CUSTOMER CHURN ANALYSIS

Big Data Analysis Project Report

Submitted by

Aadhar Mehta	**20MA20001**
Ajitesh Bharti	**20MA20004**
Chitresh Choudhary	**20MA20019**
Esha Jain	**20MA20021**
Kunal Kumar	**20MA20028**
Lovely Kaur	**20MA20031**
Pooja Sonje	**20MA20055**
Princy	**20MA20046**
Sonal	**20MA20054**
Yash Laxkar	**20MA20066**

under

Professor Bibhas Adhikari

Department of Mathematics

Indian Institute of Technology
Kharagpur

Jan 2023 - Apr 2023

Chapter 1: INTRODUCTION

1.1 PROBLEM FORMULATION:

Problem Statement: Churn Prediction for a Subscription-Based Service

Background: A company provides a subscription-based service to its customers. The service requires customers to sign up for a monthly or yearly subscription, and they can access the service during the subscription period. However, the company has been experiencing a high churn rate, where customers are canceling their subscriptions before the subscription period ends. This is resulting in loss of revenue and hindering the company's growth.

Objective: The objective of this project is to develop a churn prediction model that can accurately identify customers who are likely to churn, i.e., cancel their subscriptions before the subscription period ends. By predicting churn in advance, the company can proactively take measures to retain those customers, such as offering promotions, improving service quality, or providing personalized offers, to increase customer retention and reduce churn rate.

Data: The company has a dataset that contains historical data of its customers, including demographic information, subscription details, usage behavior, and churn status (churned or not churned). The dataset is large and contains thousands of records with multiple features.

Key Challenges:

- Imbalanced Data: The churned customers may be a small proportion of the overall customer base, resulting in imbalanced data. This can lead to biased model performance and inaccurate predictions.

- Feature Engineering: Identifying relevant features or variables that contribute to churn prediction can be challenging. The dataset may contain irrelevant or redundant features that need to be carefully analyzed and selected for model development.
- Dynamic Nature of Churn: Churn behavior may change over time, and patterns that were previously relevant may not be applicable anymore. It is important to develop a model that can adapt to changing churn patterns and provide accurate predictions.
- Real-time Prediction: The company may require real-time churn prediction to take timely actions to prevent churn. Developing a model that can provide real-time predictions can be challenging due to the need for quick processing and low-latency predictions.
- Success Metrics: The success of this project will be measured based on the following metrics:
- Accuracy: The model should have high accuracy in predicting churned customers. Precision and Recall: Precision and recall should be balanced to ensure that both false positives and false negatives are minimized, and the model provides accurate predictions.
- F1 Score: The F1 score, which is the harmonic mean of precision and recall, should be high to ensure a good balance between precision and recall.
- Real-time Prediction: The model should be able to provide real-time predictions with low-latency and high accuracy to support timely actions.
- Deliverables: The deliverables for this project will include:

- Churn prediction model: A machine learning model that accurately predicts churned customers based on the historical data.
- Model evaluation: Evaluation of the model's performance using relevant metrics, such as accuracy, precision, recall, and F1 score.
- Real-time prediction capability: Development of a model that can provide real-time predictions with low-latency to support timely actions.
- Documentation: Detailed documentation of the methodology, findings, and recommendations for the company to take proactive measures to reduce churn and improve customer retention.

1.2 DATA CLEANING:

Upon obtaining the data, our initial focus is on the cleaning and preparation of the dataset to ensure its accuracy and reliability in facilitating optimal decision-making. To achieve this, we first load the dataset into a variable, enabling us to perform a variety of operations such as filtering, grouping, and plotting. During this process, we meticulously analyze various attributes of the data, including the number of columns, rows, features, data types, and statistical attributes.

