Number of words used = 3996

Executive Summary (word count = 285)

Supervised learning algorithm was used to build churn prediction model to help solve a telecoms company's customer churn problem. Decision tree classifiers and optimisation techniques were used for feature selection. The genetic algorithm was applied to a telecoms customer dataset consisting of 6380 rows and 57 features. The Python programming language, Jupyter notebook and scikit-learn python package were used. As a pre-processing of the data, '?' was replaced by np.nan. Binary features were encoded, categorical features were frequency encoded, and missing values in the dataset were filled in by multivariate imputation. Using the processed data, two supervised learning models (with and without feature selection) were constructed. Train-test split was performed on the whole dataset. The size of the test set was set to 0.25. For the models with and without feature selection, performance on the test dataset was evaluated by several measures (accuracy, recall, precision, F1 score, ROC_AUC, MCC). Models with feature selection outperformed models without feature selection in terms of accuracy. However, for the other evaluation metrics, the models with feature selection showed lower values than those without feature selection. The average F-1 score for 10-fold cross-validation increased by 0.84% after feature selection. This implies that the decision tree classifier with feature selection is a more robust model. It is recommended that a decision tree classifier with feature selection be introduced to predict which customers will churn quickly. These results will allow telecom companies to implement customer retention strategies to prevent revenue loss. In the future, the model parameters could be further tuned by super-optimisation algorithms such as Hyperopt. Other tree-based supervised learning algorithms such as Random Forest, XGBoost, LightGBM and CatBoost can also be tested on customer data to build churn prediction models.

Introduction (word count = 736)

Service companies, especially those in the telecommunication services business, are suffering from what is known as 'customer churn' and they are losing valuable customers to competitors. Over the past few decades, the telecommunications industry has undergone significant changes with the increase in new services, technological advances and increased competition due to deregulation [1]. As a result, anticipating customer churn has become critical for players in the telecommunication industry to protect their loyal customer base, grow their organisations and improve their customer relationship management (CRM) [2] [3]. Retaining high-risk customers is one of the most difficult challenges in the telecommunication industry today [4]. With an ever-increasing number of service providers and increasing competition, today's customers have multiple options when it comes to churn. As a result, telecom industry players are realising the importance of retaining existing customers rather than acquiring new ones [2].

Machine Learning is a subfield of Artificial Intelligence and is about algorithms that learn from examples [5]. In machine learning, classification refers to the problem of predictive modelling, where a class label is predicted for a given example of the input data. Binary classification is a classification task with two class labels [6]. Churn prediction (churn or not) belongs to binary classification [7]. In general, there are two types of binary classification tasks: those that represent normal state and those that represent abnormal state [6]. For example, “not churn” is the normal state and “churn” is the abnormal state. Classes in a normal state are assigned to class label 0 and classes in an abnormal state are assigned to class label 1. Some common algorithms that can be used for binary classification are Logistic regression, k-nearest neighbours, Decision trees, Support vector machines and Naive Bayes [6]. Churn prediction is also a kind of imbalanced classification. Imbalanced classification is a classification problem when the classes in the training dataset are unevenly distributed. Imbalances in the class distribution may vary, but severe imbalances are difficult to model and may require special techniques [8].

