Customer Churn Prediction for VeriTel

Group Number: 10

Group Members:

- Anant Krishna
- Bharath Aynampudi
- Chinmay Majee
- Debapratim Ghosh
- Kunal Chandra
- Santosh Srivastava
- Satish Chilloji
- Thej Pammi

Contents:

Project Background :

VeriTel is the second largest telecom provider in the world with operations in over 15 countries directly and in other 21 countries with a partner. The company is a leader in providing telecom services in both B2B and B2C space. Although the company has been doing well in last 3-4 years, the revenue of the company seems to have been almost plateau like and stagnating. The CEO hypothesizes that this stagnation is mostly due to a large number of customers churning out of their subscriptions. Given the fact that the telecommunications industry experiences an average of 30-35 percent annual churn rate and it costs 5-10 times more to recruit a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition. In order to manage churn reduction, not only do we need to predict which customers are at high risk of churn, but also we need to know how soon these high-risk customers will churn. Therefore the company can optimize their marketing intervention resources to prevent as many customers as possible from churning. The CEO wants to deploy retention strategies in synchronizing programs and processes to keep customers longer by providing them with tailored products and services. With retention strategies in place, the CEO wants to include churn reduction as one of their business goals.

Data Collection:

We have collected the data published in Kaggle (https://www.kaggle.com/abhinav89/telecom-customer). Kaggle is the world's largest data science community with powerful tools and resources to help you achieve your data science goals and it supports a variety of dataset publication formats in non-proprietary format

Data Explanation:

In [1]:

```
import pandas as pd
import numpy as np
telecom=pd.read_csv('Telecom_customer_churn.csv')
```

The data set has 100,000 rows and 100 Columns

In [2]:

```
telecom.shape
```

Out[2]:

```
(100000, 100)
```

Glimpse of the churn data

In [8]:

```
pd.options.display.max_columns=1000
telecom.head()
```

Out[8]:

	rev_Mean	mou_Mean	totmrc_Mean	da_Mean	ovrmou_Mean	ovrrev_Mean	vceovr_Mean
0	23.9975	219.25	22.500	0.2475	0.00	0.0	0.0
1	57.4925	482.75	37.425	0.2475	22.75	9.1	9.1
2	16.9900	10.25	16.990	0.0000	0.00	0.0	0.0
3	38.0000	7.50	38.000	0.0000	0.00	0.0	0.0
4	55.2300	570.50	71.980	0.0000	0.00	0.0	0.0

Number of each data type of column

In [10]:

```
telecom.dtypes.value_counts()
```

Out[10]:

```
float64    69
object     21
int64      10
dtype: int64
```

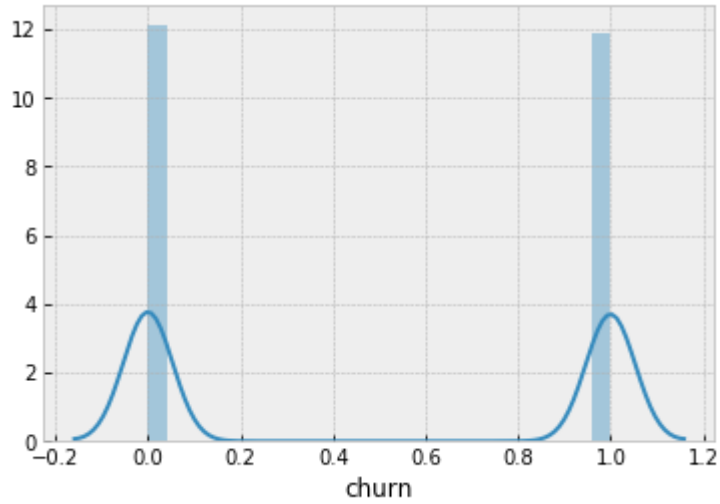
Target column churn distribution

In [7]:

```
import matplotlib.pyplot as plt
from matplotlib import pyplot
import seaborn as sns
plt.style.use('bmh')
sns.distplot(telecom['churn'])
telecom['churn'].value_counts()
```

Out[7]:

```
0    50438
1    49562
Name: churn, dtype: int64
```



Model Approach:

We will be following the tradition ML modeling approach to find the **who are the customers that are going to churn**? The data is divided into 70-30 split and the model is trained on the 70% of the data and predicts on 30% of the data.

