I have taken the dataset Breast Cancer Wisconsin Diagnostic Dataset from Kaggle. This data has some complex attributes and has 569 rows and 32 columns. Its attributes include id. Perimeter, area, radius, and several other factors involving in the diagnosis of breast cancer. I have done a comprehensive analysis on the dataset and used SVM models to analyze the data.

Analysis:

I installed a few packages and checked the libraries initially after that I have loaded the dataset

bcdiagnosis <- read.csv("c:/Users/ Harish Bodasinghi/ Desktop/ Breastcancerdiagnosis.csv")

After loading the data, I have done initial data exploration. I have used functions like head, summary, andstructure to know more about the data, it has 569 rows and 32 columns when used nrow ncol functions.

```
> summary(bcdiagnosis)
       id             diagnosis  radius_mean      texture_mean    perimeter_mean
 Min.   :     8670   0:357      Min.   : 6.981   Min.   : 9.71   Min.   : 43.79
 1st Qu.:   869218   1:212      1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17
 Median :   906024              Median :13.370   Median :18.84   Median : 86.24
 Mean   : 30371831              Mean   :14.127   Mean   :19.29   Mean   : 91.97
 3rd Qu.:  8813129              3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10
 Max.   :911320502              Max.   :28.110   Max.   :39.28   Max.   :188.50
   area_mean      smoothness_mean   compactness_mean  concavity_mean    concave.points_mean
 Min.   : 143.5   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
 1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
 Median : 551.1   Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
 Mean   : 654.9   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
 3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
 Max.   :2501.0   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
 symmetry_mean    fractal_dimension_mean   radius_se        texture_se      perimeter_se
 Min.   :0.1060   Min.   :0.04996        Min.   :0.1115   Min.   :0.3602   Min.   : 0.757
 1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606
 Median :0.1792   Median :0.06154        Median :0.3242   Median :1.1080   Median : 2.287
 Mean   :0.1812   Mean   :0.06280        Mean   :0.4052   Mean   :1.2169   Mean   : 2.866
 3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357
 Max.   :0.3040   Max.   :0.09744        Max.   :2.8730   Max.   :4.8850   Max.   :21.980
    area_se         smoothness_se      compactness_se      concavity_se
 Min.   :  6.802   Min.   :0.001713   Min.   :0.002252   Min.   :0.00000
 1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509
 Median : 24.530   Median :0.006380   Median :0.020450   Median :0.02589
 Mean   : 40.337   Mean   :0.007041   Mean   :0.025478   Mean   :0.03189
 3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205
 Max.   :542.200   Max.   :0.031130   Max.   :0.135400   Max.   :0.39600
 concave.points_se   symmetry_se       fractal_dimension_se  radius_worst    texture_worst
 Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02
 1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08
 Median :0.010930   Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41
 Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68
 3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72
 Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
 perimeter_worst     area_worst     smoothness_worst  compactness_worst concavity_worst
 Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000
 1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145
 Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267
 Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722
```

```
> nrow(bcdiagnosis)
[1] 569
> ncol(bcdiagnosis)
[1] 32
>
```

Initially when seen by using structure we find that the target variable diagnosis is a categorical variable then we try to convert it to numerical and set it like Benign as 0 and Malignant as 1.

```
> bcdiagnosis$diagnosis = factor(bcdiagnosis$diagnosis,levels = c('B', 'M'), labels = c(0, 1))
```

I have checked it again using str and head, so I have noticed that diagnosis has been changed to numerical from categorical.

