UNIVERSITY OF MALAYA

Master of Data Science

Faculty of Computer Science & Information Technology

WQD7006 Machine Learning for Data Science

Group Assignment

Title: Telco Customer Churn Prediction

Prepared by

Name	Matric Number
Lee Ming Xiang	S2021030
Valli Suppramaniam	17053409
Wan Amira Balqis	S2003952
Yong Chin Xian	17151186
Lim Han Shyong	17164409

Prepared for

Name	Dr Vimala Balakrishnan

Table of Content

1. Introduction / Background of Study

1.1 Problem Statement

Customer retention is a critical issue in any industry, particularly for companies with a subscription-based business model. Due to intense competition, telecommunications companies need to identify customers who might switch to a competitor. An accurate predictive customer churn model is therefore necessary for the firms' customer relationship management.

1.2 Churn Prediction

Churn prediction is a proactive measure for customer relationship management. Leveraging on machine learning to predict customer churning can provide insights to companies on measures required to improve their customer retention strategy. After collecting the dataset and doing the necessary data-preprocessing activities, the next step is to find the appropriate hyperparameter tuning methods, training the machines and selecting the best performers. For this project, we are looking into telecommunication companies as they have a subscription-based business model. This will allow us to make prediction models based on customer churn data. (Altexsoft, 2019)

1.3 Project Objectives

The following objectives contribute to our research:

- To identify significant predictors for telco customer churn prediction
- To evaluate the accuracy of machine learning algorithms used for telco customer churn prediction
- To achieve an optimal algorithm for telco customer churn prediction

2. Literature Review

Table 1 Literature Review Analysis

No	Research	Problem Statement	Suggested Solution(s)	Significance (e.g. evaluated as performance)
[1]	Ahmad et al. (2019)	1. Customer churn is a major concern which give direct effect on companies' revenues, especially in telecom field.	1. Use machine learning models on big data platforms to look for the major contribution on customer churning.	1.Four algorithms have been experimented: Decision Tree, Random Forest, Gradient Boost, and XGBoost. The best result is using XGBoost that achieved AUC 93.3%.
[2]	Buda et al. (2018)	1. To investigate impact of class imbalance on classification performance of convolutional neural networks.	1. Comparing different methods for addressing the imbalanced issue.	1. The effect of imbalanced datasets on classification is detrimental. 2. Oversampling is the most suitable approach to handle the imbalance dataset in convolutional neural networks.

[3]	Ismail et al. (2015)	1. To identify customers that have the potential to churn at an early stage, based on customer data, so customer retention program can be applied.	1. Proposed Multilayer perceptron neural network approach to churn prediction. 2. Compare the proposed algorithm performance with other popular algorithms.	1. Performance churn predictions among multilayer perceptrons, multiple regression analysis and logistic regression. 2. The modelling features are based on several groups of data, such as customer demographic and relationship data, billing and also usage data. 3. Using features like no of call, billing and payment data, plus customer profile such as age and gender, together with monthly commitment and internet speed to modeling. 4. The result shows that multi layer perceptrons perform significantly compared with others. Multilayer perceptron achieves very good results in accuracy(91.28%), sensitivity (93.59%) and specificity (88.28%).
[4]	Ullah et al. (2019)	1. Existing models did not provide clear insights on the behaviour of churning customers, preventing the adoption of retention strategies.	1. Proposed segmentation of churning customers using cosine similarity to form effective retention measures, following supervised classification and prediction of churning customers. 2. Identify important features in recognising churning behaviours.	1. Random Forest reported the highest accuracy score in the supervised classification and prediction part at 88.63%. 2. k-Means Clustering was used for customer profiling into three groups of customers, namely low-risk, medium-risk and high-risk customers.
[5]	Al- Shatnwai, Faris M. (2020)	1. To analyse the improvement in prediction accuracy due to the application of XGBoost together with different popular oversampling methods to an imbalance dataset.	1. Four oversampling methods i.e. Random oversampler, SMOTE, ADASYN and Borderline SMOTE are used. The performance of XGBoost and other algorithms are tested before and after the application of the oversampling methods. Conclusion is drawn based on the experiment conducted.	1. Based on the experiment conducted, the XGBoost with oversampling method showed the highest prediction accuracy of 95.6%, followed by Random Forest (95.5%), Logistic Regression (86.4%), SVM (82.7%) and SGD (80.1%). As such, the paper proposed the XGBoost with oversampling method as the preferred algorithm for the churn prediction for telecommunication firms.