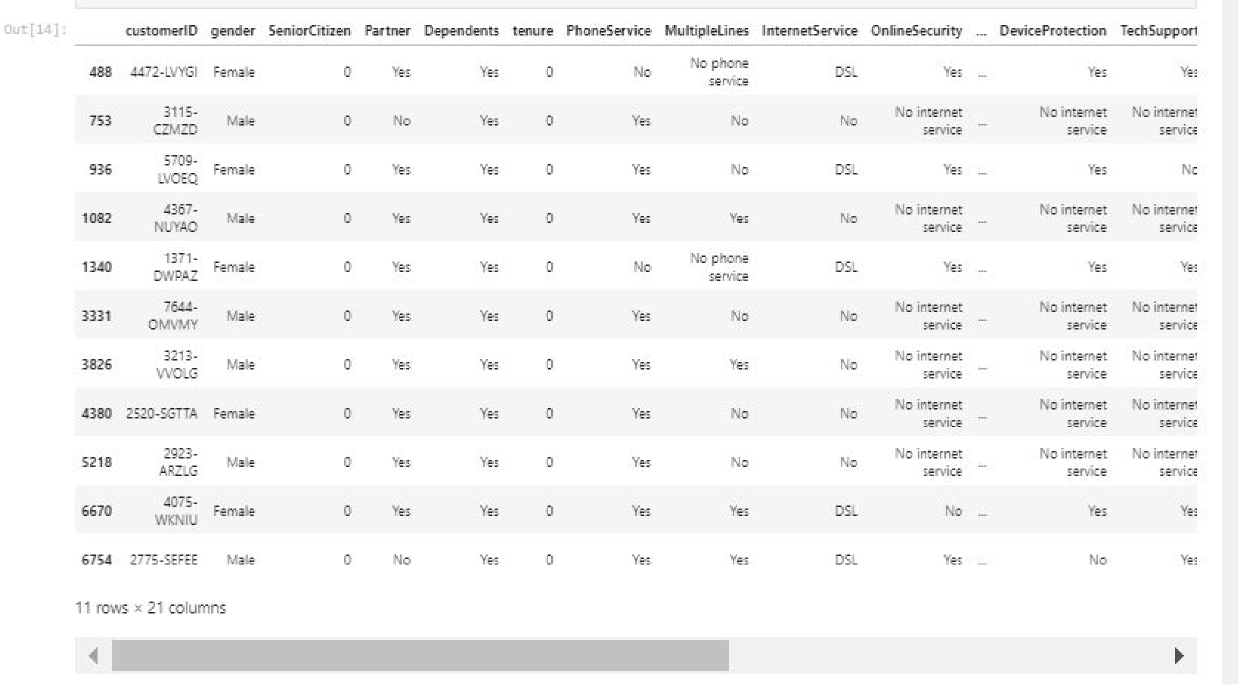
Furthermore, we examine the ratio of churned customers to non-churned customers, which stands at 27% and 73%, respectively, indicating a highly imbalanced dataset. To gain insights, we analyze the data while keeping the target values separately from other features.

Next, we create a copy of the base data to facilitate manipulation and processing and conduct a thorough check for null or missing values.

Furthermore, we examine the ratio of churned customers to non-churned customers, which stands at 27% and 73%, respectively, indicating a highly imbalanced dataset. To gain insights, we analyze the data while keeping the target values separately from other features.

Next, we create a copy of the base data to facilitate manipulation and processing and conduct a thorough check for null or missing values.

Below are the screenshots of the processing done at various stages of the project:



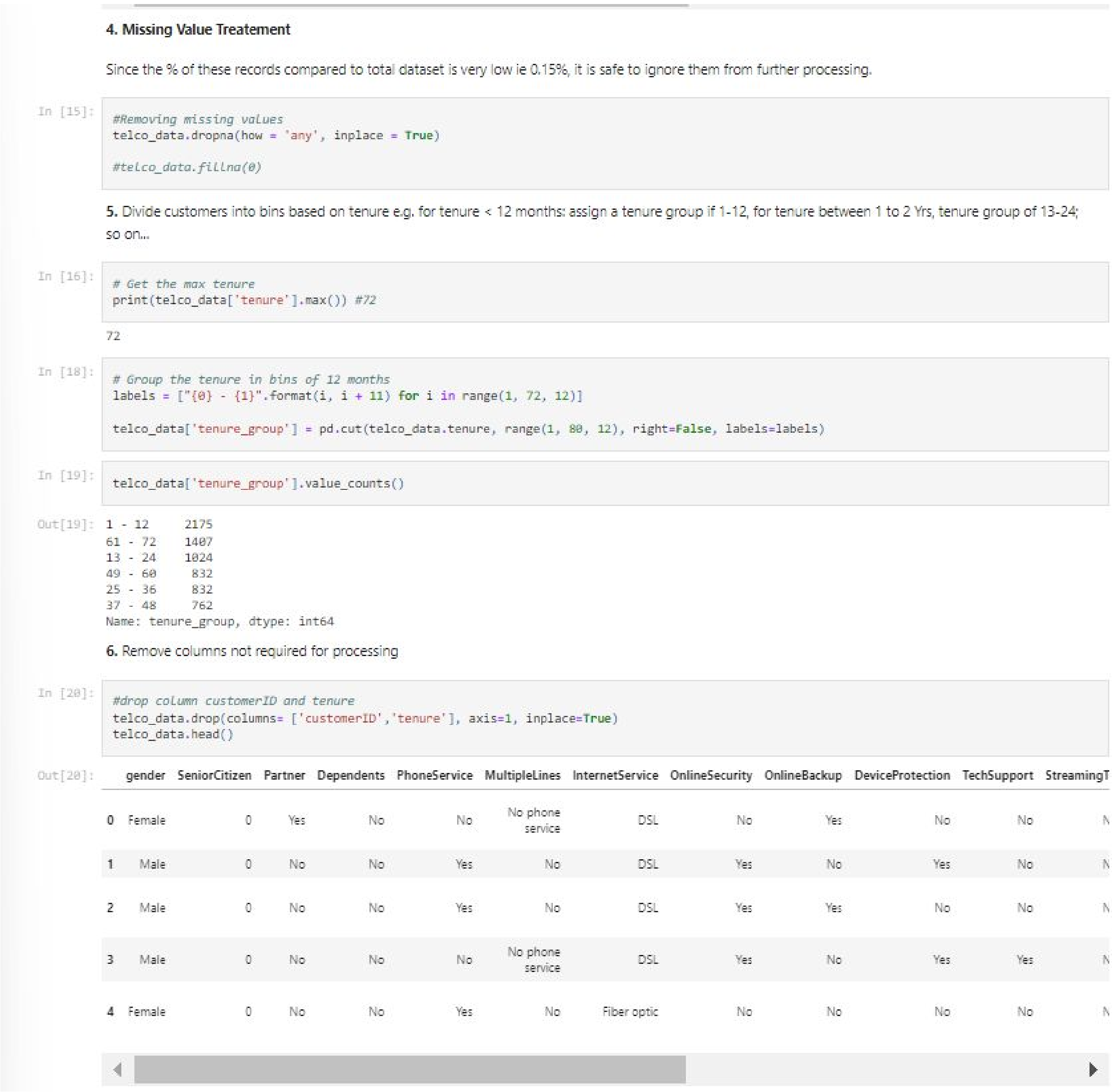
Out[14]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	Yes
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	...	Yes	No
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No internet service
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	Yes
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No internet service
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	...	Yes	Yes
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	...	No	Yes

