**Customer Default Identification Report**

**Build and Evaluate Models**

**Credit One**

In the past year, Credit One has seen an increase in the number of customers who have defaulted on loans, which they have secured from our clients or partners. An increase in customer default rates is bad for Credit One. It is our primary task to approve customers for loans in the first place. As the credit scoring service, we could lose business if the problem is not solved right away. Using the historical records for customer payments and defaults in addition to customer demographics, Data Analytics team at Credit One seeks to understand whether or not a customer is likely to default on their credit obligations.

This report is intended for the Data Science team. Our goal is to identify and predict “customer default” using specific features from historical payment, default records and customer demographics. We started our project by defining the business objectives and translating those objectives into data analytics goals. We completed Exploratory Data Analysis, EDA by fully exploring the historical data at hand.

EDA process included data preparation, data cleaning and transformation tasks. We viewed descriptive statistics of the dataset such as minimum, maximum, mean values of each attribute, checked for missing values and checked for duplicate rows. We discretized customer’s age into 6 bins to indicate ages of 20’s, 30’s through 70’s. We replaced ‘Age’ data with an ‘Age Bin’ column. After cleaning and preparing the dataset, we could visualize of the dataset through histograms, line plots, scatter plots, box plots and distribution plots.

We created covariance matrix to observe how the variables in our dataset vary together. The covariance was normalized using standard deviation to a score between -1 and 1, to make its magnitude interpretable. The result is the correlation matrix of the variables in our dataset.

In EDA, the correlation matrix showed us a set of highly-correlated features, where correlation coefficient is greater than 0.90. We eliminated redundant features by removing one of the two highly-correlated features. For example, the feature for customer bill amount, BILL\_AMT1 was highly correlated with BILL\_AMT2, where correlation coefficient was 0.95. We removed BILL\_AMT2 from the feature set. After this work, an EDA Dataset was generated to evaluate models and to predict customer default.

In this current task, we completed Feature Engineering using Recursive Feature Engineering, RFE and subsequent Dimensionality Reduction. We built three Classification models, tuned each model and selected the best model based on their Cross-Validation Score. We then used the best model with the highest CV score for prediction. Prediction accuracy compares the accuracy of predictions to the actual values in the validation (Test) dataset.

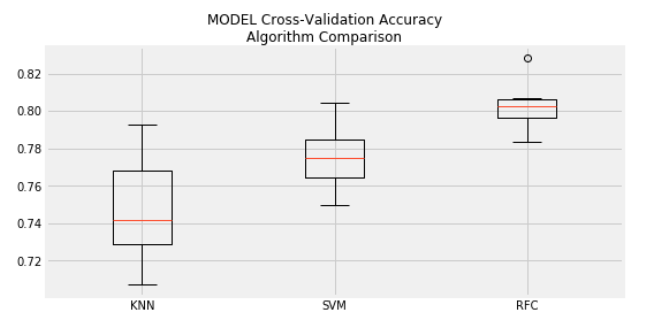
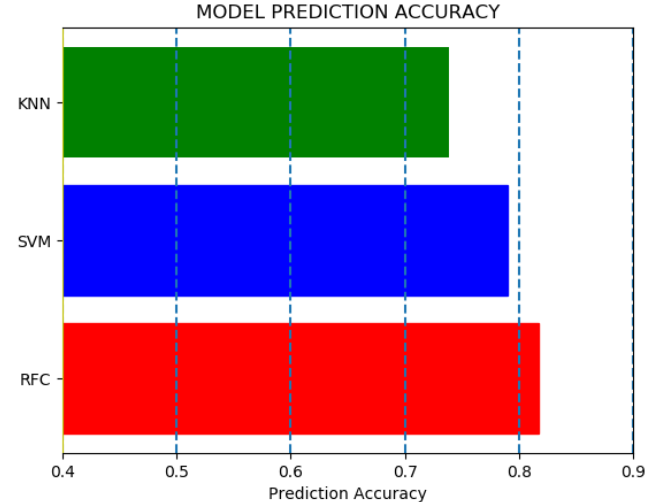
RFE resulted in 11 statistically significant, optimal number of features (independent variables) to predict customers’ behavior (dependent variable) as an indicator of their default in their payment next month. These RFE features consisted of customer’s balance limit, sex, marital status, repayment status in April, May and August, bill amount in April, July and September, amount of payment in May and June. After this Feature Engineering, RFE Dataset was generated to evaluate models and to predict customer default.

We used the EDA dataset and RFE dataset in 3 different models using machine learning algorithms:

* k-nearest neighbors, kNN classifier
* Support Vector Machine SVM classification model SVC
* Random Forest Classification using RFC.

Cross Validation, CV estimates the performance of a machine learning algorithm by splitting the dataset into n parts where n might be equal to 5 or 10 splits. Each split of the dataset is called a fold. Each algorithm that we selected is trained on (n -1) folds with one fold held back. The algorithm is then tested on the held-back fold. This procedure is repeated so that each fold of the dataset is given a chance to be held back as the test set. CV provides n different performance scores which are summarized as mean CV-score and a standard deviation. This CV-score is a reliable estimate of the performance of the algorithm on new data because the algorithm is trained and evaluated multiple times on different set of data.

We used kNN, SVC and RFC *out of the box* on the *EDA dataset* to see the CV scores.