[6]	Li and Deng (2012)	1. Identify how to retain existing Telco customers for China Telecom by using prediction models	1. Two customer churn prediction algorithms were analysed and compared which are cluster analysis and decision tree algorithm using SAS software.	1. K-Means clustering was used and segregated into 4 clusters derived from 20 attributes of the lost customers such as the brand, age, sex, on-network duration, credit rating, fee-collecting mode, network access channel, VIP class, average total talk times, average external network contacts, and average GPRS fee. 2. Two decision tree algorithms were created with the nodes setting of 3 (Tree 1) and 5 (Tree 2). Overall prediction accuracy ratio shows that Tree 1 is superior to the Tree 2 model.

3. Methodology

3.1 Data Description

The dataset is acquired from Kaggle, it consists of both training and test data. The training data contains 4250 instances with 20 features. The test dataset contains 750 instances with 20 features. The training dataset is unbalanced since there are 3652 samples (85.93%) belong to class churn=no, and 598 samples (14.07%) belong to class churn= yes. Both training and test data have no missing or null values. Among 20 features, there were only 4 columns with category class, and 16 columns with numerical data types.

3.2 Exploratory Data Analysis (EDA)

In this study, we are using both Weka and Python for the project. Data was visualized in kernel density distribution plot to understand the feature distribution, and relationship between the features. The bar plots show that the categorical features can distinguish whether the customer is churning or not. Thus, we are including these features, and apply Label Encoder to convert them to numerical data types. From the distribution plots, we observed that the total charge and total minutes have a similar pattern and correlation with the target label, churn. This observation is further supported by correlation and cluster analysis.

