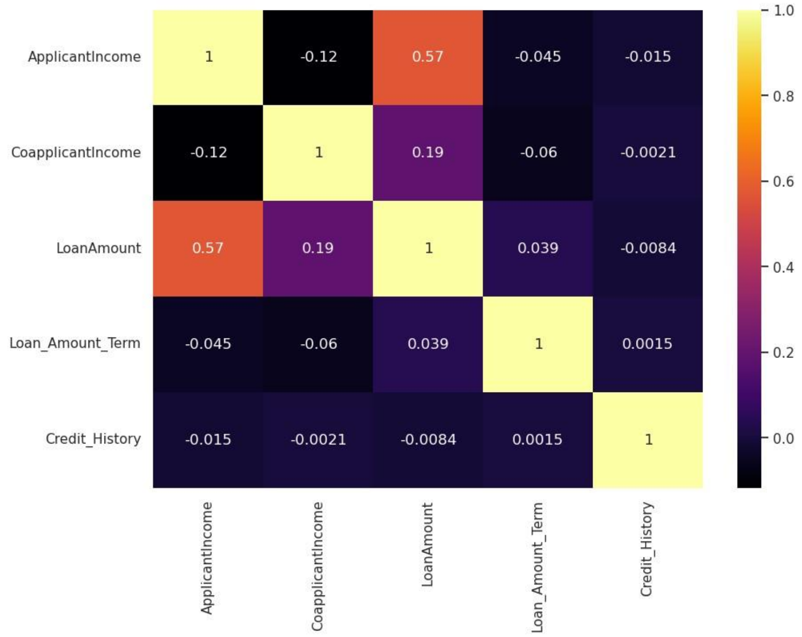




	ApplicantIncome	CoapplicantIncome	LoanAmount
count	614.000000	614.000000	592.000000
mean	5403.459283	1621.245798	146.412162
std	6109.041673	2926.248369	85.587325
min	150.000000	0.000000	9.000000
25%	2877.500000	0.000000	100.000000
50%	3812.500000	1188.500000	128.000000
75%	5795.000000	2297.250000	168.000000
max	81000.000000	41667.000000	700.000000

The distribution of Applicant income, Co Applicant Income, and Loan Amount are positively skewed and it has outliers (can be seen from both histogram and violin plot). The distribution of Loan Amount Term is negatively skewed and it has outliers.

Correlation:

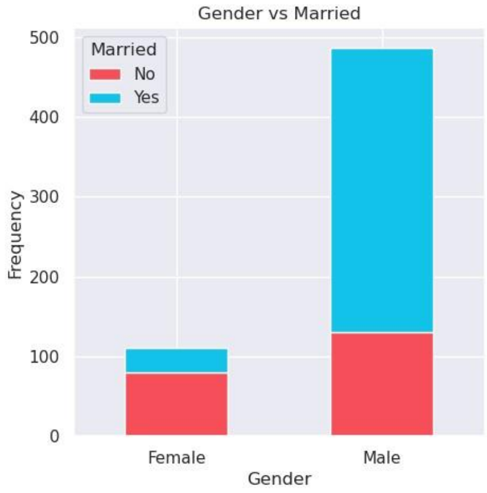


	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
ApplicantIncome	1	-0.12	0.57	-0.045	-0.015
CoapplicantIncome	-0.12	1	0.19	-0.06	-0.0021
LoanAmount	0.57	0.19	1	0.039	-0.0084
Loan_Amount_Term	-0.045	-0.06	0.039	1	0.0015
Credit_History	-0.015	-0.0021	-0.0084	0.0015	1

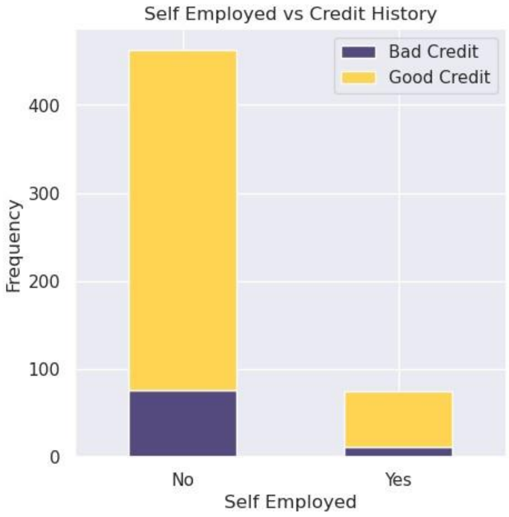
Bivariate analysis (categorical w/ categorical, categorical w/ numerical, and numerical w/ numerical)

