The report on building a binary classification model on the "Telco Customer Churn" dataset

Abstract:

The purpose of this project is to build a binary classification model using the Telco Customer Churn dataset. The goal is to predict whether a customer will churn, meaning they will discontinue their subscription, or not. The dataset contains a variety of features that provide information about the customers, including gender, type of contract, available services, etc. By analyzing the dataset and developing a model, we aim to accurately classify customers as churned or non-churned. The outcome of this project will assist the telecommunications company in identifying customers at risk of churn.

Data discreaption:

This dataset has 7043 rows and 21 columns which the last column is target of the dataset. In the following the summary of the features of dataset are mentioned:

1. Costumer ID
2. gender: (Female, Male) Whether the client is a woman or a man.
3. SeniorCitizen: Indicates whether the client is an older person (0, 1).
4. Partner: Indicates whether or not the client is partnered (Yes, No).
5. Dependents: Indicates whether the client is supported by others (Yes, No).
6. tenure: The length of time a customer has been a customer of the business (multiple different number values).
7. Phone service: If the client has a phone service, it is indicated by the words “Yes” or “No.”
8. MultipleLines: Whether the customer has more than one line (no phone service, no service, yes service).
9. InternetServices: Whether the client has a subscription to the company’s Internet service (DSL, Fiber optic, or No).
10. OnlineSecurity: Indicates if the client has access to online security (Internet service available, No, Yes).
11. OnlineBackup: Indicates whether or not the client has an online backup (Internet service unavailable, No, Yes).
12. DeviceProtection: Indicates whether the client has device protection (Internet service not available, Not Available, Yes).

13. Tech support: Whether the customer has access to tech help (no internet service, no, yes).
14. Streaming TV: Whether the customer has access (no internet service, no, yes).
15. Streaming movies: Indicates whether or not the client offers or has access streaming movies (no internet service, no, yes).
16. Contract: The type of existing contract for the customer (Month-to-Month, One-Year, Two-Year).
17. PaperlessBilling: Whether the client uses paperless billing (Yes, No).
18. PaymentMethod: The chosen payment method by the consumer (credit card, bank transfer, electronic check, paper check).
19. MontlyCharges: The monthly charge made to the consumer (various numeric quantities).
20. TotalCharges (many different numeric values): The total amount charged to the consumer.

Clean Data:

To clean dataset DataFrame.info method was used. In the first step, I dropped the customerID column. IT has no impact on analysis.The next step involved checking for missing values and data types.. I found that the TotalCharges column has 11 missing value which I remove them because the number of rows with thisfeature was limit and droping them was the easiest and effectivenes way to solve this issue. In terms of data type, I got that the type of TotalCharges, and SeniorCitizen is object and float64, in that order. I changed them to float64(because TotalCharges is number and its type should be float64) and object (as SeniorCitizen shows a costumer is old or young, respectively. Moreover,df.duplicated().any().sum() is used to check for duplicate rows in the DataFrame. By performing these data cleaning steps, the dataset was prepared for further analysis and modeling, ensuring that missing values were handled appropriately and the data types were aligned with the nature of the variables.

Data Analyze:

To get useful information about the dataset some processes were done on the dataset such as exploring the target column showing that 73.46 % of customers not churned and 26.54 churned, or 50.45% of customers were male and 49.52% were female which were almost equal. In addition, the relationship between MonthlyCharges, TotalCharges, and Churn was checked and it demonstrates when the amount of monthly charges is between 70 to 105 the churn increases rapidly, however, a lot of customers stay in getting services from this company, when the amount of their total charge going high. Moreover, I looked at the relationship between the contract and churn and I noticed the longer duration of the contract (two years) led to decreased churn figure 1. Also, the relationship between SeniorCitizen and OnlineSecurity was examined and it displayed that just 3% of customers who do have not internet service are seniors this is because of the percentage of seniors in the whole data (16.21%). This is shown in figure 2. Additionally, the services that each customer enrolled for that was investigated. It shows most customers use telephone service (90%), and almost 79% of them have internet service. Also, almost 45 percent of people do not use OnlineSecurity, OnlineBackup, DeviceProtection, and TechSupport services. In MultipleLines, StreamingTV, and StreamingMovies services the number of people using or not is almost the same.

