Logistic Regression for Predicting Customer Churn in the Telecom Sector

Ali Mirza, Andres Ramirez, Gizelle Lao, Rafael Muniz, and Trung Le

Business Insights & Analytics, Humber College Institute of Technology and Advanced Learning, Toronto, Canada

***Abstract* – The utilization of machine learning is an important aspect of supporting businesses by garnering insights that form data-driven solutions. In constantly evolving and competitive industries, machine learning presents solutions by classifying and predicting customer outcomes. For example, in the telecommunications industry, an important metric is customer churn, where existing customers cease to use a company's services and products, leave the company and explore alternatives offered by competitors [1]. In this paper, three algorithms - Decision Trees, Logistic Regression, and K-Nearest Neighbors models - are explored to understand the classification and prediction of churn for a telecommunications company. In turn, based on the accuracy score, Logistic Regression is the most effective model for predicting customer churn with an accuracy of 80%.**

***Index Terms*— Telco Customer Churn, Decision Tree, Logistic Regression, and K-Nearest Neighbors.**

I. INTRODUCTION

In the telecommunication business, as in other industries, competition is intense. Competitors often come up with strategies to take away customers for making bigger profits. Therefore, a company needs to find ways to satisfy its clients and make them stay using the company's services, thereby avoiding churn.

Based on business research, it is 5 to 10 times cheaper to retain an existing client than it is to obtain a new one [2]. Hence, the telecom business needs to predict whether an existing client will likely churn using the "Telco Customer Churn" dataset from Kaggle [3], to take the necessary measures to retain that client, for example, by targeted marketing campaigns. A lower churn will reduce cost, increase revenue, and lead to higher profitability.

In this paper, we will take a data-driven approach to solving the business problem of customer churn. In particular, Decision Tree, Logistic Regression, and K-NN machine learning models will be trained with data and tested for making predictions about customer churn, followed by our conclusions about the best-performing model and our recommendations for improvement.

This paper is divided into the following sections: Section II covers the literature survey, Section III discusses data preparation, Section IV comprises the application and evaluation of Decision Trees, Logistic Regression, and K-NN algorithms, and Section V contains our conclusion and recommendations.

II. LITERATURE SURVEY

As discussed previously, it is known that telecommunication businesses are very sensitive to their customer base, and losing it, in some way, can generate an awful financial outcome. In that sense, churn is the main aspect that companies analyze to make improvements in customer retention.

Given the importance that is given to churn in this sector, different research has been conducted with it, to predict the likelihood of a customer churn in the short-term, through the use of different machine learning algorithms.

In the first paper, Tianpei Xu et al [2] proposed a solution to this problem using different types of machine learning algorithms in a very interesting type of approach, but complex at the same time. They used Xgboost, Logistic Regression, Decision Tree, and Naive Bayes, checked which one gave the best accuracy, that was Xgboost, and decided to use that one in the first layer of the stacking model approach. After, in the second layer, using the output of the first layer, they used the remaining algorithms chosen (Logistic Regression, Decision Tree, and Naive Bayes). With this type of stacking model approach, the accuracy was higher (reaching 98.09%) than using the models isolated, as used traditionally.

On the other hand, the traditional way of using the machine learning models, using on them the same preprocessed data, with no layers of models involved, was suggested by Nilam N. A. Sjarif et al [4], which used Pearson Correlation and K Nearest Neighbor (KNN) models, and observed that the KNN performed better (with an accuracy of 97.78%). Curiously, both papers had similar results in terms of accuracy, despite the level of complexity of their solutions.

Therefore, in this paper, a churn prediction model will be proposed using three different models (KNN, Decision Tree, and Logistic Regression), in the most objective and simple approach, with just one layer of application, and measuring the different accuracies obtained by each one of them, to decide which one to use.

III. DATA PRE-PROCESSING

The dataset contained 7,043 records and 21 variables, which consisted of 18 categorical and 3 numeric variables, as described in Table 1.