There is a positive correlation between Loan Amount and Applicant Income

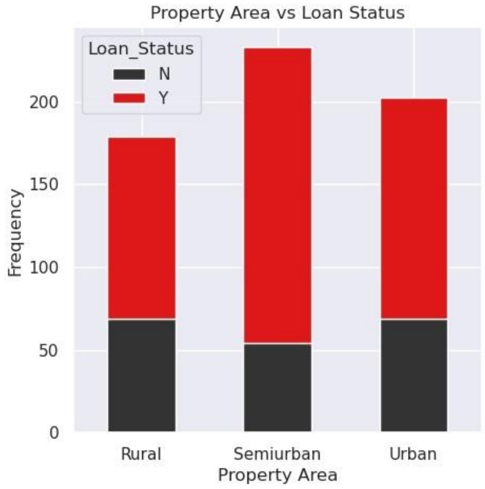
Categorical – Categorical:



Most male applicants are already married compared to female applicants. Also, the number of nonmarried male applicants is higher compared to female applicants who are not married.

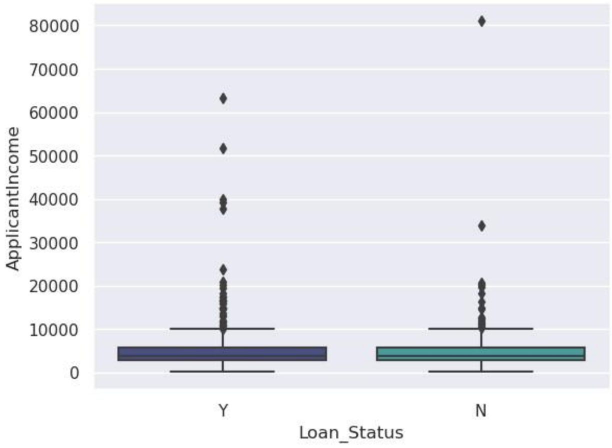


Most non self employed applicants have good credit compared to self employed applicants.

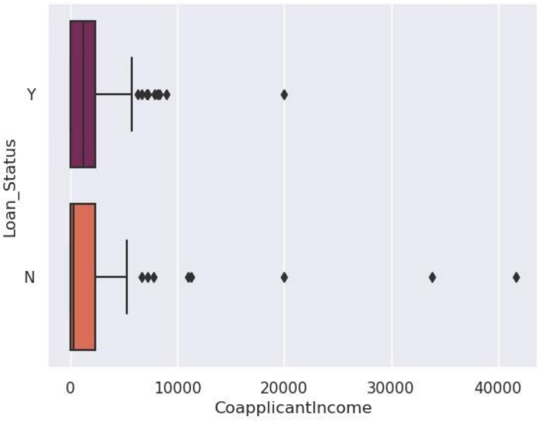


Most loans that got accepted have property in Semiurban compared to Urban and Rural.

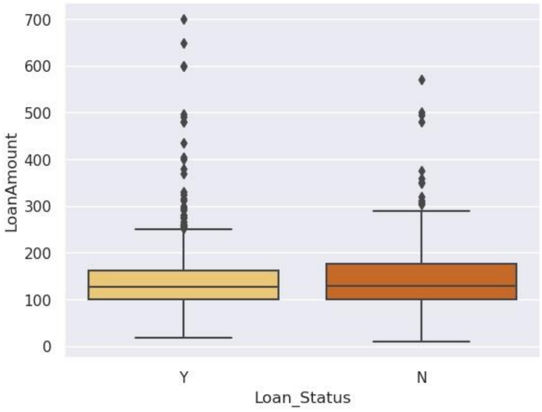
Categorical – Numerical:



It can be seen that there are lots of outliers in Applicant Income, and the distribution also positively skewed



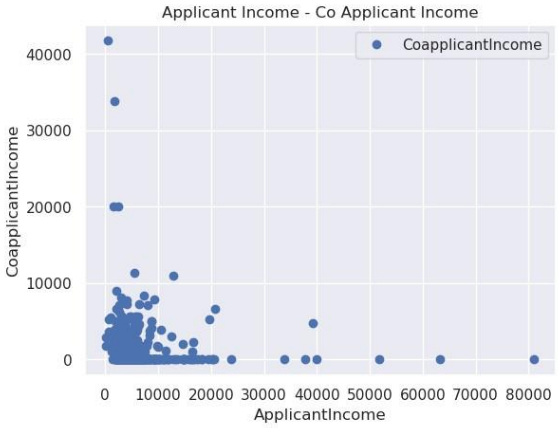
Co Applicant Income has a number of outliers, and the distribution is also positively skewed.



LoanAmount has a high number of outliers, and the distribution is also positively skewed.

