

**Graduate Diploma in Information Technology**

**School of Information Technology**

**IT7510 – Capstone Project**

**Project Research Report**

**The influence of Weather and Climate factors on Influenza Hospitalizations in New Zealand: A Predictive Modelling Approach**

**Academic Supervisor**

Adrian Hargreaves

**Student Name**

Srinivas Baddam – 22002759

**Abstract**

Influenza is one of the deadliest infectious upper respiratory diseases which demands an ongoing investigation for the purpose of effective surveillance and early prevention. The recent epidemiological evidence indicates that certain weather conditions facilitate the transmission of Influenza and the virus survives longer in temperate regions. This research aims to find out whether the climate and weather factors could influence the rate of Influenza occurrence in New Zealand using a predictive modelling approach. The outcome of this research will benefit the district health boards (DHBs) and health policy makers of New Zealand to be prepared for any future Influenza pandemics. This research used two main datasets collected from Ministry of Health (MoH) and National Institute of Water and Atmospheric Research (NIWA). These datasets have been used to perform various statistical analysis and experiment a series of predictive analytics models to identify the trends and correlations that may exist between weather factors as well as the Influenza cases in different climatic zones within New Zealand.

It was found that there are statistically significant differences between the standardised rates of Influenza hospitalisations within DHBs located in the nine climatic zones examined in this research. It is also been identified that weather factors like temperature, solar radiation, and absolute humidity are significantly correlated to the rate of influenza hospitalisations in New Zealand. For example, it was observed that when the temperature rises, the number of cases tend to decrease and when the temperature falls the number of influenza cases tends to increase. After experimenting with five different predictive algorithms, this research was successful in constructing a final predictive model using Logistic Regression, which achieved an accuracy of 89.02%. This final artefact can predict the rate of Influenza hospitalizations if the future weather data is provided in the required format. Finally, this research also provides insights for future work by clearly presenting its limitations and recommendations.

Table of Contents

[1. Introduction 4](#_Toc65089690)

[2. Literature Review 5](#_Toc65089691)

[2.1. The relevance between predictive modelling and Influenza 5](#_Toc65089692)

[2.2. What is Predictive modelling? 5](#_Toc65089693)

[2.3. Process of building a predictive model 6](#_Toc65089694)

[2.4. Tools and applications of predictive modelling 6](#_Toc65089695)

[2.5. Classification vs Regression Predictive Modelling 7](#_Toc65089696)

[3. Theory Base 7](#_Toc65089697)

[3.1. Supervised Learning vs Unsupervised Learning 8](#_Toc65089698)

[3.2. Clustering 8](#_Toc65089699)

[3.3. ANOVA 8](#_Toc65089700)

[3.4. Logistic Regression 9](#_Toc65089701)

[3.5. KNN 10](#_Toc65089702)

[3.6. CART 10](#_Toc65089703)

[3.7. Random Forest 11](#_Toc65089704)

[3.8. Naïve Bayes 11](#_Toc65089705)

[3.9. R Packages used 12](#_Toc65089706)

[4. Research Design 14](#_Toc65089707)

[4.1. Methodology 14](#_Toc65089708)

[4.2. Dataset Preparation 15](#_Toc65089709)

[4.3. Computational Environment 17](#_Toc65089710)

[4.4. Model Evaluation 18](#_Toc65089711)

[4.4. Confusion Matrix 18](#_Toc65089712)

[5. Findings 20](#_Toc65089713)

[6. Conclusion 24](#_Toc65089714)

[7. Limitations 24](#_Toc65089715)

[8. Future Work 24](#_Toc65089716)

[9. References 25](#_Toc65089717)

[10. Appendix 28](#_Toc65089718)

# 1. Introduction

Influenza is one of the deadliest airborne upper-respiratory infections that infected around one third of the world population during 1918-1919. The Influenza virus is easily spread through people when they cough, sneeze, or touch infected surfaces. Majority of the symptoms include fever, aches, chills, and cough but often it could also lead to death without proper immunisations or hygiene practices (Barry, 2020). The Influenza pandemic first hit New Zealand between October and December 1918 leaving about 9000 New Zealanders dead and ever since there have been number of deaths recorded each year in New Zealand caused by Influenza (Rice, 2018). Many countries are still struggling with Influenza disease surveillance capabilities and undoubtedly New Zealand is one of them. The early detection of the disease and accessibility to medical services will greatly increase the chances of containing Influenza virus soon after it emerges (Alison Mack, 2005).

One of the best ways to identify the influenza occurrence is to understand the factors that could trigger the disease. Many researchers suggest that the incidence of Influenza is not only caused by the biological factors such as immunity or population susceptibility but also by climatic factors that can significantly influence the rate of infection. In particular, weather factors such as temperature, humidity and solar radiation influence the rate of Influenza hospitalisations (Elankumaran, 2015). Hence this project aims to find out how climate and weather factors impact on Influenza hospitalizations in New Zealand using a predictive modelling approach.

Two research questions will be addressed in this research report. One is to determine what correlations exist between the variables within a data set provided by the Ministry of Health (MoH) as well as the correlations that may exist between the variables within the weather data provided by NIWA (National Institute of Water and Atmospheric Research) and MoH data. The second is to determine which data mining algorithms produces the best predictive model for determining the incidence influenza hospitalizations in New Zealand based on the weather data. Both NIWA and MoH datasets include 10 years of data for the period 2005 – 2015. NIWA dataset contains data from weather stations in New Zealand whereas the MoH dataset contains the demographic data relating to each patient that was hospitalised due to Influenza virus.

An algorithm in data mining or machine learning is a set of heuristics and calculations which help to create a model from the data provided. When creating the model, the algorithm analyses the data provided to find specific types of patterns or trends. The algorithm uses the results of this analysis over many iterations and then finds the optimal parameters to create the final model (Deshpande, 2015). Though there are many algorithms and data mining techniques that are widely used in scientific research, this project will make use of few classification models such as KNN, CART, logistic regression, Random Forest, and Naïve Bayes. The way these classifications models are applied is explained in the following sections. In the next section, predictive modelling, which is also known as predictive analytics and its significance to predict the Influenza rates, are discussed.

# 2. Literature Review

## 2.1. The relevance between predictive modelling and Influenza

Reliable and accurate predictions on the dynamics of infectious diseases can be valuable to public health organisations that constantly thrive to decrease or prevent disease transmission (Evan L. Ray, 2018). A great variety of predictive models have been developed for different types of diseases using model structures, targets, and covariates for prediction. These models have both strengths and weaknesses depending on the prediction targets and the measures followed while building the models. But an accurate model is always useful to make reliable predictions of the timing and severity of the infectious diseases like Influenza as they could help the health authorities know how many vaccine doses to be produced and supplied at a specific point in time. The statisticians and computer scientists have also practiced combining predictions from different models which generates a final model called an ensemble model. An ensemble model combines the predictive characteristics from several models and can often achieve a higher predictive accuracy that a single model used in isolation.

According to Jessica Kent (2019), predictive analytical models were successful in forecasting the trends in Influenza outbreaks with higher accuracy compared to historic models. The researchers at University of Massachusetts compared the accuracy of 20 predictive models to a historic baseline average using seasonal data of Influenza for the period 2010 – 2017. They found that predictive models achieved greater accuracy than any other approach. They also mentioned that approximately 9 million to 35 million individuals in United States are being impacted by Influenza and out of them between 12000 to 56000 deaths are recorded annually. In addition to that, this research team had developed an ensemble model called ‘FluSight Network ensemble’ combining the forecasts from all 20 models. This model was the top performing real time predictive model for the year 2017-2018.

On the other hand, Moran et al. (2016) believes that the area of infectious disease forecasting is still at its infancy and more forecasting technologies are needed by making use of mathematical models. The health officials are still learning how to incorporate these forecasts into their decision making. Hence a strong collaboration between the public health policy makers and predictive modelers is necessary to ensure these forecasts are appropriately communicated to the broader health community to minimise the impact of respiratory diseases.

## 2.2. What is Predictive modelling?

Predictive modelling is a process of using historic data to create, process, and validate a model that could be used to forecast future outcomes. It is a tool used in predictive analytics to determine what might possibly happen in the future of a given scenario or question. A predictive model uses patterns within data to make predictions (Finlay, 2014). The process of predictive modelling could vary depending on the tools and techniques used. But in a generalised way, the objectives and time scales are decided first then people or resources involved are identified and finally the model is built and deployed.

## 2.3. Process of building a predictive model

Building predictive models is an iterative process where a model is generated from a data mining algorithm and then refined until it produces predictively accurate results. Below are the six main steps involved in constructing a model (Max Kuhn, 2013).

1. *Scope of the predictive model:*In this step, the research or business questions are formulated, and the desired outcomes are framed.
2. *Profile and explore the data:* Building a model is data intensive. So, the data that is needed is determined and its quality to produce the desired outcomes as well as the accessibility to this data is assessed.
3. *Gather, cleanse, and integrate the data:* The data then gets cleaned (i.e., remove outliers, replace missing values, and data formatting etc.) to ensure consistent and comprehensive set of data is ready for analysis.
4. *Build the predictive model:* The goal of this step is to include or rule out different variables according to their contribution to predictive accuracy of the model. The data is divided into training and testing partitions. The initial model is built using training data and then validated on the testing data to see if the results produced are accurate and viable. If the results are satisfying, the model will be finalised to use otherwise it will be rebuilt following the above steps.
5. *Generate predictions using the model:* The final model will be integrated into a business process and is used to generate predictions based on current data. These predictions help organisations move towards data driven decision making.
6. *Monitor the model and measure the results:* The results generated by the model are regularly monitored for their efficacy. Fine tuning and adjustments are made to the model as conditions evolve.

## 2.4. Tools and applications of predictive modelling

Predictive modelling is often used in weather forecasting, meteorology, and has many applications in business. Joseph M. Carew (2020) says that in advertising and marketing, modellers use historical data about web surfers, run it through algorithms to find out what type of products they might be interested in or likely to click on. In fraud detection, predictive modelling is used to identify the outliers in each data set which will direct towards fraudulent activity. CRM (Customer relationship management) uses predictive modelling to send messages to target customers who are more likely to make a purchase.

There are many open source and proprietary predictive analytical tools available in the market, but the question is to decide what fits well to achieve the desired outcome and costs associated with them. Some of the most commonly used open-source tools are the R statistics environment, RapidMiner, Weka Data Mining, NumPy, and SciPy etc. In the proprietary predictive analytics tools, SAS advanced analytics, IBM SPSS predictive analytics enterprise, Oracle Crystal Ball, and SAP predictive analytics are few the most used tools.

## 2.5. Classification vs Regression Predictive Modelling

There are two common forms of predictive modelling, one is classification, and another is regression. Depending on the type of problem to be addressed, the analyst can adopt one of these two methods (Brownie, 2019).

***Classification predictive modelling:***Classification refers to a predictive modelling problem in machine learning where a class label (response variable) is predicted for a given example of input data. For example, given recent user behaviour, classify a customer as likely to churn (i.e., use another business) or not (i.e. remaining loyal to the business). The class labels are usually string values (i.e., ‘churn or ‘not churn’) which is usually converted to numeric values before providing them to an algorithm (i.e., ‘not churn=0’, ‘churn=1’). There are different ways to calculate the predictive accuracy of a classification model. One popular approach is to calculate the classification accuracy using confusion matrix.

***Regression predictive modelling:***Regression predictive model is a task of approximating a mapping function from input variables to a continuous output variable. Here the output variable is an integer or floating value such as amounts and sizes. For example, the prediction the likely selling price of a house. Calculating RMSE (root mean squared error) is common approach to estimate the accuracy of regression models.

The existing literature suggests that there have been studies which used classification models such as GLM (Generalized Linear Model), Random Forest, and Naïve Bayes etc. to predict the infectious diseases such as Influenza, Malaria, Typhoid, and Dengue. There are also examples of Influenza being studied from the perspective of weather factors using predictive modelling techniques (Qinneng Xu, 2017). These studies proved to be helpful and produced accurate models to predict disease occurrence. It is been found that weather factors such as temperature, solar radiation, absolute humidity, and relative humidity have a statistically significantly influence in the occurrence of Influenza.

Hence, this project will try to test some of the well-known classification models using the data sets provided by MoH and NIWA weather centre and aim to derive the best possible model to predict the Influenza rates in New Zealand. The type of classification models used and the reasons behind choosing them are explained in the next section.

# 3. Theory Base

This research uses a variety of data mining algorithms to generate predictive models. As such, these algorithms, and the techniques they incorporate provide the theory base of the research undertaken. In this section, the algorithms used in this research are discussed.

The process of predictive modelling could be conducted using either supervised or unsupervised learning methods or the combination of both depending on the individual research objectives. This research aims to use the mix of these techniques due to the nature of the data and objectives of this research. Also, since this research is time bound, only 5 categories of algorithms were adopted, which are discussed below along with the Anova and clustering algorithms.

## 3.1. Supervised Learning vs Unsupervised Learning

In Supervised learning, a predictive model is trained by examples. This mechanism learns from historical data and applies this learning to current data to predict future events. In this approach you have input variable (X) and output variable (Y) and this algorithm trains a mapping function from input to the output. When Y = f(X) both Input (X) and output (Y) data help predict the future events. Supervised learning method comprises of several classification and regression models (Azlinah Mohamed, 2019). The difference between classification and regression modelling was previously explained above.

Unsupervised learning is used when data is unlabelled (i.e., there is no response variable). The purpose of this approach is to segment the data into meaningful partitions, often based on the shared characteristics discovered within the data. For example, one form of unsupervised learning is clustering, which aims to find natural groupings that could be used to segment customers based on shared characteristics. Another example is Anomaly detection, which identifies outliers within the dataset. This approach is often used to identify bank fraud and fraudulent insurance claims.

