Telecom Customer Churn Analysis

Submitted by Group 12:

Aahan Kotian*

Cody Starke

Jaytiben Bhatt

Muhammad Z Khan

August 14th, 2023

Table of Contents

Introduction:

The Telecom Industry is heavily invested in not just growing its consumer base but also growing their loyal consumer base. The Telecom industry is highly competitive and ranges from large and well-established corporations to just newly created, local corporations. Consumers have the easy choice to jump from one telco corporation to another depending on services being provided or even due to promotions that are going on. In order to have a healthy financial outlook it is imperative that telco corporations look to increase their customer loyalty.

Increasing competition in the industry has pushed companies to prevent customers or subscribers to switch to another company occupying the same market space (Geiler et al., 2022). This is also known as customer churn and it can be particularly damaging for subscription-based services such as online gambling, music streaming, and telecommunications (Geiler et al., 2022).

Objective:

Our project focuses on analyzing the churn rate of the Telco Customer Churn dataset. Our main objective was to select an appropriate model that accurately predicts customer churn. To achieve this, we leveraged several machine learning classification algorithms to predict customer churn. The models selected were logistic regression model, random forest, decision trees, adaptive boosting, gradient boosting, KNN classification and support vector machines. We predict that logistic regression will perform the best as they are known to be the workhorse model for predicting probabilities and classification of data (Nield, 2022, p.226).

Data Preparation:

Data Procurement:

Data was collected from only one source i.e. https://github.com/treselle-systems/customer_churn_analysis/blob/master/WA_Fn-UseC_-Telco-Customer-Churn.csv

Data Quality:

Information provided in Dataset:

Demographic:

- gender: Male, Female.
- SeniorCitzen: 0 (No), 1 (Yes).
- Partner: Yes, No.
- Dependents: Yes, No.

Account related:

- CustomerID: ID
- Tenure: Number of service months with the Telco.
- Churn: (Customers who left within the last month) - Yes, No.
- Contract: Month-to-month, One year, Two year.
- PaperlessBilling: (Yes, No).
- PaymentMethod: Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic).

- *MonthlyCharges, TotalCharges: Amount*

Services signed up by the customer:

- *PhoneService: Yes, No.*
- *MultipleLines: Yes, No, No phone service.*
- *InternetService: DSL, Fiber optic, No.*
- *OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies: Yes, No, No Internet service.*

Data Preparation:

- ***Missing or incomplete records:*** We noticed that 11 rows had null values for the TotalCharges. On further review, we also noted that all these clients have a tenure of '0' . We removed all these rows from the dataset. Lastly, TotalCharges was an object in the original data set and it was changed to numeric.
- ***Inconsistent values and non-standardized categorical variables:*** We reviewed the unique values of the gender, SeniorCitizen, Partner, Dependents, PhoneService, MultipleLines, InternetService, OnlineSecuirty, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod and Churn columns. We didn't find any inconsistent values.
- ***Improperly formatted/structured data.*** We reviewed the shape of the data set using the shape() function. There were 7043 rows and 21 columns in the data set. We reviewed the formatting of the dataset using the head() function.
- ***Encoding.*** We encoded the feature SeniorCitizen from No and Yes to 0 and 1 so that it can be used in machine learning algorithms.
- ***Standardization.*** As the numerical values were distributed over different ranges, therefore we used a standard scaler to scale them to the same range.

Data Visualization:

We can see from the following graphs that the churn ratio is 26.6% among the customers. We can also see that gender does not play much role in the churn behavior of these customers i.e., the percentage of Male and Female customers who switched to other firms seems to be approximately the same.

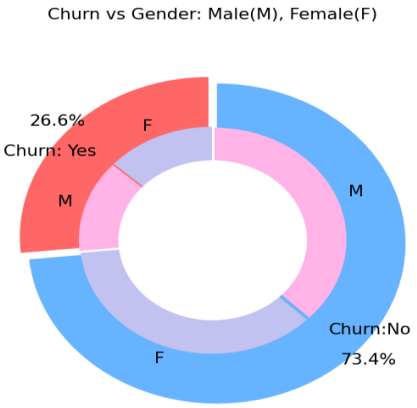

Figure 1: *Pie chart for the Frequency distribution of Churn vs Gender (Male, Female)*

We can see from the following graphs that most of the customers who switched to other firms had Month-to-Month (M2M) contracts.

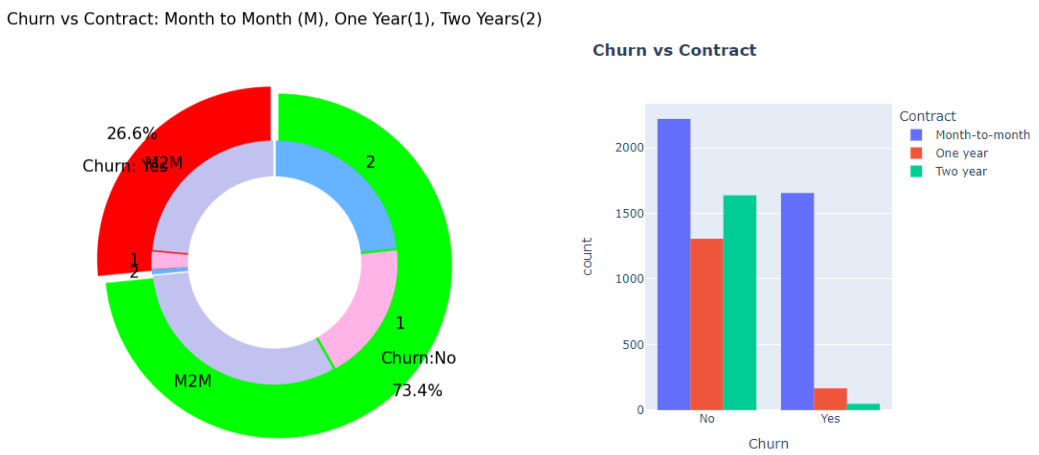

Figure 2: *Pie chart (a) and bar chart (b) for the Frequency distribution of churn vs Contracts (month to month, one year, two years)*

We can see from the following graph that most of the customers who transferred their services to other firms were paying through Electronic Checks and were using Fiber Optic services.

