Telecommunication Customer Churn Prediction

ST4248 Group Project
Final Report

Group B5
Clarissa Lee Zixin (A0205284J)
Dong Mingjie (A0194481X)
Fan Zixian (A0267684L)
Song Yida (A0211188J)

Abstract

Customer churn refers to a customer leaving a company and hence no longer consuming said company's products or services. Predicting customer churn is useful for companies as they have a huge impact on the company's financial performance, and reputation. In this project, we hence explore methods to predict churn and to identify specific features that drive customer churn.

1 Motivation

Customer churn is a pressing challenge for companies, particularly in the telecommunications sector, where competition is intense due to the abundant available options for customers. Losing customers can significantly impact a company's financial performance and reputation while acquiring new customers can be more costly than retaining existing ones. Hence, it is important for companies to understand what entices new customers to join or stay, and what factors drive customers away. For this purpose, as highlighted in previous research[1], predicting and preventing customer churn is essential for companies to remain competitive in the long term. In this project, we hence explored methods to predict churn and to identify specific features that affect customer churn.

2 Description of Data

The data set was taken from Kaggle[2]. The data was collected from a company that provides home phone and Internet services in California. It includes information on customer attributes, services, account information, and demographics. It comprises over 7,000 rows, with each row representing a unique customer. The data includes 21 columns, and the variable `Churn` indicates whether a customer has churned (left) or not. The data consists of both numerical and categorical data types.

[1]Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. Expert Systems with Applications, 39 (1), 1414–1425.

[2]Telco Customer Churn (2019) Kaggle. Available at: https://www.kaggle.com/datasets/blastchar/telco-customer-churn.

3 Problem Statement

This project aims to develop a predictive model that identifies customers at risk of churning. By doing so, the model enables the telecommunications company to take proactive measures to retain customers, increase revenue, and maintain its market share. Additionally, the model can identify the specific features that influence customer churn, enabling the company to enhance its products and services.

4 Goal/Hypothesis

The goal of this project is to develop an accurate predictive model that can predict customer churn. We hypothesize that by leveraging the customer's demographic, account, and service-related data, we can create a model that can identify the features driving customer churn. It is also expected that the model will provide insights into the specific services that are most likely to cause customer churn, as well as the customer demographics that are most likely to churn.

5 Data Pre-processing

Churn refers to a customer leaving a company and hence no longer consuming said company's products or services. A customer can choose to:

- Churn: leave
- Not churn: stay or join a company's user base

all of which affect a company's profits. In this case, we assign churn: 1, not churn: 0.

For some of the qualitative variables in our dataset, we felt that their meanings were quite different from

one another, we hence decided to separate them. For instance, `InternetService` contains the levels: `FiberOptic`, `DSL`, and `No`. `No` refers to whether or not the customer has any internet service. For predictors like this, we separated each level into individual predictors instead. After that, the data set was left with 26 predictors.

6 Multicollinearity

Multicollinearity occurs when predictors exhibit correlation, which can lead to issues in regression analysis. To detect multicollinearity, we re-evaluated the VIF on the dataset with newly created predictors (Table 1). It was observed that, certain variables returned an infinite VIF due to their creation from a previously consolidated variable, as seen in the case of `InternetService`. Furthermore, we observed high VIF values exceeding 10 for other variables, indicating significant collinearity in the dataset.

7 Data Scaling

There is some non-normal behavior in some quantitative variables (Figure 1). For scaling purpose, our research considered scaling by considering the standard scaler, min-max scaler and quantile transformer on the data set for different models.

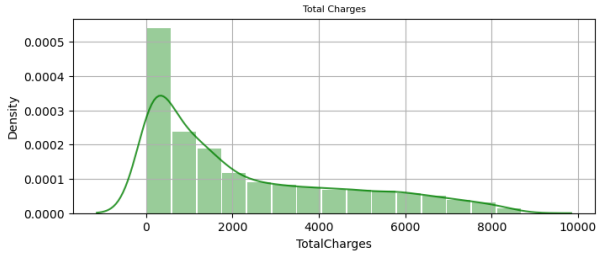

Figure 1: Histogram of `TotalCharges`

Table 1: Features' VIF

Predictors	**VIF**
Gender	1.00
Partner	1.46
Tenure	7.58
MultipleLines	5.5
OnlineBackup	6.80
TechSupport	6.48
StreamingMovies	24.16
MonthlyCharges	866.09
SeniorCitizen	1.15
Dependents	1.38
PhoneService	34.89
OnlineSecurity	6.34
DeviceProtection	6.92
StreamingTV	24.08
PaperlessBilling	1.21
TotalCharges	10.81
InternetServiceDSL	∞
InternetServiceNo	∞
ContractOneYear	∞
PaymentBankTransfer	∞
PaymentElectronicCheck	∞
InternetServiceFiberoptic	∞
ContractMonth	∞
ContractTwoYear	∞
PaymentCreditCard	∞
PaymentMailedCheck	∞

8 Imbalance Problem

The dataset has a 3:1 ratio of `No` to `Yes` for `Churn`, this imbalance may cause the classifier to tend to predict the `No` result. To handle the class imbalance issue, our research processed the data by random undersampling and oversampling. However, both treatments performed poorly compared to the original dataset result. An example for XGBoost shows this issue (Figure 2).