Table 1: Attribute Descriptions

Attribute	**Description**
customerID	Customer Identification
gender	male/female
SeniorCitizen	1 = Yes / 0 = No
Partner	Yes/No
Dependents	Yes/No
tenure	No. of months as a customer
PhoneService	Yes/No
MultipleLines	Yes/No/No phone service
InternetService	DSL/Fiber optic/No
OnlineSecurity	Yes/No/No internet service
OnlineBackup	Yes/No/No internet service
DeviceProtection	Yes/No/No internet service
TechSupport	Yes/No/No internet service
StreamingTV	Yes/No/No internet service
StreamingMovies	Yes/No/No internet service
Contract	Month-to-month, One year, Two year
PaperlessBilling	Yes/No
Payment method	Electronic check/Mailed check/ Bank transfer/Credit card
MonthlyCharges	The amount charged to the customer monthly
TotalCharges	The total amount charged since becoming a customer
Churn	The customer left within the last month? Yes/No

The 'customerID' variable was found to be an irrelevant predictor of churn and was removed from the dataset.

There were 11 missing values in the TotalCharges column. It was observed that the TotalCharges column was approximately equal to the product of the tenure and MonthlyCharges columns. This suggested that the TotalCharges variable itself was not an independent predictor of churn. As expected, it turned out to be highly correlated with tenure (0.8) and MonthlyCharges (0.7). Since the TotalCharges variable was not an independent predictor of churn, it was deleted from the dataset. This also removed the missing values from the dataset.

The data was checked for duplicates, and 27 such records were found. They were removed, reducing the dataset size from 7,043 records to 7,016 records.

The dataset was checked for any columns that had no variation (columns that contained a single unique value). It was found that no such columns existed in the dataset.

Unique values for every categorical variable were checked manually and no incorrect values were found. Numeric columns were checked for incorrect data by computing summary statistics. All values were positive and min-max values were within the expected ranges. This information was further verified from the box plots in Figure 1.1 and Figure 1.2, which show that there were no outliers in the dataset. Overall, there were no incorrect values or outliers in the dataset.

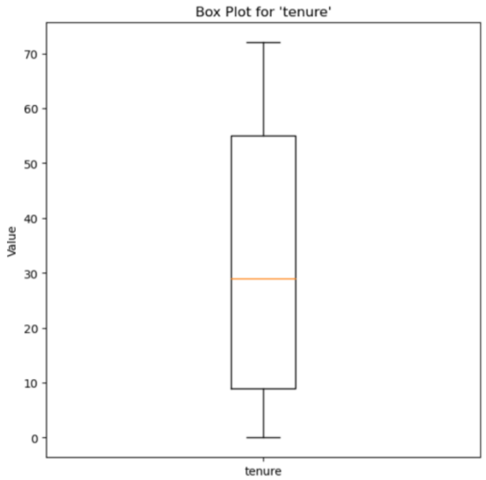

Figure 1.1: Detecting Outliers with Box Plots

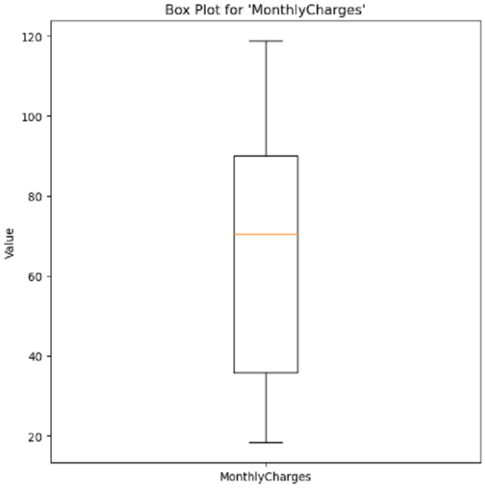

Figure 1.2: Detecting Outliers with Box Plots

The summary statistics and box plots for the numeric variables revealed that 'tenure' ranged from 0 to 72 months, while 'MonthlyCharges' ranged from 18.25 to 118.75 dollars. The scales of both variables were not drastically different from each other. Therefore, it was not necessary to rescale them.

To get the data ready for training, the categorical variables were transformed to numeric through label encoding. One-hot encoding was not used because it would have led to a huge increase in the number of variables, which would have negatively affected model performance.

All remaining variables in the dataset were deemed to be relevant predictors of churn. The goal was to maximize model accuracy, so all variables were considered. Therefore, feature selection or dimensionality reduction was not applied.