Numerical – Numerical:

```
Pearson correlation: -
0.11660458122889966

T Test and P value:
```



```
Ttest_indResult(statistic=13.835753259915661, pvalue=1.4609839484240346e40)
```

There is negative correlation between Applicant income and Co Applicant Income. The correlation coefficient is significant at the 95 per cent confidence interval, as it has a p-value of 1.46

Features Separating:

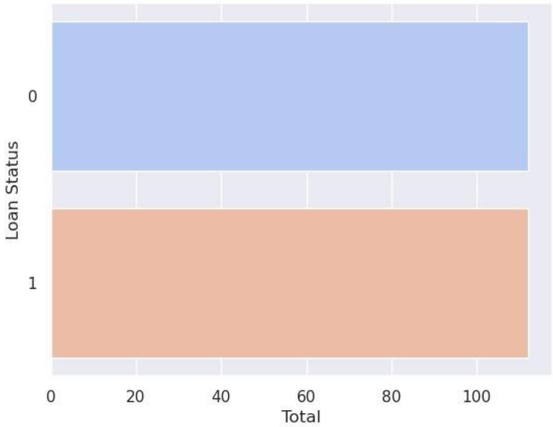
Dependent features (Loan_Status) will be seperated from independent features.

```
X = df.drop(["Loan_Status"], axis=1) y
= df["Loan_Status"]
```

SMOTE Technique:

The number of approved and rejected loans is imbalanced. In this section, the oversampling technique will be used to avoid overfitting,

```
X, y = SMOTE().fit_resample(X, y)
```



The distribution of Loan status is now balanced.

Data Normalization:

Data normalization is a preprocessing technique used in data analysis and machine learning to standardize or scale the features of a dataset. The primary goal is to transform data into a common format to prevent some features from having a disproportionate influence on the analysis.

```
X = MinMaxScaler().fit_transform(X)
```

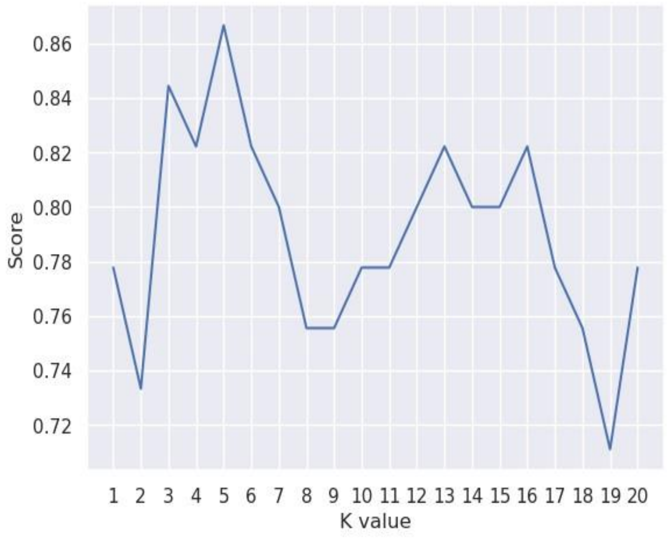
Splitting Data Set:

The data set will be split into 80% train and 20% test.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

K-Nearest Neighbour (KNN):

KNN is a straightforward machine learning algorithm that classifies or predicts by considering the class of the 'k' nearest data points based on a chosen distance metric. It's simple but can be sensitive to 'k' and distance measurement choices.



KNN best accuracy: 86.67%

Support Vector Machine (SVM):

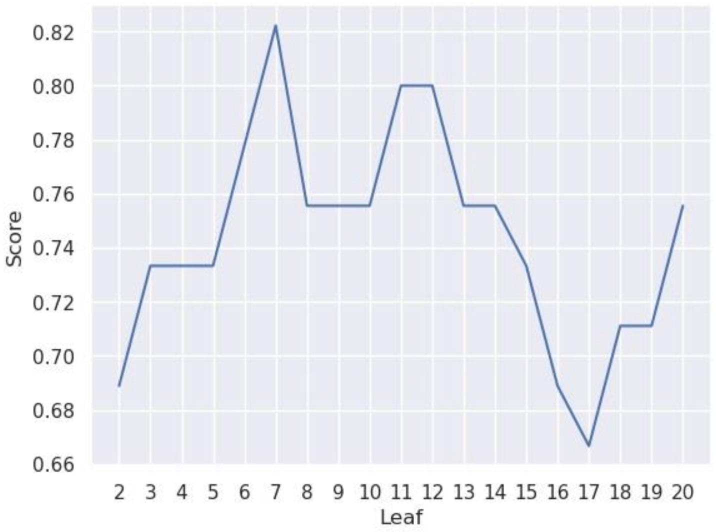
SVM is a machine learning algorithm for classification and regression. It finds the best hyperplane to separate data into different classes. It maximizes the margin (distance) between the hyperplane and the nearest data points from each class. SVM can handle nonlinear data using kernel functions, is effective for small to medium-sized datasets, and is widely used in tasks like image classification and text analysis.

```
precision    recall  f1-score   support

0       0.83      0.83      0.83        23
1       0.82      0.82      0.82        22
     accuracy                           0.82
45    macro avg       0.82      0.82      0.82
45 weighted avg       0.82      0.82      0.82
45

[[19  4]
 [ 4 18]]
SVC accuracy: 82.22%
```

