Classification Model for Customer Churn

Developed by: Shantanu Singh

Github: https://github.com/Shantanu990/DS_Project_Churn_Predict

Linkedin: https://www.linkedin.com/in/shantanu-singh-404a97141/

Table of Contents

1. Project Background & Objectives
2. Data Overview
3. Exploratory Data Analysis
4. Model Development
5. Results & Assessment
6. Next steps for improvements

Project Background

An IBM open-source customer churn dataset, sourced from Kaggle, was used for the classification model. With over 7,000 well-structured samples, the dataset was clean and had no missing values, requiring no substantial alteration or formatting. Each customer record within the dataset provided comprehensive details.

- **Customer ID, Tenure, Contract Type**
- **Geographic Details:** State, City, Lat/Long.
- **Demographic Details:** Gender, Senior citizen status, Partner and dependent status
- **Service Status:** Phone service, Internet service, Device Protection status, Streaming Movies etc.
- **Payment Method, Monthly Charges**
- **Churn Status, Churn Score, Churn Reason**

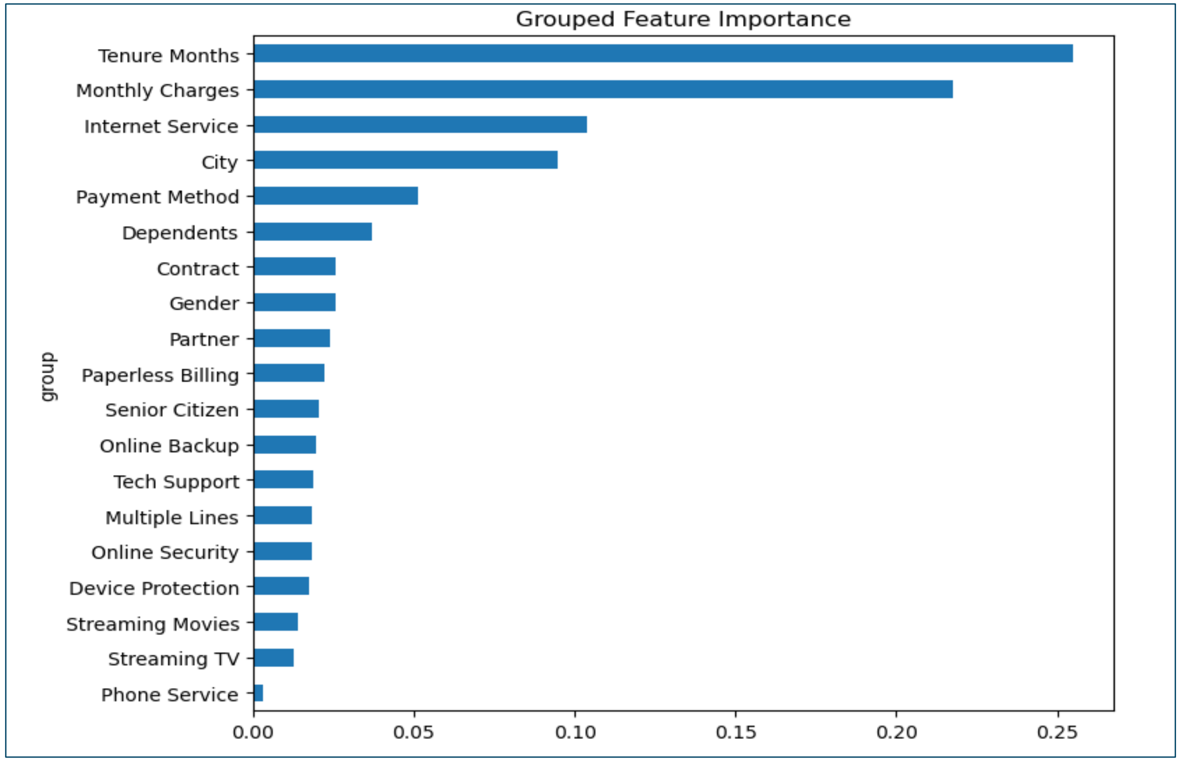
Problem Statement: Analysis of the customer churn dataset revealed that the churn score failed to provide a definitive estimation of customer churn for approximately 30% of samples where the score ranged between 65 and 80.