We tuned all of the models by varying the most significant 2 parameters for each algorithm. These parameters for each algorithm are,

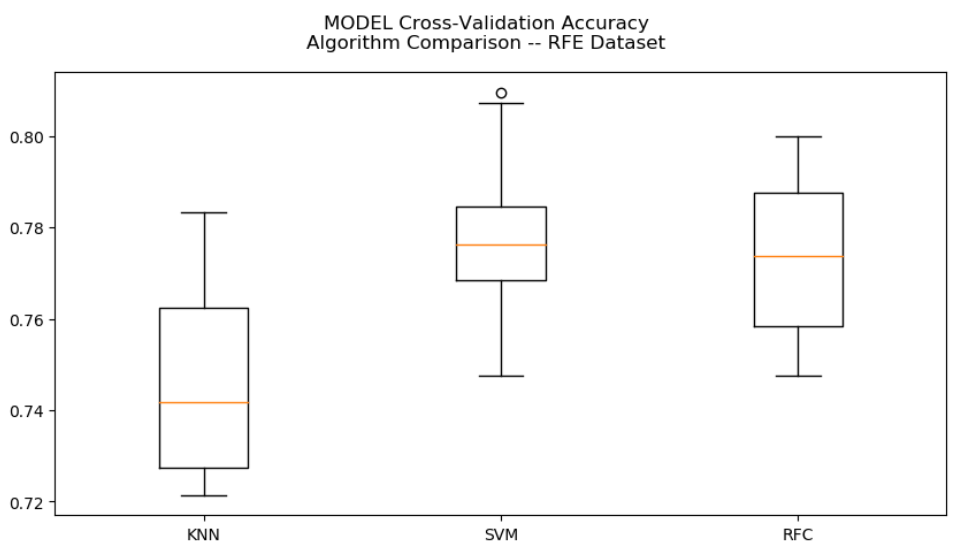
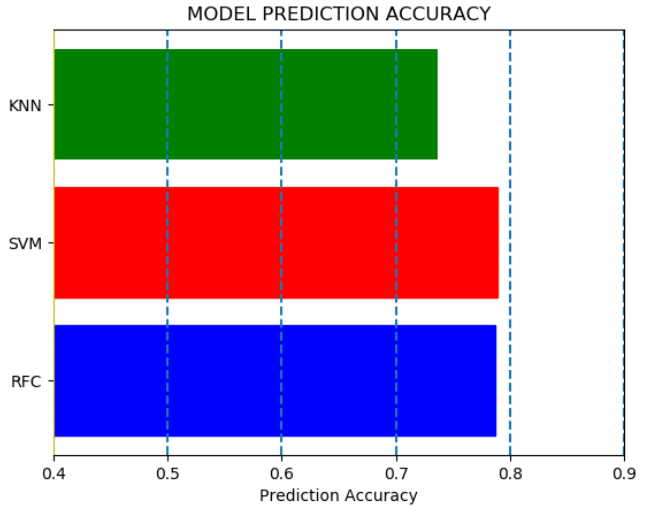
* kNN : 1. K-value specified as n neighbors from 1 to 500 2. weights specified as uniform or distance
* SVM : 1. C-values ranging from 0.1 to 2.0 2. Kernel specified as linear, poly, rbf, sigmoid
* RFC : 1. N estimators specified as n values from 2 to 500 2. Max\_feature specified as auto (sqrt) or log2

Each algorithm has many other options for parameter specifications. Default values are accepted for those that are not explicitly entered in our python script.

In order to find the maximum CV-score per model, the models are tuned using a wide range of values specified for the first parameter, along all options specified for the second set of parameters. For example, SCV model ran 40 times with the EDA dataset using 10 different C-values and 4 different kernel types. RFC ran 24 times with associated parameters to get the best CV score and so forth. We recorded the mean CV score for each model.

We selected the best model with the highest “mean CV score”.

We used kNN, SVC and RFC *out of the box* on the *RFE dataset* to see the CV scores

In order to find the maximum CV-score per model, we tuned each model using the RFE dataset using the same procedure explained for the EDA dataset.

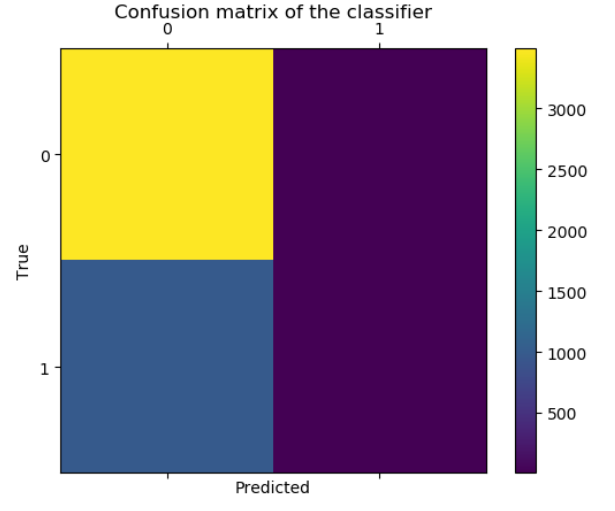
Out of box model CV scores as well as CV scores from tuned models are provided in the table below.



We observe that top 5 out of the 6 models with better CV scores had the algorithms tuned for best parameters.

SVM Classification algorithm tuned for C value = 0.9 and kernel = ‘rbf’ was selected as the best algorithm to predict customer default. Prediction Accuracy: 78%. Confusion Matrix: [[3495 5]

[ 991 9]]



0 = No default and 1 = Default

We cannot control customer spending habits, nor can we change customers’ spending or payment habits. However, we can apply predictive analytics using the customer demographics and historical payment & default data to predict a customer’s likelihood to make payments or default their credit with reasonable accuracy.

The process we used in this task can be applied to existing and new customer base to determine if a particular person should be extended credit. Based on our findings, we can reduce loan limits to existing customers that are at risk of default. We can decide credit worthiness of new customers using the same features outlined in the EDA dataset.