This observation could be attributed to two reasons. Firstly, the 3:1 ratio of `No` to `Yes` in the original data is not a severe imbalance problem. In this paper[3] by He and other researchers, it states that imbalance below 1:10 is of little significance to deal with. Secondly, the

[3]He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.

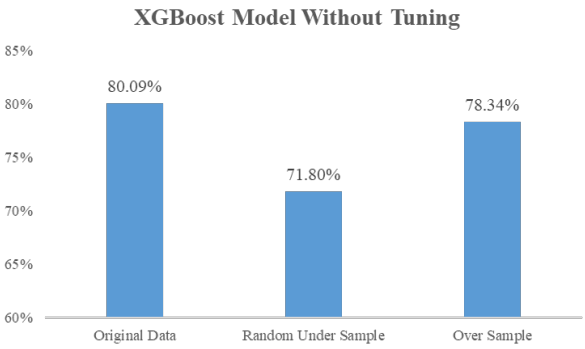

Figure 2: Comparison of XGBoost results fitted on original and sampled datasets

unbalanced distribution may reflect a 3:1 distribution in the real world. Imbalance processing will cause the model to lose such information, hence explaining their worse performance. Moving forward, our project thus decided to use the original data to fit our models.

9 Feature Selection

Boruta, a robust feature selection method, was selected to identify relevant features in the dataset. This method employs a random forest model to assess the importance of each feature, including any irrelevant features. Our study utilized cross-validation Boruta (Figure 3) to identify 11 relevant features (Figure 4), which were subsequently used to evaluate the effectiveness of feature selection in our models.

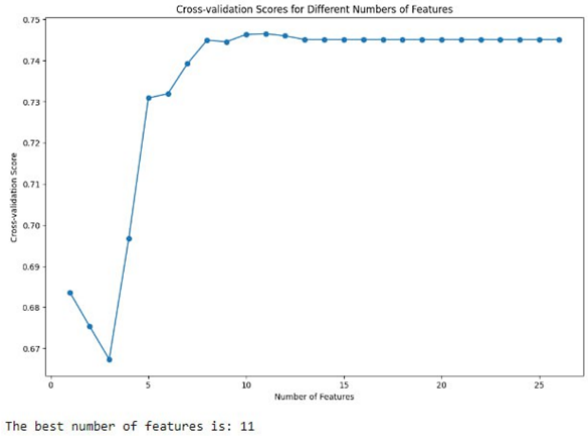

Figure 3: Boruta CV-score

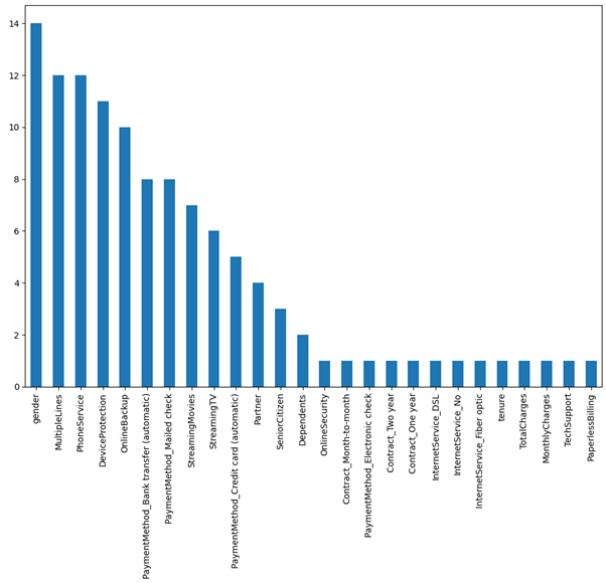

Figure 4: Boruta selected features

10 Model Training, Prediction, and Evaluation

Since this is a binary classification problem, we chose to fit these models and compare their performance:

- KNN, LDA, QDA, Logistic Regression
- Tree-based methods: Decision Tree, Bagging, Random Forest
- Boosting: AdaBoost, XGBoost
- ANN
- SVM: Linear, Polynomial, Radial Basis Function

Before fitting the models, the dataset was split into training set and testing set. The training set was at 70% (left with around 5,000 samples) and the test set at 30%. For each of the models, we compared fitting with the original scaled dataset and the Boruta dataset.

From Figure 5, it was detected that XGBoost performs the best and QDA the worst. In this report, we selected distinct methods with results that were worthy to explore, to explain in more detail: Logistic Regression, SVM, ANN and XGBoost.