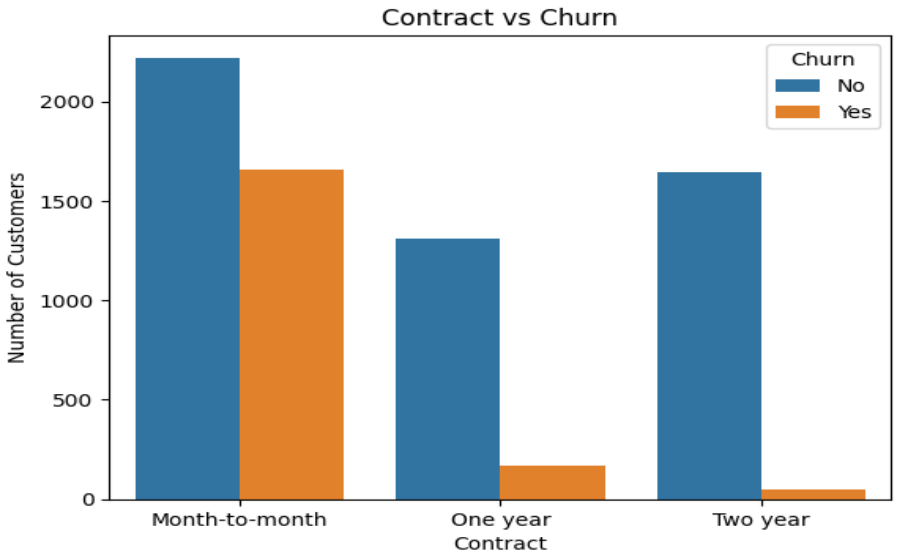

Figure 1.

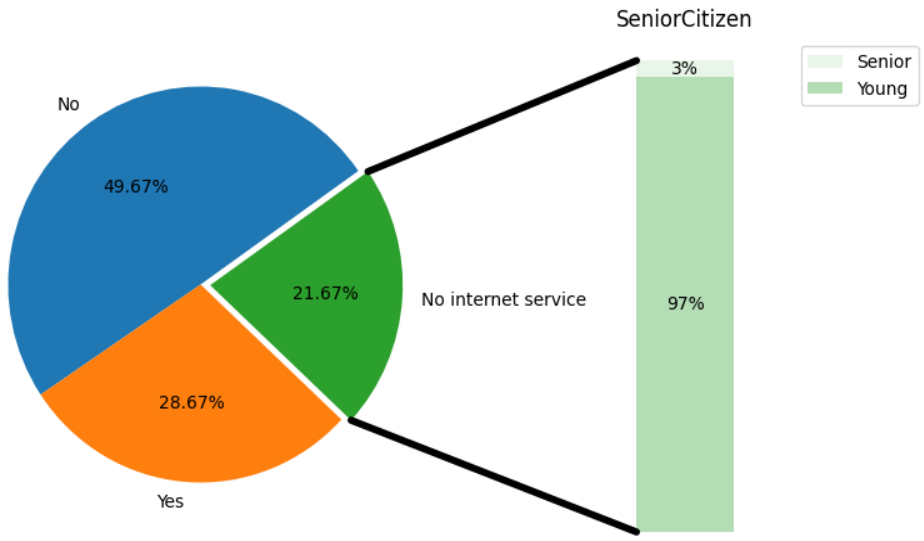

Figure 2.

Feature encoding and Engineering:

In this section, I describe the process of feature encoding and engineering. The goal of this process is to transform the categorical features into a numerical representation suitable for machine learning algorithms, as well as to perform dimensionality reduction using Principal Component Analysis (PCA). To encode the categorical features in the Telco Customer Churn dataset, I applied two methods: Label Encoding and One-Hot Encoding. Label Encoding is a technique that assigns a unique numerical label to each category in a categorical feature. I used the LabelEncoder() class from the sklearn.preprocessing module to perform this encoding. One-Hot Encoding is another popular method for encoding categorical features. It creates binary columns for each category, where a value of 1 represents the presence of the category and 0 represents its absence. I utilized the pd.get_dummies() function from the Pandas library to perform One-Hot Encoding. After applying Label Encoding to the dataset, the number of columns increased to 42.