Project Objectives:

1. Develop a **Classification Model** to accurately predict customer churn
2. The model should also be able to calculate **churn probability** for each customer.

Exploratory Data Analysis (EDA)

- Random Forest Regressor was used in Python to determine most influential features in generating Churn Value (0 or 1).
- Following encoding/scaling techniques were used for respective features: frequency encoding- *City*, one hot encoding- *Gender, Senior Citizen, Partner, Dependents, Phone Service, Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Contract, Paperless Billing, Payment Method*, standard scaler- *Tenure, Monthly Charges*.
- Random Forest Regressor model was trained using features as input and Churn Value as the dependent variable.
- Subsequently, feature_importance_ attribute of scikit-learn library was used to extract relative feature importances (depicted in the graph below).
- Tenure was the most significant contributor to Monthly Charges, Internet Service etc.

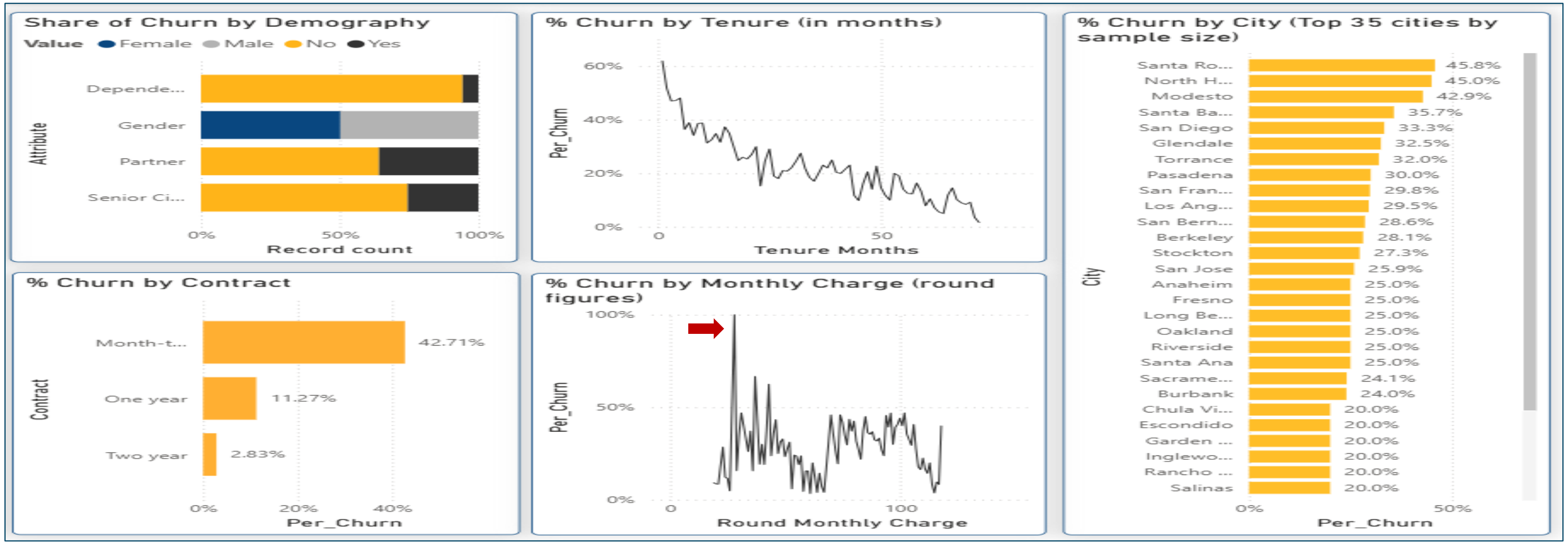


Exploratory Data Analysis (EDA)

7,043 samples of telecom service usage are analyzed, revealing that **1,869 customers (26.54%)** have churned. Percentage churn refers to the proportion of customers within a given attribute group who have discontinued their service. As shown in % Churn by City bar graph, certain cities with high sample size have relatively higher churn rates.

