**Customer Churn Analysis**

Weekly Report

Ahmedabad University

4rier Series

Prof. Mehul Raval

CSE523 - Machine Learning

**Selecting the Features of Models**

We implemented a function that covers all the dataset's features from the correlation matrix, and for each location, it checks and compares the threshold value. If the correlation on the particular location exceeds the threshold value, we simply put them in a set. The set keeps all distinct values, and thus we get all columns or features with higher correlation values.

The remaining features we get after removing these columns are the selected features that we have discarded in the test and train dataset.

However, we have done some manual checking by changing the value of the threshold.

We have got similar results for the threshold value 0.6 when considering all features. For SVM and logistic regression, there was a small amount of degradation. Surprisingly, the Naive Bayes classification got around ten percent more accuracy than earlier.

Statistics when all 31 features, when taken into consideration:

| Factors | Logistic Regression | SVM | Naive Bayes |
| --- | --- | --- | --- |
| Accuracy | 81.27962085308057 % | 80.52132701421802 % | 65.260663507109 % |
| Precision | 66.04166666666667 % | 66.42857142857143 % | 41.91564147627416 % |
| Recall | 57.74134790528234 % | 50.81967213114754 % | 86.88524590163934 % |
| F1 score | 80.85436799018186 % | 79.61121682882954 % | 67.2861386096018 % |

Statistics when only 12 features are taken into consideration:

| factors | Logistic Regression | SVM | Naive Bayes |
| --- | --- | --- | --- |
| Accuracy | 80.4739336492891 % | 79.81042654028437 % | 76.49289099526067 % |
| Precision | 65.46275395033861 % | 66.2269129287599 % | 54.04580152671755 % |
| Recall | 52.82331511839708 % | 45.71948998178507 % | 64.48087431693989 % |
| F1 score | 79.75130157938099 % | 78.48246426136785 % | 77.11458325458942 % |

We are still figuring out another better alternative to do the same work automatically.

**Explored Decision Tree Classifier**

Decision Tree Classifier works by creating a tree-like model of decisions and their possible consequences. Each node in the tree represents a decision based on a particular feature or attribute, and each branch represents the outcome of that decision. The model's outcome is determined by following the path from the root node to the leaf node.

Steps involved are:

* Tree construction
* Pruning: to avoid overfitting
* Prediction

It can also be used for feature selection and provides interpretable results. However, it is prone to overfitting and can be sensitive to small changes in the data. To avoid these issues, a random forest classifier is used.

**Explored Random Forest Classifier**

Random Forest Classifier is a popular machine-learning algorithm for classification problems. An ensemble learning method combines multiple decision trees to create a more accurate and robust model.

The basic idea behind Random Forest is to build multiple decision trees using randomly selected subsets of the training data and randomly selected subsets of the features.

Steps used in classification using random forest are:

* Build multiple decision trees by selecting random subsets of data.
* Randomly select features to be used for training
* Train each decision tree
* Prediction of the class label by aggregation of the results of each decision tree.
* Final evaluation of the performance

For evaluation of the model, accuracy, precision, recall, F1-score, confusion matrix, and AUC-ROC curve are some of the parameters that can be used.