Assignment 6: Theory of Support Vector Machine

Shashwat Patel
Metallurgical and Materials Engineering
Indian Institute of Technology,Madras
mm19b053@smail.iitm.ac.in

***Abstract*—Support vector machine is a supervised machine learning algorithm that is used for solving both classification and regression problem but it is mostly used for classification. The data is plotted in n-dimensional space with the value of each feature being the value of a particular coordinate. Support Vectors are simply the coordinates of individual observation. In this paper , we explain the theory of support vector machine and try to understand what parameters are helpful in determining whether a neutron star is a pulsar or not.**

***Index Terms*—Pulsar, SVM, Support vectors, Hyperplane, Kernel, Kernel Trick, MICE**

I. Introduction

Pulsars are a rare type of Neutron star that produces radio emissions detectable here on Earth.They emits beams of electromagnetic radiation out of its magnetic poles. This radiation is observed only when the beam of emission is pointing towards the Earth. Neutron stars are very dense and have short, regular rotational periods. Pulsars are one of the candidates for the source of ultra-high-energy cosmic rays They are of considerable scientific interest as probes of space-time, the inter-stellar medium, and states of matter. Observations of a pulsar in a binary neutron star system are used to indirectly confirm the existence of gravitational radiation [1].

Support vector machine is a supervised machine learning algorithm that is used to solve both regression and classification problems. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N is the number of features) that distinctly classifies the data points. Hyperplanes are decision boundaries that help in classifying the data points. Data points falling on either side of the hyperplane are attributed to different classes.

Support vector machine finds a lot of application in real life. This algorithm is being used in face detection, image classification, bioinformatics, text classification etc. SVMs also reduce the redundant information. [2]

Pulsars are a type of neutron star that produce electromagnetic radiation that are detected only when the emission is towards the earth. Pulsars are the potential source for high energy cosmic rays and are of high interest for the astronomers. The data is used to identify those parameters that helps in studying and classifying the neutron stars as pulsars or not. This classification is done using the support vector classifier.

This paper majorly deals with the theory of support vector machines and the mathematics behind it. We try to understand the key parameters that helps in classifying a neutron star into a pulsar or not. Two datasets have been given: ”pulsar data train.csv” and ”pulsar data test.csv”, using the training dataset we train support vector classifier model and test the model on the ”pulsar test data.csv”. The problem that we are going to solve is a classification problem so Support Vector Classifier has been described.

II. Support Vector Machine

In the SVM algorithm, data is plotted as a point in a n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, the classification is done by finding the hyper-plane that differentiates the two classes very well. Support Vectors are the coordinates of individual observation.

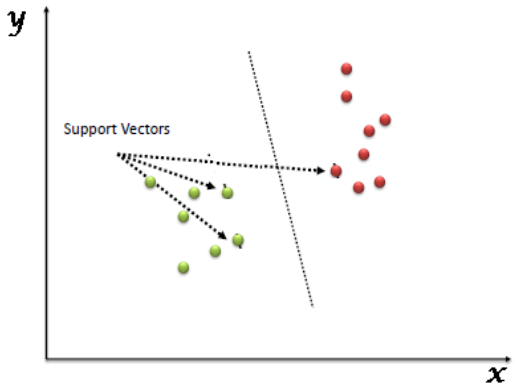

Fig. 1. Support Vector Classifier

To separate the two classes of data points, there are many different possible hyper-plane that can be chosen [3]. The main objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

*Working of support vector classifier:*Hyper-planes are decision boundaries that help classify the data points. Data points falling on either side of the hyper-plane can be attributed to different classes. The dimension of hyper-plane depends on the

number of features, if number of features is 3, then dimension of hyper-plane is 2 and geometrically, it is a plane.

