A data mining Approach to predict the breast cancer

Vignesh Muthumani (10385771)

Data Mining – CA2

Predicting the type of tumour for Breast Cancer in an individual with the help of CRISP-DM methodology using a cancer patient's record?

CONTENTS

1. INTRODUCTION

Breast cancer is one of the most common cancers in women. Almost 12% of all cancers identified are breast cancers and 25% of the cancers in women are breast related cancers.

The use of data mining algorithms in medical field is rapidly increasing because of the high prediction accuracy. Doctors can treat their patients better by identifying the correct tumor or type of cancer using machine learning algorithms. The correct identification results in treating the patient

efficiently and reduce the cost on medicine, therefore increasing the quality and healthcare value.

In this paper, we follow a Data Mining approach (CRISP-DM) in the quest of identifying the type of tumor in a patient. We will be predicting the type of tumor using some commonly used machine learning algorithms in R Studio and compare them to find out the best performing one.

2.1. BUSINESS & DATA UNDERSTANDING

The main objective of this project is to predict the correct type of tumor on patients. Which in turn could help lower the time and cost for both the patients and doctors. Through such correct diagnosis and early treatments, lives can be saved, and the risk of breast cancer can be brought down.

We gathered the information from a site which contained information about tumors, our goal was to address the issue of Breast Cancer identification which usually gets recognized only in the later stages, where it ends up hard to treat it. Our key stakeholders for this business issue would be Doctors and patients. On assessment of the dataset, we were convinced that this issue can be addressed to extent with the help of data mining by sustaining authentic information to a specific machine learning algorithm that can enable us to identify the sort of tumor a patient has. The pain point lies in the format of the data gathered from the patients, because it has to match the type of format which was used for modelling. This approach can help find malignant tumors in earlier stages which could help the medical professionals and patients.

For this research, we will be using a dataset of patient records from Wisconsin, which contains the diagnosis and a set of 10 main features describing the characteristics of the cell nuclei present in the digitized image of

a fine needle aspirate (FNA) of a breast mass. Ten real-valued features computed for each cell nucleus are as follows,

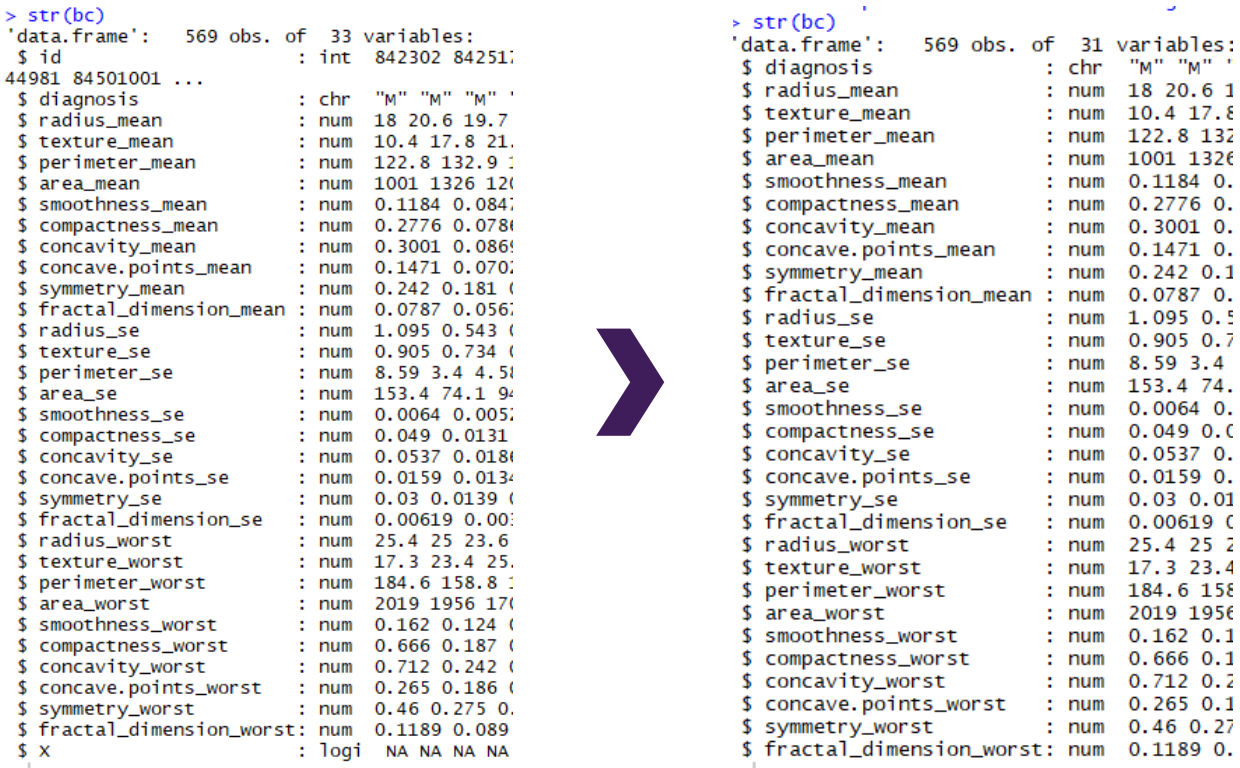
a) Radius
b) Texture (SD of gray-scale values)
c) Perimeter
d) Area
e) Smoothness (local variation in radius lengths)
f) Compactness (perimeter^2 / area - 1.0)
g) Concavity (severity of concave portions of the contour)
h) Concave points (number of concave portions of the contour)
i) Symmetry
j) Fractal dimension ("coastline approximation" - 1)

For each of these 10 features, the 'mean', standard error ('SE') and 'worst' (mean of the three largest values) were computed, resulting in 30 features.

