**Loan Application Status Prediction**

**Problem Definition**

The dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc. With the help of the dataset we need to predict whether the loan of the applicant will be approved or not.

The data set has: 614 applicants with 13 attributes.

This is classification problem.

The dataset is given in the link below:

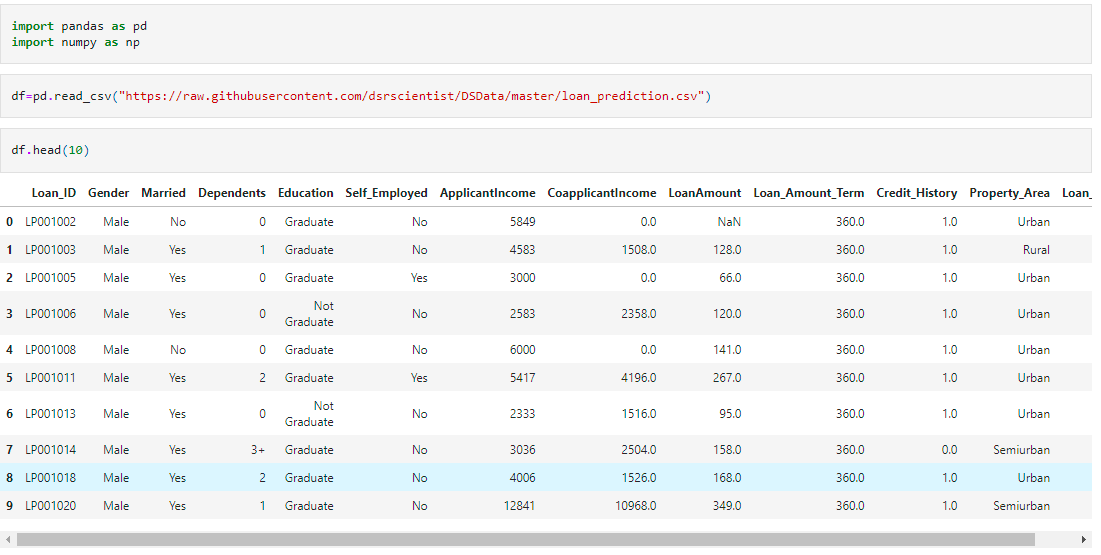
<https://github.com/dsrscientist/DSData/blob/master/loan_prediction.csv>

**Data Analysis**

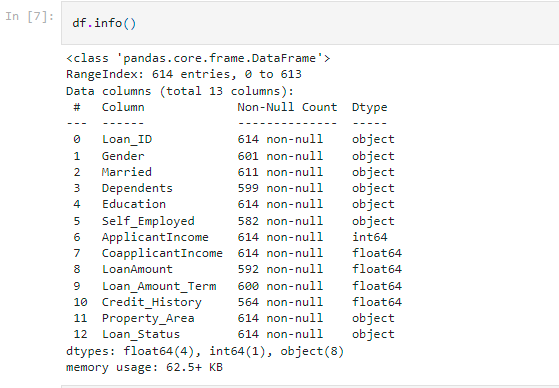
Firstly, check out the problem statement of the data set and analyze the independent variables and therefore the target variable. It is clearly mentioned within the problem statement that we have to predict the approval of the loan or not. After this, we'll clean our dataset. We will inspect the unique values of categorical data type and consider subsequent action plans.

Converting the data type of dataset to integer or float data type. Fill the NaN values if present within the dataset with the acceptable technique. Thereafter we'll analyze the connection between variables using visualization techniques. Then, we will check outliers and therefore the skewness of knowledge. We will attempt to reduce the skewness and hence outliers from the dataset.

