Configuration Manual

MSc Research Project
Net-Migration in Relation to Incidence of Cystic Fibrosis in Ireland

Fergal Bell
Student ID: X18119115

School of Computing
National College of Ireland

Supervisor: Dr Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Fergal Bell
Student ID:	18119115
Programme:	MSc Data Analytics - Research Project **Year:** 2020
Module:	
Lecturer:	Dr Catherine Mulwa
Submission Due Date:	Monday 17th August
Project Title:	Net-Migration in Relation to Incidence of Cystic Fibrosis in Ireland
Word Count:	3,313 **Page Count:** 35

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Fergal Bell

Date: 16th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Forename Surname

Contents

Student ID: 18119115

1 Introduction

This configuration manual helps the users understand how the results of the project were implemented, what artefacts were developed in the project, the software used and how to implement in detail the various solutions to come to the findings and conclusion in Net-Migration in relation to Incidence in Cystic Fibrosis in Ireland. Such components as the application environment, file storage and configurations, hardware used.

This manual is supplementary to the technical report and should be read in conjunction with the report as guidance to the technical aspects employed.

2 Application Environment

The artefacts of this report were performed in the following environment.

2.1 Hardware

- Device Name: Laptop-AER3Q2L6
- Processor: Intel®, Core™ i5-7200 CPU @ 2.50GHz 2.70GHz
- Installed RAM: 8GB (7.87 GB usable)
- System Type: 64-bit operating system, x64-based processor
- 256 GB SSD

2.2 Software

2.2.1 RStudio

RStudio is an open source statistical environment platform for data analysts. Version 1.3.959 was used in this project. There is a plethora of libraries to avail from. They are either ready installed or can be downloaded from CRAN (Comprehensive R Archive Network)[1]. There are also lots of online courses at a reasonable cost to learn some valuable skills in RStudio – such as Udemy[2]. The following libraries were used in this project:

- RStudio Libraries used across artefacts:
  a) rpart – recursive partitioning and decision trees
  b) randomForest – used to implement a collection of decision trees for classification purposes
  c) KNN – K Nearest Neighbours which uses Euclidean distances between feature variables. Used for classification and regression

[1] https://cran.r-project.org/

[2] https://www.udemy.com/?utm_source=adwords-brand&utm_medium=udemyads&utm_campaign=Brand-Udemy_la.EN_cc.ROW&utm_term=_._ag_80315195513_._ad_450687451854_._de_c_._dm__._pl__._ti_kwd-310556426868_._li_1007877_._pd__._&utm_term=_._pd__._kw_udemy_._&matchtype=e&gclid=CjwKCAjw4MP5BRBtEiwASfwAL7hwZwLaPtsTvdPM1whE-r3MIerj7T1oFHxtThecowqwAGOyWZXNfhoCoP4QAvD_BwE

d) Naïve Bayes – a classification algorithm based on Bayes Theorem of conditional probabilities
e) SVM – Support Vector Machine is a supervised classification and regression models
f) Kernel SVM – is a supervised classification model using classification algorithms and the kernel trick[3]
g) dplyr – manipulating datasets
h) e1071 – used for SVM, Naïve Bayes
i) ggplot2 – data visualisation package
j) catools – using statistical functions, sample. split, etc.
k) caret – creating a confusion matrix

2.2.2 SPSS

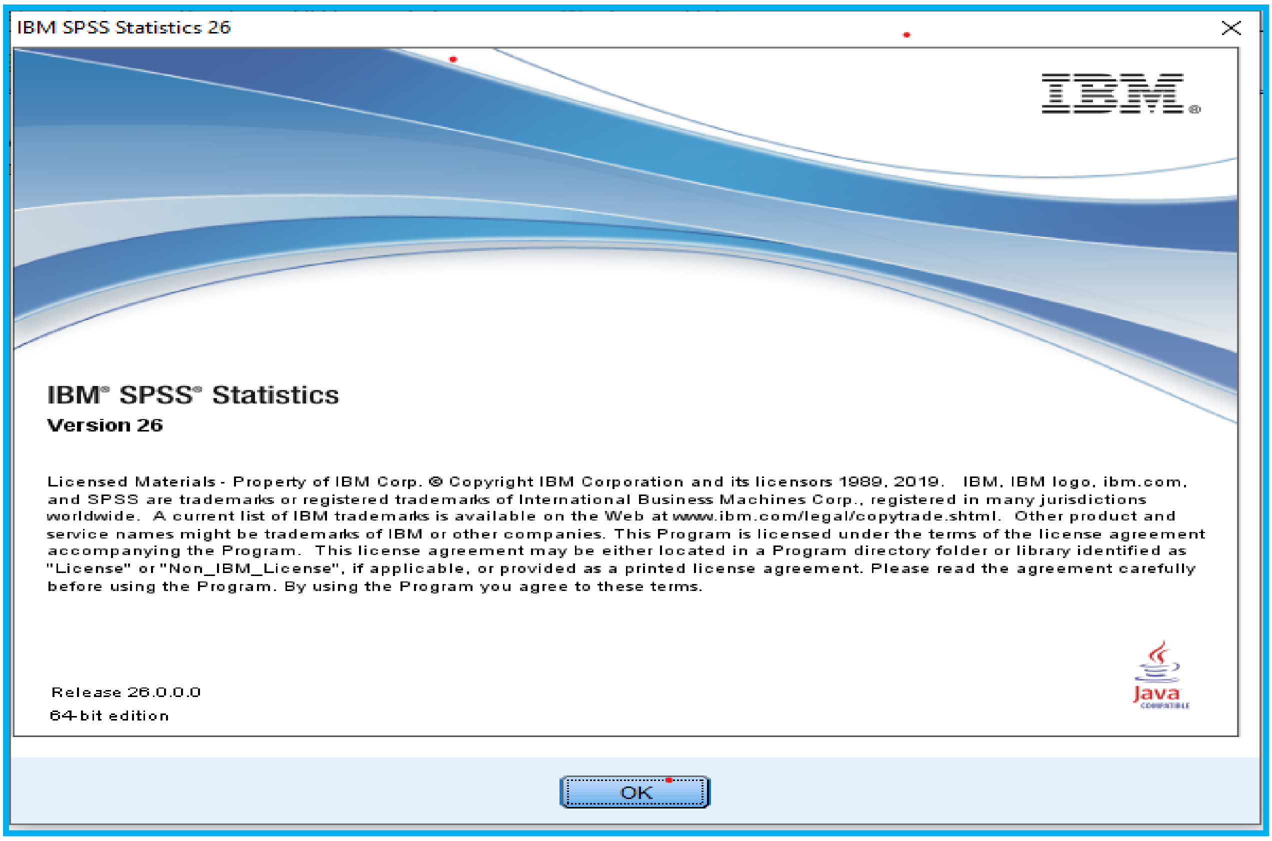

Figure 1: SPSS Version

SPSS version 26 was used in this project for PCA[4] analysis, multiple linear regression, histograms, scatterplots, normality, and other statistical tools. SPSS Survivor Manual was used as a reference for implementing some of the statistical techniques (Pallant, J, 2016).

---

[3] https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f#:~:text=The%20%E2%80%9Ctrick%E2%80%9D%20is%20that%20kernel,the%20data%20by%20these%20transformed

[4] https://en.wikipedia.org/wiki/Principal_component_analysis

2.2.3 Excel

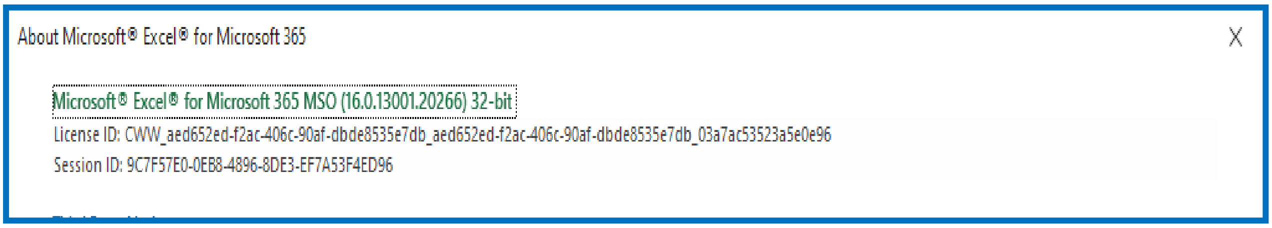

Figure 2: Version of Excel

Microsoft Excel for 365 was used in this project to store the dataset, transform the dataset to csv format that would be suitable for RStudio analysis. All data was manually inputted, and cells auto populated via VLOOKUP, index and match formulas, and other formulas which speeded up the process immensely. Help was gained via YouTube tutorials and excelchat[5]

3 Application Artefacts

3.1 Data Extraction

Excel csv file -CF Finalv7.csv

3.1.1 RStudio

3.1.2 Naïve Bayes

Dataset was extracted from excel csv file, as follows:

```
# Importing the dataset
dataset = read.csv('CF Finalv7.csv')
```

Figure 3: Importing the dataset

Changing the names of the features to a more readable format

[5] https://expert.excelchat.co/

```r
# Changing the names of the features
names(dataset)
names(dataset)[1] = "Year"
names(dataset)
names(dataset)[4] = "< 5"
names(dataset)[53] = "F508_Homo"
names(dataset)[54] = "F508_Hetero"
names(dataset)[55] = "F208"

library(dplyr)
dataset = rename(dataset, "Twenty_TwentyFour" = x20.24)
dataset = rename(dataset, "Five to Nine" = x5.9)
dataset = rename(dataset, "Ten_Fourteen" = x10.14)
dataset = rename(dataset, "Fifteen to Nineteen" = x15.19)
dataset = rename(dataset, "TwentyFive_TwentyNine" = x25.29)
dataset = rename(dataset, "Thirty to Thirty Four" = x30.34)
dataset = rename(dataset, "Thirty Five to Thity Nine" = x35.39)
dataset = rename(dataset, "Forty to Forty Four" = x40.44)
dataset = rename(dataset, "Forty Five to Forty Nine" = x45.49)
dataset = rename(dataset, "Greater than Fifty" = x.50)
```

