COLLEGE OF ENGINEERING, NORTHEASTERN UNIVERSITY

Subject: **DATA MINING IN ENGINEERING**
Subject Code: **IE7275**
Fall 2023

TELCO CUSTOMER CHURN

MILESTONE: PROJECT REPORT

GROUP 2
Santrupti Muttappa Bagali | bagali.s@northeastern.edu
Rakshit Vahi | vahi.r@northeastern.edu
Swatisama Guha Thakurta | guhathakurta.s@northeastern.edu

Name	Percentage of Efforts	Signature
Santrupti Muttappa Bagali	33%	
Rakshit Vahi	34%	Rakshit
Swatisama Guha Thakurta	33%	Swatisama Guha Thakurta

Submission Date: 12th December 2023

TABLE OF CONTENTS

1.0 PROBLEM SETTING 3

2.0 PROBLEM DEFINITION 3

3.0 DATA SOURCE 3

4.0 DATA DESCRIPTION 3

5.0 DATA EXPLORATION 4

5.1 Distribution of Gender in the Data and the Percentage that Churn *5*
5.2 Tenure of People Who Churn *6*
5.3 Distribution of categorical data *6*
5.4 Correlation of all predictors with churn *7*

6.0 DATA MINING TASKS 7

7.0 DATA MINING MODELS/METHODS 8

7.1 Random Forest *8*
7.2 KNN *11*
7.3 Support Vector Machine (SVM) *12*
7.4 Logistic regression *13*
7.5 Neural Network *14*

8.0 PERFORMANCE EVALUATION 15

8.1 Performance Metrics Used *15*
8.2 Confusion matrix *16*
8.3 Model Evaluation Summary *17*

9.0 PROJECT RESULTS 17

10.0 IMPACT OF THE PROJECT OUTCOMES 18

11.0 REFERENCE 19

1.0 PROBLEM SETTING

In the telecom industry, companies face the persistent challenge of customer churn, where subscribers discontinue their services with a telecom provider. High churn rates can significantly impact a company's revenue and profitability. Because of the competitive nature of the market, understanding and predicting customer churn is crucial for telecoms to retain subscribers and improve customer satisfaction. By delving into the nuances of customer behavior and preferences, telecom companies aim to not only mitigate churn but also foster long-term customer loyalty in an environment where adaptation and foresight are key to sustaining success.

2.0 PROBLEM DEFINITION

Within the realm of the telecommunications sector, the focal point of this project is the mitigation of customer churn. Our objective is to construct a predictive model that effectively identifies customers at a high risk of discontinuing their services. Through the lens of data analytics, we endeavor to address the following key questions:

1. Identification of Influential Factors: Uncover the pivotal factors influencing customer churn within the dynamic landscape of the telecom sector.
2. Predictive Model Development: Devise a robust predictive model capable of early detection, pinpointing customers prone to churn based on historical data and leveraging Supervised Machine Learning (ML) techniques.
3. Retention Strategy Exploration: Explore and propose targeted retention initiatives for telecom companies, harnessing the insights derived from our predictive model to minimize churn rates effectively.

In pursuit of these objectives, our project revolves around the utilization of historical customer data and the application of Supervised ML techniques. By leveraging these analytical tools, we aim to offer telecom companies a valuable resource for strategic decision-making, empowering them to address customer churn with a high degree of accuracy.

3.0 DATA SOURCE

The dataset is taken from Kaggle site. [1]

The dataset chosen is a sample data module tracks a fictional telco company's customer churn based on various factors by IBM Watson. [2]

We also referred a research paper [3] based on the same data set to understand better.

4.0 DATA DESCRIPTION

The dataset altogether has 7043 records and 21 predictor columns. Following are the predictor columns.

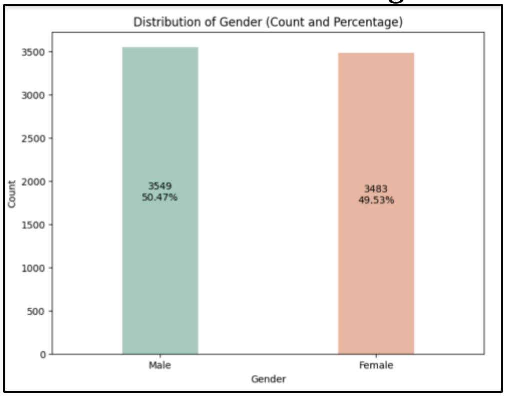
1. **CustomerID**: A unique ID that identifies each customer.
2. **Gender**: The customer’s gender: Male, Female