2.2. DATA PREPARATION

The data preparation stage is the second step of CRISP-DM methodology which involves in the process of cleansing the raw dataset to make it into a refined dataset for better precision during the modelling phase. This is usually achieved by identifying the missing values and treat them, identifying outliers and fixing them, feature selection, dimension reduction etc.

- **Removing unwanted variables:** Here, we remove the first column 'id' and last column 'x' as they don't provide any meaning information for our data analysis.



- **Checking for missing values:** Identifying the missing values in a dataset and treating them is critical as it affect the accuracy of our prediction. However, fortunately, in our dataset, we can see that there aren't any missing vales.

```
> bc[!complete.cases(bc),]
 [1] diagnosis                radius_mean              texture_mean
 [4] perimeter_mean           area_mean                smoothness_mean
 [7] compactness_mean         concavity_mean           concave.points_mean
[10] symmetry_mean            fractal_dimension_mean   radius_se
[13] texture_se               perimeter_se             area_se
[16] smoothness_se            compactness_se           concavity_se
[19] concave.points_se        symmetry_se              fractal_dimension_se
[22] radius_worst             texture_worst            perimeter_worst
[25] area_worst               smoothness_worst         compactness_worst
[28] concavity_worst          concave.points_worst     symmetry_worst
[31] fractal_dimension_worst
<0 rows> (or 0-length row.names)
>
```

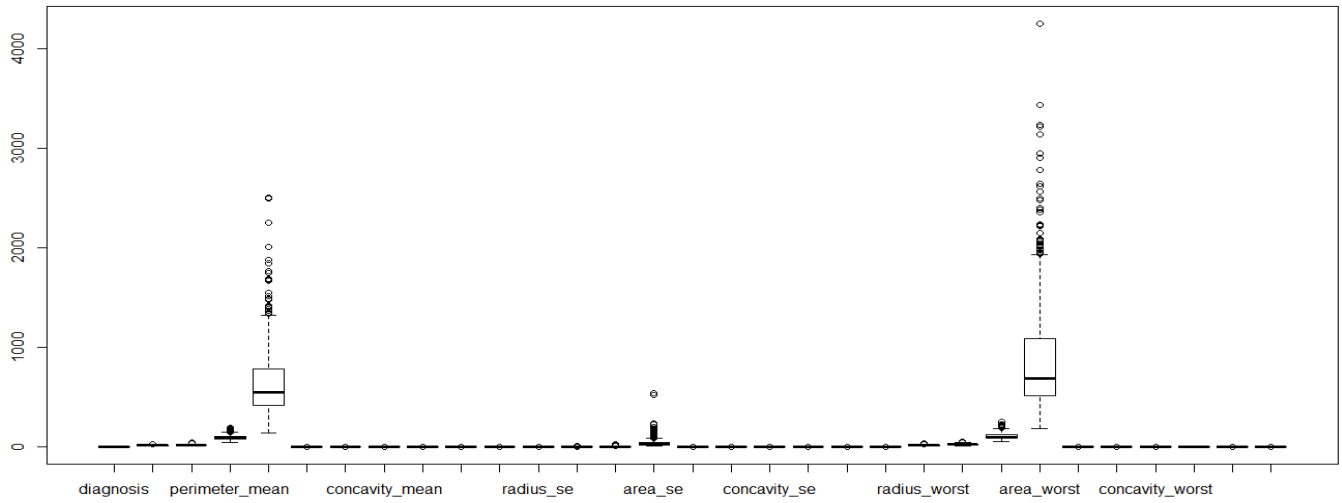
- **Converting the response variable into a factor:** The response variable in our dataset is the diagnosis column which we are going to predict. As we

can see that it is currently classified just as a 'character', we convert it into a 'factor' variable with levels 'Benign' and 'Malignant' to teach the algorithm the difference between the two levels.

```
> class(bc$diagnosis)
[1] "character"
> str(bc$diagnosis)
 chr [1:569] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" ...
```

```
> bc$diagnosis <- factor(ifelse(bc$diagnosis=="B", "Benign", "Malignant"))
> class(bc$diagnosis)
[1] "factor"
> str(bc$diagnosis)
 Factor w/ 2 levels "Benign","Malignant": 2 2 2 2 2 2 2 2 2 2 ...
```

- **Checking for outliers:** Identifying outliers is another important step, as outliers could be incorrect values in the distribution that again affect the accuracy of prediction. In our dataset, we can see some significant outliers in 'area' attribute, which in turn affect the 'area mean', 'area se', and 'area worst' variables. However, we choose to ignore these outliers because, as per medical records, these tumors could grow to any size with no restrictions. Based on this assumption, we consider the data to be legit and choose to ignore these outliers.