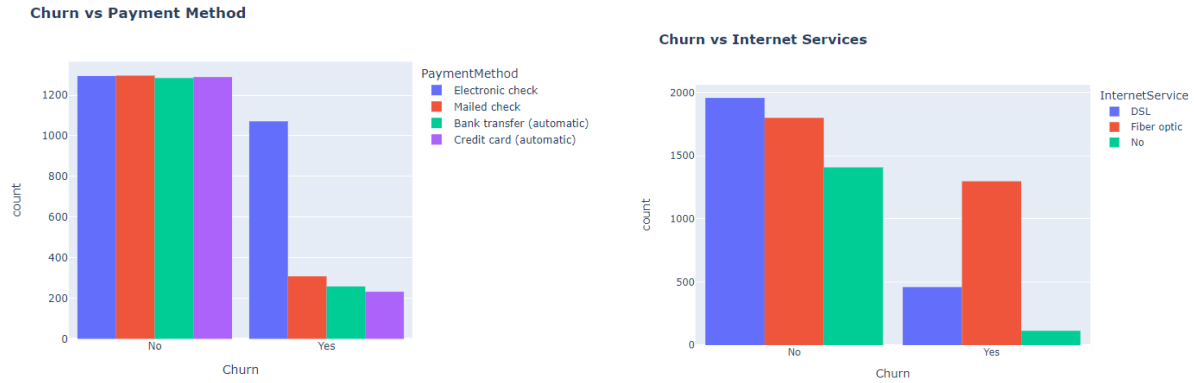

Figure 3: *Bar chart for the frequency distribution of churn vs Payment Method (a) and, Internet Services (b)*

We can see from the following graph that most of the customers who transferred their services to other firms were without dependents and were not in a relationship (without a partner).

Figure 4: *Bar chart for the frequency distribution of churn vs Dependents (a) and, Partner (b)*

We can see from the following graph that customers with higher Monthly Charges are more likely to transfer their services to other firms.

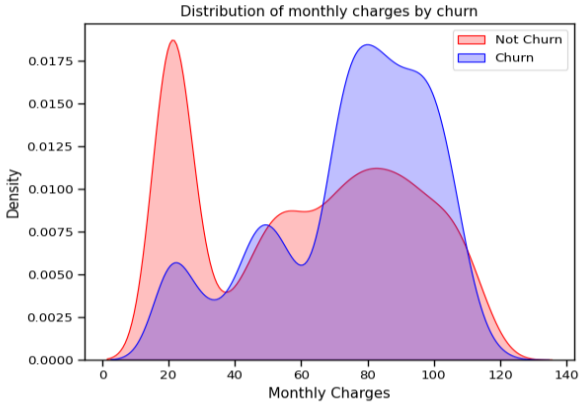

Figure 5: *Frequency distribution of monthly charges*

We can see from the following graph that customers with less Tenure with the firm are more likely to transfer their services to other firms.

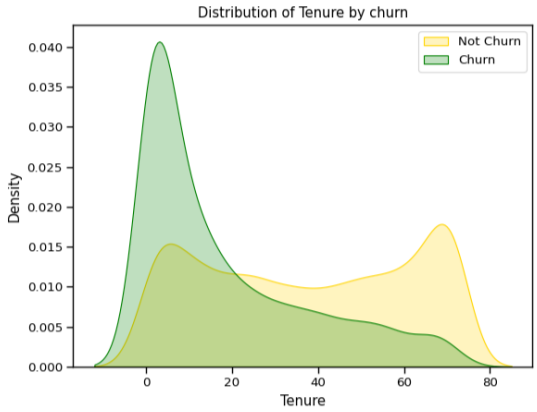

Figure 6: *Frequency distribution of tenure by churn*

Please see Appendix A for more data visualization graphs.

Tools and Challenges:

Tools commonly used for data preparation and analysis include Python, libraries like Pandas, Scikit-Learn, and potentially data visualization libraries such as Matplotlib or Seaborn. Jupyter notebooks were used to document and visualize the steps.

Challenges with the data included dealing with missing or inconsistent data, selecting appropriate features, handling categorical variables, and ensuring that the data is suitable for the chosen machine learning algorithms.

In conclusion, the data preparation process for the customer churn model involved various steps such as data loading, cleaning, transformation, and feature selection. The tools and techniques used depended on the specific requirements of the analysis, and challenges were addressed through careful consideration and domain knowledge.

Model Design

Data Preprocessing and Machine Learning Model Selection

Before creating a test set, commonly used variables were defined for easy reference. Afterwards, a test set was created to avoid later issues such as overfitting. As stated on pages 55-56 in Geron (2023), peering early into a test set may lead you to observe an interesting pattern within the data that may lead you to select a particular machine-learning model. This is also known as data snooping bias. Furthermore, testing data should be set aside prior to preprocessing as statistics used, such as the mean, should only be derived from the training data as this prevents data leakage (Luvsandorj, 2022).

When preparing the data for the machine learning algorithms, we made use of the Scikit-Learn Pipeline class to aid with the sequences of data transformation. Since numerical and categorical attributes are handled differently, two separate pipelines were made. Regarding the numerical attribute pipeline, a SimpleImputer instance was created which specified that we wanted to replace each attribute's missing value with the mean of that attribute. Furthermore, we also applied feature scaling within the numerical attribute pipeline. This was done because machine learning algorithms do not perform well when numerical attributes have varying scales (Geron, 2023, p.75). In this numerical attribute pipeline, min-max scaling was selected as it is the simplest; each value within the attribute was shifted so they range from -1 to 1 (Geron, 2023, p.76).

The categorical attribute also had a SimpleImputer instance, however, the strategy to fill missing values was different compared to the numerical attribute pipeline. The imputation strategy to fill the missing values was achieved by replacing the missing values with the fill_value parameter and specified that each attribute's missing value should be replaced with a constant value. In addition, OneHotEncoder was used to create one binary attribute per category as most machine-learning algorithms prefer working with numbers (Geron, 2023, p.72). OneHotEncoder was selected instead of OrdinalEncoder or the Pandas function get_dummies() because it is more appropriate for this data set. If OrdinalEncoder was used, the ML algorithms may have assumed that two values are more similar than distant values, which may be fine for ordered categories (Geron, 2023, p.72), but this is not the case for attributes like PaymentMethod or OnlineSecurity. The Pandas function get_dummies() was not selected because OneHotEncoder remembers which categories it was trained on while get_dummies() does not (Geron, 2023, p.73).

