Assignment 1 – Supervised Learning

Repository for data and code -

# Problem Statement

In the telecom industry, customers can choose from multiple service providers and actively switch from one operator to another. In this highly competitive market, the telecommunications industry experiences an average of 15-25% annual churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition. For many incumbent operators, retaining highly profitable customers is the number one business goal.

To reduce customer churn, telecom companies need to predict which customers are at high risk of churn.

In this project, customer-level data of a leading telecom firm is analyzed, built predictive models to identify customers at high risk of churn, and identify the main indicators of churn. The recommendations are given, and different models compared using confusion matrix, and the best model is selected. The feature engineering is also done using different mathematical procedures.

# Understanding and defining churn

Usage-based churn: Customers who have not done any usage, either incoming or outgoing - in terms of calls, internet etc. over a period of time.

The project is about predicting the potential churn of customers and identifying the high risk customers who has higher probability of leaving the telecom service provider.

# Modelling and the procedure to select the best model

The predictive model serves the following two purposes -

It will be used to predict whether a high-value customer will churn or not, in near future (i.e. churn phase). By knowing this, the company can take action steps such as providing special plans, discounts on recharge etc.

It will be used to identify important variables that are strong predictors of churn. These variables may also indicate why customers choose to switch to other networks.

Here there are large numbers of attributes, and thus dimensionality reduction technique such as PCA is used and then the classification model is built. Different models are trained and are tuned using hyperparameters.

And then the models are evaluated using appropriate evaluation metrics. Note that it is more important to identify churners than the non-churners accurately - choose an appropriate evaluation metric that reflects this business goal.

Finally, choose a model based on some evaluation metric.

The above model will only be able to achieve one of the two goals - to predict customers who will churn. We can’t use the above model to identify the important features for churn. That’s because PCA usually creates components that are not easy to interpret.

Therefore, another model is built with the main objective of identifying important predictor attributes that help the business understand indicators of churn. A good choice to identify important variables is a logistic regression model or a model from the tree family. In the case of logistic regression, make sure to handle multi-collinearity.

After identifying important predictors, display them visually - you can use plots, summary tables etc. - whatever you think best conveys the importance of features.

Finally, recommend strategies to manage customer churn based on your observations.

# Test and Train split

Test size is taken as 25% of the total data set. 4 random states are used.

A screen shot of a computer

Description automatically generated

# Aggregating the categorical features.

The categorical columns are aggregated to reduce the features.

# PCA on the remaining features.

Principle Component Aanalysis is applied on the remaining features, to reduce the number of features further, retaining as much information as possible in the data. This helps in handling the curse of dimensionality. The correlated features are reduced and only the independent ones are kept.

A screenshot of a computer program

Description automatically generated

# Logistic regression

Logistic regression is applied to classify the data on churning and non churning customers. The score of the model is computed as 80%.