If I make the table for diagnosis of the dataset then;

```
> #Benign-B is 0  & Malignant-M is 1;
> table(bcdiagnosis$diagnosis)

0 1
0 0
>
```

By splitting the data set into train and test sets and checking the dimensions of it, we get;

```
> train <- bcdiagnosis[intrain,]
> test <- bcdiagnosis[-intrain,]
> #Checking the dimensions of train and test samples
> dim(train)
[1] 370  32
> dim(test)
[1] 199  32
>
```

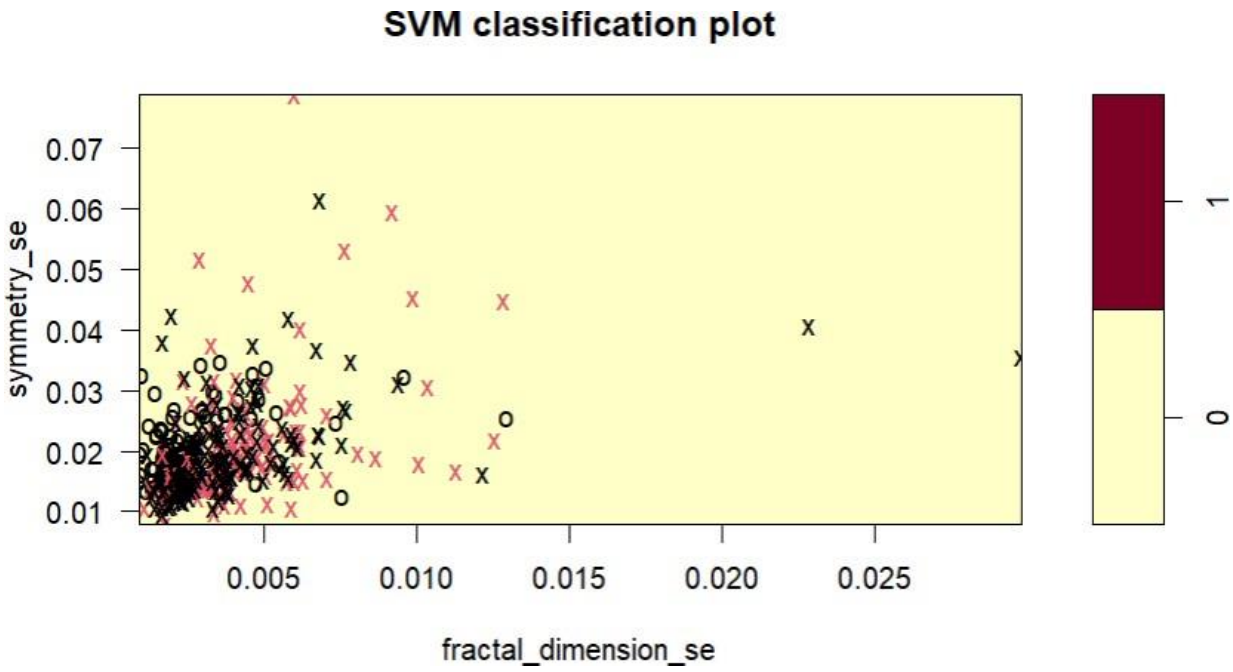
Further I have created SVM model which is the first one the train data set and we will try to predict for the test data set.

```
> # Support Vector Machine Model 1
> svm.linear.model.1 <- svm(diagnosis~symmetry_se+fractal_dimension_se,
+                           data = train,
+                           type = "C-classification",
+                           kernel="polynomial",
+                           scale = FALSE)
> svm.linear.model.1

Call:
svm(formula = diagnosis ~ symmetry_se + fractal_dimension_se, data = train,
    type = "C-classification", kernel = "polynomial", scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  1
     degree:  3
     coef.0:  0

Number of Support Vectors:  276
```

The mean of train set after prediction tends to be:

```
> pred_train <- predict(svm.linear.model.1, train)
> mean(pred_train == train$diagnosis)
[1] 0.6280323
```

Model 1 gives us this kind of plot with x axis being fractal_dimesion_se and y axis being symmetry_se



Later we try to predict for the test data set using 1 st model

```
> test_pred <- predict(svm.linear.model.1, newdata = test)
```

and find the confusion matrix for the same.

```
> #Confusion Matrix
> confusionMatrix(table(test_pred, test$diagnosis))
Confusion Matrix and Statistics

          
test_pred   0   1
        0 124  74
        1   0   0

               Accuracy : 0.6263          
                 95% CI : (0.5549, 0.6938)
    No Information Rate : 0.6263          
    P-Value [Acc > NIR] : 0.5317          

                  Kappa : 0               

 Mcnemar's Test P-Value : <2e-16          

            Sensitivity : 1.0000          
            Specificity : 0.0000          
         Pos Pred Value : 0.6263          
         Neg Pred Value :    NaN          
             Prevalence : 0.6263          
         Detection Rate : 0.6263          
   Detection Prevalence : 1.0000          
      Balanced Accuracy : 0.5000          

       'Positive' Class : 0
```

Kappa value is zero meaning that it indicates no agreement.