As we have the limited data option we will performing set of feature engineering inorder to boost the accuracy.

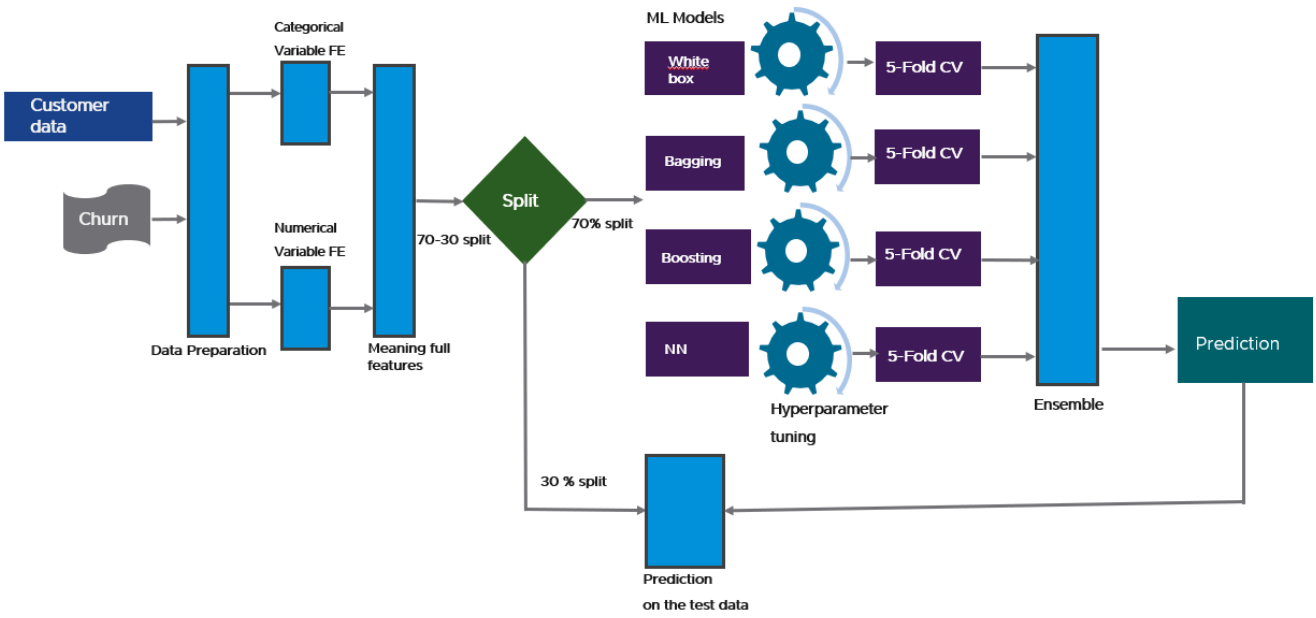
Feature Engineering on numerical features:

- Scaling
- Transformations
- Outlier cliping
- Feature Interactions
- Approximation of Addition and Multiplication.

Feature Engineering on categorical features:

- Convert to numeric
  - Label encoding
  - Frequency encoding
  - One-Hot encoding
  - Weight of evidence
- Interaction features: min,max,avg group by

The workflow of the model moves from left to right. We will be trainig the data on variety of models performing the hyperparameter tuning. Before the ensemble we will be doing the 5-Fold cross validation to overcome the issue like overfitting and underfitting the model and provide the predictions.