In order to address the above issues, companies need to correctly predict customer behaviour. There are two ways to manage customer churn, namely reactive and proactive approaches. In the reactive approach, you wait for a cancellation request from a customer and then offer an attractive plan to retain the customer. In a proactive approach, we anticipate the likelihood of churn and provide the customer with a plan accordingly. The problem is a binary classification problem that distinguishes between churners and non-churners. To solve this problem, machine learning has proven to be a very effective technique for predicting information based on previously obtained data [9]. In machine learning models, after pre-processing, feature selection plays an important role in improving classification accuracy. Researchers have developed several methods for feature selection that are effective in reducing dimensionality, computational complexity and overfitting. In churn prediction, features that are useful for predicting churn are extracted from a given input vector.
We propose to build two supervised learning model on the full data, the data with 6380 rows of telecommunication customer data. The first supervised learning model is based on Decision Trees. We used the *python programming language* [10], *Jupyter Notebook* [11] and the *scikit-learn* [12] python package. To build the Decision Trees using scikit-learn python package, data pre-processing is needed. It involves data encoding and imputation. Missing values in the data were imputed using Multivariate Imputation. Categorical features were encoded using Frequency Encoding. We also used Genetic Algorithm and Decision Trees as optimisation technique to select important features. Python package *sklearn-genetic* [13] is used for selection using Genetic Algorithm. We then built Decision Trees supervised learning model on the derived subset of features. After that, the two supervised learning models (with and without feature selection) were critically evaluated based on the evaluation metrics on the testing dataset (after train-test split). The evaluation metrics include Accuracy, Recall, Precision, F-1 score, AUC of ROC (Area Under the Curve of Receiver Operating Characteristic), and MCC (Matthews Correlation Coefficient). The robustness of the two supervised learning models (with and without feature selection) were also evaluated by applying 10-fold cross validation using *scikit-learn* python package (*from sklearn.model_selection import cross_val_score*). The evaluation metric is F-1 score. The cross-validated metrics for model with feature selection is higher than that of the model

without feature selection. This means the Decision Trees model with feature selection by Genetic Algorithm is more robust than that without feature selection.

Literature Review (word count = 834)
A lot of work has been done by researchers on customer churn prediction in the telecommunication industry. Adbelrahim et al. [9] used decision trees, random forests, the GBM tree algorithm and XGBoost (a tree-based algorithm) to predict customer churn. XGBoost outperformed the others in terms of AUC. However, accuracy can be further improved by applying an optimisation algorithm in the feature selection process.

Asthana [14] conducted a comparative analysis of machine learning models for predicting customer churn, using support vector machines, decision trees, naive Bayes and logistic regression. They also looked at the impact of boosting algorithms on classification accuracy. The results showed that SVM-POLY with AdaBoost performed better than the others. However, classification accuracy could be further improved by incorporating feature selection strategies (e.g. univariate selection).
Brandusoiu et al. [15] used three machine learning methods to predict customer churn: neural networks, support vector machines and Bayesian networks. In the feature selection process, principal component analysis was considered to reduce the dimensionality of the data. However, by using optimisation algorithms to improve the feature selection process, the accuracy of the classification can be improved. In the performance evaluation, gain measurements and ROC curves were adopted.

Burez et al. [8] applied random forests with logistic regression and resampling techniques to solve the class imbalance problem. AUC and Lift were also considered in the performance analysis. The effects of advanced sampling techniques such as CUBE were also observed, but no performance improvements were observed. However, optimisation-based sampling techniques can be used to better solve the class imbalance problem.

Coussement et al. [16] attempted to solve the churn prediction problem by support vector machines, logistic regression and random forests. Initially, the performance of the support vector machine was almost equal to that of logistic regression and random forest, but when the optimal choice of parameters was considered, the support vector machine outperformed both logistic regression and random forest in terms of PCC and AUC.

Dahiya et al [17] used two machine learning models, decision trees and logistic regression, to a churn prediction dataset. In their experiments, the WEKA tool was used. However, the above problems can be effectively addressed by employing other machine learning techniques.

Gürsoy et al. [18] used logistic regression and decision tree machine learning models to analyse a large amount of data, but the accuracy obtained was low. Therefore, further improvements are needed to employ other machine learning and feature selection techniques.

The variables that affect respected customer churn were analysed in Hadden et al [19]. They also conducted a comparative study of three machine learning models: neural networks, regression trees and linear regression. The results obtained confirmed that decision trees outperformed the others due to their rule-based architecture. The accuracy obtained can be further improved by using existing feature selection techniques.

Hadden et al. [20] reviewed all the machine learning models considered, as well as a detailed analysis of existing feature selection techniques. Among the prediction models, they found that decision trees performed better than the others. Optimisation techniques also play an important role in improving prediction techniques in terms of feature selection. After a comparative analysis of existing techniques, the authors suggested directions for future research.

