

Predicting Customer Churn for a Telecommunications Company

Machine learning project

Professor Giacomo Fiumara

Project by Mujibullah Rabin (539269)

University of Messina

Machine learning

Predicting Customer Churn for a Telecommunications Company.

CONTENTS:

* INTRODUCTION TO THE DATASET
* INTRODUCTION OF THE PROJECT
* PROJECT DETAILS
* CONCLUSION

DATASET INTRODUCTION

The dataset used for the project consists of customer information from a telecommunications company. And the dataset is including 3749 entries with 17 columns.

* **Unnamed: 0**: An integer index column, not directly relevant to the analysis.
* **CustomerID**: A unique identifier for each customer.
* **Age**: The age of the customer.
* **Gender**: The gender of the customer (Male/Female).
* **Tenure**: The number of months the customer has been with the company.
* **Service\_Internet**: The type of internet service the customer has.
* **Service\_Phone**: Indicates whether the customer has phone service (Yes/No).
* **Service\_TV**: Indicates whether the customer has TV service (Yes/No).

And many more…

And in this dataset, we have the string, float, character which i have analyse the dataset by considering them.

we can see that in the dataset some columns have missing values like ‘age’ and ‘payment method’ where these missing values must be addressed in the project.  
The dataset features both categorical and numerical variables, providing a rich set of attributes for predictive modelling.

Introduction to the project

This project analyses the dataset and aims to analyse the customer churn in telecommunications company using machine learning. And it finds out which customers are likely to churn.

For this project I am using google colabs which for machine learning google colabs would be better option and easy to work with machine learning projects and like on one of good point is the google colabs has all the libraries no need install again.

In this project I have done several stages like:

* Data processing
* Data cleaning
* Exploratory data analysis
* Feature engineering
* Model evaluation

Whereby these stages we understand the data we clean the data like addressing the to the null columns or another problem with the dataset.

And we visualize the data for better understanding, by following this in the last we use classifications methods like random forest, linear regression or any other classifications according to our dataset, and predict the customers churn in the telecommunication company using the machine learning.

The dataset includes the information of the customer and services they have subscribed. By analysing this data, we can understand customer behaviour better and develop strategies to keep them from leaving.

PROJECT

In the first part of the project all the libraries are imported that Is necessary for the project.

A screenshot of a computer program

Description automatically generated

* The pandas as pd: Provides data structures and data analysis tools. Used for data manipulation and analysis, especially with data in table format (Data Frames).
* NumPy as pd: Adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
* Seaborn: A data visualization library based on matplotlib that provides a high-level interface for drawing attractive statistical graphics.

And all the other libraries are to together provide a comprehensive toolkit for data analysis, visualization, preprocessing, and building machine learning models.

Which all the classification models we use we need to import their libraries or for visualizing of the box plot or any other kind of the plot we need import matplotlib libraries.

Now in the third part I will start with cleaning the data like dropping the unnecessary columns

In the second I have written the code to be able read the dataset from the google colabs itself with giving the path of it

A screenshot of a computer

Description automatically generated

Where it reads the dataset and to see the data is correctly reading or not and to see the head of data I used the data. head () command as we can see the data is successfully reading.

now we will try to drop the columns that are not important.

A screenshot of a computer

Description automatically generated

Here I have dropped unnamed and customer id columns where it was not important.

Also, with dropping the unnecessary columns now I have checked for missing values to identify columns that require imputation or removal.

And with data type types of each column displayed to understand how to handle them in subsequent steps. Like as they are float, integer, object and else.

Now I need to fix the missing values by replacing them with median and mode.

A screen shot of a computer

Description automatically generated

Here Missing values in the Age column are replaced with the median value and service internet and payment method is replaced with the mode. The dataset is rechecked for any remaining missing values.

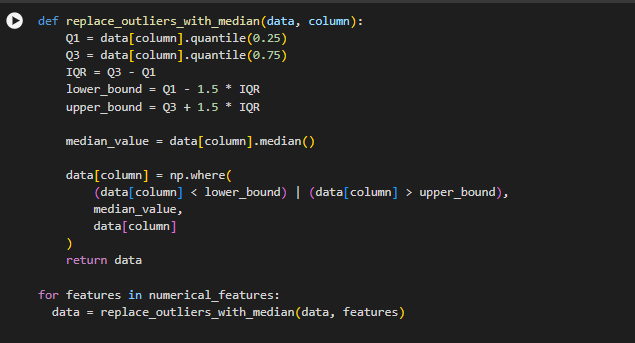
As the data is cleaned and managed, I have plotted the numerical features as float and integer using the boxplot for better understanding

A screenshot of a computer screen

Description automatically generated

So, we plotted the numerical features using the box plot. Box plots are used to visualize numerical features and identify potential outliers.