## 3.2. Clustering

Clustering is an important technique within data mining process and is often used to find natural groupings within datasets. It is a machine learning technique which enables data scientists to partition and segment the data into various groups based on their similarities. Subsets formed from large datasets are known as clusters. Clusters are formed by aggregating similar data items with shared characteristics. K-means and PAM clustering are commonly used clustering techniques in data science (Kassambara, 2017).

***K-Means Clustering:*** This is a popular clustering algorithm. The technique segments the data based on similarities. The analyst must specify number of clusters into which data should be grouped in number of specified clusters. K-means algorithm assigns each observation into one of the specified number of clusters. This algorithm does not work on categorical data, so the variables often need to be converted into numerical values when conducting this form of analysis. This algorithm does not work well with outliers. Choosing of right number of cluster groupings is very important in obtaining meaningful partitions and often relies on domain knowledge.

***PAM Clustering:*** This algorithm gets rid of problems faced in K-means clustering such as dealing with outliers and categorical data. Partitioning Around Medoids (PAM) uses Medoids instead of centroid in K means clustering. Instead of considering centre of the cluster as mean of the cluster, PAM clustering considers centroid as one of the observations in the cluster. This method is equally robust for detecting outliers and missing values. PAM clustering uses the pam function to perform partitions around the medoids (Lesmeister, 2015).

## 3.3. ANOVA

ANOVA is called as Analysis of Variance. The approach compares the means of multiple groups present in the dataset. ANOVA is referred as 1- way, 2-way or 3-way analysis which define how many factors are being examined in the model. Anova analysis finds the variance between the available groups as well as within the groups. This method is also called as a residual variance. The technique is used to determine whether statistically significant differences are present between groups of observations. The key factor behind the Anova analysis is to find out if the average mean variation between the groups is statistically significant compared to average mean variation within the available groups. Thus, this method evaluates the difference within and across the group means which tells us how different each obserbvation is from individual group. This method produces the F- statistic value (F- value) as ratio of Variance between groups / Variance within groups. If F value is less than 1 (F<1) then it indicates there is no significant differences between the means of groups which are being compared. And if the F value is greater that indicates there is significant difference present among the specified groups (Song, 2019).

Anova method can be performed using various functions. The most common function used are aov() and lm(). Anova is a type of linear model, hence lm() function is most popular to perform this analysis.

## 3.4. Logistic Regression

Logistic regression is a predictive modelling algorithm used to find the probability of success and failure events. Logistic regression is supervised learning method that predicts class membership (the category an observation belongs to – typically one of two categories). This algorithm is often applied to binary class classification which referred as binomial logistic regression, but it is also applicable to multinomial logistic regression, which can handle the classification problems with three or more classes. Logistic regression is quite popular classification algorithm, particularly within the medical community (Stephan Dreiseitl, 2002).

***Advantages***

* Logistic regression is easy to implement, interpret and efficient on training model.
* This model can work with both continuous and categorical variables.
* This model can easily extend to multinomial regression and natural probabilistic view of prediction class.
* It gives good accuracy for simple datasets and performs well when datasets are linearly separable.

***Disadvantages***

* Logistic regression leads to overfitting problem when the number of observations is less than number of features in dataset.
* Key constraint between logistic regressions is the assumption of linearity between the dependent and independent variables.
* It requires average or no multicollinearity among independent variables in the datasets.

## 3.5. KNN

KNN stands for K Nearest Neighbour. This is supervised Machine Learning algorithm which categorises the new data point into the target class, depending on the structures of its neighbouring data points. KNN is one of the simplest Machine learning algorithms which uses the labelled input dataset and predicts the output of that dataset points. This algorithm checks for similarity of data points which are its neighbour and then classifies that data points into the similar class.

KNN is non- parametric model unlike most other algorithms which means this algorithm does not make any assumptions about the given datasets. This makes KNN algorithm effective to handle the realistic data. This algorithm is also known as lazy algorithm because it remembers the training dataset instead of learning the discriminative functions from the training dataset. KNN model is applicable for both classification and regression problems (Sadegh Bafandeh Imandoust, 2013).

***Advantages***

* KNN algorithm is very easy to implement as only two parameters are required to build a model i.e., value of K and distance function (e.g., Euclidean, Hamming, Minkowski, Manhattan etc.)
* New data can be added to existing dataset seamlessly because KNN algorithm does not require training before making prediction and this will not impact the accuracy of algorithm.

***Disadvantages***

* As the dataset grows the efficiency of KNN algorithm declines.
* This algorithm does not support Curse of Dimensionality. As the number of variables increase, this model struggle to predict new data point output.
* KNN algorithm has no capability of dealing with missing values in the dataset.

## 3.6. CART

CART stands for Classification and Regression Trees. This algorithm is very popular and powerful predictive machine learning method which is used as both classification and regression modelling. R implementation of this algorithm is usually called as RPART package which stands for recursive partitioning and regression trees. Tidyverse, Caret and RPART are essential libraries required to build this algorithm.

CART algorithm of decision tree works by partitioning data on repeatedly into multiple sub-spaces, so that the output of each final terminal node is as heterogeneous as possible. This method of partitioning is usually called as recursive partitioning. The result is made up of set of rules which are used to predict the outcome of variables which can be either categorical variables for classification trees or continuous variable of regression trees (Tony Thomas, 2019). The structure of decision tree consists of three parts such as Root Node which performs the first split, Terminal Node which predict the outcome and Branches which are arrows connecting the nodes. The decision rules generated by CART algorithm are visualized by binary tree.

***advantages***

* Decision trees can handle both categorical and continuous data.
* This algorithm performs classification without demanding much computation.
* The output of decision tree provide strong suggestion of which fields are most important for classification or prediction.

***Disadvantages***

* The output of decision trees is less accurate for tasks whose goal is to predict the value for continuous attributes.
* The process of growing Decision trees is computationally expensive to train.

## 3.7. Random Forest

“The Wisdom of Crowd” is the idea behind building Random Forest algorithm in machine learning. This algorithm works by combining the predictions made by numerous decision trees of various depths. Each decision tree in random forest algorithm is trained on a subset of dataset which is called as a bootstrapped dataset. The left-out sample portions during building of each decision tree in the forest are referred to as OOB (Out of Bag) dataset. When random forest algorithm is used for classification with new dataset, the final prediction is made by considering majority of the predictions made by each discrete tree present in the random forest. On the other hand, when this algorithm used for regression with a new dataset, the final prediction is made by taking average of predictions made by each individual decision tree in the random forest (Hariom Tatsat, 2020).

***Advantages***

* Typically provides very good performance rate.
* Very less amount of tuning required as this provides remarkably good “Out of box”.
* Pre-processing of data is not required.
* Very robust to handle outliers.

***Disadvantages***

* Performance of the algorithm degrades when dataset is large.
* The outputs are less interpretable.

## 3.8. Naïve Bayes

Naïve Bayes is a supervised machine learning algorithm which is based on a simple probabilistic classification method based on Bayes theorem. Naïve Bayes algorithm considers each variable independently (i.e., assumes no dependencies exist between variables) with respect to all other variables, hence it is called as Naïve (Foster Provost, 2013). The output of the model depends on a set of independent variables that have no relation between any variables. The key principle behind this algorithm is to calculate the conditional probability which is probability of event occurring based on information about events that occurred in the past. This algorithm could be explained as below.

P (A|B) = P (B|A) P (A) / P (B)

P (A|B): Probability of ‘A’ event occurring in the given event ‘B’.

P (A): Probability occurrence of event ‘A’

P (B): Probability occurrence of event ‘B’

P (B|A): Probability of ‘B’ event occurring in the given event ‘A’.

***Advantages***

* Easy and fast to implement.
* Provides outstanding accuracy when holds independent assumption.
* Suitable for large datasets.
* Resilient to noisy data and outliers.
* No risk of overfitting.
* Provide intuitive reports such as mean, standard deviation and probability calculation of each class.

***Disadvantages***

* It is a classification method and does not suitable to perform regression.
* The real-world data variables are often dependant on each other which could be an issue for this algorithm to treat them as independent variables.

## 3.9. R Packages used

In this research, the statistical programming environment called R was used to conduct the analysis and generate predictive models. It consists of packages that contain related functions. To date, approximately 15,000 packages have developed. Below, the packages used in this research are briefly outlined.

|  |  |
| --- | --- |
| **Package Name** | **Description** |
| dplyr | It is a new package that provides a set of tools for efficiently manipulating data frames in R (dplyr, n.d.) |
| tidyverse | Provides an underlying design, philosophy, grammar, and data structures (R packages for data science, n.d.) |
| lubridate | Helps to work with dates and times. Enables parsing of date and time data (lubridate, n.d.). |
| ggpubr | Provides functions for creating and customizing ggplot2 based publication ready plots (ggpubr, n.d.) |
| rstatix | Consists of a simple pipe friendly framework to perform statistical tests like Anova and t-test (rstatix, n.d.) |
| Boruta | This algorithm helps to capture all the important variables in the dataset with respect to an outcome variable (Mazzanti, 2020) |
| Cluster | This package is used to identify groups of similar objects in the datasets that are multivariate (Types of clustering methods: Overview and quickstart R code, n.d.) |
| factoextra | Provides functions to extract and visualise the outputs delivered through multivariate data analysis. It helps to simply some cluster analysis steps (factoextra: Extract and Visualize the Results of Multivariate Data Analyses, n.d.) |
| Islr | This package provides the functions as well as data to fit the logistic regression model (ISLR, n.d.) |
| forecast | Consists of methods and tools for displaying and analysing univariate time series forecasts (Hyndman, 2020). |
| caret | Stands for Classification And REgression Training. Helps to streamline the model training process for classification and regression problems (A Short Introduction to the caret Package) |
| FNN | Known as Fast Nearest Neighbour algorithm. Helps to implement kd-tree, K nearest neighbour and regression (Fast Nearest Neighbor Search Algorithms and Applications, n.d.) |
| proc | Provides tools for smoothing, comparing, and visualising the receiver operating characteristic (ROC) curves (Robin, 2020) |
| mlbench | Includes a collection of artificial and real-world benchmark problems such as problems with disjunct classes (Package ‘mlbench’) |
| party | Provides a recusrsive partitioning laboratory which includes various tools for building tree-based regression and classification models (Torsten Hothorn, 2006) |
| randomforest | Implements Breimn’s random forest algorithm for both classification and regression models. It is also used in unsupervised learning to assess proximities among data points (Classification And Regression With Random Forest, n.d.) |
| naivebayes | This algorithm calculates the conditional a-posterior probabilities for categorical variables where the independent predictor variable uses the bayes rule (naivebayes, n.d.) |
| ggplot2 | This package uses the data provided, maps the variables, follows the primitives provided and finally produces graphical representations of the data (ggplot2, n.d.) |
| psych | Functions of this package are primarily used for multivariate analysis and scale construction using cluster analysis, factor analysis, and principal component analysis etc. (Revelle, 2020) |
| reactable | It is based on the ‘React Table’ java script library. Helps to create interactive tables and provides HTML widget that could be used in R Markdown documents (reactable). |

Table 1: R Packages used to carry out the predictive modelling process.

# 4. Research Design

## 4.1. Methodology

This project uses CRISP-DM (Cross-industry standard process for data mining) method to guide the research conducted. The main reason to follow this method is to take advantage of its structured approach to conducting data mining project. It is well proven and robust approach which was incorporated into IBM Statistical Package for the Social Sciences (SPSS) modeller. It is useful and flexible when applying analytics on complex business issues. It facilitates many tasks to be performed in different orders, enabling backtracking to previous tasks, and iterating the actions performed. CRISP-DM could also produce reports almost at any phase of the project which strengthens the client engagement (Keith McCormick, 2017).

CRISP-DM contains 6 phases in its process as explained below.

1. *Business Understanding:* This method focuses to understand the business objectives initially and converts that understanding into data mining problem.
2. *Data Understanding:* At this stage, necessary data collection is completed then proceed with identifying the data quality issues and discover insights into the data. These insights lead to hidden subsets in the data which help to form hypotheses.
3. *Data Preparation:* A final dataset is prepared from the initial raw data which is then used for data analysis.
4. *Modelling:* This phase goes through selecting appropriate modelling techniques to apply on the data. The selected models (i.e., Logistic regression, KNN etc.) will be built using certain parameters and assessed for their efficiency.
5. *Evaluation:* In this phase the model results or predictions are assessed to see if they meet the business objectives. The evaluation phase also involves in analysing any other data mining results that are generated during the process. Finally, one model gets finalised that meet all the requirements.
6. *Deployment:* A code gets deployed representing the model into an application. This will categorize the new unseen data as it comes through and creates a mechanism to use the new information and produce the predictions to resolve the original problem.

[William Vorhies](https://www.datasciencecentral.com/profile/WilliamVorhies) (2016) says that CRISP-DM ensures quality in the data science field. This methodology not only helps traditional data analytics projects but also supports more advanced data related projects such as text, image, language processing, deep learning, and artificial intelligence projects. The main disadvantage of this method is it does not perform project management related activities.

CRISP-DM is helpful especially when the project is iterative. In the world of data science, the data is always changing, affecting the model capability even before deploying it. In such cases, this method allows going back to the starting phase to do the necessary amendments. The iterative nature of CRISP-DM also helps in long-term strategy for any business as they could build quick and small models initially to see if they benefit them then continue repeat the cycle depending on what the business is trying to achieve in long run. Since this project involves in data collection, preparation, and modelling, CRISP-DM method will be suitable allowing us to move around the iterations as needed until the final model is developed.

