**Bank Loan Default Case - Python**

**By Avinash Sajeevan**

**Introduction**

Defaulting on a loan means that you have failed to make sufficient payments for an extended period. Bank will deem a loan in default when you haven't paid the minimum required payment for a certain number of months in a row, as detailed in your loan contract. Loan defaults can happen with any type of loan, whether a mortgage, credit card, or a corporate loan. Defaulting on a loan obligation is serious and can affect the creditworthiness of the individual or company in default. The risk is mainly for the bank and it can include complete or partial loss of principal, loss of interest, and disruption of cash flow.

Gone are the days where the borrowers went scot free after defaulting on bank loan. In the digital age, it is very easy for banks to locate bank defaulter. With the use of machine learning models it’s easy for the bank to predict the risk of default even at the early stage of loan application.

**Problem Statement**

The loan default dataset has 8 variables and 850 records, each record being loan default status for each customer. Each Applicant was rated as “Defaulted” or “Not-Defaulted”. New applicants for loan application can also be evaluated on these 8 predictor variables and classified as a default or non-default based on predictor variables.

**Data Summary**

The data is provided in the csv format, need to be imported using pandas library. Dataset has 8 variables and 850 records.

The data contains the details about customers:

|  |  |
| --- | --- |
| age | Age of each customer |
| ed | Education categories |
| employ | Employment status corresponds to job  status |
| address | Geographic area |
| income | Gross Income of each customer |
| debtinc | Individual’s debt payment to his or her  gross income |
| creddebt | Debt-to-credit ratio |
| othdebt | Any other debts |
| default | Customer defaulted in the past (1= defaulted, 0=Never defaulted) |

**Technological Requirements**

The following list summarizes the tools and packages used in this project

* Anaconda3 – 4.8.3 for 64 bit
* Python - 3.8.3
* Pandas – 1.0.5
* Numpy - 1.18.5
* Streamlit - 0.72
* Pickle4 - 0.0.1
* Sklearn - 0.0
* Matplotlib - 3.2.2
* Seaborn - 0.10.1
* Statsmodels - 0.11.1
* Scipy - 1.5.0
* Imblearn - 0.0

**Methods Summary**

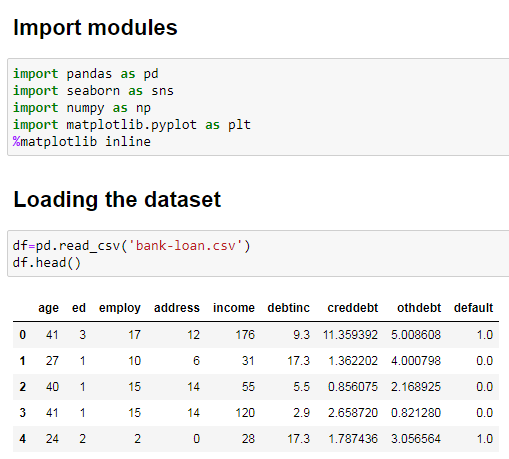
Table shows the list of data pre-processing, analysis, visualization and model building techniques applied to complete the project.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Task Details | Analytical Techniques | Visualization Techniques |
| Data Manipulation &  Preparation | 1. Perform required data manipulation and cleaning.  2. Perform Univariate  and Bivariate analysis | 1. Descriptive statistics and outlier analysis.  2. T-test and VIF check to get important features. | 1. Bar Plot, Heat map, boxplots.  2. Boxplot segmentation |
| Model Building &  Performance Check | Create Model and assess the performance  of the models. | 1. Build Logistic Regression model and used cross validation to find best model.  2. Use hyper parameter to fine tune the model. | Plot ROC-AUC Curve and Precision-Recall curve to show the model performance. |

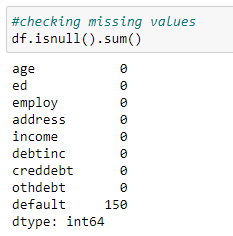
**Model to Predict Default Customers**

**- Data Manipulation & Preparation**

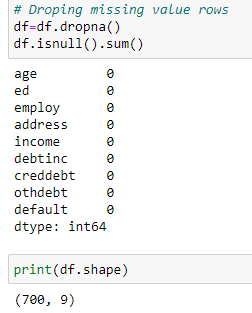
* Started by importing relevant python modules and load the dataset into pandas dataframe.