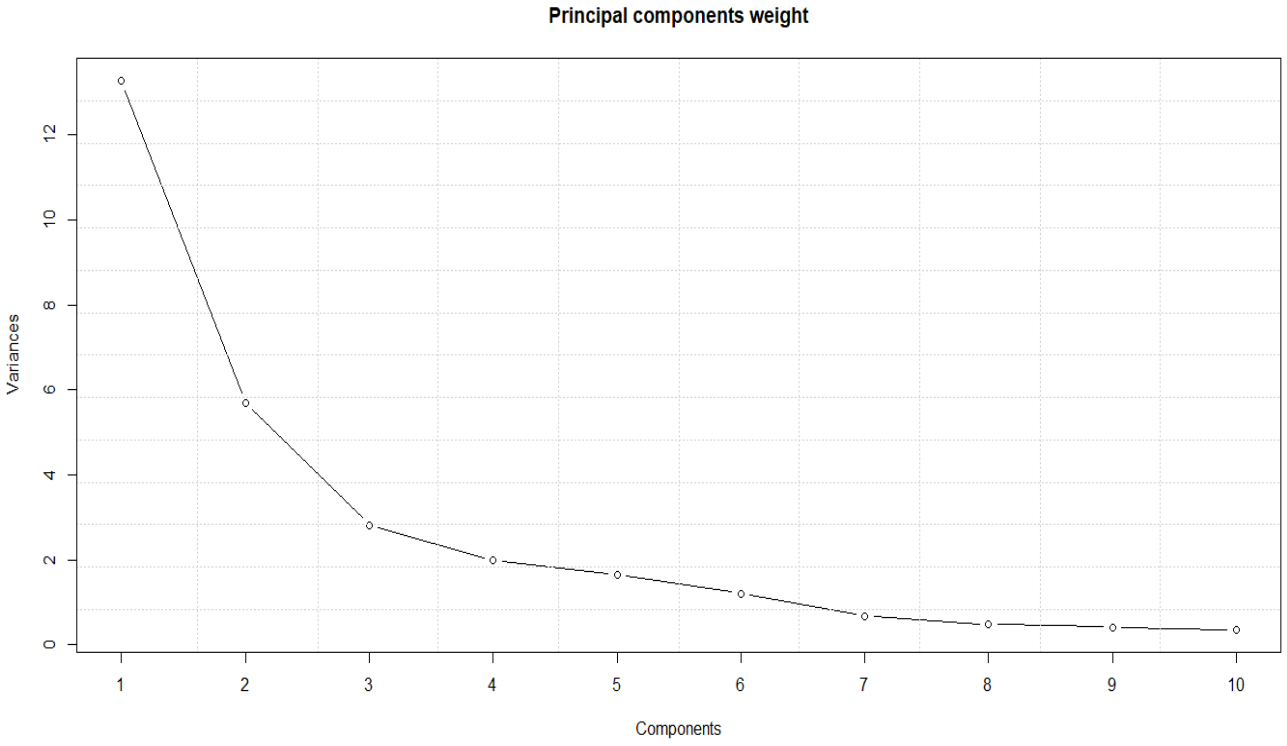


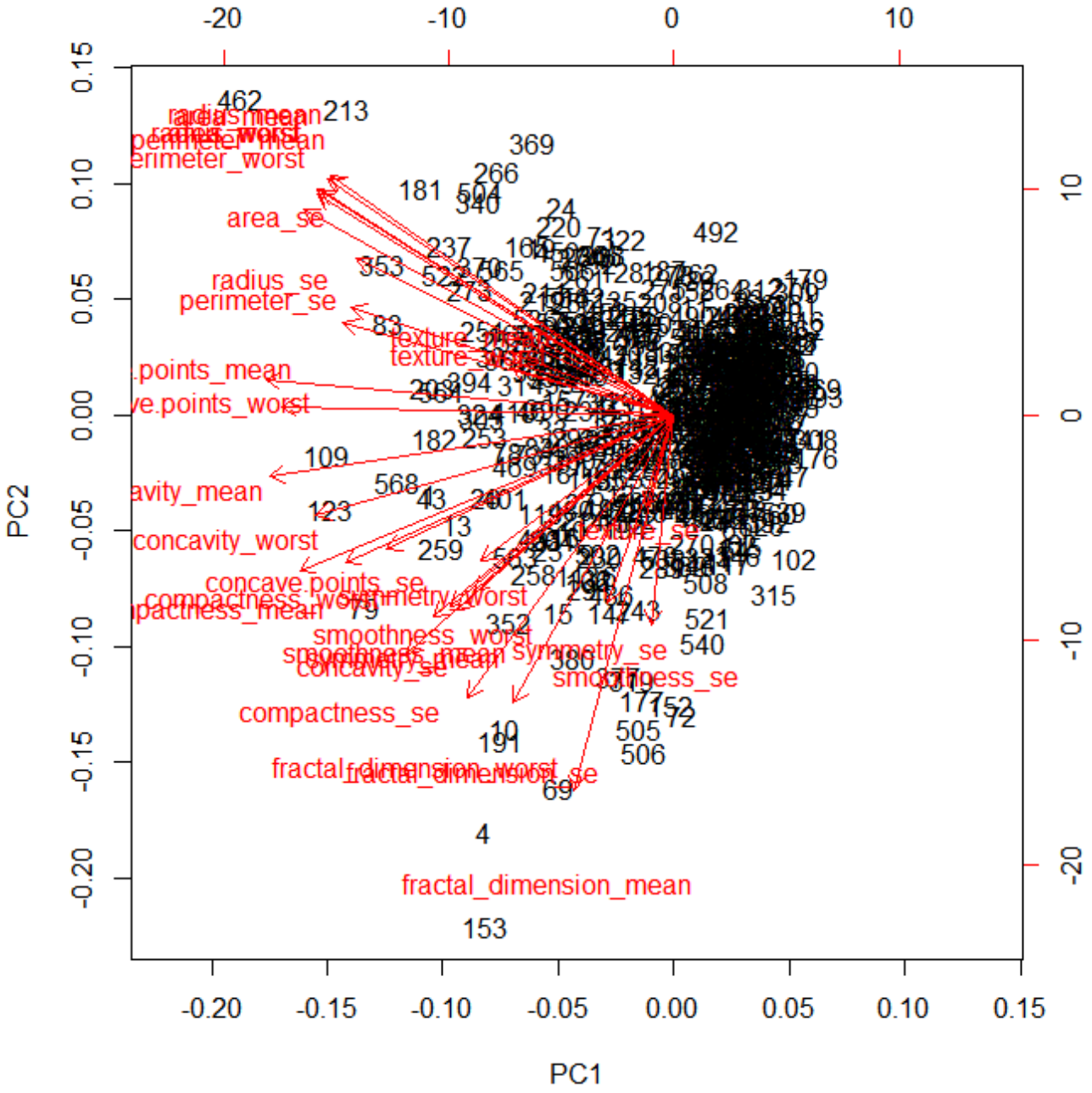
PRINCIPAL COMPONENT ANALYSIS

Too many variables can be a hindrance for effective data analysis as it makes the interpretation difficult and also decreases the efficiency by having variables that have no effect. Here, we analyze this on our dataset to reduce the variables with high correlation and standardize the data so that it can avoid data distortion caused by scale difference.

While performing the PCA on our dataset, we can see that 90% of dataset variation involves in just the first 7 components, while the remaining 23 components contributing just the balance 10%.

```
> summary(bc.pca)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759
                          PC7     PC8    PC9    PC10   PC11    PC12
Standard deviation     0.82172 0.69037 0.6457 0.59219 0.5421 0.51104
Proportion of Variance 0.02251 0.01589 0.0139 0.01169 0.0098 0.00871
Cumulative Proportion  0.91010 0.92598 0.9399 0.95157 0.9614 0.97007
                          PC13    PC14    PC15    PC16    PC17    PC18
Standard deviation     0.49128 0.39624 0.30681 0.28260 0.24372 0.22939
Proportion of Variance 0.00805 0.00523 0.00314 0.00266 0.00198 0.00175
Cumulative Proportion  0.97812 0.98335 0.98649 0.98915 0.99113 0.99288
                          PC19    PC20   PC21    PC22    PC23   PC24
Standard deviation     0.22244 0.17652 0.1731 0.16565 0.15602 0.1344
Proportion of Variance 0.00165 0.00104 0.0010 0.00091 0.00081 0.0006
Cumulative Proportion  0.99453 0.99557 0.9966 0.99749 0.99830 0.9989
                          PC25    PC26    PC27    PC28    PC29    PC30
Standard deviation     0.12442 0.09043 0.08307 0.03987 0.02736 0.01153
Proportion of Variance 0.00052 0.00027 0.00023 0.00005 0.00002 0.00000
Cumulative Proportion  0.99942 0.99969 0.99992 0.99997 1.00000 1.00000
```