Figure 4: Changing the names of the features

```r
# Choosing Important Variables
dataset = dataset[, c("Incidence","< 5")]
dataset = dataset[, c("Incidence","Five to Nine")]
dataset = dataset[, c("Incidence","Twenty_TwentyFour")]
dataset = dataset[, c("Incidence","TwentyFive_TwentyNine")]
dataset = dataset[, c("Incidence","Thirty to Thirty Four")]
dataset = dataset[, c("Incidence","Ten_Fourteen")]
dataset = dataset[, c("Incidence","Fifteen to Nineteen")]
dataset = dataset[, c("Incidence","Other")]
dataset = dataset[, c("Incidence","Other", "Age")]
dataset = dataset[, c("Incidence","Year.of.Birth")]
dataset = dataset[, c("Incidence","Other","Year.of.Birth")]
dataset = dataset[, c("Incidence","F508_Homo")]
dataset = dataset[, c("Incidence","F508_Homo", "Age")]
dataset = dataset[, c("Incidence","F508_Homo", "Year.of.Birth")]
dataset = dataset[, c("Incidence","F508_Hetero")]
dataset = dataset[, c("Incidence","F508_Hetero", "Age")]
dataset = dataset[, c("Incidence","F508_Hetero", "Year.of.Birth")]
dataset = dataset[, c("Incidence","F208")]
dataset = dataset[, c("Incidence","No.of.Other.Ethnicity")]
dataset = dataset[, c("Incidence","Thirty Five to Thity Nine")]
dataset = dataset[, c("Incidence","Forty to Forty Four")]
dataset = dataset[, c("Incidence","Greater than Fifty")]
dataset = dataset[, c("Incidence","Forty Five to Forty Nine")]
dataset = dataset[, c("Incidence","Year")]
names(dataset)
str(dataset)
```

Figure 5: Choosing important features

Figure 5 illustrates the features that were chosen in relation to incidence of cystic fibrosis. By choosing each dataset in turn, which can then be inputted into the model one at a time.

```r
# Encoding the target feature as factor
dataset$Incidence.of.CF.Yearly = as.factor(dataset$Incidence.of.CF.Yearly)
dataset$Incidence = as.factor(dataset$Incidence)

# Splitting the dataset into the Training set and Test set
install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Incidence, SplitRatio = 0.80)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Feature Scaling
training_set[-1] = scale(training_set[-1])
test_set[-1] = scale(test_set[-1])

# Fitting Naive Bayes to the Training set
library(e1071)
classifier = naiveBayes(x = training_set[-1],
                        y = training_set$Incidence)
# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-1])
```

Figure 6: Encoding target feature, splitting the dataset, feature scaling, fitting Naïve Bayes to the training set, and predicting the result.

Figure 6 illustrates how the Naïve Bayes model was implemented by first of all encoding the feature variable incidence as a factor. Then 'catools' package is used to split the data into a 80:20 ratio, (Tuszynski., 2020) feature scaling was then applied to the training set and test set. Next, fit Naïve Bayes to the model by calling the library e1071 (Meyer et al., 2019) and finally using the predict function to predict[6] the results.

```r
# Confusion Matrix

install.packages("caret")
library(caret)
cm = table(test_set[, 1], y_pred)
results = confusionMatrix(y_pred, test_set[,1])
print(results)
results = confusionMatrix(test_set[,1], y_pred)
length(y_pred)
sum(diag(cm))/sum(cm)
1-sum(diag(cm))/sum(cm)
```

Figure 7: Confusion Matrix

Figure 7 represents how the Confusion Matrix was created using the 'caret' package (Kuhn, 2019) – an example is shown in figure 8: for the dependent variable -index 1- Incidence against age group Ten to Fourteen.

[6] http://www.sthda.com/english/articles/40-regression-analysis/166-predict-in-r-model-predictions-and-confidence-intervals/

```
Confusion Matrix and Statistics

          Reference
Prediction 0.199 0.426 0.571 0.684 0.906 0.923 0.963 1.041 1.046
     0.199     0     0     0     0     0     0     0     0     0
     0.426     0   200     0     0     0     0     0     0     0
     0.571     0     0   253     0     0     0     0     0     0
     0.684   206     0     0   244     0     0     0     0     0
     0.906     0     0     0     0   237     0     0     0     0
     0.923     0     0     0     0     0     0     0     0     0
     0.963     0     0     0     0     0   209   227     0     0
     1.041     0     0     0     0     0     0     0   231     0
     1.046     0     0     0     0     0     0     0     0   219

Overall Statistics

               Accuracy : 0.7952
                 95% CI : (0.7769, 0.8125)
    No Information Rate : 0.1249
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7687
```

Figure 8: Confusion Matrix for Naïve Bayes.

3.1.3 Decision Tree Regression

Dataset was extracted from excel csv file, as follows:

```
# Importing the dataset
dataset = read.csv('CF FinalV7.csv')
```

Renaming the important features and changing data types.

```
library(dplyr)
dataset = rename(dataset, "Twenty_TwentyFour" = X20.24)
dataset = rename(dataset, "Five to Nine" = X5.9)
dataset = rename(dataset, "Ten_Fourteen" = X10.14)
dataset = rename(dataset, "Fifteen to Nineteen" = X15.19)
dataset = rename(dataset, "TwentyFive_TwentyNine" = X25.29)
dataset = rename(dataset, "Thirty to Thirty Four" = X30.34)
dataset = rename(dataset, "Thirty Five to Thity Nine" = X35.39)
dataset = rename(dataset, "Forty to Forty Four" = X40.44)
dataset = rename(dataset, "Forty Five to Forty Nine" = X45.49)
dataset = rename(dataset, "Greater than Fifty" = X.50)
dataset$Population = as.numeric(dataset$Population)
dataset$F508_Homo = as.numeric(dataset$F508_Homo)
dataset$F508_Hetero = as.numeric(dataset$F508_Hetero)
dataset$F208 = as.numeric(dataset$F208)
dataset$Other = as.numeric(dataset$Other)
dataset$Prevalence = as.numeric(dataset$Prevalence)
dataset$Incidence = as.numeric(dataset$Incidence)
dataset$No.of.Other.Ethnicity = as.numeric(dataset$No.of.Other.Ethnicity)
dataset$No.of.Irish.Ethnicity = as.numeric(dataset$No.of.Irish.Ethnicity)
dataset$Severity.of.Condition = NULL
dataset$Incidence.of.CF.Yearly = as.integer(dataset$Incidence.of.CF.Yearly)
```

Figure 9: Changing names of the dataset and amending datatypes.

Figure 9 represents the changing of variable names and datatypes for the Decision Tree classification model.

```r
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Incidence, SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Feature Scaling
training_set[-57] = scale(training_set[-57])
test_set[-57] = scale(test_set[-57])

# Fitting classifier to the Training set
library(rpart)
classifier = rpart(formula = Incidence ~
                   Ten_Fourteen,
                   data = training_set)

# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-57], type = 'vector')
str(dataset)
names(dataset)
```

Figure 10: Splitting the dataset, feature scaling and fitting classifier to Decision Tree Model

Figure 10 represents splitting the dataset into a training and test set ratio of 75:25. Feature scaling was implemented (although not necessary as Euclidean distances are not important in this type of modelling). The rpart package (Breiman,1984) was called by the RStudio library to create the classifier. Finally, the predict function was called to predict the model.

```r
# Confusion Matrix

y_pred = as.factor(y_pred)
test_set[, 57] = as.factor(test_set[, 57])

install.packages("caret")
library(caret)
results = confusionMatrix(y_pred, test_set[,57])
print(results)
```

Figure 11: Confusion Matrix for Decision Tree Model.

Figure 11 illustrates how the Confusion Matrix was created using the ‘caret’ package (Kuhn, 2019).

3.1.4 Random Forest

Dataset was extracted as follows.

```r
# Importing the dataset
dataset = read.csv('CF FinalV7.csv')
```

Changing variable names and installing ‘dplyr’

```
is.na(dataset)
sum(is.na(dataset))
names(dataset)
names(dataset)[1] = "Year"
names(dataset)[5] = "Five_Nine"
names(dataset)[4] = "less_5"
names(dataset)[53] = "F508_Homo"
names(dataset)[54] = "F508_Hetero"
names(dataset)[55] = "F208"

install.packages("dplyr")
```

Figure 12: Variable Name Changes

```
library(randomForest)

library(dplyr)
dataset = rename(dataset, "Twenty_TwentyFour" = x20.24)
dataset = rename(dataset, "Five_Nine" = x5.9)
dataset = rename(dataset, "Ten_Fourteen" = x10.14)
dataset = rename(dataset, "Fifteen_Nineteen" = x15.19)
dataset = rename(dataset, "TwentyFive_TwentyNine" = x25.29)
dataset = rename(dataset, "Thirty_Thirty_Four" = x30.34)
dataset = rename(dataset, "ThirtyFive_ThityNine" = x35.39)
dataset = rename(dataset, "Forty_Forty_Four" = x40.44)
dataset = rename(dataset, "FortyFive_FortyNine" = x45.49)
dataset = rename(dataset, "Greater_Fifty" = x.50)
dataset$Population = as.numeric(dataset$Population)
dataset$F508_Homo = as.numeric(dataset$F508_Homo)
dataset$F508_Hetero = as.numeric(dataset$F508_Hetero)
dataset$F208 = as.numeric(dataset$F208)
dataset$Other = as.numeric(dataset$Other)
dataset$Prevalence = as.numeric(dataset$Prevalence)
dataset$Incidence = as.numeric(dataset$Incidence)
dataset$No.of.Other.Ethnicity = as.numeric(dataset$No.of.Other.Ethnicity)
dataset$No.of.Irish.Ethnicity = as.numeric(dataset$No.of.Irish.Ethnicity)
```

Figure 13: Changing variable names, datatypes, calling randomForest

Figure 13 illustrates name changing of features, changing the datatypes of several of the features, and calling randomForest[7] classifier via RStudio library.

[7] https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

```
#Removing Variables
dataset$Severity.of.Condition = NULL
dataset$`less than 5` = NULL
dataset$Incidence.of.CF.Yearly = NULL
dataset$No.of.Irish.Ethnicity = NULL
dataset$No.of.Other.Ethnicity= NULL
dataset$`less than 5` = NULL
dataset$Prevalence = NULL
dataset$Children.High = NULL
dataset$Children.Moderate = NULL
dataset$Children.Low = NULL
dataset$All.Moderate = NULL
dataset$All.Low = NULL
dataset$All.High = NULL
dataset$X45.49 = NULL
dataset$Number.Deceased.End.of.year = NULL
dataset$No.of.Males...18= NULL
dataset$Median.Age..at.Death = NULL
dataset$Year= NULL
dataset$X30.34= NULL
dataset$Alive.F= NULL
dataset$X.Male = NULL
dataset$X...18yrs= NULL
```

Figure 14: Removing Variables

Figure 14 illustrates a collection of variables that can be removed from the randomForest model if needed.

```r
# Encoding the target feature as factor

dataset$Incidence = as.factor(dataset$Incidence)

str(dataset)

# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Incidence, SplitRatio = 0.80)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
forest = randomForest(Incidence ~., data = training_set, na.action = na.omit,
                      ntree =100, importance = T)

importance(forest)

# Variable Importance Plot
varImpPlot(forest, pch = 16, col = "blue", cex = 1)
```

Figure 15: Encoding, Splitting the Dataset, Creating Random Forest model.