We have imported the necessary libraries and the data set in Jupyter Notebook:



Checking the info of the data set:



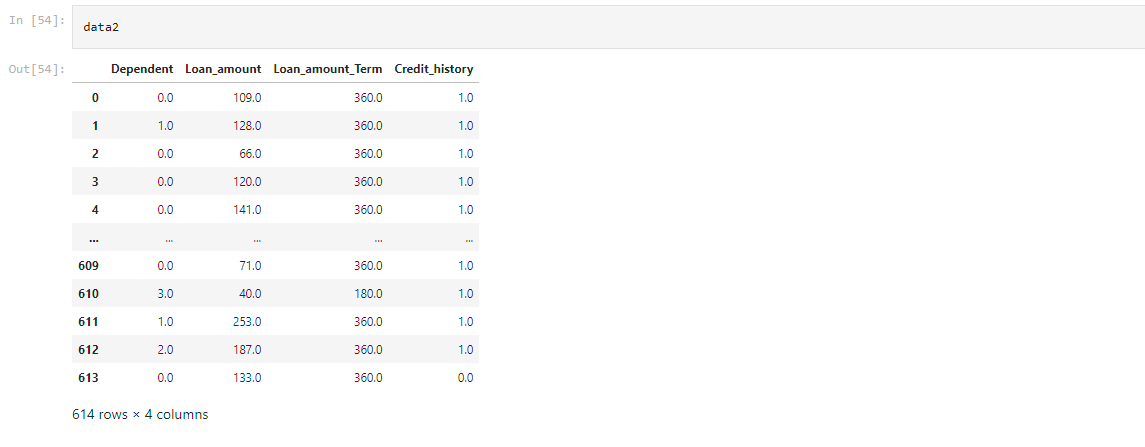
We can see the data type is mainly object type. Also the non- null values are not same. There are 12 columns in the data set.

Now we have checked the unique values of the categorical data type. And replace the object data- type with label encoder.

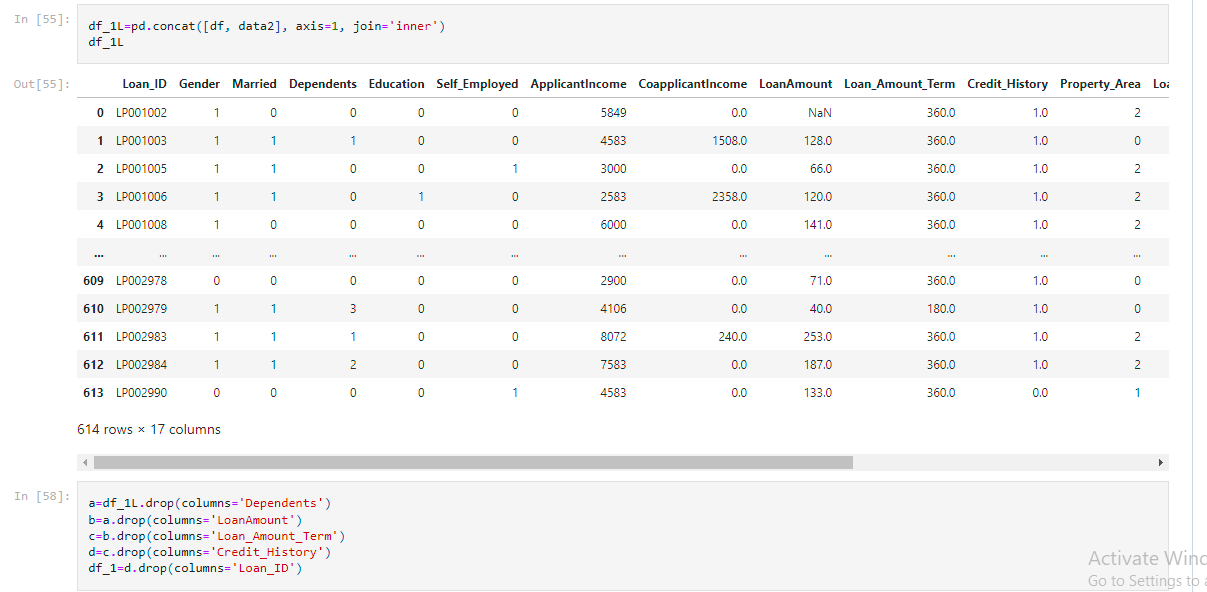


Also replacing 3+ with 3, in dependent attribute.

