Advanced Python AI and ML Tools

(AML 2203)

PROJECT REPORT

Telecom Churn Analysis

Group Members

Anto Francis (C0825095)

Omer Volkan (C0831373)

Rupesh Chandran (C0826779)

Sachin Sreekumar (C0825096)

Submitted to

Prof. Vahid Hadavi

Table of Contents

List of Figures

Abstract

Customers are the key to success for any business, even if it is a Start-up or a large MNC. As competition in the business increases, customers tend to move towards the better company that provides the service instead of staying with the same service for too long. Hence, it is essential to give the customers good service and keep them stick to the business as they are the ones who drive the company's revenue. The churning of customers is one of the significant metrics that a company would want to focus on. Companies should be aware of their customer churn rate in order to tackle the retention of customers and stay profitable in these times of business rivalry. Customer churning is a usual and frequent scenario in the telecom industry. Users always go for better plans and thus change from one service provider to another very frequently. These days, churning is so easy that users don't even have to change their mobile numbers while changing to a new service. The aim of this project is to determine the trends in customer churning and to do an exploratory analysis of the churn data. We will also be creating various classification models to predict whether a customer is likely to churn or not. In addition to that, we will be comparing the accuracy of these models to find the best churn classifier.

1. Introduction

1.1. Background

Churning in the telecommunications sector is one of the risks for their loss in revenue. According to the reports, the churning rate is approximately 2% per month, but it could increase to 67% yearly. Since customer acquisition is more hard and expensive than customer retention, good service is the only way to hold customers (Hughes 2021). It is vital for the business to use their data to identify the customers who are likely to churn. A good churn prediction model could be useful for them in order to identify and prevent their customers from changing to a new service.

1.2. Problem Statement

Customer churn rate analysis has been important for the service provider to keep their customers from moving to a new service. It is useful for them to identify the patterns of customers that churn during the past months. Machine learning algorithms can be used to create models to predict whether a customer churns or not. Once the business is able to find the patterns, it will be able to handle them easily and prevent them from churning. The aim of this project is to identify the patterns likely to churn, create machine learning models to predict churn, and compare different algorithms to identify the best classifier.

1.3. Limitations

One of the limitations of this project is that it is limited to the telecommunications sector only. Churning happens in all businesses, and it is important to address them as well. However, here we focus on only telecom data. Another limitation is that we are using basic classification models for prediction. There are advanced models which could be used to predict the churn more effectively.

2. System Design

2.1. Data Collection

We have used Telecom Churn Dataset from Kaggle for this project. It is the data of multiple users in the South-East part of Asia, and it belongs to various services. The dataset has around 100,000 records and consists of 226 features. The features cover every detail of the telecom user, such as Incoming Call usage, Outgoing usage, Internet usage, etc., for the months of June, July, August, and September. The dataset doesn't come with a target feature which is the indicator which is the indicator whether the user is churning or not.

2.2. Data Pre-processing

Initially, we selected all the premium customers from the entire dataset. We calculated the total recharge done by each customer each month by adding the total expenditure of calls and SMS with the total amount spent for data. The total expense of data was calculated by multiplying the average data recharge in each month with the total data recharged. We considered the premium users as those who spend more than or equal to 70 percentile of the average recharge amount in the first two months. This reduced our dataset record count to 30000.

Then we did a feature extraction to see whether the user was churning or not. We created a column that checks whether the total recharge in the 9th month is 0 or not. If it is 0, we set the column value to 1, which indicates that the user is churning, and 0 otherwise. We also calculated the average of certain fields in each month and removed unnecessary features from the data frame.

If any of the features in the dataset has very few unique values, then those columns will not add any good to the quality of the model. So, we identified a set of columns that has only one unique value and removed it from the data frame. It reduced our columns to 207.

As the next step, we checked for null values in each column in the dataset, and we found out that few of the columns had more than 40% null values in it. We calculated the percentage of null values in each column and removed those columns having more than 40% null values and which don't belong to September. Also, we have removed those few rows which have null values in them.

We also found the features which are highly correlated to each other. There is no use in keeping the columns which are having high correlation. It reduces the quality of the model. Hence, we identified those and dropped them. We also converted a few data types like the date to their original type.