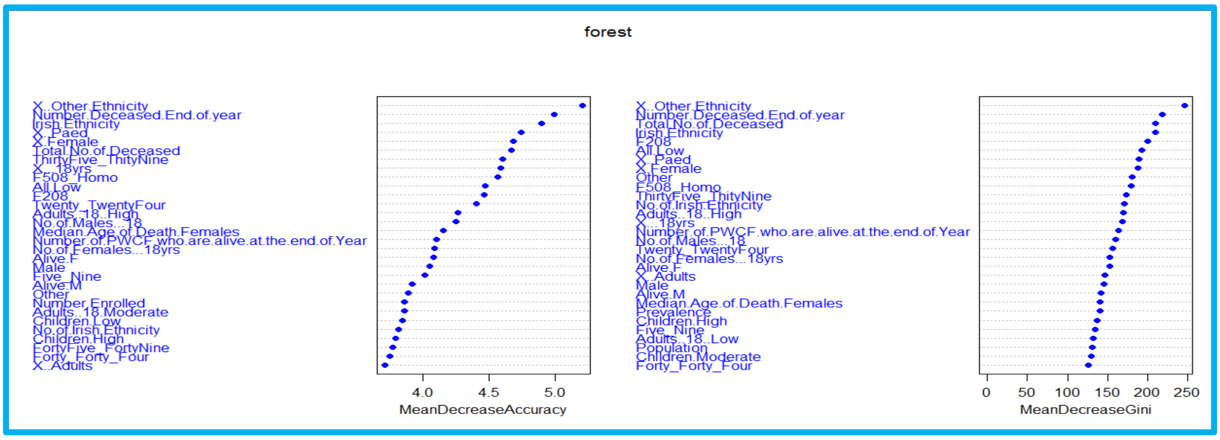

Figure 16: Variable Importance Plot

```r
# Variable Importance Plot
varImpPlot(forest, pch = 16, col = "blue", cex = 1)
```

Figure 17: Variable Importance Code.

Figure 16 and 17 show the variable importance graph[8] and code respectively for random forest. The graph basically shows the importance of each predicter of the dataset by two measures: mean decrease in Gini[9] and mean decrease in accuracy of the predictors.

---

[8] https://rdrr.io/cran/randomForest/man/varImpPlot.html

[9] https://stats.stackexchange.com/questions/197827/how-to-interpret-mean-decrease-in-accuracy-and-mean-decrease-gini-in-random-fore

The Mean Decrease Gini is a chart which shows variable importance based on the Gini impurity index. Other ethnicity reported the highest level of variable importance in the Mean Decrease Gini – caveat: this variable showed high multicollinearity with other independent variables.
It is the Mean Decrease Accuracy chart that showed up some surprising results. The chart shows how much of the model accuracy will decrease if a variable is dropped. In other words, it is another measure of how important a variable is.

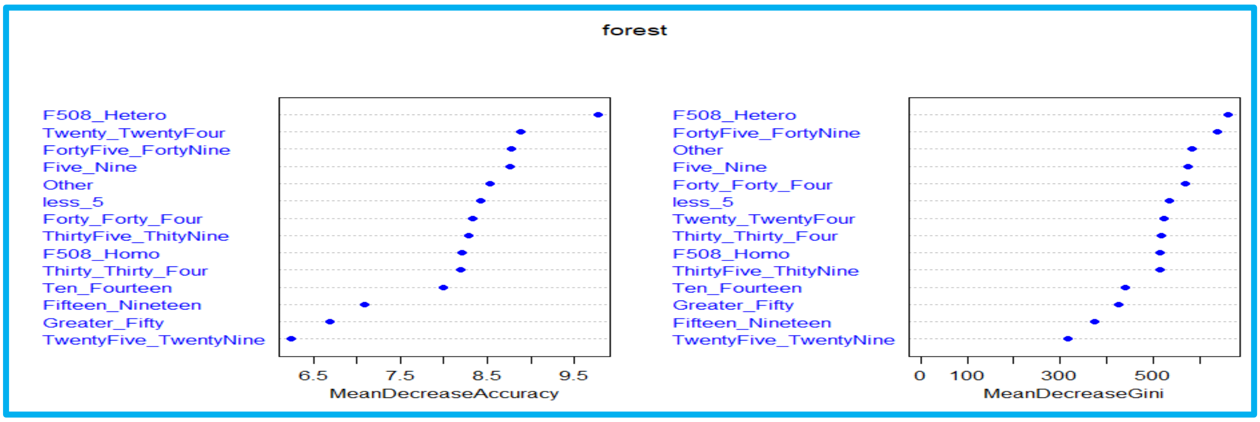

Figure 18: Variable Importance Plot for Genotypes & Age Groups

In figure 18, by excluding all the variables that have a high multicollinearity the variable plot displays a different picture. Genotype, F508_Hetero showed the highest level of Mean decrease in Gini index. Whereas, the mean decrease in accuracy shows how much the model accuracy will drop.

```
rf = predict(forest, test_set, type = "class")
confusionMatrix(rf, test_set$Incidence, positive = "Yes")

# Feature Scaling
training_set[-1] = scale(training_set[-1])
test_set[-1] = scale(test_set[-1])

# Fitting Random Forest Classification to the Training set
#install.packages("randomForest")
library(randomForest)
classifier = randomForest(x = training_set[-1],
                          y = training_set$Incidence,
                          ntree = 100)
# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-1])
```

Figure 19: Random Forest Classification Code

Figure 19 displays the code used to create the random forest classification model, as follows: predict function passed on to the forest classifier, feature scaling, fitting random classification to the training set.

```
# Making the Confusion Matrix
cm = table(test_set[, 1], y_pred)
sum(diag(cm))/sum(cm)
1-sum(diag(cm))/sum(cm)

install.packages("caret")
library(caret)
results = confusionMatrix(y_pred, test_set[,1])
print(results)
```

Figure 20: Confusion Matrix for Random Forest Model.

Figure 20 illustrates how the Confusion Matrix was created using the ‘caret’ package (Kuhn, 2019) for the test set.

3.1.5 KNN

```
# KNN

# Importing the dataset
dataset = read.csv('Cf Finalv7.csv')
names(dataset)
names(dataset)[1] = "Year"
names(dataset)
names(dataset)[4] = "< 5"
names(dataset)[53] = "F508_Homo"
names(dataset)[54] = "F508_Hetero"
names(dataset)[55] = "F208"
```

Figure 21: Importing the Dataset for KNN

The above figure shows the code for importing the dataset (cf Finalv7.csv) and renaming some of the features.

```
library(dplyr)
dataset = rename(dataset, "Twenty_TwentyFour" = X20.24)
dataset = rename(dataset, "Five to Nine" = X5.9)
dataset = rename(dataset, "Ten_Fourteen" = X10.14)
dataset = rename(dataset, "Fifteen to Nineteen" = X15.19)
dataset = rename(dataset, "TwentyFive_TwentyNine" = X25.29)
dataset = rename(dataset, "Thirty to Thirty Four" = X30.34)
dataset = rename(dataset, "Thirty Five to Thity Nine" = X35.39)
dataset = rename(dataset, "Forty to Forty Four" = X40.44)
dataset = rename(dataset, "Forty Five to Forty Nine" = X45.49)
dataset = rename(dataset, "Greater than Fifty" = X.50)
```

Figure 22: Renaming variables using the dplyr package.

Figure 22 shows the renaming of variables using the dplyr package[10] - dplyr is a new package which provides a set of tools for efficiently manipulating datasets in R.

[10] https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html

```r
# Choosing Important Variables
dataset = dataset[, c("Incidence", "Ten_Fourteen", "less_5", "Five_Nine", "Fifteen_Nineteen",
                      "Thirty_Thirty_Four")]
dataset = dataset[, c("Incidence","< 5", "Year.of.Birth")]
dataset = dataset[, c("Incidence","Five to Nine", "Year.of.Birth")]
dataset = dataset[, c("Incidence","No.of.Other.Ethnicity")]
dataset = dataset[, c("Incidence","Twenty_TwentyFour", "Year.of.Birth")]
dataset = dataset[, c("Incidence","TwentyFive_TwentyNine", "Year.of.Birth")]
dataset = dataset[, c("Incidence","Thirty to Thirty Four","Year.of.Birth")]
dataset = dataset[, c("Incidence","Ten_Fourteen", "Year.of.Birth")]
dataset = dataset[, c("Incidence","Fifteen to Nineteen", "Year.of.Birth")]
dataset = dataset[, c("Incidence","Other")]
dataset = dataset[, c("Incidence","Other", "Age")]
dataset = dataset[, c("Incidence","Other","Year.of.Birth")]
dataset = dataset[, c("Incidence","Other","Year.of.Birth")]
dataset = dataset[, c("Incidence","F508_Homo")]
dataset = dataset[, c("Incidence","F508_Homo", "Age")]
dataset = dataset[, c("Incidence","F508_Homo", "Year.of.Birth")]
dataset = dataset[, c("Incidence","F508_Hetero", "Age")]
dataset = dataset[, c("Incidence","F508_Hetero", "Year.of.Birth")]
dataset = dataset[, c("Incidence","F208")]
dataset = dataset[, c("Incidence.of.CF.Yearly","Ten_Fourteen")]
dataset = dataset[, c("Incidence.of.CF.Yearly","Other")]
dataset = dataset[, c("Incidence.of.CF.Yearly","No.of.Other.Ethnicity")]
dataset = dataset[, c("Incidence.of.CF.Yearly","X..Other.Ethnicity")]
dataset = dataset[, c("Incidence.of.CF.Yearly","TwentyFive_TwentyNine")]
dataset = dataset[, c("Incidence","Thirty Five to Thity Nine","Year.of.Birth")]
dataset = dataset[, c("Incidence","Forty to Forty Four", "Year.of.Birth")]
```

```r
dataset = dataset[, c("Incidence","Year")]
dataset = dataset[, c("Incidence", "Five to Nine")]
dataset = dataset[, c("Incidence","Year.of.Birth", "Five to Nine")]
dataset = dataset[, c("Incidence","Year.of.Birth")]
dataset = dataset[, c("Incidence","Population")]
names(dataset)
str(dataset)
dataset = dataset[, c("Incidence","Ten_Fourteen")]
dataset$`Five to Nine` = as.numeric(dataset$`Five to Nine`)
dataset$`Five to Nine` = as.factor(dataset$`Five to Nine`)
dataset$Incidence = as.factor(dataset$Incidence)
```

