1. Problem Definition

This dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc.

Independent Variables:

* Loan\_ID - This refers to the unique identifier of the applicant's affirmed purchases
* Gender - This refers to either of the two main categories (male and female) into which applicants are divided based on their reproductive functions
* Married - This refers to applicant being in a state of matrimony
* Dependents - This refers to persons who depends on the applicants for survival
* Education - This refers to number of years in which applicant received systematic instruction, especially at a school or university
* Self-employed - This refers to applicant working for oneself as a freelancer or the owner of a business rather than for an employer
* Applicant Income - This refers to disposable income available for the applicant's use under State law.
* CoapplicantIncome - This refers to disposable income available for the people that participate in the loan application process alongside the main applicant use under State law.
* Loan Amount - This refers to the amount of money an applicant owes at any given time.
* Loan\_Amount\_Term - This refers to the duration in which the loan is availed to the applicant
* Credit History - This refers to a record of applicant's ability to repay debts and demonstrated responsibility in repaying them.
* Property Area - This refers to the total area within the boundaries of the property as set out in Schedule.
* Loan\_Status - This refers to whether applicant is eligible to be availed the Loan requested.

You must build a model that can predict whether the loan of the applicant will be approved (Loan status) or not based on the details provided in the dataset.

Understanding the problem statement

* The objective is to predict the outcome of a loan application based on the variables provided in the dataset.
* When a client applies for a loan, there are two possible outcomes, it is either they are accepted (Y) or declined (N.)
* There are only two expected outcomes therefore it is easy to identify what kind of problem statement we are dealing with. This is a binary classification problem.

2.  Data Analysis

I firstly imported all necessary libraries before loading the data. I then pulled the first and last five records of the dataset to get a glance of the data rows and columns. The column names did not make sense therefore I renamed them based on the information provided in the project description. Now that it was easy to comprehend the column names, I looked and discovered that a certain column ‘s data type was incorrect which I rectified. I also took the liberty to drop the Loan Id column which is a mere id label unnecessary for data analysis. It has no impact on Loan status, the target variable.

In addition, I checked the shape and information of the data. As part of data analysis I check for duplicates and discovered none. Unique values for each column were checked to ascertain the frequency of each element in a column and the following is an extract of the most frequent elements revealed and visualized by way of a histogram.

* Gender-the most frequent gender is male with 488 occurrences
* Married majority (398) are married
* Dependents-most of the prospects have no dependents
* Education-most people are graduates
* Self-employed- majority are not self employed
* Applicant Income-the most frequent income is 2500
* CoapplicantIncome-majority of co-applicants do not have an income
* Loan Amount-most common loan amount is 120
* Loan\_Amount\_Term-the most common loan term is 360 days
* Credit History-majority have a score of 1
* Property Area-most applicants come from Semi Urban areas
* Loan Status-most applications were approved denoted by Y

After downloading the data, I needed to clean it up so that it was usable for data analysis. I made the following changes as explained below.

One of the columns, ‘Dependents’ had an incorrect data type. It was object instead of float there I change this to float.

I used a heatmap to check null values and discovered quite a handful. There were missing values in the data in both numerical and categorical columns, the former were filled by a statistical value(mean) while the latter filled by the ffill technique. The latter matches the last value of categorical item until all empty values are filled.

A statistical analysis was done on all numerical columns with focus on measures of central tendency and a description for every column was given. The statistical metrics discussed include mean, standard deviation,1st,2nd and 3rd percentiles respectively. I also looked at minimum and maximum values per every numerical column.

Categorical columns were described separately with reference to count, unique values, top and frequency in numerical terms.

A split of numerical and categorical columns was done before data visualization. This is essential because visualization depends on data type so that must clear and separate. I used the following plots for visualization; boxplot, countplot, regplot, distplot and histplot.

Visualization was done for three types of analysis named below each with an example;

* Univariate-single variable analysis

A blue and orange rectangular bars

Description automatically generated

* Bivariate two variables analysed

A graph of a person and person

Description automatically generated

Multivariate-three variables analysed

A graph of blue and orange bars

Description automatically generated

The distplot showed that there was skewness in the data as explained in the statistical analysis of the numerical columns. I tried to use three different types of methods to remove skewness namely, cube root, logp1 and power transform respectively. After attempting to use all the methods skewness was just above the benchmark.

Correlation and multicollinearity analysis was also done. I printed a table of values showing correlation of variables. At face value there was multicollinearity but after further assessment by use of Variance Inflation factors, it was revealed that its magnitude was acceptable to proceed with model building.

It is also important to note that categorical data columns would have to be transformed into numerical so that they could be read by the computer program. For this I used the label encoder method to transform the data into readable format.

The problem statement was binary classification therefore the data had to be balanced by using an oversampling method (SMOTE).There was approximately a 70 30 relationship between Y and N in the loan status column.