11 rows × 21 columns

During the data cleaning process, we identified a column where the percentage of records compared to the total dataset is minimal (0.15%). As a result, we determined that it would be safe to ignore this column from further processing. Additionally, we identified another column with an excessive number of missing values, making it safe to drop the column.

To further enhance the cleanliness of the data, we removed columns such as customer ID that were randomly generated and had no logical relationship with churning. Furthermore, we grouped columns such as tenure into smaller bins to reduce the size of the dataset.



4. Missing Value Treatement

Since the % of these records compared to total dataset is very low ie 0.15%, it is safe to ignore them from further processing.

```
In [15]:
#Removing missing values
telco_data.dropna(how = 'any', inplace = True)

#telco_data.fillna(0)
```

5. Divide customers into bins based on tenure e.g. for tenure < 12 months: assign a tenure group if 1-12, for tenure between 1 to 2 Yrs, tenure group of 13-24; so on...

```
In [16]:
# Get the max tenure
print(telco_data['tenure'].max()) #72

72
```

```
In [18]:
# Group the tenure in bins of 12 months
labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

telco_data['tenure_group'] = pd.cut(telco_data.tenure, range(1, 80, 12), right=False, labels=labels)
```

```
In [19]:
telco_data['tenure_group'].value_counts()

Out[19]:
1 - 12     2175
61 - 72    1407
13 - 24    1024
49 - 60     832
25 - 36     832
37 - 48     762
Name: tenure_group, dtype: int64
```

6. Remove columns not required for processing

```
In [20]:
#drop column customerID and tenure
telco_data.drop(columns= ['customerID','tenure'], axis=1, inplace=True)
telco_data.head()
```

Out[20]:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingT
0	Female	0	Yes	No	No	No phone service	DSL	No	Yes	No	No	N
1	Male	0	No	No	Yes	No	DSL	Yes	No	Yes	No	N
2	Male	0	No	No	Yes	No	DSL	Yes	Yes	No	No	N
3	Male	0	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	N
4	Female	0	No	No	Yes	No	Fiber optic	No	No	No	No	N

1.3 EXPLORATORY DATA ANALYSIS:

After sourcing data, we next move on to the exploratory data analysis part. EDA is an approach to analyse the datasets to summarise their main characteristics in the form of visual methods and gain insights.

After cleaning the data, the next step is to visualize it. Data visualization is the process of representing data in the form of graphs or maps, making it easier to identify trends or patterns. In this study, we employed two exploratory data analysis tools, namely univariate and bivariate analysis.

Univariate analysis involves analyzing data that consists of only one variable. It is the simplest form of analysis since it deals with only one quantity that changes. The purpose of this analysis is to describe the data and find patterns within it without dealing with causes or relationships.

On the other hand, bivariate analysis involves two different variables. This analysis deals with causes and relationships, and its purpose is to find out the relationship between the two variables.

Since the target variable in this study is churn, we analyzed every variable with respect to customers who churned. First, we analyzed the categorical variables. To visualize the data, we used the countplot in the Seaborn library to gain insights.

After analyzing the graphs, we noticed that some variables, such as gender and streaming TV, had nearly equal ratios when compared to customers who churned. This suggests that there may not be a significant relationship between these variables and the target variable. However, other variables, such as tenure group and contract, exhibited highly skewed ratios when compared to customer churn. These variables are likely to be important features in the analysis.

The insights gained from the respective countplots are as follows:

1. Gender standalone has no effect on churn
2. Ratio of churned to not churned is higher in senior citizens as compared to non senior citizens
3. People who do not have partner are more likely to churn as compared to married individuals.
4. People who are dependents are less likely to churn compared to dependents

5. Ratio of churn to not churn is similar irrespective of people having phone service.

6. Ratio of churn to not churn is similar irrespective of people having multiple lines, although people having no phone service are slightly less likely to churn.

