Churn prediction project report

EDA(Exploratory data analysis):

Exploratory data analysis (EDA) is the process of examining and analysing data to comprehend its features and spot patterns and relationships. To acquire insights and create hypotheses, it makes use of statistical and visualisation approaches.

We started to check features type and the amount of Null values in the dataset:

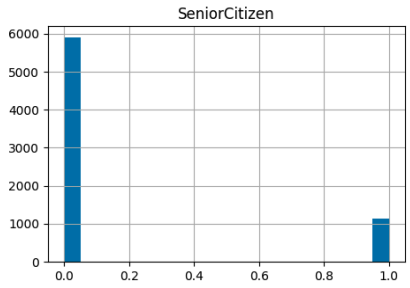
```
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
TotalCharges        object
Churn               object
dtype: object
```

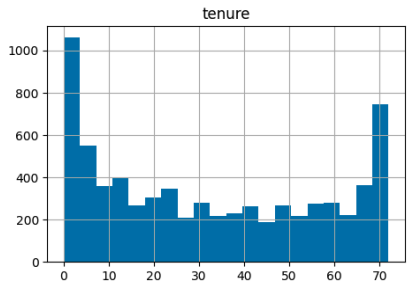
Fig1. Feature types

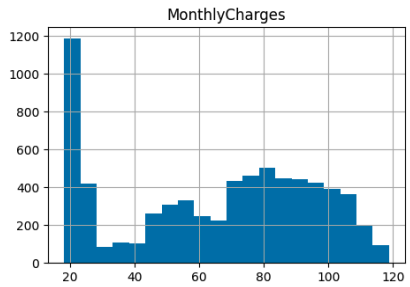
```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

Fig2. Numbre of null values

As there is no null value in the dataset, there is no need for handling the missing values.
The nex key step is to have an understanding of the value distribution per each features, to do so we started by plotting the histogram for the categorical features:



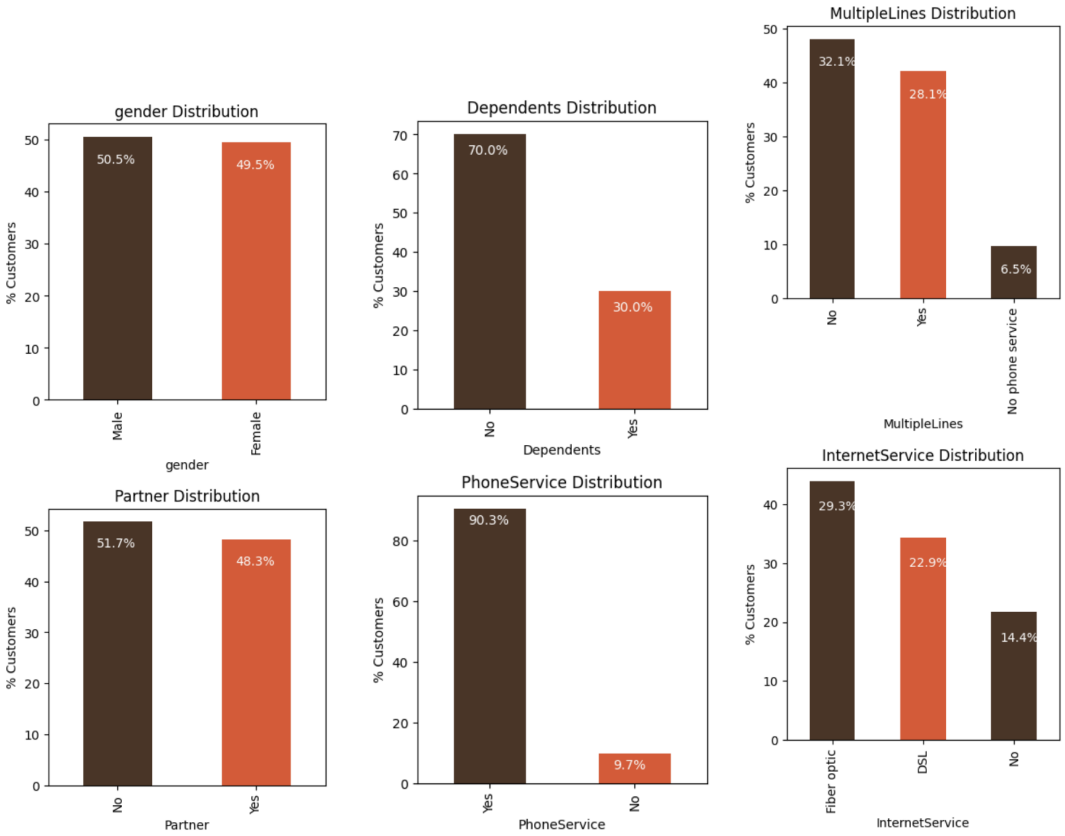




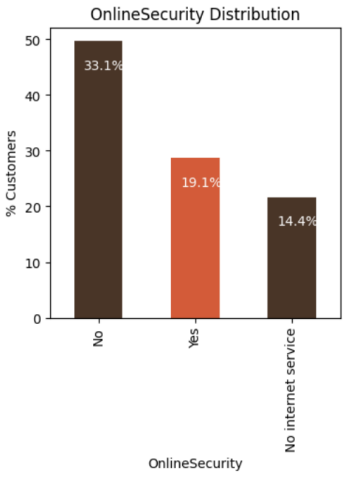
Churn prediction project report

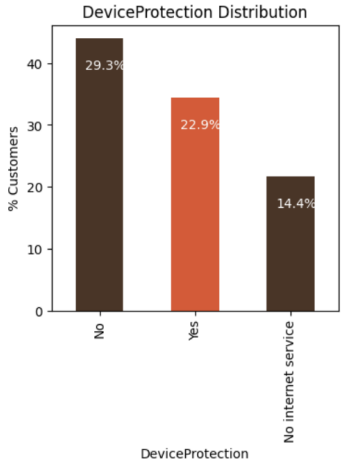