Figure 23: Choosing features for KNN Model

Figure 23 shows different variable combinations that were tried out in KNN.

```r
#Splitting the dataset into the Training set and Test set
install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Incidence, SplitRatio = 0.80)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Feature Scaling
training_set[-1] = scale(training_set[-1])
test_set[-1] = scale(test_set[-1])

# Fitting KNN to The Training Set and Predicting the test set result
library(class)
y_pred = knn(train = training_set[, -1],
             test = test_set[,-1],
             cl = training_set[, 1],
             k = 5)
```

Figure 24: Splitting dataset & Fitting the KNN Model

Figure 24 shows the code used to split the dataset into training and test sets using "catools" package from R library. Feature scaling which is important in KNN (Altman, 1991) due to the distance metric nature – all variables (apart from the target variable – incidence) are scaled to stop any one variable from dominating the model.

```
y_pred =  as.factor(y_pred)
test_set[, 1] = as.factor(test_set[, 1])
install.packages("caret")
library(caret)
results = confusionMatrix(y_pred, test_set[,1])
print(results)
```

Figure 25: KNN Confusion Matrix Code

```
Confusion Matrix and Statistics

          Reference
Prediction 0.199 0.426 0.571 0.684 0.906 0.923 0.963 1.041 1.046
     0.199   107     0     0    84     0     0     0     0     0
     0.426     0   200     0     0     0     0     0     0     0
     0.571     0     0   253     0     0     0     0     0     0
     0.684    99     0     0   160     0     0     0     0     0
     0.906     0     0     0     0   237     0     0     0     0
     0.923     0     0     0     0     0   135   122     0     0
     0.963     0     0     0     0     0    73   105     0     0
     1.041     0     0     0     0     0     0     0   230     0
     1.046     0     0     0     0     0     1     0     1   219

Overall Statistics

               Accuracy : 0.8124
                 95% CI : (0.7947, 0.8292)
    No Information Rate : 0.1249
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7889
```

Figure 26

The above confusion matrix is the result of two features – Ten to Fourteen & Year of Birth as the independent variables against the dependent variable – incidence of cystic fibrosis resulting in an accuracy of 81.24%.

3.1.6 SVM

```r
# Data Preprocessing Template
names(dataset)
names(dataset)[1] = "Year"
names(dataset)
names(dataset)[4] = "less_Five"
names(dataset)[53] = "F508_Homo"
names(dataset)[54] = "F508_Hetero"
names(dataset)[55] = "F208"
install.packages("dplyr")
library(dplyr)
dataset = rename(dataset, "Twenty_TwentyFour" = X20.24)
dataset = rename(dataset, "Five_Nine" = X5.9)
dataset = rename(dataset, "Ten_Fourteen" = X10.14)
dataset = rename(dataset, "Fifteen_Nineteen" = X15.19)
dataset = rename(dataset, "TwentyFive_TwentyNine" = X25.29)
dataset = rename(dataset, "Thirty_ThirtyFour" = X30.34)
dataset = rename(dataset, "ThirtyFive_ThityNine" = X35.39)
dataset = rename(dataset, "Forty_FortyFour" = X40.44)
dataset = rename(dataset, "FortyFive_FortyNine" = X45.49)
dataset = rename(dataset, "Greater_Fifty" = X.50)
dataset$Population = as.numeric(dataset$Population)
dataset$F508_Homo = as.numeric(dataset$F508_Homo)
dataset$F508_Hetero = as.numeric(dataset$F508_Hetero)
dataset$F208 = as.numeric(dataset$F208)
dataset$Other = as.numeric(dataset$Other)
dataset$Incidence = as.numeric(dataset$Incidence)
str(dataset)
names(dataset)
```

Figure 27: Preprocessing of Data

Figure 27 shows the preprocessing steps applied to the SVM model – renaming variables, changing datatypes, checking structure of dataset and names of variables.

```r
# Reducing Dataset to combinations of features
dataset = dataset[, c("Incidence.of.CF.Yearly","X..Other.Ethnicity")]
dataset = dataset[, c("Incidence.of.CF.Yearly","Population", "Age")]
dataset = dataset[, c("Incidence","Population")]
dataset = dataset[, c("Incidence","Net.Population")]
dataset = dataset[, c("Incidence","Ten_Fourteen")]
dataset = dataset[, c("Incidence","less_Five")]
dataset = dataset[, c("Incidence","Fifteen_Nineteen")]
dataset = dataset[, c("Incidence","Twenty_TwentyFour")]
dataset = dataset[, c("Incidence","TwentyFive_TwentyNine")]
dataset = dataset[, c("Incidence","Thirty_ThirtyFour")]
dataset = dataset[, c("Incidence","ThirtyFive_ThityNine")]
dataset = dataset[, c("Incidence","Forty_FortyFour")]
dataset = dataset[, c("Incidence","FortyFive_FortyNine")]
dataset = dataset[, c("Incidence","Greater_Fifty")]
dataset = dataset[, c("Prevalence","Population")]
dataset = dataset[, c("Incidence","Year")]
dataset = dataset[, c("Incidence","Five_Nine")]
dataset = dataset[, c("Incidence","F508_Homo", "Year.of.Birth")]
dataset = dataset[, c("Incidence","F508_Hetero", "Year.of.Birth")]
dataset = dataset[, c("Incidence","F508_Hetero")]
dataset = dataset[, c("Incidence","Other", "Year.of.Birth")]
dataset = dataset[, c("Incidence.of.CF.Yearly","Ten_Fourteen")]
str(dataset)
```

Figure 28: Reducing Dataset to Combinations of Features

Figure 28 shows the dataset being broken down in different combinations of variables for SVM modelling, if and when needed.

```
# Splitting the dataset into the Training set and Test set

install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset, SplitRatio = 0.8)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Changing & Testing Variabless
dataset$Incidence = as.factor(dataset$Incidence)
dataset$Incidence= as.integer(dataset$Incidence)
dataset$Incidence= as.numeric(dataset$Incidence)
dataset$Fifteen_Nineteen= as.numeric(dataset$Fifteen_Nineteen)
dataset$Age = as.numeric(dataset$Age)
str(dataset)
str(training_set)
str(test_set)
names(dataset)

# Feature Scaling
training_set[-1] = scale(training_set[-1])
test_set[-1] = scale(test_set[-1])
```

Figure 29: Modelling for SVM

Figure 29 shows modelling the dataset using the "caTools"[11] package from RStudio library. This entails splitting the dataset into a 80:20 ratio, changing variable types, and feature scaling.

[11] https://cran.r-project.org/web/packages/caTools/index.html

```
library(e1071)
classifier = svm(formula = Incidence.of.CF.Yearly ~ Year,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')

classifier = svm(formula = Incidence ~ Ten_Fourteen,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')

classifier = svm(formula = Incidence ~ Population,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')
classifier = svm(formula = Incidence ~ Net.Population,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')

classifier = svm(formula = Incidence.of.CF.Yearly ~ Twenty_TwentyFour,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'linear')
```

Figure 30: SVM Classifier

Figure 30 shows creating SVM classifiers for various combinations of variables using a linear kernel (Hofmann, Schölkopf and Smola, 2008) .

```
# Predicting a new result
y_pred = predict(classifier, newdata = test_set[-1])
cm = table(test_set[, 1], y_pred)
sum(diag(cm))/sum(cm)
1-sum(diag(cm))/sum(cm)
```

Figure 31: Confusion Matrix SVM

Figure 31 illustrates the code implemented to create a confusion matrix for SVM

```
install.packages("psych")
library(psych)
names(dataset)
pairs(dataset[c("Incidence", "< 5", "Five to Nine", "Ten_Fourteen", "Fifteen to Nineteen",
                "Twenty_TwentyFour", "TwentyFive_TwentyNine", "Thirty to Thirty Four",
                "Forty to Forty Four", "Forty Five to Forty Nine")])

pairs.panels(dataset[c("Incidence", "< 5", "Five to Nine", "Ten_Fourteen", "Fifteen to Nineteen",
                       "Twenty_TwentyFour", "TwentyFive_TwentyNine", "Thirty to Thirty Four",
                       "Forty to Forty Four", "Forty Five to Forty Nine")])
```

Figure 32: Code for Correlation Matrix

Figure 32 illustrates the code used to create a correlation matrix using the “psych” package from RStudio library. This code along with pairs.panels creates a visual correlation matrix with the variables along the diagonal, corelation coefficients to the right of the diagonal, and ellipses to the left – the more squeezed or squashed the elipse is the stronger the correlation between the variables – see figures 33, 34 and 35.

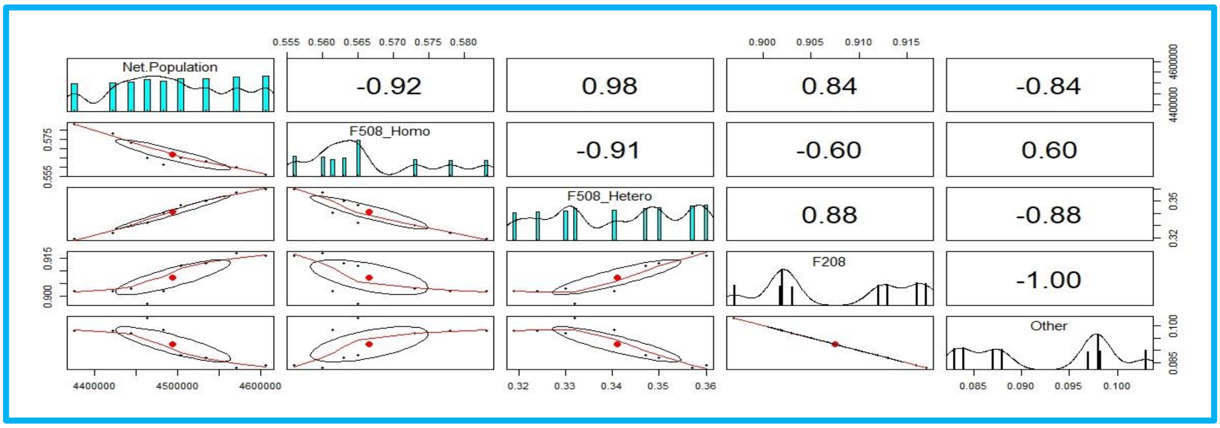

Figure 33: Correlation Matrix of Net Population[12] vs Genotypes from 2008 to 2016