Replacing the NaN values using KNN imputer for the dependent, loan\_amount, loan amount term and credit-history.

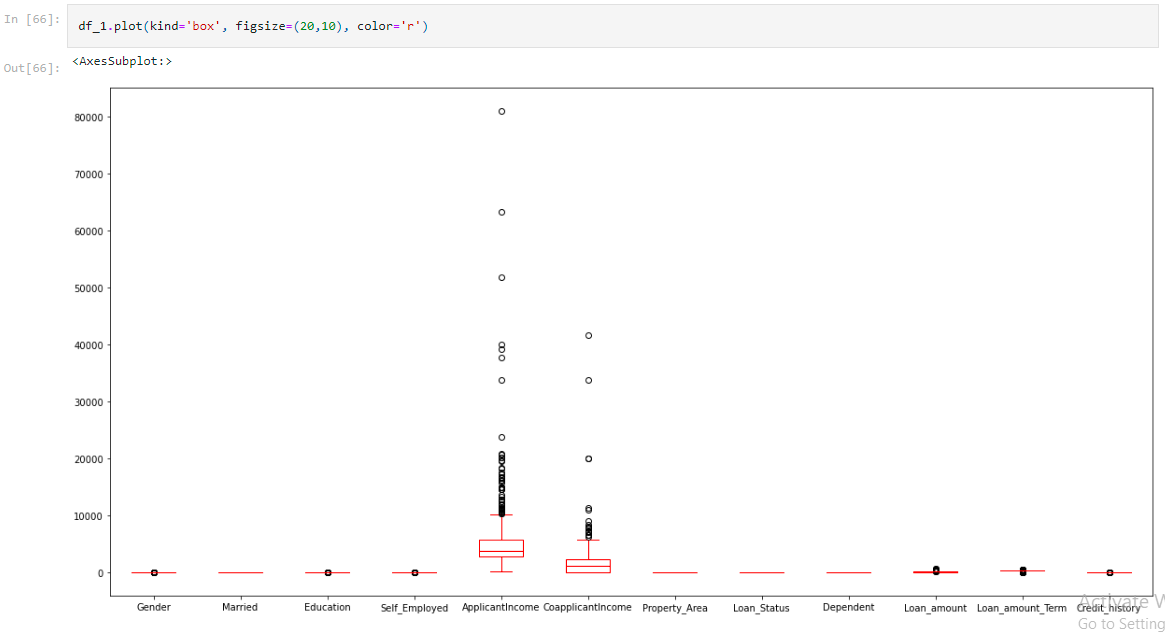


Now joining the converted columns and the data set, dropping the repeated columns (object-type)