## 4.2. Dataset Preparation

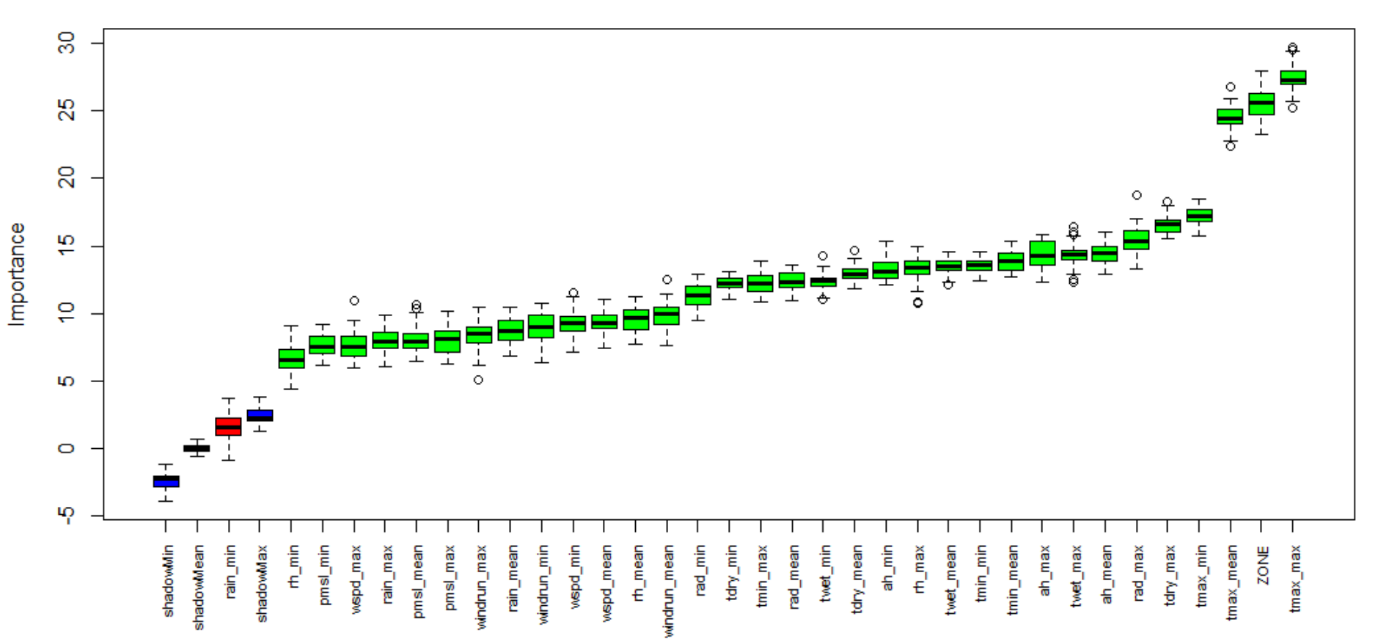
As mentioned in the above sections, two main datasets have been used to undertake this research. The first one is MoH dataset which was collected from Ministry of Health New Zealand. This dataset comprises of the data related to Influenza cases for the period of 2005 to 2015 in New Zealand. It contains details about the age, gender, ethnicity, diagnosis type of the people when they were hospitalized under different district health boards (DHBs) along with the date of admission. This dataset was integrated with the NIWA daily weather station data and grouped into one of nine climatic zones, depending on the geographical location of the DHB where each patient resides. In addition, census data recording population details for each of these zones was used to calculate the number of cases per every 100,000 people per week by zone. This preparation of the data resulted with the information about the number of Influenza cases in every hundred thousand people occurred per week in 9 different zones of New Zealand for the period 2005 – 2015. Using age and ethnicity details within the MoH dataset, the hospitalisation rates per 100,000 for each climatic zone were standardised to allow valid comparisons using ANOVA. Using this approach, the hospitalisation rate over a 10-year period within each climatic zone could be assessed to determine whether any statistically significant differences exist. This was an important first step, since generating accurate predictive models is reliant on the discovery of natural groupings within the data.

**Variables within the MoH Dataset**

Table

Description automatically generated

The second dataset which was obtained from National Institute of Water and Atmospheric Research (NIWA) contains weather information for the period 2005 – 2015. This data was used to find out if the weather factors had an impact on Influenza cases in New Zealand and to find the correlations between NIWA and MoH datasets. To perform this analysis, the processed information from both NIWA and MoH datasets was combined before applying any algorithms on it. Initially the NIWA weather dataset had 38 variables in total which relate to different weather conditions. This research made use of Boruta algorithm to select the most important weather variables out of these 38 which could better predict the Influenza cases. The Boruta algorithm is a feature selection wrapper which works with any classification method using random forest technique. Boruta performs a top-down search for the important variables by comparing their original attributes importance with the importance that could be achieved randomly and by estimating their permuted copies, then eliminating the variables that are not important (Boruta, n.d.). Below is the result obtained when Boruta algorithm was applied on the initial NIWA dataset.



The top six important variables that are resulted from Boruta analysis are tmax\_max, ah\_max, rh\_max, rain\_mean, Zone, and rad\_max. Hence this research only made use of the data supplied by these six variables to build the predictive models and the rest of the variables were neglected.

**Final Variables within the NIWA Dataset**

Table

Description automatically generated

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| AGE\_DSCH | Age of the person when hospitalised |
| GENDER | Gender of the person hospitalised |
| ETHNICGP | Ethnic group of the person hospitalised |
| DHBDOM | District Health Board domicile represented with code number. |
| END\_TYPE | End status of the person hospitalised |
| EVSTDATE | Event Start date/Date of hospitalization |
| Diag1 | Type of diagnosis represented with code. acute upper respiratory infections (J00-J06), influenza and pneumonia (J10-J18), lower respiratory infections (J20-J22). |
| Short\_stay\_ED\_Flag | Y if Length of stay <2 days and Specialty in M05-M08, else N |
| ZONE | Zone number allocated from 1 to 9 to the grouped DHBs. |
| Category | 1 = Low Influenza cases, 2 = Normal cases and 3 = High cases |
| rain\_mean | The mean value of the amount of rain |
| rh\_max | Maximum relative humidity |
| ah\_max | Maximum absolute humidity |
| tmax\_max | Maximum temperature |
| rad\_max | Maximum radiation |

Table 2: Description of Data Variables used in this research.

## 4.3. Computational Environment

This research made use of an open-source data mining software called R. This is a free data science software used for business and scientific research. This software consists of many packages and libraries to perform data science related tasks and can run on different platforms (i.e., Windows, MacOS, and UNIX platforms). It is very similar to S language and environment with some differences. R facilitates data manipulation, calculation, and graphical display on its platform. It also provides a wide variety of statistical and graphical techniques such as linear or nonlinear modelling, classification, and clustering etc. One of the main strengths of R is the ability to produce well designed publication quality plots including mathematical formulas and symbols as needed. Microsoft Excel was also used to perform certain data preparation tasks. It is a powerful tool developed by Microsoft to learn patterns in data, pre-processing the data using different formulas, and helps to visualise the data. (Microsoft Excel, n.d.).

## 4.4. Model Evaluation

All the predictive models that have been used in this research were evaluated using a function called ‘Confusion Matrix’ supplied by the R library known as Caret. The dataset was initially divided into training data (60% of the original dataset) and testing data (40% of the original dataset) on R platform. All the five models were built on the training data and then tested on the unseen testing data to find the accuracy of the models. The confusion matrix function can generate the accuracy results for every model. These results then be analysed to finalise one of the models with the highest accuracy to make predictions on Influenza occurrence.

## 4.4. Confusion Matrix

In predictive analytics, confusion matrix is one of the key functions to measure the prediction results produced by classification algorithms. It uses a function called confusion Matrix () supplied by caret package in R. It generates calculations regarding the model’s capability of classifying predictions into right or wrong categories (Chopra, 2020). The classified predictions are usually presented in a table format as shown in figure 1.

Figure 1: Example of Confusion matrix table (Chopra, 2020)

* ***TP (True Positives):*** Predicted as positive values and it is true.
* ***TN (True Negatives):*** Predicted as negative values and it is true.
* ***FN (False Negatives):*** Predicted as negative values but it is false.
* ***FP (False positives):*** Predicted as positive values but it is false.

In this research positive records ‘1’ refer to the occurrence of high Influenza cases and ‘0’ refers to the occurrence low Influenza cases depending on the changing weather conditions Once the table is produced, the accuracy is calculated using the below formula. The accuracy indicates how often the classified prediction is correct (Narkhede, 2018).

Accuracy = (TP + TN)/(TP+TN+FP+FN)

The total number of actual positives (P) = TP + FN

The total number of actual negatives (N) = FP + TN

The confusion matrix also produces some other metrics which were also considered to evaluate the models in this research. Those results are,

* ***Sensitivity:*** Calculated as number of correct positive predictions divided by the total number of positive predictions (TP/P). This result also indicates how effective the model is to identify the positive predictions (Alpaydin, 2010).
* ***Specificity:*** Calculated as number of correct negative predictions divided by the number of total negative predictions (TN/N). This score shows the model’s capability to negative predictions as negatives.
* ***Positive Prediction value:*** it is calculated as the number of correct positive predictions divided by the total number of positive predictions (TP/(TP+FP). It is also called as precision.
* ***Negative Prediction value:*** It gives the probability that a predicted negative is a true negative and is calculated with the formula TN/(TN+FN)
* ***P-value:*** It indicates the statistical significance of the model results. Usually if the p-value is less than 0.05, it is understood as the model is statistically significant or else the model results are not reliable statistically.
* ***Balanced accuracy:*** it represents how good a binary classifier is. It is helpful especially when one or more classes appear more than the other. This usually happen in the case of anomaly detection and finding the presence of a disease.it is calculated as (sensitivity + specificity) / 2 (What is balanced accuracy?, 2020).

# 5. Findings

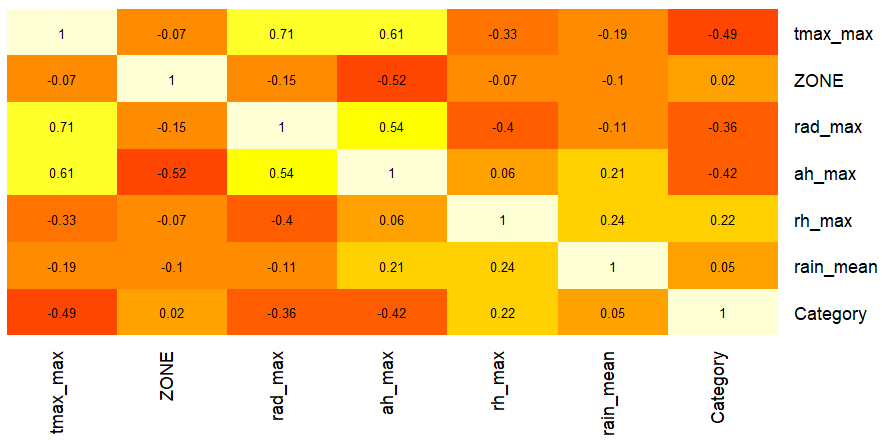
As this research aims at finding correlations between the weather factors and the influenza cases in New Zealand, some of the key correlations identified during the data analysis are shown in figure 2.

Figure 2: Correlations matrix between weather variables and Influenza cases

It is found that the maximum temperature (tmax\_max), solar radiation (rad\_max), absolute humidity (ah\_max), and relative humidity (rh\_max) are correlated to the occurrence of Influenza cases (Category) in New Zealand. This means these weather variables can influence the rate of influenza cases as well as could efficiently contribute to build predictive model to make future predictions on Influenza cases. Also, between the weather variables, it is been noticed that radiation, absolute humidity, and maximum temperature are correlated strongly to each other.

As mentioned in the above sections, this research also made use of Anova and clustering algorithms to find out if there are any statistically significant differences between the groups (Zones 1-9) in terms of number of Influenza cases as well as any associated distinct patterns within the data (i.e., High, and Low influenza cases). Figure 3 and 4 show the results produced by these algorithms.

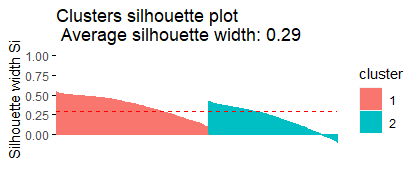
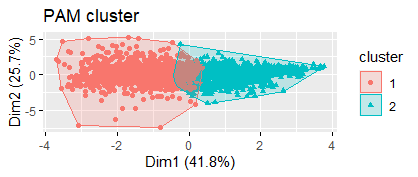
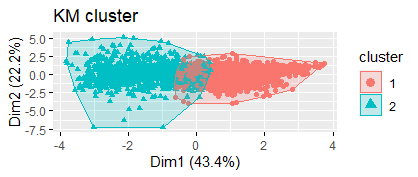
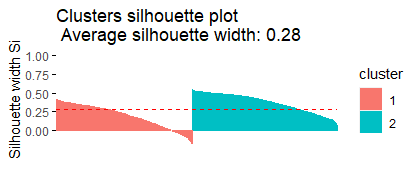
Text

Description automatically generated with low confidence

Figure 3: Anova test results obtained from R.