As a final step of cleaning, we dropped the 9th month's columns after taking a backup, as we are going to predict later in the project.

After the data cleaning and preprocessing are done, the shape of our data frame is reduced to 28000 rows and 62 columns.

2.3. Data Preparation

We have implemented various feature selection and feature extraction activities to identify the best columns needed for data analysis and model creation. We split our dataset into 70% train data and 30% data for testing the model.

2.4. Data Prediction

We used various models such as SVC, RFC, and LogisticRegression to predict whether the user churns or not. We also compared our models in order to find the best among them.

2.5. Data Visualization

We used a few visualization techniques to get a rough understanding of the proportion of churning among the customers. We also made use of the ROC curve to see the performance of our models visually.

3. Exploratory Data Analysis

We performed a certain analysis of our cleaned dataset to get insights from it. First, we calculated the percentage of users who are not using any of the services in the month of September.

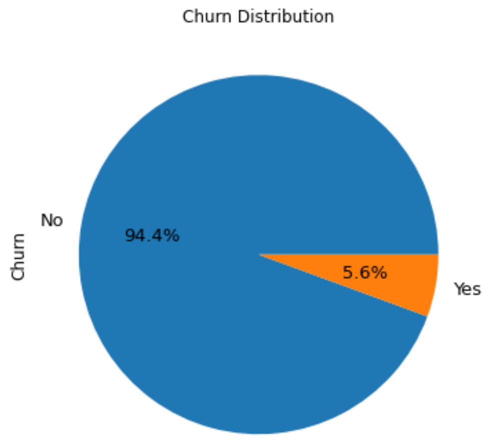


Figure 1: Proportion of Churn

From the pie chart, we identified that the percentage of churn data in premium customers is very less, which is a good thing for the provider.

We calculated the tenure of each customer in months and years to determine how long the user stays with the service provider.

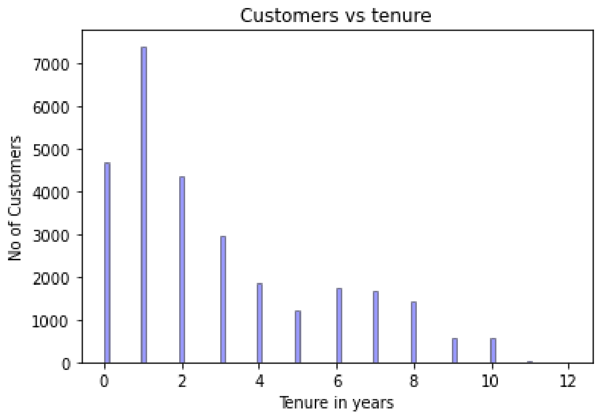


Figure 2: Customer distribution in Years

We found out that most of the customers are new to the service, and a majority of them have been using it for less than five years. When we look closely at the graph, we can see that the number of customers gradually reduces in the coming years, which is a great indication of churning.

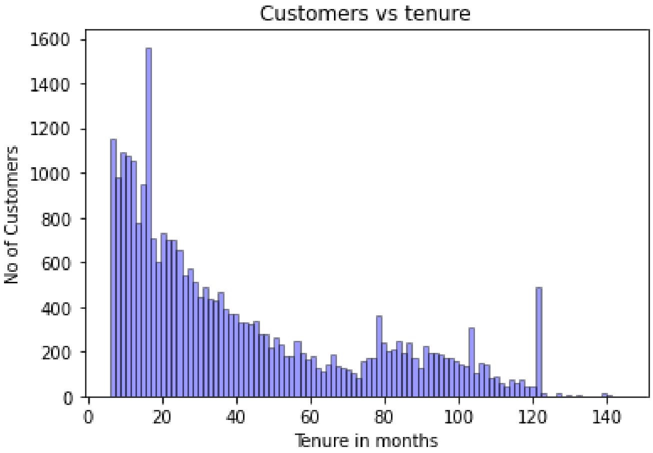


Figure 3: Customer distribution in months

Upon further analysis, we identified that some of the features such as Maximum recharge amount, revenue from customers, and Standard Outgoing calls are some of the important indicators of churning.

