The topic of this project is customer churn. Data is received from Kaggle. The URL of the site is given below.

<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

While writing the code, “<https://towardsdatascience.com/predict-customer-churn-in-python-e8cd6d3aaa7>” is used as an outline.

Packages, which are used later in the project, are imported in the first code block.

import pandas as pd

import numpy as np

import sklearn

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.model\_selection import GridSearchCV

import seaborn as sns

from sklearn.metrics import accuracy\_score, confusion\_matrix

from sklearn.metrics import f1\_score, precision\_score, recall\_score

import matplotlib.pyplot as plt

Packages related to machine learning algorithms are imported in the following sections.

The data used is in the form of CSV. Using Pandas package, data is read. Initially, data is examined to be able to receive basic information such as feature names, row number, null elements etc.

Then preprocessing phase begins. First, null elements are dropped from the dataset. All categoric data are transformed into numerical data, since many machine learning models require only numerical data. This transformation is done through replacing “Yes, No” with binary values and creating dummies for features, which include more than 2 categorical values. Then, data is scaled, so that each value in the dataset is between 0 and 1. Scaling is done to eliminate high values, so that they do not affect the model more than others and to make machine learning models work with better efficiency.

Later, data visualization is done. Correlation matrix is created to see the effect of each feature on other features. Bar plots and piecharts are created for highly correlated features to see the trends.

Then, machine learning models are created. Initially, data is separated in test and train datasets. Then Logistic Regression, XGboost and Gradient Boosting models are created and run with the same train data sets. Hyperparameter tuning is applied to XGboost and Gradient Boosting models and these models are run again with optimized parameters. In total, 5 machine learning models are run. Performance measures of accuracy, precision, recall and F1 Score are noted for each model. According to the results, Tuned Gradient Boosting model is the best performing model.

In the results section, confusion matrix considering the results of Tuned Gradient Boosting is created. Additionally, feature importance figures of XGboost, Gradient Boosting and Tuned Gradient Boosting models are created.

Flowchart of the coding section of the project can be seen below.

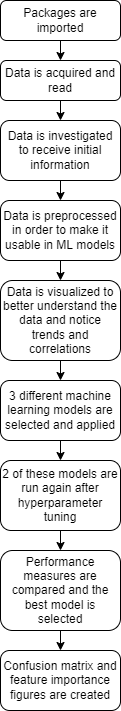


Figure 1. Flowchart of the Project

More detailed and thorough descriptions regarding each code block, functions, architecture of the tuned models, figures, etc. are present in the jupyter notebook file.