Key findings include:

- Customers on monthly contracts exhibit a disproportionately higher churn rate.
- Customers for whom the dependents or partner status is **'No'** tend to have a high churn ratio
- Churn % declines as the tenure increases.
- Churn % exhibits a non-linear dependency on monthly charges. Additionally, a notable outlier range exists where customers with low monthly charges show a disproportionately high churn percentage (highlighted in below image).
- Users who utilize phone service and non-senior citizens have notably higher churn rates.



Model Development

- A classification model was developed to determine churn probability of customers based on a set of known features. The dataset was prepared with the following feature engineering and encoding techniques:
  - One Hot Encoding- *Gender, Senior Citizen, Partner, Dependents, Phone Service, Multiple Lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Contract, Paperless Billing, Payment Method*, standard scaler- *Tenure, Monthly Charges*.
  - Frequency Encoding- *City*
  - Standard Scaler- *Tenure, Monthly Charges*
- The dataset was split into an 80:20 ratio for training and testing. Initially **Logistic Regression** was selected for model training and prediction.
- The model's performance and effectiveness was assessed through following measures:
  - **SHAP Summary and Force Plot:** The SHAP summary plot for the LR model revealed an inverse relationship: as monthly charges decreased, the model's predicted churn probability increased, and vice versa. This finding contradicts both the general understanding of customer churn and correlation analysis, which indicated a positive correlation of approximately 0.20 between 'monthly charges' and 'churn'. This suggests a potential learning anomaly in the model's interpretation of this feature, perhaps due to the **outlier samples** noticed during the EDA.
  - **Precision, Recall, F1-Score and Accuracy:** These performance metrics were used to assess overall accuracy and false flags which could cause the telecom company to overspend on customers retention or fail to proactively target customers who are actually going to churn. The results are provided in slide 9.
  - **ROC AUC:** To assess model's reliability in discriminating between a churner and non-churner.
- Since the Logistic Regression model misinterpreted the relationship between monthly charges and churn (as revealed by the SHAP summary plot), further training, testing, and evaluation of models using **XGBoost Classifier** and **TabNet Classifier** were carried out.
- The results of the models are provided in following slides.
- Tools and libraries used during EDA and model development:
  - Software: Python, Power BI, Excel
  - Libraries: xgboost, sklearn, pandas, numpy, matplotlib, seaborn, shap
  - Regression models: **XGB Classifier, TabNet, Logistic Regression**, Random Forest

Results & Assessment

SHAP summary/force plots for the **LR model** revealed an anomalous inverse relationship between monthly charges and churn probability (Figure 3). This learning pattern contradicted both the findings from the correlation analysis (Figure 4) and a direct examination of the dataset. Consequently, an XGBoost Classifier was selected for subsequent model training and evaluation.