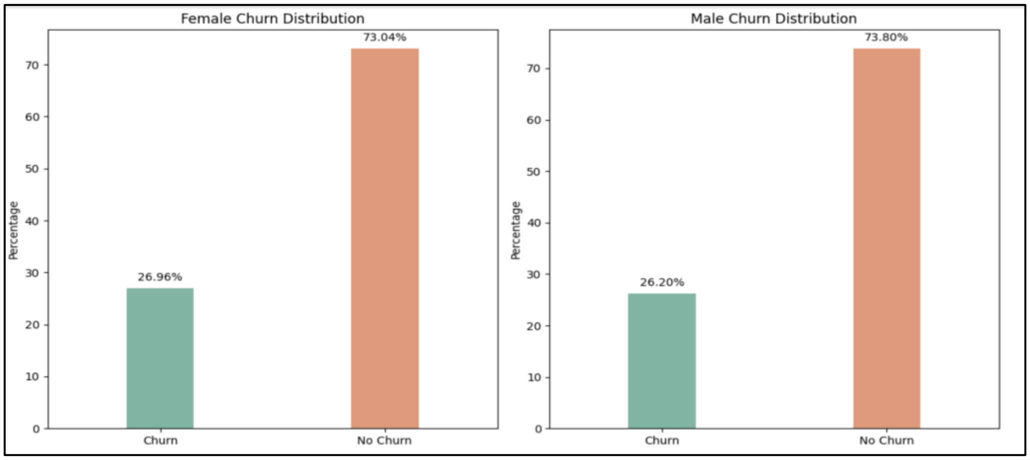
3. **Age**: The customer's current age, in years, at the time the fiscal quarter ended.
4. **Senior Citizen**: Indicates if the customer is 65 or older: Yes, No
5. **Dependents**: Indicates if the customer lives with any dependents: Yes, No. Dependents could be children, parents, grandparents, etc.
6. **Tenure in Months**: Indicates the total amount of months that the customer has been with the company by the end of the quarter specified above.
7. **Phone Service**: Indicates if the customer subscribes to home phone service with the company: Yes, No
8. **Multiple Lines:** Indicates if the customer subscribes to multiple telephone lines with the company: Yes, No
9. **Internet Service**: Indicates if the customer subscribes to Internet service with the company: No, DSL, Fiber Optic, Cable.
10. **Online Security**: Indicates if the customer subscribes to an additional online security service provided by the company: Yes, No
11. **Online Backup**: Indicates if the customer subscribes to an additional online backup service provided by the company: Yes, No
12. **Device Protection Plan**: Indicates if the customer subscribes to an additional device protection plan for their Internet equipment provided by the company: Yes, No
13. **Premium Tech Support**: Indicates if the customer subscribes to an additional technical support plan from the company with reduced wait times: Yes, No
14. **Streaming TV**: Indicates if the customer uses their Internet service to stream television programing from a third-party provider: Yes, No. The company does not charge an additional fee for this service.
15. **Streaming Movies**: Indicates if the customer uses their Internet service to stream movies from a third-party provider: Yes, No. The company does not charge an additional fee for this service.
16. **Contract**: Indicates the customer's current contract type: Month-to-Month, One Year, Two Year.
17. **Paperless Billing**: Indicates if the customer has chosen paperless billing: Yes, No
18. **Payment Method**: Indicates how the customer pays their bill: Bank Withdrawal, Credit Card, Mailed Check
19. **Monthly Charge**: Indicates the customer's current total monthly charge for all their services from the company.
20. **Total Charges**: Indicates the customer's total charges, calculated to the end of the quarter specified above.
21. **Churn Label**: Yes = the customer left the company this quarter. No = the customer remained with the company. Directly related to Churn Value.

5.0 DATA EXPLORATION

Exploratory Data Analysis (EDA) was carried out with the help of the Python programming language. Several other types of graphical representations, such as normal bar graphs and plots, were utilized within this study. By engaging in data exploration, we were able to considerably improve our understanding of the distribution of the data, which in turn made it easier to recognize patterns and trends. Following is the visualization output

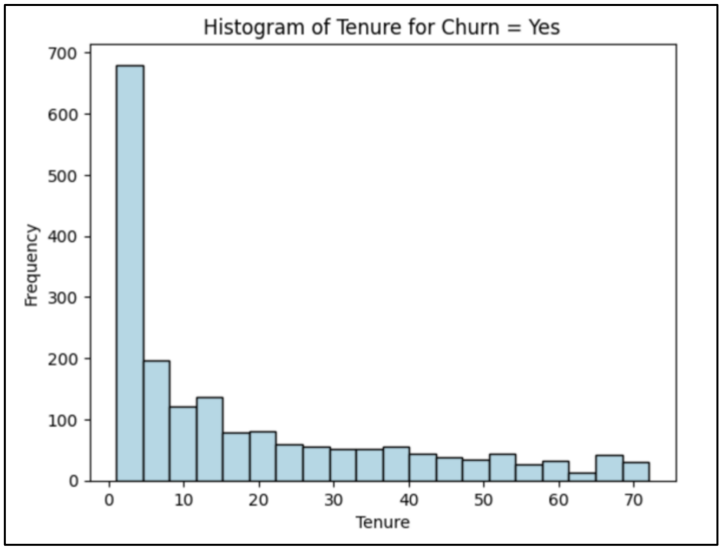
5.1 Distribution of Gender in the Data and the Percentage that Churn





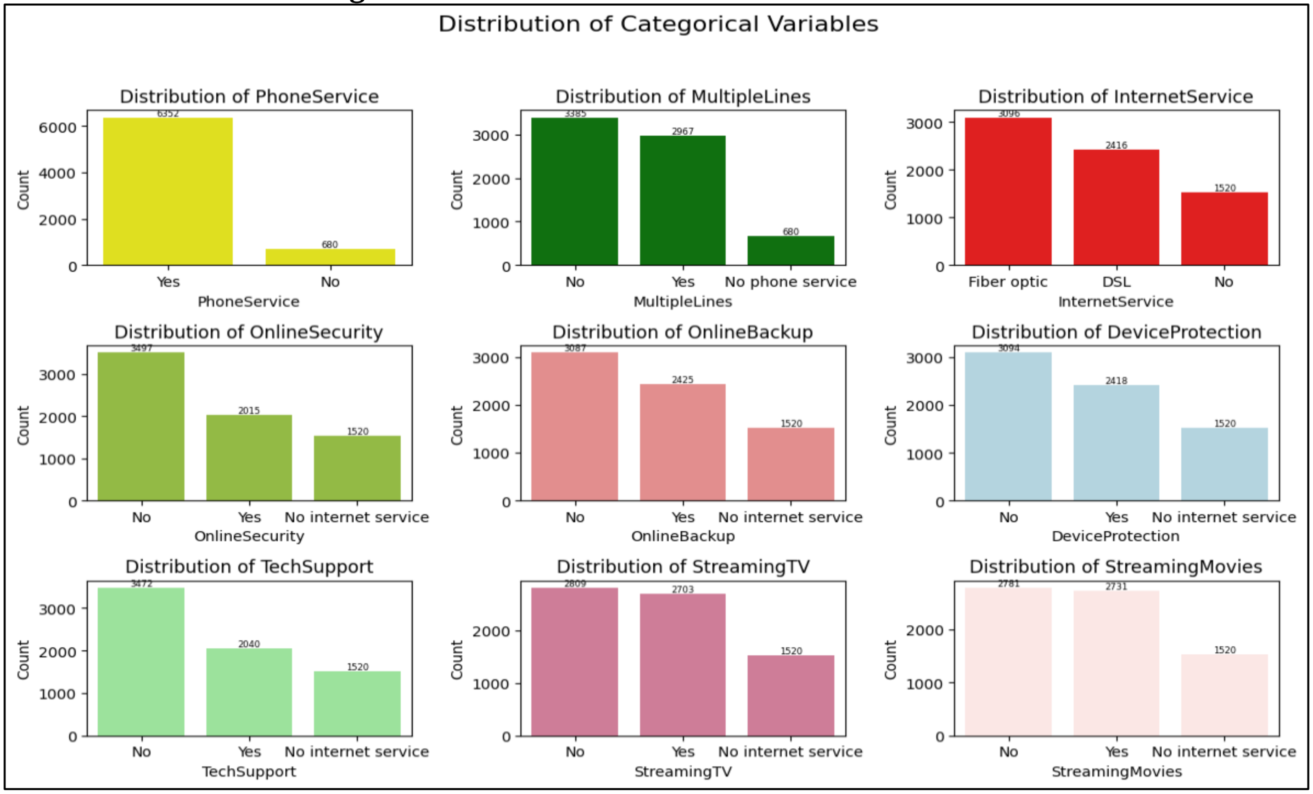
The churn rate is the same for both males and females, at 73%. Most of the customers who churn are females (50.47%).