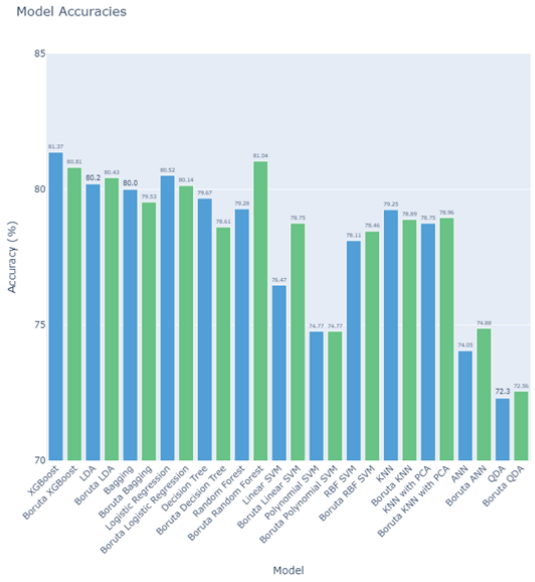

Figure 5: Test accuracy of all models

10.1 Logistic Regression

Logistic regression is used commonly for classification problems as it is easy to implement, interpret and efficient to train. In the initial fitting with the scaled dataset, we attained a high accuracy of 80.85% (Figure 6). However, we see that there are insignificant variables with high p-value (Figure 7). And some of the variables with high p-value like `PhoneService` were previously highlighted to have high VIF values (Table 1).

We hence try to remove the insignificant variables and variables with high VIF. We then get another fit with predictors with small p-value near to 0 (Figure 8). The coefficients are all also relatively high, showing that the variables are useful in predicting churn. The accuracy for this model is 80.52% (Figure 9). As there are less fitted features in this model, the accuracy has dropped.

We then compare a model fitted with the predictors identified in the feature selection stage. We observe that this performs worse at 80.14% (Figure 10), and the

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	-0.1225	0.1680	-0.7287	0.4662	-0.4518	0.2069
gender	0.0225	0.0776	0.2900	0.7718	-0.1296	0.1746
SeniorCitizen	0.2769	0.1021	2.7126	0.0067	0.0768	0.4770
Partner	-0.0212	0.0935	-0.2271	0.8203	-0.2046	0.1621
Dependents	-0.1135	0.1082	-1.0485	0.2944	-0.3255	0.0986
PhoneService	0.7239	0.7829	0.9246	0.3552	-0.8106	2.2584
MultipleLines	0.5206	0.2146	2.4262	0.0153	0.1000	0.9411
OnlineSecurity	-0.0753	0.2157	-0.3491	0.7270	-0.4981	0.3475
OnlineBackup	0.1988	0.2127	0.9349	0.3499	-0.2180	0.6157
DeviceProtection	0.2476	0.2115	1.1708	0.2417	-0.1669	0.6620
TechSupport	-0.0860	0.2158	-0.3984	0.6903	-0.5089	0.3369
StreamingTV	0.9323	0.3958	2.3556	0.0185	0.1566	1.7080
StreamingMovies	0.7770	0.3943	1.9706	0.0488	0.0042	1.5498
PaperlessBilling	0.2710	0.0889	3.0489	0.0023	0.0968	0.4452
InternetService_DSL	-0.0547	0.0878	-0.6235	0.5329	-0.2268	0.1173
InternetService_Fiber_optic	2.3949	1.0074	2.3774	0.0174	0.4205	4.3693
InternetService_No	-2.4626	0.9233	-2.6672	0.0076	-4.2723	-0.6530
Contract_Month	0.7104	0.0949	7.4867	0.0000	0.5245	0.8964
Contract_One_year	-0.0490	0.1107	-0.4425	0.6581	-0.2661	0.1680
Contract_Two_year	-0.7839	0.1590	-4.9293	0.0000	-1.0956	-0.4722
PaymentMethod_Bank_transfer	-0.0707	0.0924	-0.7654	0.4440	-0.2518	0.1104
PaymentMethod_Credit_card	-0.1400	0.0965	-1.4513	0.1467	-0.3291	0.0491
PaymentMethod_Electronic_check	0.2203	0.0743	2.9633	0.0030	0.0746	0.3660
PaymentMethod_Mailed_check	-0.1320	0.0907	-1.4562	0.1453	-0.3098	0.0457
tenure	-3.9702	0.5276	-7.5253	0.0000	-5.0042	-2.9362
TotalCharges	2.3961	0.7328	3.2696	0.0011	0.9598	3.8324
MonthlyCharges	-6.4828	3.8442	-1.6864	0.0917	-14.0174	1.0517