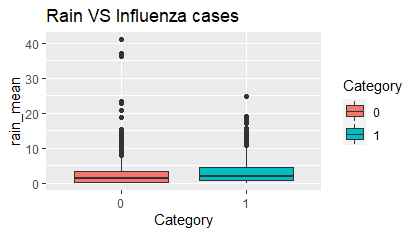
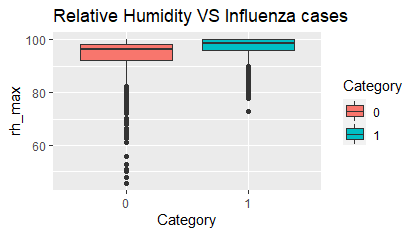
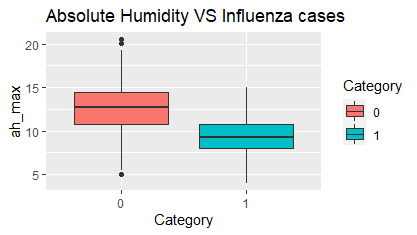
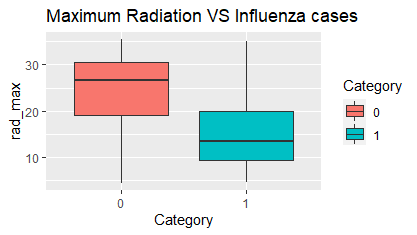
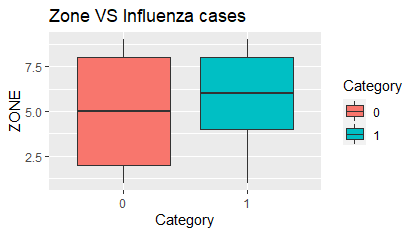
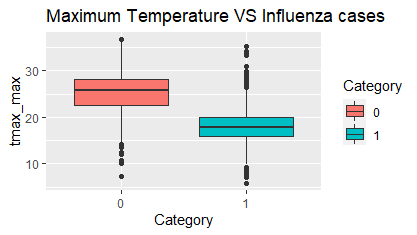
Anova test clearly identified that there are significant differences between the zones and the number of cases in each zone are different to each other. This is supported by the ges (generalised eta square) measure from Anova test resulting to 0.52 (as shown in figure 3) which means there are statistically significant differences between the standardised hospitalisation rates in each climatic zone over the 10 years of data analysed. The p-value being 0.028 indicates that these results are not only statistically significant but also the mean values of number of cases in each zone are unequal.

Figure 4: Plots for K-means and PAM clustering



The two clustering algorithms K-means and PAM (Partitioning Around Medoids) produced similar results. Since clustering technique involves in grouping the data points with similar properties, both K-means and PAM clustering resulted in identifying two distinct groups in the data. As shown in figure 4, two perceptible groups are highlighted in red and blue colours. This is also validated by the silhouette width as 0.28 for K-means clustering and 0.29 for PAM clustering. Usually if the silhouette width value is close to 1, it is understood that the objects in the cluster are like each other. Though the values obtained for K-means and PAM clustering (0.28 and 0.29 respectively) are not completely closer to 1, we can understand that there is similarity to some extent in the objects of the groups identified. These results are also supported by the confusion matrix results with accuracy = 82%.

Figure 5: Comparison between Influenza cases and weather factors using boxplots.



The boxplots shown in figure 5 are constructed between the Influenza cases in New Zealand and various weather factors. In the category variable ‘0’ refers to Low Influenza cases and ‘1’ refers to High Influenza cases. There is a noticeable change in the number of Influenza cases as the weather conditions fluctuate. For example, when there is high temperature, the influenza cases are low whereas the low temperature is causing more influenza occurrence. Similarly, with the weather factors like absolute humidity, relative humidity, and solar radiation there is a clear distinction between Influenza cases as the values of these weather variables are changing. It is also interesting to see the number of Influenza cases are different between the zones of New Zealand. Moreover, out of the five weather variables shown in figure 5, it is evident that the maximum temperature and absolute humidity are predominantly influencing the influenza cases in New Zealand.

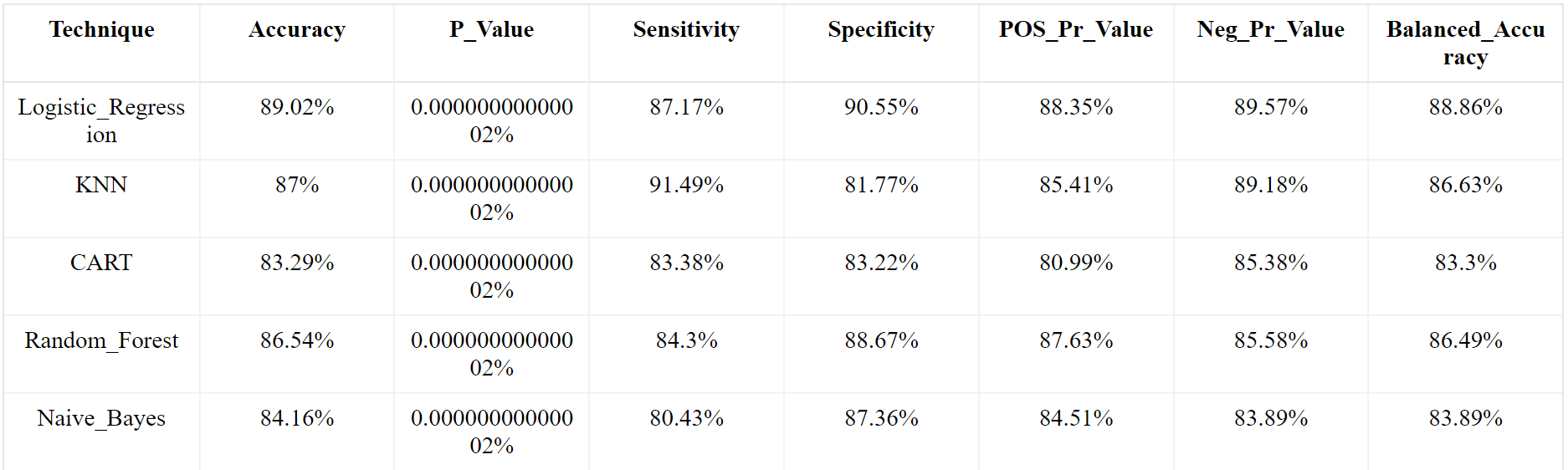
So, it is understood that the statistical analysis performed in this research is indicating two distinct groups in the data and it is obvious that the climate factors are clearly influencing the occurrence of Influenza in New Zealand. Hence, it is encouraged to use classification algorithms to build the final model as the classification is known to be a form of pattern recognition or understanding and grouping the objects into pre-set categories. As mentioned in above sections, this research made use of 5 classification algorithms such as Logistic Regression, K nearest neighbours (KNN), CART (Classification And Regression Tree), Random Forest, and Naïve Bayes. While building all the models, the variable ‘Category’ is used as dependant variable and the rest of the variables (i.e., tmax\_max, Zone, rad\_max, ah\_max, rh\_max, and rain\_mean) are used as input variables to generate predictions. A table is created with the results from five models that are generated using confusion matrix function as shown in figure 6.

Figure 6: The Predictive accuracy results produced by confusion matrix.

All the five models comparatively closer to each other in terms of the metrics produced by confusion matrix. The p-value is less than 0.05 among all the models which indicates all of them are statistically significant. The logistic regression model stands as the best suitable model to predict the Influenza hospitalizations in New Zealand with highest accuracy of 89.02%. However, KNN has higher sensitivity score (91.49%) compared to logistic regression (87.17%) but has the lowest specificity (81.77%) comparatively to all models and its overall accuracy is 87% which is the second highest among all the models. All the models achieved a balanced accuracy of above 83% with the lowest belongs to CART model (83.3%). This indicates that all the five models can generate valid predictions on Influenza cases with only few differences in the quality of predictions.

# 6. Conclusion

This research concludes that the climate factors such as temperature, solar radiation, and absolute humidity have significant corelations to the occurrence of Influenza cases in New Zealand. In particular, temperature and humidity are playing the key role to increase or decrease the influenza cases in different zones of New Zealand. According to the evaluation criteria used in this research, out of the five predictive models compared, logistic regression is the best model to predict the Influenza cases based on weather and climate conditions in New Zealand. However, the other four models will be useful to provide comparisons and deeper insights of future predictions to what the logistic regression model generates. This will help to produce more quality predictions as well as maintain and finetune the models as required.

# 7. Limitations

This research only made use of five predictive models such as Logistic Regression, K nearest neighbours (KNN), CART (Classification And Regression Tree), Random Forest, and Naïve Bayes that are available within R. It does not include any other predictive models in R, or any other predictive analytics work environments. It is possible that we might find a model better than logistic regression when the other predictive models are applied on the datasets used in this research. The final artefact is only based on the Influenza hospitalizations and Weather data for the period 2005 – 2015. it does not include any other factors or time periods. The final model could make predictions only if current weather station data is available for all the required variables that the model relies on.

# 8. Future Work

Since all the five models used in this research proved to be efficient with more than 83% accuracy, they could be combined to design an ensemble model which might generate better quality predictions. Further research could include other epidemiological and demographic data that may also help improve the accuracy of the predictive models generated. Finally, further research also could be conducted using other predictive analytics tools such as Weka Data Mining, NumPy, and SciPy etc. which may include additional data mining algorithms not available in R Studio to generate alternative predictive models to further this research area.

# 9. References

*A Short Introduction to the caret Package.* (n.d.). Retrieved from https://cran.r-project.org/: https://cran.r-project.org/web/packages/caret/vignettes/caret.html

Alison Mack, A. M. (2005). *The Threat of Pandemic Influenza: Are We Ready? Workshop Summary.* United States: National Academies Press.

Alpaydin, E. (2010). *Introduction to Machine Learning.* California, USA: MIT Press.

Azlinah Mohamed, B. W. (2019). *Supervised and Unsupervised Learning for Data Science.* Springer International Publishing.

Barry, J. M. (2020). *The Great Influenza: The Story of the Deadliest Pandemic in History.* Penguin Books Limited.

*Boruta*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/Boruta/versions/7.0.0/topics/Boruta#:~:text=Feature%20selection%20with%20the%20Boruta%20algorithm&text=The%20method%20performs%20a%20top,features%20to%20stabilise%20that%20test.

Brownie, J. (2019, May 22). Difference Between Classification and Regression in Machine Learning. *Machine Learning Mastery*.

Chopra, R. (2020). *Machine Learning* (2 ed.). New Delhi, India: Khanna publishing house.

*Classification And Regression With Random Forest*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/randomForest

Deshpande, V. K. (2015). *Predictive Analytics and Data Mining: Concepts and Practice with Rapidminer.* Elsevier Inc.

*dplyr*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/dplyr/versions/0.7.8

Elankumaran, H. S. (2015, February ). Environmental Role in Influenza Virus Outbreaks. *Annual Review of Animal Biosciences, 3*, 347-373.

Evan L. Ray, N. G. (2018, February 20). Prediction of infectious disease epidemics via weighted density ensembles. *PLOS Computational Biology*. Retrieved from https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005910

*factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. (n.d.). Retrieved from CRAN.R-project.org: https://CRAN.R-project.org/package=factoextra

*Fast Nearest Neighbor Search Algorithms and Applications*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/FNN

Finlay, S. (2014). *Predictive Analytics, Data Mining and Big Data: Myths, Misconceptions and Methods.* UK: Palgrave Macmillan.

Foster Provost, T. F. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking.* O'Reilly Media.

*ggplot2*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/ggplot2/versions/3.3.3

*ggpubr*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/ggpubr/versions/0.1.1

Hariom Tatsat, S. P. (2020). *Machine Learning and Data Science Blueprints for Finance.* O'Reilly Media.

Hyndman, R. (2020). *Package ‘forecast’.* Retrieved from https://cran.r-project.org/.

*ISLR*. (n.d.). Retrieved from Data for an Introduction to Statistical Learning with Applications in R: https://www.rdocumentation.org/packages/ISLR/versions/1.2

Kassambara, A. (2017). *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning.* STHDA.

Keith McCormick, J. S. (2017). *IBM SPSS Modeler Essentials: Effective Techniques for Building Powerful Data Mining and Predictive Analytics Solutions.* Packt Publishing.

Kelly R. Moran, G. F. (2016, November 14). Epidemic Forecasting is Messier Than Weather Forecasting: The Role of Human Behavior and Internet Data Streams in Epidemic Forecast. *The Journal of Infectious Diseases, 214*(4), 404-408. Retrieved from https://doi.org/10.1093/infdis/jiw375

Kent, J. (2019, January 22). Predictive Analytics Models Accurately Forecast Flu Trends. *Health IT Analytics*.

Lesmeister, C. (2015). *Mastering Machine Learning with R.* Packt Publishing.

Max Kuhn, K. J. (2013). *Applied Predictive Modeling.* Springer New York.

Mazzanti, S. (2020, March 18). Boruta Explained Exactly How You Wished Someone Explained to You. *towards data science*.

*Microsoft Excel*. (n.d.). Retrieved from Microsoft: https://www.microsoft.com/en-nz/microsoft-365/excel

*naivebayes*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/e1071/versions/1.7-3/topics/naiveBayes

Narkhede, S. (2018, May 10). Understanding Confusion Matrix. *towards data science*. Retrieved from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

*Package ‘mlbench’.* (n.d.). Retrieved from cran.r-project.org: https://cran.r-project.org/web/packages/mlbench

Qinneng Xu, Y. R. (2017, May 2). Forecasting influenza in Hong Kong with Google search queries and statistical model fusion. *PLOS ONE*. Retrieved from https://doi.org/10.1371/journal.pone.0176690

*R packages for data science*. (n.d.). Retrieved from Tidyverse: https://www.tidyverse.org/#:~:text=The%20tidyverse%20is%20an%20opinionated,Learn%20the%20tidyverse

*reactable.* (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/reactable

Revelle, W. (2020). *Package 'Psych'.* Retrieved from cran.r-project.org: https://cran.r-project.org/web/packages/psych/psych.pdf

Rice, G. W. (2018). *Black Flu 1918: The Story of New Zealand's Worst Public Health Disaster.* New Zealand: Canterbury University Press.