7. People having fibre optics as internet service have higher churn rate

8. People having no online security have higher churn rates

9. People having no device protectionhave higher churn rate

10. People having no tech support have higher churn rate

11. Streaming of tv and movies have seemingly no effect on churning.

12. People who have monthly contract have high churn rate

13. People with paperless billing churn more

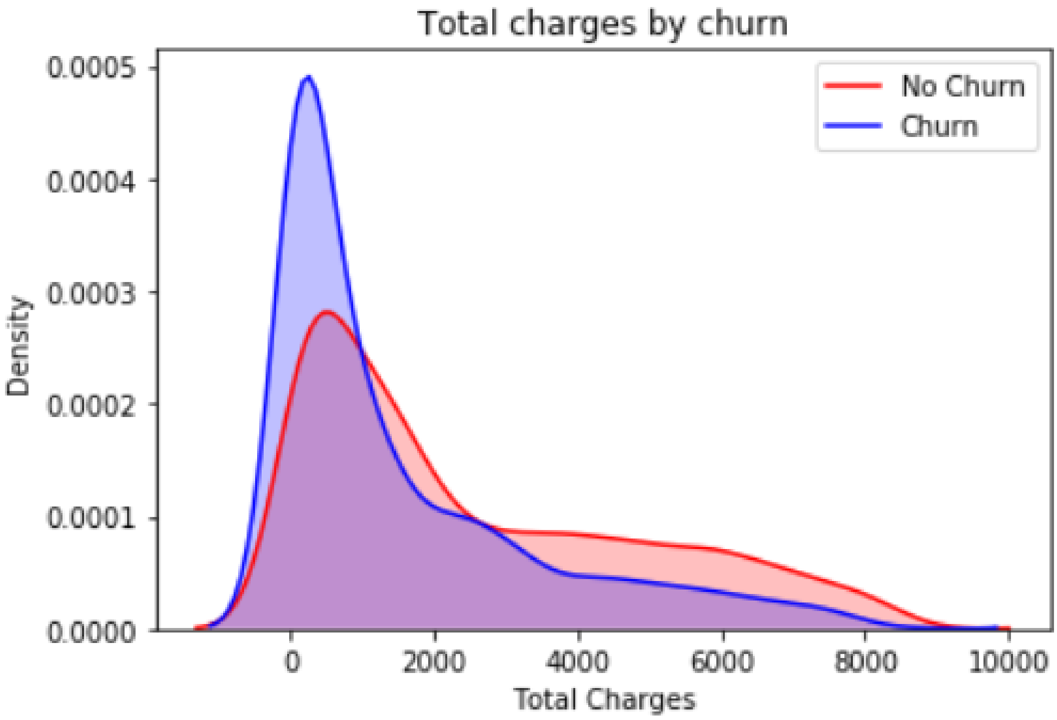
14. People having e-check as payment method have considerably high churn rate

15. People who have the tenure group 1-12 months are more likely to churn

As most of our variables are categorical, it is essential to convert them into a numerical format since machine learning models cannot work with strings. To achieve this, we used binary numeric variables by assigning the value of 1 to ”Yes” and 0 to ”No”. Additionally, we converted categorical variables into dummy numerical variables using encoding techniques such as dummy encoding or one-hot encoding.

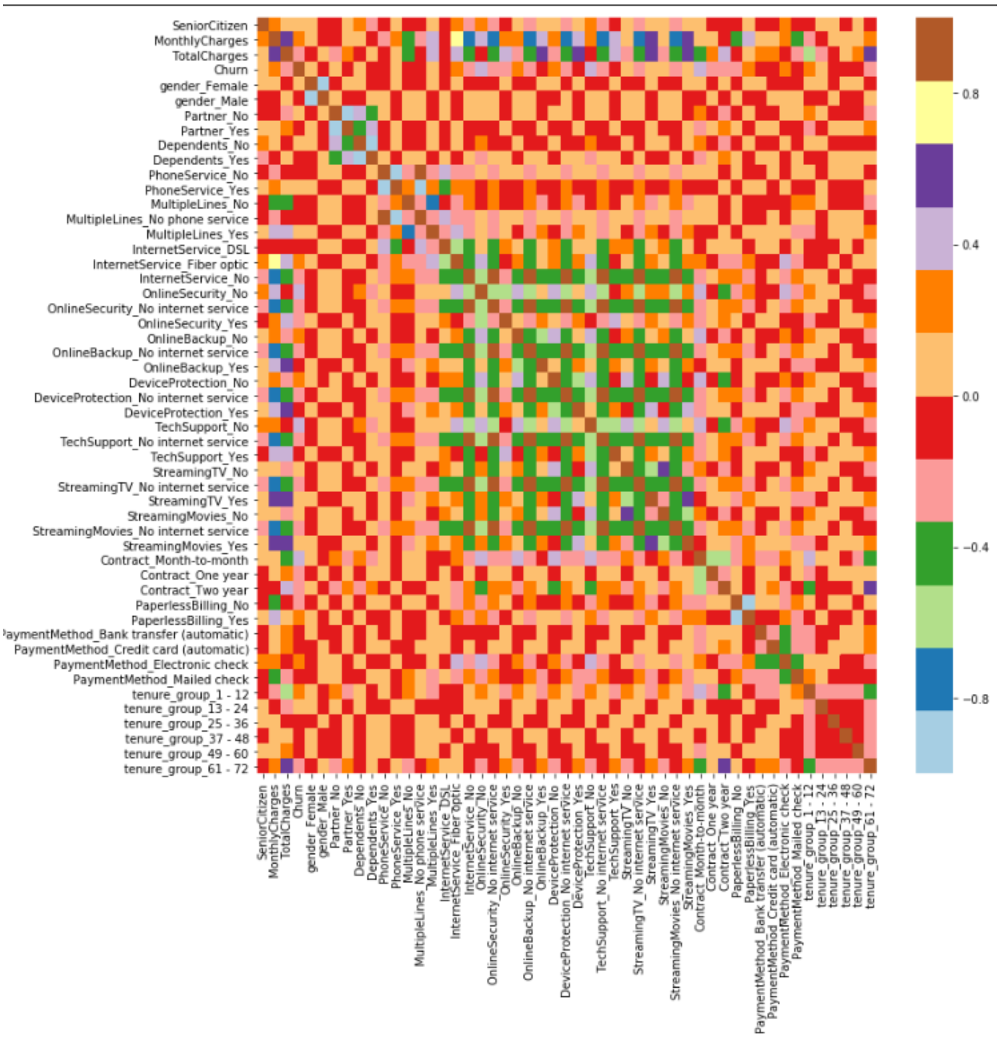
In this study, we utilized the ”get_dummies” method from Pandas to create one-hot encoded features for our categorical values. This method helped us to transform each categorical variable into multiple binary variables, with each variable indicating the presence or absence of a particular category.