Lastly, we made use of ColumnTranformer to handle categorical and numerical columns simultaneously which streamlines the data transformation steps. To summarize, our pipeline splits the customer churn data into numerical and categorical groups, preprocesses the categorical and numerical groups in parallel, concatenates the preprocessed data from the categorical and numerical features and then passes the preprocessed data into the models we selected (Luvsandorj, 2022). The benefit of this approach is that storing interim results for the training and testing dataset is no longer a requirement (Luvsandorj, 2022).

In this project, we compared seven machine learning models and their algorithms with their default hyperparameters and used accuracy to measure classification performance. Recall that accuracy is the proportion of predictions that are correct (Bruce et al., 2020, p.220). The seven algorithms selected were logistic regression, a random forest classifier, a decision tree classifier, an adaptive boosting classifier, a gradient boosting classifier, the K-Nearest Neighbours classifier, and a support vector machine. The accuracy scores for the test set and training set are in Table 1.

Table 1: *Results of each classification model accuracy on the testing and training data. Bolded values in the training and testing columns are the models with the highest accuracy.*

Model	Training Accuracy	Testing Accuracy
Logistic Regression	80.43	**82.26**
Random Forest	**99.88**	78.25
Decision Tree	99.88	73.70
Adaptive Boosting	80.85	79.25
Gradient Boosting	83.02	79.53
K-Nearest Neighbours	85.80	75.62
Support Vector Machine	82.26	82.26

The model that performed the best on the training data set was the RandomForestClassifier with an accuracy score of 99.88%. However, it is well known that random forests tend to overfit data when the default hyperparameters are used (Bruce et al. 2020, p.269). We will address hyperparameter tuning a random forest model later. Note that we could have selected the DecisionTreeClassifier as it too had the highest training accuracy, but decided to move forward with the random forest since decision trees are the fundamental components of random forests (Geron, 2023, p. 195) and random forests also have one important extension. In addition to sampling records, the algorithm also bootstraps sampling variables at each split (Bruce et al., 2020, p. 261). The model that performed the best on the testing data was a logistic regression model. Logistic regression is likely the best model for predicting customer churn because fitting a model to the testing set is more indicative of real-world performance. The reason for this is because models are trained on the training data making the model is optimized for that data. Thus, it does not really represent how the model would perform on data it has not seen before.

Cross Validation

Evaluation of the random forest and logistic regression was done by using Scikit-Learn's k-fold cross-validation feature. Briefly, we randomly split the training set into ten non-overlapping subsets called folds. The k-fold cross-validation feature trains and evaluates the random forest and logistic regression model ten times by picking a different fold for evaluation and using the nine other folds for training (Geron, 2023, p.90). When a ten-fold cross-validation was performed on the random forest the mean accuracy was 78.81% with a standard deviation of ~2%. Although the random forest does not look as good as it did earlier, the accuracy is still acceptable and may be more realistic as it is close to its testing accuracy value of 78.25%. A ten-fold cross-validation was also performed on the logistic regression model. The mean accuracy of the logistic regression model was 80.25% with a standard deviation of 1.04%. These values are very close to the logistic regression models' training accuracy of 80.43% and its testing accuracy of 82.26% making it a promising model as well.

Feature Importance

As mentioned previously, random forests tend to overfit data when the hyperparameters are set to their default. Despite this, one of the advantages of a random forest is its ability to determine which predictors are important automatically (Bruce et al., 2020, p.265). The RandomForestClassifier collects information about the importance of each feature and makes it accessible via feature_importances. After evaluating the output, it was decided to drop the five features which do not seem to play a significant role in the random forest model from the data set. These features were DeviceProtection, InternetService, TechSupport, OnlineSecurity and PhoneService.

Another test set was created using the train_test_split function before training the data with the dropped features on a RandomForestClassifier and a LogisticRegression model. Once again, we used accuracy to measure the model performance. For the random forest model, the training accuracy score dropped 0.02% from 99.88% to 99.86% and the testing accuracy minimally improved 0.21% from 78.25% to 78.46%. The logistic regression model accuracy score decreased on the training (from 80.43% to 79.77%) and testing set (from 82.26% to 79.25%). After evaluating the models, logistic regression models may be more sensitive when features are dropped from the dataset. There are a few caveats however; the first is we only used the features that the RandomForestClassifier collected information about and the second is we did not perform a vigorous investigation on the feature effects on our models as this is beyond the scope of this assignment. For a detailed discussion on determining feature importance refer to Bruce et al., 2020, p.265-269.

Hyperparameter Tuning

Many statistical machine learning algorithms are considered to be black box algorithms with knobs that can affect the performance of the model (Bruce et al., 2020, p.269). These knobs are called hyperparameters which need to be set before fitting a model (Bruce et al. 2020, p.269). Using the data with the dropped features, the RandomForestClassifier and LogisticRegression model were fine-tuned by adjusting their respective hyperparameters.

The approach used to adjust the hyperparameters for the RandomForestClassifier was to use SciKit_Learn's RandomizedSearchCV as it is the preferred method when the hyperparameter grid space is larger (Geron, 2023, p.93). In the RandomForestClassifier, we chose six different hyperparameters to tune, however, the two most important hyperparameters for random forest models are min_samples_leaf and max_leaf_nodes (Bruce et al., 2020, p.269). The reason why min_samples_leaf and max_leaf_nodes are important to fine-tune is because the random forest algorithm will fit smaller trees and therefore is less likely to produce untrustworthy results (Bruce et al. 2020, p.269-270). These two aforementioned hyperparameters were included while fine-tuning the random forest model. After performing hyperparameter tuning, and once again using accuracy to measure model performance on the training set and the testing set, the RandomForestClassifier produced non-spurious results achieving an accuracy score of 80.53% on the training data and 79.67% on the testing data. Recall that RandomForestClassifier had an accuracy score of 99.86% on the training data and an accuracy score of 78.46% on the testing data before fine-tuning the hyperparameters of the random forest model.

