Customer Churn Dashboard

Chatterbox Telco Pvt Ltd

CS2500 Data Science
& Engineering Challenge

Name : Kajanan Selvanesan

Index Number : 190287R

Date : 01.04.2022

Problem definition

Problem

Customer churn occurs when a company loses consumers for various reasons, including bad service and lower rates elsewhere.It is one of the most significant and challenging concerns for telecommunication companies, credit card companies, cable service providers, and other businesses.

CEO of Chatterbox Telecom Pvt Ltd in the Banana Republic wants to analyze this customer churn in his/her company.

Motivation

Since acquiring new customers costs more than retaining existing ones, analyzing customer churn and finding ways to reduce it.

Objectives

- Analyze the data to provide various insights.
- Build a model to obtain the prediction for a new customer.
- Create a dashboard that should show various insights into the provided dataset and allow the user to obtain the prediction for a new customer.

Solution

Possible Technologies

We may analyze the data using R or Python to reveal various insights and create a model to predict customer churn for a new customer. Both are large-community open-source programming languages. New libraries or tools are added to their respective catalogs regularly. R is mainly used for statistical analysis, whereas Python offers a more comprehensive data science approach.

Plenty of technologies are available to create a web-based interactive dashboard with their open-source edition. Such as:

- **PowerBI:** Power BI is an interactive data visualization software product developed by Microsoft with primary focus on business intelligence.
- **Chart.js:** Chart.js is a free, open-source JavaScript library for data visualization.
- **D3.js:** D3.js is a JavaScript library for producing dynamic, interactive data visualizations in web browsers.
- **Grafana:** Grafana is a multi-platform open source analytics and interactive visualization web application.
- **Dash:** Dash is the original low-code framework for rapidly building data apps in Python, R, Julia, and F#.

Selected Technologies

Python code is more maintainable and robust than R code. Python didn't have many data analysis and machine learning libraries a few years ago. Python has recently caught up and offers cutting-edge APIs for machine learning and artificial intelligence. Numpy, Pandas, Scipy, Scikit-learn, and Seaborn are five Python libraries that can be used to perform most data science tasks. On the other hand, Python is more replicable and accessible than R. Python is the ideal choice if we need to use the outcomes of our study in an application or website. Therefore, **I have planned to go with Python**.

We can build our own very customized interactive dashboards from scratch in the Dash. But it makes it easy to plot graphs and visualize data. The beauty of this is that all this is done in pure Python or R. We can use data frames and then feed those to the graph elements in Dash. So Dash is more customizable, and it is based on Flask, Plotly.js, and React.js. However, Dash abstracts away all of the technologies and protocols required to build a full-stack web app with interactive data visualization, making it more convenient for the developers. Therefore we don't

need any prior knowledge of React and Flask. It is enough to know Python and a little bit of web development. Since I have chosen Python already, **Dash will be a better choice among other similar technologies**.

Procedure to Implement The Dashboard with Dash

Dash is a python framework created by Plotly for creating interactive web applications. We can install the Dash library in our working python environment using the `pip install dash` command. Typically Dash application has two major parts that are **layout** and **callbacks**.

The layout describes the view of our dash applications. Dash has **Dash Core components** (graphs, dropdowns, and sliders) and **Dash HTML components** (headings, paragraphs, images, etc.) to make those gorgeous views easily.

Callbacks provide interactivity to our dash applications. These are like python functions. Typically, they are triggered when a user tries to change anything on Dash Core components.

Let's consider the procedure,

1. Import necessary libraries.
2. Read the data and pre-process it.
3. Analyze the data and build a model if you need.
4. Initialize a Dash app.
5. Define the layout property of the initialized Dash app.
6. Initialized a callback using @app.callback().
7. View the application on the local server by running app.run_server().

[Local server: http://127.0.0.1:8050/]

Features of the dashboard

1. Allow the user to see the various insights into the provided dataset.
2. Allow the user to obtain the prediction for a new customer.

Using these features the company can take any action like give an offer before a customer gets out from this company with the help of a predictor, increase the call/message charges as much as possible without affecting the customers, etc.

Project timeline

Task	Start	End
Data pre-processing	01-04-2022	10-04-2022
Analyze the data	11-04-2022	30-04-2022
Build a model	01-05-2022	15-05-2022
Build a interactive dashboard	16-05-2022	31-05-2022
Testing	01-06-2022	05-06-2022