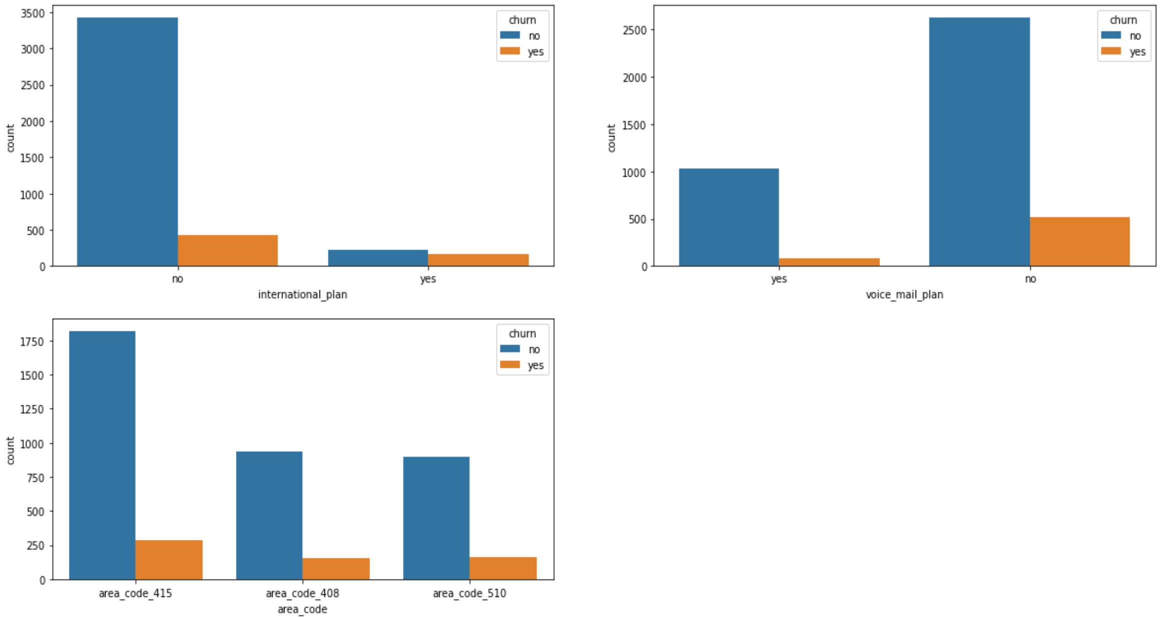
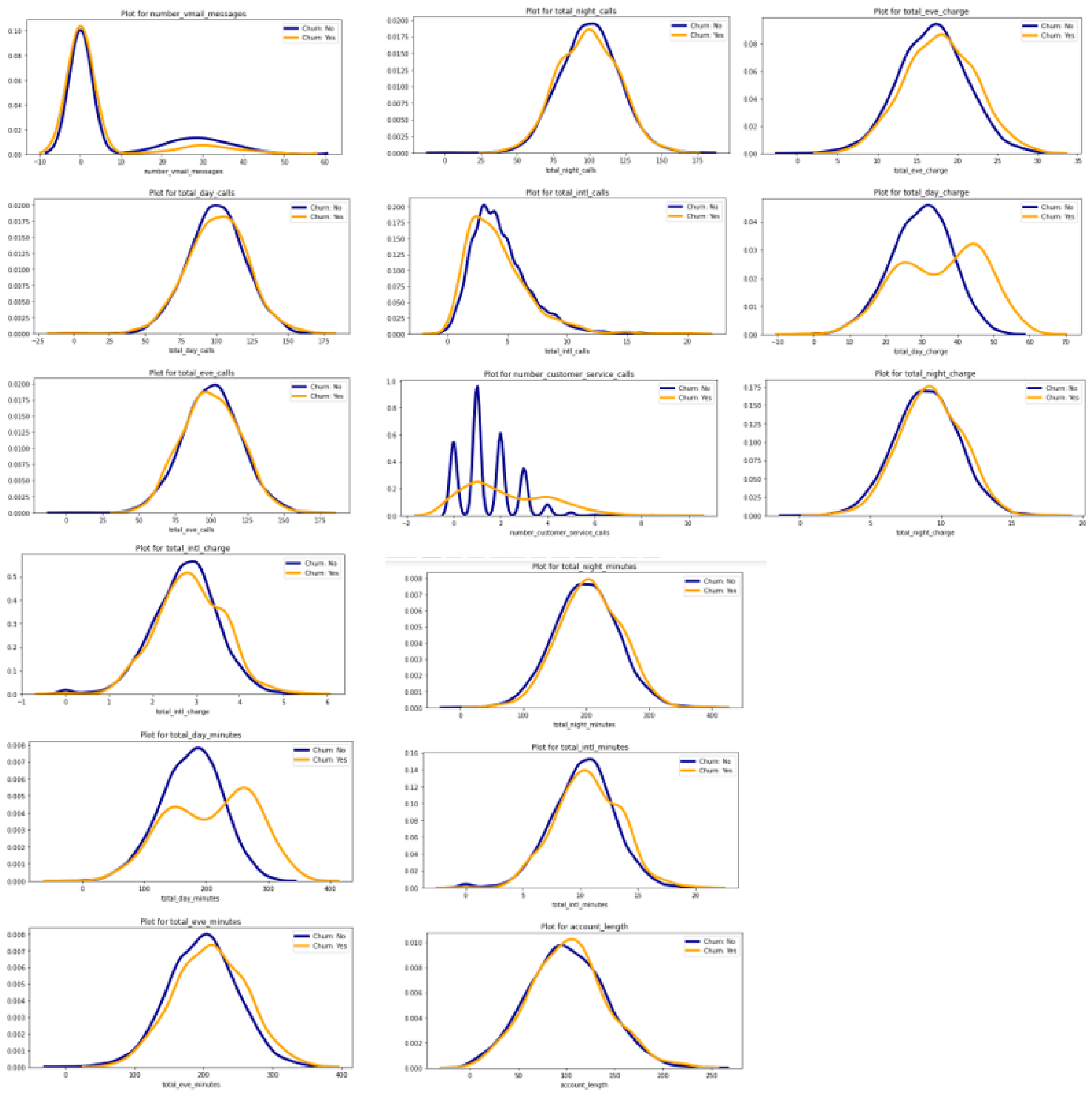


