Utilizing Logistic Regression and Decision Tree Classifier in Customer Churn Prediction

Introduction

For a business with various services, in this case, a fictional Telco company, it is essential to keep old customers because studies suggest that depending on the industry you are in, acquiring a new customer can cost five to seven times more than retaining an old one .[1] Therefore, this machine learning project is built to predict customer churn, which promises to help businesses make better customer retention programs.

This report consists of 4 main parts following by the references and appendices. The first part "Problem formulation" discusses the problem and the objective of this project. The second part "Method" describes the deployed methods. Then, the results from applying the models are presented and compare in the section "Result". Finally, summary of the findings and further future improvements are discussed in the section "Conclusion".

Problem Formulation

This project aims to predict whether a customer is likely to leave a service or subscription. My machine learning problem is framed as a supervised learning task, where the model is trained using labeled data pairs, consisting of input features and a corresponding output label, which seeks to predict the label from unseen data.

The data points of the problem represent the statistics of 7073 customers of Telco. The concerned label is whether the customer is going to cancel the service or not (1 for yes, 0 for no). The features include services that each customer has signed up for, customer account information, and demographic info about customers.

The dataset was collected from Kaggle (https://www.kaggle.com/datasets/blastchar/telco-customer-churn) [2]. It is a highly rated dataset for its accuracy and usability.

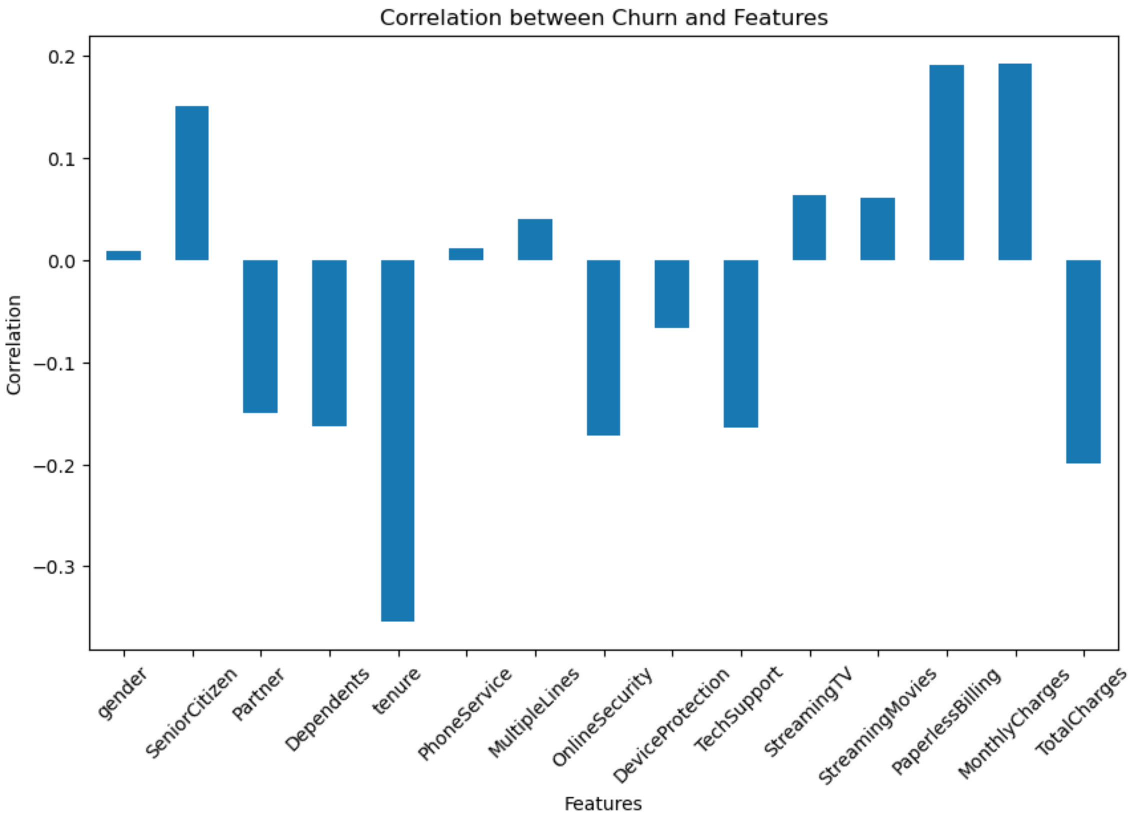
Methods

Dataset overview:

There are 7043 datapoints in the dataset, corresponding to 7043 customer statistics with no missing data in any field except for column 'TotalCharges'. To prepare the data for the Machine learn task, I used numpy.to_numeric() method to convert data of 'object' data type to numeric value, pandas.map() to handle Yes/No information, and pandas.get_dummies() to process other categorical data.