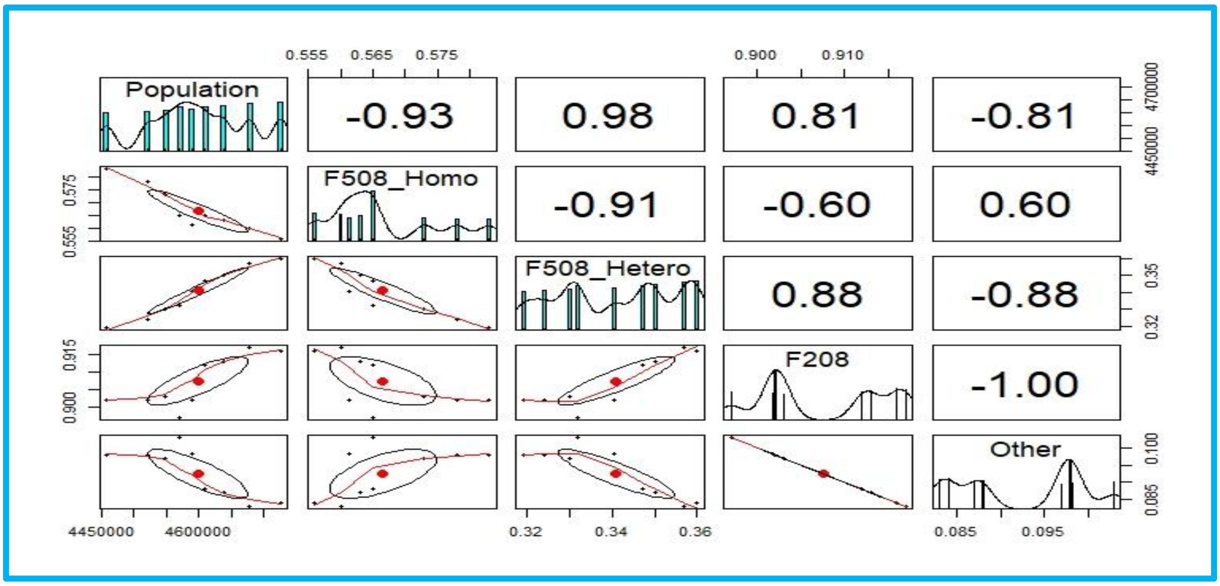

Figure 34: Correlation Matrix Population vs Genotypes from 2008 to 2016.

[12] Excluding non-Irish from population for 2008 to 2016.

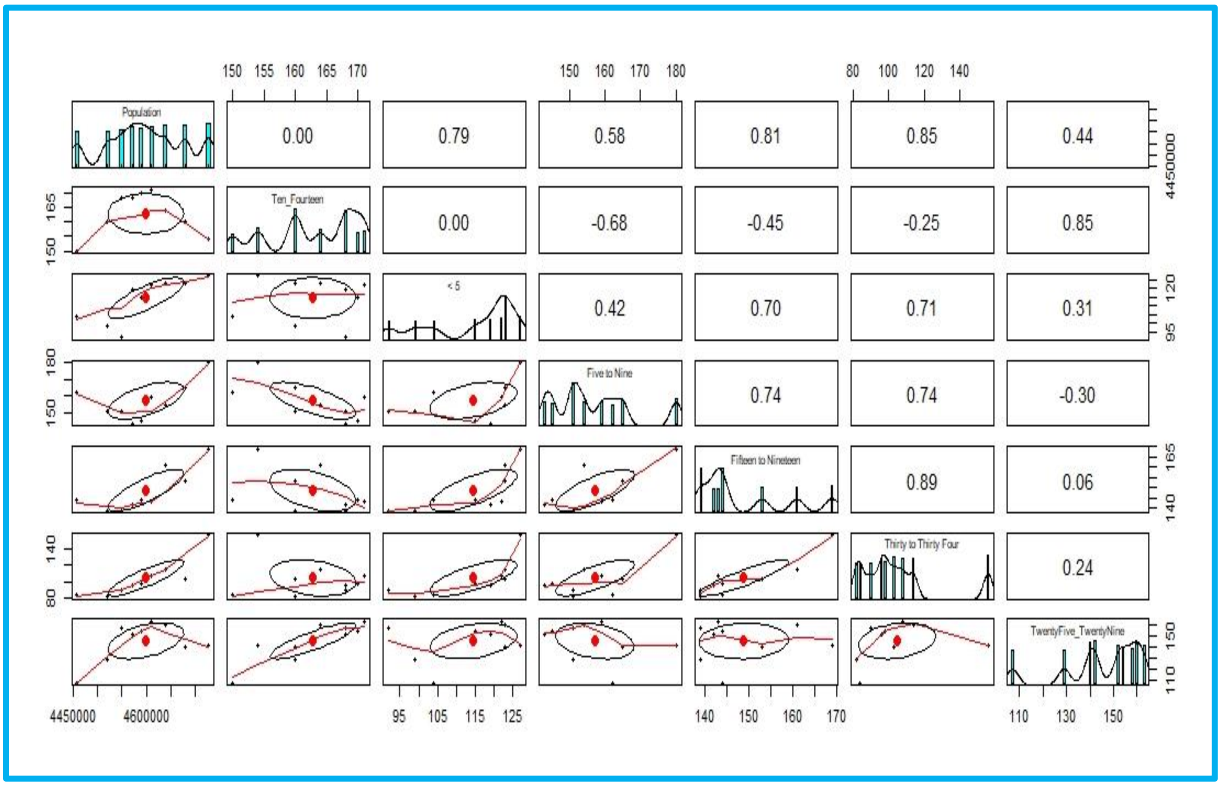

Figure 35: Correlation Matrix vs Age Groups from 2008 to 2018.

```
# Test of Independence
chisq.test(dataset$Incidence.of.CF.Yearly, dataset$Ten_Fourteen)
chisq.test(dataset$Incidence, dataset$Ten_Fourteen)
chisq.test(dataset$Incidence.of.CF.Yearly, dataset$`Five to Nine`)
chisq.test(dataset$Incidence.of.CF.Yearly, dataset$Twenty_TwentyFour)
chisq.test(dataset$Incidence.of.CF.Yearly, dataset$TwentyFive_TwentyNine)
chisq.test(dataset$Incidence.of.CF.Yearly, dataset$Year)
chisq.test(dataset$Incidence, dataset$Population)
chisq.test(dataset$Population, dataset$`Five to Nine`)
chisq.test(dataset$Population, dataset$`Fifteen to Nineteen`)
chisq.test(dataset$Population, dataset$Twenty_TwentyFour)
chisq.test(dataset$Population, dataset$TwentyFive_TwentyNine)
chisq.test(dataset$Population, dataset$`Thirty to Thirty Four`)
chisq.test(dataset$Population, dataset$`Thirty Five to Thity Nine`)
chisq.test(dataset$Population, dataset$`Forty to Forty Four`)
chisq.test(dataset$Population, dataset$`Forty Five to Forty Nine`)
chisq.test(dataset$Population, dataset$`Greater than Fifty`)
```

Figure 36 Test of Independence Between Variables

Figure 36 illustrates the code used to test for independence[13] between a combination of variables to explore if there is a significant correlation between the variables of the same dataset. All the variables tested had an exceptionally low p-value – example figure 37 below.

[13] https://data-flair.training/blogs/chi-square-test-in-r/#:~:text=Introduction%20to%20Chi%2DSquare%20Test,selected%20from%20the%20same%20population.

```
> chisq.test(dataset$Incidence, dataset$Ten_Fourteen)

        Pearson's Chi-squared test

data:  dataset$Incidence and dataset$Ten_Fourteen
X-squared = 60768, df = 48, p-value < 2.2e-16
```

Figure 37: Chi-Square Test of Independence for Incidence vs Ten to Fourteen Age Group.

3.1.7 Kernel SVM

```
# Kernal SVM CF
dataset = read.csv('CF FinalV7.csv')
names(dataset)
names(dataset)[1] = "Year"
names(dataset)
names(dataset)[4] = "< 5"
names(dataset)[53] = "F508_Homo"
names(dataset)[54] = "F508_Hetero"
names(dataset)[55] = "F208"

library(dplyr)
dataset = rename(dataset, "Twenty_TwentyFour" = X20.24)
dataset = rename(dataset, "Five to Nine" = X5.9)
dataset = rename(dataset, "Ten_Fourteen" = X10.14)
dataset = rename(dataset, "Fifteen to Nineteen" = X15.19)
dataset = rename(dataset, "TwentyFive_TwentyNine" = X25.29)
dataset = rename(dataset, "Thirty to Thirty Four" = X30.34)
dataset = rename(dataset, "Thirty Five to Thity Nine" = X35.39)
dataset = rename(dataset, "Forty to Forty Four" = X40.44)
dataset = rename(dataset, "Forty Five to Forty Nine" = X45.49)
dataset = rename(dataset, "Greater than Fifty" = X.50)
dataset$Population = as.numeric(dataset$Population)
dataset$F508_Homo = as.numeric(dataset$F508_Homo)
dataset$F508_Hetero = as.numeric(dataset$F508_Hetero)
dataset$F208 = as.numeric(dataset$F208)
dataset$Other = as.numeric(dataset$Other)
dataset$Incidence = as.numeric(dataset$Incidence)
names(dataset)
```

Figure 38: Preprocessing Data for Kernel SVM Model.

Figure 38 illustrates the preprocessing steps applied to the Kernel SVM model – renaming variables, changing datatypes, checking structure of dataset and names of variables

```r
# Choosing Combinations of Variables from dataset
dataset = dataset[, c("Incidence.of.CF.Yearly","X..Other.Ethnicity")]
dataset = dataset[, c("Incidence.of.CF.Yearly","Population")]
dataset = dataset[, c("Incidence","Population")]
dataset = dataset[, c("Incidence","Ten_Fourteen")]
dataset = dataset[, c("Incidence","< 5", "Year.of.Birth")]
dataset = dataset[, c("Incidence","Five to Nine", "Year.of.Birth")]
dataset = dataset[, c("Incidence","Thirty to Thirty Four")]
dataset = dataset[, c("Incidence","Thirty Five to Thity Nine")]
dataset = dataset[, c("Incidence","Forty to Forty Four")]
dataset = dataset[, c("Incidence","Forty Five to Forty Nine")]
dataset = dataset[, c("Incidence","Greater than Fifty")]
dataset = dataset[, c("Incidence","F508_Homo")]
dataset = dataset[, c("Incidence","F508_Hetero")]
dataset = dataset[, c("Incidence","Other")]
dataset = dataset[, c("Incidence","Other", "Year.of.Birth")]
dataset = dataset[, c("Prevalence","Population")]
str(dataset)
table(dataset$Incidence)
```

Figure 39: Choosing Combinations of Variables for kernel SVM

Figure 39 illustrates the combination of variables chosen for Kernel SVM model.

```r
# Encoding the target feature as factor
dataset$Incidence = factor(dataset$Incidence,
                           levels = c("0.199", "0.426", "0.571", "0.684",
                                      "0.906", "0.923", "0.963", "1.041",
                                      "1.046"),
                           labels = c("1", "2", "3", "4", "5", "6", "7", "8",
                                      "9"))
dataset$Incidence = as.numeric(dataset$Incidence)

# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Incidence, SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Feature Scaling
training_set[-1] = scale(training_set[-1])
test_set[-1] = scale(test_set[-1])
```

Figure 40:

Figure 40 illustrates code used for changing dependent variable incidence into a factor with labels 1 to 9. Splitting the dataset in training and test sets with a ratio of 75:25; feature scaling the training and test sets.

```
# Fitting Kernel SVM to the Training set
# Create your classifier here
library(e1071)
classifier = svm(formula = Incidence ~.,
                 data = training_set,
                 type = 'C-classification',
                 kernel = 'radial')

# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-1])

# Making the Confusion Matrix
cm = table(test_set[, 1], y_pred)
cm
sum(diag(cm))/sum(cm)
1-sum(diag(cm))/sum(cm)
```

Figure 41: Code for Implementing kernel SVM classifier with a “radial kernel”.