Now we will conduct an analysis of our monthly and total charges numerical variables with respect to our target variables. To visualize this, we will be utilizing the kernel density estimate plot from the Seaborn library. The kdeplot function enables us to estimate the probability density function of the continuous or non-parametric curve of our dataset in one or more dimensions, providing us with a more effective method for visualizing our data.



Based on our analysis, we have found that high monthly charges are correlated with high churn rates, while low total charges are associated with higher churn. However, when we consider all three factors, namely tenure, monthly charges, and total charges, the relationship becomes clearer. Specifically, we observed that customers with a shorter tenure and higher monthly charges tend to have lower total charges. This combination of higher monthly charges, lower tenure, and lower total charges is strongly linked to high churn rates, as shown in our data visualization.

After converting our categorical variables to numerical, we utilized the corr function from the pandas library to construct a correlation matrix among the variables. The resulting matrix is presented below. To gain a more comprehensive understanding of the correlations, we can utilize the seaborn library's heatmap function to generate a graphical representation.



Derived Insights:

1. HIGH Churn seen in case of Month to month contracts, No online security, No Tech support, First year of subscription and Fibre Optics Internet

2. LOW Churn is seens in case of Long term contracts, Subscriptions without internet service and The customers engaged for 5+ years

3. Factors like Gender, Availability of PhoneService and of multiple lines have

almost NO impact on Churn

Bivariate Analysis: After conducting the univariate analysis, the next step is to perform bivariate analysis to gain a more thorough understanding of the data. Bivariate analysis is a systematic statistical approach used to examine the relationship between two variables (features/attributes) of data. In this phase, we investigated the connection between gender and other features in relation to our target variable, churn, in order to gain additional insights, as illustrated below.

1. Electronic check medium are the highest churners

2. Contract Type - Monthly customers are more likely to churn because of no contract terms, as they are free to go customers.

3. No Online security, No Tech Support category are high churners

4. Non senior Citizens are high churners

Conclusion of Exploratory Data Analysis:

1. Churn rate is surprisingly high at lower Total Charges.

2. However, Higher Monthly Charges at lower tenure results into lower Total Charge.

3. Higher Monthly Charge, Lower tenure and Lower Total Charges are linked to High Churn

4. HIGH Churn seen in case of Month to month contracts, No online security, No Tech support, First year of subscription and Fibre Optics Internet

5. LOW Churn is seen in case of Long term contracts, Subscriptions without internet service and customers engaged for 5+ years.

6. Factors like Gender, Availability of PhoneService have almost NO impact on

Churn

7. Electronic check medium are the highest churners
8. Contract Type - Monthly customers are more likely to churn because of no contract terms, as they are free to go customers.
9. No Online security, No Tech Support categories are high churners
10. Non senior Citizens are high churners

1.4 MODELLING:

After doing the Exploratory Data Analysis (EDA), we move forward to the modelling part. For that purpose, we have divided the data frame into a churn variable y containing only 1 or 0 values denoting whether a customer has churned or not respectively. We then made another variable x which contains every feature of the data frame except churn value. Then we divided the data into the training dataset and testing dataset.

We will apply the following machine learning model to test our accuracy of the data:

(i) Decision Tree Classifier:

It is a popular machine learning algorithm used for classification tasks. It creates a tree-like model of decisions and their possible consequences. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

It finds the feature that provides the best split of the data into separate classes. This is done using measures like Information Gain or Gini Impurity. Here we will use the Gini impurity measure for splitting our data nodes.It is a non-parametric, supervised learning algorithm and is hierarchical in structure. As decision tree can lead to low bias by overfitting the training model by going to maximum depth we limit height of the tree to 6.