4. Model Creation and Comparison

4.1. Preparation of Data

Before creating the model, it is important to improve the quality of the dataset by reducing the bias. As our dataset is not balanced, the prediction could have some bias toward the non-churn outcome. Hence, we used a library function called SMOTE to oversample the dataset. Oversampling is beneficial in order to reduce the effect of bias in the dataset.

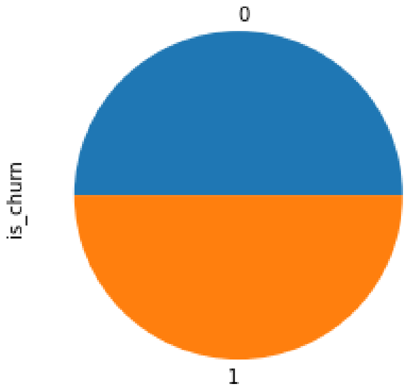


Figure 4: Proportion of churn - no bias

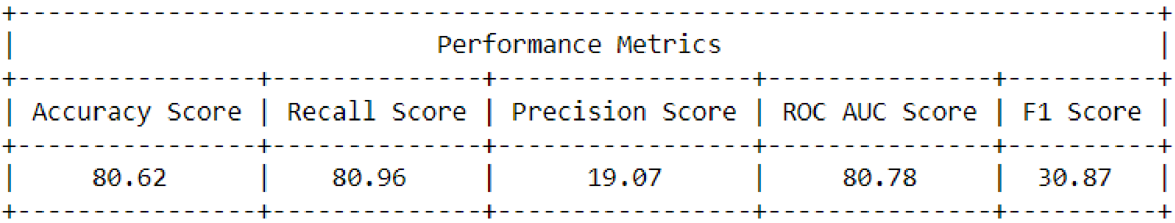
Once we implemented the oversampling technique, we divided the dataset into test and train sets and made it ready for classification.

4.2. Classification Models

Once the dataset was ready for model creation, we imported the necessary libraries and created functions for performance metric calculations. The listing below each model one by one:

4.2.1. Support Vector Classifier

SVC model has been created with a linear kernel to get a model with an accuracy of 80.62%.



```
+-----------------------------------------------------------------------------------+
|                                Performance Metrics                                |
+-----------------+---------------+-----------------+-----------------+-----------+
| Accuracy Score  | Recall Score  | Precision Score |  ROC AUC Score  | F1 Score  |
+-----------------+---------------+-----------------+-----------------+-----------+
|      80.62      |     80.96     |      19.07      |      80.78      |   30.87   |
+-----------------+---------------+-----------------+-----------------+-----------+
```

Figure 5: Performance Table: SVC

The performance table shows that the model has a good accuracy of 80.62%. It also has a good Recall and ROC score. However, the F1 score is less due to low precision.

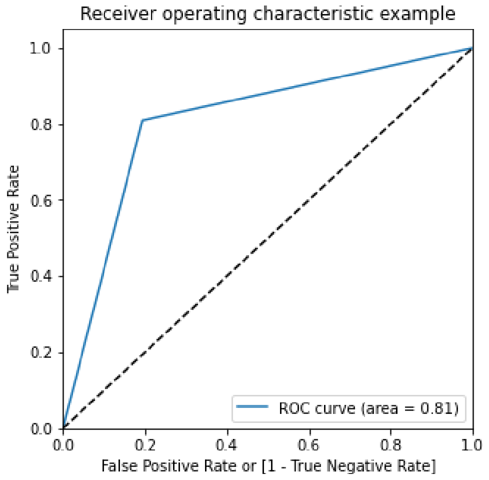
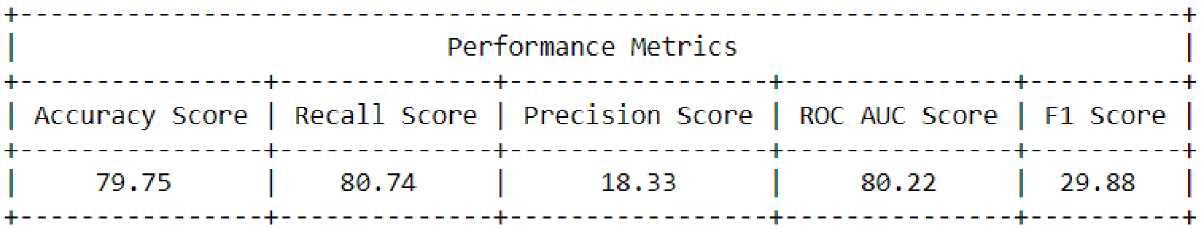


Figure 6: ROC Curve: SVC

The ROC curve of the SVC model indicates 81% are under the curve.