As the plots shows, most of the customers are senior citizens which can make this feature not much a distinctive one to categorize records based on that. However as the number of records is not huge, its existence in our feature list is not a big deal. The value distribution for tenure and Montly charges show considerable number per each group of values which can make them helpful features.

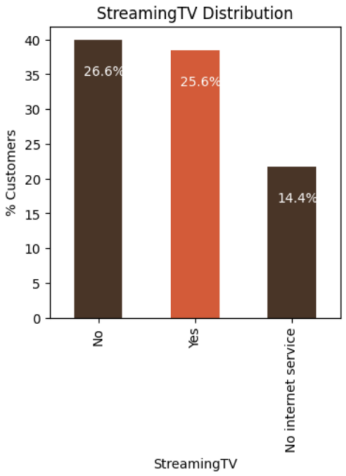
We also did the same analysis for numerical features and the result is as below:

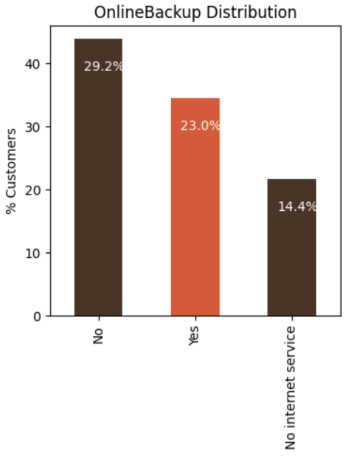


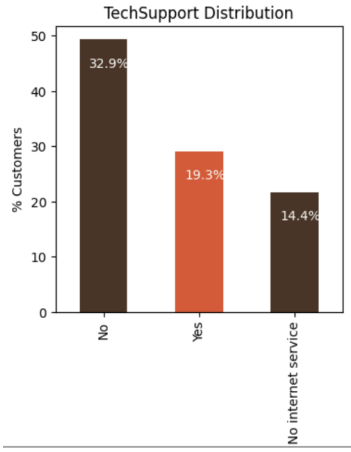
Churn prediction project report

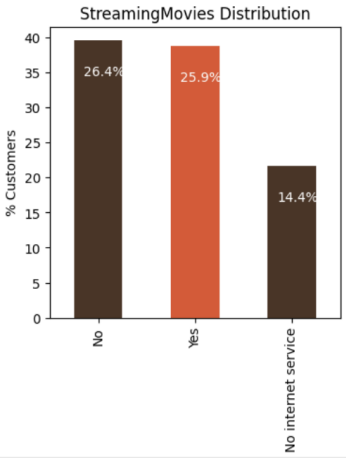


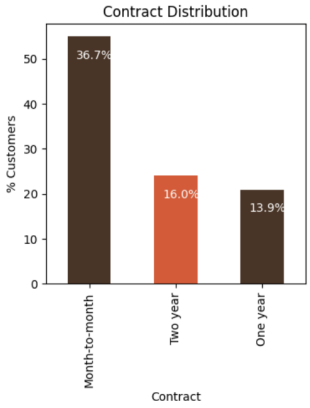


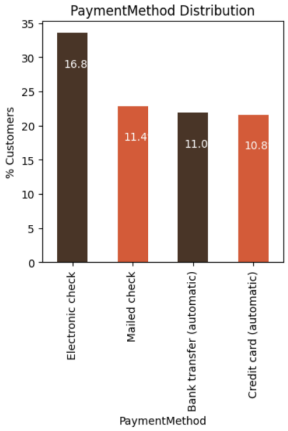


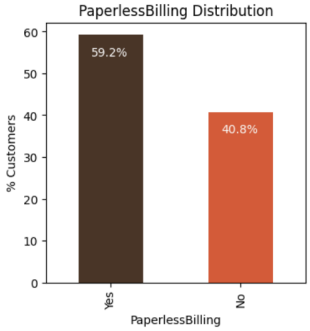


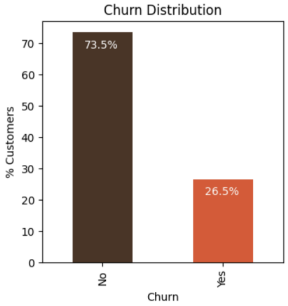












Churn prediction project report

Which indicates that all the numerical features can be helpful as the number of records per each value group in their distibutiona are considerable. You can find the related pie charts for all the features in the code notebook.

Pre-processing:

Preprocessing is crucial for creating a machine learning-based solution since it aids in getting the data ready for the algorithms to use. Missing data are handled, categorical data are encoded, the input characteristics are scaled, and outliers are eliminated using preprocessing procedures. Preprocessing approaches can change the data into a format that is better suited for machine learning algorithms, improving the model's accuracy and performance.

As the “TotalCharges” feature presents numerical values in string format, we cast all its values from string into number.
After such conversion, we observed null values as “TotalCharges”, we then remove all records with this situation from our dataset.

Since we want to train different ML models on our dataset and most of them does not accept non-numerical values, we then convert all the categorical features into numerical ones using Python “LabelEncoder” library from “sklearn” package.

Prediction:

Data normalisation is crucial for developing machine learning models because it enhances convergence, lessens the impact of outliers, allows for feature comparison, and makes models easier to understand. By ensuring that features are on a same scale, normalising data, we can more easily compare features and pinpoint the most crucial ones. This may ultimately enhance the machine learning model's performance and accuracy.