5.2 Tenure of People Who Churn

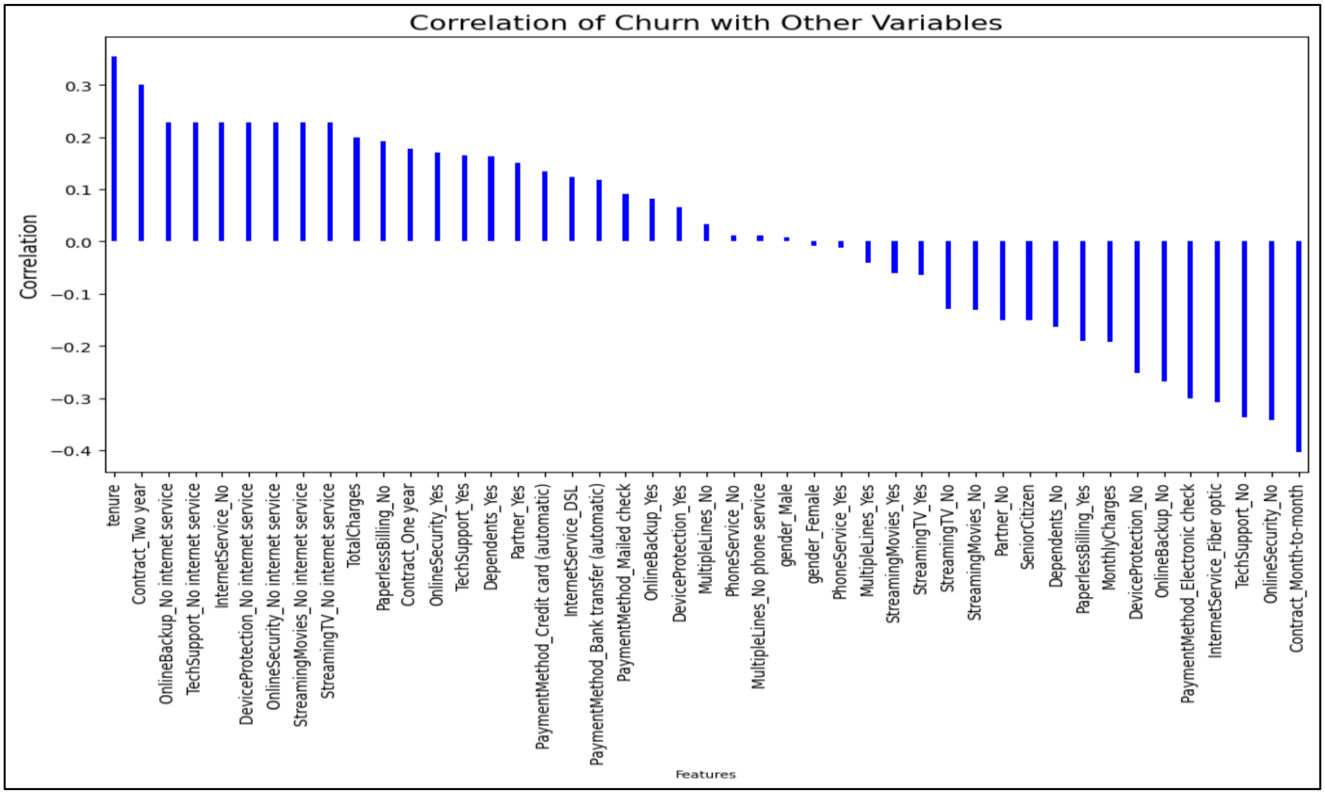


The majority of churned customers have a tenure of less than 20 months. The churn rate decreases after 20 months of tenure. This suggests that customers who have been with the company for longer than 20 months are more likely to stay. There is a significant peak in the number of churned customers at around 10 months of tenure.

5.3 Distribution of Categorical Data



5.4 Correlation of all Predictors with Churn



We can see that predictors like “tenure”, “Contract_Two year” are highly positively correlated with churn, where as “OnlineSecurity_No” and “Contract_Month-to-month” are highly negatively correlated with churn. Therefore, it can be inferred from the graph that features such as tenure, Contract, OnlineSecurity, TechSupport and much more play an important role in churn prediction.

6.0 DATA MINING TASKS

The steps taken in the process of data mining tasks are:

Step 1: Finding out the predictors that are numerical and categorical columns. We discovered that one of the columns, “TotalCharges”, which has all numerical value was a categorical column. The column was encoded to numerical column.