Figure 6: Logistic Regression with all predictors

		Predicted	
		Not churn	Churn
Actual	Not Churn	1417	241
	Churn	163	289

Figure 7: Confusion Matrix for Figure 2

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	-1.5955	0.1151	-13.8669	0.0000	-1.8210	-1.3700
SeniorCitizen	0.2936	0.0998	2.9412	0.0033	0.0980	0.4893
MultipleLines	0.3313	0.1050	3.1543	0.0016	0.1254	0.5371
OnlineSecurity	-0.2657	0.1068	-2.4881	0.0128	-0.4750	-0.0564
TechSupport	-0.2668	0.1090	-2.4473	0.0144	-0.4804	-0.0531
StreamingTV	0.5642	0.1167	4.8363	0.0000	0.3356	0.7929
StreamingMovies	0.4156	0.1150	3.6154	0.0003	0.1903	0.6409
PaperlessBilling	0.2773	0.0887	3.1257	0.0018	0.1034	0.4512
InternetService_DSL	-0.5359	0.0891	-6.0118	0.0000	-0.7106	-0.3612
InternetService_Fiber_optic	0.9812	0.1746	5.6202	0.0000	0.6390	1.3234
InternetService_No	-2.0408	0.2159	-9.4544	0.0000	-2.4638	-1.6177
Contract_Month	0.2214	0.0901	2.4579	0.0140	0.0449	0.3980
Contract_One_year	-0.5403	0.0998	-5.4155	0.0000	-0.7358	-0.3447
Contract_Two_year	-1.2766	0.1527	-8.3591	0.0000	-1.5760	-0.9773
PaymentMethod_Bank_transfer	-0.2924	0.1115	-2.6225	0.0087	-0.5109	-0.0739
PaymentMethod_Credit_card	-0.3582	0.1176	-3.0466	0.0023	-0.5886	-0.1277
PaymentMethod_Mailed_check	-0.3570	0.1168	-3.0568	0.0022	-0.5859	-0.1281
tenure	-1.3875	0.1808	-7.6749	0.0000	-1.7419	-1.0332
TotalCharges	0.6513	0.1912	3.4065	0.0007	0.2766	1.0260
MonthlyCharges	-0.8344	0.2060	-4.0512	0.0001	-1.2380	-0.4307

Figure 8: Logistic Regression with significant predictors

		Predicted	
		Not churn	Churn
Actual	Not Churn	1411	242
	Churn	169	288

Figure 9: Confusion Matrix for Figure 4

variables have high p-values (Figure 11).

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	-2.1643	0.1614	-13.4104	0.0000	-2.4806	-1.8480
tenure	-1.3657	0.1804	-7.5717	0.0000	-1.7192	-1.0122
OnlineSecurity	-0.5031	0.1007	-4.9952	0.0000	-0.7005	-0.3057
TechSupport	-0.4823	0.1042	-4.6303	0.0000	-0.6864	-0.2781
PaperlessBilling	0.3554	0.0872	4.0779	0.0000	0.1846	0.5263
MonthlyCharges	0.1362	0.1315	1.0353	0.3005	-0.1216	0.3940
TotalCharges	0.6779	0.1913	3.5436	0.0004	0.3029	1.0528
InternetService_DSL	-0.6005	0.0889	-6.7554	0.0000	-0.7747	-0.4262
InternetService_Fiber_optic	0.0846	0.1454	0.5818	0.5607	-0.2004	0.3695
InternetService_No	-1.6484	0.1682	-9.8028	0.0000	-1.9780	-1.3188
Contract_Month	1.6036	0.2141	7.4880	0.0000	1.1838	2.0233
Contract_One_year	0.7666	0.2177	3.5219	0.0004	0.3400	1.1931

Figure 10: Logistic Regression with feature selected predictors

		Predicted	
		Not churn	Churn
Actual	Not Churn	1401	240
	Churn	179	290

Figure 11: Confusion Matrix for Figure 6

For logistic regression, we hence decide that the model in Figure 8 performs the best. Upon further exploration of this model (Figure 12), we observe that fibre optics has the greatest positive coefficient, meaning that customers are more likely to churn if fibre optics are offered. This is contradictory to our understanding as fibre optics provide the fastest internet connections so we thought that customers would be more likely to stay because of it.

We also noticed that payment methods matter less to customers when they decide to leave a company. This makes sense as customers would be more concerned about the qualities of the telecommunication product or service instead of the payment method.

It is also interesting to note that `OnlineSecurity` has the smallest coefficient, as it seems to be a vital part of internet usage. This could be because the data was from 2019, and people were not as concerned about their personal data as compared to other factors. Or the

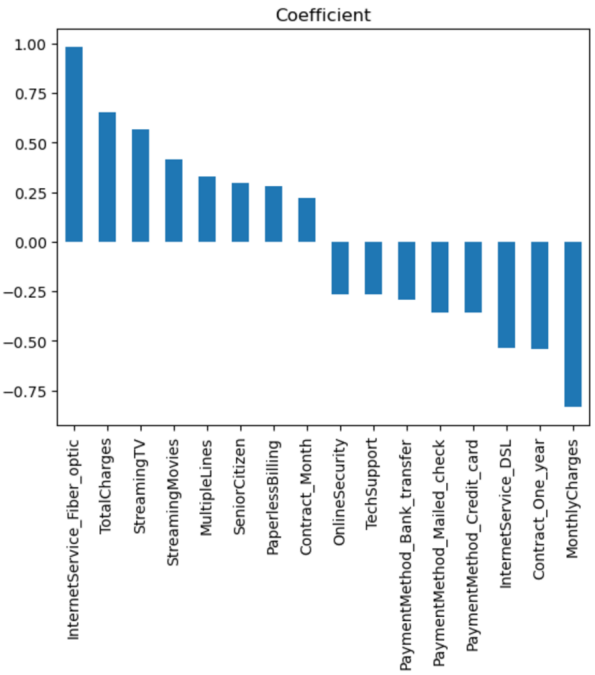

Figure 12: Fitted coordinates for Figure 4

customers were using services from other companies to protect their data.