We create our pipeline with two steps, in which the first step is data normalization using “StandardScaler” from “sklearn” and the second step is classification. We create a list of such pipelines per 7 ML model of our choice. We also use 7-fold cross-validation for having a more robust accuray measurement and the models’ results based on that are as below:

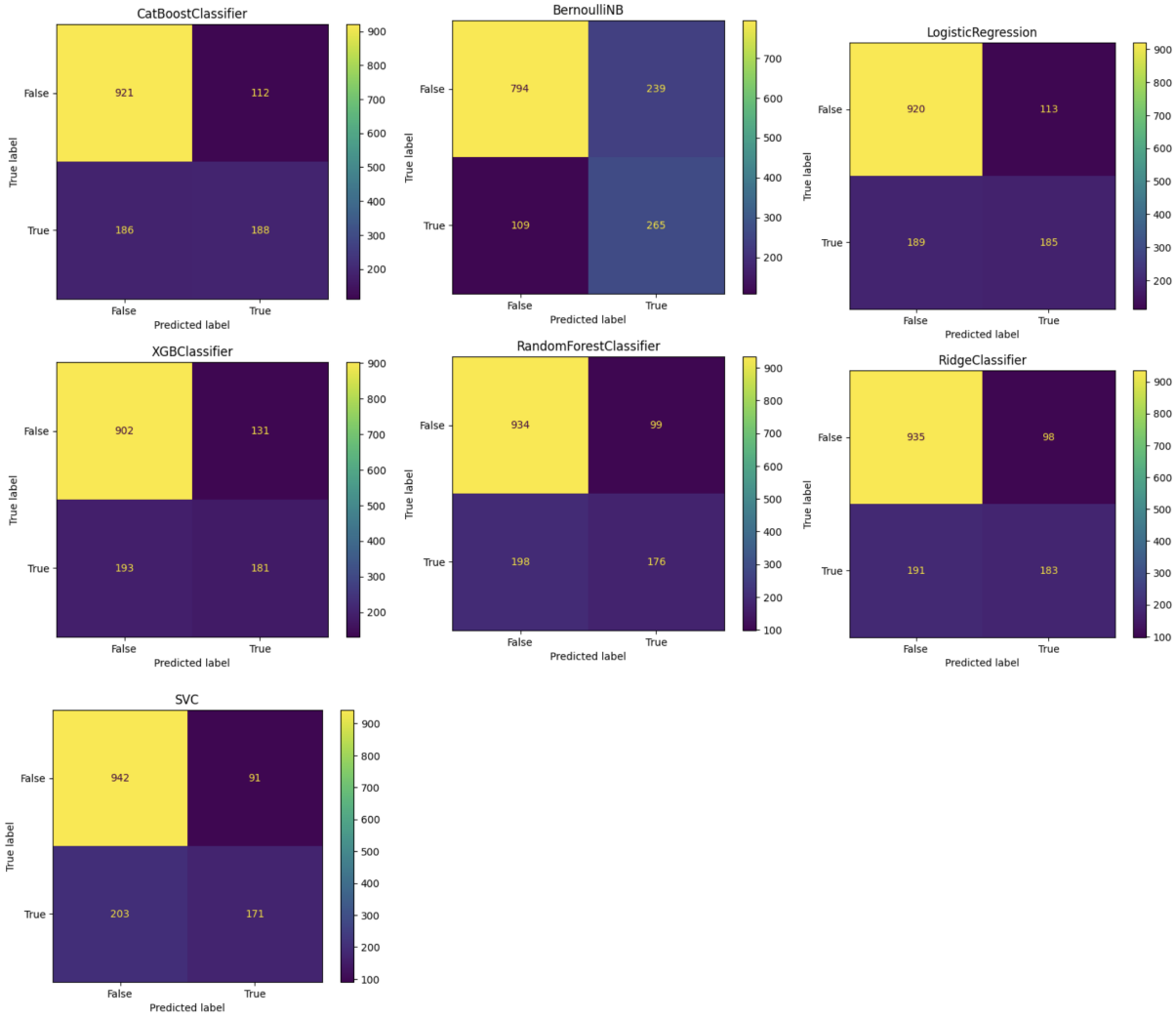
```
The mean cross-validation accuracy:

        CatBoostClassifier 0.797 +/- 0.009      in 33.369 secs
        XGBClassifier 0.786 +/- 0.006   in 7.71 secs
        LogisticRegression 0.804 +/- 0.009      in 0.246 secs
        RidgeClassifier 0.799 +/- 0.008         in 0.166 secs
        BernoulliNB 0.753 +/- 0.007     in 0.154 secs
        RandomForestClassifier 0.791 +/- 0.009  in 5.897 secs
        SVC 0.798 +/- 0.010     in 11.72 secs
```

Churn prediction project report

Based on the result, the RidgeClassifier seems to perform better in comparison with other models.

