BIRMINGHAM CITY University

CMP6200/DIG6200
Individual Undergraduate Project
2022–2023

Literature Review and Methods

Project Title:

Customer churn prediction of telecommunication companies using supervised machine learning

Course: **CMP6200 Individual Honours Project**

Student Name: **Mitra Bitaraf Fazel**

Student Number: **20121126**

Supervisor: **Dr. Atif Azad**

Table of Contents

List of Figures

List of tables

1. Report Introduction

In the current competitive global market, companies require in-depth understanding of the costumer behaviour and needs in order to provide personalised services and goods, and maintain or expand their market. Customer churn prediction is a field that studies company-costumer interaction to predicts the customers' behaviour and help companies to provide the required service. This will enable companies to maintain their customers and acquire new ones, while providing the customers with better services.

There are various approaches to analyse customer churn and to predict it. One of these approaches is the application of machine learning which can handle and explore complex features and large data sets for predicting customer behaviour and needs. In line with the above, this report presents a critical review of the literature on customer churn prediction approaches, its influential factors, and the application of machine learning for customer churn prediction. This literature review is the basis for developing an efficient supervised machine learning model to be trained with a large dataset of customers.

In the following section, the aim and objectives of this report is presented, followed by the literature review in Chapter 2. Then, in Chapter 3, the methodology for developing the machine learning model is presented. Appendices providing extra information and the references used are presented at the end of the report.

1.1. Aim and Objectives

1.1.1 Aim

The aim of this study is to develop a supervised machine learning model for predicting customer churn in the telecommunications industry, utilizing historical customer data to identify patterns and trends that may indicate an increased likelihood of churn. The goal is to accurately predict which customers are at risk of leaving the company, and to use this information to develop targeted interventions to reduce churn rates and improve customer retention.

1.1.2 Objectives

- Identify patterns and trends in customer behaviour that predict churn.

- Develop a predictive model that utilizes these important factors and variables, using machine learning techniques, with the aim of accurately identifying customers who are at risk of churning.
- Use the prediction model to target at-risk customers with retention campaigns and incentives.
- Use insights gained from the prediction model to inform decisions about product development and marketing strategies.
- Evaluate the performance of the predictive model using appropriate evaluation metrics in order to assess its effectiveness in predicting customer churn.
- Identify areas where further research is needed, and make recommendations for future work in the field of customer churn prediction in telecommunications companies.

1.2. Literature Search Methodology

In order to perform a c comprehensive review on customer churn prediction approaches and the application of machine learning for the above, a set of themes were defined that cover the study area. For each theme a subset of keywords was identified and listed. For each key word, a number of credible academic publications were critically studied, summarised, and presented under the relevant section.

2. Literature Review

2.1. Themes

The current literature review focuses on the theme of customer churn prediction, which is the main topic of investigation. To structure the review, several themes were identified and explored, including: (1) definition and importance of customer churn prediction and its methods; (2) supervised machine learning (SML) and its applicability for customer churn prediction; (3) SML algorithms, specifically decision tree classification and random forest classification; (4) data cleaning and pre-processing; (5) extraction, including the analysis of correlations between different features; and (6) evaluation metrics, which are used to quantify the performance of SML. Key words were used to search the literature for relevant studies and information.

2.2. Review of Literature

In the following, the literature on the above themes is critically reviewed, to study and understand the theory, application, advantages, and disadvantages of the available methods and tools outlines in the key words in the previous section.

2.2.1 Customer churn prediction

Keyword: A description of customer churn prediction

Customer churn prediction is the process of using data mining and machine learning techniques to predict which customers are most likely to stop doing business with a company. This can be used to identify high-risk customers, so that the company can take proactive steps to retain them, such as offering special promotions or incentives. Churn prediction models typically use historical data on customer behaviour, such as purchase history and customer demographics, to make predictions about future behaviour. These models are commonly used in industries such as telecommunications, finance, and e-commerce (Churn, 2003; Zaki and O'malley, 2007)

Keyword: The importance of customer churn prediction (why, for whom, and how?)

The prediction of customer churn is critical for businesses and organizations that rely on recurring revenue from customers. It is particularly relevant in industries such as telecommunications, finance, e-commerce, and SaaS where customer acquisition costs are high and retaining customers is vital. By predicting customer churn, businesses can take proactive measures to retain customers and reduce the costs associated with acquiring new ones, thereby improving overall revenue. Additionally, customer churn prediction can inform better customer segmentation and targeting strategies, inform product development and marketing efforts, and improve the customer experience. Ultimately, predicting customer churn can lead to increased customer lifetime value, reduced operational costs, and improved profitability (Reichheld and Sasser, 1996; Srivastava and S. Rangaswamy, 2002; Zaki and O'malley, 2007).

Keyword: The methods have been used for customer churn prediction

The prediction of customer churn, or the likelihood that a customer will discontinue their relationship with a company, can be accomplished through various methods in the field of machine learning. Some of the commonly used techniques include logistic regression, decision trees, random forests, gradient boosting, neural networks, support vector machines, K-Means clustering, association rule mining, and hybrid models. Each method presents its own

advantages and disadvantages and the choice of which one to use will depend on the characteristics of the data and the specific business problem at hand. The use of labelled data in supervised machine learning methods, such as decision trees and logistic regression, enables the training of models that can predict customer churn based on past customer behaviour. On the other hand, unsupervised learning methods like K-Means clustering can identify customer groups with similar characteristics and behaviours, aiding in the prediction of customer churn. The combination of multiple methods in hybrid models can also improve the accuracy of churn prediction. (Wilcox, 2002; Churn, 2003; Zaki and O'malley, 2007; Raza and Besar, 2017; Proença et al., 2010).

Table 1. Comparison of customer churn prediction methods (Wilcox, 2002; Churn, 2003; Zaki and O'malley, 2007; Raza and Besar, 2017; Proença et al., 2010)

	Method	Advantage	Disadvantage
Supervised Machine Learning	Logistic Regression	Simple to implement, easy to interpret, handles categorical and numerical data, can handle multiple input variables	Assumes linearity between the input variables and output variable, assumes independence of input variables
	Decision Trees	Simple to interpret, handles categorical and numerical data, can handle multiple input variables, can be visualized	Prone to overfitting, assumption of independence of input variables
	Random Forest	Handle high-dimensional data, non-linear relationships and handle missing data, can be used for feature selection	Prone to overfitting, not easy to interpret
	Gradient Boosting	Handle high-dimensional data, non-linear relationships and handle missing data, can be used for feature selection, powerful method for feature selection	Prone to overfitting, not easy to interpret
	Neural Networks	Handle high-dimensional data, non-linear relationships and handle missing data, can handle complex data, can handle interactions among the input variables	Can be difficult to interpret, require a large amount of data

Support Vector Machines	Handle high-dimensional data, non-linear relationships and handle missing data, can handle complex data, can handle interactions among the input variables	Can be difficult to interpret, require a large amount of data
K-Means Clustering	Unsupervised learning, can identify groups of customers who have similar characteristics and behaviour	Assumes spherical shape of clusters, sensitive to initial conditions, not easy to interpret
Association rule mining	Can find association between different variables and customer churn	Not easy to interpret, require a large amount of data
Hybrid models	Combining multiple methods can improve the accuracy of the prediction	Complexity, not easy to interpret

Among the methods above, supervised machine learning methods provide a wide range of advantages to explore large datasets with nonlinear interdependencies. The following summarises these advantages, as well as the disadvantages.