4.2.2. Logistic Regression

Secondly, we used the Logistic Regression algorithm to create the model. Before that, we reduced our features in order to take the most important features. We used Principal Component Analysis to implement the dimensionality reduction.



```
+----------------------------------------------------------------------------+
|                            Performance Metrics                             |
+----------------+--------------+-----------------+---------------+----------+
| Accuracy Score | Recall Score | Precision Score | ROC AUC Score | F1 Score |
+----------------+--------------+-----------------+---------------+----------+
|     79.75      |    80.74     |      18.33      |     80.22     |  29.88   |
+----------------+--------------+-----------------+---------------+----------+
```

Figure 7: Performance Table: Logistic Regression

The table says that the accuracy is 79.75%. The result is somewhat similar to the SVC model.

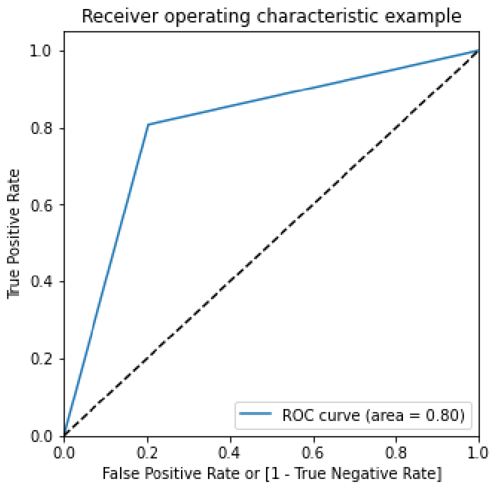
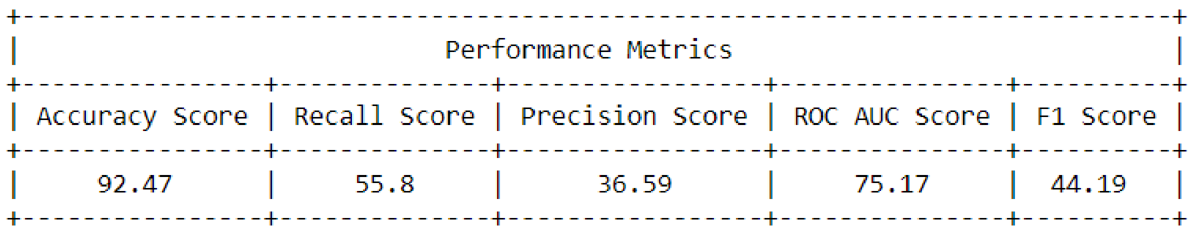


Figure 8: ROC Curve: Logistic Regression

The area under the curve for Logistic Regression shows 80%.

4.2.3. Random Forest Classifier

We created a Random Forest Classification model using its default hyperparameters.



```
+----------------------------------------------------------------------------+
|                            Performance Metrics                             |
+----------------+--------------+-----------------+---------------+----------+
| Accuracy Score | Recall Score | Precision Score | ROC AUC Score | F1 Score |
+----------------+--------------+-----------------+---------------+----------+
|     92.47      |     55.8     |      36.59      |     75.17     |  44.19   |
+----------------+--------------+-----------------+---------------+----------+
```

Figure 9: Performance Table: RFC

As shown in the table, the accuracy of the model increases when the Random Forest algorithm is used. F1 score shows an improvement when compared to other models as precision increases. However, the recall and ROC scores have been reduced.