Figure 1 Bar plot showing the relationship between the categorical attributes with the target label



:

Figure 2 Kernel density estimation (KDE) plots showing the features distribution corresponding to the target label

3.2.2 Correlation Analysis

Pearson Correlation analysis is conducted to analyse multicollinearity issues among the attributes. The correlation among the attributes are noted to be insignificant (i.e. correlation coefficient value of <= +/- 2%) except for the following attributes:

Table 2 Correlation table by using Pearson Correlation Analysis

Attribute1	Attribute 2	Correlation Value	Action to be taken
total_day_charge	total_day_minutes	100%	Drop attribute: total_day_charge
total_eve_charge	total_eve_minutes	100%	Drop attribute: total_eve_charge
total_night_charge	total_night_minutes	100%	Drop attribute: total_night_charge
total_intl_charge	total_intl_minutes	100%	Drop attribute: total_intl_charge

As such, the following variables are dropped in the subsequent steps: total_day_charge, total_eve_charge, total_night_charge, total_intl_charge

3.2.2 Cluster Analysis Based on K-Means

In building a cluster model, we dropped the 4 variables that are highly correlated such as total_day_charge, total_eve_charge, total_night_charge, total_intl_charge to further analyse the features relationship. 2 clusters were used in this case and validated by using elbow technique after 10 iterations. From the analysis, we can conclude that area code, voicemail plan, international plan, number of voicemail messages, and area code are the main features to build the cluster for customer churn.

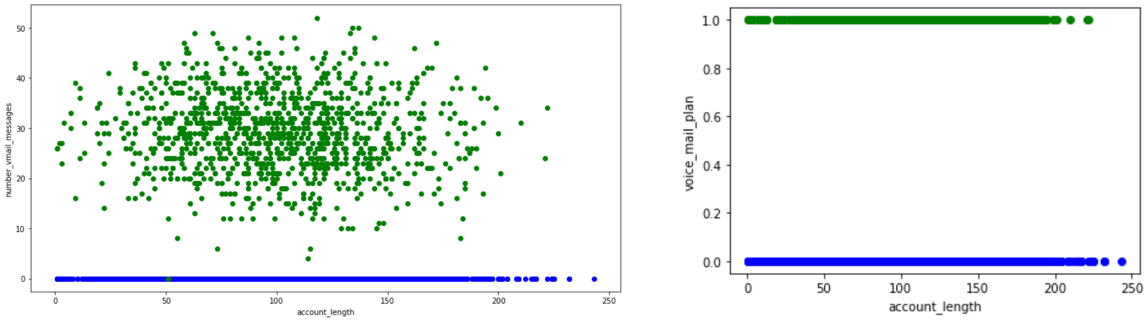


Figure 3 Cluster Analysis Result

3.3 Data Processing

Based on the EDA analysis, we validated the importance of the categorical features for our modelling algorithms. Thus, we are using Label Encoder to convert the categorical column with more than 2 attributes, such as state and area code. For features such as ‘internation_plan’, ‘voice_mail_plan’ , and target label ‘churn’, we are mapping ‘yes’ to 1 , and ‘no’ to 0.

3.4 Machine Learning Algorithms

In this study, the following algorithms were used to test the models. Selection for the best model is based on measures such as “Correctly classified Instances”, “Precision”, “Recall”, “F-Measure” and “Area Under the Receiver Operating Characteristic Curve (ROC AUC)”.