Step 2: Calculated number of missing values in the dataset. The column “TotalCharges” had 11 missing records. The records with the missing values were dropped.
Before dropping records: 7043 records and 21 predictor columns
After dropping records: 7032 records and 21 predictor columns

Step 3: Dropping the “Customer-ID” column and encoding the “churn” to binary values.
Churn [Yes: 1, No: 0]
Before Step 3: 7032 records and 21 predictor columns
After Step 3: 7032 records and 20 predictor columns

Step 4: Creating dummy variable for the categorical data
Before creation of dummies: 7032 records and 20 predictor columns

After creations of dummies: 7032 records and 46 predictor columns

Step 5: X set has all the columns and Y set has Churn predictor.
Dividing the data set into train, validate and test set in the percentage of 60, 20, 20.
Before dividing: 7032 records and 46 predictor columns
X: 7032 records and 45 predictor columns
Y: 7032 records and 1 predictor columns
After dividing:
X_train: 4210 records and 45 predictor columns
X_val: 1406 records and 45 predictor columns
X_test: 1407 records and 45 predictor columns
y_train: 4210 records and 1 predictor columns
y_val: 1406 records and 1 predictor columns
y_test: 1407 records and 1 predictor columns

Step 6: Training Models and hyper tuning the models on the validate set.

Step 7: Testing the performance metrics of each model on the test data set and comparing them.

7.0 DATA MINING MODELS/METHODS

In the realm of predictive modeling, choosing the right algorithm is crucial for the success of a churn prediction system. Our model development involved a thorough exploration of diverse supervised machine learning algorithms, each highlighting unique strengths. Here, we introduce key models, outlining their characteristics and the reasons for their inclusion in our evaluation. Additionally, we incorporated a Neural Network as our baseline model, leveraging its capacity to capture complex patterns for an enhanced understanding of customer churn dynamics.

In our model training and hyperparameter tuning process, a particular emphasis was placed on optimizing the recall value of the positive class (where churn=yes). By prioritizing recall, we aimed to fine-tune the model to effectively minimize false negatives, ensuring that customers at risk of churning were identified with high accuracy. This strategic focus contributes to the model's ability to proactively detect potential churners and enhances the overall effectiveness of customer retention strategies.

7.1 Random Forest

Random forest is one of the most widely used machine learning algorithm, which combines the output of multiple decision trees to reach a result. A random forest can handle data containing continuous variables, in case of regression, and categorical variables, in case of classification. Random Forest was adopted due to its ability to handle non-linearity, resilience to outliers, and ease of use and fine tuning. The Following approach was used to build a random forest.

- **Pre-processing**: Even though a random forest can handle categorical data, it was required to use an encoding technique as the python library sklearn works only with numerical data. One-hot encoding and label encoding techniques were tested, and it was observed that there was no significant difference in the outputs. Therefore, it was decided to use one-hot encoding because the rest of the model were use the same.

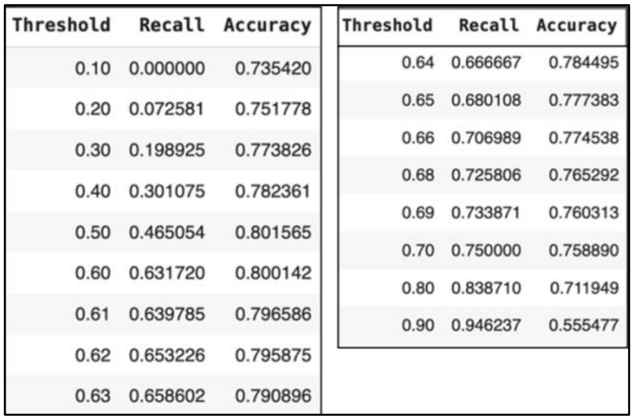
- **Baseline Model**: a random forest baseline model was created with no parameters. Random State parameter was set to 42 to make sure that the results were same at every run. The baseline model had an accuracy of 99.85% on the train set with a recall of 100%, and an accuracy of 78.73% with a recall of 48% on the validate set, which clearly indicated an overfitting of the model on the train set.
- **Hyperparameters Tuning**: Random Forest has the following hyperparameters that can be tuned:

n_estimator	Number of trees in the forest. Increasing the number of trees in the forest generally improves performance but also increases the computational cost.
criterion	This is used to specify the technique used to measure the quality of the split. For classification problems either gini or entropy is used.
max_depth	Maximum depth of every decision tree in the forest. Increasing max_depth may lead to overfitting. Therefore, it's important to tune this parameter well.
min_samples_split	Minimum number of samples needed to split a node. This controls how finely the tree is allowed to partition the input space.
min_samples_leaf	Minimum number of samples needed to be there at the lead node. It helps in controlling the size of terminal nodes in each tree.
bootstrap	Whether to use bootstrapped samples when building trees. If True, each tree is built on a random subset of the data with replacement.

To tune the hyperparameters RandomizedSearchCV method was adopted. In this method the algorithm randomly selects a subset of hyperparameters from the defined parameter distribution and evaluates the model on the train set to find the optimal values of the hyperparameters. After performing a random search the following optimal values of hyperparameters were achieved:

1. n_estimator: 198
2. criterion: ‘gini’
3. max_depth: 15
4. min_samples_split: 9
5. min_samples_leaf: 11
6. bootstrap: True

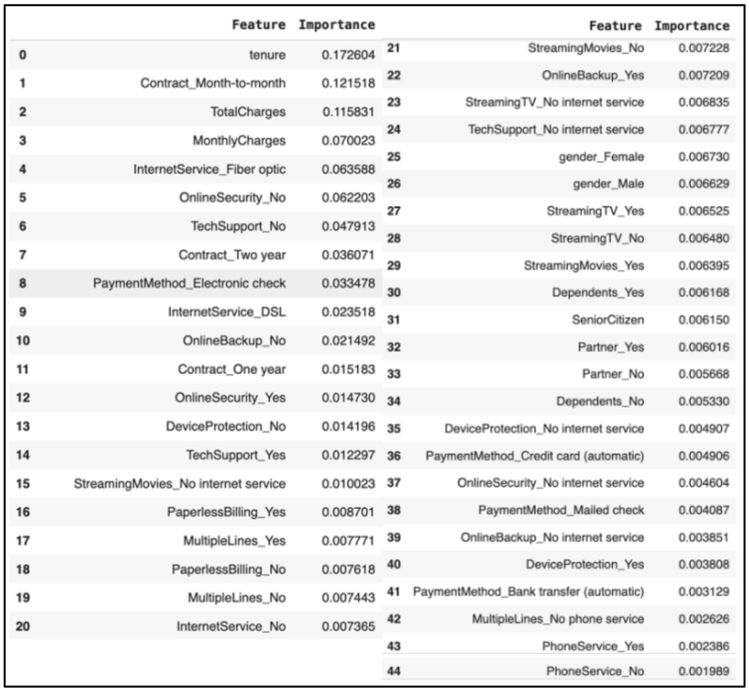
- **Model Training and Validation**: After creating and fine-tuning the model to achieve a good overall accuracy, it was observed that the recall value remained suboptimal. Recognizing the significance of prioritizing recall, the approach of threshold tuning was adopted. This involved exploring various threshold values, each validated on the validation set to identify the optimal threshold that strikes the best tradeoff between recall (positive class) and overall accuracy. The objective was to fine-tune the model to excel not only in accuracy but also in capturing a higher proportion of relevant instances, thereby enhancing its recall performance. Below is the table with a range of threshold values along with their respective accuracies and recalls:



Threshold	Recall	Accuracy
0.10	0.000000	0.735420
0.20	0.072581	0.751778
0.30	0.198925	0.773826
0.40	0.301075	0.782361
0.50	0.465054	0.801565
0.60	0.631720	0.800142
0.61	0.639785	0.796586
0.62	0.653226	0.795875
0.63	0.658602	0.790896
0.64	0.666667	0.784495
0.65	0.680108	0.777383
0.66	0.706989	0.774538
0.68	0.725806	0.765292
0.69	0.733871	0.760313
0.70	0.750000	0.758890
0.80	0.838710	0.711949
0.90	0.946237	0.555477

It can be observed that at a threshold of 0.60 we achieve a Recall of 63.17% along with an accuracy of 80.01%. By slightly increasing the threshold to 0.62 we can achieve a recall of **65.32%** along with an accuracy of **79.58%.** A decision was made to go ahead with the random forest with a threshold of 0.62 as with a small decrease in accuracy there was a significant improvement in the recall of the positive class.

Random forest gave the opportunity to also find the importance of each feature in predicting the churn (target). Below is the table with the features listed in descending order of their importance.



	Feature	Importance
0	tenure	0.172604
1	Contract_Month-to-month	0.121518
2	TotalCharges	0.115831
3	MonthlyCharges	0.070023
4	InternetService_Fiber optic	0.063588
5	OnlineSecurity_No	0.062203
6	TechSupport_No	0.047913
7	Contract_Two year	0.036071
8	PaymentMethod_Electronic check	0.033478
9	InternetService_DSL	0.023518
10	OnlineBackup_No	0.021492
11	Contract_One year	0.015183
12	OnlineSecurity_Yes	0.014730
13	DeviceProtection_No	0.014196
14	TechSupport_Yes	0.012297
15	StreamingMovies_No internet service	0.010023
16	PaperlessBilling_Yes	0.008701
17	MultipleLines_Yes	0.007771
18	PaperlessBilling_No	0.007618
19	MultipleLines_No	0.007443
20	InternetService_No	0.007365
21	StreamingMovies_No	0.007228
22	OnlineBackup_Yes	0.007209
23	StreamingTV_No internet service	0.006835
24	TechSupport_No internet service	0.006777
25	gender_Female	0.006730
26	gender_Male	0.006629
27	StreamingTV_Yes	0.006525
28	StreamingTV_No	0.006480
29	StreamingMovies_Yes	0.006395
30	Dependents_Yes	0.006168
31	SeniorCitizen	0.006150
32	Partner_Yes	0.006016
33	Partner_No	0.005668
34	Dependents_No	0.005330
35	DeviceProtection_No internet service	0.004907
36	PaymentMethod_Credit card (automatic)	0.004906
37	OnlineSecurity_No internet service	0.004604
38	PaymentMethod_Mailed check	0.004087
39	OnlineBackup_No internet service	0.003851
40	DeviceProtection_Yes	0.003808
41	PaymentMethod_Bank transfer (automatic)	0.003129
42	MultipleLines_No phone service	0.002626
43	PhoneService_Yes	0.002386
44	PhoneService_No	0.001989

7.2 KNN

K-Nearest Neighbors (KNN) is a simple and versatile machine learning algorithm used for classification and regression. Its key characteristics include simplicity, non-parametric nature, adaptability to local patterns, and robustness to outliers. KNN doesn't require a training phase and is effective when decision boundaries are non-linear or when data exhibits local patterns. However, it can be computationally expensive and requires careful tuning of parameters like the number of neighbors (k) and the distance metric for optimal performance.

- **Distance used:** Gower distance is a metric used in KNN for datasets with mixed numerical and categorical features. It calculates the dissimilarity between instances by considering attribute types and uses a weighted average of attribute-wise distances. In KNN, Gower distance helps find the nearest neighbors, enabling the algorithm to make predictions for classification or regression based on the majority class or average values of these neighbors. Gower distance is valuable for handling diverse datasets and requires careful consideration of attribute weights and distance functions for optimal performance.
- **Baseline model:** The baseline model of the KNN was created with no parameters and has an accuracy of 81% and a recall of 70% on train set. Whereas on validate set, it has an accuracy of 76% and a recall of 62%.
- **Hypertuning parameters:**

n_neighbors	The n_neighbors parameter in KNeighborsClassifier controls model complexity, with smaller values leading to sensitivity and larger values to simplicity. It plays a key role in the bias-

	variance tradeoff and requires careful tuning for optimal performance.
metric	The metric parameter specifies the distance metric used to measure the similarity between data points, influencing the K-Nearest Neighbors (KNN) algorithm's behavior.

Next step is to find the optimal K value. The K value for which the recall and the accuracy is considerable. We found the optimal K value by checking the accuracy and recall for K ranging from 1 to square root of the length of the X_train set on the train set. The distance matrix is precomputed using the gower_matrix function.
The optimal value of n_neighbors is found to be 8 with accuracy to be **76%** and recall to be **63%** on the validate set.

7.3 Support Vector Machine (SVM)

SVM was chosen as one of our modelling algorithms because of its proficiency in handling high-dimensional data and capturing complex patterns within the dataset. SVM's ability to delineate decision boundaries in the feature space aligns seamlessly with the binary nature of our target variable, making it a robust candidate for churn prediction. We implemented SVM for churn prediction by adopting the following approach.