Advantages of supervised machine learning method

Supervised machine learning is a widely used method for customer churn prediction that can achieve high accuracy when trained on a large amount of labelled data. It is capable of handling large amounts of data and extracting useful information, as well as automating the prediction process and handling multiple input variables. However, it requires labelled data that may be difficult or expensive to obtain, and can overfit the data if not properly validated, leading to poor performance on new data. Additionally, supervised machine learning algorithms can be complex and difficult to interpret, and assume independence of input variables, which may not always be true. They can also be inflexible when it comes to handling new, unseen data that deviates from the training data. The choice of method for a specific problem will depend on the characteristics of the data and the business problem, as well as the available resources (Raza and Besar, 2017).

2.2.3 Supervised machine learning (SML)

Applications of SNL include classification and regression which are discussed in the following. Figure 1 shows schematically how SML estimates a trend in a set of data, i.e., regression, versus compartmenting a data set by labelling them, i.e., classification. Each of these applications address different problems in engineering, medicine, data science, etc. The following explores each of these SNL applications.

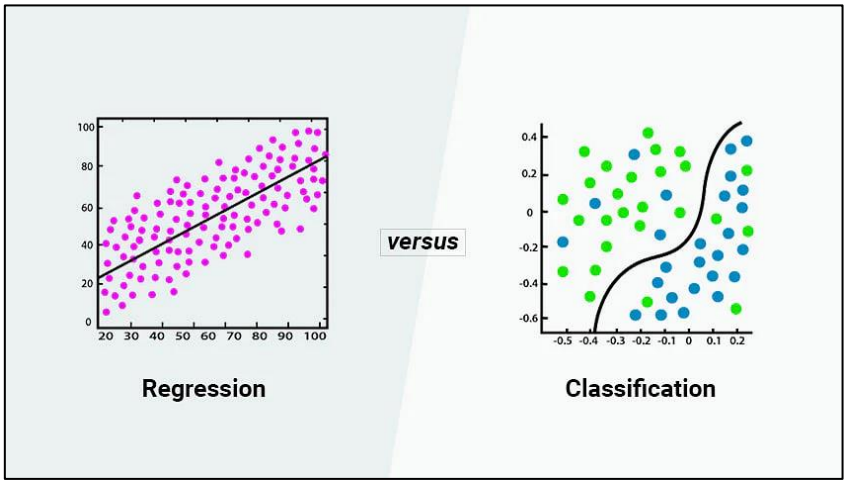

Figure 1. Schematic of the application of supervised machine learning for regression versus classification problems (https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article)

Keyword: Classification task

Supervised machine learning methods carry out classification tasks by learning the relationship between the input variables and the output variable (churn or non-churn) from a labelled training dataset. An example is to train a SML for separating spam emails from ham email (see figure 2). SML can be trained to recognise specific common features in spam emails and correlate them with the spam label.

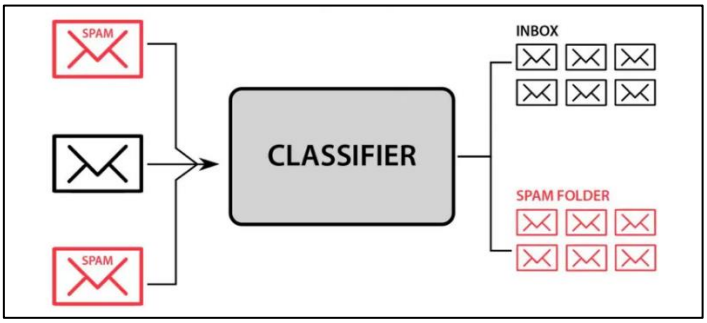

Figure 2. SML for classifying email into spam and ham emails (https://www.analytixlabs.co.in/blog/classification-in-machine-learning).

The process of classification in supervised machine learning involves several steps: data preparation, feature selection, model training, evaluation, tuning, and deployment. The first step is to clean and pre-process the data. Then, the input features are selected, followed by training a model on labeled training data. The trained model is evaluated on a separate dataset, and its parameters may be adjusted for improved performance. Finally, the satisfactory model is deployed for making predictions on new data. The steps and techniques can vary for different methods, but the overall process remains similar.

Keyword: Regression task

Supervised machine learning methods can also be used to carry out regression tasks, which involve predicting a continuous output variable based on a set of input variables. The underlying theory and equations for different regression methods can vary, but a common approach is linear regression.