3.4.1 Random Forest

Random forest is a “bagging” learning method that can be used for classification and regression by utilising an ensemble of decision trees. The classifier refers to the prediction of each tree and uses the majority voting system to predict the final outcome, instead of depending on a single decision tree.

3.4.2 Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm that can be used for both classification and regression problems. In SVM, the classification is performed by finding the hyper-plane that separates the cases into two classes. Separating plane is usually determined by only a handful of data points. The points that help determine the hyperplane are called Support Vectors and the hyperplane itself is a classifying machine. In SVM, although a linear hype-plane is easy to have for two classifiers, the kernel trick enables the handling of non-linear classifiers.

3.4.3 Multilayer Perceptron

A deep learning class of feedforward artificial neural network, it's a most basic neural network among others. Generally deep learning is a subset of machine learning, but it's getting famous as it only needs minimal guidance compared with traditional machine learning. Thus, we used one of the modals here for churn predictions, to compare the performance with others.

Due to an imbalanced dataset, we applied stratified cross validation during model evaluations, this is to ensure the testing data matched the real situation of the dataset. We configured the model with 2 hidden layer 10 nodes, as it is half of the feature’s sizes, and 2 layers giving a better performance score.

3.4.4 Xgboost

XGBoosting stands for extreme gradient boosting, which is a decision tree based ensemble technique (Morde,V., 2019). The package includes efficient linear model solver and tree learning algorithm. It is built sequentially by minimizing the errors from previous models, and using weak learning to improve the model prediction. In hyperparameter tuning, we have tested with variable learning rate, maximum depth of tree, the regular lambda, gamma, and regular alpha, and the colsample by tree and node, to further enhance the model computation accuracy and efficiency.

3.4.5 Decision Tree (J48)

A decision tree is one of the most common supervised models in machine learning which helps with decision making (Gupta, 2017). The data is continuously split into multiple outputs according to the parameters in the dataset and provides a tree-like structure. We utilized J48 as the decision tree algorithm. J48 is known to deal with specific characteristics, varying attribute costs and missing attribute estimations of the data (Saravanan and Gayathri, 2018). Venkatesan (2015) states that pruning will improve the accuracy of the model. Hence, we tested the algorithm pruned and unpruned, removing correlated attributes to find the difference in model accuracy.

3.5 Cross Validation & Random Sampling

3.5.1 Data Imbalance Issue & Random Sampling

The label attribute is ‘churn’ and it consists of 3652 instances of ‘yes’ and 598 instances of ‘no’ in the full dataset. Based on this count and other preliminary model runs, e.g. SVM, showed that the model is unable to classify any cases under ‘yes’ as the dataset is biased towards the ‘no’ category. As such, the data imbalance issue needs to be corrected to address the bias and predictability issue. This is addressed by using a new dataset based on random sampling of 551 instances of “No” and all 598 instances of “Yes” from the label attribute “churn”. As this study requires a dataset that contains more than 1000 instances, this sampling method is deemed suitable as it considered all the 598 instances of “yes” and about an equal proportion from the “no” category to correct the imbalance issue. To ensure, there is no bias in the new dataset based on random sampling, another set of random samples are taken and the results from the two datasets are compared. As the results from both datasets are almost similar, the method is concluded as suitable and only the first random sample dataset was used in this analysis and the results are compared against the full dataset.

3.5.2 Cross Validation

All the models in this study are evaluated using the 10-fold cross validation method as well as split-validation of 70% (train) & 30% (test). The final reported numbers are based on cross validation as this method is deemed as more robust as compared to split validation.

4. Results & Discussion

This section we discuss the model performance, we group the experimental results into 3 different categories, (1) Base model; (2) Model after hyperparameter and dropping correlated features; (3) Model with random sampling to address unbalanced data. The table below shows all the evaluation metrics for each model. We are showing the accuracy, precision, recall, F-score and ROC curve (receiver operating characteristic curve) in the table as each of them have different meanings to the model. And for overall purpose, we used ROC as the evaluation benchmark. We evaluate the confusion matrix as it gives the True Positive (TP) and True Negative (TN) values. For our churn situation, our main focus target is the churn customer. All the evaluation metrics are the results after weighted average the model scores.

