Analytics-Vidhya-Approach

Can you predict whether a customer will churn or not?

1.Understanding Problem Statement

Decreasing the Customer Churn is a key goal for any business. Predicting Customer Churn (also known as Customer Attrition) represents an additional potential revenue source for any business. Customer Churn impacts the cost to the business. Higher Customer Churn leads to a loss in revenue and the additional marketing costs involved with replacing those customers with new ones.

In this challenge, as a data scientist of a bank, you are asked to analyze the past data and predict whether the customer will churn or not in the next 6 months. This would help the bank to have the right engagement with customers at the right time.
Objective

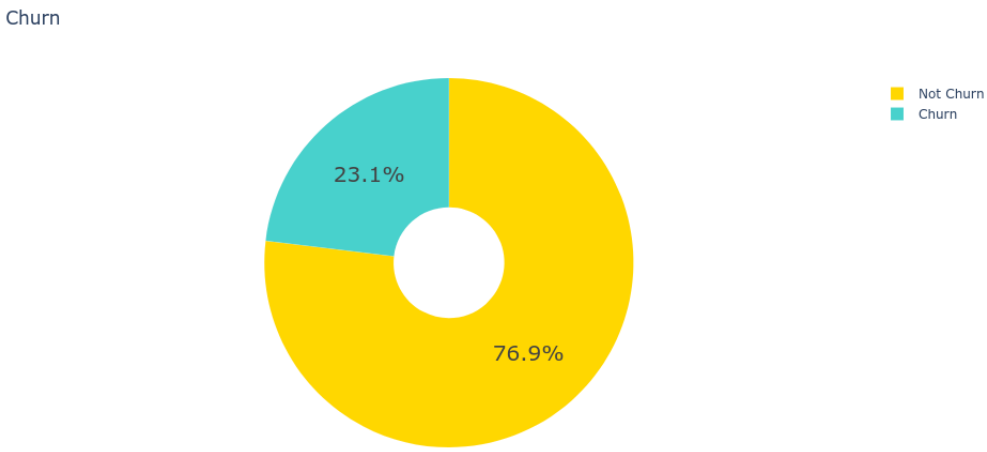
Our objective is to build a machine learning model to predict whether the customer will churn or not in the next six months.

2. Data:

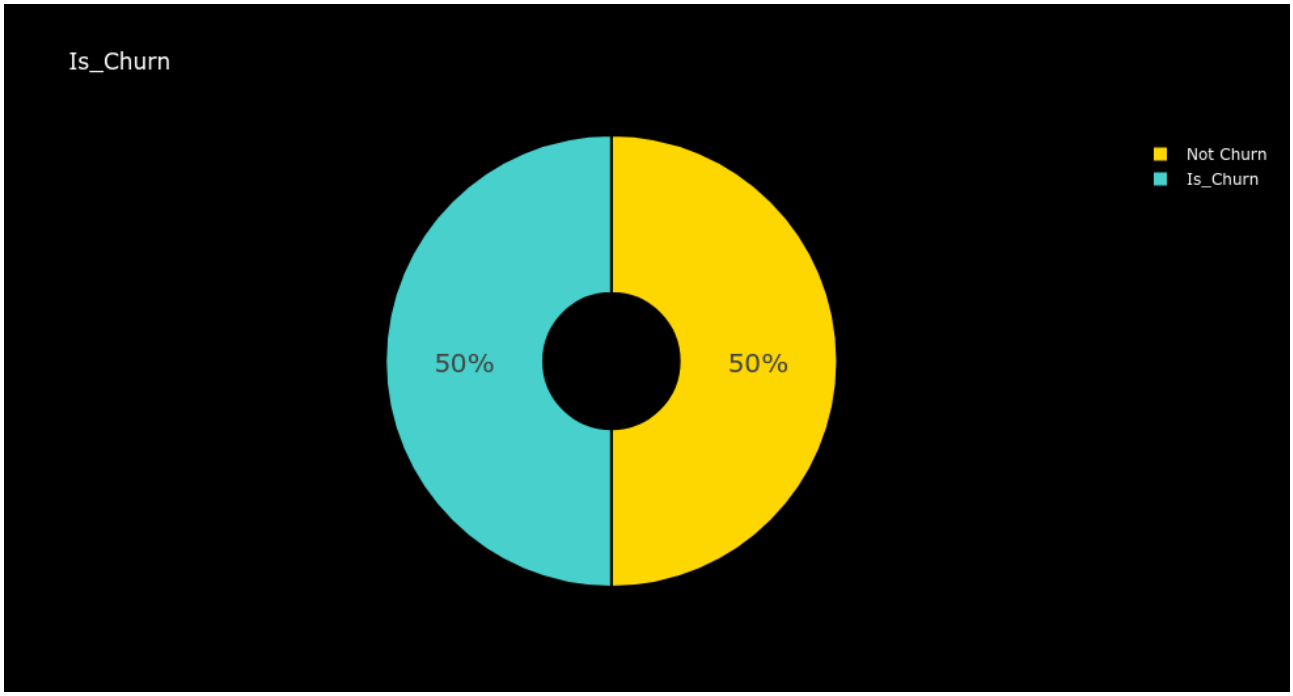
Null Values: The data had no NAN/Null values
Shape of Dataset : 6650 - rows and 11 columns.

-- Problems:
- The data set was heavily imbalanced, where only **23.1%** of the customers churned and **76.9%** did not.
To tackle this problem- I utilized *SMOTE* (*Please read more about this on*:https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/)

Before SMOTE:



After SMOTE:



3.Outliers:
Columns such as Age and Balance had to be cleaned for outliers.

4.Mistyped Data : Some values in Product_Holding Columns were mistyped as '+3' and '3+' which identified the column as a categorical column, had to replace those values with '3' and change the data type to 'int64'.

Steps Taken:

1.Cleaned the dataset

2.Data Visualization.

3.One-Hot Encoding: Some of the values such as Gender, Income, Credit Category were in like Male, Female, Income more than 5L etc, used Pandas Get Dummies function to encode them.

4.Used drop_first = True to avoid the dummy variable trap.

5.Split the Data in Test and Train split: The test ratio was 20% of the dataset, while 80% was training Data.

6.Min Max Scaling: Variables such as Age, Balance were in a very different range as the other variables were in between 0 to 1, scaled the dataset to get them in the same range.

7.Outlier Removal: The columns such as Age and Balance had values that were too high. So here I used Isolation Forest to identify them and have them removed from the data.

Read more about Isolation forest here:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

8. Correlation: Checking the correlation between the Features and target I dropped ['Credit_Category_Poor', 'Income_Less than 5L','Income_5L - 10L', 'Vintage','Balance']) columns.

9. SMOTE: Used Smote and oversampled the minority class.

10. Models: Final Model used was Extreme Gradient Boosting with Hyperparameter Tuning. Parameters were tuned using RandomizedSearchCV.

---- --- ---- ----- ------ ---- ---- -- ----- ----- -------- ----------- ------ -------- ----------- ---------
---------- ----------