PROJECT REPORT ON LOAN PREDICTION

INTRODUCTION :-

This is a loan prediction project developed using python. We have used various algorithms like logistic regression, random forest and decision tree and tested their accuracy and performance. This project covers the whole process from problem statement to model development and evaluation:

1. Problem Statement
2. Hypothesis Generation
3. Data Collection
4. Exploratory Data Analysis (EDA)
5. Data Pre-processing
6. Model Development and Evaluation
7. Conclusion.

Problem Statement:

This is a classification problem where we have to predict whether a loan will be approved or not. Specifically, it is a binary classification problem where we have to predict either one of the two classes given i.e. approved (Y) or not approved (N). Another way to frame the problem is to predict whether the loan will likely to default or not, if it is likely to default, then the loan would not be approved, and vice versa. The dependent variable or target variable is the Loan Status, while the rest are independent variable or features. We need to develop a model using the features to predict the target variable.

Hypothesis Generation

Hypothesis Generation is the process of listing out all the possible factors that can affect the outcome i.e. which of the features will have an impact on whether a loan will be approved or not. Some of the hypothesis are:

1.Education - Applicants with higher education level i.e. graduate level should have higher chances of loan approval

2.Income: Applicants with higher income should have more chances of loan approval

3.Loan amount: If the loan amount is less, the chances of loan approval should be high

4.Loan term: Loans with shorter time period should have higher chances of approval

5.Previous credit history: Applicants who have repaid their previous debts should have higher chances of loan approval

6.Monthly instalment amount: If the monthly instalment amount is low, the chances of loan approval should be high

And so on

Some of the hypothesis seem intuitive while others may not. We will try to validate each of these hypothesis based on the dataset.

Data Collection and description

The data has been provided to us as separate train and test data split up. The training set will be used for training the model, i.e. our model will learn from this data. It contains all the independent variables and the target variable. The test set contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data. There are 13 columns of features and 614 rows of records in the training set and 12 columns of features and 367 rows of records in the test set. The dataset variables are summarized as below:

No	Variable	Type	Description
1	Loan_ID	Numerical - Discrete	Unique Loan ID
2	Gender	Categorical - Nominal	Male / Female
3	Married	Categorical - Nominal	Applicant married (Y/N)
4	Dependents	Categorical - Ordinal	Number of dependents (0, 1, 2, 3+)
5	Education	Categorical - Nominal	Applicant Education (Graduate / Under Graduate)
6	Self-employed	Categorical - Nominal	Self employed (Y/N)
7	Applicant Income	Numerical - Continuous	Applicant income
8	Co applicant Income	Numerical - Continuous	Co applicant income
9	Loan Amount	Numerical - Continuous	Loan amount in thousands
10	Loan Amount Term	Numerical - Discrete	Term of loan in months
11	Credit History	Categorical - Nominal	credit history meets guidelines (0, 1)