Decision Tree:

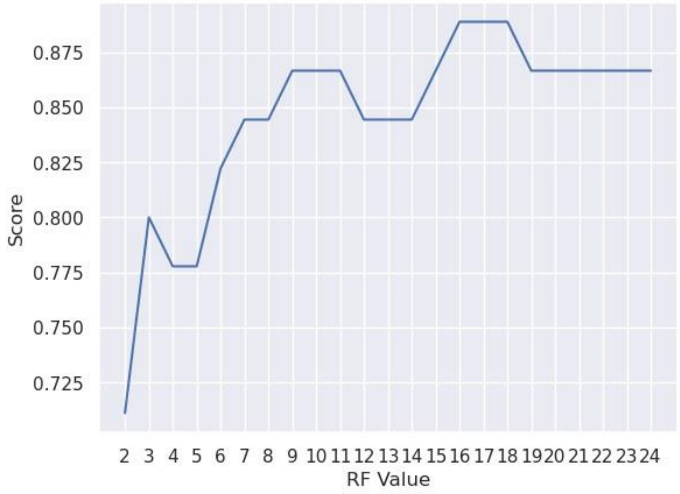
A decision tree is a tree-like model used for classification and regression. It splits data into subsets based on features, creating a flowchart-like structure. It helps make decisions by following the path from the root node to leaf nodes, which represent outcomes. Decision trees are easy to understand and interpret, making them valuable for explaining decisions in various applications.

Decision Tree Accuracy: 82.22%



Random Forest:

Random Forest is a machine learning technique that combines multiple decision trees to make more accurate predictions. It works by creating a collection of decision trees, each trained on a random subset of the data and using a random subset of features. These trees then vote (for classification) or average (for regression) their predictions to provide a robust and reliable final output. Random Forest is known for its ability to handle complex data, reduce overfitting, and offer insights into feature importance.



Random Forest Accuracy: 88.89%

Logistic Regression:

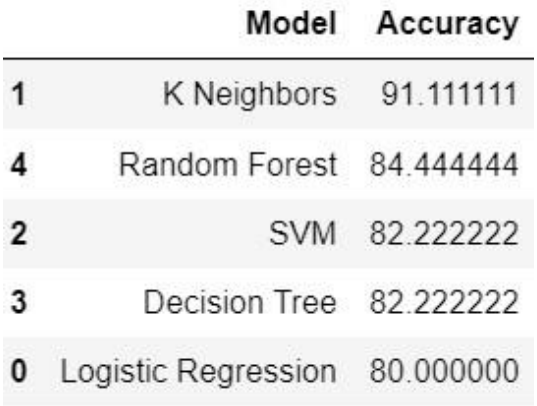
Logistic regression is a statistical model used for binary classification. It calculates the probability of an event happening based on input features and a logistic function. It's simple, interpretable, and widely used in fields like medicine and marketing for tasks such as spam detection and predicting disease outcomes.

```
precision    recall  f1-score   support

0       0.83      0.87      0.85        23
1       0.86      0.82      0.84        22
     accuracy                           0.84
45    macro avg       0.85      0.84      0.84
45 weighted avg       0.84      0.84      0.84
45

[[20  3]
 [ 4 18]]
LR accuracy: 84.44%
```

Model Comparison:

All models can achieve up to 70% accuracy.

The highest accuracy is 91%.



	Model	Accuracy
1	K Neighbors	91.111111
4	Random Forest	84.444444
2	SVM	82.222222
3	Decision Tree	82.222222
0	Logistic Regression	80.000000

REFERENCES

1] Amruta S. Aphale and R. Prof. Dr. Sandeep. R Shinde, “Predict Loan Approval in Banking System Machine Learning Approach for Cooperative Banks Loan Approval”, International Journal of Engineering Trends and Applications (IJETA), vol. 9, issue 8, 2020)

[2] Loan Prediction Using Ensemble Technique, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016

[3] Exploratory data analysis https://en.wikipedia.org/wiki/Exploratory_data_analysis

[4] Pandas Library https://pandas.pydata.org/pandas-docs/stable/

[5] MeanDecreaseAccuracy https://dinsdalelab.sdsu.edu/metag.stats/code/randomforest.hml