The dataset was split into training and test sets in a 60:40 ratio, respectively. Out of a total of 7,016 records, 4,209 were allocated to the training set while 2,807 were assigned to the test set.

IV. MODEL TRAINING AND PERFORMANCE

Decision Tree

The Decision Tree model was chosen because it is one of the effective ways of dealing with the churn prediction problem, as already mentioned in the literature survey. It also gives a good idea, to whoever is going to analyze it with a deeper perspective, because it gives the main characteristics that are the decision points in each layer, providing the important variables that are used to make the “decision”.

In this case, the model was executed and, after fine-tuning the model’s parameters [5], the following parameters in Table 2 showed the best result:

Table 2: Classification Tree Parameters

Parameter	Value
max_depth[1]	6
max_leaf_nodes[2]	30
min_samples_leaf[3]	40
min_samples_split[4]	130

The performance of the model, using the parameters listed above, is summarized in Table 3 and Table 4:

Table 3: Classification Report for the Decision Tree

	precision	recall	f1-score
No	0.82	0.90	0.86
Yes	0.61	0.43	0.51

Table 4: Confusion Matrix for the Decision Tree

Accuracy 0.78		Prediction	
		No	Yes
Actual	No	1,880	198
	Yes	414	315

The table above shows that the mentioned model has an accuracy of 78%. This result needs to be compared with the next models to decide which one has the better performance. But going further in the performance, it shows, in summary, that the model accurately got 315 true positives (correct number of predictions of clients that churned) and 1,880 true negatives (correct number of predictions of clients that didn’t churn).

Another aspect that we noticed from Table 4 is the difference between precisions (82% for “No” and 61% for “Yes”) and recalls (90% for “No” and 43% for “Yes”), which demonstrates that the model is not performing in the best way possible.

Logistic Regression

In this case, we applied logistic regression to a dataset for classification tasks. Logistic regression is a

[1] “The maximum depth of the tree.” [5]

[2] “Grow a tree with max_leaf_nodes in the best-first fashion. Best nodes are defined as relative reduction in impurity.” [5]

[3] “The minimum number of samples required to be at a leaf node.” [5]

[4] “The minimum number of samples required to split an internal node” [5].

popular linear classification algorithm used to model the relationship between a "yes/no" or "1/0" dependent variable and independent variables. The goal of this analysis is to predict the binary outcome based on the given set of features [6]. The data was then split into training and testing sets for model evaluation.

To find the best-performing logistic regression model, we performed hyperparameter tuning using GridSearchCV. We defined the hyperparameters to search over as follows: Penalty: Regularization term, either 'l1' or 'l2'. C: Inverse of regularization strength, with values [0.001, 0.01, 0.1, 1, 10]. Solver: Optimization algorithm, set as 'liblinear'. Max_iter: Maximum number of iterations for the solver to converge, with values [100, 500, 1000].

Penalty: We considered both L1 and L2 regularization because they have different effects on the model. L1 regularization can lead to sparse models by encouraging some coefficients to be exactly zero, while L2 regularization penalizes large coefficients [7].

C: We chose a range of C values to control the strength of regularization. Smaller C values result in stronger regularization, helping prevent overfitting, while larger C values allow the model to fit the training data more closely [8].

Solver: Since we have a relatively small dataset, 'liblinear' was chosen as the solver as it performs well on small datasets [9].

Max_iter: We experimented with different values to ensure that the optimization solver had enough iterations to converge to a solution.

After conducting GridSearchCV, the best hyperparameters were found to be: C = 0.1, Penalty = 'l1', Solver = 'liblinear', Max_iter = 100.

The classification report in Table 5 provides insights into the model's performance for both classes ('No' and 'Yes'). The report includes metrics such as precision, recall, and F1-score for each class. Additionally, the support column indicates the number of occurrences of each class in the test set. Also, we can observe the following: for class 'No' the model achieved a precision of 85%, recall of 90%, and an F1-score of 87%. This indicates that the model performs well in predicting negative instances. In contrast, for class 'Yes' the model shows a lower precision of 65%, recall of 53%, and an F1-score of 59%. While the performance for this class is relatively lower, it still provides some valuable predictive capabilities.