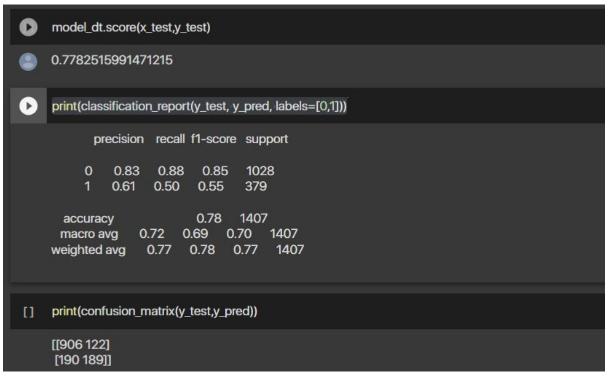
$$E(S) = -p_{(+)}\log p_{(+)} - p_{(-)}\log p_{(-)}$$

$$\text{Information Gain} = E(Y) - E(Y|X)$$

Gini Index

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

Then we got our result as the classification report for the above mentioned model.



```
model_dt.score(x_test,y_test)

0.7782515991471215

print(classification_report(y_test, y_pred, labels=[0,1]))

          precision   recall  f1-score   support

   0        0.83      0.88      0.85      1028
   1        0.61      0.50      0.55       379

accuracy                        0.78      1407
macro avg   0.72      0.69      0.70      1407
weighted avg 0.77     0.78      0.77      1407

[]  print(confusion_matrix(y_test,y_pred))

[[906 122]
 [190 189]]
```

As we can see the results in the above image, our f1-score is 0.85 for non-churners and 0.55 for churners, and our accuracy is nearly 80As we can also see that the accuracy is quite low, and as it's an imbalanced dataset, we shouldn't consider Accuracy as our metrics to measure the model, as Accuracy is cursed in imbalanced datasets.

Hence, we need to check recall, precision & f1 score for the minority class, and it's quite evident that the precision, recall & f1 score is too low for Class 1, i.e. churned customers.

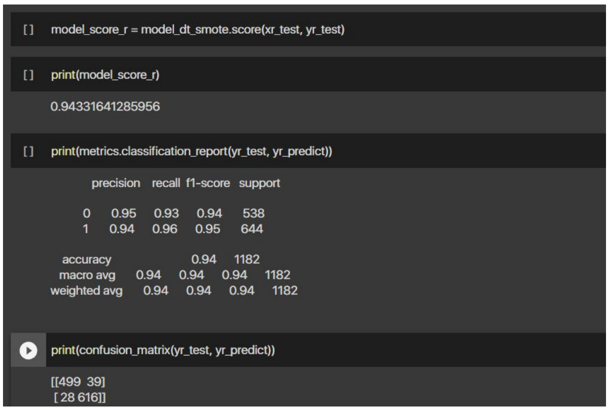
Hence, moving ahead to call SMOTEENN (UpSampling + ENN). SMOTEENN (Up-

Sampling + ENN):

SMOTEENN is a combination of two different techniques used for handling imbalanced datasets in machine learning, namely SMOTE and Edited Nearest Neighbors (ENN).

SMOTE (Synthetic Minority Over-sampling Technique) is an oversampling technique that generates synthetic examples of the minority class by interpolating between existing examples. The aim is to create a balanced dataset by increasing the number of examples in the minority class without introducing bias.

Then we got our result as the classification report for the above mentioned model:



```
[ ]  model_score_r = model_dt_smote.score(xr_test, yr_test)

[ ]  print(model_score_r)

     0.94331641285956

[ ]  print(metrics.classification_report(yr_test, yr_predict))

               precision   recall  f1-score   support

           0       0.95      0.93      0.94       538
           1       0.94      0.96      0.95       644

    accuracy                           0.94      1182
   macro avg       0.94      0.94      0.94      1182
weighted avg       0.94      0.94      0.94      1182

     print(confusion_matrix(yr_test, yr_predict))

     [[499  39]
      [ 28 616]]
```

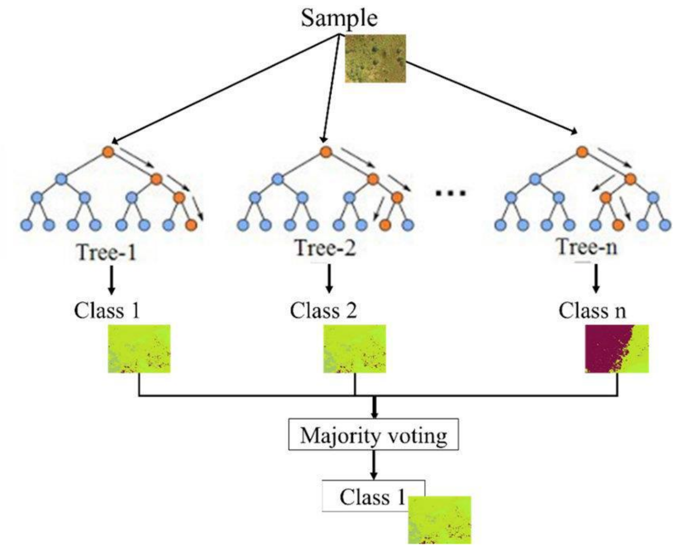
Now we can see quite better results, i.e. Accuracy: 94%, and a very good recall, precision & f1 score for minority class.