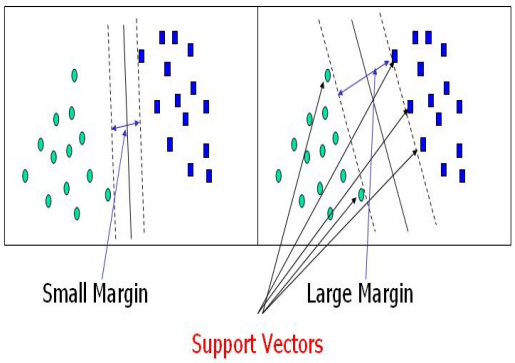

Fig. 2. Support Vectors and the margin

Figure 2 shows the support vectors and the margin which help in classifying the data. Support vectors are the data points that are closer to the hyper-plane the most and influence the position and orientation of the hyper-plane. Using these support vector, the margin of the classifier is maximized. Changing the position of the support vectors will also influence the position of hyper-plane [3]. In SVM, if the output of our linear function and is greater than 1, we identify it with one class and if the output is -1, it is identified with another class [4].

$$c(x, y, f(x)) = \begin{cases} 0 & y * f(x) \geq 1 \\ 1 - y * f(x) & else \end{cases} \quad (1)$$

Equation 1 is the loss function that helps in maximizing the margin of the classifier. The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, then the loss value is calculated. A regularization parameter is also set to balance the margin maximization and the loss. After adding the parameter our cost function looks like [4]:

$$min_w \lambda ||w||^2 + \sum_{i=1}^{n} (1 - y_i < x_i, w >) \quad (2)$$

This is the case for a linear decision boundary, but while considering non linear data,a linear decision boundary will misclassify the data. For this we create another dimension to deal with this. This is one of the method but this method is complex in the sense that the transformation is computationally expensive. Kernel trick [5] helps in this computation and it is time effective as well. In general, the kernel is linear and we get a linear classifier. But by using a nonlinear classifier, we don't need to do the transformations, we only have to change the dot product to that of the required space that we want and it will be done.

There are different types of kernels used in SVM, polynomial kernel is used generally during image processing, Gaussian kernel is a general purpose kernel used when there is no prior knowledge, hyperbolic tangent kernel used in neural network, ANOVA radial basis kernel used in regression analysis and many other different kinds of kernel.

III. The Problem

Pulsars are a type of neutron star that produce electromagnetic radiation that are detected only when the emission is towards the earth. Pulsars are the potential source for high energy cosmic rays and are of high interest for the astronomers. "Pulsar train data" has been used to train a support vector classifier model and "Pulsar test data" is used to predict a neutron star as pulsar or not.

The dataset consists of features like mean, standard deviation, skewness, excess kurtosis of Integrated profile and DM-SNR curve, target_class. The train data has 12528 observations while the test data consists of 5370 observations.

In training data, 1735 observations are missing in excess kurtosis of integrated profile, 1178 observations missing in standard deviation of DM-SNR curve, 625 observations missing in skewness of DM-SNR curve. In total around 3.5% of training data is missing. In test data, 767 observations missing in excess kurtosis of integrated profile, 524 observations missing in standard deviation of DM-SNR curve. 244 observations missing in skewness of DM-SNR curve. In total around 3.57% of test data is missing. After combining both the datasets, imputation was done to fill up the missing values. MICE [6] imputation based on the method of linear regression was used to fill up the missing values. MICE assumes that the missing data are Missing at Random, which means that the probability that a value is missing depends only on observed value and can be predicted using them. By default, linear regression is used to predict continuous missing values.

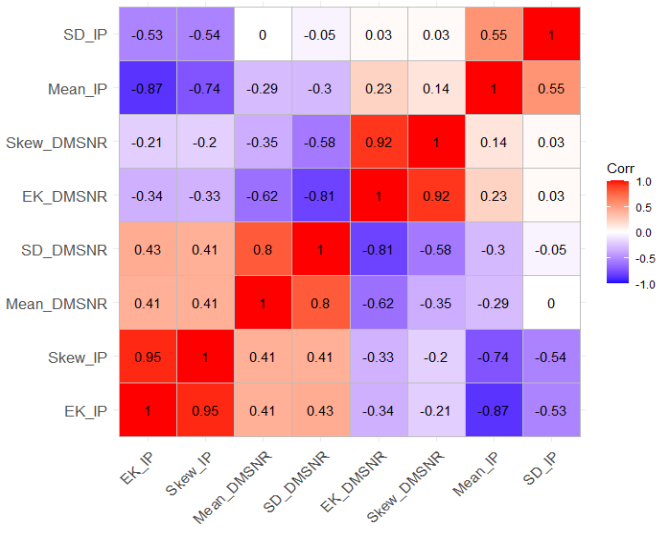

Fig. 3. Correlation Plot

Figure 3 is the correlation plot between the different features present in the dataset. There are lot of strong positive and negative cor relationship between the features. For example, Skewness and excess kurtosis of integrated profile showed a

very strong positive correlation between them. There are lot of other features that show a strong correlation between them, these all features have been visualized.