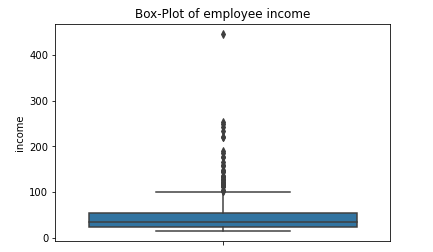
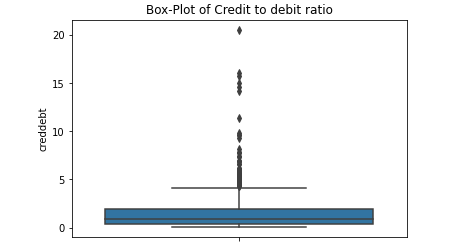
* There are 8 variables and 850 records in the data. Checked for missing values.

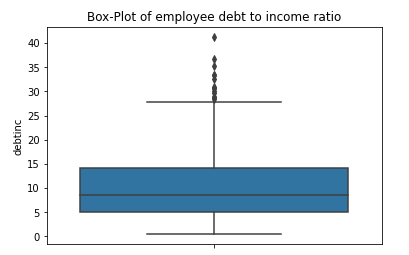
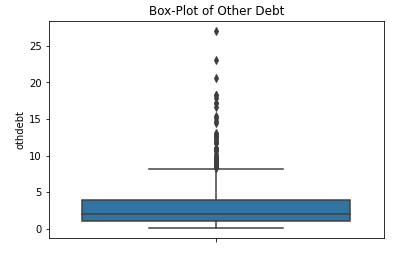


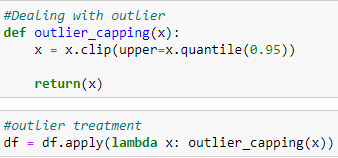
* There are 150 missing values in the dataset and it is found that missing values are in the default column. So dropped those missing values.

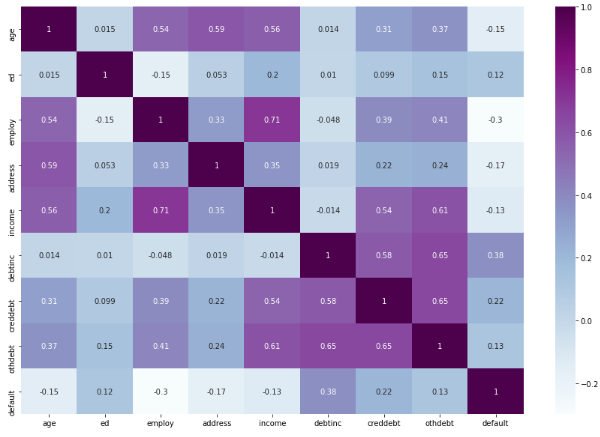


* Got 700 rows of data after dropping the missing values. That will be good for building the model.
* Checked for outliers by using box plots for each variables.

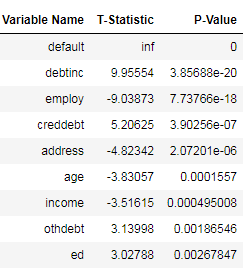
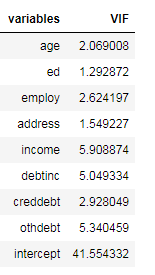


