Predicting Customer Churn in a Telecommunications Company Using Machine Learning Techniques

By Niraj Kumar Sahu.
For **Speakx**

Student of B.Tech CSE at Lovely Professional University

Reg no. 12102759

1. Introduction

In today's highly competitive business world, customer retention has become a top priority for companies in the industry. The ability to predict customer behaviour and implement effective retention strategies can impact an organization's profitability and long-term success.

This data, which includes a wide range of customer behaviours from service subscriptions to financial data to publicly available content, enables in-depth analysis to uncover patterns and customer forecasts. By using advanced analytics, businesses can gain insights that will reduce stress and increase retention. and reporting the development of customer retention plans.
Through data-driven insights and predictive modelling, our goal is to make Churn prediction model.

2. Data Pre-processing

Data preprocessing is an important step in preparing the dataset for subsequent analysis and modelling. In this section, we describe the preliminary procedures used to ensure the quality and suitability of model prediction data.

Data set : Telco Customer Churn

The dataset consists of 7,043 entries and 21 columns, encompassing various customer attributes and behaviours. Before conducting any analysis or modeling, it is essential to preprocess the data to ensure its quality, consistency, and suitability for further analysis. The pre-processing steps undertaken are outlined below:

```
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   customerID       7043 non-null   object
 1   gender           7043 non-null   object
 2   SeniorCitizen    7043 non-null   int64
 3   Partner          7043 non-null   object
 4   Dependents       7043 non-null   object
 5   tenure           7043 non-null   int64
 ……..
 15  Contract         7043 non-null   object
 16  PaperlessBilling 7043 non-null   object
 17  PaymentMethod    7043 non-null   object
 18  MonthlyCharges   7043 non-null   float64
 19  TotalCharges     7043 non-null   object
 20  Churn            7043 non-null   object
```

2.1. Handling Categorical Variables:

Several columns in the dataset contain categorical variables, such as 'gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', and 'Churn'. These variables need to be appropriately encoded for numerical analysis.

2.2.Handling Special Cases:

Some categorical variables have special cases, such as 'No phone service' in the 'MultipleLines' column and 'No internet service' in the 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', and 'StreamingMovies' columns.

These cases were replaced with 'No' to ensure uniformity and consistency across the dataset.by organizing rows and columns based on their similarity.

2.3. Data Encoding:

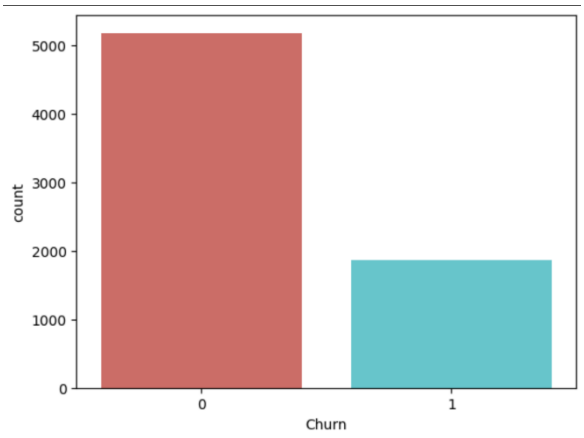
To transform categorical variables into numerical format, the LabelEncoder from the sklearn.preprocessing module was employed. Each categorical column was encoded individually to maintain the integrity of the data.

2.4 Handling missing values:

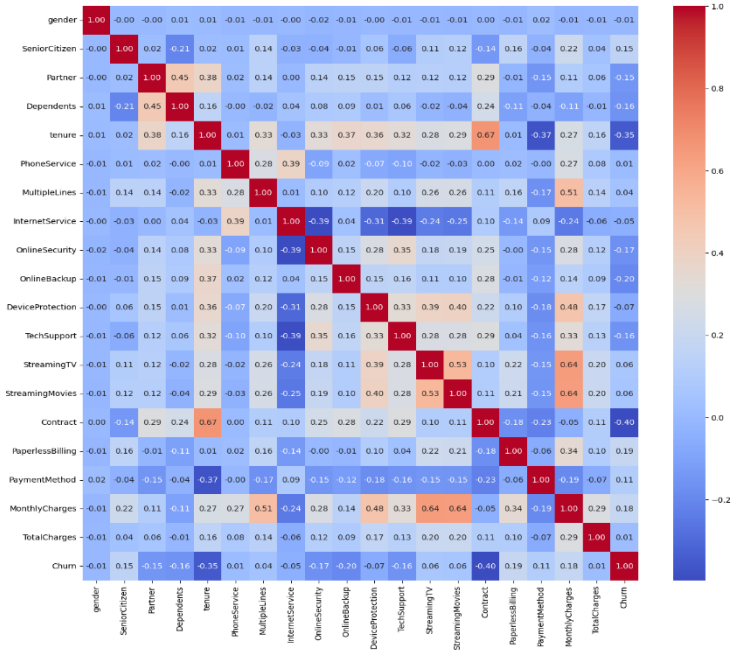
After analysis, there were no missing values in the data set. Therefore, there is no need to load or process missing data. To make the number easier to identify, it must be converted to a document number. But this issue is not explicitly addressed in the code snippet provided. Additional considerations may be necessary depending on the specific needs of the scan.