Figure 41 illustrates using the “e1071”[14] package from the RStudio library using a radial kernel. Prediction function and confusion matrix implemented.

4 SPSS

4.1 Introduction

SPSS[15] was used to analyse the Cystic Fibrosis dataset via multiple linear regression, PCA[16] analysis to show patterns, multicollinearity, correlations, etc.

4.1.1 Multiple Regression

Descriptive Statistics

	Mean	Std. Deviation	N
Incidence of CF/Yearly	34.67	13.145	10128
Year	2012.19	2.579	10128
Year of Birth	1992.69	12.267	10128
Age	19.50	12.188	10128
<5	114.56	11.531	10128
5-9	157.11	11.075	10128
10-14	162.79	6.846	10128
15-19	148.83	10.196	10128
20-24	148.63	13.141	10128
25-29	145.55	16.356	10128
30-34	104.80	21.697	10128
35-39	66.14	17.380	10128
40-44	36.44	11.230	10128
45-49	15.09	7.541	10128
>50	12.34	8.203	10128
Male	649.58	51.686	10128
Female	480.41	34.980	10128
%Male	.57445566745659	.00305284251258	10128
%Female	.42929097551342	.01057468604299	10128
<18yrs	530.08	24.827	10128
>= 18yrs	599.90	63.185	10128
% Adults	.49402390438247	.00000000000000	10128
% Paed	.50597609561753	.00000000000000	10128
No of Males >=18	351.01	44.542	10128
% No of Males>=18	.30963258901074	.01772102138959	10128
No of Females>= 18yrs	245.26	23.565	10128
% No of Females>=18	.21675365186787	.00584532398890	10128
Alive/F	480.40	34.966	10128
% No of Females/Alive	.42600	.000000	10128
Alive/M	649.57	51.666	10128
% No of Males	.57400	.000000	10128
Median Age at Death	26.587	3.3227	10128
Median Age of Death/Males	29.717	3.9730	10128
Median Age of Death/Females	27.969	4.2754	10128
Irish Ethnicity	.97800	.000000	10128
Number Enrolled	1271.64	138.301	10128
Total No of Deceased	Double-click to activate	60.192	10128
Number of PWCF who are alive at the end of Year		86.424	10128
Number Deceased/End of year	17.73	5.098	10128
No of Irish Ethnicity	1077.76	63.036	10128
% Other Ethnicity	.02200000000000	.00000000000000	10128

Figure 42: Descriptive Statistics of Cystic Fibrosis Dataset

[14] https://cran.r-project.org/web/packages/e1071/index.html

[15] https://www.ibm.com/analytics/spss-statistics-software

[16] Principal Component Analysis

Correlations

		Incidence of CF/Yearly	Year	Year of Birth	Age	<5	5-9	10-14	15-19	20-24	25-29	30-34	35-39
Pearson Correlation	Incidence of CF/Yearly	1.000	.324	.043	.024	.348	-.383	.823	-.051	-.788	.872	.108	.176
	Year	.324	1.000	.141	.069	.846	.579	.004	.816	-.801	.440	.834	.968
	Year of Birth	.043	.141	1.000	-.978	.121	.081	-.002	.110	-.112	.054	.108	.139
	Age	.024	.069	-.978	1.000	.057	.041	.002	.061	-.056	.038	.068	.064
	<5	.348	.846	.121	.057	1.000	.417	.003	.702	-.711	.312	.708	.802
	5-9	-.383	.579	.081	.041	.417	1.000	-.684	.735	-.068	-.300	.742	.578
	10-14	.823	.004	-.002	.002	.003	-.684	1.000	-.450	-.513	.845	-.249	-.116
	15-19	-.051	.816	.110	.061	.702	.735	-.450	1.000	-.474	.061	.891	.838
	20-24	-.788	-.801	-.112	-.056	-.711	-.068	-.513	-.474	1.000	-.827	-.528	-.730
	25-29	.872	.440	.054	.038	.312	-.300	.845	.061	-.827	1.000	.243	.318
	30-34	.108	.834	.108	.068	.708	.742	-.249	.891	-.528	.243	1.000	.757
	35-39	.176	.968	.139	.064	.802	.578	-.116	.838	-.730	.318	.757	1.000
	40-44	.341	.943	.134	.064	.806	.525	.017	.806	-.820	.470	.737	.938
	45-49	.259	.956	.134	.067	.749	.681	-.119	.829	-.712	.336	.882	.915
	>50	.035	.938	.132	.065	.740	.809	-.272	.848	-.572	.178	.875	.924
	Male	.249	.987	.139	.068	.845	.645	-.068	.842	-.725	.366	.868	.943
	Female	.314	.993	.142	.067	.845	.582	-.020	.811	-.773	.395	.821	.958
	%Male	-.165	.592	.075	.049	.533	.748	-.371	.756	-.214	.028	.788	.533
	%Female	.236	.086	.010	.008	.107	-.325	.148	.243	-.343	.301	-.071	.192
	<18yrs	.244	.922	.135	.059	.878	.592	-.125	.790	-.650	.224	.771	.881
	>= 18yrs	.282	.995	.139	.070	.815	.617	-.018	.827	-.765	.430	.861	.956
	% Adults	.	.	.	.	.	.	.	.	.	.	.	.
	Paed	.	.	.	.	.	.	.	.	.	.	.	.
	of Males >=18	.094	.956	.134	.067	.736	.684	-.217	.901	-.655	.264	.860	.973
	No of Males>=18	-.119	.777	.106	.057	.486	.632	-.354	.843	-.472	.122	.731	.864
	of Females>= 18yrs	.242	.976	.133	.072	.793	.650	-.083	.866	-.724	.389	.923	.932
	% No of Females>=18	.145	.660	.077	.062	.414	.483	-.071	.675	-.510	.380	.826	.618
	Alive/F	.314	.993	.141	.067	.845	.582	-.020	.811	-.773	.396	.820	.958
	% No of Females/Alive	.	.	.	.	.	.	.	.	.	.	.	.
	Alive/M	.249	.987	.138	.069	.846	.644	-.068	.842	-.725	.367	.867	.943
	% No of Males	.	.	.	.	.	.	.	.	.	.	.	.
	Median Age at Death	-.132	.684	.095	.049	.273	.664	-.360	.716	-.347	.056	.667	.736
	Median Age of Death/Males	-.009	.152	.013	.020	-.029	.134	.204	-.095	-.132	.274	.186	.124
	Median Age of Death/Females	.273	.527	.056	.054	.295	.133	-.015	.556	-.547	.322	.478	.575
	Irish Ethnicity	.	.	.	.	.	.	.	.	.	.	.	.
	Number Enrolled	.284	.990	.143	.065	.842	.589	-.028	.784	-.764	.372	.779	.970
	Total No of Deceased	.443	.977	.139	.067	.910	.437	.123	.747	-.855	.504	.775	.931
	Number of PWCF who are alive at the end of Year	.276	.992	.140	.068	.848	.621	-.049	.832	-.746	.379	.851	.952
	Number Deceased/End of year	.058	-.240	-.033	-.017	-.040	-.591	.097	-.136	-.012	-.026	-.344	-.107
	No of Irish Ethnicity	.294	.999	.141	.069	.862	.600	-.032	.829	-.780	.404	.841	.969
	% Other Ethnicity	.	.	.	.	.	.	.	.	.	.	.	.
	No of Other/Ethnicity	.214	.921	.131	.063	.765	.642	-.090	.793	-.620	.296	.831	.855
	Children Low	.060	.176	.038	-.001	.161	-.225	.055	.110	-.300	.080	-.267	.363
	Children Moderate	-.095	-.872	-.130	-.054	-.745	-.749	.171	-.648	.548	-.154	-.692	-.843
	Children High	-.081	.535	.083	.029	.516	.700	-.348	.379	-.238	-.165	.465	.528
	Adults>=18 Low	-.219	-.870	-.126	-.057	-.846	-.578	.063	-.738	.676	-.315	-.674	-.856
	Adults>=18 Moderate	-.353	-.597	-.097	-.028	-.521	-.390	-.057	-.264	.467	-.144	-.351	-.521
	Adults>=18 High	.145	.843	.126	.051	.715	.680	-.142	.690	-.591	.217	.624	.832
	All Low	-.241	-.882	-.129	-.056	-.743	-.566	-.067	-.606	.651	-.379	-.648	-.831
	All Moderate	-.113	-.839	-.124	-.052	-.704	-.724	.106	-.588	.552	-.211	-.651	-.803
	All High	.172	.723	.111	.040	.621	.678	-.096	.448	-.453	.117	.542	.648
	ΔF508_Homo	-.531	-.929	-.130	-.065	-.831	-.322	-.241	-.663	.863	-.588	-.755	-.857
	ΔF508_Hetero	.326	.996	.142	.067	.826	.588	.017	.790	-.795	.442	.808	.960
	ΔF208	.012	.849	.124	.055	.633	.767	-.249	.761	-.535	.171	.688	.864
	Other	-.012	-.849	-.124	-.055	-.633	-.767	.249	-.761	.535	-.171	-.688	-.864
	Prevalence	.187	.986	.140	.068	.845	.687	-.157	.876	-.710	.300	.864	.973

Double-click to activate

Double-click to activate

Figure 43: Correlations of The Cystic Fibrosis Dataset

Figure 43 illustrates the correlations between the dependent variable, incidence, and the independent variables – anything above 0.3 is preferable. It also demonstrates the correlations between the independent variables. It would be advisable to remove any bivariate correlations above 0.7 in the analysis (Pallant, 2016, p.159).

