Configuration Manual

MSc Research Project
Data Analytics

Siddharth Sharma
Student ID: x16148371

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Tomer

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Siddharth Sharma
Student ID:	x16148371
Programme:	Data Analytics
Year:	2018
Module:	MSc Research Project
Supervisor:	Mr. Vikas Tomer
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	6

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:	
Date:	20th December 2018

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

Siddharth Sharma
x16148371

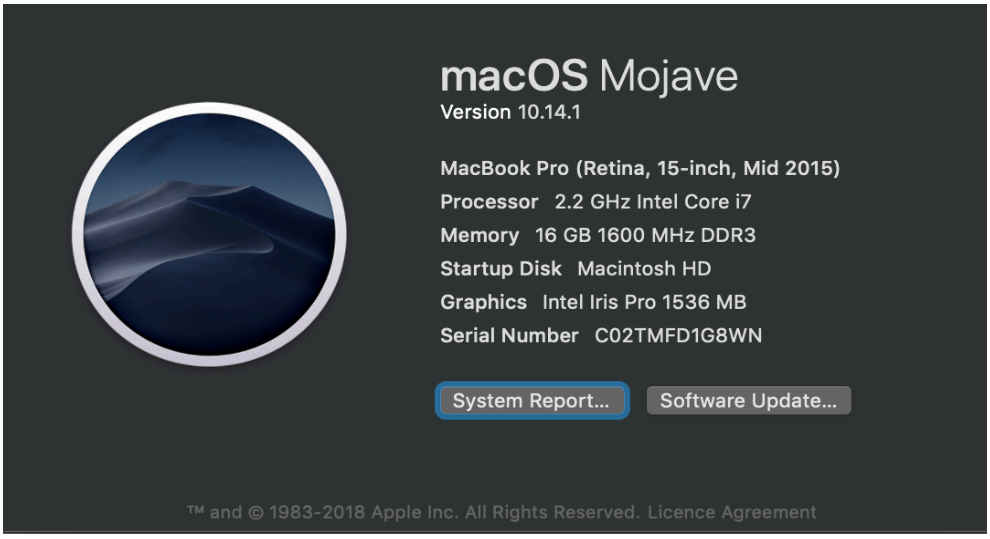
1 Introduction

This configuration manual has been created for a better understanding of the significant information and technical process used for the classification of breast cancer. The research focuses on classifying breast cancer with the help of RNA-Seq gene expression data having five tumors: BRCA (Breast invasive carcinoma), KIRC (Kidney Renal clear cell Carcinoma), COAD (Colon Adenocarcinoma), LUAD (Lung Adenocarcinoma) and PRAD (Prostate Adenocarcinoma). Machine learning techniques that were implemented are support vector machine- radial basis function, k-Nearest neighbours, decision trees (C5.0) and ensemble learning techniques: stacking and bagging. The evaluation was done with the help of classification accuracy, kappa, f-measure, precision and recall. Implementation of the research is described step by step in the following sections.

2 System Configuration

2.1 Hardware

Implementation for the research was carried put on MacOS. Specifications can be seen in figure 1:



macOS Mojave
Version 10.14.1

MacBook Pro (Retina, 15-inch, Mid 2015)
Processor 2.2 GHz Intel Core i7
Memory 16 GB 1600 MHz DDR3
Startup Disk Macintosh HD
Graphics Intel Iris Pro 1536 MB
Serial Number C02TMFD1G8WN

™ and © 1983-2018 Apple Inc. All Rights Reserved. Licence Agreement

Figure 1: Hardware Configuration

2.2 Software

2.2.1 R Console & R Studio

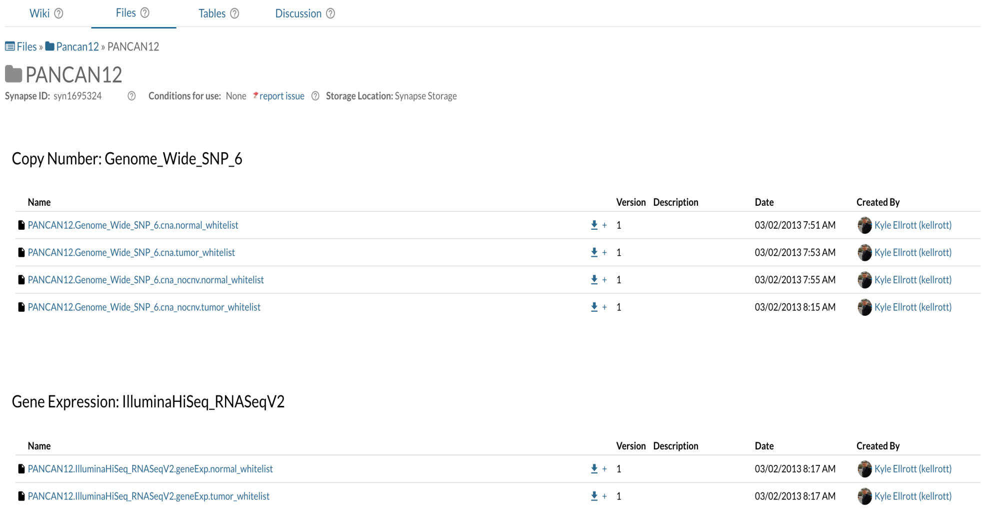
R is a programming language which is used for computation of statistical models and implementation of various data mining algorithms with the help of packages specific to particular functions. R is a widely used language by data analysts and data scientists for gaining knowledge and building statistical & data mining models. R console can be downloaded at R Console Rstudio is an interface were coding, debugging and visualisations can be done. RStudio is aviaible at: RStudio