Figure 3

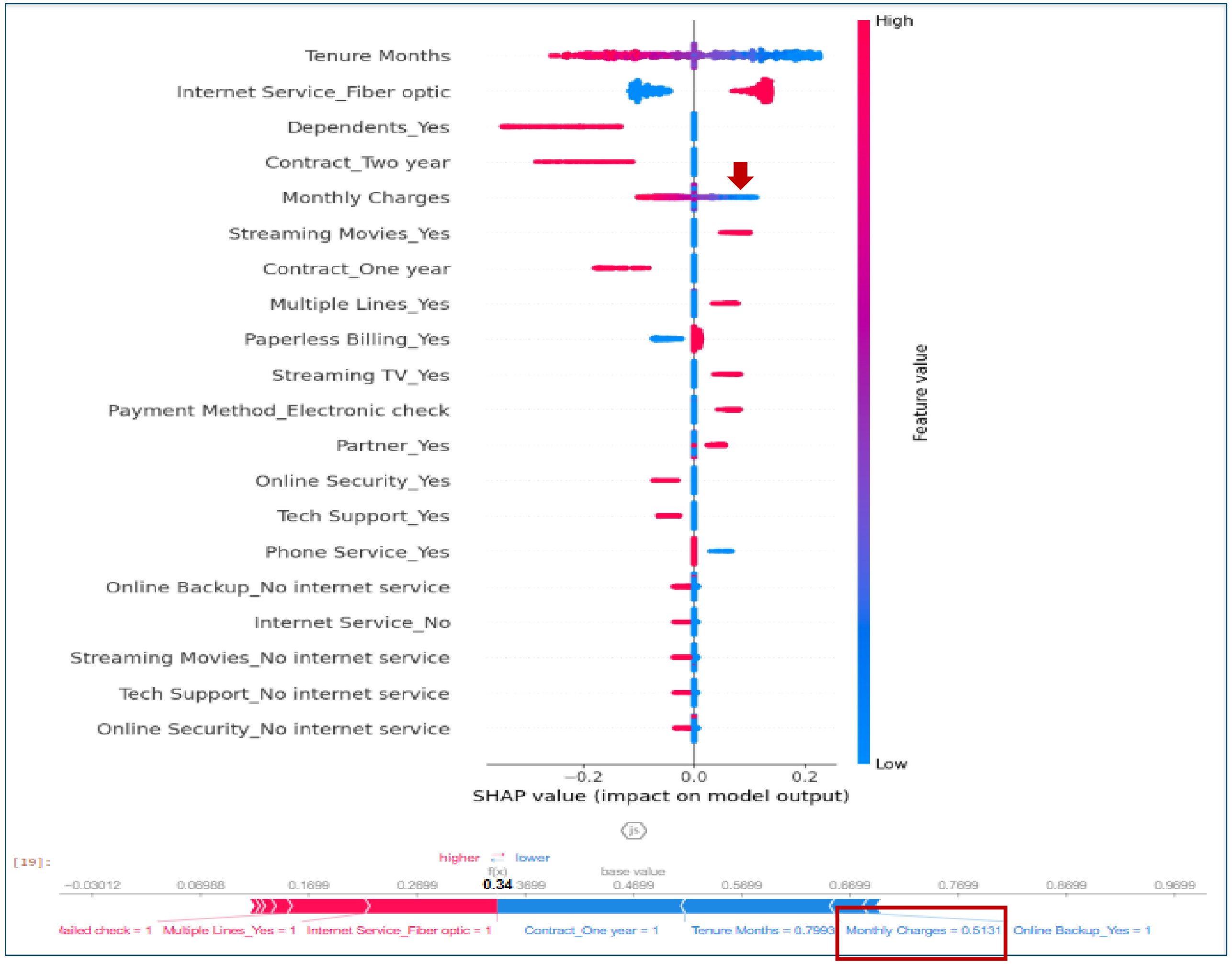