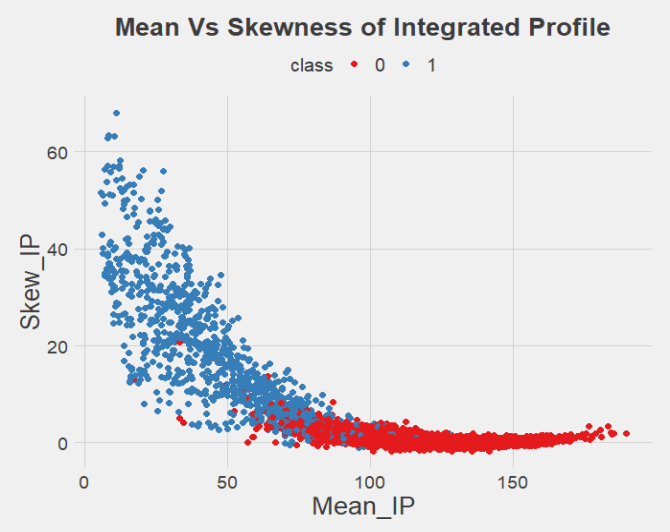

Fig. 4. Mean & Skewness of Integrated Profile

Figure 4, shows the plot of mean vs skewness of integrated profile. A negative cor-relationship can be seen between the features. Most of the neutron stars which are not pulsars have higher values of mean integrated profile and lower skewness value.

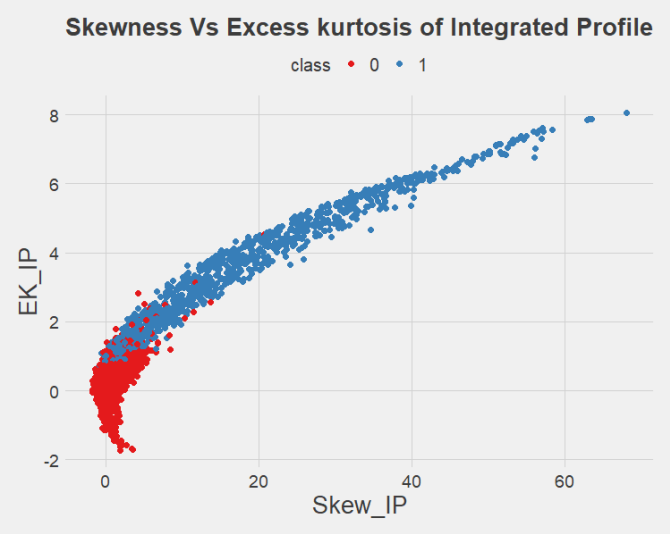

Fig. 5. Excess kurtosis & Skewness of Integrated Profile

Figure 5, shows the plot of skewness vs excess kurtosis of integrated profile neutron star. It is observed that there is a positive correlation between them. Neutron stars which are not pulsar have a lower value of skewness and excess kurtosis.

It is also observed that excess kurtosis and mean of integrated profile have negative correlation between them. Standard deviation and skewness of integrated profile have negative correlation as well.

Figure 6, shows the plot of mean vs standard deviation of integrated profile. There is somewhat a positive correlation between them. It is observed that pulsars have low mean as well as low standard deviation of integrated profile.