Hyperparameter tuning on the LogisticRegression model made use of SciKit-Learn's GridSearchCV as only two hyperparameters were used to fine-tune the model. The first was the penalty argument which chooses a regularization technique to control overfitting (Nield, 2022, p.199). The second was the solver argument which selects the minimizer (Bruce et al., p.211). It

was also taken into consideration that not all penalties are compatible with every solver (SciKit-Learn developers, 2023) and therefore we used the SciKit-Learn's LogistcRegression documentation to optimize how many penalties and solvers we could use. We selected the penalties l2 and none as they were compatible with four different solvers (lbfgs, newton-cg, sag, and saga). The penalty selected was l2 and the solver selected was lbfgs once fine-tuning the model was completed. Unsurprisingly, hyperparameter tuning had no effect on the logistic regression models accuracy (79.77% on the training set 79.25 on the test set) as l2 regularization and the lbfgs solver are the default hyperparameters for Scikit-Learn's LogisticRegression model (Geron, 2023, p.172). This suggests that logistic regression models may be more sensitive to feature selection when predicting customer churn.

Model Evaluation

Performance Measures

Although accuracy was used as the performance metric to measure machine learning model performance, accuracy is generally not the preferred performance metric for classifiers (Geron, 2023, p.107) and is horrendously misleading for classifications problems (Neild, 2022, p.220). A much better way to measure the performance of a classifier is to look at confusion matrices (Geron, 2023 p.108). A confusion matrix is a grid that shows the predictions against actual outcomes displaying true positives, true negatives, false positives, and false negatives (Neild, 2022, p.220). In addition to confusion matrices, we also plotted precision/recall curves using Scikit-Learn's PrecisionRecallDisplay to select a good precision/recall tradeoff. Recall that there is a trade-off between precision and recall; increasing precision reduces recall and vice versa (Geron, 2023, p.111).

Since there is a trade-off between recall and specificity, we also used the metric "Receiver Operator Characteristics" (ROC) curves because it captures the trade-off between precision and recall (Bruce et al., 2020, p.224). The ROC curve plots recall (sensitivity) on the y-axis and specificity on the x-axis (Bruce et al., 2020, p.224). Even if the ROC curve is a valuable graphical tool, it does not have the capability to measure the performance of a classifier (Bruce et al., 2020 p.226). Instead, ROC curves are used to produce the area under the curve (AUC), a metric that assesses how well a classifier handles the trade-off between accuracy and the need to identify, in our case, customer churn (Bruce et al., 2020, p.228). To compare our logistic regression model to our random forest model, we used the area under the curve metric with their respective ROC curves to determine what model to use.

Random Forest Performance Measures

Looking at the RandomForestClassifier confusion matrices, the model's performance on the training set correctly classified 79.44% of the observations; 67.88% of clients will remain with the service provided by the telecommunication company while 11.56% will leave or churn (Figure 7). Furthermore, 15.02% were wrongly classified as remaining with the company (false positive or type I errors) while 5.55% were misclassified as customers that left (false negatives or type II errors) (Figure 7). Overall, 20.57% of the training data were misclassified. The random forest model confusion matrix on the testing set performed very similarly. Looking at Figure 7, it correctly classified 79.67% of the data (67.73% true positives and 11.94% true negatives) and misclassified 20.33% of the observations (14.64% false positives and 5.69% false negatives). Note that a perfect classifier would only have true positives and true negatives with non-zero values only on its main diagonal (Geron, 2023, p.109).

Precision measures the accuracy of a predicted positive outcome while recall measures the strength of the model to predict a positive outcome (Bruce et al., 2020, p.223). Looking at the precision-recall curve for the RandomForestClassifer (Figure 8), the average precision (AP) value is 63%, which also offers the greatest recall (for a more in-depth discussion about mean average precision, see Geron, 2023, p.529). The AUC metric for the RandomForestClassifer was 0.85 on the training set and 0.83 for the testing set (Figure 8). Recall that the AUC metric is the value in which our RandomForestClassifier handles the trade-off between overall accuracy and identifying customer churn. A completely ineffective classifier (the diagonal red line) would have an AUC of 0.5 (Bruce et al., 2020, p.226).

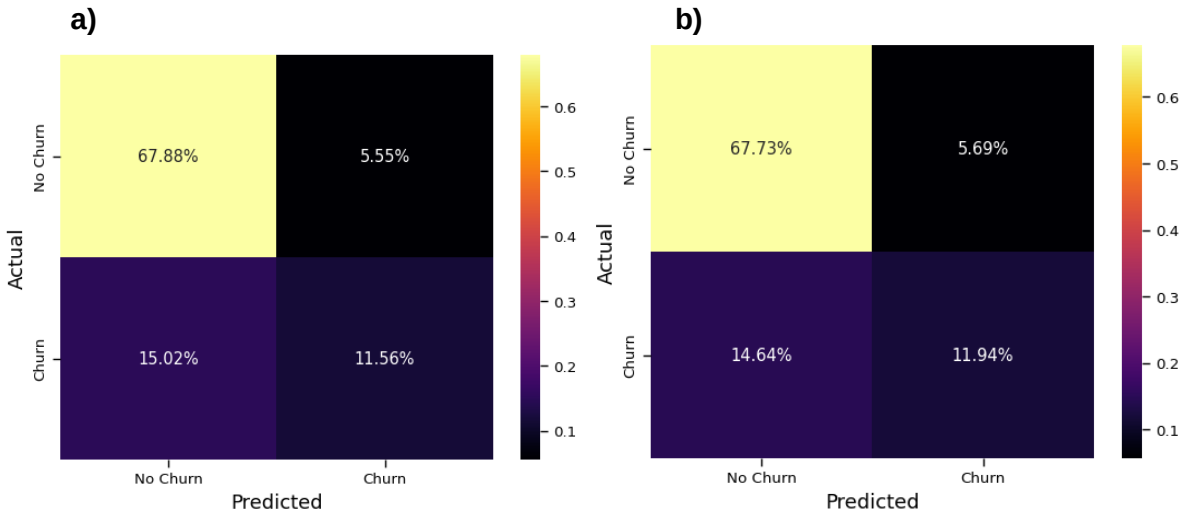

Figure 7: *Confusion matrices for the RandomForestClassifier on the training set (a) and on the testing set (b).*