We then analyse all models' confusion matrix plot for further analysis which are as below:



Based on the cross-validataion accuracy and confusion matrix, we select the Catboost and RidgeClassifier for further hyperparameter tuning.

Hyperparameter tuning:

We first use GridSearch CV for finding the best values of models hyperparamters and perform the optimization on both "Ridge" and "Catboost" classifiers. Based on the result of such process, Catboost classifier performance improved more than Ridge after the tuning process.

Churn prediction project report

```
{'alpha': 0.01, 'class_weight': None, 'fit_intercept': True}
0.7994856038846612
```

Ridge accuracy after tuning.

```
{'depth': 6, 'iterations': 100, 'l2_leaf_reg': 5, 'learning_rate': 0.1}
0.8080189380267935
```

Catboost accuracy after tuning.

We try to achieve more improvement by using Baysian optimization as tuner for Catboost hypeparameters and get the following result:

iter	target	baggin...	depth	l2_lea...	learni...
1	0.7868	3.745	11.56	7.588	0.1836
2	0.7832	1.56	4.404	1.523	0.2612
3	0.7896	6.011	9.373	1.185	0.2913
4	0.7932	8.324	4.911	2.636	0.06319
5	0.7918	3.042	7.723	4.888	0.09446
6	0.7896	3.044	7.501	4.801	0.05983
7	0.7889	3.198	7.83	4.939	0.0284
8	0.7967	0.04478	3.363	5.198	0.1121
9	0.7939	9.708	9.386	9.945	0.1184
10	0.7932	4.715	4.444	8.218	0.03799
11	0.7882	9.383	9.357	5.051	0.2996
12	0.7861	7.752	10.94	3.045	0.118
13	0.796	0.02319	3.13	5.202	0.1712
14	0.7918	0.4394	3.42	4.995	0.03083
15	0.7854	7.426	7.726	3.703	0.01571
16	0.7882	0.09072	3.209	5.554	0.1535
17	0.7839	2.062	11.73	1.129	0.03014
18	0.7939	4.484	3.791	1.465	0.2527
19	0.7932	0.8307	10.19	9.298	0.2918
20	0.7932	3.645	5.581	7.23	0.2035

However, as we couldn't get further improvement by this approach, we can conclude that the because Catboost model does not have a considerable complex structure, there is no point in using any profound tuning approach like Baysian optimization.

Evaluation:

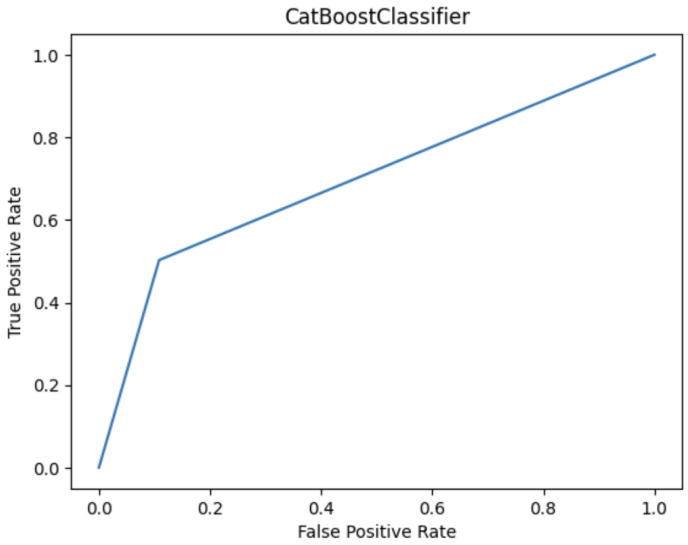
We use Classification report function from "sklearn" for evaluating our Catboost model and its result is as follow:

	precision	recall	f1-score	support
Churn	0.82	0.91	0.87	1033
Non-churn	0.65	0.46	0.54	374
accuracy			0.79	1407
macro avg	0.74	0.68	0.70	1407
weighted avg	0.78	0.79	0.78	1407

Churn prediction project report

We also plotted the roc_curve for it as follow:

The ROC curve can be used to analyse and demonstrate the performance of a binary classification model by charting the trade-off between true positive rate and false positive rate at various classification levels. The area under the curve (AUC), which can be used to summarise the model's overall performance, provides a way to visualise the model's performance. A greater AUC denotes better performance.