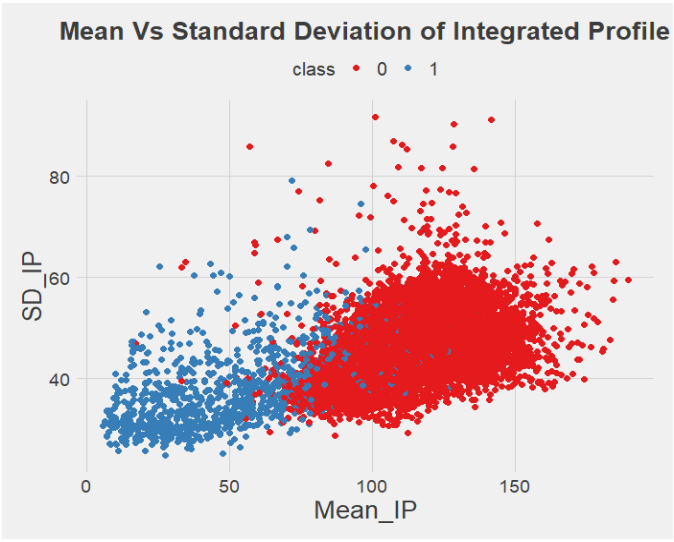

Fig. 6. Mean & Standard deviation of Integrated Profile

Figure 7 shows the plot of mean vs standard deviation of DM-SNR curve. There is a positive correlation between the features. There is no such distribution of pulsars seen in the plot.

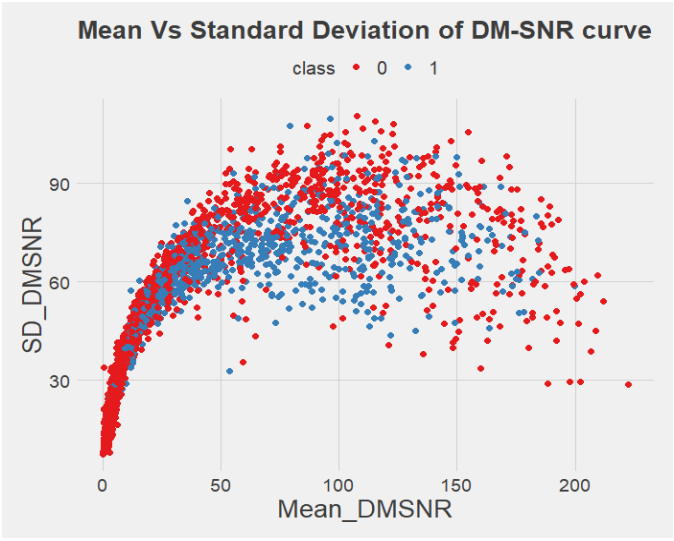

Fig. 7. Mean & Standard deviation of DM-SNR curve

It is also observed that standard deviation and excess kurtosis of DM-SNR have a very strong negative correlation between them.

Figure 8 shows the plot of mean vs excess kurtosis of DM-SNR curve. There is exponential decrease of excess-kurtosis

with increasing values of mean DM-SNR curve. It is observed that neutron stars which are pulsars have low value of excess kurtosis and a general high value of mean.

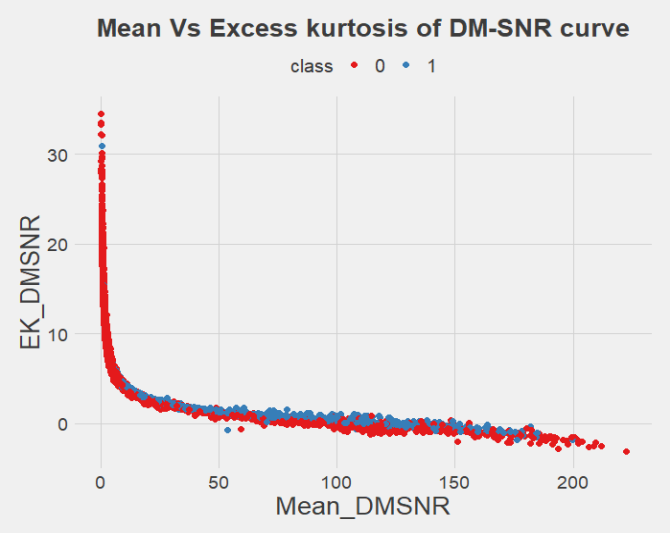

Fig. 8. Mean & Excess kurtosis of DM-SNR curve

Figure 9 shows the plot of skewness vs excess kurtosis of DM-SNR curve. A strong positive correlation is observed between the features. It is observed that pulsars have lower value skewness and excess kurtosis. All these features help in determining there is indeed a relationship between the features and it will really help in deciding the decision boundary during support vector machine application.