4.1 Base Model Evaluation Metric & Confusion Matrix

Table 3 Model results without data pre-processing and parameter tuning

Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	AUC-ROC (%)

Random Forest	95.7	95.7	95.7	95.5	92.2
SVM	85.9	-	86.0	-	50
MLP	88.9	87.9	88.0	88.3	79.8
Xgboost	95.8	96.0	96.0	96.0	91.5
Decision Tree (J48)	93.8	93.6	93.8	93.6	83.7

Table 4 Confusion Matrix before data pre-processing and parameter tuning

Model	TP (/3652)	FN (/3652)	FP (/598)	TN (/598)
Random Forest	3626	26	155	443
SVM	3652	0	598	0
MLP	3478	174	299	299
Xgboost	3619	33	136	462
Decision Tree (J48)	3556	96	168	430

4.2 Tuned Model Evaluation Metric & Confusion Matrix

Table 5 Model results after data pre-processing and parameter tuning

Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	AUC-ROC (%)
Random Forest	95.5	95.4	95.5	95.2	91.8
SVM	85.9	-	85.9	-	50.0
MLP	87.9	87.1	87.9	87.4	80.6
Xgboost	95.7	95.0	96.0	95.0	92.0
Decision Tree (J48)	94.2	94.0	94.2	94.0	85.1

Table 6 Confusion Matrix after data pre-processing and parameter tuning

Model	TP (/3652)	FN (/3652)	FP (/ 598)	TN (/598)
Random Forest	3624	28	164	434
SVM	3652	0	598	0
MLP	3444	208	308	290

Xgboost	3588	64	120	478
Decision Tree (J48)	3576	76	169	429

4.3 Tuned Model with Random Sampling Evaluation Metric & Confusion Matrix

Table 7 Model results applied under sampling techniques after data pre-processing and parameter tuning

Model	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	AUC-ROC (%)
Random Forest	88.5	88.7	88.5	88.5	90.4
SVM	74.8	74.8	74.8	74.8	74.8
MLP	74.3	74.5	74.3	74.3	79.3
Xgboost	84.2	84.0	84.0	84.0	90.7
Decision Tree (J48)	85.1	85.4	85.1	85.1	86.0

Table 8 Confusion matrix with under sampling techniques, after data pre-processing and parameter tuning

Model	TP (/551)	FN (/551)	FP (/598)	TN (/598)
Random Forest	502	49	83	515
SVM	390	161	131	467
MLP	424	127	168	430
Xgboost	445	106	76	522
Decision Tree (J48)	490	61	110	488

4.4 Machine Learning Model Performance Discussion

For Xgboost, the performance for base model and tune model is almost similar, however when we compare the confusion matrix, we can observe that although the TP percentage for no churn label is dropping from 99% to 98% , the TN percentage for churn label is increasing from 77% to 80% . While with random sampling data, the TP percentage is 75% and the TN percentage 94.5%. In this experiment, we can see that the overall tuned Xgboost tuned model, without random sampling, is working better with model accuracy of 95.7% at both target labels with TP 98% and TN 80%.

For Decision Tree (J48), the performance for base model and tuned model have no significant difference. The tuned model which has been pruned (94.2%) shows better accuracy compared to the unpruned model (93.8%). However, if we look at the tuned model for random sampling, it shows a significant decrease in accuracy compared to the pruned model where it went from 94.2% to 85.1%. Similar to Xgboost above, the overall pruned J48 model has a better performance with an accuracy of 94.2% where the TP is 98% and TN is 72%.