Huang et al. [21] adopted different classifiers to a churn prediction dataset and received results confirming that random forests outperformed other classifiers in terms of AUC and PR-AUC analysis. However, the accuracy can be further improved by using optimisation techniques for feature extraction.

Idris et al. [22] attempted to combine genetic programming and AdaBoost machine learning models and compared them with other classification models. The results showed that the combination of genetic programming and AdaBoost had better accuracy than other classification models. However, the accuracy could be further improved by using other optimisation techniques such as gravitational search algorithms and biogeography-based optimisation.

Kisioglu et al. [23] used Bayesian belief networks to the prediction of customer churn. In their experimental analysis, correlation analysis and multiple covariance tests were conducted. The results confirmed the applicability of Bayesian belief networks to churn prediction. These results also suggest directions for future research.

Adwan et al. [24] Applied a multilayer perceptron neural network to predict churn. In this model, artificial neural networks, a back-propagation learning algorithm, are used as the error minimisation criterion. The model uses the following performance metrics: confusion matrix, accuracy, hit rate and churn rate. The model performs well on all performance metrics. For all performance metrics, it was shown that the performance of the model improved from a certain point in time and then remained constant even when the number of hidden layers increased. The model [25] also adopts a multi-layer perceptron for prediction and uses the Accuracy performance metric, which measures good accuracy.

In the study [26], convolutional neural networks were used to predict churn rates. Compared to other methods, convolutional neural networks are very powerful methods as they have feature extraction capabilities in their own right. The performance metrics used in these studies were accuracy, ROC, precision, recall, confusion matrix, AUC, error rate and F-score.

Research Design (word count = 1495)

I. **Data Pre-processing**

First, the missing values in the dataset are represented by '?'. The '?' in the dataset is replaced NAN values so that imputation of the data can be performed.

Second, since all data is input as objects, it is needed to distinguish numerical data from categorical data for later processing. The next step is to cast some object columns to numerical (float) columns.

Third, the 'churn' column is encoded into binary 1/0 numerical column.
After that, frequency encoding is used to deal with other categorical columns. There are many ways to handle categorical columns. One-hot encoding is a good way to do this, but if the cardinality of the features is high, it leads to too many new columns and memory limits are reached. To get rid of the curse of multidimensionality, one-hot encoding was not used. Label encoding is a good solution for ordinal data, but it is not a suitable solution for nominal data. For this dataset, frequency coding is the best solution. Frequency coding is a method that uses the frequency of a category as a label. It helps in the understanding of the model when the frequency has some relationship to the variable of interest and assigns weights in direct or inverse proportion depending on the nature of the data [27].

Fourth, Multivariate Imputation [28] is performed on the data. Missing values need to be dealt with appropriately. The potential bias due to missing values are dependent on the mechanism that caused the missing values and the analysis method used to correct them. Therefore, careful planning and attention should be given when analysing test data containing missing values. Multivariate Imputation is a complex technique for dealing with missing values. It is superior to the single imputation method because it takes into account the uncertainty in the imputation of missing values. Multivariate imputation estimates each feature from all other features. This is a strategy for estimating missing values by modelling each feature with a missing value as a function of the other features in a round robin fashion. Multivariate Imputation is used to fill in the missing values in the dataset. *IterativeImputer* [29] from the *sci-kit learn* python package is used.

II. **Optimisation technique applied for feature selection**

Genetic Algorithm is adopted for feature selection. *sklearn-genetic* [30] python package is used. Genetic Algorithm is chosen because it is the most frequently used metaheuristic for feature selection, and usually perform better than traditional feature selection techniques [31]. *GeneticSelectionCV* in *sklearn-genetic* is used. The estimator is Decision Tree Classifier. The *GeneticSelectionCV* was run for different *max_features* (maximum number of features) from 2 to 30. The scoring is Matthews correlation coefficient (MCC). After running the

GeneticSelectionCV for different *max_features*, the subset of features is chosen from the experiment with the highest score. The score for each experiment is taken from *selector.generation_score*, which is a list of scores (Matthews' correlation coefficients) for each generation.