3. Data visualization

A countplot was generated to visualize the distribution of churn within the dataset. The x-axis represents the 'Churn' variable, indicating whether a customer has churned or not, while the y-axis displays the count of customers falling into each category.



A correlation matrix heatmap was generated to visualize the pairwise correlations between different variables in the pre-processed dataset. The heatmap illustrates the strength and direction of the relationships between variables, with higher correlation coefficients indicating stronger associations.



By going through heatmap it is observed that StreamingTV and StreamingMovies colums have near same corelation so one of the column is dropped to reduce the computation complexity and improve feature selection.

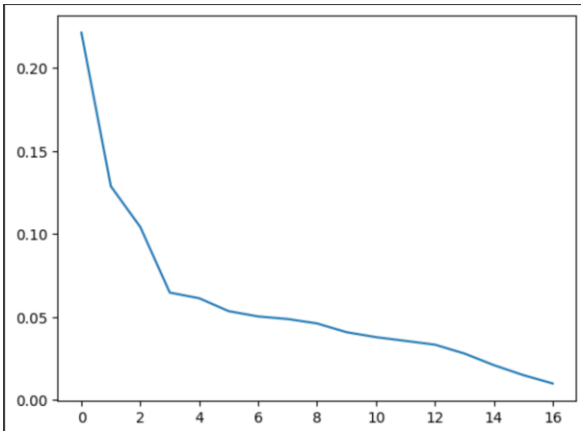
Also it is observed that the “Gender” column does not have much corelation withing dataset. The StandardScaler from the sklearn.preprocessing module was utilized to standardize the features in both the training and testing datasets. Standardization ensures that all features have a mean of 0 and a standard deviation of 1, which is essential for certain machine learning algorithms that are sensitive to feature scaling.

Dimensionality Reduction with PCA:

Principal Component Analysis (PCA) was applied to reduce the dimensionality of the dataset while preserving as much variance as possible. The number of principal components was set to 17, which effectively captures the variance in the data while reducing the feature space.

Explained Variance Ratio:

After applying PCA, the total explained variance ratio was calculated to assess the cumulative amount of variance retained by the selected number of principal components. The reported value of 0.9999999999999998 indicates that approximately 99.99% of the variance in the original dataset is preserved by the 17 principal components.



By looking the PCA graph it concluded that there is no need for applying full PCA.

3. Model selection

While the dataset primarily classification model, we ventured into implementing and analyzing classification models on the same dataset. Upon thorough examination of the dataset, we discovered significant similarities conducive to classification modeling. Hence, we proceeded to explore the following classification models:

1. Logistic regression:
In the logistic regression model's classification report, we observe distinctive patterns in the performance metrics:

```
classification_report
              precision    recall  f1-score   support

           0       0.86      0.90      0.88      1036
           1       0.68      0.58      0.63       373

    accuracy                           0.82      1409
   macro avg       0.77      0.74      0.75      1409
weighted avg       0.81      0.82      0.81      1409
```

Precision: Precision measures the proportion of correctly predicted positive cases out of all cases predicted as positive. For class 0, the precision is 0.86, indicating that 86% of the predicted non-churn cases are correct. For class 1, the precision is 0.68, indicating that 68% of the predicted churn cases are correct.

Recall: Recall (also known as sensitivity or true positive rate) measures the proportion of correctly predicted positive cases out of all actual positive cases. For class 0, the recall is 0.90, indicating that 90% of the actual non-churn cases are correctly identified. For class 1, the recall is 0.58, indicating that 58% of the actual churn cases are correctly identified.