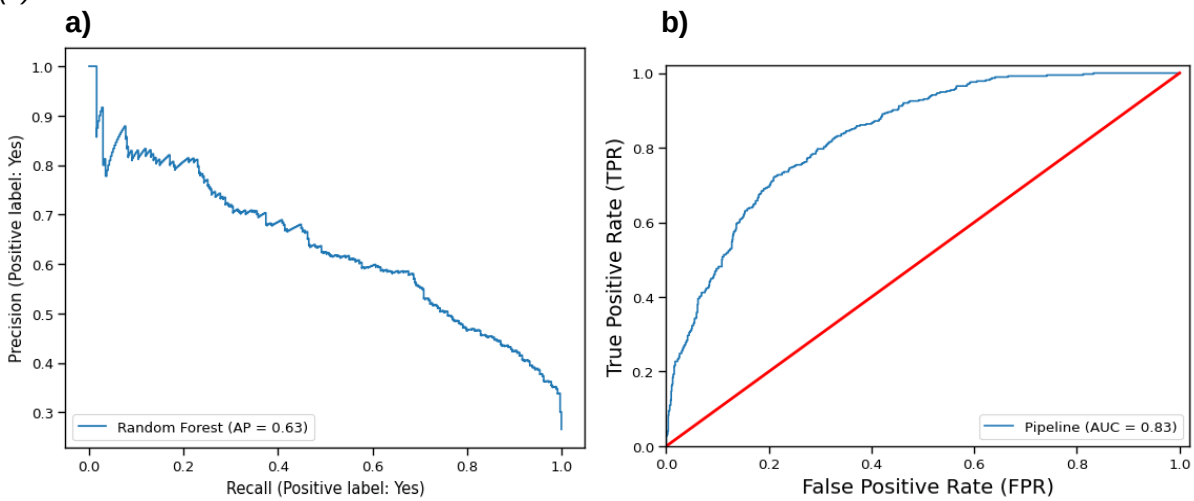

Figure 8: *The precision vs recall with its average precision value (a) and the ROC AUC curve with its ROC AUC score (b) for the RandomForestClassiefier.*

Logistic Regression Performance Measures

Regarding the LogisticRegression model confusion matrices, the model's performance on the training set correctly classified 79.51% of the observations; 65.55% of clients will remain with the service provided by the telecommunication company while 13.96% will leave or churn (Figure 9).

Furthermore, 12.62% were wrongly classified as remaining with the company while 7.88% were misclassified as customers that left (Figure 9). Overall, 20.5% of the training data were misclassified. The logistic regression model confusion matrix on the testing set performed very similarly, correctly classifying 79.25% of the data (65.25% true positives and 14.00% true negatives) and misclassifying 20.75% of the observations (12.58% false positives and 8.17% false negatives – Figure 9).

Looking at the precision recall curve for the LogisticRegression model the average precision (AP) value is 62% (Figure 10). This is very similar to the AP of the RandomForestClassifier. The AUC metric for the LogisticsRegression model was 0.84 on the training set, and like the RandomForestClassifier the AUC value was 0.83 for the testing set (Figure 10).

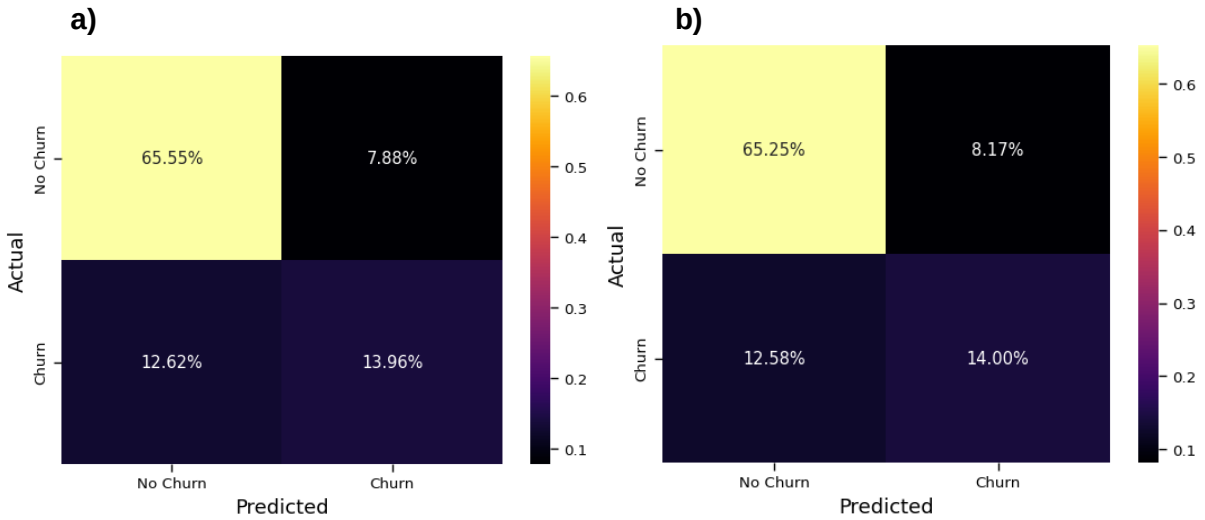

Figure 9: *Confusion matrices for the Logistic Regression model on the training set (a) and on the testing set (b).*