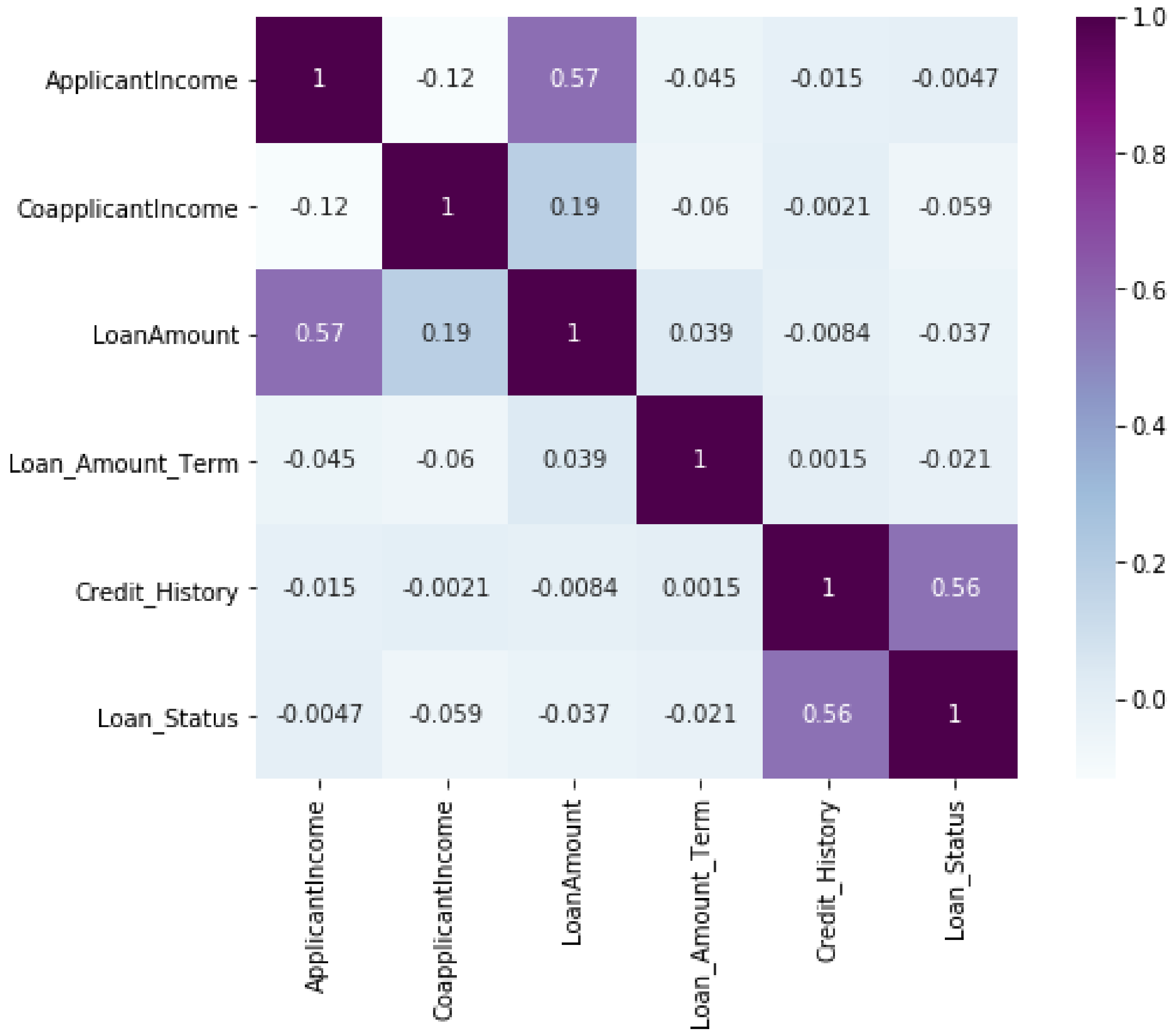
No	Variable	Type	Description
12	Property Area	Categorical - Ordinal	Urban / Semi Urban / Rural
13	Loan Status	Categorical - Nominal	Loan approved (Y/N)

Approach

Our approach to the solution of this project is first doing the exploratory data analysis where we will explore the data in order to gain a better understanding of the features and target variable. Then we will do the data pre-processing which is a data mining technique that involves transforming raw data into an understandable format. Then we will do the model development and evaluation where we first clarify the evaluation metrics for classification problems and do the model building and feature engineering. It is where we use build various models like logistic regression, decision tree, and random forest tree.

Data analysis and visualization:

lets look at the correlation between all the numerical variables. We use the `corr()` to compute pairwise correlation of columns, excluding NA/null values using pearson correlation coefficient. Then we use the heat map to visualize the correlation. Heatmaps visualize data through variations in coloring. The variables with darker color means their correlation is more.



Model Development and Evaluation

Model Building : Part I

We first make our first model to predict the target variable. We start with Logistic Regression which is used for predicting binary outcome.

- Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.
- Logistic regression is an estimation of Logit function. Logit function is simply a log of odds in favor of the event.
- This function creates a s-shaped curve with the probability estimate, which is very similar to the required step wise function

Logistic Regression using stratified k-folds cross validation

To check how robust our model is to unseen data, we use Validation. It is a technique which involves reserving a particular sample of a dataset on which you do not train the model. Later, we test your model on this sample before finalizing it.

Feature Engineering

Based on the domain knowledge, we came up with new features that might affect the target variable. We created the following three new features:

- **Total Income** - As discussed during bivariate analysis we will combine the Applicant Income and Coapplicant Income. If the total income is high, chances of loan approval might also be high.
- **Equated Monthly Installment** - EMI is the monthly amount to be paid by the applicant to repay the loan. Idea behind making this variable is that people who have high EMI's might find it difficult to pay back the loan. We can calculate the EMI by taking the ratio of loan amount with respect to loan amount term.
- **Balance Income** - This is the income left after the EMI has been paid. Idea behind creating this variable is that if this value is high, the chances are high that a person will repay the loan and hence increasing the chances of loan approval.

Model Building : Part II

After creating new features, we can continue the model building process. So we will start with logistic regression model and then move over to more complex models like Random Forest .

We will build the following models in this section.