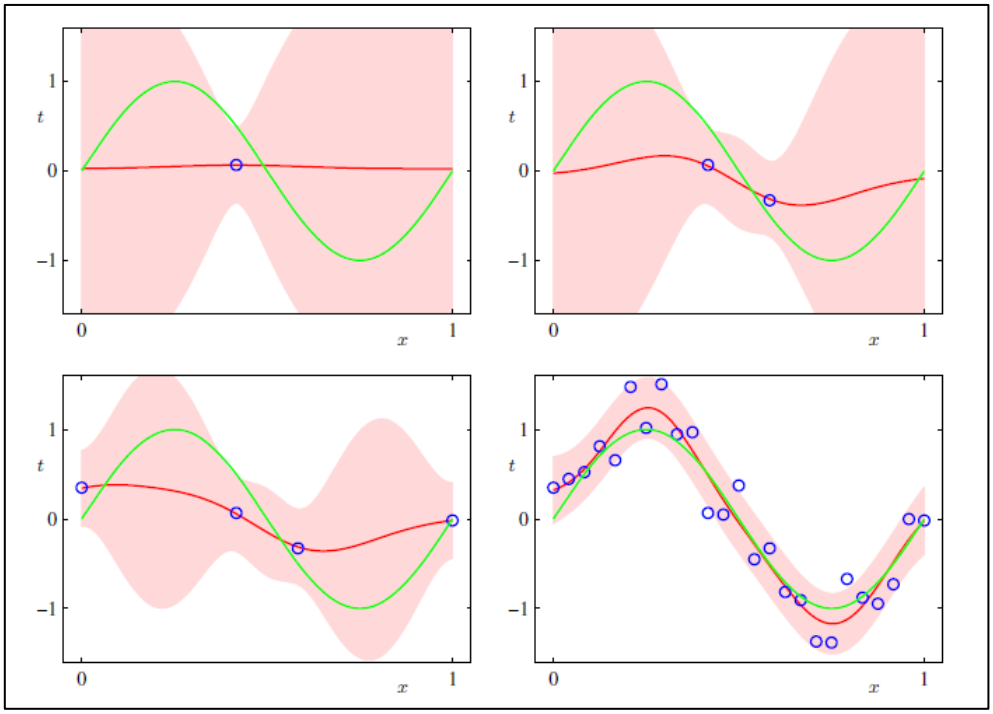

Figure 3. Application of SML for regression with different number of data, where green line in the plots represent the target pattern, blue circles are the sample points used to train the SML, and the red lines are the SML estimations (https://stats.stackexchange.com/questions/311266/gaussian-basis-function-in-bayesian-linear-regression).

Linear regression is a method that models the relationship between the input variables and the output variable as a linear equation. The equation can be represented as:

$y = b_0 + b_1x_1 + b_2x_2 + ... + b_n*x_n$

Where y is the output variable, x_1, x_2, ..., x_n are the input variables, and b_0, b_1, b_2, ..., bn are the coefficients of the equation. The goal of linear regression is to find the values of the coefficients that best fit the data.

To find the values of the coefficients, the method of least squares is used. The method of least squares is a technique for finding the values of the coefficients that minimize the sum of the squared residuals. The residual is the difference between the predicted value and the actual value. The sum of the squared residuals is represented by the following equation:

$SSE = \Sigma(y - y')^2$

Where y is the actual value, y' is the predicted value, and Σ represents the sum over all data points.

Once the coefficients have been estimated, the linear regression model can be used to make predictions on new, unseen data. The model can be used to predict the output variable for a given set of input variables by plugging the values of the input variables into the equation and solving for the output variable (James et al., 2013; Hastie, 2009; Mitchell and McGraw-Hill, 1997).

2.2.4 Supervised machine learning algorithms

As discussed previously, decision tree and random forest are two of the SML algorithm that are widely used due to the advantages which they offer.

Keyword: Decision trees classification

The Decision Tree Algorithm is a supervised learning method that can be used for both classification and regression tasks. It builds a tree-like model of decisions and their possible outcomes based on the concepts of entropy and information gain. The goal is to reduce the entropy by making a series of decisions that split the data into subsets based on the values of the input variables. The algorithm starts with the root node that represents the entire dataset and calculates the entropy of the dataset. It then selects the input variable with the highest information gain and splits the data into subsets. This process is repeated recursively until a stopping criterion is met. The final result is a tree structure that represents the decisions and

their consequences with leaf nodes representing class labels. The tree can be used for predictions by traversing the tree and arriving at a leaf node that represents the prediction.

An example of using the Decision Tree Algorithm is in a guessing game, where a person eliminates options by asking questions and categorizing objects to arrive at the correct one. For example, a person imagines an object such as a car or a bus, and the other person eliminates options by asking questions such as "does it have wheels?" until the correct object is identified. However, the decision tree algorithm can be prone to overfitting, which can be addressed through techniques like pruning, cross-validation, and bagging (James et al., 2013; Hastie et al., 2009; Mitchell and McGraw-Hill, 1997).

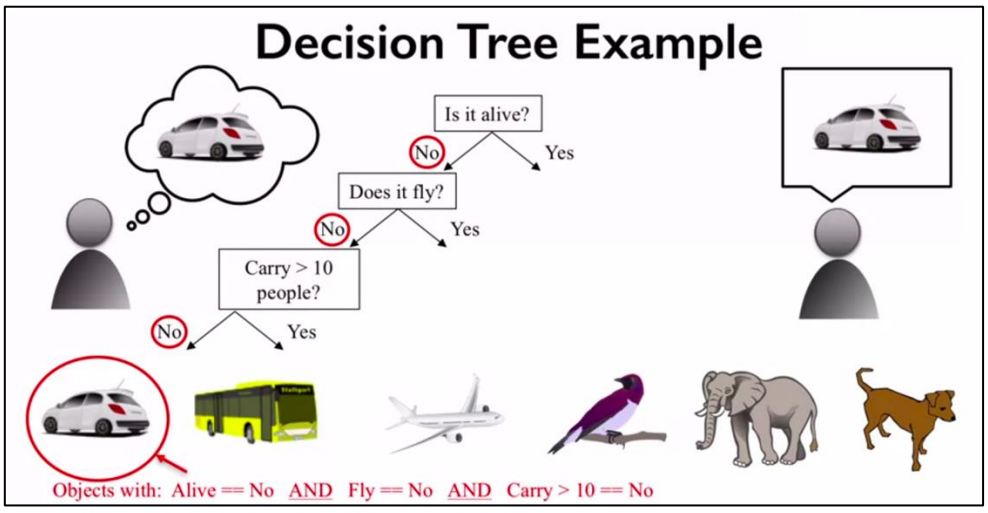

Figure 4. Example of decision tree in a guessing game (https://machine-learning-and-data-science-with-python.readthedocs.io/en/latest/assignment5_sup_ml.html)

The decision tree algorithm uses the following main equations to calculate information gain and entropy.

Information Gain: This equation calculates the decrease in entropy after a dataset is split on an attribute. It is calculated as the difference between the current entropy and the weighted average entropy of the subsets. The equation is as follows:

$IG(S, A) = \text{Entropy}(S) - \Sigma (|S_v| / |S|) * \text{Entropy}(S_v)$

Where S is the current dataset, A is the attribute used to split the dataset, S_v is the subset of S for a given value of A, $|S_v|$ is the number of instances in S_v, and |S| is the number of instances in S.

Entropy: This equation calculates the impurity of a dataset. It is calculated as the sum of the probability of each class *i* multiplied by the log base 2 of the probability of the class *i*. The equation is as follows:

Entropy(S) = -Σ p(*i*|*S*) * log2(p(*i*|*S*))

Where S is the dataset, *i* is the class, and p(*i*|*S*) is the probability of class *i* in dataset *S*.

These equations are used to calculate the information gain and entropy at each decision point in the decision tree algorithm. The input variable with the highest information gain is selected to split the data, reducing the entropy and making the decision tree more accurate (Mitchell, McGraw-Hill, 1997; Breiman; 2001).