In this part we manage the outliers with the median of the data.



Here a function is defined and applied to replace outliers in numerical features with the median value.

Then I have plotted the categorical features for better understanding of and visualize it using the boxplot

A screenshot of a computer screen

Description automatically generated

Encoding the features, Categorical features are encoded using LabelEncoder to convert them into numerical format for modeling.

A screenshot of a computer

Description automatically generated

And over here I have plotted the encoded features.

In the exploratory data analysis (EDA) phase of a machine learning project, it's essential to understand the relationships between different features and the target variable.

A screenshot of a graph

Description automatically generated

**Histogram and KDE Plots**: Visualize the distribution of each numerical feature to understand their spread and central tendency.

So here in these plots it visualizes A screenshot of a computer screen

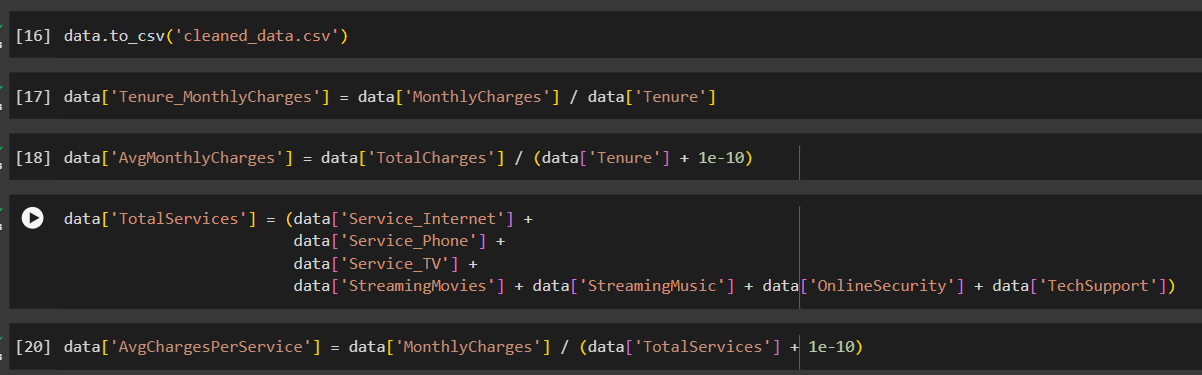
Description automatically generatedthe relationships between features and churn.

**Box Plots Grouped by Churn**: Compare the distribution of numerical features for customers who have churned versus those who have not to identify significant differences.

Now we will go with the correlation matrix where we will visualize it in the heatmap where it will show the correlation between features.  
The correlation matrix is a crucial tool in exploratory data analysis (EDA) for understanding the relationships between features in your dataset.

A screenshot of a computer

Description automatically generated

And we use the feature engineering where we will create new features to improve the performance of the modelling.

New features are engineered to potentially improve model performance by creating combinations and averages of existing features.

After feature engineering we will start with the modelling.

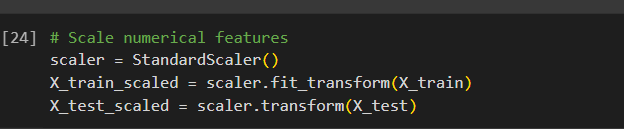
First, we will split the data to train and test set.

A screenshot of a computer

Description automatically generated

So as here in the code the dataset is split into training and testing sets for model training and evaluation.

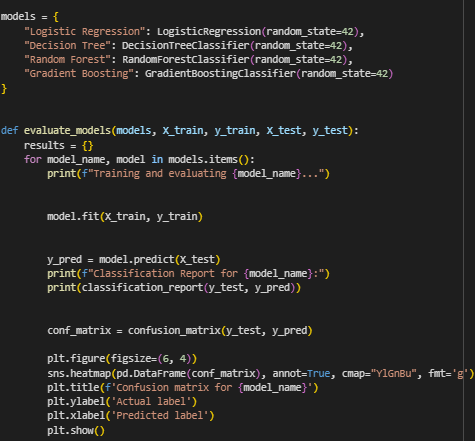
In machine learning feature scaling is important to ensure they contribute equally to the model.