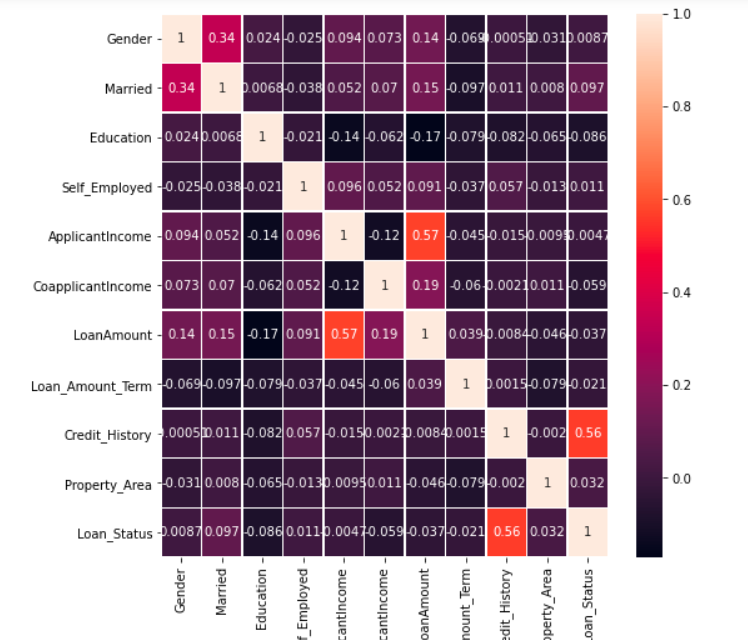


The data set is completely float/integer type. All columns are completed.

Checking the statistical description of the data set and visualizing the data set by box plot.



There are lot of outliers in Applicant Income and co-applicant Income, removing the outliers. Checking the correlation using heat map.



**EDA Concluding Remarks:**

1. Replaced all object to float.
2. Replaced NaN
3. No major correlation between the data given.
4. Removed outliers and reduce the skewness.

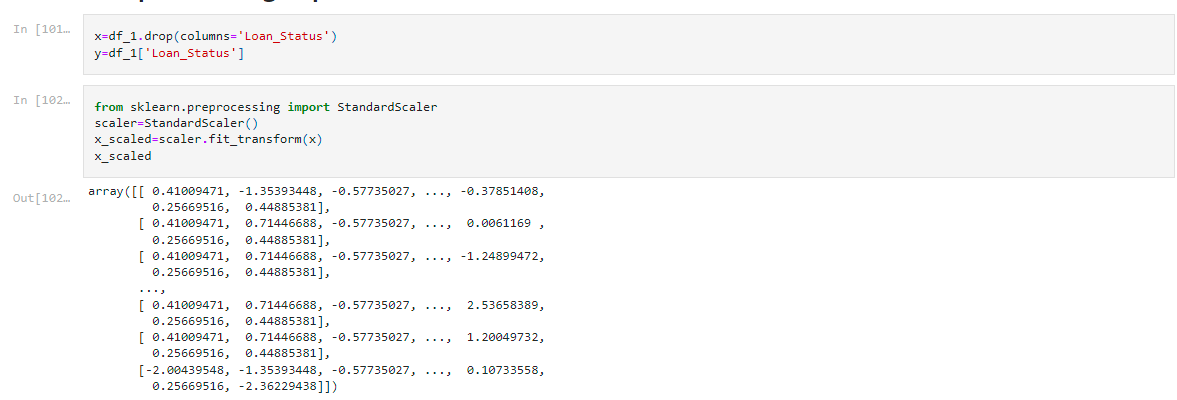
**Pre-Processing Pipeline**

So far we've taken care of the datatype and missing values of our dataset. We have tried to nornmalize the data set. Now we'll split the data set in dependent and target variable, say x-dataset (dependent) and y-dataset (target).

For better application of the techniques we would like to scale the x-dataset. It is assumed that data is generally distributed within each feature. We will scales the data such that distribution centered around 0, with standard deviation of 1. Centering and Scaling happens independently on each feature Based on the sort of model we are building, we will need to normalize the info in such how that the range of all the variables is nearly similar. We will do that easily in python using the StandardScaler function. Once more we will check multicoliinearity for x dataset. Multicollinearity occurs when two or more independent variables are highly correlated with each other during a regression model. VIF score of an experimental variable represents how well the variable is explained by other independent variables. We will check VIF should be less than 5 for each variable.

Now the data set is split into target and dependent variable. In this dataset loan status will be our target variable(y) and rest all will be the dependent variable.

Scaling data is important to increase prediction accuracy



Now checking the variance of the data set.



The VIF is less than 5, good to go ahead.

**Building Machine Learning Models**

Data-Driven decision-making has large involvement of Machine Learning Algorithms. For a business problem, the professional never believe one algorithm. One always applies multiple relevant algorithms supported the matter and selects the simplest model supported the simplest performance metrics shown by the models.