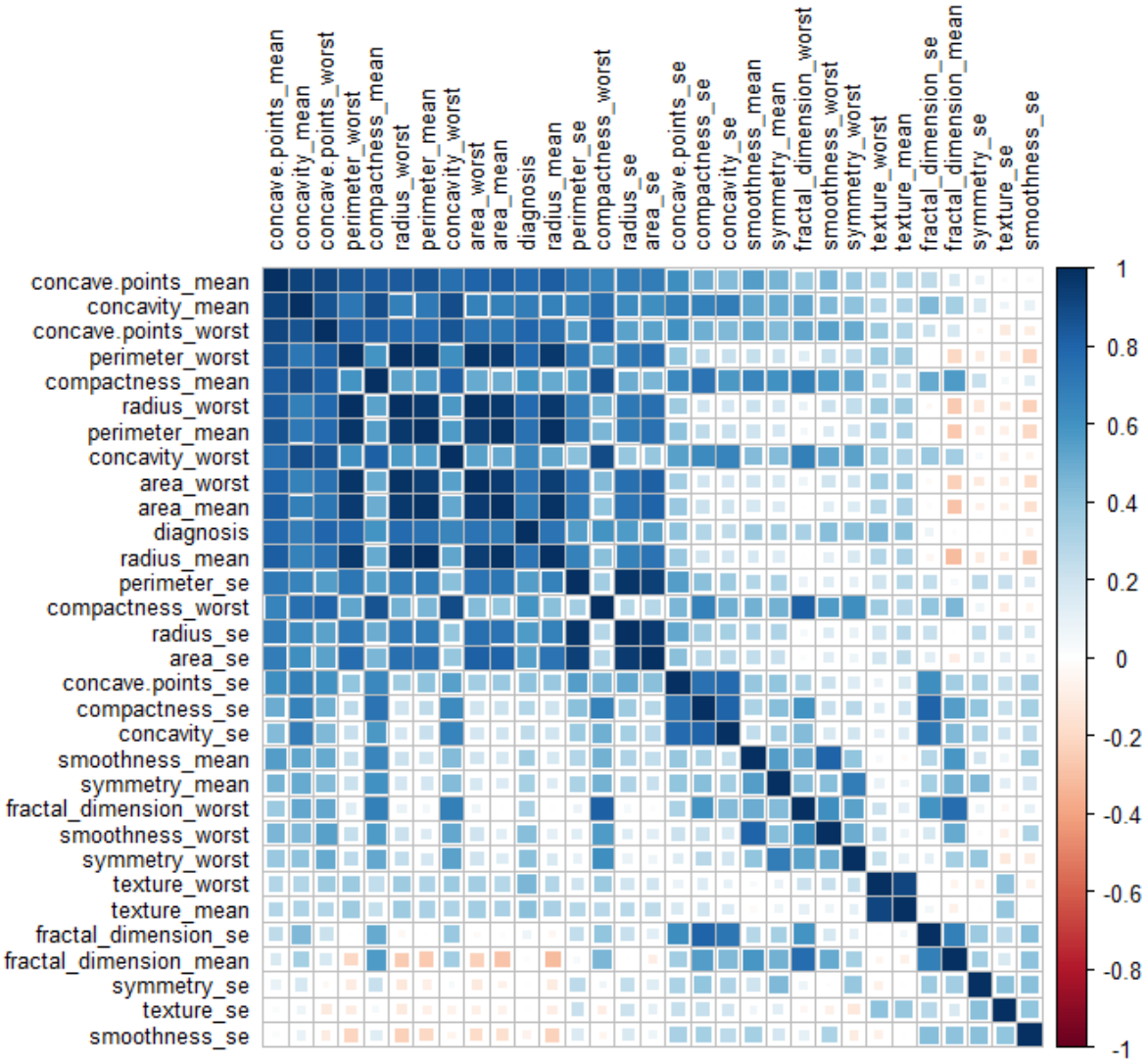




CORRELATION ANALYSIS

A model is considered to come out well if the response variable has high correlation with the independent variables, while the independent variables are not correlated with each other. Here, we perform a

correlation analysis to see which of these independent attributes show high correlation with each other.