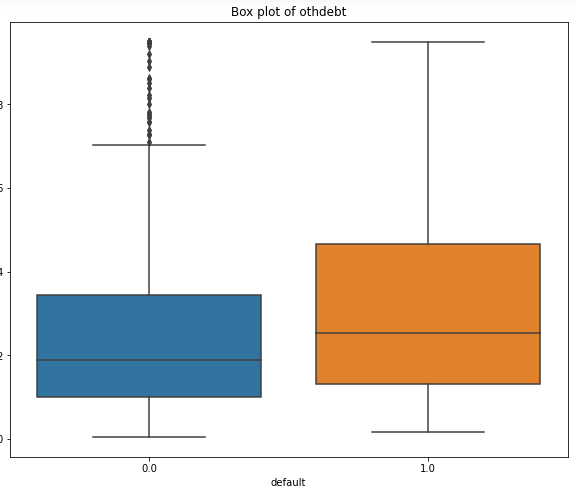
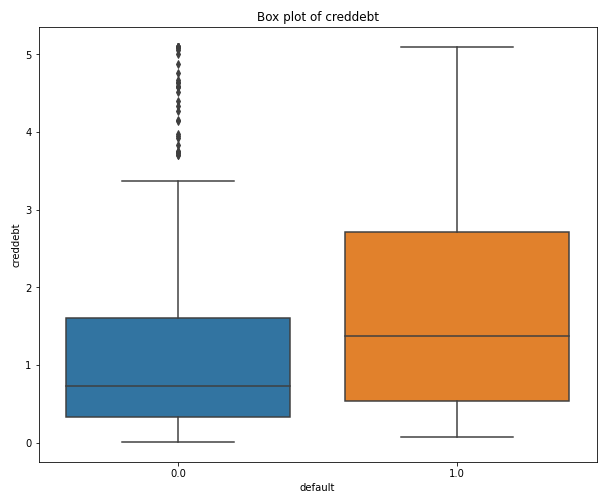
* Found few outliers in the data especially income, debtinc, creddebt, othdebt. So used winsorization method to handle the outliers.  
    
  
* Checked variable importance and multicollinearity using T-test and Variance Inflation Factor (VIF). Also used heat map to visualize correlation.

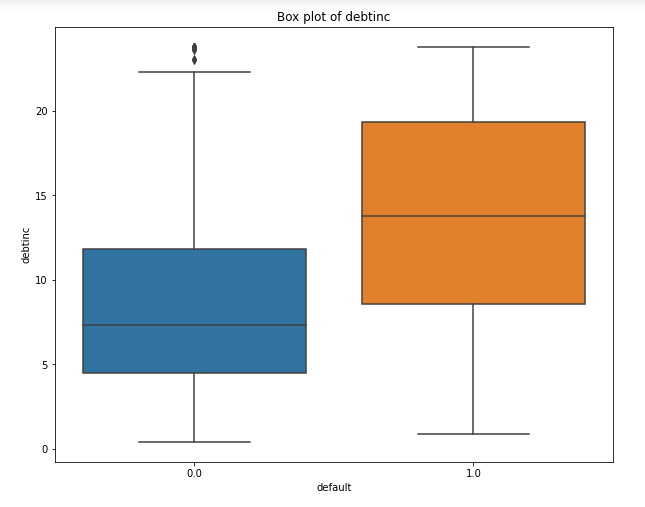
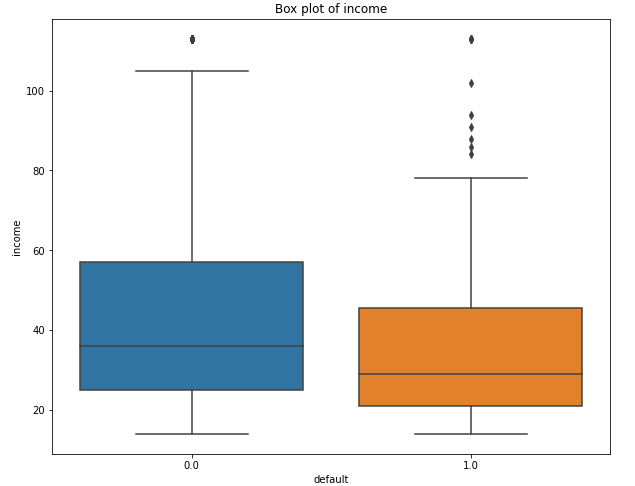


* Performed Independent T Test on each variable with 95% confidence level and found that all the variables are with in-significance level.

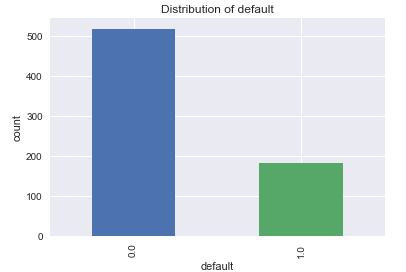
 

* Applied VIF test to find the multicollinearity between the variables, VIF Factor for these variables seems to be with in acceptance levels.
* Checked the effect of independent variables with the target variable using box plot and got few observations.
* Customers with lower values in age, geographic area, employment status, income are likely to default.
* Customers with higher values in individual’s debt payment, debt-to-credit ratio, any other debts are likely to default.