10.2 Support Vector Machine

Support Vector Machines (SVMs) are a powerful class of supervised learning algorithms used for classification. SVMs work by locating the hyperplane that best separates different classes of data in a high-dimensional space. The SVM algorithm can use several kernel functions to transform the data into a higher-dimensional space, making it easier to locate a hyperplane that can accurately separate the different classes.

Linear SVM	Poly SVM	RBF SVM
Confusion Matrix: $\begin{bmatrix} 1517 & 63 \\ 370 & 160 \end{bmatrix}$ Accuracy: 0.794	Confusion Matrix: $\begin{bmatrix} 1580 & 0 \\ 530 & 0 \end{bmatrix}$ Accuracy: 0.748	Confusion Matrix: $\begin{bmatrix} 1537 & 43 \\ 417 & 113 \end{bmatrix}$ Accuracy: 0.781

Figure 13: Confusion matrix and accuracy for SVM

In our experimentation with SVMs, we tested various kernel functions and found that the RBF kernel achieved the highest accuracy, while the polynomial kernel did not perform as well (Figure 13). Though the accuracy of Linear SVM seems high here, the performance of it is extremely unstable as it varies between 50% and 80%. Besides, the confusion matrix shows that the Polynomial SVM classified all values as positive, which is an unacceptable outcome. It shows that SVM with a polynomial kernel cannot capture the pattern of the data and is not suitable here. These phenomena may be attributed to the curse of dimensionality. The exponential growth in the number of possible feature combinations with the increase in dimensions can make it challenging for the SVM to effectively capture the underlying patterns in the data. This is also consistent with a paper by Kotsiantis and other researchers[4], who reported that the curse of dimensionality has a negative impact on SVM classification accuracy.

Besides, from the importance of the features plots we can see that the importance weights of these features are distributed more evenly in RBF SVM (Figure 16) compared to Polynomial SVM (Figure 15), which means that RBF SVM is better able to get complex relationships between input features and output. This is likely the reason why RBF SVM outperforms all other SVMs. We can also observe that the total charge and the tenure, which were given little weight in Linear SVM (Figure 14), are actually very important features in Poly SVM and RBF SVM. Because these features and output do not have a linear relationship, Linear SVM cannot use them and therefore its performance is not stable.

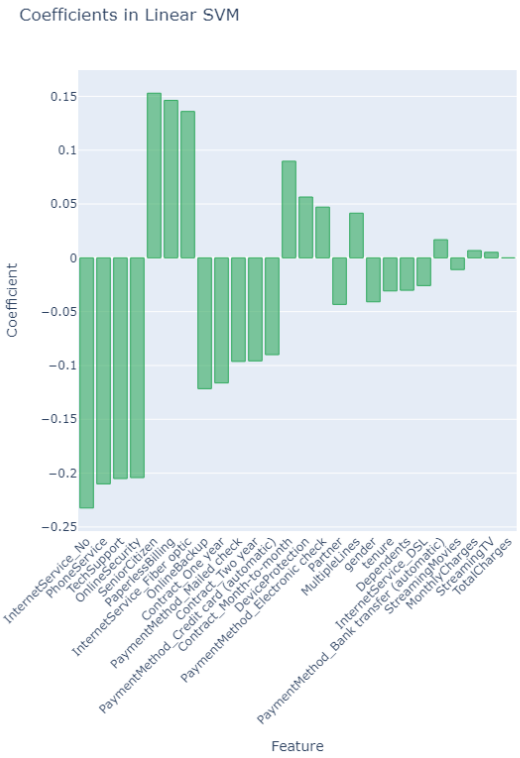

Figure 14: Linear SVM

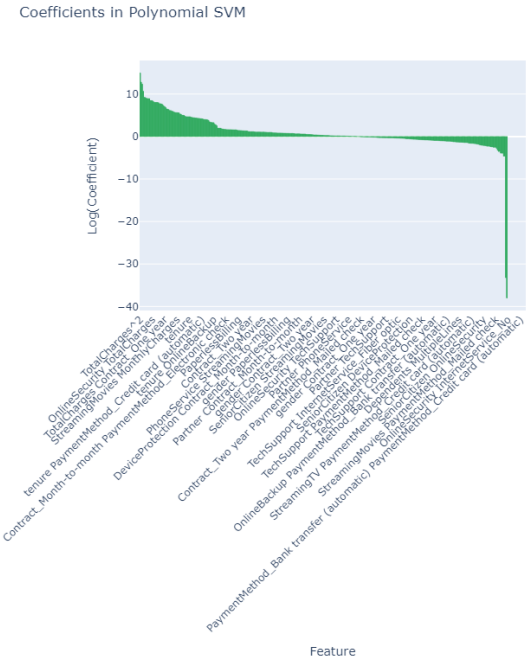

Figure 15: Polynomial SVM

[4]Kotsiantis, S. B. (2006). Handling imbalanced datasets: A review. GESTS International Transactions on Computer Science and Engineering, 30(1), 25-36.