- **Pre-processing:** In our SVM-based churn prediction, we applied Min-Max Scaling to standardize feature scaling within the range of 0 to 1, effectively addressing SVM's sensitivity by considering the minimum and maximum values of the features. This ensures uniformity, preventing larger-magnitude features from dominating and enhancing SVM's classification effectiveness. We independently applied Min-Max Scaling to each set—training, validation, and testing. Exclusively using fit-transform on the training data ensured that scaling parameters originated solely from the training set. This approach safeguards model integrity, avoiding any influence on test and validation data and preventing potential data leakage.
- **Hyperparameter Tuning:** In addition to feature scaling, we performed hyperparameter tuning to optimize SVM's performance. Through a diligent random search on training data, we have identified the best hyperparameters as below:
- **Kernel Trick:** In some cases, data might not be linearly separable in its original feature space. That’s why, we have used the Radial Basis Function (RBF) kernel in SVM to capture intricate, non-linear data patterns, by transforming the input features into a higher-dimensional space, allowing SVM to construct decision boundaries that adapt to complex relationships within the data.
- **C parameter:** The parameter C in SVM serves as a regularization term controlling balance between fitting the training data well and generalization to new, unseen data. C was set as 40 to achieve a smooth decision boundary and correctly classifying training points. It also helped prevent overfitting or underfitting.

- **Gamma value**: Gamma value in SVM controls the width of the kernel and influences the shape of the decision boundary. Gamma is set as ‘auto’ to find the suitable value based on the input variable.
- **Class weightage:** Our dataset has slight class imbalance. SVM is susceptible to slight imbalances in the dataset, potentially leading to biased predictions. So, we have used class weightage in hyperparameters during model training to manage a balanced representation of both classes. We have identified the ideal class weightage value through random search. Specifically, we assigned a weight of 1.71 to instances of class 0, indicating its higher importance in the learning process. Conversely, instances of class 1 were assigned a weight of 1, maintaining the standard weight.
- **Random Search:** The random search algorithm randomly samples hyperparameter values from the specified distribution for each iteration. Random search was only applied on train data. Due to the randomness involved, we have observed different results in each run. So, we have picked the best hyperparameters combination that improved our accuracy and recall both.
- **Model Train & Validation:** We have trained our model with all the hyperparameters that we have identified. We have also monitored our validation accuracy and recall value to tune our hyper parameters. After the model was trained and validated, it was rigorously tested on an independent test data to assess its generalization performance and ensure its effectiveness in making predictions on unseen data. The SVM model achieved a validation accuracy of **79.02%,** and the recall value for the positive class is **66%.**

7.4 Logistic regression

Logistic regression was selected for churn prediction due to its interpretability, efficiency in binary classification, simplicity, and the model's ability to provide probabilistic outputs, aligning well with the requirements of the task.

- **Pre-processing**: Here also, we have independently applied Min-Max Scaling to each dataset-Train, Validate and Test.
- **Hyperparameter Tuning:** We have also applied random search on training data to get the best hyperparameters combination for our model. Then we meticulously tuned key hyperparameters to enhance its predictive performance. The chosen hyperparameter configuration is as follows:
- **C (Regularization Strength):** The parameter C was fine-tuned to 26.23 to control the regularization strength, balancing model complexity and generalization.
- **Penalty:** We opted for 'L2' regularization, adding a penalty term to the loss function to prevent overfitting.
- **Max Iterations:** The maximum number of iterations for optimization was set to 300 to ensure convergence during the training process.
- **Solver:** 'lbfgs' was selected as the optimization algorithm, known for its suitability in the context of medium-sized datasets.

- **Class Weightage:** We know that the performance of the logistic regression model can be impacted by even a little imbalance in the class distribution. So, class weights were provided to ensure equitable representation during model training. We have assigned a weight of 1.61 to instances of class 0, indicating its higher importance during training. Conversely, instances of class 1 were assigned a weight of 1, maintaining the standard weight.
- **Random Search:** The random search algorithm randomly samples hyperparameter values from the specified distribution for each iteration. Random search was only applied on train data. Due to the randomness involved, we have observed different results in each run. So, we have picked the best hyperparameters combination that improved our accuracy and recall both.
- **Model Train & Validation:** We have trained our model with all the hyperparameters that we have identified. We have also monitored our validation accuracy and recall value to tune our hyper parameters. After the model was trained and validated, it was rigorously tested on an independent test data to assess its generalization performance and ensure its effectiveness in making predictions on unseen data. The Logistic Regression model achieved a validation accuracy of **79.08%,** and the recall value for the positive class is **65%**.

7.5 Neural Network

After applying various machine learning models, we have chosen neural network as our baseline model. Neural network is capable of handling non-linear patterns, allowing for a more comprehensive exploration of the complex relationships present in the dataset. Neural network, with its ability to adapt to different levels of feature complexity, can better accommodate the intricate dynamics of customer churn.

- **Key components of Neural network:** We have defined a sequential neural net with one input layer, two hidden layers and one output layer. We used Rectified Linear Unit (ReLU) activation function for hidden layers. The output layer has one neuron with a sigmoid activation function suitable for binary classification.
- **Hyperparameters tuning:** We have tuned numerous hyperparameters during the training process as below:
- **Learning rate:** A custom learning rate is defined(custom_leraning_rate=0.001) and the Adam optimizer is utilized with specified learning rate. We have adjusted the learning rate to find a balance that facilitates effective optimization and model training. the learning rate of 0.001 was chosen for our neural network in churn prediction to achieve a balance between convergence speed and stability, avoiding issues such as overfitting.
- **Compilation of the Model:** The model is compiled using the binary crossentropy loss function, which is appropriate for binary classification tasks. The model aims to minimize this loss during the optimization process.
- **Epochs and batch size:** In our neural net training process, we have utilized 25 epochs, where each epoch represented a complete pass through the entire training dataset. Additionally, we employed a batch size of 32 indicating the number of training samples processed in each

iteration. We have chosen number of epochs based on training loss convergence and learning rate.