Initially we used default settings for Multilayer perceptron, we are getting accuracy of 88.9% and ROC of 79.8%. Although the figure looks promising, from the confusion matrix we found that the total of TN(True Negative = churn) only 299, and FP(False positive) also 299, meaning the model only classified around 50% correctly, doesn't seem to be a good model. We continue to test with the tuned model, the accuracy slightly decreases to 87.9% and ROC slightly increases to 80.6%, but the confusion matrix doesn't seem to improve much, the TN even drops to 290 and FP increases to 308, which show even more wrongly classified. After applied under sampling methods, the accuracy dropped to 74.3% and ROC dropped 79.3%, but the confusion matrix shows that TN increased to 424, which means classified dramatically improved.

For Random Forest, it was found that for this particular problem, hyperparameter tuning using the GridSearchCV tool in Python did not result in improvements in the model's performance. Hence, both the base model and the "tuned" model used the same parameters in both sections, with the features selected for prediction being the only difference. It was found that the fluctuation in performance before and after the feature selection process was small. Nevertheless, it is worth mentioning that the reported scores in the 4.1 and 4.2 sections are weighted averages in Weka. To better handle the imbalance data issue, the 4.3 section with under sampling techniques was included to address the problem. The dataset produced using the sampling techniques has a similar weight for both "churn" and "no churn" classes, making it a balanced dataset. For Random Forest, it was found that the accuracy, precision, recall and F-score consistently range around 88%. The accuracy score is comparable with Ullah et al. (2019), who obtained a score of 89.59% using Random Forest with a similar dataset without employing sampling techniques to address the imbalance data issue.

Random Forest can produce feature importance scores. In Weka, the importance scores are calculated using average impurity decrease. The table below shows the feature importance scores generated by Weka when using the undersampled dataset. It is worth mentioning that higher the mean impurity decrease, the more important the feature is. As shown in the figure, the top five significant features, according to the mean impurity decrease criteria in Weka, are area_code, state, account_length, total_day_calls and total_day_minutes. The area_code and state attributes are both related to the location of a user, while account_length represents the number of months of the customer's subscribership with the telco. The total_day_calls feature refers to the total number or frequency of day calls of a user, while the total_day_minutes feature refers to the total minutes of day calls of a user.