III. **Supervised learning technique applied**

Decision Tree Classifier is applied because it is fast to run and its result is highly explainable, which is essential for making business decisions [32]. *DecisionTreeClassifier* from *scikit-learn* python package is used.

IV. **Evaluation of the techniques applied in terms of accuracy of their result**

The testing dataset is created by using the *train_test_split* of the *sci-kit learn* python package. The size of the testing set is set to 0.25. The model performance over the testing dataset (after the train-test split of the whole dataset) is evaluated on some chosen metrics for both models with and without feature selection. The chosen evaluation metrics are Accuracy, Recall, Precision, F-1 score, AUC of ROC (Area Under the Curve of Receiver Operating Characteristic), and MCC (Matthews Correlation Coefficient). Then, the *cross_val_score* the *sci-kit learn* python package is used to calculate the cross-validated metrics. 10-fold cross validated F-1 score is calculated for both model with and without feature selection.

Accuracy is the ratio of total correct predictions to the total number of samples Accuracy is the ratio of correct predictions but may give the illusion of high accuracy [33]. Precision tries to answer what proportion of positive identifications was actually correct. Recall tries to answer what proportion of actual positives was identified correctly. F1-score is the harmonic mean instead of precision and recall. F1-score value is close to the lowest values of precision and recall. The value of AUC varies from 0 to 1. A model that predicts 100% incorrectly has an AUC of 0.0 and a model that predicts 100% correctly has an AUC of 1.0. The Mathews Correlation Coefficient (MCC) is used in machine learning to measure the quality of binary and multi-class classification. It takes into account true positives and false positives and is often considered a balanced measure that can be used even when the classes are of very different sizes. The MCC is essentially the value of the correlation coefficient from -1 to +1. A coefficient of 1 represents a perfect prediction, 0 represents an average random prediction and -1 represents an inverse prediction. This statistic is also known as the Phi coefficient. Accuracy and Recall should be as high as possible for the model. F1-score is a useful metric for a balance between precision and recall and there's an uneven class distribution. F1-score adopts the Harmonic Mean at the place of Arithmetic Mean by punishing the extreme values more.

The formula of the evaluation metrics is given below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where

- TP is the number of true positives, i.e., the number of samples that are correctly classified in a class.
- FP is the number of false positives, i.e., the number of samples that are not classified in a class.
- TN is the number of true negatives, i.e., the number of samples that are not correctly classified in a class.
- FN is the number of false negatives, i.e., the number of samples that are incorrectly classified in a class.

V. Algorithmic parameters

Decision Tree Classifier for supervised learning model:

For implementing the Decision Tree Classifier, *sklearn.tree.DecisionTreeClassifier* is used. Table 1 below shows the parameters used in *DecisionTreeClassifier*. *max_depth* determines the maximum depth of the tree. By default, it is set to none. In this case, the decision tree tends to be over-fitted. The *max_depth* parameter is one way of regularising the decision tree, it limits the way the tree grows and prevents over-fitting. The parameter *splitter* is the way in which the decision tree searches for features to split. By default, it is set to 'best'. This means that for each node, the algorithm considers all features and chooses the best split; if the splitter parameter is set to 'random', a random subset of features is considered. The splitting is then performed based on the best features in the random subset. The size of the random subset is determined by the *max_features* parameter. This is where the name Random Forest comes from.

Parameter	**Meaning of parameter**	**Chosen value**
max_depth	the maximum depth of the tree	9
random_state	It controls the randomness of the estimator. Using an integer will produce the same results across different calls.	123
splitter	the strategy used to choose the split at each node. splitter is set to be "best" to choose the best split.	"best"
criterion	the function to measure the quality of a split. criterion is set to be "gini" for the Gini impurity.	"gini"

Table 1: parameters used in DecisionTreeClassifier.