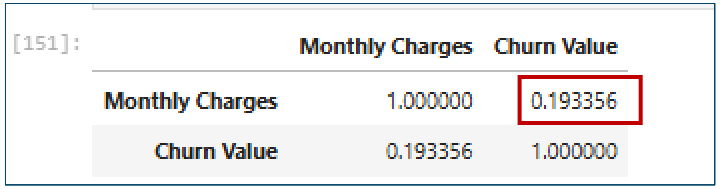
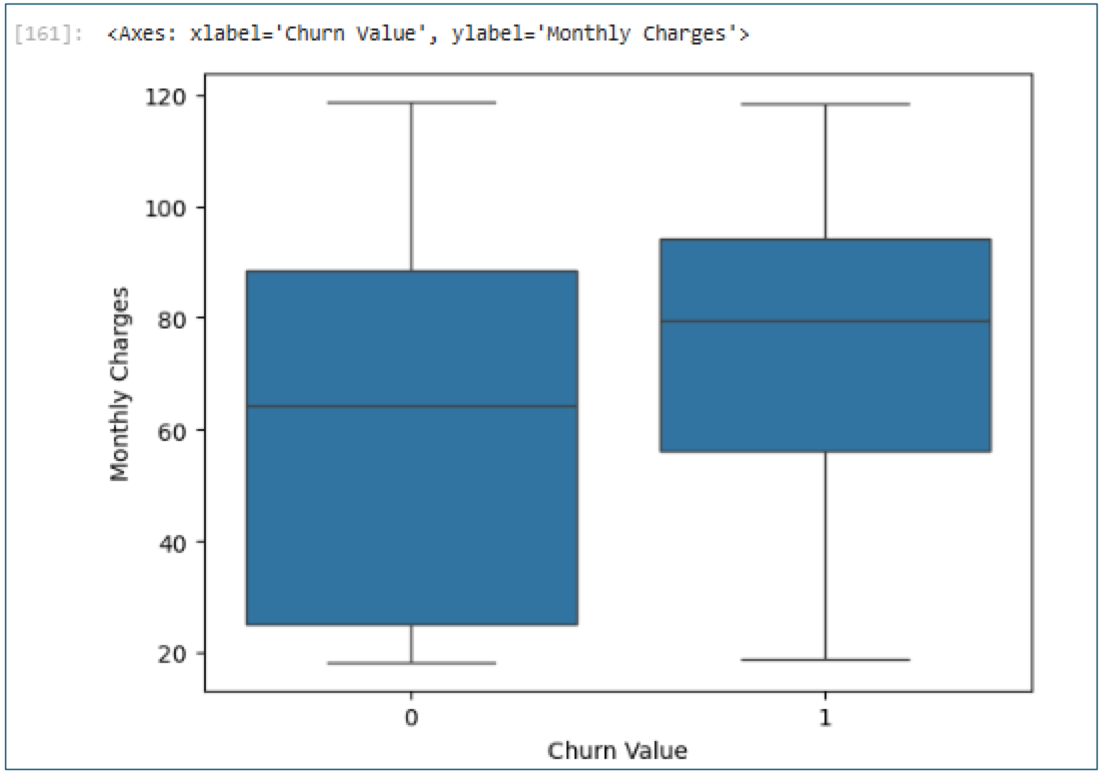