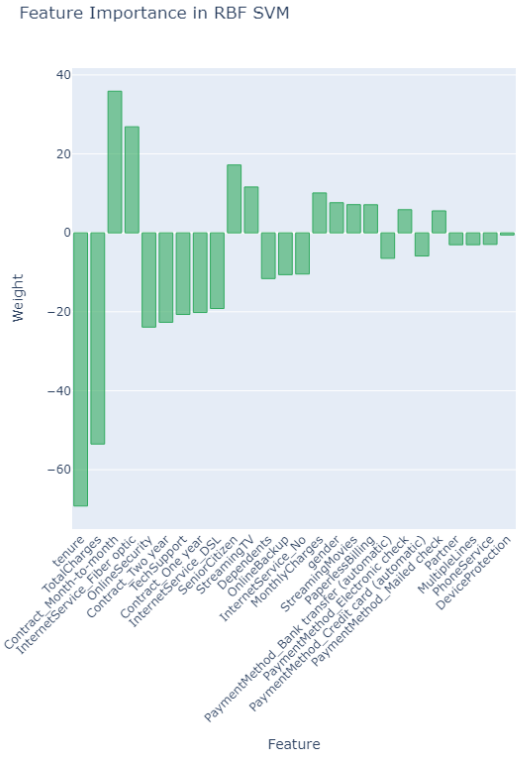

Figure 16: RBF SVM

10.3 Artificial Neural Network (ANN)

In the deep learning section, we choose to use the Artificial Neural Network (ANN) model for training and predicting the results because the data is unordered and non-image-based.

The ANN model introduces data through the input layer, adds complex nonlinear relationships, and establishes associations in the hidden layer. Multiple hidden layers work together to produce very convoluted parameters for training, and then the activation results are transformed to obtain the final classification results.

Given the training dataset of about 5000, our research finally considers two hidden layers and three hidden layer neural network models (Figure 17, which are determined by the amount of data and the difficulty of the problem. For example, a model with 2-hidden layers can end up with hundred parameters after hidden layers and the number of nodes enhances the parameter dimensionality. And a 3-hidden layer model can apply a more significant number of nodes and end up with thousands of parameters; such parameter numbers are the most acceptable for our training set. If deeper and more complex hidden layers are considered, the training data will not be able to support the computation of the model.

During the model's training, parameters considered to be tuned include the number of hidden layers, the neuron nodes per layer, and the number of epochs. And our research also considers the data after different scaling approaches, since a paper by Bengio and other researchers [5] stated that in deep learning unscaled data can cause different feature scales that affect the convergence and prediction accuracy of the model.

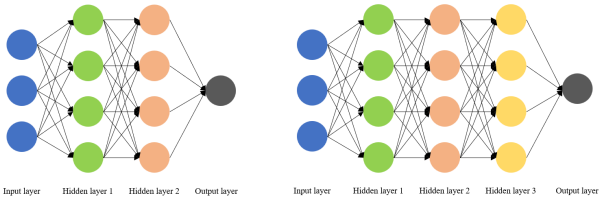

Figure 17: Selected two Neural Network Structures

Our research investigates the performance of 2- and 3-layer ANN models in predicting outcomes. The prediction accuracy of both models is analyzed, and the results are evaluated post-tuning. Notably, our findings align with previous assumptions, indicating that scaling data can enhance prediction accuracy. Specifically, the results reveal that min-max and quantile scaling techniques yielded superior outcomes, suggesting that the data under examination possessed non-bell shape

[5] Bengio, Y., Goodfellow, I., & Courville, A. (2016). Deep learning. MIT Press.

characteristics.