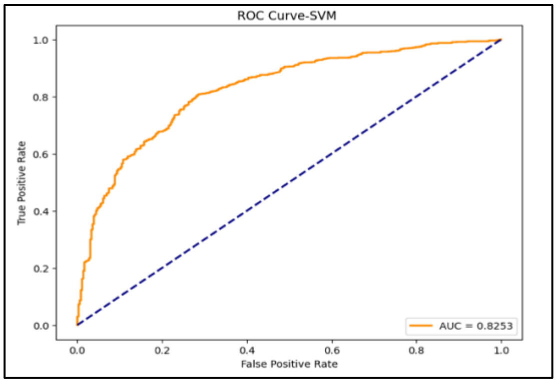
- **Model Train and Validation:** We have utilized TensorFlow library for building and training our neural network and integrated Keras API into TensorFlow for high-level model construction and training. We have also monitored our validation accuracy to tune our hyperparameters. The Neural Network achieved a validation accuracy of **78.09%,** and the recall value for the positive class is **66%**.

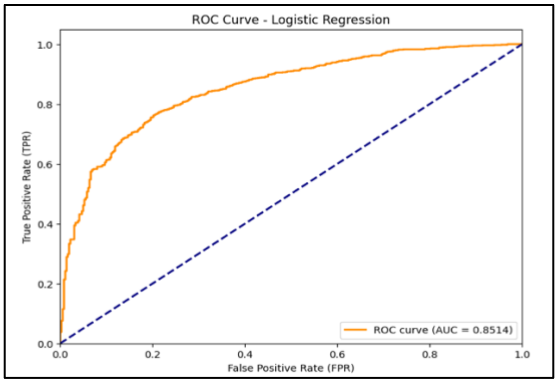
8.0 PERFORMANCE EVALUATION

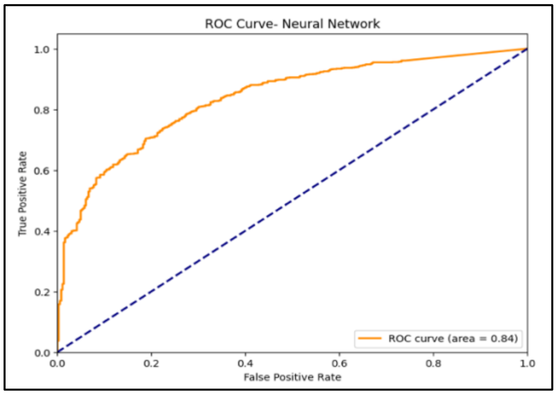
Performance evaluation is an important step in assessing the effectiveness and reliability of the data mining algorithms. It is the analysis of several metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve, among others. These metrics help in understanding how well the data mining model can generalize to unseen data and capture the underlying patterns within the dataset. Additionally, the use of confusion matrices helps in understanding a model’s ability to correctly classify instances. Performance evaluation methods help in fine tuning models which ultimately helps in selecting the best model.

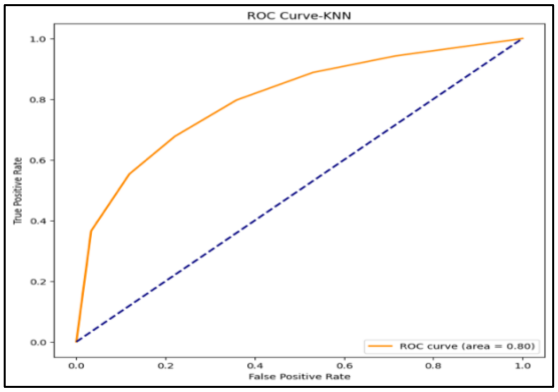
8.1 Performance Metrics Used

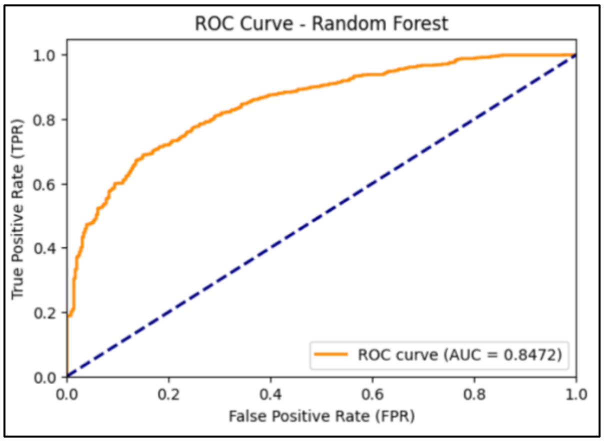
In evaluating our data mining models, we placed particular emphasis on accuracy to measure overall correctness and on recall for its significance in capturing positive instances effectively and especially minimizing false negative instances. This approach is crucial in telecom churn prediction, ensuring that our model effectively identifies customers at risk of churning while minimizing the risk of overlooking potential churn cases. Then we have focused on creating and analyzing Receiver Operating Characteristic (ROC) curves for each model, using the test data. The ROC curves provided a visual way to see how well our models balanced true positive and false positive rates at different thresholds. Additionally, we have closely monitored area under the ROC curve (AUROC) value for each model. AUROC provided a summarized metric, showing model’s overall ability to distinguish between positive and negative instances.







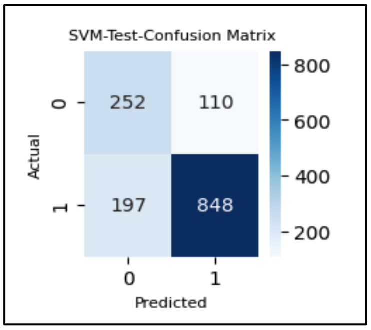


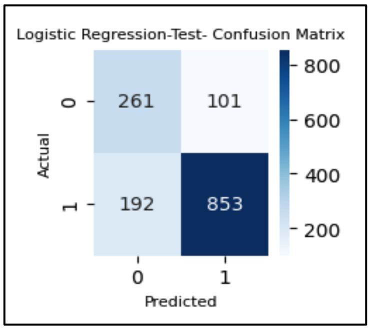


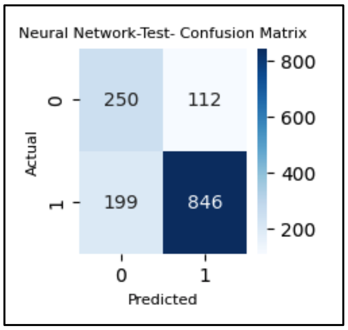
8.2 Confusion matrix

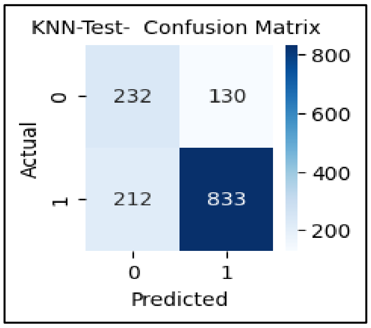
Here we've depicted the confusion matrix for the test data of each model, illustrating whether instances were correctly classified or not.