A screenshot of a computer program

Description automatically generated

# Confusion matrix for the logistic regression on test data

A screen shot of a computer

Description automatically generated

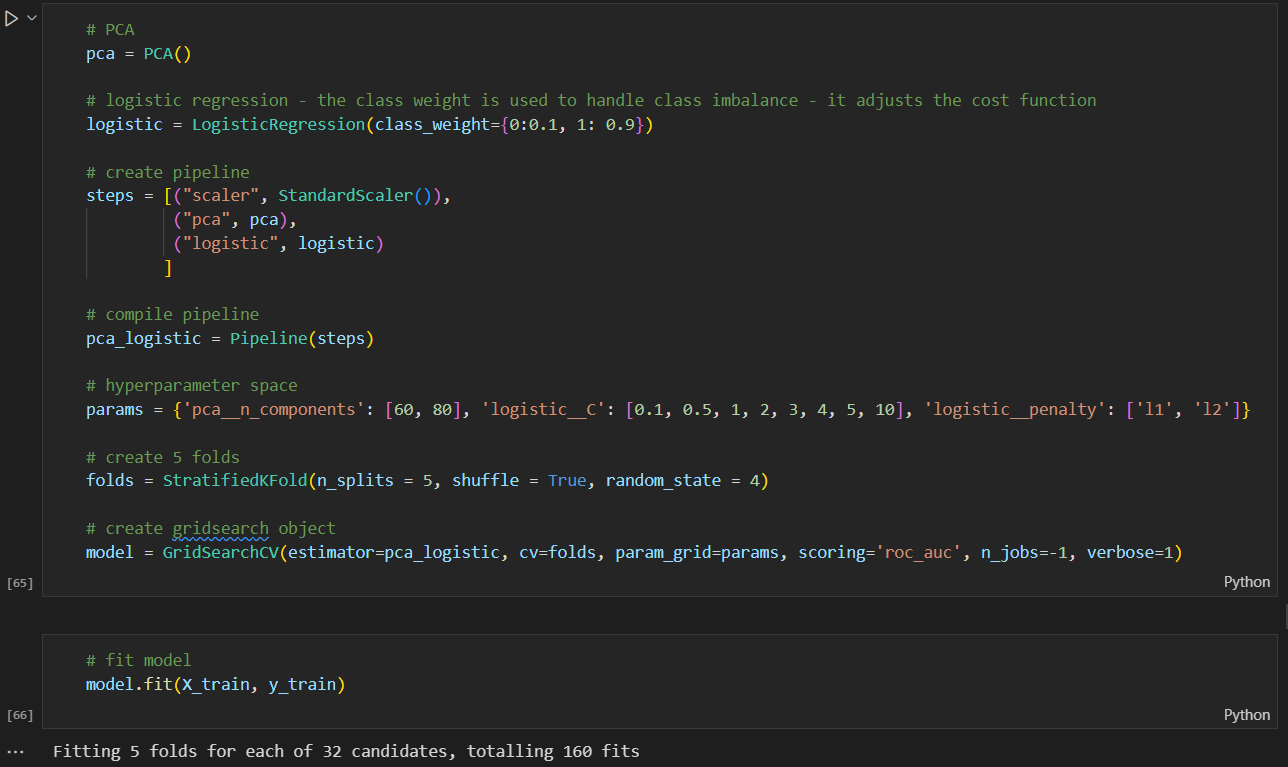
# Hyperparameter tuning and logistic regression

A diagram of a model

Description automatically generated

Grid search cross validation method is used to do the hyperparameter tuning. A 5 fold search is done to do the hyperparameter tuning. Following are the steps to hyper parameter tuning as depicted in the figure above –

* Select the right type of model.
* Review the list of parameters of the model and build the HP space
* Finding the methods for searching the hyperparameter space
* Applying the cross-validation scheme approach
* Assess the model score to evaluate the model



# Model performance post hyperparameter tuning for logistic regression.

A screen shot of a computer program

Description automatically generated

As shown above the performance of the model has improved, and the AUC is 90%.

|  |  |  |
| --- | --- | --- |
| Logistic regression | Before hyperparameter tuning | Post-hyperparameter tuning |
| Sensitivity | 85% | **83%** |
| Specificity | 80% | **84%** |
| AUC | 89% | **90%** |
| Precision | 27% | **31%** |
| Recall | 85% | **83%** |
| Accuracy | 80% | **84%** |
| F1 score | 41% | **45%** |
| AUC | 89% | **90%** |

The scorecard of the model performance of the logistic regression in the above table shows that the performance of the model has drastically improved after hyper parameter tuning.

# Decision tree – random forest after hyper parameter tuning

A screenshot of a computer program

Description automatically generated

# Model performance – random forest

A screen shot of a computer program

Description automatically generated

# Comparison of Classification Model Recommendation

Hence the best model is PCA along with logistic regression, as following are the model performance metrics after doing hyperparameter tuning –

|  |  |  |
| --- | --- | --- |
| Metrics after hyperparam tuning | Logistic Regression | Radom forest decision trees |
| Sensitivity | 83% | **50%** |
| Specificity | 84% | **98%** |
| AUC | 90% | **93%** |
| Precision | 31% | **73%** |
| Recall | 83% | **50%** |
| Accuracy | 84% | **94%** |
| F1 score | 45% | **60%** |
| AUC score | 90% | **93%** |

The Random Forest fares better than Logistic regression on all fronts except Recall and Sensitivity. Based on the scenarios at hand, the model can be chosen. In all cases, where the sensitivity of the data is not crucial, Random forest appears good. But the recall of the Random forest is way lesser than that of the Logistic regression. Recall is the ability to predict correctly when an event is classified as positive. This scenario is important in this business context of predicting the churn of customers when they will churn out. So, for this business scenario**, the recommended model needs to be the Logistic regression model after hyperparameter tuning.**