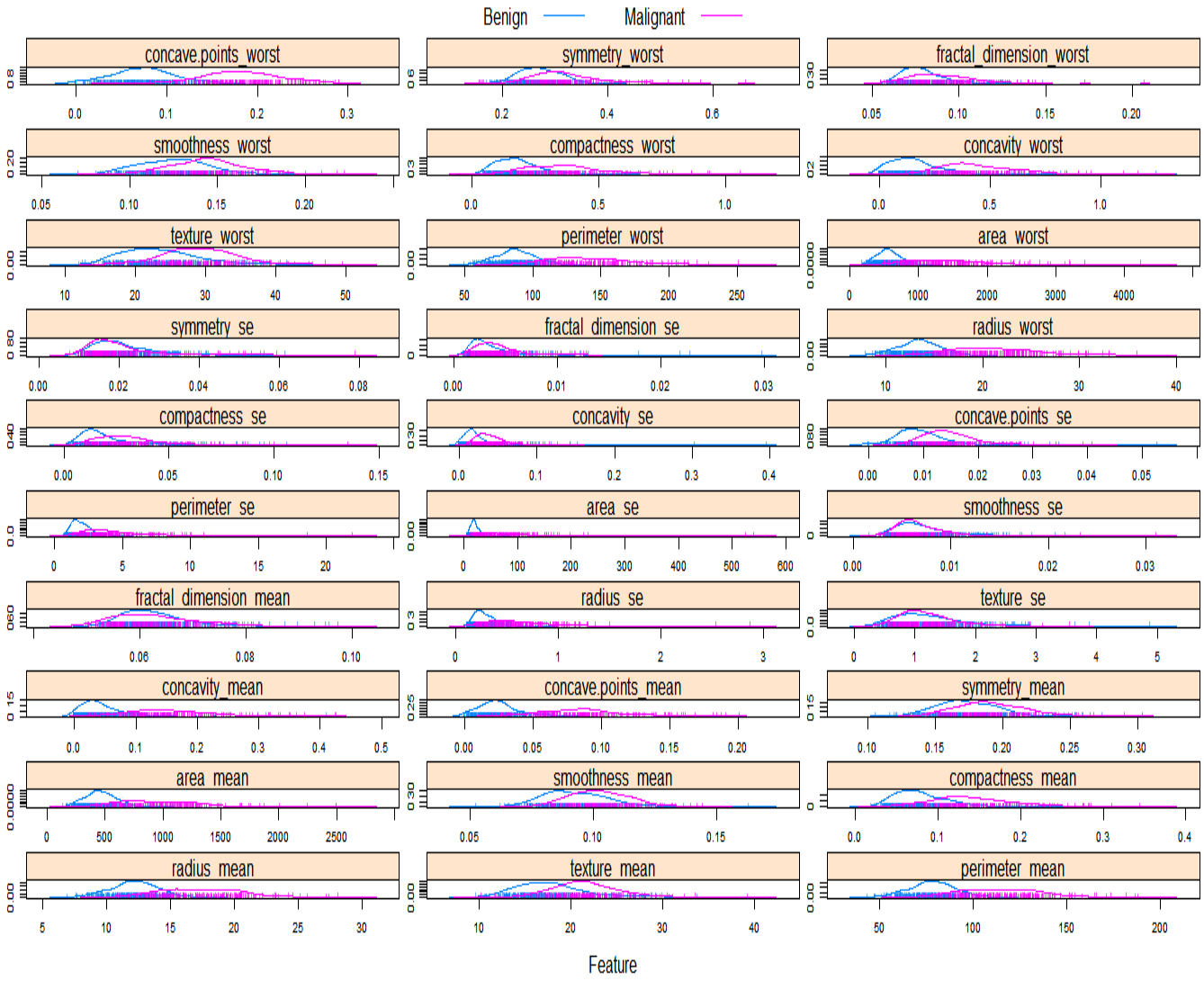


Through this Pearson correlation analysis, we can see high correlations between the following variables,

- Perimeter mean – radius worst
- Area worst – radius worst
- Perimeter worst – radius worst, area worst, area mean, radius mean
- Texture mean – texture worst

FEATURE DENSITY PLOT

Here we perform a data exploration to check the feature density for each of the attributes to see how the proportion of our two cases (Benign and Malignant) is in each of the variables.



2.3. DATA MODELLING

Modelling is the main phase in the CRISP-DM methodology which involves in the process of fitting models based on our dataset and using this model for predicting the new data.

We have used the following 'classification' machine learning algorithms for this stage,

- Naïve Bayes
- Random Forest
- Support Vector Machine (SVM)

Here, we split our refined dataset into train and test set in 70:30 ratio to model the train set and make predictions with the help of test set. Then we compare these predictions with the actual test set values to calculate its accuracy.

```
> dim(bc)
[1] 569  31
> dim(train)
[1] 398  31
> dim(test)
[1] 171  31

> prop.table(table(train$diagnosis))

   Benign Malignant
0.6281407 0.3718593
> prop.table(table(test$diagnosis))

   Benign Malignant
 0.625731  0.374269
```

We can also see that our proportion of split in train set and test set is pretty similar making the split fair for modelling and prediction.

a) Naïve Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the

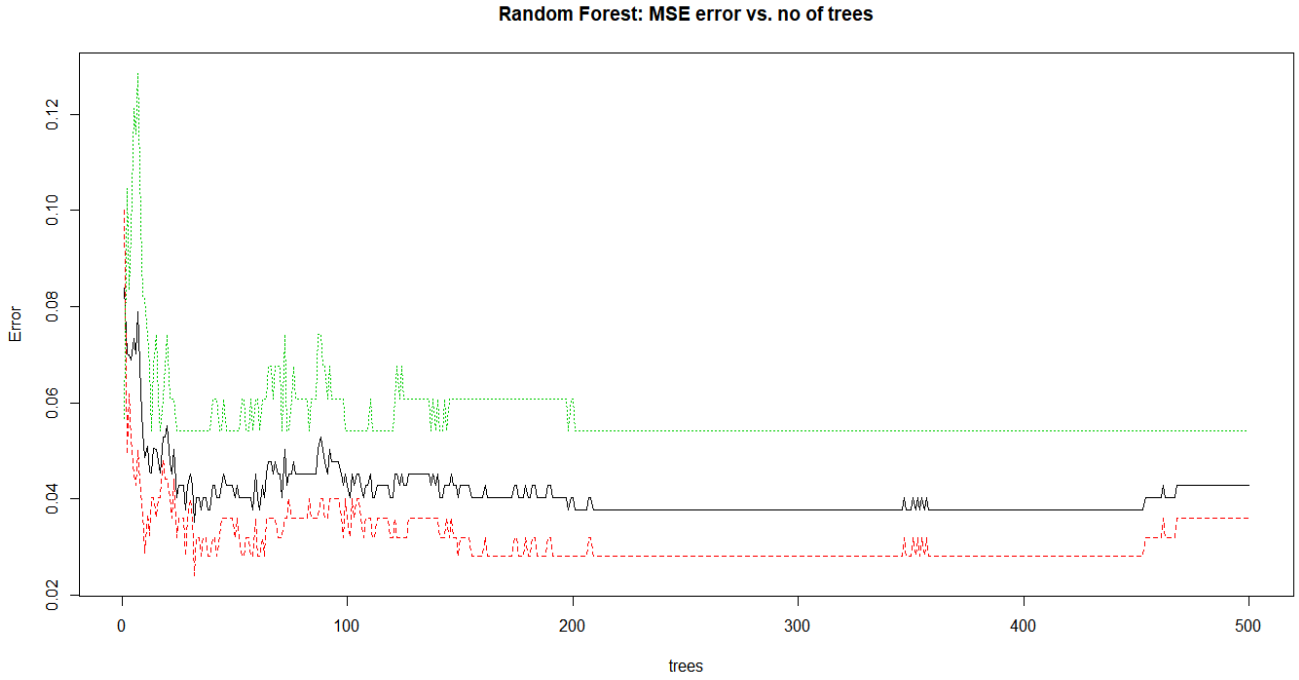
presence of any other feature [5]. With the help of Naïve Bayes classification technique, we were able to build a model with 92% of accuracy in prediction of the test set with 13 incorrect predictions.

```
Confusion Matrix and Statistics

           Reference
Prediction  Benign Malignant
  Benign       101         7
  Malignant      6        57

               Accuracy : 0.924
                 95% CI : (0.8735, 0.9589)
    No Information Rate : 0.6257
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8372
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9439
            Specificity : 0.8906
         Pos Pred Value : 0.9352
         Neg Pred Value : 0.9048
             Prevalence : 0.6257
         Detection Rate : 0.5906
   Detection Prevalence : 0.6316
      Balanced Accuracy : 0.9173

       'Positive' Class : Benign
```

b) RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the

mode of the classes (classification) or mean prediction (regression) of the individual trees [3]. With the help of Random Forest classification technique, we were able to build a model with an accuracy of 96% in prediction of the test set with 6 (2 false-positive and 4 false-negative) incorrect predictions.




```
Confusion Matrix and Statistics

          Reference
Prediction  Benign Malignant
  Benign       105         4
  Malignant      2        60

               Accuracy : 0.9649
                 95% CI : (0.9252, 0.987)
    No Information Rate : 0.6257
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9246
 Mcnemar's Test P-Value : 0.6831

            Sensitivity : 0.9813
            Specificity : 0.9375
         Pos Pred Value : 0.9633
         Neg Pred Value : 0.9677
             Prevalence : 0.6257
         Detection Rate : 0.6140
   Detection Prevalence : 0.6374
      Balanced Accuracy : 0.9594

       'Positive' Class : Benign
```

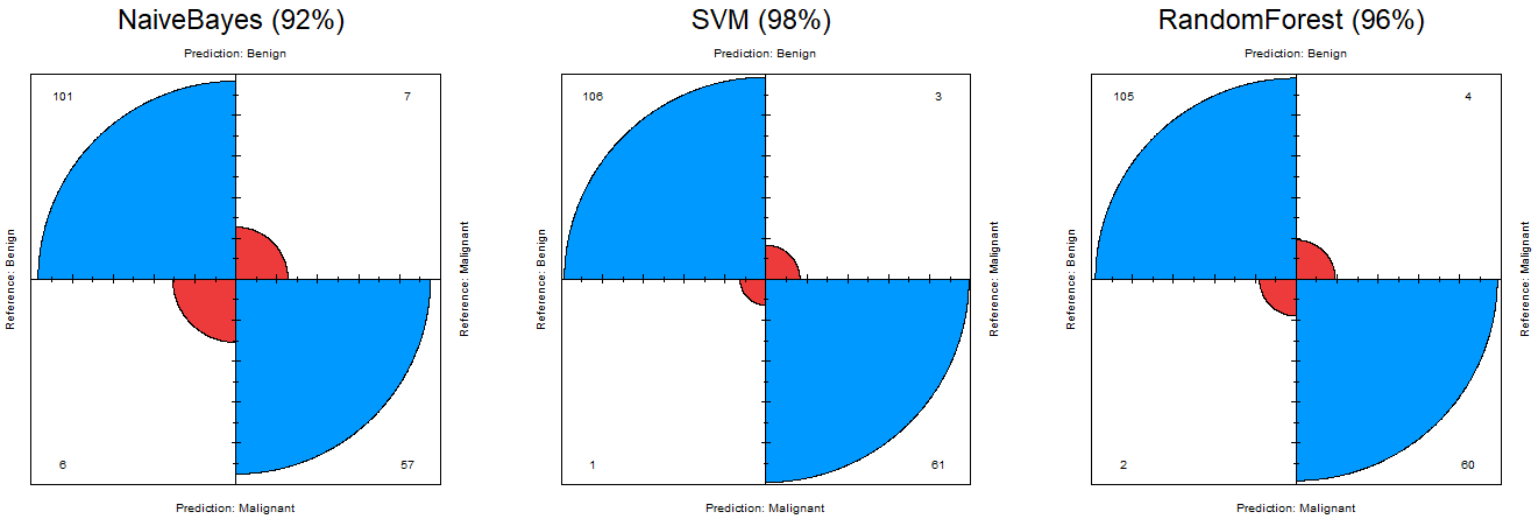
c) SUPPORT VECTOR MACHINE (SVM)

A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are called support vectors [4]. SVM is also insensitive to the outliers, which is an added advantage for us, as we chose to retain the outliers in our dataset. With the help of Support Vector Machines (SVM) classification technique, we were able to build a model with an accuracy of 98% in prediction of the test set with only 4 (1 false-positive and 3 false-negative) incorrect predictions.

```
Confusion Matrix and Statistics

           Reference
Prediction  Benign Malignant
  Benign       106         3
  Malignant      1        61

               Accuracy : 0.9766
                 95% CI : (0.9412, 0.9936)
    No Information Rate : 0.6257
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9497
 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9907
            Specificity : 0.9531
         Pos Pred Value : 0.9725
         Neg Pred Value : 0.9839
             Prevalence : 0.6257
         Detection Rate : 0.6199
   Detection Prevalence : 0.6374
      Balanced Accuracy : 0.9719

       'Positive' Class : Benign
```

2.4. DATA EVALUATION AND DEPLOYMENT

This is the stage where we pick the desired model and deploy it for making the predictions. Now that we have built 3 models in our case, let's compare their performances to determine which model performs the best with the help of four-fold plot technique.