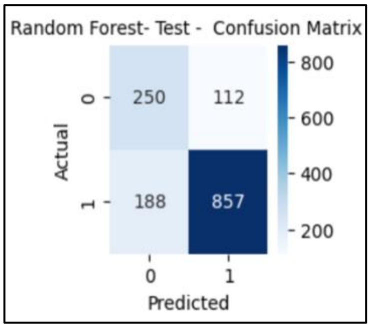
- **True Positives (TP):** The upper-left corner reflects true positives, representing customers correctly identified as likely to churn.
- **False Negative (FN):** The top-right corner signifies false negatives, indicating instances where the model missed predicting customers who were likely to churn.
- **False Positive (FP):** The bottom-left corner represents false positives, denoting cases where the model incorrectly predicted customer churn when they actually stayed.
- **True Negative (TN):** The bottom-right corner corresponds to true negatives, signifying instances where the model accurately identified customers who did not churn.











8.3 Model Evaluation Summary

Below table shows Accuracy, Recall and AUROC score of all the models.

Models	Test Accuracy	Recall (Positive Class)	AUROC Score
Random Forest	78.67%	69%	0.8472
KNN	75.69%	64%	0.80
SVM	78.18%	70%	0.8253
Logistic Regression	79.17%	72%	0.8514
Neural Network	77.9%	69%	0.84

Table 1: Accuracy, Recall and AUROC score of all the models

9.0 PROJECT RESULTS

After an in-depth analysis of every model's performance using test data, the findings show that logistic regression performed better than the other models in terms of accuracy and recall. Logistic regression showed highest test accuracy at **79.17%,** showcasing a remarkable recall rate of **72%** for positive instances and a strong AUROC score of **0.8514**. Notably, logistic regression excelled in identifying true positives while effectively minimizing false negatives, underscoring its proficiency in capturing positive churn instances accurately. It not only enhances overall accuracy but also mitigates the risk of overlooking churn cases, ensuring a comprehensive approach to identifying and addressing potential customer attrition. Logistic regression also effectively captures important features in churn prediction when the data exhibits a simpler structure with linearly contributing features and minimal nonlinearity in feature interactions.

Random Forest and SVM also demonstrated commendable performance, exhibiting robust accuracy and recall value. The utilization of a Neural Network as a baseline served as a valuable reference point, contributing to a comprehensive understanding of the data and model performance. All the models demonstrated robust AUROC scores above 0.8, indicating their ability to distinguish between positive and negative instances. This suggests a high level of predictive performance and reliability across the models, with implications for their effectiveness in real-world applications where accurate classification is crucial. Also, our analysis revealed that features such as tenure, total charges, and the type of contracts played significant roles. These factors emerged as key contributors, influencing the prediction of customer churn within the telecom service.

10.0 IMPACT OF THE PROJECT OUTCOMES

Our project's influence is set to transform the telecommunications industry by offering a predictive algorithm that not only achieves high accuracy but also guarantees strong recall, thus reducing the chance of missing good cases. The telecom business greatly values this predictive algorithm due to its significant impact on addressing the essential issue of customer churn. According to Forbes [4], the cost of acquiring a new customer in the telecom business can be four to five times more than the cost of retaining an existing customer. Our initiative directly confronts this difficulty, providing a strategic solution that may promptly and significantly influence corporate outcomes.

The predictive methods, as demonstrated by the effectiveness of logistic regression, random forest, SVM, KNN and neural network in our experiment, enable telecom businesses to accurately estimate customer churn. By implementing this approach, it enables these firms to proactively and specifically address issues, so preventing client loss and ensuring customer loyalty. The model's exceptional accuracy enables a dependable forecast of possible churn, while the focus on recall guarantees precise identification of positive events, indicating consumers who are at risk of leaving.

The ramifications for the telecommunications industry are significant. By accurately forecasting customer churn, organizations may promptly adopt targeted interventions, customized retention strategies, and loyalty programs that cater to the specific requirements of consumers who are at risk of leaving. This not only ensures the protection of current sources of income but also substantially decreases the expenses linked to gaining new clientele.

The immediate impact on the company is twofold. To begin with, the predictive algorithm offers a financial benefit by reducing significant expenses that would otherwise be allocated to gaining new clients. Additionally, it promotes consumer allegiance and the ability to maintain customers, thereby enhancing long-term profitability and market competitiveness. The project's results establish it as a revolutionary influence in the telecommunications sector, providing a practical and efficient resolution to one of its most urgent challenges.

11.0 REFERENCE

[1] BlastChar. (2018, February 23). *Telco customer churn*. Kaggle. https://www.kaggle.com/datasets/blastchar/telco-customer-churn

[2] *Telco customer churn*. IBM. (n.d.). https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=samples-telco-customer-churn

[3] Sharmila K. Wagh, Aishwarya A. Andhale, Kishor S. Wagh, Jayshree R. Pansare, Sarita P. Ambadekar, S.H. Gawande, Customer churn prediction in telecom sector using machine learning techniques, Results in Control and Optimization, Volume 14, 2024, 100342, ISSN 2666-7207, https://doi.org/10.1016/j.rico.2023.100342.

[4] Kumar, S. (2023, September 12). *Council post: Customer retention versus customer acquisition*. Forbes. https://www.forbes.com/sites/forbesbusinesscouncil/2022/12/12/customer-retention-versus-customer-acquisition/