Keyword: Random Forest classification

The Random Forest algorithm is an ensemble learning method that combines multiple decision trees to produce predictions. The algorithm leverages the concept of decision trees and bootstrap aggregating (bagging) to arrive at its predictions. The process of building a Random Forest starts with bootstrapping the data, which involves taking a random sample with replacement from the dataset. For each bootstrapped sample, a decision tree is trained. At each decision point in the tree, a random subset of the input variables is selected and the best one is used for splitting the data. This process is repeated for a specified number of decision trees, which are grown independently and in parallel. The final prediction is made by averaging the predictions of all the decision trees in the forest. The Random Forest algorithm reduces variance and correlation in predictions, leading to a more robust model that generalizes better to new, unseen data. However, the algorithm can be prone to overfitting and techniques such as cross-validation and pruning may be employed to mitigate this issue (Breiman, 2001; James et al., 2013).

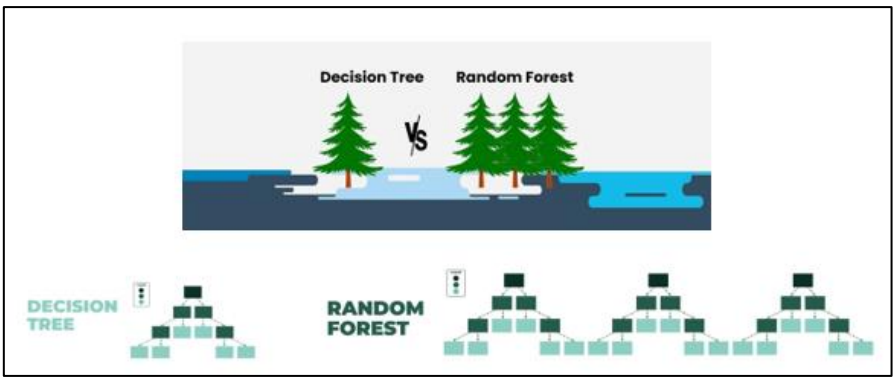

Figure 5. C**omparing the structure of a random forest versus a decision three algorithm (https://www.datatrained.com/post/decision-tree-vs-random-forest)**

2.2.5 Data cleaning and pre-processing

Keyword: Data splitting e.g., balanced and imbalanced, Random state

Data splitting is an important step in the machine learning process that involves dividing a dataset into multiple subsets for training and evaluating a model. The subsets are typically a training set, a validation set, and a test set. The main goal of data splitting is to provide an accurate estimate of the model's performance on unseen data. This is achieved by evaluating the model on subsets that are independent of the training set, which helps prevent overfitting.

There are several methods for data splitting, including balanced data splitting, imbalanced data splitting, random state data splitting, k-fold cross-validation, leave-one-out cross-validation, time series data splitting, and stratified sampling. Each method has its own advantages and limitations, and the choice of method depends on the characteristics of the data and the problem being solved. For example, the balanced data splitting method ensures that the class distribution is roughly the same in all subsets, while the imbalanced data splitting method can be useful when the data is naturally imbalanced. The random state data splitting method is easy to implement but may lead to overfitting or underfitting. It is important to experiment with different methods and evaluate the performance of the model to select the best one for a particular problem (Kuhn and Johnson, 2013; Bhargava and Agrawal, 2018).

Keyword: Scaling

Scaling refers to the process of transforming variables to a common scale to ensure that all features have equal influence in a machine learning model. It is an important step in pre-processing for many algorithms as the features' range and distribution can impact the performance of the model.

There are several techniques for scaling, including:

1. Standardization: Subtracting the mean and dividing by the standard deviation, which results in a distribution with mean 0 and standard deviation 1. Mathematically, this can be expressed as:

$x' = (x - \mu) / \sigma$

where x is the original feature, x' is the standardized feature, μ is the mean and σ is the standard deviation.

2. Min-Max Scaling: Rescaling the feature values between two specific values by subtracting the minimum value and dividing by the range. The equation can be expressed as:

$x' = (x - xmin) / (xmax - xmin)$

where x is the original feature, x' is the rescaled feature, $xmin$ is the minimum value and $xmax$ is the maximum value. When the min-max scaling is performed between 0 and 1, it is referred to as normalisation (Patel, 2019).

Keyword: Shapley values

The Shapley value in the context of machine learning is a measure of the contribution of each feature to the prediction performance of a model. It assigns a unique value to each feature based on its contribution to the prediction accuracy in different combinations with other features. The Shapley value is calculated as the average of the marginal contribution of each feature over all possible combinations of features and is represented mathematically as: $V(i) = (1/n!) * \Sigma(p!) * (n\text{-}p\text{-}1)! * (\Pi\, v(S))$, where n is the number of features, p is the size of the coalition (combination of features), and $v(S)$ is the contribution of the coalition to the prediction accuracy of the model. The Shapley value provides insight into the relative importance of each feature for the prediction performance of the model (Patel, 2019).

Keyword: data visualisation

Data visualization is an important part of preprocessing in machine learning as it helps to understand and prepare the data for modelling. It presents the data in a way that makes patterns, trends, and relationships easier to identify, allowing for insights into the data that might not be obvious from raw data. These insights inform preprocessing steps like normalizing or scaling the data to improve model performance. Data visualization plays a key role in the preprocessing step of the machine learning pipeline (Alpaydin, 2010).

Keyword: Encoding

The encoding of categorical data involves converting categorical variables into numerical representations that can be used by machine learning algorithms. This is a crucial preprocessing step and there are several methods for encoding including one-hot encoding, label encoding, ordinal encoding, binary encoding, count encoding, target encoding, and frequency encoding. The method selected depends on the problem specifications and characteristics of the data.

Encoding impacts the success of the modelling process and it is important to choose the appropriate encoding method for the data (Alpaydin, 2010; Gnat, S., 2021).

Keyword: Missing values

Dealing with missing values in machine learning is an important preprocessing step as missing data can significantly impact the performance of machine learning algorithms. There are several methods for addressing missing values, including mean/median/mode imputation, K-Nearest Neighbours imputation, linear interpolation imputation, multiple imputation, regression imputation, model-based imputation, and data deletion. The choice of method will depend on the specific characteristics of the data, the amount of missing data, and the requirements of the problem. The appropriate method should be chosen based on a thorough understanding of the data and the problem, and a combination of methods may be used in some cases to achieve optimal results (Gnat, S., 2021).

Keyword: Outliers

Outliers can have a detrimental effect on the accuracy of machine learning models, as they can significantly skew the distribution of the data and cause a shift in the mean. To address this issue, several methods are employed in machine learning, including data cleaning, winsorisation, data transformation, outlier detection, and robust modelling. The appropriate method for dealing with outliers will depend on the characteristics of the data, the amount of outlier data, and the requirements of the problem. In some instances, a combination of methods may be used to achieve optimal results (Alpaydin, 2010; Gnat, S., 2021).