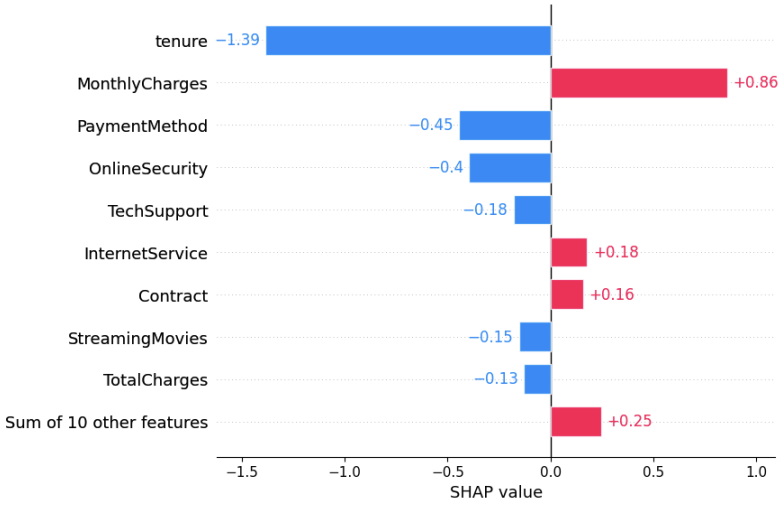
An ideal ROC curve will be closer to the upper left corner of the plot, indicating a high true positive rate (TPR) and a low false positive rate (FPR) in the model. Based on this criteria, our model seems to have an acceptable performance.

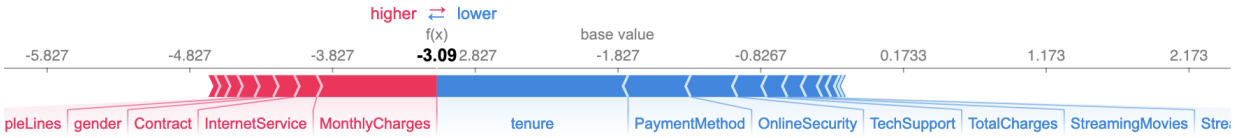
Feature importance analysis:

A SHAP (SHapley Additive exPlanations) value plot shows how each attribute affects a model's ability to predict an outcome for a given instance. It employs SHAP values, a model-independent technique for determining the relative relevance of each information in a prediction. The plot shows the contribution of each attribute as a horizontal bar chart, with positive values signifying an increase in prediction value and negative values signifying a drop in prediction value. A model's performance can be enhanced and errors can be identified by analysing a SHAP value plot to acquire insights into how a model generates predictions and which features are most crucial for the model's output.

With this aim, we plot three types of SHAP value plots for our final model as follow:

Churn prediction project report





As the two above plot shows, “tenure”, “PaymentMethod” and “MonthlyChrges” are the top 3 most influential features that can considerably effect the model’s final output. These plots can be helpful for choosing most effective features for further dimensionality reduction usings techniques such as LDA (Linear Discriminant Analysis) in the case of improving model’s time and computation complexity.