- Logistic Regression
- Decision Tree
- Random Forest

Logistic Regression

After using the logistic regression model using 5 fold stratified cross validation we got the following results

```
1 of kfold 5
accuracy_score 0.7983870967741935

2 of kfold 5
accuracy_score 0.8225806451612904

3 of kfold 5
accuracy_score 0.7786885245901639

4 of kfold 5
accuracy_score 0.7868852459016393

5 of kfold 5
accuracy_score 0.8278688524590164

Mean validation accuracy:  0.8028820729772608
```

The mean validation accuracy for this model is 0.803

Decision Tree

Decision tree is a type of supervised learning algorithm(having a pre-defined target variable) that is mostly used in classification problems. In this technique, we split the population or sample into two or more homogeneous sets(or sub-populations) based on most significant splitter / differentiator in input variables.

Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable. The results of this model are as follows

```
1 of kfold 5
accuracy_score 0.7258064516129032

2 of kfold 5
accuracy_score 0.7419354838709677

3 of kfold 5
accuracy_score 0.7049180327868853

4 of kfold 5
accuracy_score 0.680327868852459

5 of kfold 5
accuracy_score 0.7049180327868853

Mean validation accuracy:  0.7115811739820201
```

The mean validation accuracy for this model is 0.71, which is lower than Logistic Regression,

Random Forest

- RandomForest is a tree based bootstrapping algorithm wherein a certain no. of weak learners (decision trees) are combined to make a powerful prediction model.
- For every individual learner, a random sample of rows and a few randomly chosen variables are used to build a decision tree model.
- Final prediction can be a function of all the predictions made by the individual learners.
- In case of regression problem, the final prediction can be mean of all the predictions.

There are some parameters worth exploring with the sklearn RandomForestClassifier:

- n_estimators
- max_features

n_estimators = ususally bigger the forest the better, there is small chance of overfitting here. The more estimators you give it, the better it will do. We will use the default value of 10.

max depth of each tree (default none, leading to full tree) - reduction of the maximum depth helps fighting with overfitting. We will limit at 10.

The results of this model are as follows

```
1 of kfold 5
accuracy_score 0.8225806451612904

2 of kfold 5
accuracy_score 0.8145161290322581

3 of kfold 5
accuracy_score 0.7377049180327869

4 of kfold 5
accuracy_score 0.7295081967213115

5 of kfold 5
accuracy_score 0.8114754098360656

Mean validation accuracy:  0.7831570597567425
```

The mean validation accuracy for this model is 0.783.

Comparison of the 3 models

After running the 3 models and having their mean validation accuracy we can see that the logistic regression has the highest accuracy with 0.803 followed by random forest with 0.783 and the decision tree has the least accuracy with 0.71.

Model optimizing using grid search cv

We try to improve the accuracy by tuning the hyperparameters for this model. We will use grid search to get the optimized values of hyper parameters. GridSearch is a way to select the best of a family of hyper parameters, parametrized by a grid of parameters.

We use GridSearchCV in sklearn.model_selection for an exhaustive search over specified parameter values for an estimator. GridSearchCV implements a “fit” and a “score” method. It also implements “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used.

The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid, hence GridSearchCV.

We tune the max_depth and n_estimators parameters. max_depth decides the maximum depth of the tree and n_estimators decides the number of trees that will be used in random forest model.

After optimizing the data the mean validation accuracy has improved from 0.783 to 0.813 in the random forest model.

Conclusion

After trying and testing 3 different algorithms, the best accuracy is achieved by Logistic Regression (0.8028), followed by RandomForest (0.7831)), and Decision Tree performed the worst (0.7115). While new features created via feature engineering helped in predicting the target variable, it did not improve the overall model accuracy much. Compared to using default parameter values, GridSearchCV helped improved the model's mean validation accuracy by providing the optimized values for the model's hyperparameters. On the whole, a logistic regression classifier provides the best result in terms of accuracy for the given dataset, without any feature engineering needed. Because of its simplicity and the fact that it can be implemented relatively easy and quick, Logistic Regression is often a good baseline that data scientists can use to measure the performance of other more complex algorithms. In this case, however, a basic Logistic Regression has already outperformed other more complex algorithms like Random Forest, for the given dataset.

Future work

There are many things that can be tried to improve the models' predictions. We can create and add more variables, try different models with different subset of features and/or rows, etc. Some of the ideas are listed below:

- Combine the applicants with 1,2,3 or more dependents and make a new feature as discussed in the EDA part.
- Make independent vs independent variable visualizations to discover some more patterns.
- Arrive at the EMI using a better formula which may include interest rates as well.
- Try ensemble modeling (combination of different models). More about ensemble techniques can be found at the references.
- Try neural network using Tensorflow or PyTorch

In summary, this project is helpful to me in understanding how a machine learning project is approached and what are the steps one should go through to build a robust model.