2.2.6 Evaluation metrics

Keyword: Confusion Matrix

The Confusion Matrix is a tool used to evaluate the performance of a classification model by comparing predicted values to actual values. It comprises four quadrants: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), representing the number of correctly and incorrectly predicted observations as positive or negative. Evaluation metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC can be calculated from the quadrants. The Confusion Matrix provides a comprehensive and concise summary of a model's performance, highlighting areas for improvement and strong performance. It can be used for

binary classification problems and extended to multi-class classification problems. (Brett, 2015; Alpaydin, 2010).

Keyword: Accuracy Score

Accuracy is a commonly used evaluation metric in machine learning that measures the proportion of correct predictions made by a model. It is calculated by dividing the number of correct predictions by the total number of predictions. The accuracy score ranges from 0 to 1, where 1 represents a perfect model and 0 represents a model that makes only incorrect predictions.

The formula for accuracy is:

Accuracy = (*True Positive* + *True Negative*) / (*True Positive* + *True Negative* + *False Positive* + *False Negative*)

True Positive (*TP*) represents the number of observations that are correctly predicted as positive. *True Negative* (*TN*) represents the number of observations that are correctly predicted as negative. *False Positive* (*FP*) represents the number of observations that are incorrectly predicted as positive. *False Negative* (*FN*) represents the number of observations that are incorrectly predicted as negative (Brett, 2015; Alpaydin, 2010).

Keyword: Precision Score

Precision is a commonly used evaluation metric in machine learning that measures the proportion of true positive predictions made by a model among all positive predictions. It is calculated by dividing the number of true positive predictions by the total number of positive predictions made by the model. The precision score ranges from 0 to 1, where 1 represents a perfect precision (all positive predictions are true) and 0 represents a model that only makes incorrect positive predictions.

The formula for precision is:

Precision = *True Positive* / (*True Positive* + *False Positive*)

True Positive (*TP*) represents the number of observations that are correctly predicted as positive. *False Positive* (*FP*) represents the number of observations that are incorrectly predicted as positive.

Precision is particularly useful when the cost of false positives is high or when the goal is to minimize the number of false alarms. For example, in a medical diagnosis system, a high

precision score is desired because a false positive can lead to further unnecessary tests or treatments. In contrast to accuracy, precision is not affected by class imbalance, and it's less sensitive to the threshold of the classifier. However, precision alone is not enough to evaluate the model's performance, particularly when the goal is to maximize recall, in such cases, the F1-Score can be used as it's a balance between precision and recall (Brett, 2015; Alpaydin, 2010).

Keyword: *Recall Score*

Recall is another commonly used evaluation metric in machine learning, it measures the proportion of true positive predictions made by a model among all actual positive observations. It is calculated by dividing the number of true positive predictions by the total number of actual positive observations. The recall score ranges from 0 to 1, where 1 represents a perfect recall (all actual positive observations are predicted as positive) and 0 represents a model that fails to predict any actual positive observations as positive.

The formula for recall is:

Recall = *True Positive* / (*True Positive* + *False Negative*)

True Positive (*TP*) represents the number of observations that are correctly predicted as positive. *False Negative* (*FN*) represents the number of observations that are incorrectly predicted as negative (Brett, 2015; Guyon and Elisseeff, 2003; Kuhn and K. Johnson, 2013).

Keyword: *F1 Score*

F1 score is a commonly used evaluation metric in machine learning that is a combination of precision and recall. It is the harmonic mean of precision and recall, and it ranges from 0 to 1, where 1 represents a perfect score and 0 represents a model that makes only incorrect predictions. The F1 score is useful when you want to balance precision and recall.

The formula for F1 score is:

F1 score = 2 * (*Precision* * *Recall*) / (*Precision* + *Recall*)

F1 score gives equal weight to precision and recall, which means that it will be high if both precision and recall are high, and it will be low if either precision or recall is low. It's particularly useful when the data set is imbalanced and in cases when the goal is to maximize both precision and recall.

The F1 score is a good metric to use when the goal is to balance precision and recall, however, it can be affected by the threshold of the classifier, which means that the F1 score can be increased or decreased by changing the threshold. To overcome this limitation, a combination of precision, recall, and F1 score with other evaluation metrics such as ROC-AUC can be used to evaluate the performance of the model (Brett, 2015; Guyon and Elisseeff, 2003; Kuhn and K. Johnson, 2013).

Keyword: AUC-ROC

AUC-ROC (Area Under the Receiver Operating Characteristic curve) is a commonly used evaluation metric in machine learning, particularly in binary classification problems. It is a measure of a classifier's performance, which represents the ability of a model to distinguish between two classes, i.e., positive and negative. It ranges from 0 to 1, where 1 represents a perfect score and 0 represents a model that makes random predictions.

The AUC-ROC score is calculated by finding the area under the ROC curve, which is a plot of the TPR against the FPR at different classification thresholds. A model with a higher AUC-ROC score has a better ability to distinguish between the positive and negative classes, and therefore it's considered a better model. AUC-ROC is a robust evaluation metric, and it's not affected by the threshold of the classifier, it's useful when the goal is to maximize the true positives while minimizing the false positives. It's worth noting that AUC-ROC is a good evaluation metric when the data is imbalanced, and it's not affected by the threshold of the classifier, it's also useful when the goal is to maximize the true positives while minimizing the false positives (Brett, 2015; Guyon and Elisseeff, 2003; Kuhn and K. Johnson, 2013).

In the table below, the advantages and disadvantages of the evaluation metrics are summarised.

2.3. Summary

This section provided a literature review on customer churn prediction, supervised machine learning (SML), and various algorithms used for SML, including decision trees and random forest classification. The review also covers the data cleaning and pre-processing steps involved, such as data splitting, exploratory data analysis, extraction, encoding, missing values, outliers, scaling, and finally, the evaluation metrics used to assess the model's performance, including confusion matrix and accuracy score. In conclusion, this is a comprehensive literature review on the topic of customer churn prediction and supervised machine learning.