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.875[a]	.765	.765	.133415

a. Predictors: (Constant), Age, 10-14, <5, 30-34, 35-39, 5-9

b. Dependent Variable: Incidence

Table 1: Model Summary

In the model summary (table 1) explains how much of the variance in the dependent variable, incidence is explained by the model. In this case, the value is 0.765, which is 76.5%. This means that all the independent variables in this model explain 76.5% of the variance in incidence.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B		Correlations			Collinearity Statistics	
		B	Std. Error	Beta	t	Sig.	Lower Bound	Upper Bound	Zero-order	Partial	Part	Tolerance	VIF
1	(Constant)	-5.871	.087		-67.836	.000	-6.041	-5.702					
	<5	.008	.000	.328	37.469	.000	.007	.008	.309	.349	.180	.303	3.302
	5-9	.001	.000	.035	3.018	.003	.000	.001	-.399	.030	.015	.170	5.877
	10-14	.034	.000	.852	106.727	.000	.034	.035	.810	.728	.514	.364	2.746
	30-34	.002	.000	.161	15.710	.000	.002	.002	.066	.154	.076	.220	4.545
	35-39	-.003	.000	-.187	-19.953	.000	-.003	-.003	.119	-.195	-.096	.264	3.786
	Age	.000	.000	-.002	-.469	.639	.000	.000	.019	-.005	-.002	.995	1.005

a. Dependent Variable: Incidence

Table: 2: Coefficients of Chosen variables

Table 2 illustrates the any problems that can occur with multicollinearity. Two columns to inspect are tolerance and VIF (Variance Inflation Factor)[17]. The "rule of thumb" for VIF is anything above VIF values of 10 would be a concern here. The highest VIF value is 5.877 which is well below the cut-off of 10.

Evaluating each of the independent variables

By looking in table 2 in the standardised coefficients the Beta value of 0.852 has the largest beta coefficient, which is for the 10 to 14 age group. This means this variable makes the strongest contribution to explaining the dependent variable, incidence.

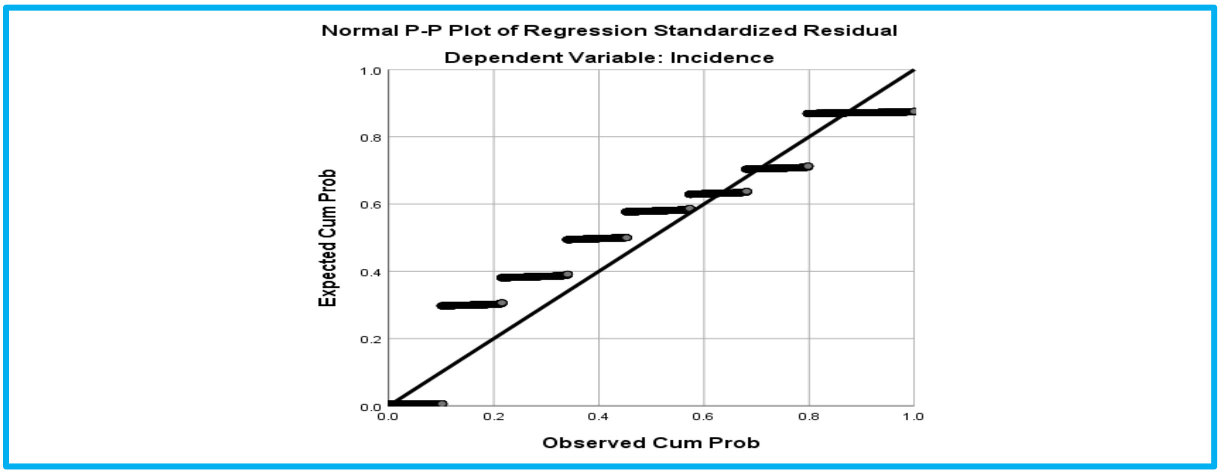

Figure 43: Probability Plot (P-P) of the Regression Standardised Residual

Figure 43 shows that the model has no major deviations from normality.

[17] https://en.wikipedia.org/wiki/Variance_inflation_factor

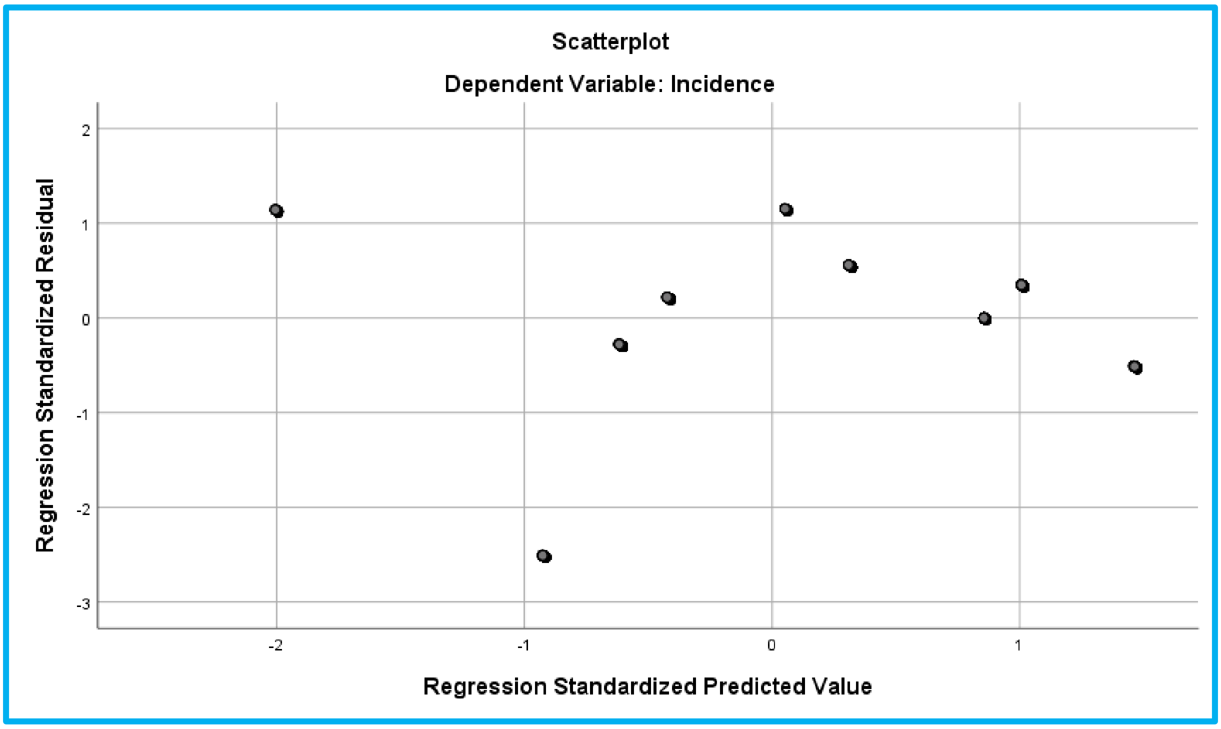

Figure 44: Scatterplot of the Standarsised Residuals

Figure 44 illustrates that the residual points are roughly distributed above and below the zero point – which suggest no violations of the assumptions.

4.1.2 PCA – Principal Component Analysis

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.572
Bartlett's Test of Sphericity	Approx. Chi-Square	140276.815
	df	36
	Sig.	.000

Table 3: KMO & Bartlett's Test

Table 3 sets out whether the data set is suitable for factor analysis[18]. The first statistical test is the Kaiser-Meyer-Olkin Measure of Sampling Adequacy. The minimum acceptable is 0.5 or above. In this model it is 0.572, and Bartlett's Test of Sphericity is significant (p = 0.000).

[18] https://en.wikipedia.org/wiki/Factor_analysis

Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	3.720	41.335	41.335	3.720	41.335	41.335	3.592	39.913	39.913
2	3.451	38.344	79.679	3.451	38.344	79.679	3.579	39.766	79.679
3	.976	10.848	90.527						
4	.370	4.114	94.642						
5	.198	2.196	96.837						
6	.157	1.744	98.582						
7	.110	1.224	99.806						
8	.013	.142	99.947						
9	.005	.053	100.000						

Extraction Method: Principal Component Analysis.

Table 4: Total Variance Explained

Table 4 explains how many components to extract based on eigenvalues greater than 1. In this model only two components recorded eigenvalues greater than1 (3.720, 3.451) these two components explained 79.68% of the variance. Furthermore, component 3 came remarkably close to the criteria with an eigenvalue of 0.976. If this result is included 90.53% of the variance is explained.

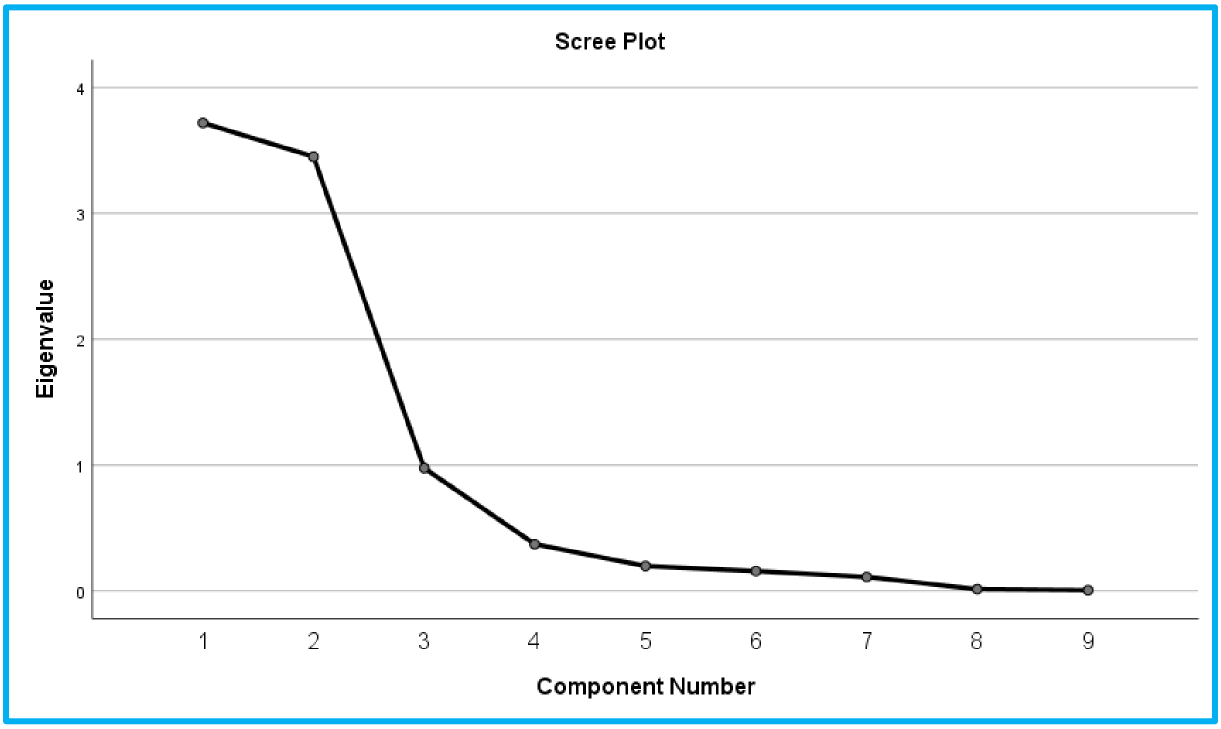

Figure 45: Scree Plot

Figure 45 shows a scree plot[19] of the model components. It shows a distinct change at the "elbow" of the plot at 3. Or at component 4, depending on the context 4 components could be included.