Explainable Machine Learning

In the Churn model once we have know who all the customer are going to Churn the next question that the end user look for **Why this customers are going to churn?** over the period of time this will be the important to know root cause for the churn.

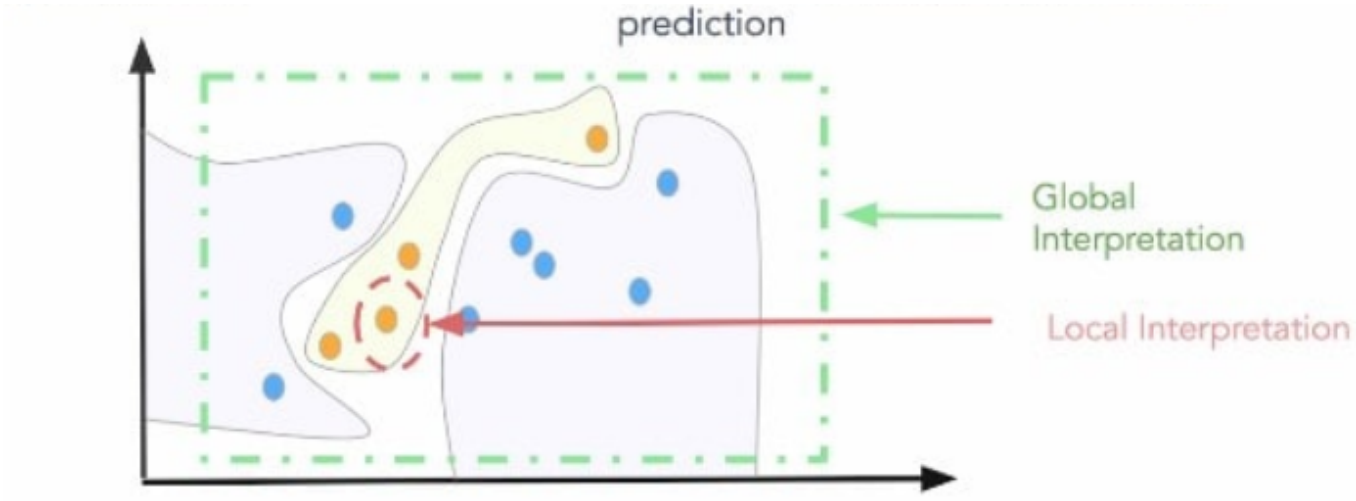
In machine learning complex model has big issue with transparency, we don't have any strong prove why model give that prediction and which feature are impacting the model prediction, which features are strongly contributing, and which are negative contribution for model prediction. By feature importance graph we can see which features importance by passing complete training and test dataset, but for single row of features or for any given instance it is very difficult to understand why and how model predict output.

We will be looking on the following 3 Explainable technique.

- Lime
- Shap
- eli5

Lime

LIME is a local surrogated model which normally use Linear regression or decision tree model to explain the prediction at local boundary. In advance You need to select the K number which is kernel weight and number of features the lower K value easier it is to interpret the model, and the Higher K value produces models with higher fidelity. LIME currently uses an exponential smoothing kernel to define the neighbourhood. A smoothing kernel is a function that takes two data instances and returns a proximity measure. The kernel width determines how large the neighbourhood is: A small kernel width means that an instance must be very close to influence the local model, a larger kernel width means that instances that are farther away also influence the model.



ummarizing Global and Local Interpretation (Source: DataScience.com)

Shap

SHAP goal is to explain the prediction of an given instance x by computing the contribution of each feature to the prediction. The feature values of a data instance act as players in a coalitional game theory. SHAP prediction output is a fair distribution of all the feature Shapley values. Shapely value is actually distribution, it's a average of model contribution made by each player(features) over all permutation of player(features).The baseline for Shapley values is the average of all predictions. In the plot, each Shapley value is an arrow that pushes to increase (positive value) or decrease (negative value) the prediction.

eli5

ELI5 help to debug machine learning classifier and explain their top prediction via a easy to understand and good visualize way. However, it doesn't support true model-agnostic interpretations and support for models are mostly limited to tree-based and other parametric\linear models. When you want to predict ELI5 does this by showing weights for each feature depicting how influential it might have been in contributing to the final prediction decision across all trees. ELI5 provides an independent implementation of this algorithm for XGBoost and most scikit-learn tree ensembles which is definitely on the path towards model-agnostic interpretation but not purely model-agnostic like LIME.