F1-score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. For class 0, the F1-score is 0.88, and for class 1, the F1-score is 0.63.

Support: Support indicates the number of actual occurrences of each class in the dataset. In this case, there are 1,036 instances of class 0 and 373 instances of class 1.

Accuracy: The overall accuracy of the logistic regression model is 0.82, indicating that 82% of the predictions are correct.

2. Gradient Boosting Classifier Accuracy Report:

The accuracy of the Gradient Boosting Classifier model is reported as approximately 0.813, or 81.3%. This accuracy value represents the proportion of correctly classified instances out of the total number of instances in the dataset.

An accuracy of 81.3% indicates that the Gradient Boosting Classifier model performs well in predicting the target variable, likely customer churn in this context. However, it's important to consider this accuracy in the context of the dataset and the problem being addressed. Depending on the specific requirements and constraints of the application, further evaluation of the model's performance may be necessary, including assessing other metrics such as precision, recall, and F1-score, especially in cases of class imbalance.

the utilization of the SMOTEENN sampling technique and various machine learning models for predicting customer churn. We explore how the SMOTEENN technique, which combines over-sampling and under-sampling, affects model performance and compare the accuracy of different models.

SMOTEENN Sampling:

Purpose: SMOTEENN (Synthetic Minority Over-sampling Technique and Edited Nearest Neighbors) is employed to address class imbalance by oversampling the minority class (churn) and undersampling the majority class (non-churn).
Implementation: The SMOTEENN algorithm is applied to the dataset, resulting in a resampled dataset with balanced class distribution.
Effect on Dataset: The resampled dataset contains an equal number of instances for both churn and non-churn classes, improving the model's ability to learn from minority class instances.
Model Evaluation:

Model Selection: Six machine learning models (Random Forest, Gradient Boosting, Logistic Regression, K-Nearest Neighbors, Decision Tree, and Naive Bayes) are evaluated using the resampled dataset.
Model Training: Each model is trained using the resampled training data and evaluated on the resampled test data.
Hyperparameter Tuning: GridSearchCV is used to optimize hyperparameters for models that require tuning, such as Random Forest, Gradient Boosting, Logistic Regression, and Decision Tree.
Model Evaluation Metrics: The accuracy of each model is calculated using the accuracy_score function, which measures the proportion of correctly classified instances out of the total instances.

Results:

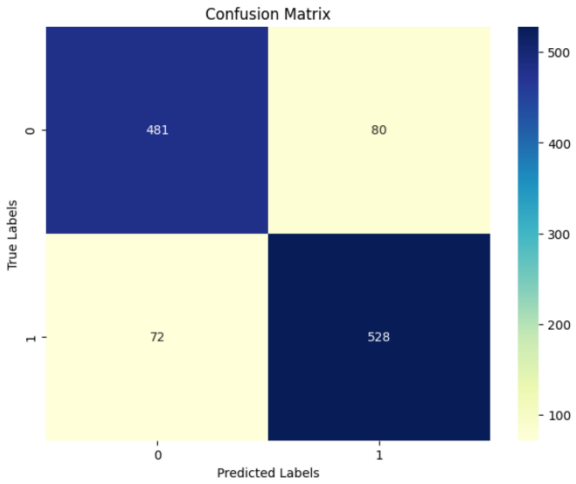
Model Performances:

Random Forest: Test Accuracy - 93.6%
Gradient Boosting: Test Accuracy - 93.8%
Logistic Regression: Test Accuracy - 90.3%
K-Nearest Neighbors: Test Accuracy - 88.4%
Decision Tree: Test Accuracy - 91.5%
Naive Bayes: Test Accuracy - 85.8%

Best Model: The Gradient Boosting Classifier achieved the highest accuracy of 93.8% on the resampled test data.

4.Conclusion

We can also we see through the Confusion matrix that our model is performing well.



The utilization of SMOTEENN sampling technique significantly improved the performance of machine learning models in predicting customer churn. The Gradient Boosting Classifier emerged as the best-performing model, with an accuracy of 93.8%. Overall, this approach demonstrates the effectiveness of SMOTEENN sampling in

handling imbalanced datasets and highlights the importance of selecting appropriate models for churn prediction tasks. Further optimizations and model refinements could potentially enhance predictive accuracy and provide more actionable insights for customer retention strategies.