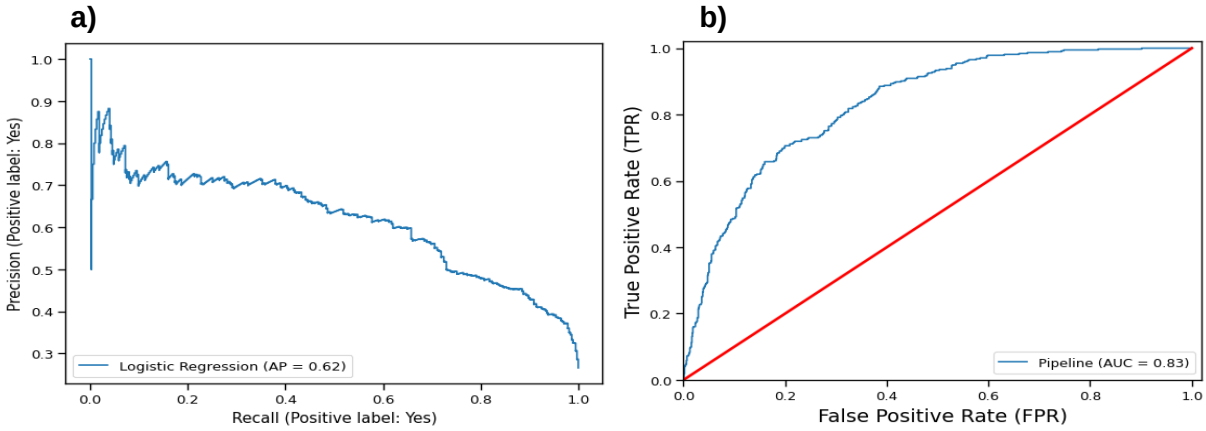

Figure 10: *The precision vs recall with its average precision value (a) and the ROC AUC curve with its ROC AUC score (b) for the LogisticRegression model.*

Conclusions

The objective of this assignment was to compare a variety of classification models and create a useful model to predict customer churn from the Telco Customer Churn dataset. We predicted that logistic regression would perform the best as they are known to be the workhorse model for predicting probabilities and classification of data (Nield, 2022, p.226). After the model evaluation was completed for the seven models selected, we decided to move on with the machine learning algorithm that performed best on the training set (RandomForestClassifer) and the testing set (LogisticRegression).

For the RandomForestClassifier, we did create a useful model after hyperparameter tuning as it correctly classified 79.44% of the training data and 79.25% of the testing data when looking at the confusion matrices. The AUC metric for this model on the testing data was 0.83. Regarding LogisticRegression, it too was a useful model but was not responsive to hyperparameter tuning. Instead, our logistic regression model was more sensitive to feature selection as the accuracy of the testing model dropped from 82.26% to 79.25%. Nevertheless, despite having a lower accuracy score after dropping features that did not play a significant role in the random forest model, the LogisticRegression model was very comparable to the RandomFoerstClassifier. The LogisticRegression model was able to correctly classify 79.51% of the training data and 79.25% of the testing data. In addition, the AUC score was identical to that of the RandomForestClassifier at 0.83 on the testing data.

With reference to what we learned about our dataset, it is possible that the data may have been noisier than anticipated. One possible indication of this occurred during hyperparameter tuning. As previously discussed, hyperparameters played a significant role to prevent the RandomForestClassifier from overfitting the data. One possible reason why the random forest tended to overfit the data before and after feature selection is that the data may have been noisy. As stated on page 269 in Bruce et al. (2023), not adjusting the hyperparameters min_samples_leaf and max_leaf_nodes will often cause random forest models to overfit the data, especially if it is noisy.

To improve our models in the future, one step that was not taken during data preprocessing was checking the distribution of numerical features. It is possible that MonthlyCharges, TotalCharges, or tenure may have had a heavy tail. Since min-max scaling and standardization flatten values into a small range, and machine learning algorithms typically do not like this, an attempt to make the distribution symmetrical before feature scaling may have improved our models (Geron, 2023, p.76). Another step we could have taken is to make sure that machine learning was applicable to our problem. The use of a dummy classifier to serve as a baseline model could be useful by comparing the prediction accuracy of our selected machine learning algorithms in this assignment to the baseline model to see if our selected machine learning algorithms outperformed the dummy classifier model (Iglesias Moreno, 2021).

In conclusion, both the RandomForestClassifier and LogisticRegression were successful in predicting customer churn. Our prediction was almost correct as LogisticRegression did the best job in predicting customer churn before features were removed from the dataset. Nevertheless, LogisticRegression performed almost identical to the RandomForestClassifier after hyperparameter tuning was performed on the random forest model. Since the logistic regression required fewer steps to essentially achieve the same AP and AUC values as the RandomForestClassifier, the claim by Nield (2022, p.226) seems to hold true; logistic regression is the workhorse model for predicting probabilities and classification on data.

References

1) *Bruce, Peter., Bruce Andrew., & Gedeck Peter. Practical Statistics for Data Scientists. 2nd ed. O'Reilly, 2020.*
2) *Geiler, L., Affeldt, S., & Nadif, M. (2022). A survey on machine learning methods for churn prediction. International Journal of Data Science and Analytics, 14(3), 217-242.*
3) *Geron, Aurelien. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. 3rd ed. O'Reilly, 2023.*
4) *Iglesias, Moreno. (2023, August, 10). End-to-end Machine Learning Project: Telco Customer Churn. Medium. https://towardsdatascience.com/end-to-end-machine-learning-project-telco-customer-churn-90744a8df97d*
5) *Prasad, Bharti. Telecom Customer Churn Prediction. https://www.kaggle.com/code/bhartiprasad17/customer-churn-prediction*
6) *Luvsandorj, Zolzaya. (2023, August, 02). From ML Model to ML Pipeline. Medium. https://towardsdatascience.com/from-ml-model-to-ml-pipeline-9f95c32c6512*
7) *Nield, Thomas. Essential Math for Data Science. O'Rielly, 2022*
8) *Scikit-Learn Developers. (2023, August, 05). 1.1.11 Logistic regression. SciKit-Learn. https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression*

Appendix:

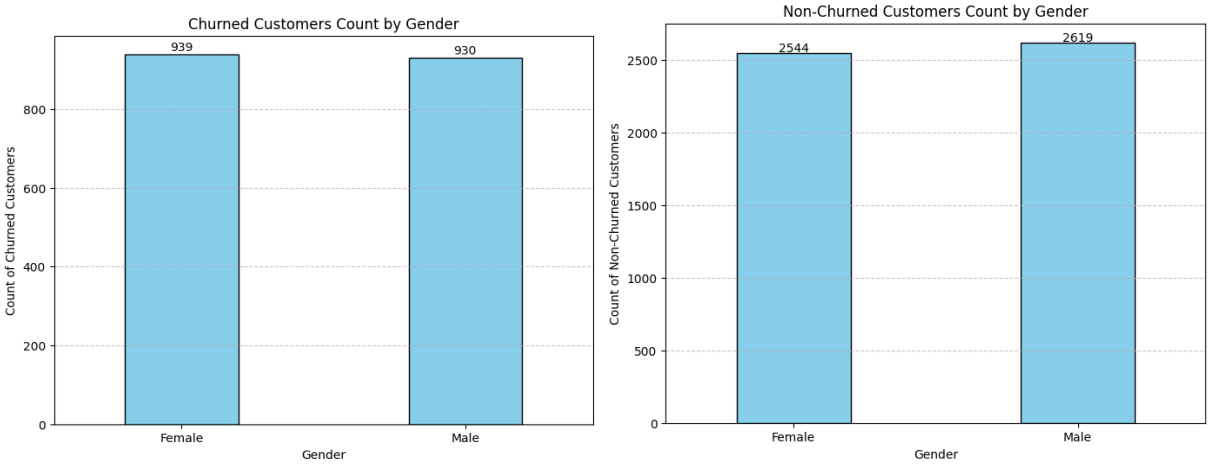

Figure 11: *Bar chart for the customer churn count (a) and, non-churned count (b) by gender*

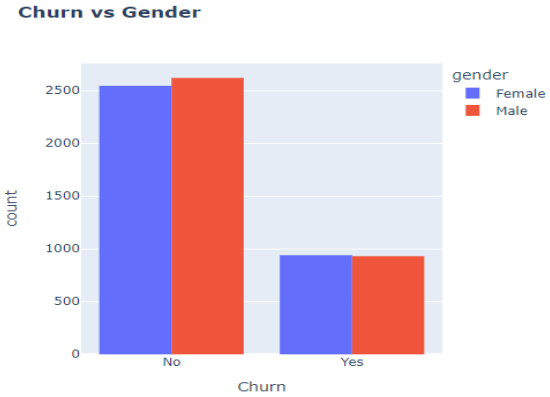

Figure 12: *Bar chart for the frequency distribution of churn vs gender*

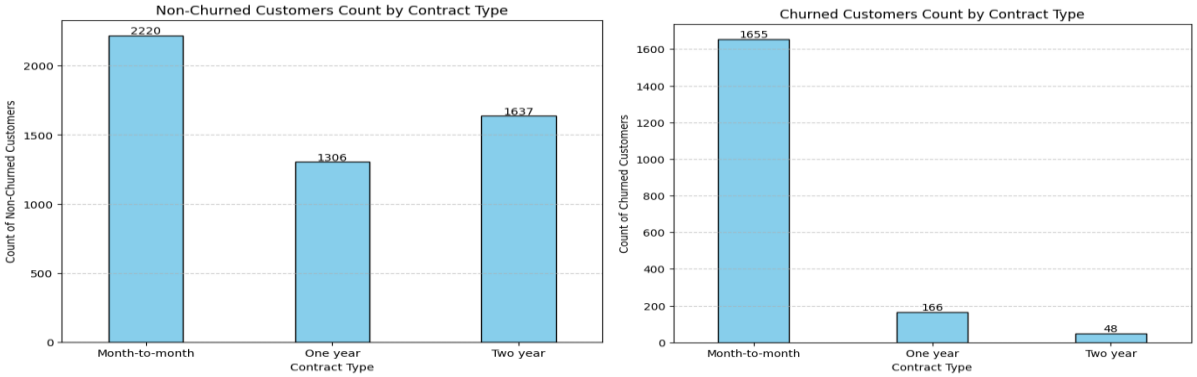

Figure 13: *Bar chart for the customer non-churned count (a) and, customer churned count by Contract type*

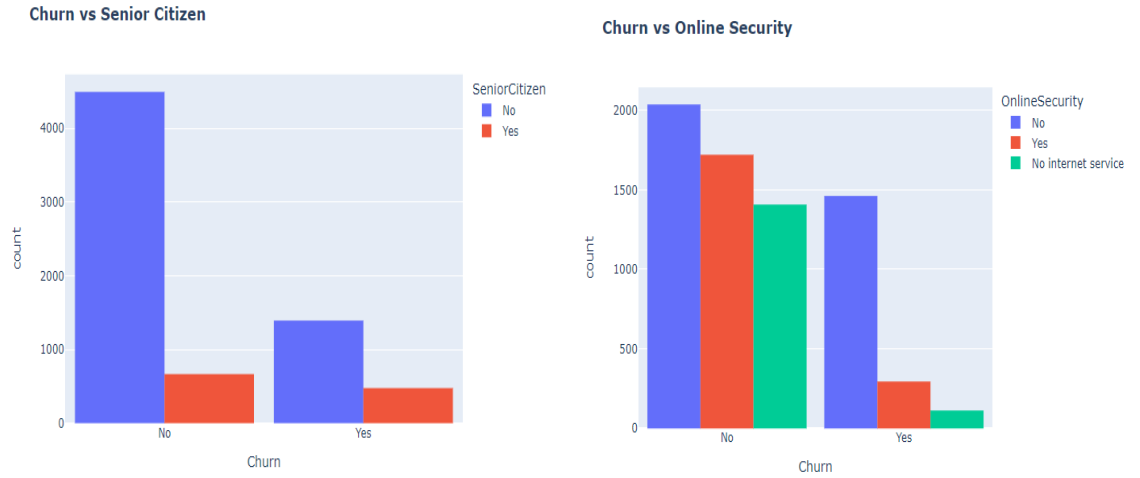

Figure 14: *Bar chart for the frequency distribution of churn vs senior citizen (a) and, churn vs online security (b)*