Feature selection:

As stated in the problem formation, this project aims to predict the likelihood of a customer canceling the service, which means irrelevant data such as 'customerID' are omitted. Furthermore, after visualising data of correlation between the features and the label, features 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'OnlineSecurity', 'DeviceProtection', ‘PaperBilling’, ‘MonthlyCharges’, and ‘TotalCharges’ show **much stronger correlations to the label** than others (see appendix), therefore these are chosen. Moreover, categorical features ‘InternetService’, ‘Contract’, ‘PaymentMethod’, and ‘OnlineBackup’ **show significant impacts on churn rate**, thus they are also included.



Choice of ML model

Logistic Regression is chosen because the concerned label is categorical (whether a customer is going to churn or not). It is reasonable to use this model because it is a popular model to handle binary classification problems.

Furthermore since this is a classification problem, it is possible to use Decision Tree Classifier. Because it break down complex data into parts, it is expected to provide a useful comparison to Logistic Regression.

Choice of Loss Function

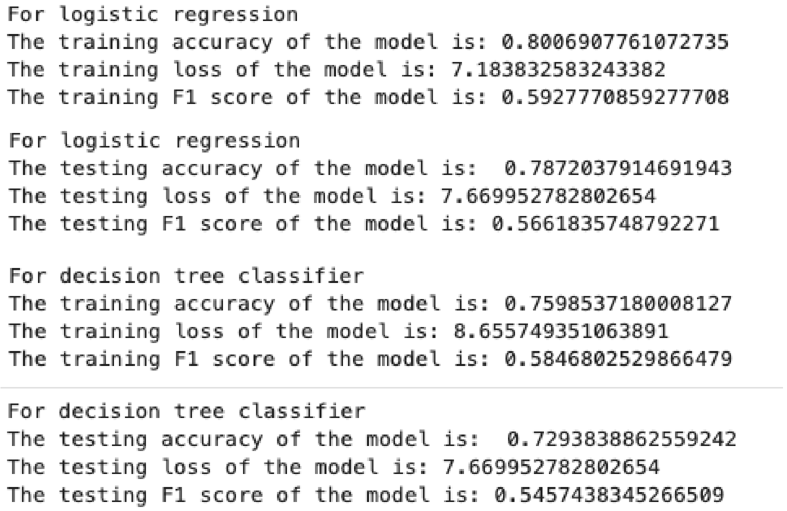
The logistic loss function is chosen because it is convenient with the help of a ready-made library for logistic regression. I also choose the same loss function for Decision Tree Classifier to make it easy to compare this model performance to that of Logistic Regression.

Model validation

The data is partitioned following the 70/30 ratio, that is 70% for training and 30% for testing. According to this publication (https://scholarworks.utep.edu/cs_techrep/1209/)[3], empirical data shows that this approach is likely to produce the best result. The data is partitioned using library function train_test_split().

Results

The results of the two methods evaluated using accuracy score, loss, and F1 score are shown below.



```
For logistic regression
The training accuracy of the model is: 0.8006907761072735
The training loss of the model is: 7.183832583243382
The training F1 score of the model is: 0.5927770859277708

For logistic regression
The testing accuracy of the model is:  0.7872037914691943
The testing loss of the model is: 7.669952782802654
The testing F1 score of the model is: 0.5661835748792271

For decision tree classifier
The training accuracy of the model is: 0.7598537180008127
The training loss of the model is: 8.655749351063891
The training F1 score of the model is: 0.5846802529866479

For decision tree classifier
The testing accuracy of the model is:  0.7293838862559242
The testing loss of the model is: 7.669952782802654
The testing F1 score of the model is: 0.5457438345266509
```

For logistic regression, the testing accuracy, loss, and F1 is approximately that of the testing set, thus it it reasonable to conclude that the model does not show sign of serious overfitting. On the other hand, for Decision Tree Classifier, there is a notable discrepancy between the training loss and the testing loss.

Because logistic regression has better accuracy, lower loss, and higher F1 score in comparision to Decision Tree Classifier, it can be concluded that logistic regression is the better model this project. Therefore, logistic regression is chosen to be the final method.

As stated in the previous section, my data is partitioned following the 70/30 ratio, that is 70% for training and 30% for testing. The data is partitioned using library function train_test_split(). The final test error of the chosen method using this construction of the test set is approximatly 7.669952782802654.

Conclusion

In conclusion, the project has managed to make prediction with the accuracy of about 79% on the test set using Logistic Regression as the prefered method. This is a satisfactory result given the limited amount of data and the simplicity of the implementation. However, there are still room for future improvement. First, it would always be beneficial to have more data. Secondly, using another methods of sets partitioning such as k-fold cross-validation may yield better prediction results.

Reference

[1] https://www.forbes.com/sites/forbesbusinesscouncil/2022/12/12/customer-retention-versus-customer-acquisition/
[2] https://www.kaggle.com/datasets/blastchar/telco-customer-churn
[3] https://scholarworks.utep.edu/cs_techrep/1209/

Appendices

- Code in Github repository: https://github.com/ponyo19/Customer-Churn-Prediction