Robin, X. (2020). *Package 'Proc'.* Retrieved from https://cran.r-project.org/.

*rstatix*. (n.d.). Retrieved from RDocumentation: https://www.rdocumentation.org/packages/rstatix

Sadegh Bafandeh Imandoust, M. B. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background. *S B Imandoust et al. Int. Journal of Engineering Research and Applications, 3*(5), 605-610.

Song, J. (2019). *Big Data Analysis Using Machine Learning for Social Scientists and Criminologists.* Cambridge Scholars Publishing.

Spinu, V. (n.d.). *lubridate*. Retrieved from RDocumentation: https://www.rdocumentation.org/packages/lubridate/versions/1.7.9.2

Stephan Dreiseitl, L. O.-M. (2002, October). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics, 35*(5-6), 352-359.

Tony Thomas, A. P. (2019). *Machine Learning Approaches in Cyber Security Analytics.* Springer Singapore.

Torsten Hothorn, K. H. (2006). party: A Laboratory for Recursive Partytioning. Retrieved from https://cran.microsoft.com/snapshot/2015-11-17/web/packages/party/vignettes/party.pdf

*Types of clustering methods: Overview and quickstart R code*. (n.d.). Retrieved from Data Novia: https://www.datanovia.com/en/blog/types-of-clustering-methods-overview-and-quick-start-r-code/#:~:text=Clustering%20methods%20are%20used%20to,Hierarchical%20clustering

Vorhies, W. (2016, July 26). CRISP-DM – a Standard Methodology to Ensure a Good Outcome. *Data Science Central*.

What is balanced accuracy? (2020). *Statistical Odds & Ends*. Retrieved from https://statisticaloddsandends.wordpress.com/2020/01/23/what-is-balanced-accuracy/

*What is R?* (n.d.). Retrieved from R: https://www.r-project.org/about.html

# 10. Appendix

Below is the code developed in RStudio to complete all the four iterations and produce the final outcomes of this project.

#Iteration: 1 - Data Pre-processing

#install.packages("dplyr")

library**(**dplyr**)**

### Date packages

#install.packages("tidyverse")

#install.packages("lubridate")

library**(**tidyverse**)**

library**(**lubridate**)**

MOH **<-** read.csv**(**"MoHSubset.csv"**)**

head**(**MOH**)**

summary**(**MOH**)**

# Recode variables in MoHsubset

# Gender: M = 2, F = 1

MOH**$**GENDER **<-** ifelse**(**MOH**$**GENDER **==** "M", 2, 1**)**

# Age

MOH **<-** MOH %>%

mutate**(**AGEGRP **=** case\_when**(**

AGE\_DSCH **<=** 4 **~** 1,

AGE\_DSCH **>=** 5 **&** AGE\_DSCH **<=** 14 **~** 2,

AGE\_DSCH **>=** 15 **&** AGE\_DSCH **<=** 24 **~** 3,

AGE\_DSCH **>=** 25 **&** AGE\_DSCH **<=** 44 **~** 4,

AGE\_DSCH **>=** 45 **&** AGE\_DSCH **<=** 64 **~** 5,

AGE\_DSCH **>=** 65 **~** 6,

**TRUE** **~** 10

**)**

**)**

# Convert to date format

# Need to pre-format EVSTDATE column as a date in Excel first

# Use Custom format mm/d/yyyy option in Excel (need to type format)

# Check for NAs using summary command to ensure correct conversion

MOH**$**DATE **<-** as.Date**(**MOH**$**EVSTDATE, "%m/%d/%Y"**)**

# Add climate zone variable - based on TLADOM value

# Northern New Zealand - Zone 1

# Climate: Kaitaia (18183), Whangarei (1287), Auckland (1962), Tauranga (1615)

# DHBs: Northland (11), Waitemata (21), Auckland (22), Counties Manukau (23), Bay of Plenty (47)

# Central North Island - Zone 2

# Climate: Hamilton (2112), Taupo (1858), Rotorua (1770)

# DHBs: Waikato (31), Lakes (42)

# South-West North Island - Zone 3

# Climate: New Plymouth (2283), Wanganui (3715), Palmerston North (3243), Wellington (25354)

# DHBs: Taranaki (71), Wanganui (82), Mid Central (81), Capital and Coast (91), Hutt (92)

# Eastern North Island - Zone 4

# Climate: Gisborne (2810), Napier (2980), Masterton (Martinborough - 21938)

# DHBs: Tairawhiti (51), Hawkes Bay (61), Wairarapa (93)

# Northern South Island - Zone 5

# Climate: Nelson (4271), Blenheim (4326)

# DHBs: Nelson Marlborough (101)

# Western South Island - Zone 6

# Climate: Westport (7342), Hokitika (3910), Milford Sound (Pop. 120 not used)

# DHBs: West Coast (111)

# Eastern South Island - Zone 7

# Climate: Kaikoura (4506), Christchurch (4843), Timaru (5086)

# DHBs: Canterbury (121)

# Inland South Island - Zone 8

# Climate: Lake Tekapo (Pop. 400 Not used), Manapouri (Pop. 100 Not used), Queenstown (5451), Alexandra (5578)

# DHBs: South Canterbury (123), Otago (131)

# Southern New Zealand - Zone 9

# Climate: Dunedin (7339), Invercargill (11104)

# DHBs: Southland (160) (Dunedin TLA, Invercargill TLA)

MOH **<-** MOH %>%

mutate**(**ZONE **=** case\_when**(**

# Zone 1

DHBDOM **==** 11 **~** 1,

between**(**DHBDOM, 21, 23**)** **~** 1,

DHBDOM **==** 47 **~** 1,

# Zone 2

DHBDOM **==** 31 **~** 2,

DHBDOM **==** 42 **~** 2,

# Zone 3

DHBDOM **==** 71 **~** 3,

between**(**DHBDOM, 81, 82**)** **~** 3,

between**(**DHBDOM, 91, 92**)** **~** 3,

# Zone 4

DHBDOM **==** 51 **~** 4,

DHBDOM **==** 61 **~** 4,

DHBDOM **==** 93 **~** 4,

# Zone 5

DHBDOM **==** 101 **~** 5,

# Zone 6

DHBDOM **==** 111 **~** 6,

# Zone 7

DHBDOM **==** 121 **~** 7,

# Zone 8

DHBDOM **==** 123 **~** 8,

DHBDOM **==** 131 **~** 8,

# Zone 9

DHBDOM **==** 160 **~** 9,

**TRUE** **~** 10

**)**

**)**

# Check for unidentified DHBs

MOH %>% count**(**ZONE **==** 10**)**

# Recode SHTSTAY

# Short stay = 1, Long stay = 0

MOH**$**SHTSTAY **<-** ifelse**(**MOH**$**Short\_Stay\_ED\_Flag **==** "N", 0, 1**)**

# Check 297372 = TRUE

MOH %>% count**(**SHTSTAY **==** 1**)**

# Recode ENDTYPE

MOH **<-** MOH %>%

mutate**(**ENDCODE **=** case\_when**(**

END\_TYPE **==** "DA" **~** 1,

END\_TYPE **==** "DC" **~** 2,

END\_TYPE **==** "DD" **~** 3,

END\_TYPE **==** "DF" **~** 4,

END\_TYPE **==** "DI" **~** 5,

END\_TYPE **==** "DL" **~** 6,

END\_TYPE **==** "DN" **~** 7,

END\_TYPE **==** "DO" **~** 8,

END\_TYPE **==** "DP" **~** 9,

END\_TYPE **==** "DR" **~** 10,

END\_TYPE **==** "DS" **~** 11,

END\_TYPE **==** "DT" **~** 12,

END\_TYPE **==** "DW" **~** 13,

END\_TYPE **==** "EA" **~** 14,

END\_TYPE **==** "ED" **~** 15,

END\_TYPE **==** "EI" **~** 16,

END\_TYPE **==** "ER" **~** 17,

END\_TYPE **==** "ES" **~** 18,

END\_TYPE **==** "ET" **~** 19,

**TRUE** **~** 20

**)**

**)**

# Check - should return all false

MOH %>% count**(**ENDCODE **==** 20**)**

# Recode diag01 - use summary command to ensure no 999 values

# J001 - J006 (acute upper respiratory infections) -> 1

# J10 - J18 (Influenza and pneumonia) -> 2

# J20 - J22 (Other acute lower respiratory infections) -> 3

MOH **<-** MOH %>%

mutate**(**DIAG **=** case\_when**(**

diag01 **==** "J00" **~** 1,

diag01 **==** "J010" **~** 1,

diag01 **==** "J011" **~** 1,

diag01 **==** "J012" **~** 1,

diag01 **==** "J013" **~** 1,

diag01 **==** "J014" **~** 1,

diag01 **==** "J018" **~** 1,

diag01 **==** "J019" **~** 1,

diag01 **==** "J020" **~** 1,

diag01 **==** "J028" **~** 1,

diag01 **==** "J029" **~** 1,

diag01 **==** "J030" **~** 1,

diag01 **==** "J038" **~** 1,

diag01 **==** "J039" **~** 1,

diag01 **==** "J040" **~** 1,

diag01 **==** "J041" **~** 1,

diag01 **==** "J042" **~** 1,

diag01 **==** "J050" **~** 1,

diag01 **==** "J051" **~** 1,

diag01 **==** "J060" **~** 1,

diag01 **==** "J068" **~** 1,

diag01 **==** "J069" **~** 1,

diag01 **==** "J100" **~** 2,

diag01 **==** "J101" **~** 2,

diag01 **==** "J108" **~** 2,

diag01 **==** "J110" **~** 2,

diag01 **==** "J111" **~** 2,

diag01 **==** "J118" **~** 2,

diag01 **==** "J120" **~** 2,

diag01 **==** "J121" **~** 2,

diag01 **==** "J122" **~** 2,

diag01 **==** "J128" **~** 2,

diag01 **==** "J129" **~** 2,

diag01 **==** "J13" **~** 2,

diag01 **==** "J14" **~** 2,

diag01 **==** "J150" **~** 2,

diag01 **==** "J151" **~** 2,

diag01 **==** "J152" **~** 2,

diag01 **==** "J153" **~** 2,

diag01 **==** "J154" **~** 2,

diag01 **==** "J155" **~** 2,

diag01 **==** "J156" **~** 2,

diag01 **==** "J157" **~** 2,

diag01 **==** "J158" **~** 2,

diag01 **==** "J159" **~** 2,

diag01 **==** "J160" **~** 2,

diag01 **==** "J168" **~** 2,

diag01 **==** "J170" **~** 2,

diag01 **==** "J171" **~** 2,

diag01 **==** "J172" **~** 2,

diag01 **==** "J173" **~** 2,

diag01 **==** "J180" **~** 2,

diag01 **==** "J181" **~** 2,

diag01 **==** "J182" **~** 2,

diag01 **==** "J188" **~** 2,

diag01 **==** "J189" **~** 2,

diag01 **==** "J200" **~** 3,

diag01 **==** "J201" **~** 3,

diag01 **==** "J202" **~** 3,

diag01 **==** "J204" **~** 3,

diag01 **==** "J205" **~** 3,

diag01 **==** "J206" **~** 3,

diag01 **==** "J208" **~** 3,

diag01 **==** "J209" **~** 3,

diag01 **==** "J210" **~** 3,

diag01 **==** "J218" **~** 3,

diag01 **==** "J219" **~** 3,

diag01 **==** "J22" **~** 3,

**TRUE** **~** 999

**)**

**)**

MOH %>% count**(**DIAG **==** 1**)** # 81739

MOH %>% count**(**DIAG **==** 2**)** # 150152

MOH %>% count**(**DIAG **==** 3**)** # 120577

MOH %>% count**(**DIAG **==** 999**)**

# Ethnicity

# 3 major groupings; Maori, Pacific and Other

# Ethnicity codes > 53 not used for this analsysis

# These are records where the ethnicity details are unknown

MOH **<-** MOH %>%

mutate**(**ETHNICITY **=** case\_when**(**

ETHNICGP **==** 21 **~** 1,

between**(**ETHNICGP, 30, 37**)** **~** 3,

between**(**ETHNICGP, 10, 12**)** **~** 2,

between**(**ETHNICGP, 40, 53**)** **~** 2,

**TRUE** **~** 6

**)**

**)**

# Add CASEYEAR variable to enable grouping by year

MOH **<-** MOH %>%

mutate**(**CASEYEAR **=** year**(**DATE**))**

# Standardisation

# Need to use standardised rates NOT raw case numbers

# Libraries used for date calculations

#install.packages("tidyverse")

#install.packages("lubridate")

library**(**tidyverse**)**

library**(**lubridate**)**

# Count rows within these groups - helps to identify groups with no cases

#Also provides the total number of cases by zone, ethnicity, agegrp, and year

cases.grouped **<-** MOH %>% count**(**CASEYEAR, ZONE, ETHNICITY, AGEGRP**)**

# Use only ethnicity 1 through to 3

cases.filtered **<-** cases.grouped %>% filter**(**ETHNICITY **<=**3**)**

# Convert to data frame for easier handling and rename count column ('n') to cases

hospCases.df **<-** as.data.frame**(**cases.filtered**)**

names**(**hospCases.df**)[**names**(**hospCases.df**)** **==** "n"**]** **<-** "Cases"