Figure 4



[151]:

	Monthly Charges	Churn Value
Monthly Charges	1.000000	0.193356
Churn Value	0.193356	1.000000

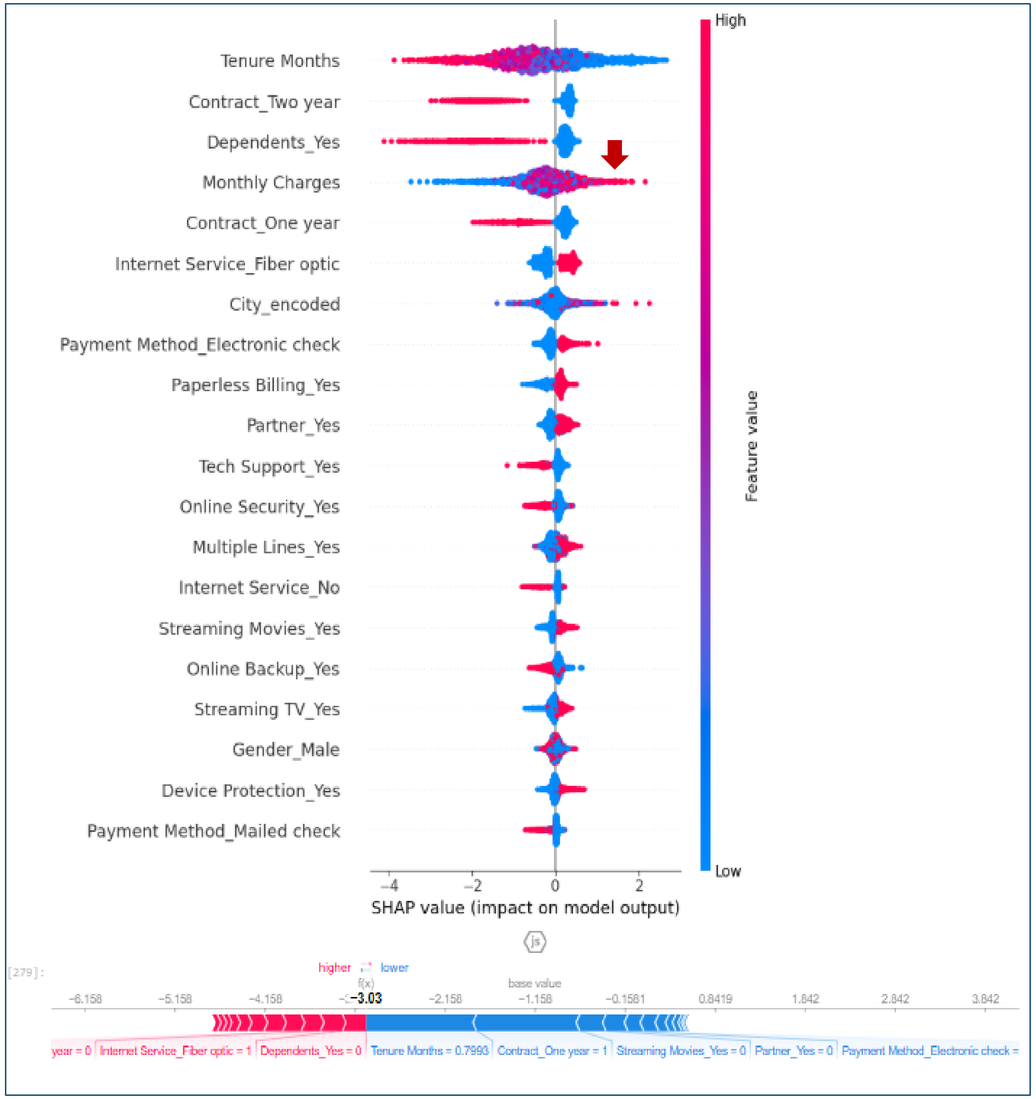


[161]: <Axes: xlabel='Churn Value', ylabel='Monthly Charges'>

Results & Assessment

SHAP summary/force plots for the **XGB Classifier** showed that the model successfully identified and represented the correct, relationship for Monthly Charges: higher charges = higher churn, lower charges = lower churn. This confirms that XGBoost has learned a more accurate and interpretable representation of this feature's impact on churn. Model's interpretation of top features is as follows:

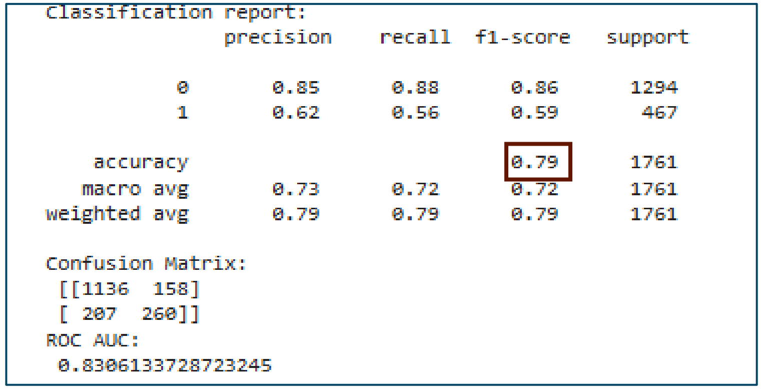
- **Tenure Months:** Low tenure increases churn, high tenure decreases churn. Consistent with actual data.
- **Contract:** Both, two and one-year contracts significantly decrease churn probability. Consistent with actual data.
- **Internet Service_Fibre optic:** Customer having fibre optic increases churn probability. Consistent with actual data due to competitive alternatives for internet services.
- **Dependent_Yes:** Customers who have dependents less likely to churn. Consistent with actual data.
- **Monthly Charges:** Customers with higher monthly charges are more likely to churn while those with lower charges are less likely to churn. Consistent with actual data.