Actual: 0 Predicted : 0

y=No (probability **0.718**, score **-0.936**) top features

Contribution[?]	Feature
+1.930	thal
+1.466	sex
+1.327	trestbps
+0.918	ca
+0.851	exang
+0.533	chol
+0.081	oldpeak
-0.146	<BIAS>
-1.064	age
-1.858	slope
-3.102	thalach

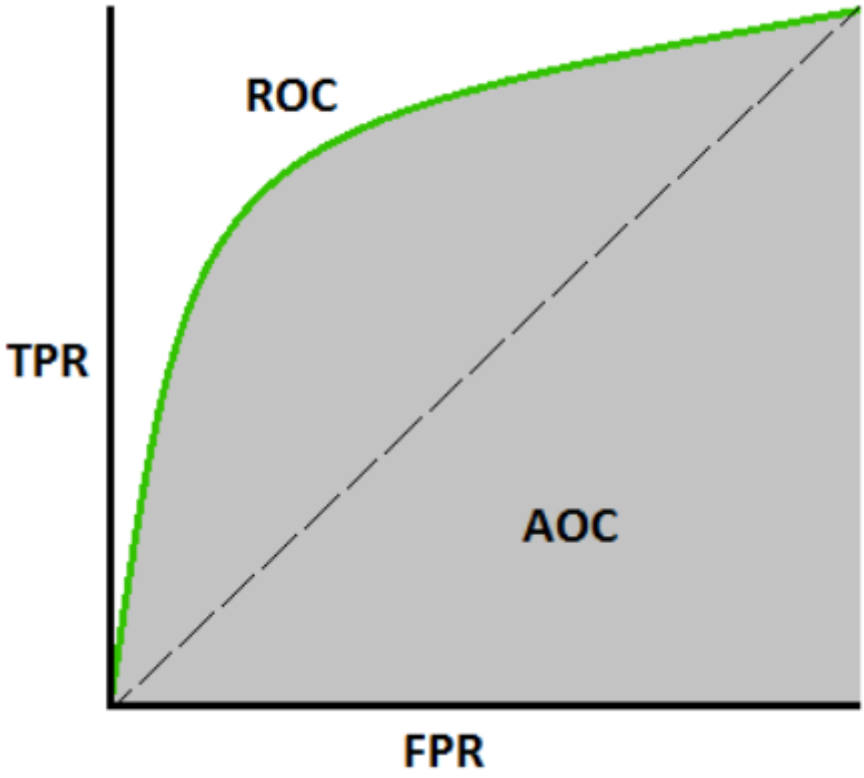
Evaluation metrics:

Different performance metrics are used to evaluate different Machine Learning Algorithms. For now, we will be focusing on the ones used for Classification problems. We can use the following classification performance metrics:

- AUC(Area under Curve)
- Accuracy
- Precision and recall

AUC-ROC

The AUC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.



Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all kinds predictions made.

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Accuracy = $\frac{TP+TN}{TP+FP+FN+TN}$

Precision

Precision is a measure that tells us what proportion of customers that we predicted as churn are actually had churn.

Precision = $\frac{TP}{TP+FP}$

Recall

Recall is a measure that tells us what proportion of customers that actually had churn and how much the algorithm is predicted as churn.

Recall = $\frac{TP}{TP+FN}$

Tools and Techniques:

- R or Python : Data mining, Model buliding
- GitHub : Versioning and Sharing
- Tableau : Data Visualization and Deployment