# Import population details for each DHB. This data will allow us to

# calculate a standardized hospital rates for Influenza cases

pop **<-** read.csv**(**"DHBpopV4.csv"**)**

# Recode data

# Add age groups - last two age groups (65-79 & 80+) in DHBpopv4 are merged to match the hospital data

pop.agegrp **<-** pop %>%

mutate**(**AGEGRP **=** case\_when**(**

agegp **==** "00-04" **~** 1,

agegp **==** "05-14" **~** 2,

agegp **==** "15-24" **~** 3,

agegp **==** "25-44" **~** 4,

agegp **==** "45-64" **~** 5,

agegp **==** "65-79" **~** 6,

agegp **==** "80+" **~** 6,

**TRUE** **~** 999

**)**

**)**

# Add ethnicity

pop.ethnic **<-** pop.agegrp %>%

mutate**(**ethnicity **=** case\_when**(**

ethnic **==** "Maori" **~** 1,

ethnic **==** "Other" **~** 2,

ethnic **==** "Pacific" **~** 3,

**TRUE** **~** 999

**)**

**)**

# Add zone value to each row

pop.recoded **<-** pop.ethnic %>%

mutate**(**ZONE **=** case\_when**(**

# Zone 1 - Northland, Waitemata, Auckland, Counties Manukau, Bay of Plenty

area **==** "Northland" **~** 1,

area **==** "Waitemata" **~** 1,

area **==** "Auckland" **~** 1,

area **==** "Counties Manukau" **~** 1,

area **==** "Bay of Plenty" **~** 1,

# Zone 2 - Waikato, Lakes

area **==** "Waikato" **~** 2,

area **==** "Lakes" **~** 2,

# Zone 3 - Taranaki, Whanganui, Mid Central, Capital and Coast, Hutt

area **==** "Taranaki" **~** 3,

area **==** "Whanganui" **~** 3,

area **==** "MidCentral" **~** 3,

area **==** "Capital and Coast" **~** 3,

area **==** "Hutt" **~** 3,

# Zone 4 - Tairawhiti, Hawkes Bay, Wairarapa

area **==** "Tairawhiti" **~** 4,

area **==** "Hawkes Bay" **~** 4,

area **==** "Wairarapa" **~** 4,

# Zone 5 - Nelson Marlborough

area **==** "Nelson Marlborough" **~** 5,

# Zone 6 - West Coast

area **==** "West Coast" **~** 6,

# Zone 7 - Canterbury

area **==** "Canterbury" **~** 7,

# Zone 8 - South Canterbury, Otago

area **==** "South Canterbury" **~** 8,

area **==** "Otago" **~** 8,

# Zone 9 -

area **==** "Southern" **~** 9,

**TRUE** **~** 10

**)**

**)**

# Don't include rows that have no flu cases in hospCases.df

# Select all rows not in ZONE 6 with ethnicity and agegrp = 3

# Select all rows not in ZONE 8 with ethnicity 3 and agegrp = 3 or 6

# The rows not included do not have any flu cases to perform analysis

pop.filtered **<-** pop.recoded %>% filter**(!(**ZONE **==** 6 **&** ethnicity **==** 3 **&** AGEGRP **==** 3**))**

pop.filtered **<-** pop.filtered %>% filter**(!(**ZONE **==** 8 **&** ethnicity **==** 3 **&** AGEGRP **==** 3**))**

pop.filtered **<-** pop.filtered %>% filter**(!(**ZONE **==** 8 **&** ethnicity **==** 3 **&** AGEGRP **==** 6**))**

# Group data for cases and population to perform a one way anova test

# Pool together data across whole 10 year period

pop.grouped **<-** pop.filtered %>%

group\_by**(**ZONE, ethnicity, AGEGRP**)** %>%

summarize**(**StudyPopTotal **=** sum**(**Pop**))**

# Define standard population - 2013, all DHBs, 3 ethnicities and age groups

pop.standard **<-** pop.filtered %>%

filter**(**year **==** '2013'**)** %>%

group\_by**(**ZONE, ethnicity, AGEGRP**)** %>%

summarize**(**StandPopTotal **=** sum**(**Pop**))**

# Calculate sum of flu cases for selected grouping

cases.grouped **<-** hospCases.df %>%

group\_by**(**ZONE, ETHNICITY, AGEGRP**)** %>%

summarize**(**CaseTotal **=** sum**(**Cases**))**

# Add study population totals to hospital data

hosp.collated **<-** cases.grouped

hosp.collated**$**StudyPop **<-** pop.grouped**$**StudyPopTotal

# Add standard population totals

hosp.collated**$**StandPop **<-** pop.standard**$**StandPopTotal

# Calculate crude flu rate and std rate for each observation in zones

hosp.collated**$**crude **<-** hosp.collated**$**CaseTotal**/**hosp.collated**$**StudyPop**\***100000

hosp.collated**$**std **<-** hosp.collated**$**CaseTotal**/**hosp.collated**$**StudyPop**\***hosp.collated**$**StandPop

# Total by Zone and calculate standardized rate

stand.rates **<-** hosp.collated %>%

group\_by**(**ZONE**)** %>%

summarize**(**ZoneSTD **=** sum**(**std**)/**sum**(**StandPop**)\***100000**)**

#Iteration: 2 - Statistical analysis

#ANOVA Analysis

library**(**tidyverse**)**

library**(**ggpubr**)**

library**(**rstatix**)**

#Compute the mean and the SD (standard deviation)

hosp.collated %>%

group\_by**(**ZONE, ETHNICITY**)** %>%

summarize**(**ZoneSTD **=** sum**(**std**)/**sum**(**StandPop**)\***100000**)** %>%

get\_summary\_stats**(**ZoneSTD, type **=** "mean\_sd"**)**

# Find Outliers

stand.rates1 **<-** hosp.collated %>%

group\_by**(**ZONE, ETHNICITY**)** %>%

summarize**(**ZoneSTD **=** sum**(**std**)/**sum**(**StandPop**)\***100000**)**

stand.rates1 %>%

group\_by**(**ZONE, ETHNICITY**)** %>%

identify\_outliers**(**ZoneSTD**)**

#Visualization

ggboxplot**(**stand.rates1, x **=** "ZONE", y **=** "ZoneSTD"**)**

# Build the linear model

modelX **=** lm**(**ZoneSTD **~** ZONE, data **=** stand.rates1**)**

modelY **<-** lm**(**ZoneSTD **~** ZONE, data **=** stand.rates**)**

# Create a QQ plot of residuals

ggqqplot**(**residuals**(**modelX**))**

# All the points fall approximately at the reference line

#which indicates the normality of the data

ggqqplot**(**residuals**(**modelY**))**

#Majority of the data points are at reference line

#which indicates the normality of the data

#Shapiro Test

shapiro\_test**(**residuals**(**modelY**))**

# A tibble: 1 x 3

# variable statistic p.value

# <chr> <dbl> <dbl>

# 1 residuals(modelY) 0.949 0.678

# The p-value 0.678 is greater than 0.05.

#Hence Shapiro test proves the normality of the data.

#Perform ANOVA test

aov **<-** stand.rates %>% anova\_test**(**ZoneSTD **~** ZONE**)**

aov

#ANOVA Table (type II tests)

#Effect DFn DFd F p p<.05 ges

#ZONE 1 7 7.57 0.028 \* 0.52

#From the above Anova table, it is clear that there are significant differences

#between the groups with p-value = 0.028 and ges = 0.52 (52% of change between number of cases by zone)

#Clustering

hospWeather.df **<-** read.csv**(**"hospWeather.csv"**)**

# Use Boruta technique to assess attribute importance and select the variables

#that contributes more to build models compared to others

#install.packages("Boruta")

library**(**Boruta**)**

# Scale variables - ignore YEAR, WEEK and Cases columns

hosp.weather.scaled **<-** as.data.frame**(**scale**(**hospWeather.df**[**, **-**c**(**2**:**4**)]**, scale **=** **TRUE))**

Boruta.CAT **<-** Boruta**(**Category**~**., data **=** hosp.weather.scaled, doTrace **=** 2**)**

# attStats lists attributes in descending order of importance (higher the value to more important the variable)

attStats**(**Boruta.CAT**)**

# Visualise the Boruta results

plot**(**Boruta.CAT, xlab **=** "", xaxt **=** "n"**)**

lz**<-**lapply**(**1**:**ncol**(**Boruta.CAT**$**ImpHistory**)**,**function(**i**)**

Boruta.CAT**$**ImpHistory**[**is.finite**(**Boruta.CAT**$**ImpHistory**[**,i**])**,i**])**

names**(**lz**)** **<-** colnames**(**Boruta.CAT**$**ImpHistory**)**

Labels **<-** sort**(**sapply**(**lz,median**))**

axis**(**side **=** 1,las**=**2,labels **=** names**(**Labels**)**,

at **=** 1**:**ncol**(**Boruta.CAT**$**ImpHistory**)**, cex.axis **=** 0.7**)**

#the tentative atrributes picked - tmax\_max + ZONE + rad\_max + ah\_max + rh\_max + rain\_mean

imp.df **<-** hospWeather.df**[**, c**(**'tmax\_max', 'ZONE','rad\_max', 'ah\_max', 'rh\_max', 'rain\_mean', 'Category'**)]**

# Perform correlation analysis

library**(**gplots**)**

#Scale the data

imp.df.scaled **<-** scale**(**imp.df, center **=** **TRUE**, scale **=** **TRUE)**

heatmap.2**(**cor**(**imp.df.scaled**)**, Rowv **=** **FALSE**, Colv **=** **FALSE**, dendrogram **=** "none",

cellnote **=** round**(**cor**(**imp.df.scaled**)**,2**)**,

notecol **=** "black", key **=** **FALSE**, trace **=** 'none', margins **=** c**(**10,10**))**

# Analysis shows correlations between rad, tmax, ah, rh variables and Category

#Clustering Techniques

install.packages**(**c**(**"cluster", "factoextra"**))**

#KMeans

library**(**cluster**)**

library**(**factoextra**)**

set.seed**(**123**)**

df.scaled **<-** scale**(**imp.df**)**

km.res **<-** kmeans**(**df.scaled**[**, **-**c**(**7**)]**, 3, nstart **=** 25**)**

print**(**km.res**)**

# K-means clustering with 3 clusters of sizes 1582, 1274, 1384

#

# Cluster means:

# tmax\_max ZONE rad\_max ah\_max rh\_max rain\_mean

# 1 -0.8630754 0.4438080 -0.9504619 -0.8559084 0.3570747 -0.05799816

# 2 0.8825426 0.4760129 0.8377745 0.1136978 -0.8299093 -0.41910934

# 3 0.1741518 -0.9454803 0.3152501 0.8736966 0.3557892 0.45209421

#1 - Low, 2 - Normal and 3 - High

# Within cluster sum of squares by cluster:

# [1] 4206.455 5035.929 5682.344

# (between\_SS / total\_SS = 41.3 %)

fviz\_cluster**(**km.res, imp.df, ellipse.type **=** "convex", ggtheme **=** theme\_bw**())**

#PAM clustering

library**(**cluster**)**

library**(**factoextra**)**

pam.res **<-** pam**(**df.scaled**[**, **-**c**(**7**)]**, 3**)**

print**(**pam.res**)**

# Medoids:

# ID tmax\_max ZONE rad\_max ah\_max rh\_max

# [1,] 1905 0.9032366 -0.09490461 1.0361644 0.9706662 -0.3779538

# [2,] 1568 -0.3420063 -0.47452303 -0.2504336 0.2357345 0.1854961

# [3,] 2716 -0.8673432 0.66433224 -1.2323110 -1.1606356 0.4036057

# rain\_mean

# [1,] -0.50618748

# [2,] 0.04221485

# [3,] -0.36567644

# Objective function:

# build swap

# 1.800260 1.776735

fviz\_cluster**(**pam.res, geom **=** "point",

main **=** "PAM Clusters",

ellipse.type **=** "convex",

ggtheme **=** theme\_bw**())**

#Remove Normal =2 cases from the data imp.df

imp.df1 **<-** subset**(**imp.df, Category**!=**2**)**

#KMeans clustering with new data with out normal cases

set.seed**(**123**)**

df.scaled1 **<-** scale**(**imp.df1**)**

km.res1 **<-** kmeans**(**df.scaled1**[**, **-**c**(**7**)]**, 2, nstart **=** 25**)**

print**(**km.res1**)**

# K-means clustering with 2 clusters of sizes 1122, 951

#

# Cluster means:

# tmax\_max ZONE rad\_max ah\_max rh\_max rain\_mean

# 1 -0.7011780 0.1477624 -0.7372458 -0.5922753 0.3332528 0.08232842

# 2 0.8272574 -0.1743316 0.8698105 0.6987728 -0.3931753 -0.09713196

#K-means clustering with 2 clusters of sizes 1122, 951

#1 - Low cases and 3 - High cases

# Within cluster sum of squares by cluster:

# [1] 3636.120 5064.547

# (between\_SS / total\_SS = 30.0 %)

#Visualization of clusters

fviz\_cluster**(**km.res1, imp.df1, geom **=** "point",

main **=** "KM cluster",

ellipse.type **=** "convex",

ggtheme **=** theme\_gray**())**

#PAM clustering with new data without using normal level cases

library**(**cluster**)**

library**(**factoextra**)**

pam.res1 **<-** pam**(**df.scaled1**[**, **-**c**(**7**)]**, 2**)**

print**(**pam.res1**)**

# Medoids:

# ID tmax\_max ZONE rad\_max ah\_max rh\_max

# 1947 950 0.687712 -0.09929047 0.8041866 0.5720145 -0.003534997

# 1881 916 -0.769566 -0.09929047 -0.7954281 -0.5925899 0.273918950

# rain\_mean

# 1947 0.15625073

# 1881 0.09214033

# Objective function:

# build swap

# 1.956989 1.899875

fviz\_cluster**(**pam.res1, geom **=** "point",

# palette.colors(n= Null, palette = "0kabe\_Ito", alpha, recycle = FALSE),

main **=** "PAM cluster",

ellipse.type **=** "convex",

ggtheme **=** theme\_grey**())**

#silhouette Analysis for K-Means

# Visualize silhouette information

library**(**"cluster"**)**

library**(**"factoextra"**)**

sil **<-** silhouette**(**km.res1**$**cluster, dist**(**df.scaled1**[**, **-**c**(**7**)]))**

fviz\_silhouette**(**sil**)**

#Average Silhouette width is 0.29

# cluster size ave.sil.width

# 1 1 1122 0.36

# 2 2 951 0.20

# Identify observation with negative silhouette

neg\_sil\_index **<-** which**(**sil**[**, "sil\_width"**]** **<** 0**)**

sil**[**neg\_sil\_index, , drop **=** **FALSE]**

#Silhouette Analysis for PAM clustering

require**(**cluster**)**

sil.pam.res **<-** pam**(**df.scaled1**[**, **-**c**(**7**)]**, 2**)**

# Visualize silhouette information

fviz\_silhouette**(**sil.pam.res**)**

#Average Silhouette width is 0.28

# cluster size ave.sil.width

# 1 1 1000 0.19

# 2 2 1073 0.37

#Add the cluster values to imp.df1 as a new column

imp.df1**$**kclust **<-** as.numeric**(**km.res1**$**cluster**)**

with**(**imp.df1, table**(**kclust, Category**))**

# Category

# kclust 1 3

# 1 187 935

# 2 765 186

#Confusion Matrix

#(935+765)/(187+765+935+186) = 82%,

#This suggests a good accuracy of the clustering results.