Table 5: Classification Report - Logistic Regression

	precision	**recall**	**f1-score**
No	0.85	0.90	0.87
Yes	0.65	0.53	0.59

The confusion matrix in Table 6 provides a more detailed view of the model's performance. It shows the number of true positives, true negatives, false positives, and false negatives. We can see that the model correctly classified 1,864 instances of class 'No' and 390 instances of class 'Yes'. However, it misclassified 214 instances of class 'No' as the class 'Yes' and 339 instances of class 'Yes' as the class 'No'. However, the overall model accuracy is 80%.

Table 6: Confusion Matrix - Logistic Regression

Accuracy 0.80		**Prediction**	
		No	Yes
Actual	No	1,864	214
	Yes	339	390

In conclusion, logistic regression was applied to a binary classification task, and hyperparameter tuning using GridSearchCV was performed to find the best model. The chosen hyperparameters (C = 0.1, Penalty = 'l1', Solver = 'liblinear', Max_iter = 100) resulted in a model with an accuracy of approximately 80%. The model showed good performance in predicting class 'No' but had some difficulty in classifying class 'Yes'. Further exploration and experimentation with feature engineering or trying different algorithms may improve the model's performance in the class 'Yes'. Overall, the logistic regression model provides a promising start, but further refinement may be necessary for more challenging classification tasks.

K-Nearest Neighbors (K-NN)

Another important model to effectively understand the classification and prediction of customer churn includes K-Nearest Neighbors (KNN), utilized in this report. The KNN model can identify patterns based on similarities between the data points. It can be used for classification tasks with categorical outputs or prediction tasks with numerical outputs [10].

It is integral to determine the appropriate value of K when creating the model, and it is achieved by selecting the value with the highest accuracy. The process of selecting K is arbitrary, but for binary classification models, it is recommended to use odd values for K [11]. In the case of predicting customer churn with a binary output (‘Yes’ and ‘No’), K = 35 was selected as it achieved an accuracy of 79% based on Table 7, which displays the highest K values based on accuracy. The rationale behind choosing the range of 40 was based on conducting various tests, and a larger set seemed appropriate for the size of the data set.

Table 7: Top 5 Highest K Values by Accuracy

k	Accuracy
35	0.790524
29	0.790524
28	0.790167
10	0.790167
30	0.790167

Following the model training, tests on the validation set were conducted to obtain the classification predictions as shown in Table 8 below.

Table 8: Confusion Matrix for KNN

Accuracy 0.79		Prediction	
		No	Yes
Actual	No	1,876	202
	Yes	386	343

This reflects that the model achieved a 79% accuracy in predicting values for customer churn. There were 1,876 true negatives, showing the model correctly predicted customers who wouldn’t churn and they didn’t. Yet, 202 false positives show the model incorrectly predicted customers would churn when they stayed. Also, the model identified 386 false negatives which reflect customers who were predicted to not churn but left. There were 343 true positives, accurately indicating the churned customers as predicted.

Further, the classification report shown in Table 9 reflects a higher precision score of 83% for the ‘No’ category and 63% in the ‘Yes’ category, indicating its accuracy in predicting instances for both ‘No’ and ‘Yes’. Additionally, the recall results show the model’s better performance in predicting ‘No’ as evidenced by the 90% but poor performance of 47% in recalling ‘Yes’. The F1-score has values of 86% for ‘No’ and 54% for ‘Yes’. In turn, these demonstrate that the model performs better at predicting ‘No’ values, where customers have not churned.

Table 9: Classification Report - KNN

	precision	recall	f1-score
No	0.83	0.90	0.86
Yes	0.63	0.47	0.54

V. CONCLUSION

In this paper, we explored and utilized the application of three machine learning algorithms: Classification Tree, Logistic Regression, and K-NN. Then we analyzed each model’s effectiveness in predicting customers who are likely to churn in the telecommunication industry based on the “Telco Customer Churn” dataset collected from Kaggle. As per Table 10 and Figure 2, our analysis showed that the Logistic Regression algorithm stood out as the best model with an overall accuracy of 80%.