Component Matrix[a]

	Component					
	1	2	3	4	5	6
20-24	**-.963**					
<5	**.794**	**.362**		**-.459**		
25-29	**.780**	**-.555**				
30-34	**.700**	**.640**				
Incidence	**.688**	**-.638**				
5-9		**.900**				
10-14	**.414**	**-.888**				
15-19	**.592**	**.753**				
Year of Birth			**.983**			

Extraction Method: Principal Component Analysis.

a. 6 components extracted.

Figure 46: Component matrix

Figure 46 shows a component matrix of the unrotated loadings of each of the variables on 6 components. Very few variables load on 3 to 6. This suggests a two-factor solution maybe more suitable.

Results

The 59 variables were subjected to PCA. Initially 41 variables were excluded due to high collinearity based on the multiple regression modelling. This left a total of 18 variables to analyse, this was further reduced to 9 variables. The scree plot showed that 2 components had a cumulative variance of 79.68%.

[19] https://en.wikipedia.org/wiki/Scree_plot

5 Excel

5.1 Introduction

Datasets were extracted manually from the CF Registry of Ireland annual reports from 2008 to 2016. Two final Datasets were created: CF Final V7, CF Final V10: 59 variables with 10,128 rows; 21 variables with 10,128 rows – 597,552, 212,688 datapoints respectively.

5.2 Datasets

Year	Year of Birth	Age	<5	5-9	10-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49	>50	Male	Female	%Male	%Female	<18yrs	>= 18yrs	% Adults	% Paed	No of Males >=18	% No of Males>=18	No of Females
2008	2008	0	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2005	3	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2006	2	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2007	1	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2005	3	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2004	4	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2007	1	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2004	4	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2005	3	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2008	0	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2006	2	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2008	0	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2008	0	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2007	1	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2005	3	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2007	1	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2007	1	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2005	3	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2007	1	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	
2008	2004	4	104	162	150	144	175	107	84	40	21	5	3	576	428	57.40%	42.60%	508	496	49.40%	50.60%	288	28.70%	

Figure 47: Dataset Year to % No of Males

Incidence of CF/Yearly	Severity of Condition	Alive/F	% No of Females/Alive	Alive/M	% No of Males	Median Age at Death	Median Age of Death/Males	Median Age of Death/Females	Irish Ethnicity
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%
16	Low	428	42.60%	576	57.40%	23	23	21	97.80%

Figure 48: Dataset Incidence of CF/Yearly to Irish Etnicity

Number Enrolled	Total No of Deceased	Number of PWCF who are alive at the end of Year	Number Deceased/End of year	No of Irish Ethnicity
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981
1065	135	1004	17	981

Figure 49: Dataset: Number Enrolled to No. of Irish Ethnicity

AR	AS	AT	AU	AV	AW	AX	AY	AZ
Children Low	Children Moderate	Children High	Adults>=18 Low	Adults>=18 Moderate	Adults>=18 High	All Low	All Moderate	All High
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45
0.04	0.34	0.73	0.3	0.44	0.35	0.24	0.48	0.45

Figure 50: Dataset: Children Low to All High (Severity of Condition)

6 Accuracy of Models

6.1 Introduction

Excel was used to calculate the overall accuracy of the models used.

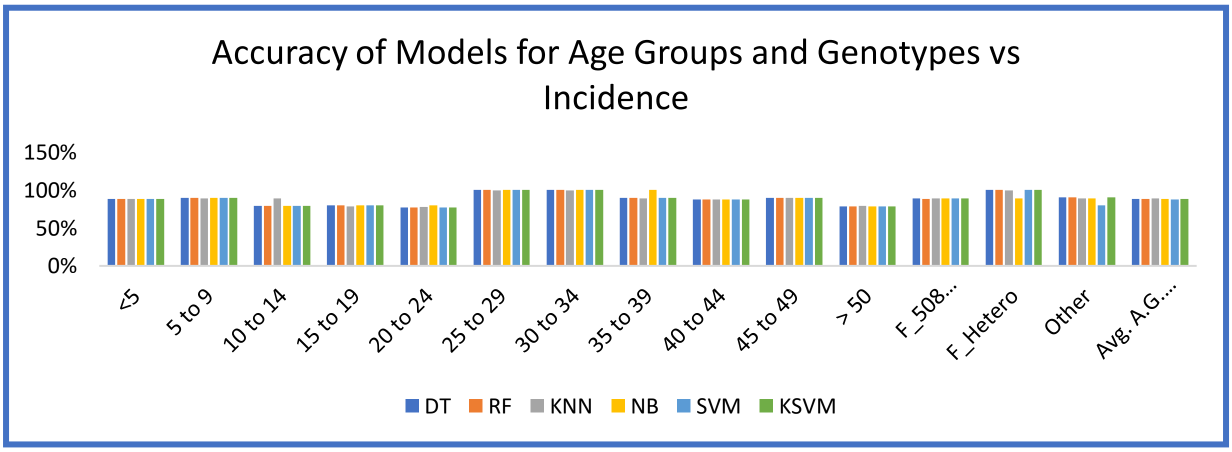

Figure 51: Accuracy of Models on Selected Variables.

Figure 51 shows the accuracy arrived at using machine learning models Decision Tree to Kernel SVM.

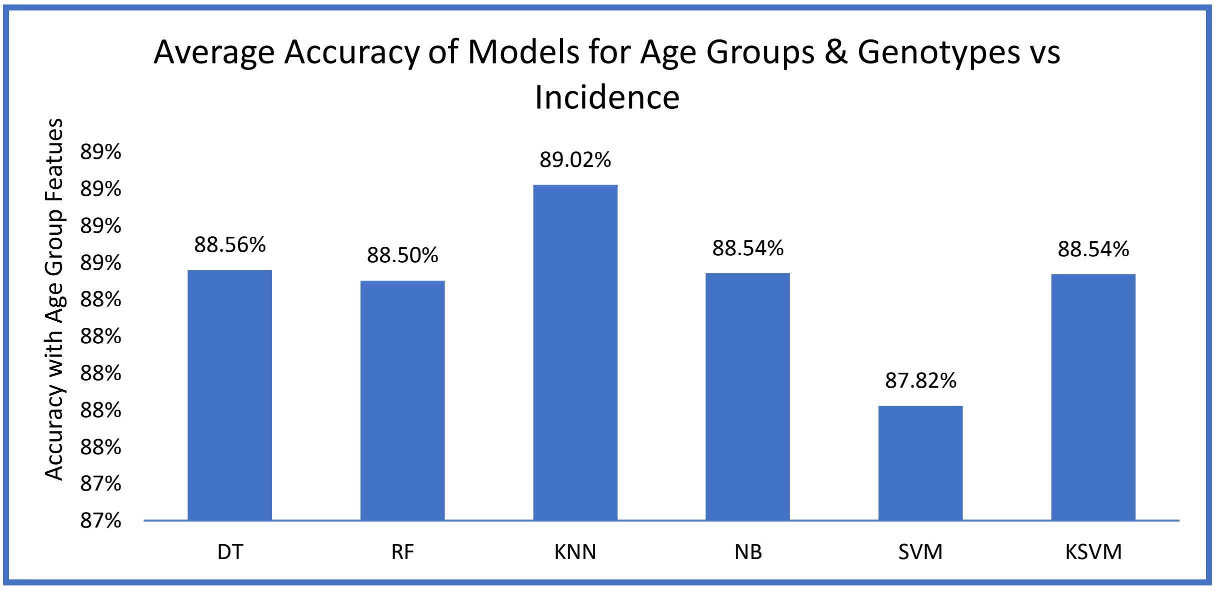

Figure 52: Average Accuracy of Models

Figure 52 shows the average accuracy of machine learning models on the selected variables. KNN (K Nearest Neighbours) is the best performing algorithm in this research project.

7 References

Altman, N., 1991. *An Introduction To Kernel And Nearest Neighbor Nonparametric Regression*. [online] Ecommons.cornell.edu. Available at: <https://ecommons.cornell.edu/bitstream/handle/1813/31637/BU-1065-MA.pdf;jsessionid=FEC7E6B53138480A841AA60EC059D01B?sequence=1> [Accessed 13 August 2020].

Breiman, L., 1984. *Classification And Regression Trees*.

Hofmann, T., Schölkopf, B. and Smola, A., 2008. Kernel methods in machine learning. *The Annals of Statistics*, 36(3), pp.1171-1220.

Kuhn, M., 2019. *The Caret Package*. [online] Topepo.github.io. Available at: <https://topepo.github.io/caret/> [Accessed 11 August 2020].

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C. and Lin, C., 2019. *Package 'E1071'*. [online] Cran.r-project.org. Available at: <https://cran.r-project.org/web/packages/e1071/e1071.pdf> [Accessed 11 August 2020].

models, H. and Welling, S., 2020. *How To Interpret Mean Decrease In Accuracy And Mean Decrease GINI In Random Forest Models*. [online] Cross Validated. Available at: <https://stats.stackexchange.com/questions/197827/how-to-interpret-mean-decrease-in-accuracy-and-mean-decrease-gini-in-random-fore> [Accessed 12 August 2020].

Pallant, J., 2016. *SPSS Survival Manual*. 6th ed. Maidenhead: Open University Press.

Tuszynski, J., 2020. *Package 'Catools'*. [online] Cran.r-project.org. Available at: <https://cran.r-project.org/web/packages/caTools/caTools.pdf> [Accessed 11 August 2020].