* Checked the distribution of default and non-defaulted customers in the dataset, to check whether the dataset is imbalanced or balanced data set.



* Found dataset is highly imbalanced with default value:

0 517

1. 183

* Decided to use Generating Synthetic Minority Oversampling Technique (SMOTE) in order to avoid biasedness of the estimates and overfitting of the model.

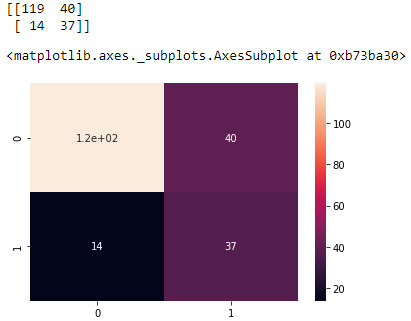
**-** **Model Building & Performance Check**

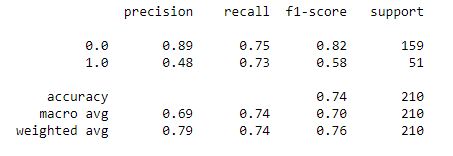
* Splitted data into train and test using train\_test\_split from sklearn.model\_selection and up-sample the default of training data using the SMOTE algorithm.

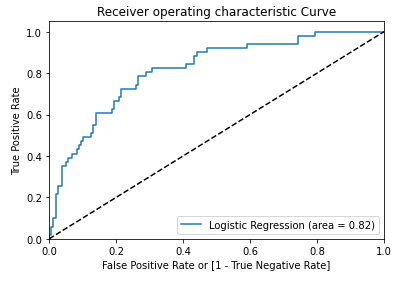


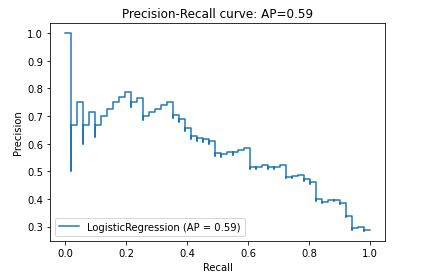
**Logistic Regression**

* Build the logistic regression model with all the variables and default probability for cut-off is taken as 0.5.
* Examined the model using confusion matrix and classification report, the overall accuracy of the default model is around 74% and recall score (ability of the model to find all the positive samples - find all the default customers) is 73%.

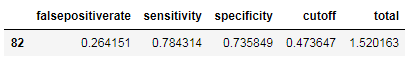




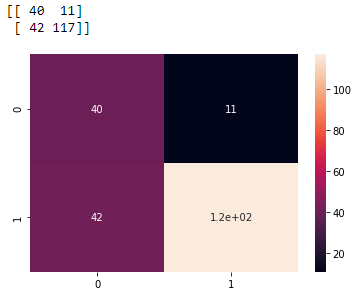
* Calculated Area Under the Curve score and plotted AUC - ROC Curve and Precision - Recall Curve  
    
    
   

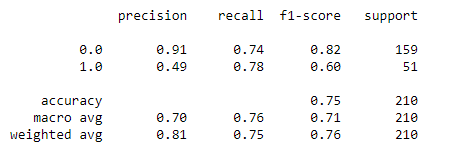


* Recall score of 73% is good. The objective of the model is to identify the customers who will default. In this case we need to find the optimum cut-off value.
* Found the optimum cutoff value where the sensitivity and specificity is maximum: **0.473647**



* Created confusion matrix and classification report using this cut-off instead of sklearn default cutoff.



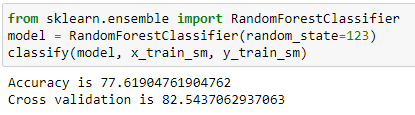


* The accuracy of the model with new cutoff is around 75% and recall score (ability of the model to find all the positive samples - find all the default customers) is 78%.
* Overall accuracy of the model is increased from 74% to 75% by taking optimum cutoff as 0.457116, Model performance i.e. recall score (ability of the model to find all the positive samples - find all the default customers) has increased from 73% to 78% and precision score (ability of model not to label non default customers as default customers) increased from 48% to 49%.