R Libraries Libraries used for the project are:

- **caret:** Used for classification and regression models.
- **randomForest:** Used for buliding random Forest algorithm
- **C50:** Used to build decision trees with algorithm C5.0
- **class:** Used to build k-Nearest Neighbours
- **catools:** Used for round-off error free sem and cumsum.
- **gbm:** Used to build Gradient Boosting Machine
- **corrpot:** Used for plotting correlation matrix.

3 Data Extraction

Data has been taken from the website: Synapse



Files » Pancan12 » PANCAN12

PANCAN12

Synapse ID: syn1695324 Conditions for use: None report issue Storage Location: Synapse Storage

Copy Number: Genome_Wide_SNP_6

Name	Version	Description	Date	Created By
PANCAN12.Genome_Wide_SNP_6.cna.normal_whitelist	1		03/02/2013 7:51 AM	Kyle Ellrott (kellrott)
PANCAN12.Genome_Wide_SNP_6.cna.tumor_whitelist	1		03/02/2013 7:53 AM	Kyle Ellrott (kellrott)
PANCAN12.Genome_Wide_SNP_6.cna_nocnv.normal_whitelist	1		03/02/2013 7:55 AM	Kyle Ellrott (kellrott)
PANCAN12.Genome_Wide_SNP_6.cna_nocnv.tumor_whitelist	1		03/02/2013 8:15 AM	Kyle Ellrott (kellrott)

Gene Expression: IlluminaHiSeq_RNASeqV2

Name	Version	Description	Date	Created By
PANCAN12.IlluminaHiSeq_RNASeqV2.geneExp.normal_whitelist	1		03/02/2013 8:17 AM	Kyle Ellrott (kellrott)
PANCAN12.IlluminaHiSeq_RNASeqV2.geneExp.tumor_whitelist	1		03/02/2013 8:17 AM	Kyle Ellrott (kellrott)

Figure 2: Data Extraction

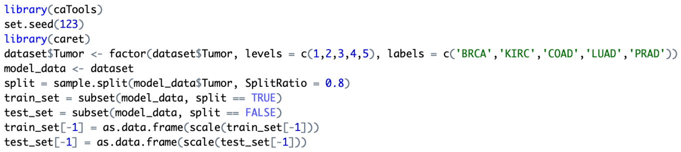
The dataset used for the research has data concerning 800 patients and 14,000 genes expression levels for the 800 patients.

4 Data Preprocessing

Data preprocessing has been done in RStudio using R. Following steps were taken:

4.1 Training and Testing Data

Here random sample is selected from the original dataset into training and testing datasets in the ratio 80:20 with the help of following R code.

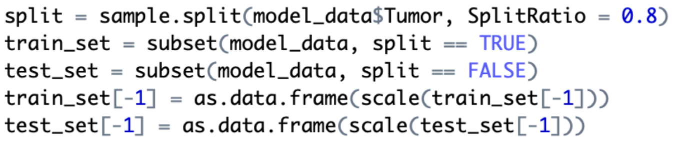


```
library(caTools)
set.seed(123)
library(caret)
dataset$Tumor <- factor(dataset$Tumor, levels = c(1,2,3,4,5), labels = c('BRCA','KIRC','COAD','LUAD','PRAD'))
model_data <- dataset
split = sample.split(model_data$Tumor, SplitRatio = 0.8)
train_set = subset(model_data, split == TRUE)
test_set = subset(model_data, split == FALSE)
train_set[-1] = as.data.frame(scale(train_set[-1]))
test_set[-1] = as.data.frame(scale(test_set[-1]))
```

Figure 3: Training and Testing Data

4.2 Data Scaling

Data scaling has been done so that higher variables cannot dominate variable with low value.