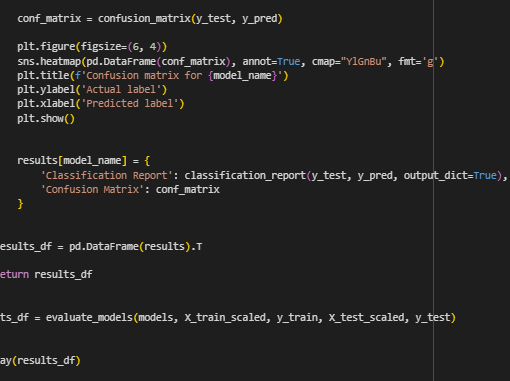
So here I have applied scaling on numerical feature to standardize the numerical features.

Now I will use default models  
I have used 4 models according to the project guidline

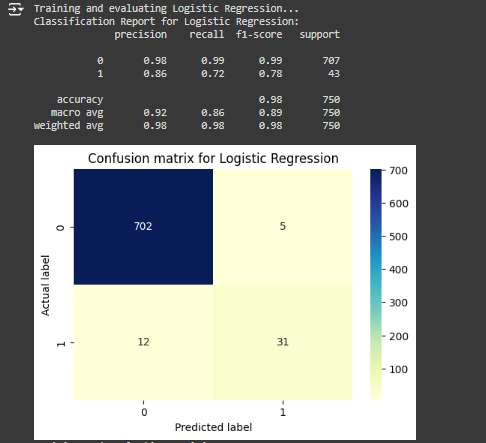
1. Logistic regression
2. Decision tree
3. Random forest
4. Gradient boosting



So here I used the defuilt models without hypertunning so then I can use the model with grid search and hypertunning to see the differences.



After applying the models here I visualized all models I have used.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

In this evaluation, we used the default parameters for each model to establish baseline performance. These default settings provide a good starting point for understanding the capabilities and limitations of each model before applying more advanced techniques like hyperparameter tuning.

now I have used the grid and models

In the context of machine learning, a "grid" refers to a set of hyperparameters that we want to explore to find the best configuration for a given model.

Hyper parameters are parameters that they don’t learn from the data instead the set prior the training set.

A screen shot of a computer program

Description automatically generated

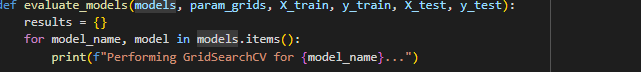
Here in each param\_grids correspond to a model and specifies the hyperparameters and their respective values to be tested.

A screenshot of a computer

Description automatically generated

Here I have initialized all the models.

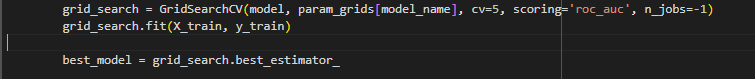
Then I have used a function to evaluate the models



The evaluate\_models function is designed to train, tune, and evaluate each model using GridSearchCV for hyperparameter tuning and cross-validation.

The process of exploring these hyperparameter grids is called "grid search". GridSearchCV from the sklearn.model\_selection

Now for each model I have initialized the grid search.



Here grid search is initialized, and it gets the best model for grid search.

After the best model is has been identified using gridsrearchCV it is used to make predictions on the test dataset:

A screen shot of a computer code

Description automatically generated

Now we can visualize it using the confusion matrix to provide the detail of each model’s performance on each class.

A computer screen shot of a code

Description automatically generated

Then the results dictionary is converted into a DataFrame for better visualization and analysis:

The evaluation function is executed with the specified models, parameter grids, and scaled training and testing datasets

Finally, the results DataFrame is displayed

A screenshot of a computer screen

Description automatically generatedAnd here are the result with advance technique hyperparameter tunning:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

In conclusion:

This project aimed to predict customer churn for a telecommunications company by leveraging machine learning techniques. The dataset, comprising 3749 entries and 17 columns, included a variety of customer attributes, such as age, tenure, and service details. The following steps were undertaken to achieve the project's objectives:

* Data Preprocessing:
* Exploratory Data Analysis (EDA):
* Feature Engineering
* Model Training and Evaluation:

Among the evaluated models, the Random Forest and Gradient Boosting classifiers demonstrated superior performance in predicting customer churn, highlighting their effectiveness in handling this type of classification problem.