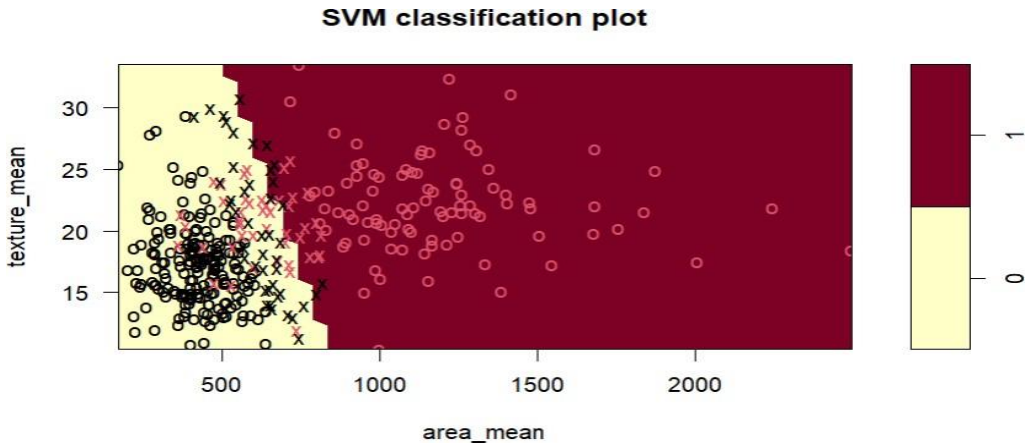
Then second model is being tried

```
> # Support Vector Machine Model 2
> svm.linear.model.2 <- svm(diagnosis~texture_mean+area_mean,
+                           data = train,
+                           type = "C-classification",
+                           kernel="linear",
+                           scale = FALSE)
```

We also calculate the mean for this train set also;

```
> pred_train <- predict(svm.linear.model.2, train)
> mean(pred_train == train$diagnosis)
[1] 0.8948787
```

We get the plot for second svm model as;



We also try to predict for the test data set

```
> #Prediction (Test Set)
> test_pred <- predict(svm.linear.model.2, newdata = test)
```

The confusion matrix states that;

```
> #Confusion Matrix
> confusionMatrix(table(test_pred, test$diagnosis))
Confusion Matrix and Statistics

          
test_pred   0   1
        0 117  19
        1   7  55

               Accuracy : 0.8687          
                 95% CI : (0.8135, 0.9124)
    No Information Rate : 0.6263          
    P-Value [Acc > NIR] : 2.599e-14       

                  Kappa : 0.71            

 Mcnemar's Test P-Value : 0.03098         

            Sensitivity : 0.9435          
            Specificity : 0.7432          
         Pos Pred Value : 0.8603          
         Neg Pred Value : 0.8871          
             Prevalence : 0.6263          
         Detection Rate : 0.5909          
   Detection Prevalence : 0.6869          
      Balanced Accuracy : 0.8434          

       'Positive' Class : 0
```

Kappa value seems to be 0.71 which is a strong agreement.