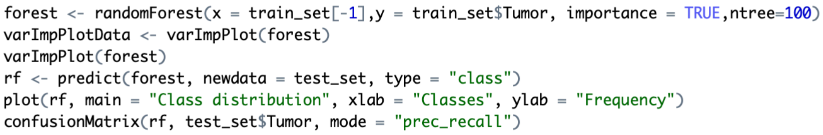


```
split = sample.split(model_data$Tumor, SplitRatio = 0.8)
train_set = subset(model_data, split == TRUE)
test_set = subset(model_data, split == FALSE)
train_set[-1] = as.data.frame(scale(train_set[-1]))
test_set[-1] = as.data.frame(scale(test_set[-1]))
```

Figure 4: Scaling of Data

4.3 Feature Selection

Feature selection helps us in removing the undesired variables that have a negative impact on the performace of model as it makes the algorithm's task of learning quite tough. The most significant features are selected using random forest and high correaltion matrix using R code as shown in the figure:



```
forest <- randomForest(x = train_set[-1],y = train_set$Tumor, importance = TRUE,ntree=100)
varImpPlotData <- varImpPlot(forest)
varImpPlot(forest)
rf <- predict(forest, newdata = test_set, type = "class")
plot(rf, main = "Class distribution", xlab = "Classes", ylab = "Frequency")
confusionMatrix(rf, test_set$Tumor, mode = "prec_recall")
```

Figure 5: Feature Slection by random Forest

5 Section 5

5.1 Class Imbalance

There was no calss imbalance in the dataset but to check the class imbalance of the dataset following R code was used :

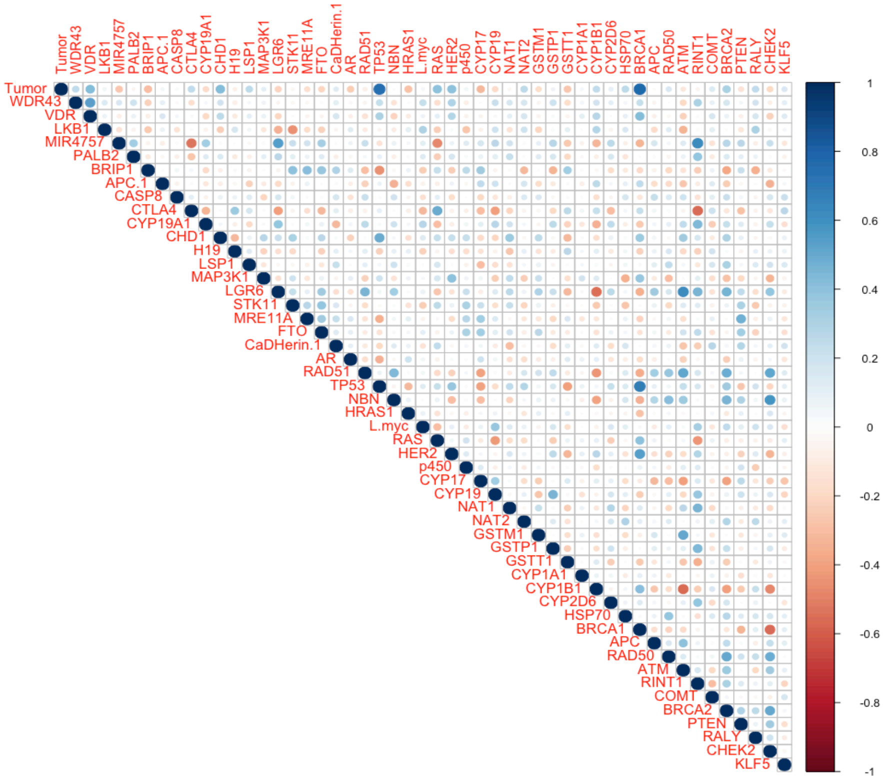
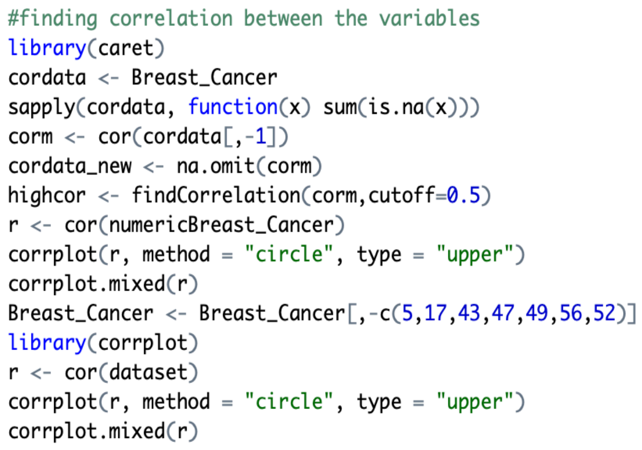