The blue planes on the above charts denote the True-Positives and True-Negatives, while the red planes denote the False-Positives and False-Negatives. It is evident from the graph that SVM has the smallest red planes of all, meaning it has the least number of incorrect predictions

Hence, we can see that among these three models, Support Vector Machine (SVM) has performed the best with an accuracy of 98% making it the most effective technique for making predictions.

K-FOLD CROSS VALIDATION

Although SVM works better on our test set, it is quite important and beneficial to do a cross validation on the dataset just to double-check the accuracy and see if it's consistent. Since our dataset is also smaller, it is quite important to

perform the k-fold cross validation technique to see if the model's performance is consistent.

K-fold Cross Validation is a simple, yet important technique where the dataset is split evenly into k-folds. During each iteration, one of the folds act as a test set, while all the other folds act as train set. Finally, the accuracy of all the iterations are averaged to get the final accuracy.

```
# K-fold cross validation for svm with 10 folds
ControlParameters <- trainControl(method = 'repeatedcv',
                                  number = 10, repeats = 5,
                                  savePredictions = TRUE,
                                  classProbs = TRUE)


modelsvm <- train(diagnosis~., data=bc, method = "svmLinear",
                  trControl = ControlParameters)

modelsvm

tune.svm(diagnosis~., data= bc) #gives error estimation
```

Here, we have set the 'K' to 10, hence it has folds, with 5 iterations.

```
> modelsvm
Support Vector Machines with Linear Kernel

569 samples
 30 predictor
  2 classes: 'Benign', 'Malignant'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 513, 512, 513, 512, 512, 512, ...
Resampling results:

  Accuracy   Kappa
  0.9704675  0.9362348

Tuning parameter 'C' was held constant at a value of 1
> tune.svm(diagnosis~., data= bc) #gives error estimation

Error estimation of 'svm' using 10-fold cross validation: 0.02637845
```

From the above output, we can see that our accuracy through 10-fold cross validation is 97%, which is just 0.6% lower than our initial test, though it doesn't make a big difference, it proves that our model is consistent in performance. We can also see that the error estimation for the model is only 2%, yet again proving that SVM model outperforms all the other models.

DEPLOYMENT

The deployment stage involves in deploying the chosen model into the system for making the predictions. In our case, since SVM model performed the best, we will be deploying that to test a new non-diagnosed data and see if the model can predict the correct type of tumor.

```
# testing the best model for prediction
patient <- read.csv(file.choose(), header=T, stringsAsFactors=F)
patient$X <- NULL

#malignent patient
M <- patient[35,]           ## 35th patient
M[,c(1,2)]

#benign patient
B <- patient[20,]               ## 20th patient
B[,c(1,2)]

#delete diagnosis column for testing
M$diagnosis <- NULL
B$diagnosis <- NULL

# patient cancer diagnosis prediction function
# for printing output
cancer_diagnosis_predict_print <- function(new, method=learn_svm) {
  new_pre <- predict(method, new[,-1])
  new_res <- as.character(new_pre)
  return(paste("Patient ID: ",new[,1],"  =>  Result: ", new_res, sep=""))
}

cancer_diagnosis_predict_print(B)
cancer_diagnosis_predict_print(M)
```

Results:

```
> cancer_diagnosis_predict_print(B)
[1] "Patient ID: 8510426  =>  Result: Benign"
> cancer_diagnosis_predict_print(M)
[1] "Patient ID: 854039  =>  Result: Malignant"
```

As we can see, the model predicts the type of tumor for the given patient ID.

3. CONCLUSION

With a data mining approach, we were successfully able to perform data analysis on the Wisconsin's Breast Cancer dataset by following the CRISP-DM methodology of understanding the data, refining the data by fixing the inconsistencies, then by modelling the data for making prediction and finally by deploying the SVM model which performed the best. Then we did test for the deployed SVM model by feeding data which was not diagnosed, and the results were impressive by predicting the correct type of tumor.

4. REFERENCES

[1] UCI – Wisconsin Breast Cancer Dataset [online] Available at: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[2] Saiyed, S., Bhatt, N. and Ganatra, A. (2016). A Survey on Naive Bayes Based Prediction of Heart Disease Using Risk Factors. International Journal of Innovative and Emerging Research in Engineering, Vol. 3., Issue (2).

[3] Ho, T. (2018). Random Decision Forests. [online] Available at: https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf [Accessed 14 Aug. 2018].

[4] Sayed, S. Support Vector Machine. [online] Saedsayad.com. Available at: http://saedsayad.com/support_vector_machine.htm [Accessed 14 Aug. 2018].

[5] Ray, S. (2018). Naive Bayes Algorithm (with code in Python). [online] Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ [Accessed 14 Aug. 2018].