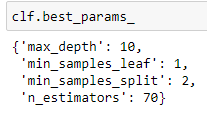
**Checking other models**

* Checked other classification models like Random Forest Classifier, Decision Tree Classifier and SGD Classifier using cross validation. Found Random Forest Classifier has more accuracy and cross validation score of 77.62% and

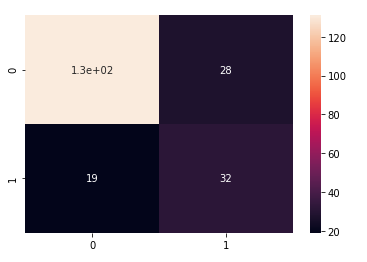
82.54% respectively.

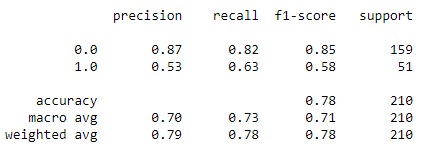


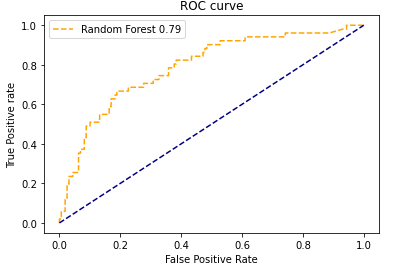
* Tuned the model using hyper parameters and found the best parameters using grid search CV.

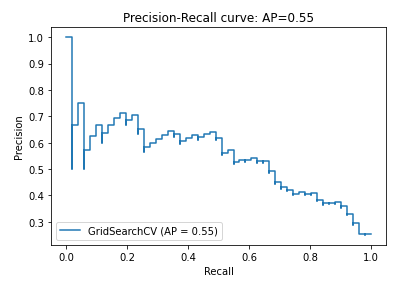


* Examined the model using confusion matrix and classification report, the overall accuracy of the random forest model is around 78% and recall score (ability of the model to find all the positive samples - find all the default customers) is 63%.

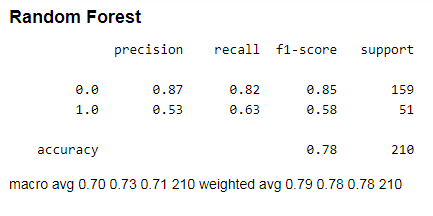


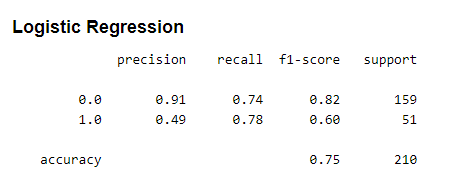


* Calculated Area Under the Curve score and plotted AUC - ROC Curve and Precision - Recall Curve for Random Forest  
    
  

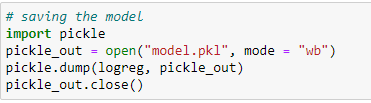


* Compared the accuracy, precision, recall and F1 score of both the models.





* Based on the recall and F1-score (harmonic mean of precision and recall), logistic regression model with F1 score (for positive labels - default customers) of 0.60 is giving better results than random forest model with F1 score of 0.58. So we will use the logistic regression model to predict if the customer default or not.
* Save the model using Pickle.



Model is now saved as ‘model.pkl’

**Deployment**

* Created an app for using the model using streamlit.
* Deployed the app on heroku.  
    
  App deployed: <https://bank-loan-default-python.herokuapp.com/>

Github repository: <https://github.com/avinashsajeevan/Bank-Loan-Default-Prediction--Python>

**Summary**Banks play a big role in the market economies. In other to avoid another global financial crisis, it is important they give close attention to customer’s loan application, risk exposure and probability of default. Here, model building with Logistic regression seems very appropriate. The recall rate of 78% makes it best suitable model for bank loan default prediction. The high recall is better as the banks don’t want to lose money, and would be a good idea to alarm the bank even if there is a slight doubt about defaulter. It is important to mention that due to the relatively small sample size of the data which might not have been able to gain enough statistical and explanatory power. In the future data analysis, I’ll endeavor to apply large sample size.