Results & Assessment

- The initial run of the XGBoost Classifier gave an accuracy of 77%. Upon closer inspection of the predicted values, two distinct patterns of misclassification were identified:
  - **False Negatives for Device Protection:** The model frequently predicted a churn value of '0' for customers who actually churned ('1') and had 'Device Protection' enabled. This suggests the model did not adequately capture the relationship between device protection and churn.
  - **False Positives for Specific Demographics:** Conversely, the model often predicted a churn value of '1' for customers who did not churn ('0') and had 'Senior Citizen' status as 'No', 'Dependent' status as 'No', and 'Partner' status as 'No'. This indicates the model overemphasized these specific demographic combinations, leading to incorrect churn predictions.
- To address these identified misclassification patterns, sample weights were strategically adjusted. For customers with 'Device Protection' enabled and an actual churn value of '1' (false negatives), their sample weights were increased from 1.0 to 1.2. Conversely, for customers with 'Senior Citizen' status as 'No', 'Dependent' status as 'No', and 'Partner' status as 'No', but an actual churn value of '0' (false positives), their sample weights were increased to 1.15.
- This re-weighting led to a 2% improvement in overall accuracy in the subsequent model run, achieving **79%** (as shown in the figure 6 below). Consequently, the XGB Classifier is now capable of providing predictions for the 30% of cases previously lacking definitive estimation (highlighted in problem statement) from the original predictor, with an approximate 80% accuracy.
- However, despite this notable improvement in overall accuracy, the model's ability to precisely predict actual churn (Class 1) still presents a challenge, as reflected in its precision and recall values for this class. This is likely attributable to class imbalance, specifically the limited number of samples available for the churn class (only 1,869 samples). The probability scores for each sample were also determined as shown in figure 7.

Figure 6



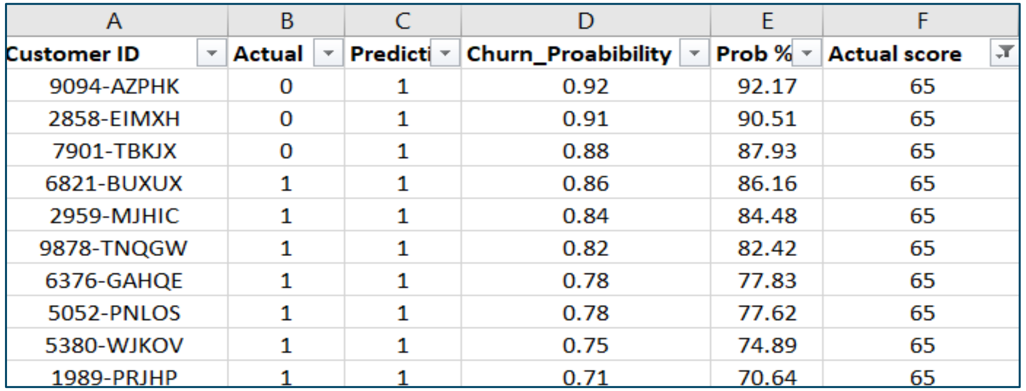
```
Classification report:
              precision    recall  f1-score   support

           0       0.85      0.88      0.86      1294
           1       0.62      0.56      0.59       467

    accuracy                           0.79      1761
   macro avg       0.73      0.72      0.72      1761
weighted avg       0.79      0.79      0.79      1761

Confusion Matrix:
 [[1136  158]
 [ 207  260]]
ROC AUC:
 0.8306133728723245
```