(ii) Random Forest Classifier:

Random Forest is a Machine Learning Algorithm used for classification, regression and other tasks. It belongs to the family of ensemble methods, which combine multiple weak learners to create a strong learner.

In a random forest classifier, multiple decision trees are trained on different subsets of the data using a random selection of features for each tree.

It is based on ensemble learning, which integrates multiple classifiers to solve a complex issue and increases the model's performance. Random forests are bagged decision tree models that split on a subset of features on each split.it works on the principle of bootstrapping where randomly sample our data and feed it to numerous decision tree.

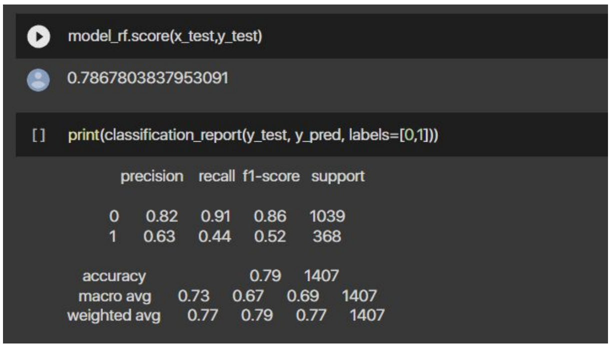


Advantage of use random forest are

1. Reduced risk of overfitting

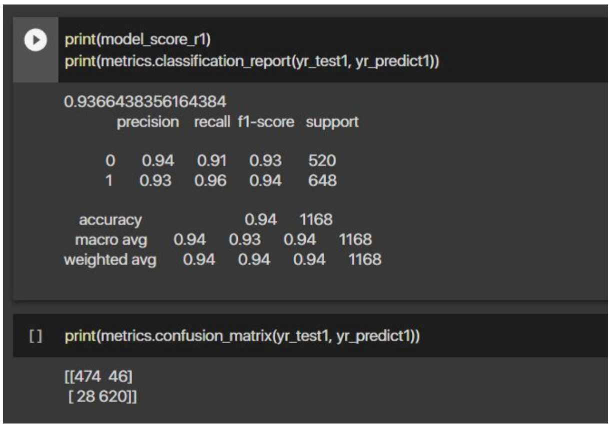
2. provides flexibility: Since random forest can handle both regression and classification tasks with a high degree of accuracy.

Then we got our result as the classification report for the above mentioned model:



```
model_rf.score(x_test,y_test)

0.7867803837953091

[ ] print(classification_report(y_test, y_pred, labels=[0,1]))

            precision   recall  f1-score   support

       0       0.82      0.91      0.86      1039
       1       0.63      0.44      0.52       368

accuracy                           0.79      1407
   macro avg       0.73      0.67      0.69      1407
weighted avg       0.77      0.79      0.77      1407
```

As we can see the results in the above image, our f1-score is 0.86 for non-churners and 0.52 for churners, and our accuracy is nearly 79Now we created SMOTEEN for the above Forest Classifier Model Then we got our result as the classification report for the above mentioned model:



```
print(model_score_r1)
print(metrics.classification_report(yr_test1, yr_predict1))
```

```
0.9366438356164384
              precision    recall  f1-score   support

           0       0.94      0.91      0.93       520
           1       0.93      0.96      0.94       648

    accuracy                           0.94      1168
   macro avg       0.94      0.93      0.94      1168
weighted avg       0.94      0.94      0.94      1168
```

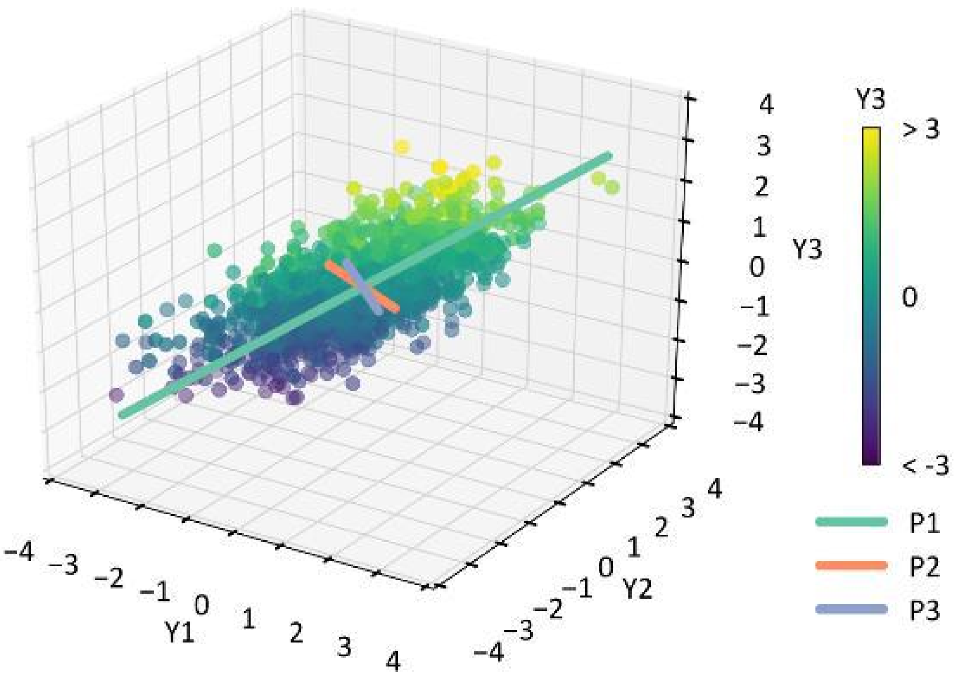
```
print(metrics.confusion_matrix(yr_test1, yr_predict1))
```

```
[[474  46]
 [ 28 620]]
```