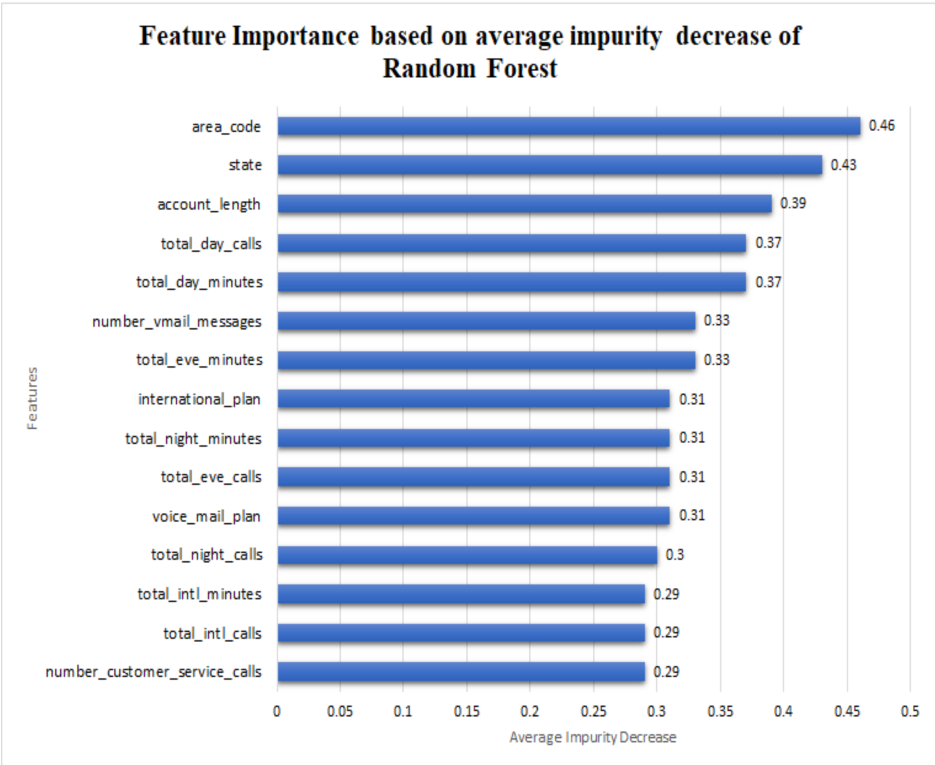


Figure 4 Feature importance based on mean impurity decrease of Random Forest

This study also compares the performance of the classifier using the corrected Paired T-Test criterion leveraged by Weka. The comparison of performance was only performed on the models using the under-sampling method. It was found that in terms of accuracy (Percent correct), Random Forest is statistically significantly better than other classifiers.

The following table displays the precision of the churning class (negative class) before and after using the undersampling technique. It is evident that the technique resulted in a significant increase in the precision scores.

Table 9 Precision comparison for imbalanced vs undersampling data

Model	Precision before undersampling (%)		Precision after undersampling (%)	
	No Churn	Churn	No Churn	Churn
Random Forest	99.2	72.6	91.1	90.9
SVM	100	0	70.7	72.3
MLP	94.3	48.5	76.7	76.8
Xgboost	98.2	79.9	80.8	87.3
Decision Tree (J48)	97.9	71.7	88.9	88.4

5. Conclusion

In present case study, we have identified the important features that contribute in predicting customer churn in the telecom industry. Based on the study, the best model algorithm is Random Forest which is statistically better than other supervised machine learning model with accuracy more than 88.5% for random sampled data, and 95% for imbalanced data. From the feature importance analysis from the best model, the three main key predictors for telco customer churn prediction are area code, state and the account length. In general, according to the evaluation and confusion matrix, we can conclude that the Random Forest with optimal hyperparameter tuning and random sampling, is giving better prediction for both churn and no churn class, as it is improving the precision for churn label is improving from initial 72.6% to 90.9%, while maintaining the precision for no churn label at 91.1%.

References

Ahmad, A.K., Jafar, A. & Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*. https://doi.org/10.1186/s40537-019-0191-6

Al-Shatnwai A.M, Faris.M (2020), Predicting Customer Retention using XGBoost and Balancing Methods. *International Journal of Advanced Computer Science and Applications.* Vol. 11, No. 7, 2020

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*. https://doi.org/10.1016/j.neunet.2018.07.011

Ismail, M. R., Awang, M. K., Rahman, M. N. A., & Makhtar, M. (2015). A multi-layer perceptron approach for customer churn prediction. *International Journal of Multimedia and Ubiquitous Engineering*. https://doi.org/10.14257/ijmue.2015.10.7.22

Morde, V. (2019). XGBoost Algorithm: Long May She Resign! *Towards data science*. Retrieved from https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d#:~:text=XGBoost%20is%20a%20decision%2Dtree,images%2C%20text%2C%20etc

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2914999

Altexsoft (2019). Customer Churn Prediction Using Machine Learning: Main Approaches and Models. *Kdnuggets*. Retrieved from https://www.kdnuggets.com/2019/05/churn-prediction-machine-learning.html

Li, G. & Deng, X. (2012). Customer Churn Prediction of China Telecom Based on Cluster Analysis and Decision Tree Algorithm. *International Conference on Artificial Intelligence and Computational Intelligence.* https://link.springer.com/chapter/10.1007/978-3-642-34240-0_42

Gupta, P. (2017). Decision Trees in Machine Learning. *Towards data science.* Retrieved from https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

Saravanan, N. & Gayathri, V. (2018) Performance and Classification Evaluation of J48 Algorithm and Kendall's Based J48 Algorithm (KNJ48). *International Journal of Computational Intelligence and Informatics, Vol. 7: No. 4.* https://www.periyaruniversity.ac.in/ijcii/issue/marnew/2_mar_18.pdf

Venkatesan, E. V. (2015). Performance Analysis of Decision Tree Algorithms for Breast Cancer Classification. *Indian Journal of Science and Technology.*