Standardization:

I using StandardScaler() from the sklearn. preprocessing module to standardize the dataset. This preprocessing step ensures that all features have zero mean and unit variance, which helps in handling features with different scales and improves the performance of certain machine learning algorithms.

Dimensionality Reduction using PCA:

After encoding the categorical features, the resulting dataset contained a large number of columns, which may lead to increased computational complexity and potential overfitting. To address this issue, we employed Principal Component Analysis (PCA) for dimensionality reduction. By applying PCA to the encoded dataset, we reduced the dimensionality while retaining a significant portion of the data's variability. This reduction enhances computational efficiency and helps mitigate the risk of overfitting, especially when using machine learning algorithms.

Train Model:

In this step, the dataset was split into training and testing sets using the train_test_split module from the scikit-learn library. The aim of this split is to have a portion of the data reserved for model evaluation and testing, ensuring that the trained models' performance can be properly assessed. After splitting the data, different classifier algorithms were utilized to train the models. The choice of classifier algorithms will depend on the specific problem and the nature of the data. In this report, the following classifier algorithms were used:

Linear Classifier, Random Forest, Gradient Boost, XGBoost, KNN, Decision Tree, Bagging Classifier, Lpgestic Regration.

Evaluate Model:

After training the models on the training data, they were evaluated using the test data to evaluate their performance and determine the best model. The following metrics were used for evaluation:

Confusion Matrix, F-Measure, Accuracy. The confusion matrix provides a summary of the model's predictions and the actual labels. It shows the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The F-measure, also known as the F1 score, is a metric that combines precision and recall. It provides a single value that represents the model's performance in terms of both false positives and false negatives. The accuracy metric represents the overall correctness of the model's predictions. It calculates the ratio of correct predictions to the total number of predictions. Based on the evaluation metrics, the best model in terms of F-measure and accuracy is the SGDClassifier Classifier, which achieved an F-measure of -0.06 and an accuracy of 79.24 % on the test data (50 step training)

A visual representation with ROC Curve:

In addition to the metrics mentioned earlier, a Receiver Operating Characteristic (ROC) curve was used to visualize the performance of the models. The ROC curve illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) for different classification thresholds. The ROC curve plot shows the performance of the best model. The closer the curve is to the top-left corner, the better the model's performance. The area under the ROC curve (AUC) is a scalar value that represents the overall performance of the model. I created the ROC curve of the best version of each algorithm.

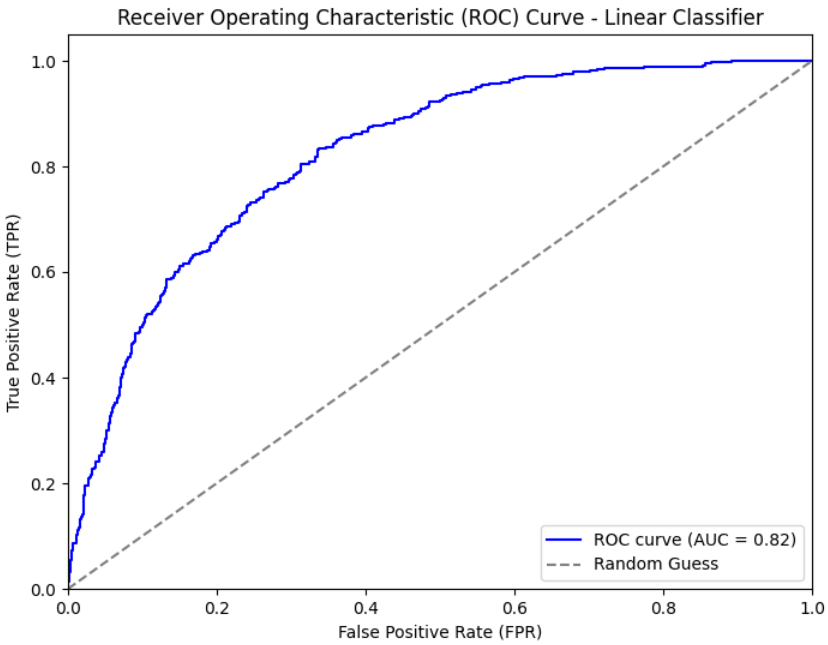

Figure 3. The ROC Curve of linear Classifier