A machine learning model is made by learning and generalizing from training data, then applying that acquired knowledge to new data it's never seen before to form predictions and fulfill its purpose. Lack of knowledge will prevent you from building the model, and access to data isn't enough. The data set is going to be divided into 2 datasets; training and testing.

Whenever we build any machine learning model, we feed it with initial data to coach the model. Then we feed some unknown data (test data) to know how well the model performs and generalized over unseen data. If the model performs well on the unseen data, it’s consistent and is in a position to predict with good accuracy on a good range of input data; then this model is stable. But this is often not the case always! Machine learning models are not always stable and that we need to evaluate the steadiness of the machine learning model. That's where Cross Validation comes into the image

“In simple terms, Cross-Validation is a technique used to assess how well our Machine learning models perform on unseen data”

We can increase the model performance using hyperparameters. Thus, finding the optimal hyperparameters would help us achieve the best-performing mode. Grid Search uses a special combination of all the required hyperparameters and their values and calculates the performance for every combination and selects the simplest value for the hyperparameters. This makes the processing time-consuming and expensive supported the amount of hyperparameters involved.

The most popular sort of Cross-validation is K-fold Cross-Validation. It’s an iterative process that divides the train data into k partitions. Each iteration keeps one partition for testing and therefore the remaining k-1 partitions for training the model. Subsequent iteration will set subsequent partition as test data and therefore the remaining k-1 as train data then on. In each iteration, it'll record the performance of the model and at the top give the typical of all the performance. Thus, it's also a time-consuming process. After that we will train our data set with training dataset and then check the accuracy of our model with the help of Test dataset.

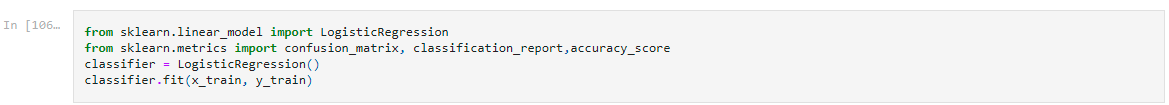


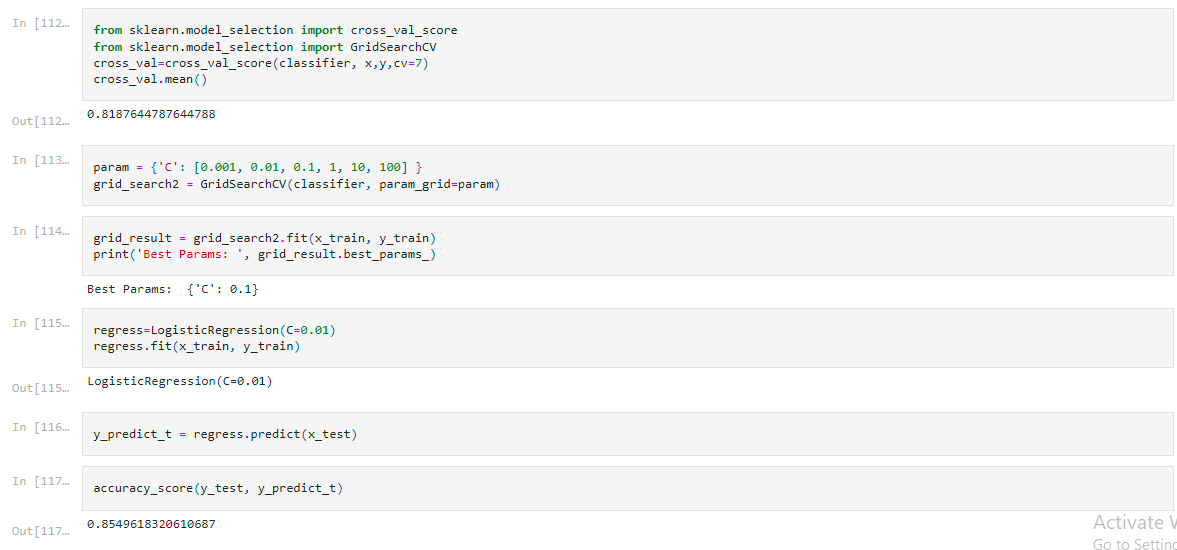
**Now we can start with different algorithms for prediction.**