Table 2 below shows other parameters not specified in *DecisionTreeClassifier*. They use the default value.

min_samples_leaf	the minimum number of samples required to be at a leaf node.	1
min_weight_fraction_leaf	the minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.	0.0
max_features	is the number of features to consider when looking for the best split.	None
max_leaf_nodes	Grow a tree with max_leaf_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity. If None then unlimited number of leaf nodes.	None
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	0.0
class_weight	Weights associated with classes in the form {class_label: weight}. If None, all classes are supposed to have weight one.	None
ccp_alpha	Complexity parameter used for Minimal Cost-Complexity Pruning. Ts default value is 0.0. The subtree with the largest cost complexity that is smaller than ccp_alpha will be chosen. By default, no pruning is performed.	0.0

Table 2: Other parameters not specified in DecisionTreeClassifier.

Genetic Algorithm for feature selection:
For implementing the Genetic Algorithm, *GeneticSelectionCV* is used. Table 3 below shows the parameters used in *GeneticSelectionCV*.

Parameter	**Meaning of parameter**	**Chosen value**
estimator	A supervised learning estimator with a fit method.	estimator = DecisionTreeClassifier(max_depth = 9, random_state = 123, splitter = "best", criterion = "gini")
cv	Determines the cross-validation splitting strategy.	rkf = RepeatedStratifiedKFold(n_repeats = 20, n_splits = 5)
verbose	Controls verbosity of output.	0
scoring	a scorer callable object / function with signature scorer(estimator, X, y).	mcc = make_scorer(matthews_corrcoef)
max_features	The maximum number of features selected.	i, this is the variable to vary, ranging from 2 to 30
n_population	Number of population for the genetic algorithm.	200
crossover_proba	Probability of crossover for the genetic algorithm.	0.5
mutation_proba	Probability of mutation for the genetic algorithm.	0.2
n_generations	Number of generations for the genetic algorithm.	10
crossover_independent_proba	Independent probability for each attribute to be exchanged, for the genetic algorithm.	0.5
mutation_independent_proba	Independent probability for each attribute to be mutated, for the genetic algorithm.	0.1
n_gen_no_change	If set to a number, it will terminate optimization when best individual is not changing in all of the previous n_gen_no_change number of generations.	10
caching	If True, scores of the genetic algorithm are cached.	True
n_jobs	Number of cores to run in parallel. Defaults to 1 core. If *n_jobs=-1*, then number of jobs is set to number of cores.	-1
tournament_size	Tournament size for the genetic algorithm.	*default=3*

Table 3: the parameters used in *GeneticSelectionCV*.

The *estimator* is chosen to be the same as the supervised learning algorithm: Decision Tree Classifier with the same parameter. *cv*, the cross-validation scheme is chosen to be 5-fold, 20 repeat scheme. The score for each experiment is taken from *selector.generation_score*, which is a list of scores (Matthews' correlation coefficients) for each generation. In general, the scores improve with each generation, so the final score is used as the final score of the model (selector.generation_score[-1]). *crossover_proba*, the crossover rate determines how genes are passed on to offspring. It determines at what point in time and with what probability the offspring will inherit genes from both parents.

crossover_independent_proba is the probability that the feature will pass on to the child. *mutation_proba*, the mutation rate determines how many chromosomes are mutated in a generation and the mutation rate ranges from 0 to 1. The purpose of mutation is to prevent the Genetic Algorithm from converging to a local optimum. *mutation_independent_proba* is combined with *mutation_proba* to determine the likelihood of putting a feature to the feature set.The initial population (of size *n_population*) is created randomly from the sample space of the feature set. The extent of these sets is limited by the parameter *max_features*. This parameter sets the maximum size of each feature subset.To determine the next generation of members, a tournament is used to select members. The number of members in a tournament is set by *tournament_size*. *tournament_size* means that several members of the population are selected to compete against each other based on a scoring metric. The winner of the tournament is selected as the parent of the next generation. The *n_gen_no_change* parameter monitors whether the best member of the population has not changed over several generations. In this case, the search has found the best state. Consider changing the selection further by increasing the probability of mutation or crossover.