Table 2Comparison of the evaluation metrics (Brett, 2015; Guyon and Elisseeff, 2003; Kuhn and K. Johnson, 2013; Alpaydin, 2010; Jain and Mahajan, 2016)

Evaluation metric	Advantage	Disadvantage
Confusion Matrix	Provides a clear and concise summary of a model's performance	Only useful for binary classification problems
Accuracy	Easy to understand and widely used as a baseline	Can be affected by class imbalance
Precision	Useful when the cost of false positives is high	Can be affected by class imbalance
Recall	Useful when the cost of false negatives is high	Can be affected by the threshold of the classifier
F1-Score	A balance between precision and recall	Can be affected by the threshold of the classifier
AUC-ROC	A robust evaluation metric not affected by the threshold of the classifier	Not useful for multi-class classification problems

3. Design and Methods

3.1. Introduction

In this chapter, we will outline the methodology adopted for implementing machine learning algorithms in our project. Firstly, we will discuss the approach taken for the development process. Secondly, we will examine each method described in the previous literature review, highlighting its advantages and limitations. Finally, based on the analysis of options and limitations, we will draw conclusions and specifications for the design, which will serve as a guide for the concept solution, design, development, and testing phases.

The design and methods for customer churn prediction require careful consideration of various factors, including feature selection, model selection, and evaluation metrics. This project's plan is to provide a comprehensive overview of the current state-of-the-art in the field of customer churn prediction, with a focus on the design and methods used to effectively predict customer churn.

3.2. Methodology

Waterfall: a linear, sequential approach where each phase of the project must be completed before moving on to the next. Here's an example of a waterfall methodology.

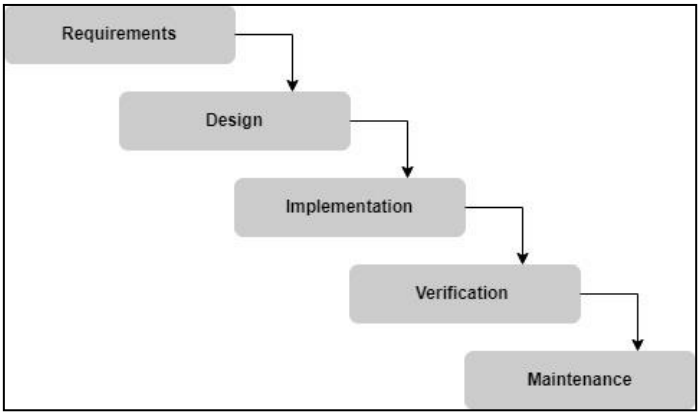

Figure 6. Waterfall diagram of the methodology

Agile: a flexible, iterative approach where requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams.

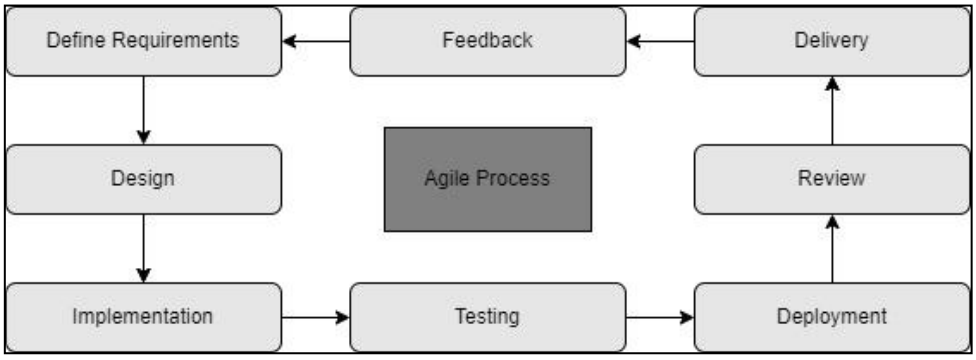

Figure 7. Agile diagram of the methodology

3.3. Limitations and Options

The methods proposed by the various authors in the literature were critically reviewed and summarised in the following table.

Table 3. Limits and options of the project themes

Theme	Limitations	Options
Supervised Machine Learning	Data quality and size	Size: The size of a dataset can have a significant impact on the performance of a machine learning model. In general, larger datasets are more likely to result in models with better performance compared to smaller datasets, although the relationship between dataset size and performance is not always straightforward. Quality: More diverse and inclusive dataset, e.g., data from different sources, companies, and countries would enhance the performance of the model.
Supervised machine learning algorithm	Overfitting	Hyperparameters are parameters that control the behaviour of machine learning models, and hyperparameter tuning is the process of finding the optimal values for these parameters. Hyperparameter tuning can be used to improve the performance of models.

3.4. Design Specification/User Requirements

3.4.1. Data cleaning and pre-processing

- Splitting the data into training, validation, and test sets using train_test_split from scikit-learn library
- remove duplicates using df.drop_duplicates(), this function removes all the duplicates in the data frame and keeps the first occurrence.
- handle missing values using df.isna().sum(), these functions count the number of missing values in each column and df.dropna() removes the missing values from the data frame
- Handle outliers using df[df < (Q1 - 1.5 * IQR)].fillna(value=Q1 - 1.5 * IQR): Remove outliers based on the IQR (interquartile range) method.
- Transform the data into a suitable format for analysis (e.g., Encoding) using pd.get_dummies(df[column]) Creates a new data frame with a binary column for each unique category in the specified column
- Creating new features from the raw data to capture important patterns and relationships. (Shapley values in Python Pandas can be calculated using the shap library, use the shap.Explainer class to calculate Shapley values.)

3.4.2. Model Development

- Selecting the appropriate machine learning algorithms to build the model, such as decision trees and random forests (the DecisionTreeClassifier class is used to train a decision tree, and the RandomForestClassifier class is used to train a random forest model on the data)
- Tuning the hyperparameters of the models to optimize the performance of the model
- Using the training data and

3.4.3. Testing and Evaluation

- Evaluating the performance of the model using the validation data
- Evaluating the performance of the final models using the test data
- comparing the performance of different models

3.4.4. Deployment

- Deploying the best performing model in a production environment and monitoring its performance

3.5. Concept Solution

The proposed solution to the design specification for this project is outlined in the following diagram. The diagram presents a conceptual representation of the solution.

