University of Messina

A blue and white logo

Description automatically generated with low confidence

Bachelor of Data Analysis

ACADEMIC YEAR - 2022/2023

Machine Learning

(Project report)

Supervisor: Students:

Prof. Giacomo Fiumara Sahil Nakrani

**Understanding the Dataset**

The dataset consists of 3749 entries with 17 columns. ( But here we are not considering 2 columns which are ‘Unnamed: 0’ and ‘CustomerId’. Because which are just indexes for the entries. )

In this dataset, there is two columns ‘Unnamed: 0’ and ‘CustomerID’ which are only showing the index. So, I have removed it to not have unnecessary feature.

A black background with yellow text

Description automatically generated

**Basic Information and Statistical Details**

A screenshot of a computer

Description automatically generated

A screenshot of a black screen

Description automatically generated

**Data Preprocessing**

**Handling the missing values**

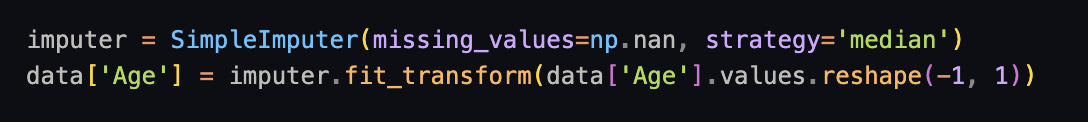
We utilize both the functions “isnull()” to inspect null values.

**A computer screen shot of a black background

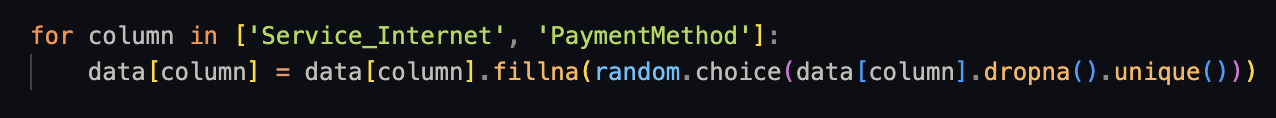
Description automatically generated**

Here, feature ‘Age’ is numerical feature and [ ‘Service\_Internet’, ‘PaymentMethod’ ] are categorical features.

* Replacing the numerical missing values with the median of the column.

****

* Replacing the categorical missing values with the random instance of the column.

****

Here, we can also replace categorical feature’s missing values with mode ( most frequent instance ) but we have replaced with random instance. Just because already mode is having majority of distribution and as per me, it is not good to put again the mode to let the model to rely on single instance ( category ).

**Label Encoding for the categorical features**

For the Exploratory Data Analysis (EDA) and for the model training part. For example terraforming the values of categorical value ‘yes’ and ‘no’ to 1 and 0.

First, I have listed all the categorical features. I’m using ‘**LabelEncoder()’** function to encode the categories of the features.

**A screen shot of a computer

Description automatically generated**

**Handling the Outliers**

Outliers are datapoints that deviate significantly from the rest of the dataset and can have a substantial impact on analysis.

A white graph with black lines

Description automatically generated

In the above box-plot, you can see there are outliers in two right sided features which are ‘MonthlyCharges’ and ‘TotalCharges’. And we are going to handle it with the technique called inter quartile range.

So, first I have made the function to handle the outliers and then I will apply it on all the numerical features and I will save the index of the outlier.

A computer screen shot of a program

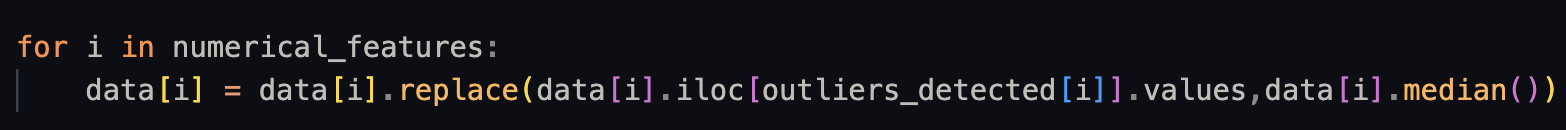
Description automatically generated

For example :

**A black background with white numbers

Description automatically generated**

Then, I have just replaced that detected outliers with the median of the column. We can also remove the outliers, but since our dataset is small so it is better to replace it with middle value not to remove.

****

Let’s see now the box-plot.

**A white graph with black lines

Description automatically generated**

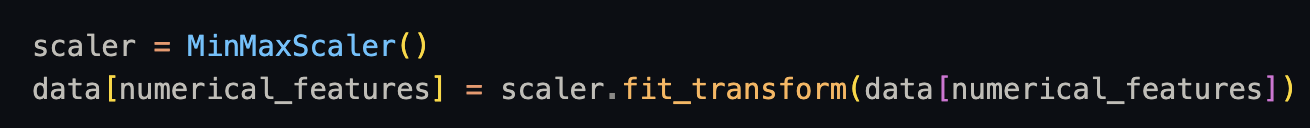
**Scaling the Feature**

Before scaling the data, features are not in one range. They are in different range which is not good for training phase.

**A white graph with blue lines

Description automatically generated**

here, I’m using a MinMaxScaler() to scale the data not a Standard Scaler. Because we have some features which is showing time period and standard scaler is converting the range between -1 and 1. Negative values are not usual for the time period. That’s why here I have used MinMaxScaler to have range between 0 and 1.