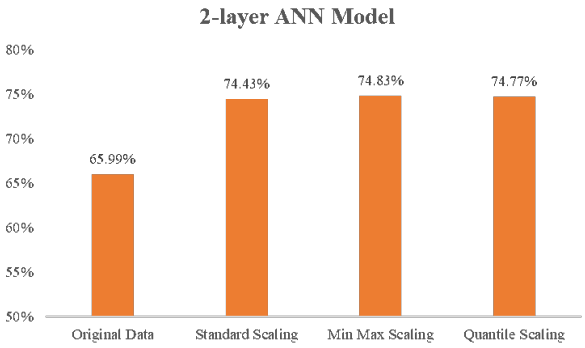

Figure 18: 2 layer ANN model performance after tuning

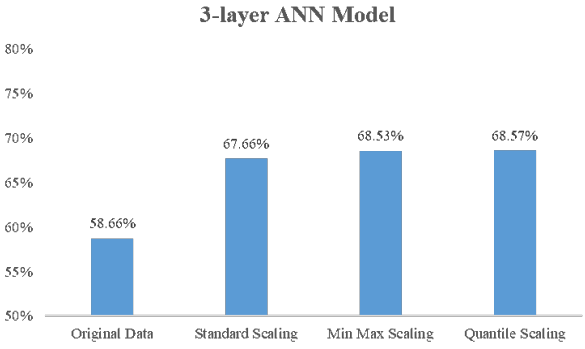

Figure 19: 3 layer ANN model performance after tuning

Comparing the results of the two models, it is observed that the 2-layer ANN model (Figure 18) outperformed the 3-layer model (Figure 19). This observation may be attributed to the relatively minor amount of data and simpler structure of the 2-layer model, which is less prone to over-fitting or gradient vanishing.

In addition to analyzing the performance of ANN models based on the number of layers, our research also conducts further analyses on the impact of feature selection on prediction accuracy. As in Figure 20, Our findings indicate that fitting the model to selected features will lead to a slight improvement in prediction accuracy following scaling. Interestingly, reducing the number of nodes and epochs did not significantly alter the accuracy, suggesting that a smaller, more focused set of features may produce more favorable outcomes. This may be attributed to the smaller amount of data used in the analysis, as fewer features are less likely to produce overfitting, and the selected features by Boruta method may represent more critical information for prediction accuracy. However, it is noteworthy that the complexity of the data, including periodic data and excessive noise, may limit the effectiveness of the ANN model in comparison to simpler models. As the best accuracy is only 74.88%, we won't consider it for prediction purposes.

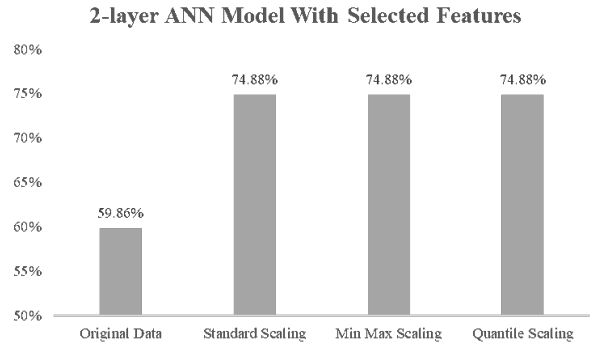

Figure 20: 2 layer ANN with selected features performance after tuning

10.4 XGBoost

For Ensemble learning methods, we considered bagging and boosting methods and finally found that the XGBoost model could achieve the best results after parameter tuning and it beats all other models in terms of accuracy.

XGBoost (eXtreme Gradient Boosting) is a powerful model similar to AdaBoost, which also improves its prediction accuracy by integrating weak decision tree classifiers. The difference is that XGBoost uses second-order gradient information as the loss function, while Adaboost uses exponential gradient information. Regarding the classification problem, it used logloss

as the loss function to train the model. And XGBoost will perform regularization during the training process, which is similar to the algorithm of tree pruning to remove nodes if they contribute no value to the prediction, hence can avoid the huge influence of outliers, which makes its prediction more accurate. In this study, parameters to be tuned include max_depth, learning_rate, n_estimators, reg_alpha, and reg_lambda, with the last 2 corresponding to the regularization term.

After parameter tuning, the prediction accuracy of the XGBoost model trained from the original data can reach 81.37% (Figure 21), beating all other models. From the feature importance graph (Figure 22), it can be observed that the most important features are `TotalCharges`, `MonthlyCharges`, and `Tenure`, indicating they can help reduce much of the node impurity (relative to the number of times the feature is used to split the data across all trees in the model). The SHAP diagram implements game theory to calculate the Shapley values, which measure the contribution strength and direction of different variables to the final prediction result, and hence help to interpret the perplexing XGBoost model. Taking the first person as an example, in Figure 23 it can be observed that people with `Month-to-Month` contracts contribute +0.66 to its final prediction, which is the strongest positive impact on the prediction. This makes sense as a shorter contract means more ease of movement to another company and hence a customer leaving is more likely. To understand the difference between the importance and SHAP graph, the `MonthlyCharges` with high feature importance appeared in the importance graph (Figure 22) and not in the SHAP graph (Figure 23), indicating that it may be of great importance to model training, but has little effect on the prediction of individual instances. This also gives us the motivation to focus on interactions between features.