**1) Logistic Regression**

Logistic regression is usually used where we've to classify the info into two or more classes. One is binary and therefore the other is multi-class logistic regression. Because the name suggests, the binary class has 2 classes that are Yes/No, True/False, 0/1, etc. In multi-class classification, there are quite 2 classes for classifying data. But, before we go allow us to first define the logistic regression.

Now will import Logistic regression from sklearn. We will check accuracy of test data.





Cross value mean= 0.8187644787644788

Accuracy score=0.8549618320610687

### 2) Random Forest

Random forest may be a Supervised Machine Learning Algorithm that's used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average just in case of regression. One of the foremost important features of the Random Forest Algorithm is that it can handle the info set containing continuous variables as within the case of regression and categorical variables as within the case of classification. It performs better results for classification problems.



Accuracy is 0.8549618320610687

# 3) KNN algorithm

K Nearest Neighbor algorithm falls under the Supervised Learning category and is employed for classification (most commonly) and regression. It’s a flexible algorithm also used for imputing missing values and resampling datasets. Because the name (K Nearest Neighbor) suggests it considers K Nearest Neighbors (Data points) to predict the category or continuous value for the new Data point. The algorithm’s learning is:

1. Instance-based learning: Here we don't learn weights from training data to predict output (as in model-based algorithms) but use entire training instances to predict output for unseen data. 2. Lazy Learning: Model isn't learned using training data prior and therefore the learning process is postponed to a time when prediction is requested on the new instance.

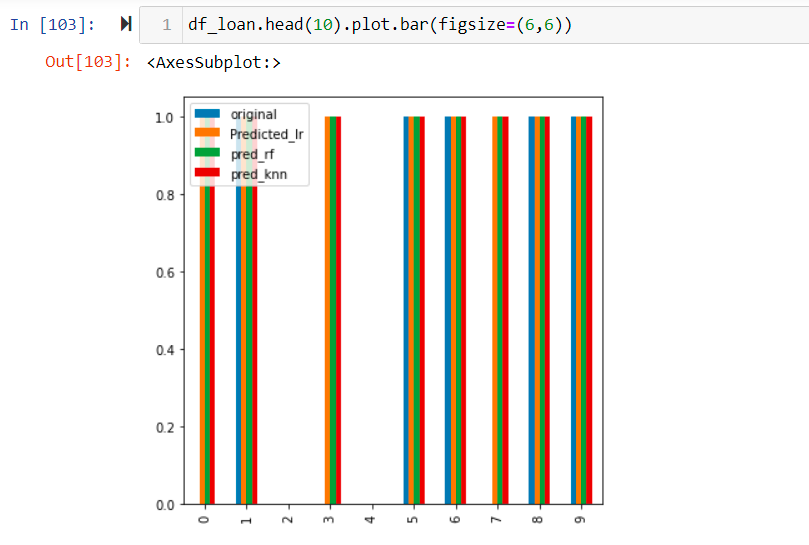


Accuracy: 0.8473282442748091

Now comparing the results of all three models.

Although each one is giving same accuracy (approx.).





**Concluding Remark**

Taking Logistic Regression is the better model as it has highest accuracy and better cross validation mean. Now saving the model.



<https://github.com/himaniuniyal/DataTrained-Project/blob/main/Evaluation%20project/week2/Loan%20Application%20Status%20Prediction.ipynb>

Attached link of jupyter notebook.

Thanks!