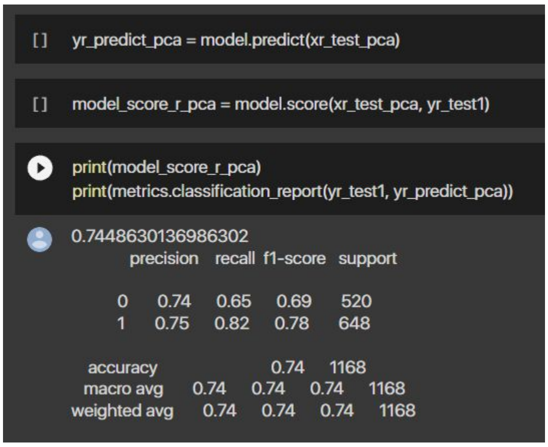
Using RF Classifier, we got quite better results i.e. Accuracy: 94% which is quite better than what we got using Decision Tree.

(iii) Principal Component Analysis:

As we have around 20 features in the database there is high chances that our model may suffer from overfitting increasing the variance. Therefore it is sensible to apply PCA. By applying the above method in our model, the model: 1. Reduces the data dimension 2. Reduces variance by solving problem of overfitting 3. Provides better visualization of the model



Then we got our result as the classification report for the above mentioned model:



```
[ ]  yr_predict_pca = model.predict(xr_test_pca)

[ ]  model_score_r_pca = model.score(xr_test_pca, yr_test1)

     print(model_score_r_pca)
     print(metrics.classification_report(yr_test1, yr_predict_pca))

     0.7448630136986302
               precision    recall  f1-score   support

           0       0.74      0.65      0.69       520
           1       0.75      0.82      0.78       648

    accuracy                           0.74      1168
   macro avg       0.74      0.74      0.74      1168
weighted avg       0.74      0.74      0.74      1168
```

Here we can see that the accuracy after applying the PCA was 74% and f1-score is 0.69. With PCA, we couldn't see any better results, hence let's finalise the model which was created by RF Classifier, and save the model so that we can use it in a later stage.

1.5 RESULT:

Conclusion of Exploratory Data Analysis:

1. Churn rate is surprisingly high at lower Total Charges.
2. However, Higher Monthly Charges at lower tenure results into lower Total Charge.
3. Higher Monthly Charge, Lower tenure and Lower Total Charges are linked to High Churn
4. HIGH Churn seen in case of Month to month contracts, No online security, No Tech support, First year of subscription and Fibre Optics Internet
5. LOW Churn is seen in case of Long term contracts, Subscriptions without internet service and customers engaged for 5+ years.

6. Factors like Gender, Availability of PhoneService have almost NO impact on Churn

7. Electronic check medium are the highest churners

8. Contract Type - Monthly customers are more likely to churn because of no contract terms, as they are free to go customers.

9. No Online security, No Tech Support categories are high churners

10. Non senior Citizens are high churners

After doing EDA we move forward to apply various machine learning algorithm for model prediction part. After applying classifier like Decision Tree Classifier we got our f1-score as 0.85 for non-churners and 0.55 for churner class and accuracy of about 79 %. But the accuracy obtained was quite low and we move for applying Smotteen ,an upsampling technique and improved our accuracy to 94%. Similarly we applied Random Forest Classifier from which we got an accuracy of about 79% and then by applying the upsampling method we reached to an accuracy of 94%. We also applied other algorithms like PCA but couldn't improve our accuracy further and so we finalize our model which was created using RF Classifier.

REFERENCES

1. https://towardsdatascience.com/end-to-end-machine-learning-project-telco-customer-churn-90744a8df97d

2. https://medium.com/@islamhasabo/predicting-customer-churn-bc76f7760377

3. https://www.javatpoint.com/machine-learning-random-forest-algorithm

4.https://scikit-learn.org/stable/modules/tree.html: :text=Decision

5.https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

6.https://www.paddle.com/resources/churn-prediction

7.https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

8.https://www.section.io/engineering-education/seaborn-tutorial/

9.https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

10.https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

11.https://www.geeksforgeeks.org/ml-principal-component-analysispca/