Experimental Results and Analysis (word count = 273)
Findings:
For the evaluation metrics on the testing dataset, the Decision Tree Classifier with feature selection outperform that without feature selection in terms of Accuracy. However, the Decision Tree Classifier with feature selection has lower values in other evaluation metrics compared to that without feature selection. Notably, F-1 score decreases by 0.0507 and MCC decreases by 0.0094 after feature selection. Note that F-1 score and MCC are more important for model evaluation with imbalanced dataset.

Moreover, for the 10-fold cross-validation, Decision Tree Classifier with feature selection has a higher F-1 score than that without feature selection. After feature selection, the average 10-fold cross-validated F-1 score increases by 0.84%. This means that the Decision Tree Classifier with feature selection is a more robust model.

Feature selection is an important issue in classification as it can have a significant impact on the accuracy of the classifier [34]. The dimensionality of the dataset can be reduced, thus reducing processor and memory usage. The data becomes easier to understand and easier to study.

Model	**Accuracy**	**Recall**	**Precision**	**F1-score**	**ROC_AUC**	**MCC_metric**
Decision Tree Classifier *without* feature selection	0.7009	0.2207	0.4103	0.287	0.5509	0.128
Decision Tree Classifier *with* feature selection	0.7122	0.1632	0.4277	0.2363	0.5407	0.1186

Table 4: Evaluation metrics on the testing dataset for supervised learning models

Model	**10-fold cross-validated F1-score**
Decision Tree Classifier *without* feature selection	0.237
Decision Tree Classifier *with* feature selection	0.239

Table 5: Evaluation of the robustness of the supervised learning models

Discussion on how these results can help the business solve the customer churn problem: These results can be used to deploy a supervised learning model to predict which customers will churn quickly. Churn prediction involves analysing whether a customer is about to churn and why. Churn prediction allows us to understand the causes of churn and implement retention strategies to prevent loss of revenue. In the telecommunications industry, it is estimated that the cost of acquiring a new customer is more than five times the cost of retaining a customer, further emphasising that churn forecasting is a very important area for telecommunications companies [35]

Conclusion (word count = 373)
To help solve the customer churn problem in the telecommunication company, supervised learning: Decision Tree Classifier and optimisation technique for feature selection: Genetic Algorithm were applied on a telecommunication customer dataset with 6380 rows and 57 features. Genetic Algorithm is selected because it is the most frequently used metaheuristic for feature selection, and usually perform better than traditional feature selection techniques. Decision Tree Classifier is used because it is fast to run and its model output is highly explainable, which is suitable for business application.

Python programming language, Jupyter Notebook and the scikit-learn python package were used. Data pre-processing includes replacement of '?' by np.nan, Binary Feature Encoding, Frequency Encoding for categorical features as well as Multivariate Imputation to fill missing values in the dataset. Two supervised learning models (with and without feature selection) were built with the processed data.

Train-test split was performed on the whole dataset. The size of the testing set is set to 0.25. The model performance over the testing dataset was evaluated on some metrics (Accuracy, Recall, Precision, F1-score, ROC_AUC and MCC) for both models with and without feature selection. Model with feature selection outperformed that without feature selection in terms of Accuracy. However, the model with feature selection has lower values in other evaluation metrics compared to that without feature selection. After feature selection, the average 10-fold cross-validated F-1 score increases by 0.002, which is an increase by 0.84%. This means that the Decision Tree Classifier with feature selection is a more robust model. It is suggested to deploy the Decision Tree Classifier with feature selection to predict which customers will churn quickly. These results allow the telecommunication company to implement customer retention strategies to prevent loss of revenue.