		Predicted	
		Not churn	Churn
Actual	Not Churn	1431	149
	Churn	244	286

Figure 21: Confusion Matrix for XGBoost fitted on all predictors

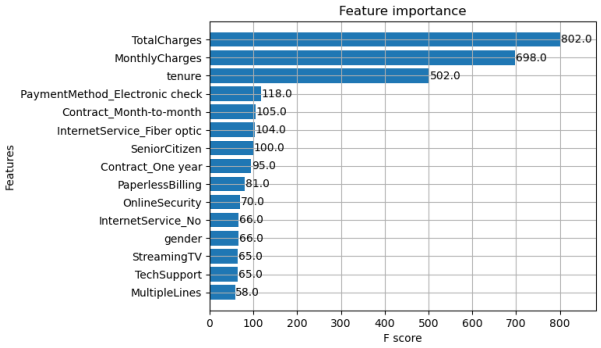

Figure 22: XGBoost Feature Importance Graph (trained with all features)

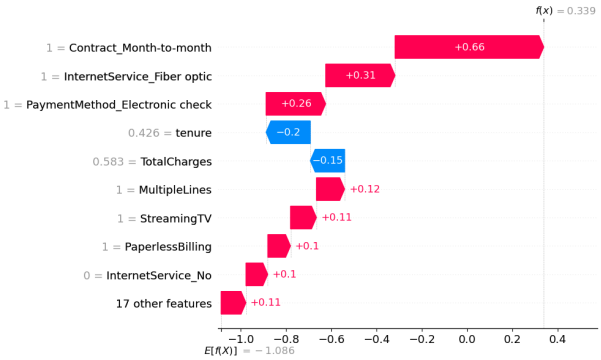

Figure 23: XGBoost SHAP Diagram for the first people to be predicted as an example (trained with all features)

XGBoost has fitted again with the Boruta selected features, and the accuracy is slightly lower but still reaches 81.33% (Figure 24). There has been a large change in the SHAP diagram (Figure 26) regarding features and the diagram shape, which makes us sure that some unnecessary variables have indeed been removed.

		Predicted	
		Not churn	Churn
Actual	Not Churn	1426	154
	Churn	240	290

Figure 24: Confusion Matrix for XGBoost fitted on selected Boruta features

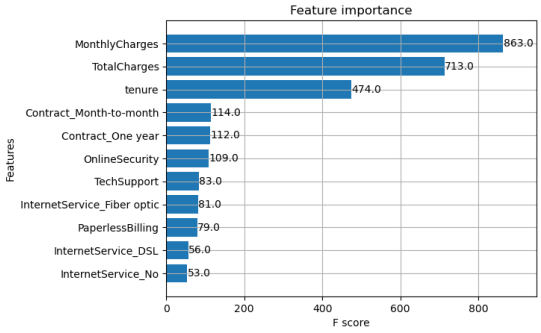

Figure 25: XGBoost Feature Importance Graph (trained with Boruta features)

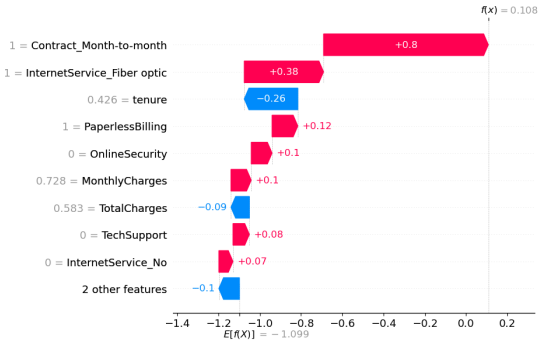

Figure 26: XGBoost SHAP Diagram for the first people to be predicted as an example (trained with Boruta features)

However, after detecting the tree diagram obtained from two training sessions, it is found that the model tree after selecting features (Figure 28) is more complex and deeper than the tree fitted on all predictors (Figure 27), which means that there might be an overfitting problem. Also, by reviewing the similar two confusion matrix (Figure 21, 24), there is no pattern showing that any of them may fall into the trouble of strong bias. Therefore, our research decides to select the XGBoost model trained with all features to give future predictions.

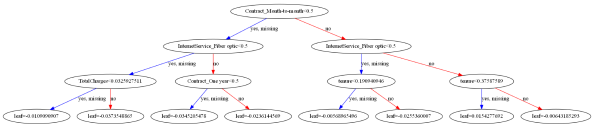

Figure 27: XGBoost Tree plot (trained with All features)

Figure 28: XGBoost Tree plot (trained with Boruta features)

11 Conclusion

In conclusion, our study determined that the XGBoost model, trained on our original scaled dataset, exhibited superior performance in predicting churn among telecommunication consumers. This was attributed to the model's exceptional accuracy and the provision of beneficial regularization properties.

We opted not to utilize the features selected by Boruta, despite its effectiveness in reducing training time, due to its inability to identify useful features that would improve the predictive model. We postulate that the poor performance of feature selection could be attributed to the large number of predictors in this dataset. This observation is consistent with existing literature[6] that highlights that in datasets with many features, some features may have low discriminatory power making it difficult to identify the most relevant features.

[6]Zheng, Z., Wu, X., & Srihari, R. K. (2007). Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the 20th International Conference on Machine Learning (ICML-07) (pp. 856-863).