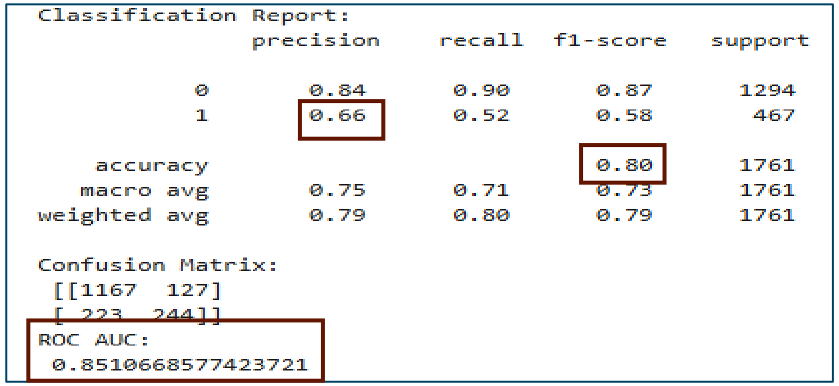
Figure 7



A	B	C	D	E	F
Customer ID	Actual	Predicti	Churn_Proability	Prob %	Actual score
9094-AZPHK	0	1	0.92	92.17	65
2858-EIMXH	0	1	0.91	90.51	65
7901-TBKJX	0	1	0.88	87.93	65
6821-BUXUX	1	1	0.86	86.16	65
2959-MJHIC	1	1	0.84	84.48	65
9878-TNQGW	1	1	0.82	82.42	65
6376-GAHQE	1	1	0.78	77.83	65
5052-PNLOS	1	1	0.78	77.62	65
5380-WJKOV	1	1	0.75	74.89	65
1989-PRJHP	1	1	0.71	70.64	65

Results & Assessment

- A third classifier model was trained using **TabNet**, which achieved an overall accuracy of **80%**, representing a small 1% improvement over the XGBoost Classifier (Figure 8).
- The TabNet model also recorded a slightly higher **ROC AUC of 0.85** compared to XGB, suggesting only a marginal improvement in its ability to discriminate between classes 0 and 1. This trend is also reflected in the precision for class 1, which improved from **0.62 with XGB to 0.66 with TabNet**, indicating that TabNet made somewhat more accurate class 1 predictions.
- Nevertheless, the performance gain remains limited, as with XGB, due to the relatively small number of class 1 samples.



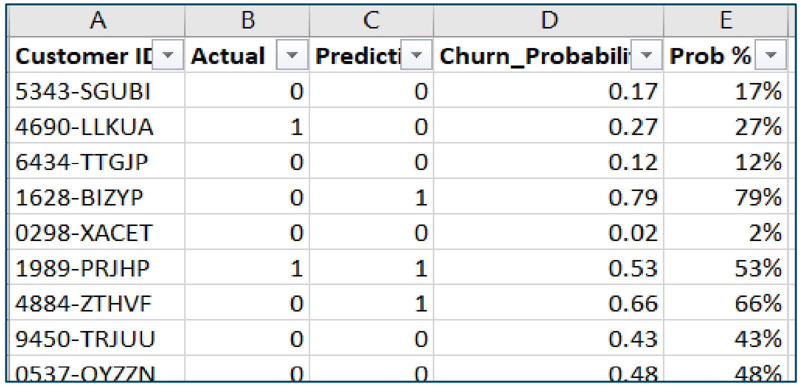
```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.90      0.87      1294
           1       0.66      0.52      0.58       467

    accuracy                           0.80      1761
   macro avg       0.75      0.71      0.73      1761
weighted avg       0.79      0.80      0.79      1761

Confusion Matrix:
 [[1167  127]
 [ 223  244]]
ROC AUC:
 0.8510668577423721
```

Figure 8



A	B	C	D	E
Customer ID	**Actual**	**Predicti**	**Churn_Probabili**	**Prob %**
5343-SGUBI	0	0	0.17	17%
4690-LLKUA	1	0	0.27	27%
6434-TTGJP	0	0	0.12	12%
1628-BIZYP	0	1	0.79	79%
0298-XACET	0	0	0.02	2%
1989-PRJHP	1	1	0.53	53%
4884-ZTHVF	0	1	0.66	66%
9450-TRJUU	0	0	0.43	43%
0537-OYZZN	0	0	0.48	48%

Figure 9

Next steps for improving the classification model

Following strategies can be used to further improve the classification model:

- **Data Augmentation:** Adding more samples where churn value is 1 will help the model learn feature relationships more accurately.
- **Create synthetic samples:** Create synthetic samples for minority class by using methods such as SMOTE or ADASYN.
- **Hyperparameter tuning:** Use Bayesian Optimization to find better configurations for hyperparameters and contribute to better minority class prediction.

Thank You