Table 10: All model’s accuracy

Model	Accuracy
Decision Tree	0.78
Logistic Regression	**0.80**
K-NN	0.79

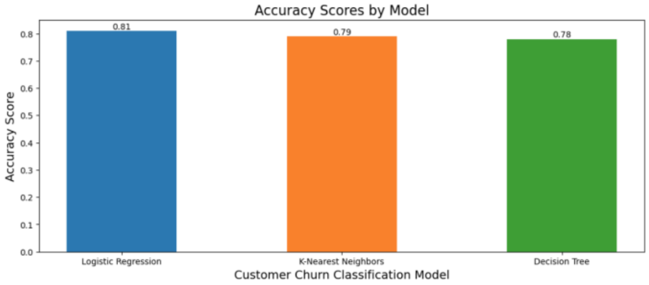

Figure 2: All model’s accuracy

Our research findings offer significant insights into the telecommunication industry, enabling them to classify customers who are about to churn based on product preferences, provided services, payment patterns, and usage. This study can be leveraged to enhance their product offerings and improve customer service and lead to higher customer retention rates and improve overall customer lifetime value. In addition, by understanding product preferences toward loyal customers' profiles, telecommunication companies can develop targeted segmentation strategies and offer tailored products that cater to specific needs, not only increasing customer loyalty but also attracting potential customers.

To improve model prediction accuracy, it is crucial to collect more data, explore alternative machine learning algorithms and employ effective feature engineering techniques. Furthermore, by integrating the most recent data, current product and service offerings, and in-depth feedback from clients, model performance can be enhanced, resulting in more precise predictions that empower telecommunication companies to devise more effective customer retention plans.

VI. REFERENCES

[1] Andrews, R., Zacharias, R., Antony, S., & James, M. M. (2019). Churn prediction in the telecom sector using machine learning. International Journal of Information, 8(2).

[2] Xu, T., Ma, Y., & Kim, K. (2021, May 21). Telecom Churn Prediction System Based on Ensemble Learning Using Feature Grouping. Applied Sciences. https://www.mdpi.com/2076-3417/11/11/4742/pd

[3] BLASTCHAR. (2018). *Telco Customer Churn*. Kaggle. Retrieved August 5, 2023, from https://www.kaggle.com/datasets/blastchar/telco-customer-churn

[4] Sjarif, N. N. A., Yusof, M. R., Wong, D. H.-T., Ya'akob, S., Ibrahim, R., & Osman, M. Z. (2019, July). A Customer Churn Prediction using Pearson Correlation Function and K Nearest Neighbor Algorithm for Telecommunication Industry. Int. J. Advance Soft Compu. Appl, 11(2). http://188.247.81.52/PapersUploaded/2019.2.4.pdf

[5] Scikit-learn. (n.d.). sklearn. tree.DecisionTreeClassifier — scikit-learn 1.3.0 documentation. Scikit-learn. Retrieved August 5, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[6] IBM. (n.d.). *What is logistic regression?* Retrieved August 5, 2023, from IBM: https://www.ibm.com/topics/logistic-regression#:~:text=Resources-,What%20is%20logistic%20regression%3F,given%20dataset%20of%20independent%20variables.

[7] Pykes, K. (2023, August 4). *Fighting Overfitting With L1 or L2 Regularization: Which One Is Better?* Retrieved from Neptune.AI: https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization#:~:text=L2%20regularization%20takes%20the%20square,the%20cost%20only%20increases%20linearly.&text=By%20this%20I%20mean%20the,to%20arrive%20at%20one%20point.

[8] Kumar, V. (2017, October 26). *What does a large C parameter mean in GridSearchCV?* Retrieved from Stack Overflow: https://stackoverflow.com/questions/46938122/what-does-a-large-c-parameter-mean-in-gridsearchcv#:~:text=As%20documented%20here%2C%20C%20is,prone%20to%20overfit%20the%20data.

[9] Saturn Cloud. (2023, July 10). *What Are the Different Python Solvers for Logistic Regression and How Do They Work?* Retrieved from Saturn Cloud: https://saturncloud.io/blog/what-are-the-different-python-solvers-for-logistic-regression-and-how-do-they-work/

[10] Shmueli, G., Bruce, P. C., Gedeck, P., Patel, N. R. (2019). Data Mining for Business Analytics: Concepts, Techniques, and Applications in Python. United Kingdom: Wiley.

[11] Brownlee, J. (2016, April 15). K-Nearest Neighbors for Machine Learning. Machine Learning Mastery. Retrieved [Date you accessed the page], from https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/