#Iteration: 3 - Building Predictive models

#Five different algorithms have been applied on the final data set

# Those are Logistic Regression, KNN, CART, Random Forest and Naive bayes.

#1. Logistic Regression Model

library**(**ISLR**)**

library**(**tidyverse**)**

library**(**lubridate**)**

#Convert the Category variabe into binary

Model\_Data **<-** imp.df1 %>%

mutate**(**FluRate **=** case\_when**(**

Category **==** 1 **~** 0,

Category **==** 3 **~** 1,

**)**

**)**

#Scale the data to normalize the variables

scale**(**Model\_Data, center **=** **TRUE**, scale **=** **TRUE)**

#Convert FluRate into factor variable.

Model\_Data**$**FluRate **<-** factor**(**Model\_Data**$**FluRate**)**

#Split data into train (60%) and test (40%)

set.seed**(**1234**)**

train.index **<-** sample**(**c**(**1**:**1244**)**, 829, replace **=** **FALSE)**

test.df **<-** Model\_Data**[**train.index, **-**c**(**7**)** **]**

train.df **<-** Model\_Data**[-**train.index, **-**c**(**7**)** **]**

#Build the logistic regression model

log\_reg **<-** glm**(**FluRate **~** tmax\_max **+** ZONE **+** rad\_max **+** ah\_max **+** rh\_max **+** rain\_mean, data **=** train.df, family **=** 'binomial'**)**

summary**(**log\_reg**)**

# Deviance Residuals:

# Min 1Q Median 3Q Max

# -3.1728 -0.5130 0.2028 0.5698 3.1492

#

# Coefficients:

# Estimate Std. Error z value Pr(>|z|)

# (Intercept) 1.15330 1.65811 0.696 0.48671

# tmax\_max -0.11842 0.02643 -4.481 7.43e-06 \*\*\*

# ZONE -0.36057 0.04468 -8.070 7.05e-16 \*\*\*

# rad\_max 0.02886 0.01523 1.896 0.05799 .

# ah\_max -0.80129 0.06852 -11.695 < 2e-16 \*\*\*

# rh\_max 0.11851 0.01773 6.682 2.35e-11 \*\*\*

# rain\_mean 0.07758 0.02473 3.137 0.00171 \*\*

# ---

# Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

#

# (Dispersion parameter for binomial family taken to be 1)

#

# Null deviance: 1718.32 on 1243 degrees of freedom

# Residual deviance: 946.45 on 1237 degrees of freedom

# AIC: 960.45

#

# Number of Fisher Scoring iterations: 5

#Predict the model on test data

library**(**forecast**)**

library**(**caret**)**

p1 **<-** predict**(**log\_reg, test.df, type**=**'response'**)**

str**(**p1**)**

# Convert p1 into factor variable in order to build confusion matrix

pred\_glm **<-** ifelse**(**p1**>**0.5, 1, 0**)**

pred\_glm **<-** as.factor**(**pred\_glm**)**

#Confusion Matrix

confusionMatrix**(**pred\_glm, test.df**$**FluRate**)**

# Confusion Matrix and Statistics

#

# Reference

# Prediction 0 1

# 0 326 43

# 1 48 412

#

# Accuracy : 0.8902

# 95% CI : (0.8669, 0.9107)

# No Information Rate : 0.5489

# P-Value [Acc > NIR] : <2e-16

#

# Kappa : 0.7781

#

# Mcnemar's Test P-Value : 0.675

#

# Sensitivity : 0.8717

# Specificity : 0.9055

# Pos Pred Value : 0.8835

# Neg Pred Value : 0.8957

# Prevalence : 0.4511

# Detection Rate : 0.3932

# Detection Prevalence : 0.4451

# Balanced Accuracy : 0.8886

#

# 'Positive' Class : 0

#Goodness of Fit Test

with**(**log\_reg, pchisq**(**null.deviance **-** deviance, df.null **-** df.residual, lower.tail **=** F**))**

#the p-value is 1.844561e-163 which way lesser then 0.05.

#So the model log\_reg is statistically significant.

#2. KNN Model

#install library

library**(**FNN**)**

library**(**caret**)**

library**(**pROC**)**

library**(**mlbench**)**

#Prepare the data set to be used for the model,

# and remove the Normal cases (=2) from category variable.

imp.df **<-** hospWeather.df**[**, c**(**'tmax\_max', 'ZONE','rad\_max', 'ah\_max', 'rh\_max', 'rain\_mean', 'Category'**)]**

imp.df1.knn **<-** subset**(**imp.df, Category**!=**2**)**

str**(**imp.df1.knn**)**

#Convert the values in Category variable (1= Low and 3= high)

#and then convert it to factor variable.

imp.df1.knn**$**Category**[**imp.df1.knn**$**Category **==** 1**]<-** 'Low'

imp.df1.knn**$**Category**[**imp.df1.knn**$**Category **==** 3**]<-** 'high'

imp.df1.knn**$**Category **<-** factor**(**imp.df1.knn**$**Category**)**

#Data Partition into training = 60% and testing = 40%

set.seed**(**1234**)**

ind **<-** sample**(**2, nrow**(**imp.df1.knn**)**, replace **=** T, prob **=** c**(**0.6, 0.4**))**

training **<-** imp.df1.knn**[**ind **==** 1,**]**

testing **<-** imp.df1.knn**[**ind **==** 2,**]**

#Build KNN Model

trControl **<-** trainControl**(**method **=** "repeatedcv",

number **=** 10,

repeats **=** 3**)**

set.seed**(**222**)**

fit **<-** train**(**Category **~**.,

data **=** training,

method **=** 'knn',

tuneLength **=** 20,

trControl **=** trControl,

preProc **=** c**(**"center", "scale"**))**

fit

# k-Nearest Neighbors

#

# 1265 samples

# 6 predictor

# 2 classes: 'high', 'Low'

#

# Pre-processing: centered (6), scaled (6)

# Resampling: Cross-Validated (10 fold, repeated 3 times)

# Summary of sample sizes: 1139, 1139, 1139, 1138, 1138, 1138, ...

# Resampling results across tuning parameters:

#

# k Accuracy Kappa

# 5 0.8619334 0.7204020

# 7 0.8664142 0.7297140

# 9 0.8656248 0.7281800

# 11 0.8677245 0.7324062

# 13 0.8637770 0.7245695

# 15 0.8648331 0.7266410

# 17 0.8664141 0.7299868

# 19 0.8690534 0.7351926

# 21 0.8672057 0.7314517

# 23 0.8658850 0.7286836

# 25 0.8661454 0.7292295

# 27 0.8653558 0.7276252

# 29 0.8656246 0.7281381

# 31 0.8645664 0.7259226

# 33 0.8614022 0.7192990

# 35 0.8621916 0.7208694

# 37 0.8632478 0.7230992

# 39 0.8624562 0.7214629

# 41 0.8648226 0.7263930

# 43 0.8632416 0.7231234

#

# Accuracy was used to select the optimal model using the largest value.

# The final value used for the model was k = 19.

plot**(**fit**)**

#ROC curve variable importance

varImp**(**fit**)**

#Importance

#tmax\_max 100.00

#ah\_max 83.34

#rad\_max 76.84

#rh\_max 41.28

#rain\_mean 12.60

#ZONE 0.00

#Apply the model on the testing data

pred **<-** predict**(**fit, newdata **=** testing**)**

confusionMatrix**(**pred, testing**$**Category**)**

# Confusion Matrix and Statistics

#

# Reference

# Prediction high Low

# high 398 68

# Low 37 305

#

# Accuracy : 0.87

# 95% CI : (0.8449, 0.8925)

# No Information Rate : 0.5384

# P-Value [Acc > NIR] : < 2.2e-16

#

# Kappa : 0.737

#

# Mcnemar's Test P-Value : 0.003415

#

# Sensitivity : 0.9149

# Specificity : 0.8177

# Pos Pred Value : 0.8541

# Neg Pred Value : 0.8918

# Prevalence : 0.5384

# Detection Rate : 0.4926

# Detection Prevalence : 0.5767

# Balanced Accuracy : 0.8663

#

# 'Positive' Class : high

#3. CART (Classification And Regression Tree) algorithm

#Prepare data set for CART model

imp.df **<-** hospWeather.df**[**, c**(**'tmax\_max', 'ZONE','rad\_max', 'ah\_max', 'rh\_max', 'rain\_mean', 'Category'**)]**

imp.df1.cart **<-** subset**(**imp.df, Category**!=**2**)**

str**(**imp.df1.cart**)**

#Scale the data

scale**(**imp.df1.cart, center **=** **TRUE**, scale **=** **TRUE)**

#Change the category variable to factor variable

imp.df1.cart**$**CategoryF **<-** factor**(**imp.df1.cart**$**Category**)**

str**(**imp.df1.cart**)**

#partition data into training and validation (train:60 - Valid:40)

set.seed**(**1234**)**

pd **<-** sample**(**2, nrow**(**imp.df1.cart**)**, replace **=** **TRUE**, prob **=** c**(**0.6, 0.4**))**

traincart **<-** imp.df1.cart**[**pd**==**1,**]**

validcart **<-** imp.df1.cart**[**pd**==**2,**]**

#decision tree with party package

library**(**party**)**

tree **<-** ctree**(**CategoryF**~**tmax\_max**+**ZONE**+**rad\_max**+**ah\_max**+**rh\_max**+**rain\_mean,

data **=** traincart, controls **=** ctree\_control**(**mincriterion **=** 0.99, minsplit **=** 500**))**

tree

# Conditional inference tree with 7 terminal nodes

#

# Response: CategoryF

# Inputs: tmax\_max, ZONE, rad\_max, ah\_max, rh\_max, rain\_mean

# Number of observations: 1265

#

# 1) tmax\_max <= 22; criterion = 1, statistic = 541.581

# 2) ah\_max <= 12.9; criterion = 1, statistic = 38.54

# 3) rh\_max <= 82; criterion = 1, statistic = 46.902

# 4)\* weights = 10

# 3) rh\_max > 82

# 5) ah\_max <= 7.5; criterion = 0.998, statistic = 12.616

# 6)\* weights = 134

# 5) ah\_max > 7.5

# 7) ZONE <= 5; criterion = 1, statistic = 25.865

# 8)\* weights = 281

# 7) ZONE > 5

# 9)\* weights = 246

# 2) ah\_max > 12.9

# 10)\* weights = 48

# 1) tmax\_max > 22

# 11) ah\_max <= 9.9; criterion = 1, statistic = 92.01

# 12)\* weights = 94

# 11) ah\_max > 9.9

# 13)\* weights = 452

plot**(**tree**)**

#predict the model on the test data

predict\_tree **<-** predict**(**tree, newdata **=** validcart**)**

#Confusion Matrix

confusionMatrix**(**predict\_tree, validcart**$**CategoryF**)**

# Confusion Matrix and Statistics

# Reference

# Prediction 1 3

# 1 311 73

# 3 62 362

#

# Accuracy : 0.8329

# 95% CI : (0.8054, 0.858)

# No Information Rate : 0.5384

# P-Value [Acc > NIR] : <2e-16

#

# Kappa : 0.6646

#

# Mcnemar's Test P-Value : 0.3894

#

# Sensitivity : 0.8338

# Specificity : 0.8322

# Pos Pred Value : 0.8099

# Neg Pred Value : 0.8538

# Prevalence : 0.4616

# Detection Rate : 0.3849

# Detection Prevalence : 0.4752

# Balanced Accuracy : 0.8330

#

# 'Positive' Class : 1

#4. Random Forest Model

#Prepare data set for the model

imp.df **<-** hospWeather.df**[**, c**(**'tmax\_max', 'ZONE','rad\_max', 'ah\_max', 'rh\_max', 'rain\_mean', 'Category'**)]**

imp.df1.rf **<-** subset**(**imp.df, Category**!=**2**)**

str**(**imp.df1.rf**)**

#Scale the data

scale**(**imp.df1.rf, center **=** **TRUE**, scale **=** **TRUE)**

#Convert Category variable into factor variable

imp.df1.rf**$**Category **<-** as.factor**(**imp.df1.rf**$**Category**)**

str**(**imp.df1.rf**)**

table**(**imp.df1.rf**$**Category**)**

# 1 3

#952 1121

#data partition into 60-40 split

set.seed**(**123**)**

ind **<-** sample**(**2, nrow**(**imp.df1.rf**)**, replace **=** **TRUE**, prob **=** c**(**0.6, 0.4**))**