After data cleaning I standardised the data by using standard scaler. The purpose of this was to bring all the variables to the same scales without changing the functional properties.

3.EDA Concluding Remarks

* Distribution of data was checked, and I attempted to remove skewness by using three different methods. However, skewness could not be removed entirely
* I visualized outliers in the data by use of a boxplot then removed them via the Z score method. Only 6% of the data was lost in the process
* Null values were filled accordingly
* Correlation and Multicollinearity was checked. Multicollinearity was present but not to the extent that warrants for elimination. Variance inflation factors were below the benchmark of 10.

4.  Pre-processing Pipeline

It is important to note that raw data is mostly not fit for the purpose of analysis. It may contain missing or null values, and other characters. If these discrepancies are included in analysis they have the potential to cause wrong conclusions on the data and subject matter. Considering the above, I followed the following steps in the Data Pre-processing Pipeline.

Data Cleaning:

* Handling missing values-filled numerical null with mean values of the data and used the ffill method in python to fill categorical missing values
* Outliers-these were dealt with by using the Z score method

Feature Scaling:

* Standardizing or normalizing features to bring them to a similar scale without changing functional properties. This will help bring out true relationships during modelling. The method of scaling used is Standardisation.

Feature Transformation

* I checked for skewness in the data as it had been revealed in the statistical analysis and used three logp1, cuberoot and power transform to try and remove it from the data.

Encoding Categorical Data

* Converting categorical variables into numerical representations.
* This has been done by using the label encoder method.
* Feature Selection
* Selecting the most relevant features for modelling.

5.  Building Machine Learning Models

We are dealing with a binary classification problem therefore had to import all the necessary libraries. In building machine learning models, we must try various classification algorithms then select the best model based on a certain criterion. I therefore first computed the following algorithms to get a view of Accuracy score, confusion matrix and classification report for each algorithm. We train different models on our dataset and observe which algorithm works better for our problem.

Support vector machine

Random Forest Classifier,

ExtraTreesClassifier

KNeighborsClassifier

LogisticRegression

DecisionTreeClassifier

GradientBoostingClassifier

AdaBoostClassifier

Bagging Classifier

In principle, I chose these algorithms is because they are appropriate when dealing with classification problems. They work best with categorical variables. In addition, they are user friendly.

It is imperative to note that the accuracy score cannot be simply the only metric that is used to select a model as a best performing one. It needs to be validated. I had to compute Cross validation score for each, and every algorithm then work out the difference between Accuracy score and cross validation score. The model selection criteria were the based on one that had the least difference between Accuracy score and cross validation score. I selected ExtraTreesClassifier which also had an accuracy score of 82.6%.

It is not enough to just accept the model, so I did some hyper parameter tuning to determining the right combination of hyperparameters that maximizes the model performance. Based on the size of the dataset, which is relatively small, I used the Grid Search CV method for parameter tuning. The accuracy score increased to 89.01% after parameter tuning.

To get a more comprehensive measure of model accuracy I then plotted the AUC ROC to demonstrate the predictive power of my model. In this case it was 81%.

After saving the model I did a test run of the model on the test data and found out that the predictions were mostly equal to the original figures in the dataset.

6.Concluding remarks

* In this machine learning project, I built a binary classifier using the nine algorithms to predict the target variable, loan status. I selected Extra trees classifier based on the criteria explained above. I achieved an accuracy of 82.6% by using the Extra trees classifier model which is a very good performance. After hyperparameter tuning, which is primarily meant to maximize model performance, the accuracy increased to 89.01%
* Credit approval is highly dependent on credit history hence we saw its high correlation with Loan Status. In the UK market generally, your ability to be approved for any from of credit depends on your credit history, normally rated on a point-based system with various categories from poor to excellent rating.
* To deal with the issue the issue of loan status imbalance of data, I used an oversampling technique, SMOTE imported from the imblearn module in python.
* ROC AUC of the selected models is biased towards 1. Therefore, we can conclude that our model does a great job in predicting the outcome of a loan application.This project shows the importance and relevance of using machine learning for loan prediction.
* In order to evaluate the model performance for the different classifiers, three classification metrics were used as explained below.
* Classification report - Classification report is another way to evaluate the classification model performance. It displays the precision, recall, f1 and support scores for the model
* Mathematically, precision can be defined as the ratio of TP to (TP + FP).
* Mathematically, recall can be given as the ratio of TP to (TP + FN).
* f1-score is the weighted harmonic mean of precision and recall. The best possible f1-score would be 1.0 and the worst would be 0.0.
* Support is the actual number of occurrences of the class in our dataset.
* Confusion matrix - A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give us a clear picture of classification model performance and the types of errors produced by the model. It gives us a summary of correct and incorrect predictions broken down by each category. The summary is represented in a tabular form.