Future work can include further tuning of model parameters by hyperoptimisation algorithm such as *Hyperopt* [36] because it is better than other parameter optimisation algorithms like Grid Search [37] and Randomised Search [38]. Other tree based supervised learning algorithms such as *Random Forest* [39]*, XGBoost [40], LightGBM [41]* and *CatBoost [42]* can also be tested with the customer data to build the churn prediction model because these algorithms are highly interpretable, easy to implement, and their model accuracies are known to be high compared to Decision Tree Classifier.

Reference

[1] Y. Huang, B. Huang, and M.-T. Kechadi, “A rule-based method for customer churn prediction in telecommunication services,” Advances in Knowledge Discovery and Data Mining, pp. 411–422, 2011.

[2] A. Idris and A. Khan, “Customer churn prediction for Telecommunication: Employing various various features selection techniques and tree based ensemble classifiers,” 2012 15th International Multitopic Conference (INMIC), 2012.

[3] R. Mohanty and K. J. Rani, “Application of computational intelligence to predict churn and non-churn of customers in Indian telecommunication,” 2015 International Conference on Computational Intelligence and Communication Networks (CICN), 2015.

[4] V. L. Miguéis, D. Van den Poel, A. S. Camanho, and J. Falcão e Cunha, “Modeling Partial Customer Churn: On the value of first product-category purchase sequences,” Expert Systems with Applications, vol. 39, no. 12, pp. 11250–11256, 2012.

[5] T. M. Mitchell, Machine learning. New York: McGraw Hill, 2017.

[6] R. Kumari and S. Kr., “Machine learning: A review on Binary Classification,” International Journal of Computer Applications, vol. 160, no. 7, pp. 11–15, 2017.

[7] B. Huang, M. T. Kechadi, and B. Buckley, “Customer churn prediction in Telecommunications,” Expert Systems with Applications, vol. 39, no. 1, pp. 1414–1425, 2012.

[8] J. Burez and D. Van den Poel, “Handling class imbalance in customer churn prediction,” Expert Systems with Applications, vol. 36, no. 3, pp. 4626–4636, 2009.

[9] A. K. Ahmad, A. Jafar, and K. Aljoumaa, “Customer churn prediction in telecom using machine learning in Big Data Platform,” Journal of Big Data, vol. 6, no. 1, 2019.

[10] “Welcome to Python.org,” Python.org. [Online]. Available: https://www.python.org/. [Accessed: 27-Feb-2022].

[11] “Project jupyter,” Project Jupyter. [Online]. Available: https://jupyter.org/. [Accessed: 27-Feb-2022].

[12] “Learn: Machine learning in python - scikit-learn 0.16.1 documentation,” scikit. [Online]. Available: https://scikit-learn.org/. [Accessed: 27-Feb-2022].

[13] “Sklearn-Genetic,” PyPI. [Online]. Available: https://pypi.org/project/sklearn-genetic/. [Accessed: 27-Feb-2022].

[14] P. Asthana, “A comparison of machine learning techniques for customer churn prediction,” International Journal of Pure and Applied Mathematics, vol. 119, no. 10, pp. 1149–1169, 2018.

[15] I. Brandusoiu, G. Toderean, and H. Beleiu, “Methods for churn prediction in the pre-paid Mobile Telecommunications Industry,” 2016 International Conference on Communications (COMM), 2016.

[16] K. Coussement and D. Van den Poel, “Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques,” Expert Systems with Applications, vol. 34, no. 1, pp. 313–327, 2008.

[17] K. Dahiya and S. Bhatia, “Customer churn analysis in telecom industry,” 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2015.

[18] U¸S Gürsoy, “Customer churn analysis in telecommunication sector,” ˙Istanbul Üniversitesi ˙I¸sletme Fakültesi Dergisi, vol. 39, no. 1, pp. 35–49, 2010.

[19] J. Hadden, A. Tiwari, R. Roy and D. Ruta, “Churn prediction: Does technology matter,” International Journal of Intelligent Technology, vol. 1, no. 2, pp. 104–110, 2006.

[20] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, “Computer Assisted Customer Churn Management: State-of-the-art and future trends,” Computers &amp; Operations Research, vol. 34, no. 10, pp. 2902–2917, 2007.