Here is the box-plot after scaling the numerical features.

**A diagram of a diagram

Description automatically generated with medium confidence**

**EDA ( Exploratory Data Analysis )**

First, we are going to make a distribution graph for the numerical features using histogram.

**A graph of different numbers

Description automatically generated with medium confidence**

By this histogram, we can see the distribution, frequencies, and also skewness in the distribution. For example, in the distribution of the TotalCharges, we can see that it is left side skewed, so we can say that majority of people have total charges less than 3000.

Now here is the relation between a numerical features and the target feature which is Churn. This relation is visualized with the box-plot.

**A diagram of a box plot

Description automatically generated**

**A diagram of a box plot

Description automatically generated**

**A diagram of a box plot

Description automatically generated**

**A diagram of a box plot

Description automatically generated**

Also I have visualized the relation with the help of pairwise scatter plot. Where we can identify the relational trends, actual trends in the data, and also distribution and outliers. Here, I have mentioned the pairwise scatter plot below.

**A screenshot of a graph

Description automatically generated**

Then, let’s create a heatmap of the correlation matrix. This is very important step of EDA because, by correlation matrix, we can identify the relation between the feature and other features and asl between the feature and target feature.

By correlation matrix, we can also look the percentage of dependencies of the features on other features. Because, correlation coefficient ranges between -1 and 1. Where 1 mean strongly positive correlation and -1 means strongly negative correlation.

So, let’s visualize the heatmap of the correlation matrix.

A graph of a number of data

Description automatically generated with medium confidence

**Feature Engineering**

I have already scaled the features, and also have put the labels for the categorical features.

But now I’m going to create some extra additional features which can effect this learning process or can affect the predictions.

Combine Existing Features :

* **MonthlyCharges\_to\_Tenure :** identifies high-cost users who might be at risk of churn.
* **TotalCharges\_to\_Tenure :** highlights customers who have spent a lot over their tenure and might be more sensitive to service issues.

Domain Knowledge :

* **TotalServices :** helps to identify customers having reliant on the company’s services and might churn if dissatisfied.
* **isSeniorCitizen :** senior citizens might have different churn behaviour.

**A screen shot of a computer program

Description automatically generated**

**Modelling and Evaluation**

Frist we are going to split the data into target variables and other features.

A black background with yellow and green text

Description automatically generated

I have tried to train model now, but it is overfitting. May be because we have very less data and three features with high missing values. So, that’s why I have used feature selection technique to select most important features.

**A computer screen with text

Description automatically generated**

So, we are selecting 10 most important features from our dataset to train the model on.Then, we are splitting the data into training and test sets using train\_test\_split() function.

****

Here, test\_size=0.3 means 30% testing set and 70% training set. Random\_state=42 means every time it is using same sample set for training and testing set.

Here, I’m testing the model by creating 4 different models :

* Random Forest Classifier
* Logistic Regression Classifier
* Gradient Boost Classifier
* Decision Tree Classifier

But first, I’m making a common function to train all the models by that function. Which is mentioned below.

A computer screen shot of a program code

Description automatically generated

Here, I’m also using a GridSearchCV() function to split the dataset in several part and train on each split to get best model and best parameters which will suitable to improve the accuracy of the model.

And by this function as you can see, I’m printing best parameters, classification report, confusion matrix, and returning the best model.

A computer screen shot of text

Description automatically generated

By the upper function, we can see the importance of the features for that specific model to train the model. And by below mentioned function, by checking the accuracy of the train and test, it is overfitting or not.

A screen shot of a computer code

Description automatically generated

And here is the example that how I’m training the model with some range of parameters, and with the help of functions which I made before. This is the example of the random forest classifier and below this I have mentioned result of this model. Like this I have trained other models and got the results which you can see in the code file.

A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A chart of confusion matrix

Description automatically generated

A black background with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

But in the Logistic Regression, I have noticed that it is giving reliable result.

A screenshot of a computer screen

Description automatically generated

A chart of confusion matrix

Description automatically generated

A black background with white text

Description automatically generated

**Conclusion**

In this project, we undertook a comprehensive approach to predict customer churn using various machine learning models. We started with data preprocessing and exploratory data analysis to understand the key characteristics and patterns within the dataset. Following this, we implemented and evaluated several machine learning models, including Logistic Regression, Decision Trees, Random Forest, and Gradient Boosting, using a range of performance metrics such as accuracy, precision, recall, F1-score.

Our initial findings revealed that while some models performed well on training data, they exhibited signs of overfitting when evaluated on the testing data. This was evident from the significant discrepancies between training and testing accuracies. To address this, we applied cross-validation techniques and observed the models' performance across multiple folds, ensuring robust and reliable performance estimates. But nothing changes that much so as per my opinion, there is less data in the dataset that’s why it may overfit in this training process.

Further, we engaged in hyperparameter tuning using GridSearchCV to optimize the models' performance. This process allowed us to refine the model parameters and achieve improved evaluation metrics, reducing the overfitting issue significantly. Specifically, the Random Forest model, after tuning, demonstrated a balanced trade-off between bias and variance, making it one of the top performers in our analysis.

As per the dataset target feature, these findings offer strategic insights into customer behaviour, enabling the organization to implement targeted interventions to reduce churn and enhance customer retention.