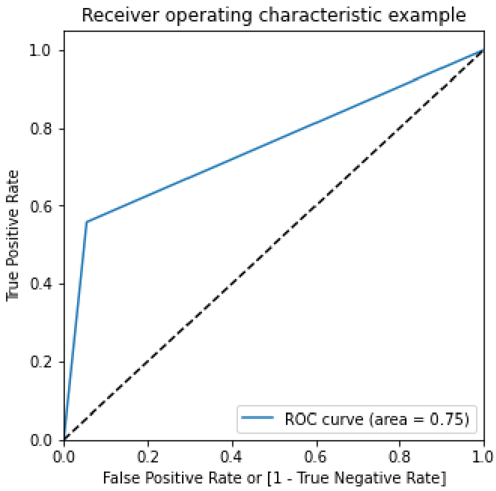
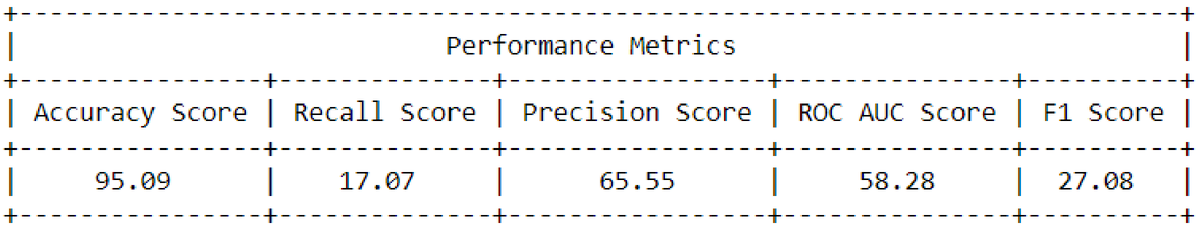


Figure 10: ROC Curve: RFC

ROC Curve shows a sharp deviation before 0.6 Positive Rate and got an area of 75% under the curve.

We selected some parameters and did hyperparameter tuning using GridsearchCV and KFold to analyze the performance improvement.



```
+--------------------------------------------------------------------------------+
|                               Performance Metrics                              |
+----------------+-------------+-----------------+---------------+----------+
| Accuracy Score | Recall Score | Precision Score | ROC AUC Score | F1 Score |
+----------------+-------------+-----------------+---------------+----------+
|     95.09      |    17.07     |      65.55      |     58.28     |  27.08   |
+----------------+-------------+-----------------+---------------+----------+
```

Figure 11: Performance Table: RFC after Hyperparameter Tuning

The table after hyperparameter tuning indicates that the accuracy has risen to 95%. However, the recall has deteriorated to 17%, which is again a bad sign.

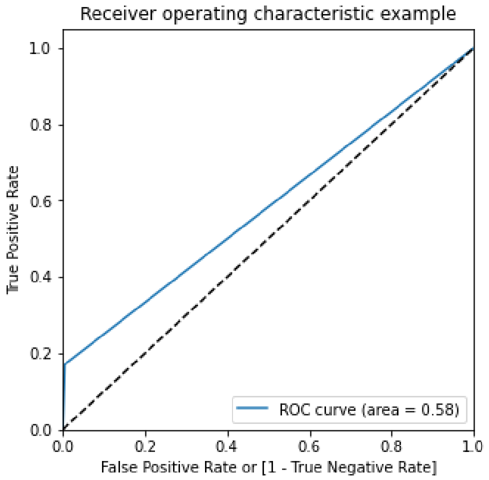


Figure 12: ROC Curve: RFC after Hyperparameter Tuning

The graph shows that the model has become even worse, with a ROC score of 58%.

5. Conclusion

As the competition in Telecom Sector has been increasing on a daily basis, service providers should be aware of customer satisfaction, and they should be knowledgeable about the parameters which lead to customer churn. In this project, we have identified some of the parameters that indicate churning in the telecom market. Also, we have created a model with an accuracy of 81%-95% to predict churning.

6. References

- Hughes, A. (2019). Churn reduction in the telecom industry. *Database Marketing Institute*.

- Umashankarsomaskar. (2022, March 7). Telecom Churn Dataset. Kaggle. Retrieved April 19, 2022, from https://www.kaggle.com/code/umashankarsomaskar/telecom-churn-pca-rf-hyper-pram-tune/data