[21] Y. Huang, F. Zhu, M. Yuan, K. Deng, Y. Li, B. Ni, W. Dai, Q. Yang, and J. Zeng, “Telco churn prediction with Big Data,” Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015.

[22] A. Idris, A. Khan, and Y. S. Lee, “Genetic programming and Adaboosting based churn prediction for Telecom,” 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2012.

[23] P. Kisioglu and Y. I. Topcu, “Applying bayesian belief network approach to customer churn analysis: A case study on the telecom industry of Turkey,” Expert Systems with Applications, vol. 38, no. 6, pp. 7151–7157, 2011.

[24] O. Adwan, H. Faris, K. Jaradat, O. Harfoushi, and N. Ghatasheh, “Predicting customer churn in telecom industry using multilayer preceptron neural networks:modeling and analysis,” Life Science Journal, vol. 11, no. 3, pp. 75–81, 2014.

[25] A. Sharma and P. Kumar Panigrahi, “A neural network based approach for predicting customer churn in cellular network services,” International Journal of Computer Applications, vol. 27, no. 11, pp. 26–31, 2011.

[26] A. Chouiekh and E. H. El Haj, “Deep convolutional neural networks for customer churn prediction analysis,” International Journal of Cognitive Informatics and Natural Intelligence, vol. 14, no. 1, pp. 1–16, 2020.

[27] S. Galli, Python feature engineering cookbook: Over 70 recipes for creating, engineering, and transforming features to build machine learning models. Birmingham ; Mumbai: Packt, 2020.

[28] S. van Buuren and K. Groothuis-Oudshoorn, “Mice: Multivariate imputation by chained equations INR,” Journal of Statistical Software, vol. 45, no. 3, 2011.

[29] “Sklearn.impute.IterativeImputer,” scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html#sklearn.impute.IterativeImputer. [Accessed: 28-Feb-2022].

[30] “Sklearn-Genetic,” PyPI. [Online]. Available: https://pypi.org/project/sklearn-genetic/. [Accessed: 28-Feb-2022].

[31] S. C. Yusta, “Different metaheuristic strategies to solve the feature selection problem,” Pattern Recognition Letters, vol. 30, no. 5, pp. 525–534, 2009.

[32] F. K. Došilović, M. Brčić and N. Hlupić, "Explainable artificial intelligence: A survey," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2018, pp. 0210-0215, doi: 10.23919/MIPRO.2018.8400040.

[33] S. A. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. H. Mason, “Defection detection: Measuring and understanding the predictive accuracy of customer churn models,” Journal of Marketing Research, vol. 43, no. 2, pp. 204–211, 2006.

[34] E. M. Karabulut, S. A. Özel, and T. İbrikçi, “A comparative study on the effect of feature selection on classification accuracy,” Procedia Technology, vol. 1, pp. 323–327, 2012.

[35] H. Jain, A. Khunteta, and S. Srivastava, “Telecom churn prediction and used techniques, datasets and performance measures: A Review,” Telecommunication Systems, vol. 76, no. 4, pp. 613–630, 2020.

[36] “Hyperopt: Distributed asynchronous hyper-parameter optimization,” Hyperopt Documentation. [Online]. Available: http://hyperopt.github.io/hyperopt/. [Accessed: 28-Feb-2022].

[37] “3.2. tuning the hyper-parameters of an estimator,” scikit. [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html. [Accessed: 28-Feb-2022].

[38] “Sklearn.model_selection.RANDOMIZEDSEARCHCV,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. [Accessed: 28-Feb-2022].

[39] L. Breiman, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[40] T. Chen and C. Guestrin, “XGBoost,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[41] G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS), pp. 3146–3154, 2017.

[42] A. Dorogush, V. Ershov, and A. Gulin, “CatBoost: Gradient boosting with categorical features support,” in Proc. Workshop ML Syst. Neural Inf. Process. Syst. (NIPS), pp. 1–7, 2017