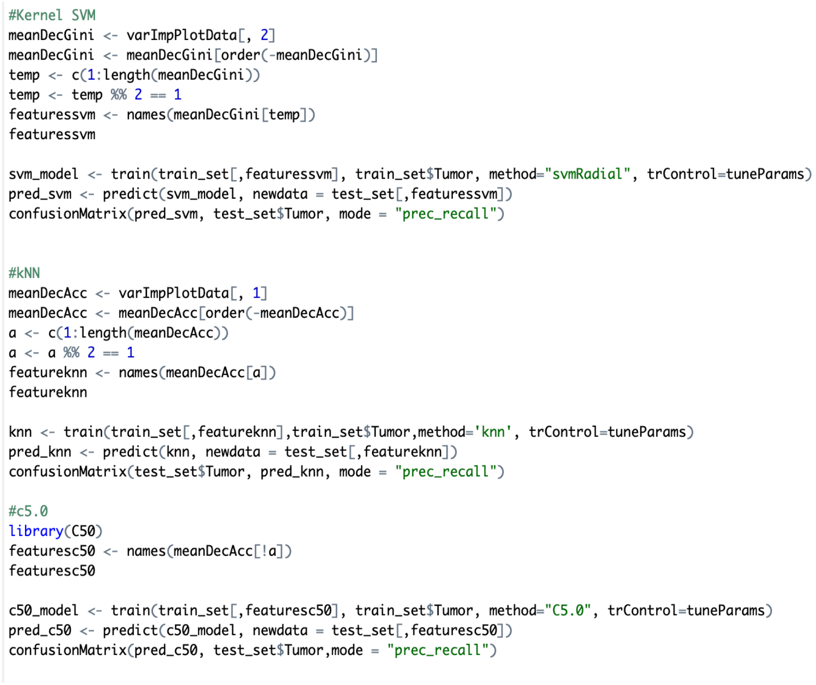
Figure 6: Correlation Matrix

plot(dataset$Tumor)



```
#finding correlation between the variables
library(caret)
cordata <- Breast_Cancer
sapply(cordata, function(x) sum(is.na(x)))
corm <- cor(cordata[,-1])
cordata_new <- na.omit(corm)
highcor <- findCorrelation(corm,cutoff=0.5)
r <- cor(numericBreast_Cancer)
corrplot(r, method = "circle", type = "upper")
corrplot.mixed(r)
Breast_Cancer <- Breast_Cancer[,-c(5,17,43,47,49,56,52)]
library(corrplot)
r <- cor(dataset)
corrplot(r, method = "circle", type = "upper")
corrplot.mixed(r)
```

Figure 7: Feature Selection by High Correlation Matrix



```
#Kernel SVM
meanDecGini <- varImpPlotData[, 2]
meanDecGini <- meanDecGini[order(-meanDecGini)]
temp <- c(1:length(meanDecGini))
temp <- temp %% 2 == 1
featuressvm <- names(meanDecGini[temp])
featuressvm

svm_model <- train(train_set[,featuressvm], train_set$Tumor, method="svmRadial", trControl=tuneParams)
pred_svm <- predict(svm_model, newdata = test_set[,featuressvm])
confusionMatrix(pred_svm, test_set$Tumor, mode = "prec_recall")


#kNN
meanDecAcc <- varImpPlotData[, 1]
meanDecAcc <- meanDecAcc[order(-meanDecAcc)]
a <- c(1:length(meanDecAcc))
a <- a %% 2 == 1
featureknn <- names(meanDecAcc[a])
featureknn

knn <- train(train_set[,featureknn],train_set$Tumor,method='knn', trControl=tuneParams)
pred_knn <- predict(knn, newdata = test_set[,featureknn])
confusionMatrix(test_set$Tumor, pred_knn, mode = "prec_recall")

#c5.0
library(C50)
featuresc50 <- names(meanDecAcc[!a])
featuresc50

c50_model <- train(train_set[,featuresc50], train_set$Tumor, method="C5.0", trControl=tuneParams)
pred_c50 <- predict(c50_model, newdata = test_set[,featuresc50])
confusionMatrix(pred_c50, test_set$Tumor,mode = "prec_recall")
```

Figure 8: Feature Slection for Bagging

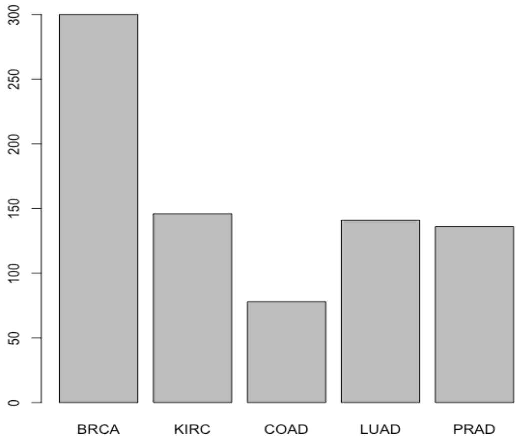


Figure 9: Class Imbalance