Capstone Project 1: Machine Learning

Data Pre-processing:

As we move from EDA to machine learning, some pre-processing needs to be done in order to build our models.

We have 18 categorical and 3 numerical features. Since scikit-learn does not accept categorical features, we need to encode them. We do this by 'Label Encoding' the categorical features. In the next step we standardize the 3 numerical columns.

Now we look at the correlation matrix again to see any new correlation since all the columns are numeric now after label encoding. We can see the high correlation of tenure - total charges and monthly charges - total charges, same like we saw in EDA.

We now visualize the principal components using the principal component analysis. The 2 principal components represent atleast 50% variance in the data. We plot these 2 principal components.

Model Building:

First we define some functions to tune the parameters and see the performance of the model. We now split the data into training and test sets on a 80:20 ratio.

Baseline Model-

The next step we build a baseline model. Here we see the performance of the models using default parameters. The classifiers we choose to train and test our data are:

1. Logistic Regression
2. K-Nearest Neighbors
3. Support Vector Machine
4. Decision Tree
5. Random Forest
6. Gaussian Naive Bayes

The accuracy of the classifiers with default parameters are:

Classifier	Accuracy Score
Logistic Regression	79%
K-Nearest Neighbors	76%
Support Vector Machine	79%
Decision Tree	71%
Random Forest	77%
Gaussian Naive Bayes	73%

Logistic Regression and SVM performed the best with an accuracy of 79% and Decision Tree performed the worst with an accuracy of 70%.

Parameter Tuning-

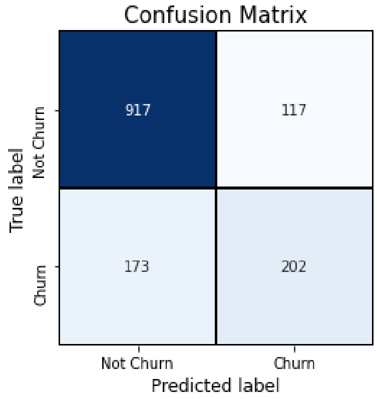
Now we try to improve the performance of the models with parameter tuning. We will use both Grid Search CV and Randomized Search CV.

1. Logistic Regression:-

The following parameter was tuned:

- 'C': [0.001, 0.01, 0.1, 1, 10, 100]

The best value obtained was C = 0.1.
We got a slight increase of 0.28% in accuracy.
Accuracy = 79%
Confusion matrix =



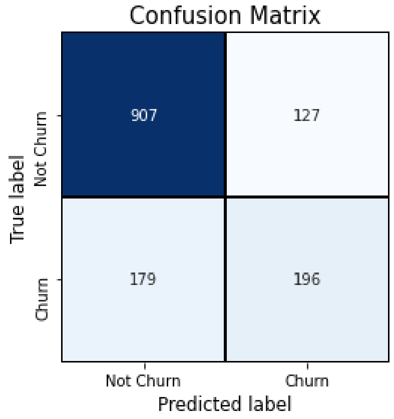
Precision = 63%
Recall = 54%
f1-score = 58%
AUC = 82%
Feature importance = Monthly charge is highly positively related with churn whereas tenure, phone service & contract are highly negatively related to churn.

2. K-Nearest Neighbors:-

The following parameter was tuned:

- 'n_neighbors': (1 to 49)

The best value obtained was n_neighbors = 38.
We get an increase of 2.6% in accuracy.
Accuracy = 78%
Confusion matrix =



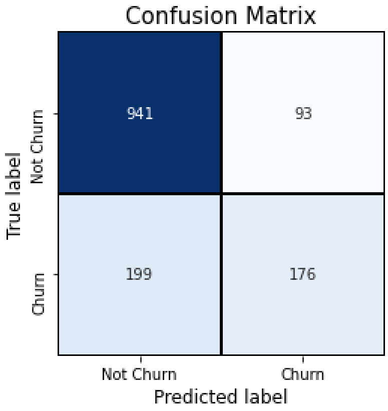
Precision = 61%
Recall = 52%
f1-score = 56%
AUC = 81%

3. **<u>Support vector Machine</u>:-**

The following parameters were tuned:

- 'C': (1 to 19)
- 'kernel': ['linear', 'poly', 'rbf', 'sigmoid']
- gamma': ['scale', 'auto']