I have further went to test for model 3 of SVM on the training set

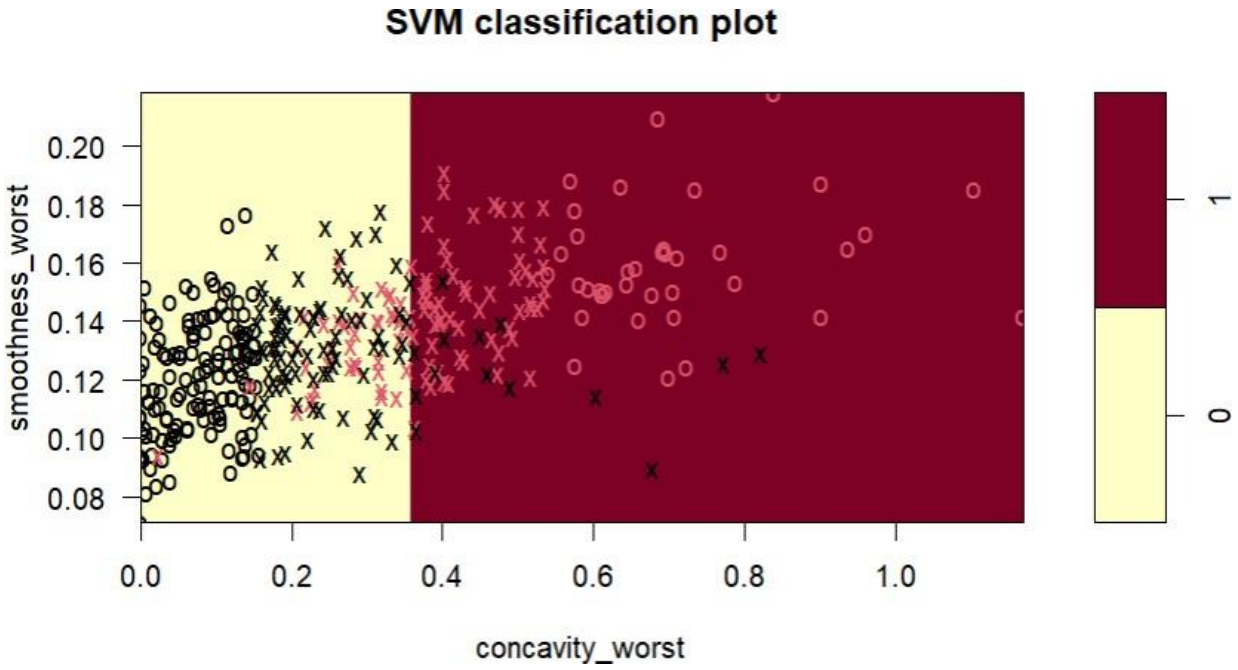
```
> # Support Vector Machine Model 3
> svm.linear.model.3 <- svm(diagnosis~smoothness_worst+concavity_worst,
+                           data = train,
+                           type = "C-classification",
+                           kernel="sigmoid",
+                           scale = FALSE)
```

```
> svm.linear.model.3

Call:
svm(formula = diagnosis ~ smoothness_worst + concavity_worst, data = train,
    type = "C-classification", kernel = "sigmoid", scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  sigmoid
       cost:  1
     coef.0:  0

Number of Support Vectors:  200
```

Mean prediction for this model tends to be;

```
> pred_train <- predict(svm.linear.model.3, train)
> mean(pred_train == train$diagnosis)
[1] 0.8571429
```

Plotting the model 3 looks like this;



```
> #Prediction (Test Set)
> test_pred <- predict(svm.linear.model.3, newdata = test)
```

Confusion matrix for test set will be

```
> #Confusion Matrix
> confusionMatrix(table(test_pred, test$diagnosis))
Confusion Matrix and Statistics


test_pred   0   1
        0 115  28
        1   9  46

               Accuracy : 0.8131
                 95% CI : (0.7517, 0.8649)
    No Information Rate : 0.6263
    P-Value [Acc > NIR] : 8.813e-09

                  Kappa : 0.579

 Mcnemar's Test P-Value : 0.003085

            Sensitivity : 0.9274
            Specificity : 0.6216
         Pos Pred Value : 0.8042
         Neg Pred Value : 0.8364
             Prevalence : 0.6263
         Detection Rate : 0.5808
   Detection Prevalence : 0.7222
      Balanced Accuracy : 0.7745

       'Positive' Class : 0
```

Here Kappa is 0.5 which means it is a good agreement.