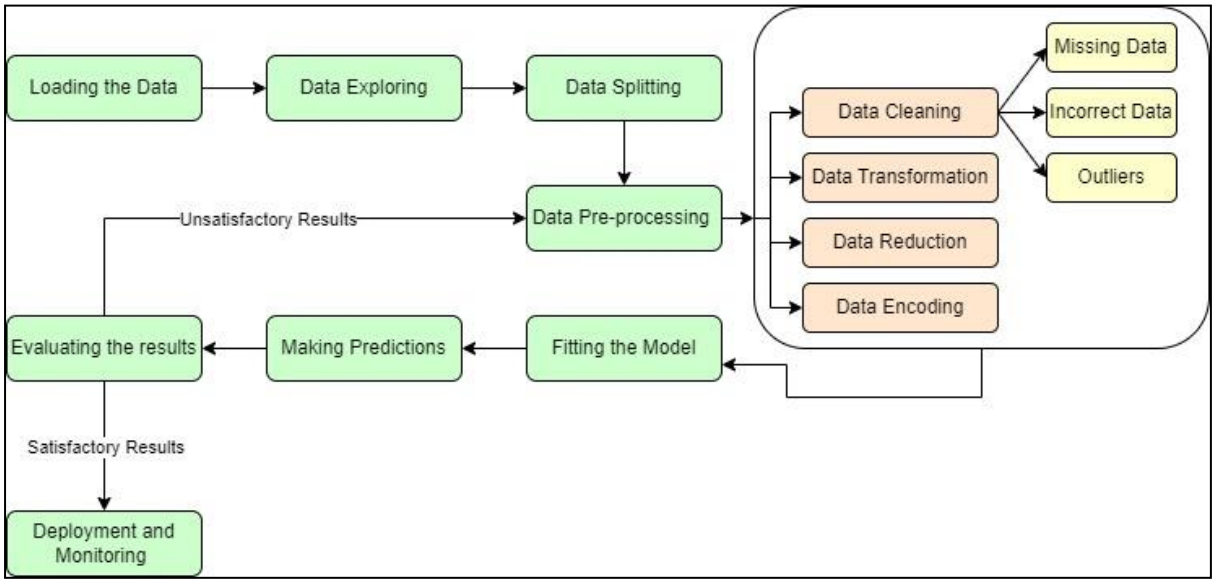

Figure 8. Conceptual representation of the proposed solution

3.6. Design and Development

The first step is to import the required libraries, namely `Pandas, Shap, numpy as np, seaborn as sns, matplotlib.pyplot as plt, sklearn as skl,` using *import* keyword.

The next step is to extract the data set from where it is stored (in the current case, GoogleDrive). The data is converted to *numpy* array, and then split into the data into training, validation, and testing subsets using *train_test_split* function from scikit-learn Python library.

The *random_state* parameter is used to set the random seed for the *train_test_split* function, setting its value to 42 as the common value for this parameter, even though any other integer works as well.

Then, the data features and target are visualised as scatter, bar, line, etc. plots for exploratory data analysis, as well as heat map correlation plot using Pearson method. This allows to explore

and quantify the correlation between features and target which, in turn, can be useful for the model development process.

The next steps are checking the missing values to impute if there is any, handling the outliers, normalising the data, and dropping the low correlation columns by Shapley values, using the procedures discussed in the previous sections, for better performance of the model.

Then, using the fitting function, i.e. "*.fit()"*, to train the model, followed by validating the model and, final, testing the model.

For evaluation purposes, the model results are shown via confusion matrix which enables calculation of the accuracy, precision, recall, and the F1 scores.

In the case of low performance of the model, or other issues such as over fitting, multiple iterations of the above procedure can be carried out to achieve the desired performance.

3.7. Testing and Testing Strategies

In machine learning, a confusion matrix is a commonly used tool for evaluating the performance of a binary or multiclass classification algorithm as discussed in literature review in 2.2.6. It provides a visual representation of the accuracy of the algorithm, as well as its ability to correctly predict or classify each class. The matrix consists of rows representing actual classifications and columns representing predicted classifications, and cells within the matrix display the number of observations that were classified as a particular class in the actual data and the predicted data. From the values in the matrix, several evaluation metrics can be calculated, including accuracy, precision, recall, and the F1-score, which the calculation of all have been discussed in the literature review. The confusion matrix is a useful tool for comparing the performance of different algorithms, as well as for tuning a single algorithm to improve its performance. The following figure shows a generic confusion matrix where the correlation between variables is displayed in each cell.

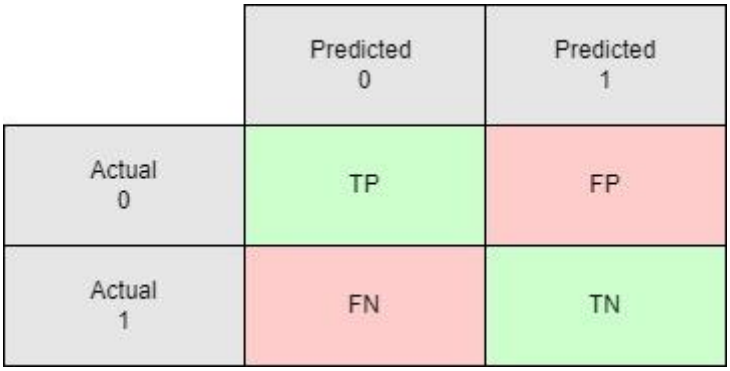

Figure 9. A generic confusion matrix showing the correlation between two actual and predicted values

3.8. Summary

This report focuses on the aim and objectives, literature search methodology, literature review, design and methods, and testing and evaluation of the individual honours project for course CMP6200. The aim of the project is to develop a customer churn prediction model using supervised machine learning algorithms and evaluate the model performance using appropriate metrics. The literature review encompasses customer churn prediction, supervised machine learning, supervised machine learning algorithms, data cleaning and pre-processing, and evaluation metrics. The design and methods section covers the methodology, limitations and options, design specification/user requirements, concept solution, design and development, and testing and testing strategies. Finally, the appendix provides a Gantt chart for the project timeline and a list of references.

Appendix:

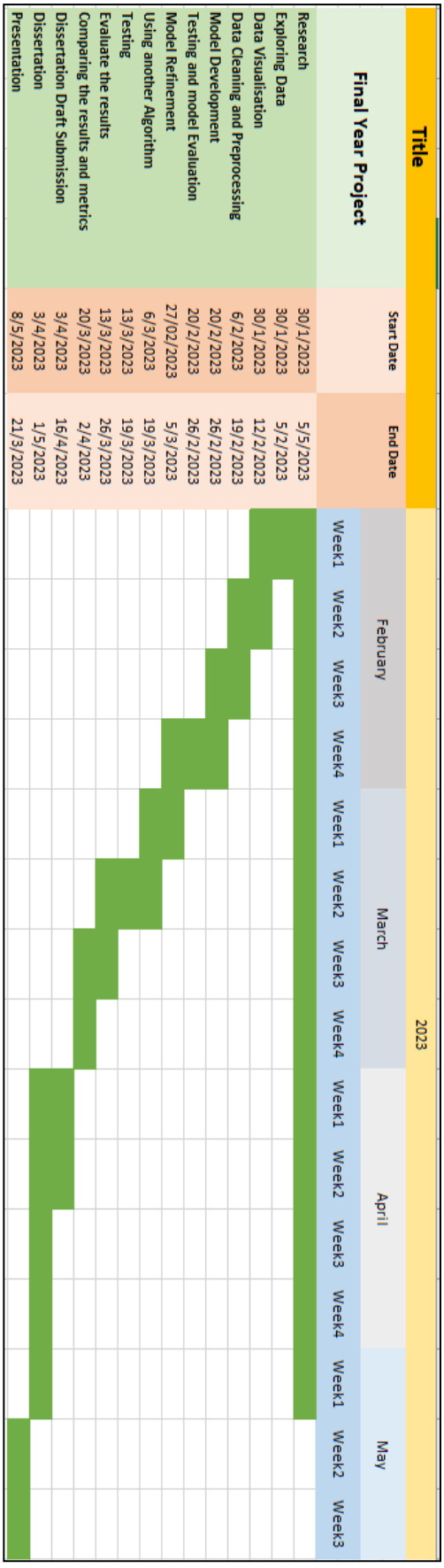

Figure 10. Gantt chart of the project

References

"The Click Reader. (2021). Decision Tree Regression Explained with Implementation in Python. Medium. Retrieved from https://medium.com/@theclickreader/decision-tree-regression-explained-with-implementation-in-python-1e6e48aa7a47".

Alpaydin, E. (2010). Introduction to Machine Learning (2nd ed.). Cambridge, MA: MIT Press.

Barocas, S., Hardt, M. and Narayanan, A., (2017). Fairness in machine learning. Nips tutorial, 1, p.2. [Accessed 13 Jan 2023].

Bhargava, A., & Agrawal, R. (2018). Data Preprocessing for Machine Learning. Springer.

Blattberg, R. C., & Deighton, J. (2001). The impact of customer retention on profitability: A study of the telecommunications industry. Journal of Interactive Marketing, 16(2), 2-21.

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Brett, B. (2015). Machine Learning with R. Packt Publishing.

Brownlee, J. (2016). Why you should use a fixed seed when evaluating machine learning algorithms. [Webpage]. https://machinelearningmastery.com/why-you-should-use-a-fixed-seed-when-evaluating-machine-learning-algorithms/.

Brownlee, J. (2019). Train-Test Split for Evaluating Machine Learning Algorithms. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/.

Çelik, O. and Osmanoglu, U.O., 2019. Comparing to techniques used in customer churn analysis. *Journal of Multidisciplinary Developments*, *4*(1), pp.30-38.

Chan, J.Y.L., Leow, S.M.H., Bea, K.T., Cheng, W.K., Phoong, S.W., Hong, Z.W. and Chen, Y.L., (2022). Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review. Mathematics, 10(8), p.1283.

Charbuty, B. and Abdulazeez, A., 2021. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), pp.20-28.

Churn, S. (2003). Predicting customer churn: An analysis of the telecommunications industry. Journal of Management Information Systems, 20(1), 191-215.

Dalvi, P.K., Khandge, S.K., Deomore, A., Bankar, A. and Kanade, V.A., 2016, March. Analysis of customer churn prediction in telecom industry using decision trees and logistic regression. In *2016 symposium on colossal data analysis and networking (CDAN)* (pp. 1-4). IEEE.

De Ville, B., (2013). Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics, 5(6), pp.448-455.

Garg, A. and Tai, K., 2013. Comparison of statistical and machine learning methods in modelling of data with multicollinearity. International Journal of Modelling, Identification and Control, 18(4), pp.295-312.

Ghorbani, A. and Zou, J., 2019, May. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning* (pp. 2242-2251). PMLR.

Gnat, S. (2021). Impact of categorical variables encoding on property mass valuation. Procedia Computer Science, 192, 3542-3550

Google Developers. (2021, May 10). Machine Learning Crash Course: Validation. [online]. Available at: https://developers.google.com/machine-learning/crash-course/validation/another-partition.

Google Developers. (n.d.). Introduction to Machine Learning. [online] Available at: https://developers.google.com/machine-learning/crash-course/ml-intro.

Guyon, I., & Elisseeff, A. (2003). Evaluation Metrics for Machine Learning. JMLR.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

IBM. (n.d.). Supervised Learning [online]. Available at: https://www.ibm.com/topics/supervised-learning .

Jain, R., & Mahajan, A. (2016). Classification and Diagnosis of Medical Images. Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Karthikeyan, A., & Aravindhan, R. (2016). A beginner's guide to univariate & multivariate linear regression. Journal of Applied Sciences and Research, 12(1), 1-10.

Kelleher, A. D., & Bailey, B. (2015). Avoiding data leakage. Journal of Machine Learning Research, 16(1), 3133-3181.

Kuhn, K., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

Lalwani, P., Mishra, M.K., Chadha, J.S. and Sethi, P., 2022. Customer churn prediction system: a machine learning approach. *Computing*, pp.1-24.

Liu, Y., Wang, Y. and Zhang, J., 2012. New machine learning algorithm: Random forest. In *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3* (pp. 246-252). Springer Berlin Heidelberg.

Merrick, L. and Taly, A., 2020. The explanation game: Explaining machine learning models using shapley values. In *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4* (pp. 17-38). Springer International Publishing.

Mitchell, T. (1997). Machine Learning. McGraw-Hill.

Murphy, K.P., 2022. *Probabilistic machine learning: an introduction*. MIT press.

Raza, M. A., & Besar, M. H. B. (2017). A review of customer churn prediction in telecommunications industry. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(2), 1-10.

Reichheld, F. F., & Sasser, W. E. (1996). The economics of customer retention. Harvard Business Review, 74(1), 46-54.

Scikit-learn Development Team. (n.d.). sklearn.model_selection.train_test_split. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

Somvanshi, M., Chavan, P., Tambade, S. and Shinde, S.V., 2016, August. A review of machine learning techniques using decision tree and support vector machine. In *2016 international conference on computing communication control and automation (ICCUBEA)* (pp. 1-7). IEEE.

Srivastava, J. J., & Rangaswamy, S. (2002). Retention management: A framework for retaining customers. Journal of Marketing, 66(2), 33-52.

Tan, P., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining. Pearson Education.

Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G. and Chatzisavvas, K.C., 2015. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, *55*, pp.1-9.

Walker, B. C., & Johnson, W. D. (2003). The role of customer retention in the profitability of firms. International Journal of Service Industry Management, 14(3), 241-253.

Weng, J. (2021, October 12). Data splitting for model evaluation. [Webpage]. https://towardsdatascience.com/data-splitting-for-model-evaluation-d9545cd04a99.

Wilcox, D. S. (2002). A survey of techniques for the reduction of the customer churn rate in the telecommunications industry. Journal of Marketing Research, 39(2), 222-238.

Zaki, M. J., & O'Malley, D. J. (2007). Customer churn prediction using decision tree analysis. Journal of Data Mining and Knowledge Discovery, 18(1), 63-90.