trainrandom **<-** imp.df1.rf**[**ind**==**1,**]**

testrandom **<-** imp.df1.rf**[**ind**==**2,**]**

#Build the random forests model

library**(**randomForest**)**

set.seed**(**222**)**

rf **<-** randomForest**(**Category**~**., data **=** trainrandom,

ntree **=** 450,

mtry **=** 2,

importance **=** **TRUE**,

proximity **=** **TRUE**,

**)**

print**(**rf**)**

# Type of random forest: classification

# Number of trees: 450

# No. of variables tried at each split: 2

#

# OOB estimate of error rate: 11.8%

# Confusion matrix:

# 1 3 class.error

# 1 471 86 0.15439856

# 3 63 643 0.08923513

attributes**(**rf**)**

#Prediction and confusion matrix - train data

library**(**caret**)**

p1 **<-** predict**(**rf, trainrandom**)**

head**(**p1**)**

head**(**trainrandom**$**Category**)**

confusionMatrix**(**p1, trainrandom**$**Category**)**

#Accuracy is 100% as the model already has seen the test data

#The model also predicted and classified all the records correctly

#Prediction and confusion matrix - test data

predict\_rf **<-** predict**(**rf, testrandom**)**

head**(**p2**)**

head**(**testrandom**$**Category**)**

confusionMatrix**(**predict\_rf, testrandom**$**Category**)**

# Confusion Matrix and Statistics

#

# Reference

# Prediction 1 3

# 1 333 47

# 3 62 368

#

# Accuracy : 0.8654

# 95% CI : (0.84, 0.8882)

# No Information Rate : 0.5123

# P-Value [Acc > NIR] : <2e-16

#

# Kappa : 0.7305

#

# Mcnemar's Test P-Value : 0.1799

#

# Sensitivity : 0.8430

# Specificity : 0.8867

# Pos Pred Value : 0.8763

# Neg Pred Value : 0.8558

# Prevalence : 0.4877

# Detection Rate : 0.4111

# Detection Prevalence : 0.4691

# Balanced Accuracy : 0.8649

#

# 'Positive' Class : 1

#Error rate of random forest

plot**(**rf**)**

#The below code in comments is useful if we need to fine tune the model for any reasons.

# Tune mtry

# t <- tuneRF(trainrandom[, -7], trainrandom[,7],

# stepFactor = 0.5,

# plot = TRUE,

# ntreeTry = 100,

# trace = TRUE,

# improve = 0.05)

#With the tuned values the model did not improve the accuracy on test data.

#No. of nodes for the trees

hist**(**treesize**(**rf**)**,

main **=** "No. of rows for the Trees",

col **=** "green"**)**

#variable Importance

varImpPlot**(**rf,

sort **=** T,

n.var **=** 4,

main **=** "Top 4 variable importance"**)**

importance**(**rf**)**

# 1 3 MeanDecreaseAccuracy MeanDecreaseGini

# tmax\_max 45.650799 52.70335 67.03800 224.41193

# ZONE 17.797351 31.51845 38.14676 37.85366

# rad\_max 11.229667 23.39180 27.17204 104.99514

# ah\_max 46.339293 46.09225 62.05836 144.13088

# rh\_max 10.744726 24.97613 28.08973 57.66704

# rain\_mean 7.649185 16.21476 17.84215 52.70118

#MeanDecreaseAccuracy = This graph tests how worse the model performs without each variable.

#tmax\_max and ah\_max imp variables however rain-mean contributed negligence importance

#MeanDecresaseGini = This graph measures how pure the nodes are at the end of the tree without each variable

#tmax\_max and ah\_max stand among all variables

#partial Dependence plot

partialPlot**(**rf, trainrandom, tmax\_max, "1"**)**

partialPlot**(**rf, trainrandom, tmax\_max, "3"**)**

partialPlot**(**rf, trainrandom, ah\_max, "1"**)**

partialPlot**(**rf, trainrandom, ah\_max, "3"**)**

#Extract single tree

getTree**(**rf, 5, labelVar **=** **TRUE)**

# Multi-dimensional scaling plot of proximity matrix

MDSplot**(**rf, trainrandom**$**Category**)**

#5. Naive Bayes Model

library**(**naivebayes**)**

library**(**dplyr**)**

library**(**ggplot2**)**

library**(**psych**)**

#Pre pare the data to build model

imp.df **<-** hospWeather.df**[**, c**(**'tmax\_max', 'ZONE','rad\_max', 'ah\_max', 'rh\_max', 'rain\_mean', 'Category'**)]**

imp.df1.naive **<-** subset**(**imp.df, Category**!=**2**)**

str**(**imp.df1.naive**)**

#Convert the values of Category variable into binary 0, 1

imp.df1.naive**$**Category**[**imp.df1.naive**$**Category **==** 1**]<-** 0

imp.df1.naive**$**Category**[**imp.df1.naive**$**Category **==** 3**]<-** 1

xtabs**(~**Category**+**ZONE, data **=** imp.df1.naive**)**

#Scale the data to normalize the variables

scale**(**imp.df1.naive, center **=** **TRUE**, scale **=** **TRUE)**

#Change the category variable into factor variable

imp.df1.naive**$**Category **<-** as.factor**(**imp.df1.naive**$**Category**)**

str**(**imp.df1.naive**)**

#Visualization

pairs.default**(**imp.df1.naive**[-**7**])**

#Box plots for all variables vs Category

imp.df1.naive %>% ggplot**(**aes**(**x **=** Category, y **=** tmax\_max, fill **=** Category**))** **+**

geom\_boxplot**()** **+**

ggtitle**(**"Maximum Temperature VS Influenza cases"**)**

# the plot shows that where the tmax\_max values are higher, the cases are tend to be low

# and where the tmax\_max values are lower the cases are high.

#This tells us we could build a classification model using this data.

imp.df1.naive %>% ggplot**(**aes**(**x **=** Category, y **=** ZONE, fill **=** Category**))** **+**

geom\_boxplot**()** **+**

ggtitle**(**"Zone VS Influenza cases"**)**

imp.df1.naive %>% ggplot**(**aes**(**x **=** Category, y **=** rad\_max, fill **=** Category**))** **+**

geom\_boxplot**()** **+**

ggtitle**(**"Maximum Radiation VS Influenza cases"**)**

imp.df1.naive %>% ggplot**(**aes**(**x **=** Category, y **=** ah\_max, fill **=** Category**))** **+**

geom\_boxplot**()** **+**

ggtitle**(**"Absolute Humidity VS Influenza cases"**)**

imp.df1.naive %>% ggplot**(**aes**(**x **=** Category, y **=** rh\_max, fill **=** Category**))** **+**

geom\_boxplot**()** **+**

ggtitle**(**"Relative Humidity VS Influenza cases"**)**

imp.df1.naive %>% ggplot**(**aes**(**x **=** Category, y **=** rain\_mean, fill **=** Category**))** **+**

geom\_boxplot**()** **+**

ggtitle**(**"Rain VS Influenza cases"**)**

#All the above plots are indicating that the different values of different variables

#are contributing to differentiate the number of cases into 'low - 0' and 'high - 1' category.

#This means we could potentially classify the number of cases based on these weather variables.

#Density Plot of ZONE

imp.df1.naive %>% ggplot**(**aes**(**x **=** ZONE, fill **=** Category**)** **)** **+**

geom\_density**(**alpha **=** 0.8, color **=** 'black'**)** **+**

ggtitle**(**"Density Plot of ZONE"**)**

# We could also notice Zones 1-4 have more lower cases than the other zones

#But the significant amount of data is overlapping between zones.

#Density Plot of tmax\_max

imp.df1.naive %>% ggplot**(**aes**(**x **=** tmax\_max, fill **=** Category**)** **)** **+**

geom\_density**(**alpha **=** 0.8, color **=** 'black'**)** **+**

ggtitle**(**"Density Plot of tmax\_max"**)**

#Density Plot of rad\_max

imp.df1.naive %>% ggplot**(**aes**(**x **=** rad\_max, fill **=** Category**)** **)** **+**

geom\_density**(**alpha **=** 0.8, color **=** 'black'**)** **+**

ggtitle**(**"Density Plot of rad\_max"**)**

#Density Plot of ah\_max

imp.df1.naive %>% ggplot**(**aes**(**x **=** ah\_max, fill **=** Category**)** **)** **+**

geom\_density**(**alpha **=** 0.8, color **=** 'black'**)** **+**

ggtitle**(**"Density Plot of ah\_max"**)**

#Density Plot of rh\_max

imp.df1.naive %>% ggplot**(**aes**(**x **=** rh\_max, fill **=** Category**)** **)** **+**

geom\_density**(**alpha **=** 0.8, color **=** 'black'**)** **+**

ggtitle**(**"Density Plot of rh\_max"**)**

#Density Plot of rain\_mean

imp.df1.naive %>% ggplot**(**aes**(**x **=** rain\_mean, fill **=** Category**)** **)** **+**

geom\_density**(**alpha **=** 0.8, color **=** 'black'**)** **+**

ggtitle**(**"Density Plot of rain\_mean"**)**

#Most of the above plots clearly are making distinction between high and low

#number of cases. This shows the potential to build a model but the overlap

#indicates that the model might not be 100% accurate.

#Data Partition into train:60% - Valid:40% split

set.seed**(**1234**)**

ind **<-** sample**(**2, nrow**(**imp.df1.naive**)**, replace **=** T, prob **=** c**(**0.6,0.4**))**

trainnb **<-** imp.df1.naive**[**ind **==**1,**]**

testnb **<-** imp.df1.naive**[**ind **==**2,**]**

#Build Naive Bayes Model using trainnb data

model **<-** naive\_bayes**(**Category **~**., data **=** trainnb, usekernel **=** T**)**

model

plot**(**model**)**

#Predict

p **<-** predict**(**model, trainnb, type **=** 'prob'**)**

head**(**cbind**(**p, trainnb**))**

#confusion matrix - train data

p1 **<-** predict**(**model, trainnb**)**

**(**tab1 **<-** table**(**p1, trainnb**$**Category**))**

# 0 1

# 0 474 83

# 1 105 603

1 **-** sum**(**diag**(**tab1**))** **/** sum**(**tab1**)**

#[1] 0.1486166

#mis-classification rate is 14.86%

#Accuracy is 85.13%

#confusion matrix and predict the model on test data

predict\_nb **<-** predict**(**model, testnb**)**

confusionMatrix**(**predict\_nb, testnb**$**Category**)**

# Confusion Matrix and Statistics

#

# Reference

# Prediction 0 1

# 0 300 55

# 1 73 380

#

# Accuracy : 0.8416

# 95% CI : (0.8146, 0.8661)

# No Information Rate : 0.5384

# P-Value [Acc > NIR] : <2e-16

#

# Kappa : 0.6802

#

# Mcnemar's Test P-Value : 0.1329

#

# Sensitivity : 0.8043

# Specificity : 0.8736

# Pos Pred Value : 0.8451

# Neg Pred Value : 0.8389

# Prevalence : 0.4616

# Detection Rate : 0.3713

# Detection Prevalence : 0.4394

# Balanced Accuracy : 0.8389

#

# 'Positive' Class : 0

#Iteration: 4 - Select the best model

# Choose the final model out of the 5 modls built according to the confusion matrix results.

#Make table with P-value, sensitivity, specificity, and balanced accuracy

#of all models to find out which model will best suit to predict influenza hospitalizations in New Zealand.

models **<-** data.frame **(**Technique **=** c**(**"Logistic\_Regression", "KNN", "CART", "Random\_Forest", "Naive\_Bayes"**)**,

Accuracy **=** c**(**0.8902, 0.87, 0.8329, 0.8654, 0.8416**)**,

P\_Value **=** c**(**2e**-**16, 2e**-**16, 2e**-**16, 2e**-**16, 2e**-**16**)**,

Sensitivity **=** c**(**0.8717, 0.9149, 0.8338, 0.8430, 0.8043**)**,

Specificity **=** c**(**0.9055, 0.8177, 0.8322, 0.8867, 0.8736**)**,

POS\_Pr\_Value **=** c**(**0.8835, 0.8541, 0.8099, 0.8763, 0.8451**)**,

Neg\_Pr\_Value **=** c**(**0.8957, 0.8918, 0.8538, 0.8558, 0.8389**)**,

Balanced\_Accuracy **=** c**(**0.8886, 0.8663, 0.8330, 0.8649, 0.8389**))**

models

library**(**reactable**)**

reactable**(**models, defaultColDef **=** colDef**(**align **=** "center"**)**, bordered **=** **TRUE**,

columns **=** list**(**"Accuracy" **=** colDef**(**format **=** colFormat**(** percent **=** **TRUE))**,

"P\_Value" **=** colDef**(**format **=** colFormat**(**percent **=** **TRUE))**,

"Sensitivity" **=** colDef**(**format **=** colFormat**(**percent **=** **TRUE))**,

"Specificity" **=** colDef**(**format **=** colFormat**(**percent **=** **TRUE))**,

"POS\_Pr\_Value" **=** colDef**(**format **=** colFormat**(**percent **=** **TRUE))**,

"Neg\_Pr\_Value" **=** colDef**(**format **=** colFormat**(**percent **=** **TRUE))**,

"Balanced\_Accuracy" **=** colDef**(**format **=** colFormat**(**percent **=** **TRUE))**

**))**

# Based on the confusion matrix metrics, Logistic regression is the model with 89.02% accuracy.

# So Logistic regression is the final model that could produce quality predictions on Influenza-

#-occurence in New Zealand