The best values obtained were C = 1, kernel = 'rbf', gamma = 'scale'.
Accuracy remains the same.
Accuracy = 79%
Confusion matrix =



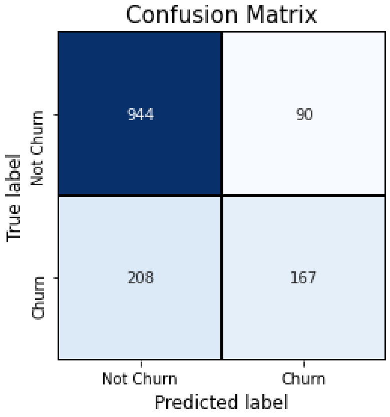
Precision = 65%
Recall = 47%
f1-score = 55%
AUC = 78%

4. **<u>Decision Tree</u>:-**

The following parameters were tuned:

- 'max_depth': (2 to 6)
- 'max_features': (1 to 20)
- 'min_samples_leaf': (1 to 19)
- 'criterion': ['gini', 'entropy']

The best values obtained were min_samples_leaf = 4, max_features = 13, max_depth = 5, criterion = 'entropy', random_state = 10.
We get a huge increase of 7.24% in accuracy.
Accuracy = 79%
Confusion matrix =



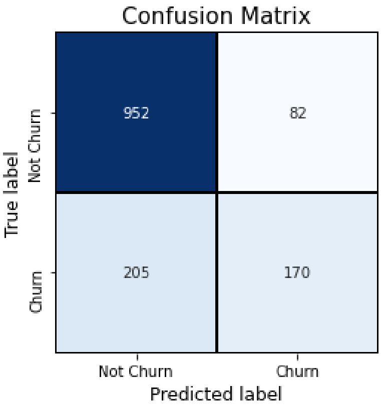
Precision = 65%
Recall = 45%
f1-score = 53%
AUC = 81%
Feature importance = We get tech support and monthly charges highly related to churn.

5. **Random Forest**:-

The following parameters were tuned:

- 'max_depth': (2 to 6)
- 'max_features': ['sqrt']
- 'min_samples_leaf': (1 to 19)

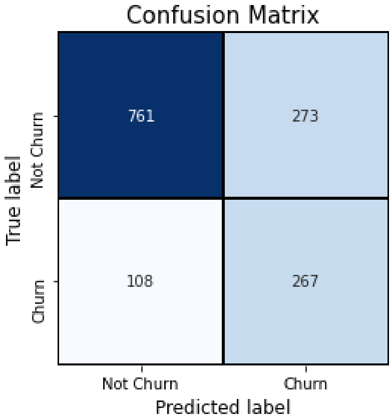
The best values obtained were min_samples_leaf = 13, max_features = 'sqrt', max_depth = 6.
We get an increase of nearly 2.4% in accuracy.
Accuracy = 79%
Confusion matrix =



Precision = 67%
Recall = 45%
f1-score = 54%
AUC = 83%
Feature importance = We get contract and tenure highly related to churn.

6. **Gaussian Naive Bayes**:-

No parameters were tuned here.
So the accuracy remains the same.
Accuracy = 73%
Confusion matrix =



Precision = 49%
Recall = 71%
f1-score = 58%
AUC = 80%

The performance of the classifiers after tuning the parameters are:

Classifier	Accuracy Score
Logistic Regression	79% (0.23% ↑)
K-Nearest Neighbors	78% (2.6% ↑)
Support Vector Machine	79%
Decision Tree	79% (7.24% ↑)
Random Forest	79% (2.4% ↑)
Gaussian Naive Bayes	73%

After tuning the parameters we saw some good improvements in the performance of the models. Decision Tree improved the most by an increase of 7.24% in accuracy. Logistic Regression, K-Nearest Neighbors and Random Forest improved their performances too. Support Vector Machine and Gaussian Naive Bayes remained the same. The Gaussian Naive Bayes model is the lowest performing model with an accuracy of 73%.

All the models are performing equally (except the Gaussian Naive Bayes model) with an accuracy of 79% (approx).

Voting Ensemble-

We now use the Voting Classifier to combine all the above mentioned machine learning classifiers. The models are pitted against each other and selected upon best performance by voting. Such a classifier can be useful for a set of equally well performing model in order to balance out their individual weaknesses.

The voting ensemble gives an accuracy score of 80%.