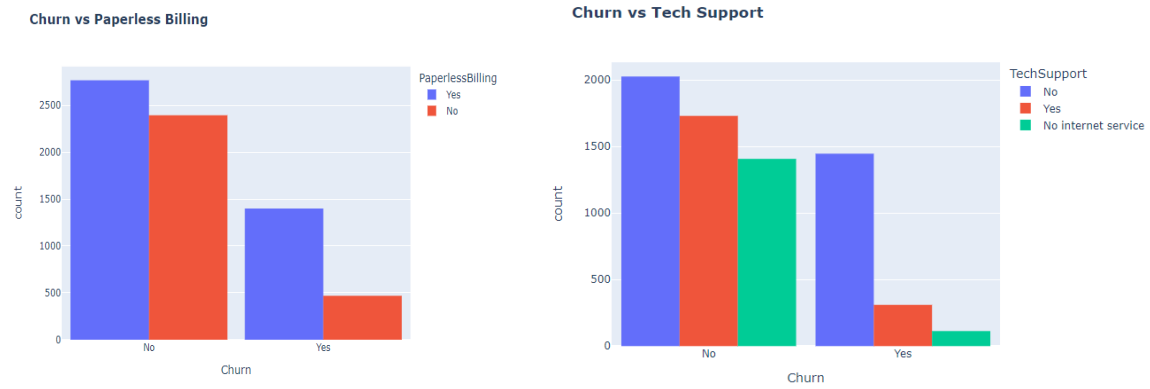

Figure 15: *Bar chart for the frequency distribution of churn vs Paperless billing (a) and, churn vs Tech-support (b)*

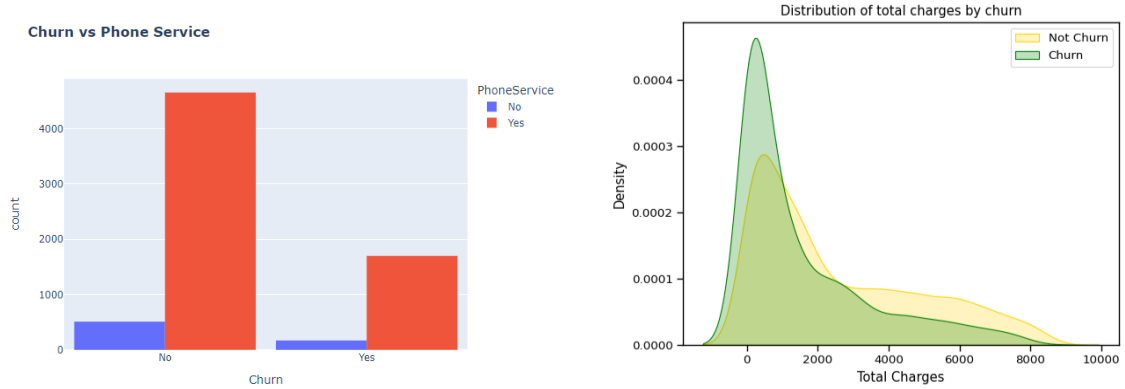

Figure 16: *Bar chart for the frequency distribution of churn vs phone service (a) and, frequency distribution of total charges by churn*

Summary

Telecom Customer Churn Analysis - Group 12

Introduction: In a highly competitive telecom Industry, preventing customer churn and retaining customers is critical. Since the telecom industry is characterized by consumers that can easily change providers, there is a need for a proactive approach to predict customer churn. In addition, it is also beneficial to determine what services are offered by the telecom industry that help to retain the customers. By leveraging exploratory data analysis, machine learning algorithms and feature importance, we aimed to predict customer churn with hopes of identifying key variables influencing consumer needs.

Data Source: Our project leveraged the Telco Customer Churn dataset, which compiled telecom customer churn data. Our dataset is from the following source. https://github.com/treselle-systems/customer_churn_analysis/blob/master/WA_Fn-UseC_-Telco-Customer-Churn.csv

Objective: Our project focuses on analyzing the churn rate of the Telco Customer Churn dataset. Our main objective was to select an appropriate model that accurately predicts customer churn. To achieve this, we leveraged several machine learning classification algorithms to predict customer churn. The models selected were logistic regression, random forest, decision trees, adaptive boosting, gradient boosting, KNN classification and support vector machines. We predict that logistic regression will perform the best as they are known to be the workhorse model for predicting probabilities and classification of data (Nield, 2022, p.226).

Model Design: To summarize our model design, we developed a pipeline that splits the customer churn data into numerical and categorical groups, preprocesses the categorical and numerical groups in parallel, concatenates the preprocessed data from the categorical and numerical features and then passes the preprocessed data into the models we selected (Luvsandorj, 2022). The benefit of this approach is that storing interim results for the training and testing dataset is no longer a requirement (Luvsandorj, 2022).

We then evaluated each model using accuracy as a preliminary performance metric for the seven models stated in the objective section. It was decided to perform cross-validation, feature importance, and hyperparameter tuning on the machine learning algorithm that performed best on the training set (RandomForestClassifer) and the testing set (LogisticRegression).

Model Evaluation: For the RandomForestClassifier, we did create a useful model after hyperparameter tuning as it correctly classified 79.44% of the training data and 79.25% of the testing data when looking at the confusion matrices. The AUC metric for this model on the testing data was 0.83. Regarding LogisticRegression, it too was a useful model but was not responsive to hyperparameter tuning. Instead, our logistic regression model was more sensitive to feature selection as the accuracy of the testing model dropped from 82.26% to 79.25%. Nevertheless, despite having a lower accuracy score after dropping features which do not play a significant role in the random forest model, the LogisticRegression model was very comparable to the RandomFoerstClassifier. The LogisticRegression model was able to correctly classify 79.51% of the training data and 79.25% of the testing data. In addition, the AUC score was identical to that of the RandomForestClassifier at 0.83 on the testing data.

Conclusions: In conclusion, both the RandomForestClassifier and LogisticRegression were successful in predicting customer churn. Our prediction was almost correct as LogisticRegression did the best job in predicting customer churn before features were removed from the dataset. Nevertheless, LogisticRegression performed almost identical to the RandomForestClassifier after hyperparameter tuning was performed on the random forest model. Since the logistic regression required fewer steps to essentially achieve the same AP and AUC values as the RandomForestClassifier, the claim by Nield (2022, p.226) seems to hold true; logistic regression is the workhorse model for predicting probabilities and classification on data.

In this project, we successfully addressed the challenge of analyzing customer churn in a telecom company. By employing a combination of data preprocessing, exploratory analysis, feature importance, and machine learning modeling, we developed an effective predictive tool. The insights gained from the analysis not only provided a better understanding of churn dynamics but also helped us to better understand the customers and thus improve customer retention.