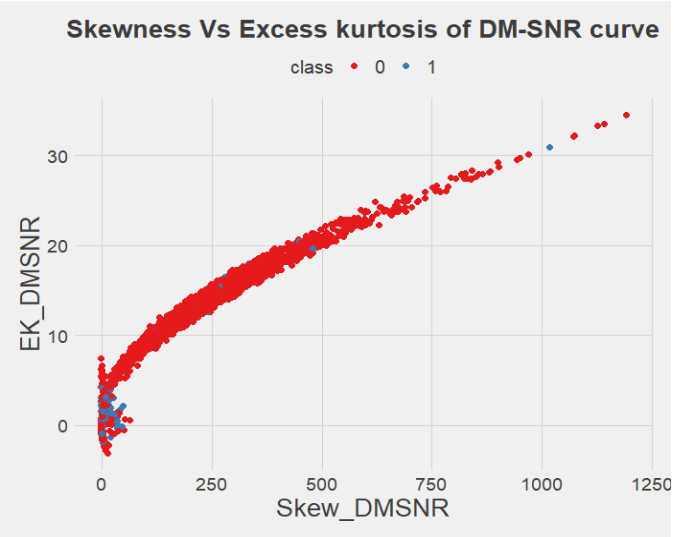

Fig. 9. Skewness & Excess kurtosis of DM-SNR curve

*Model building:*the dataset has been feature scaled first. A linear kernel has been used to build up the model. The training dataset was split into two datasets: one true training data, another a validation dataset.

The true training data has 10023 observations while the validation dataset has 2505 observations. The observations were split randomly in the ratio of 0.8.

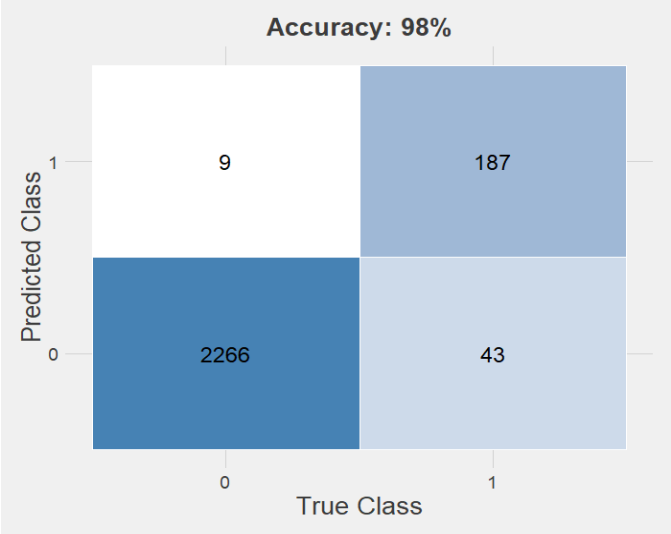

Fig. 10. Confusion Matrix

Figure 10 shows the confusion matrix of validation dataset. From the plot it is observed that a linear kernel SVM model had an accuracy of 98%. Out of 2275 observations of a neutron star not being a pulsar, only 9 observations were predicted to be neutron star. Out of 230 observations actually being a pulsar, 187 observations were correctly predicted. It shows that the linear kernel was a pretty good fit for our dataset and this can be seen due to strong linear relationship between our features.

From the observations, it can be said that all features given in the dataset are pretty important while predicting a neutron star as a pulsar or not.

In the test dataset out of 5370 observations, 403 observations were predicted to be pulsars.

IV. Conclusion

The support vector classifier is a pretty good model for predicting whether a neutron star is a pulsar or not. All the features in the given dataset are pretty important for predicting pulsars. The linear model had an accuracy of 98% on a validation dataset. In total 52 observations were predicted wrong out of 2505 observations.

Further improvements in the model are possible, the outliers were not accounted for, missing data was directly imputed using the MICE algorithm further work on missing data can be done. Instead of only using a linear kernel, different kernels can be used to check which kernel works better. Other classification techniques like logistic regression, random forest can be used as well.

References

[1] https://en.wikipedia.org/wiki/Pulsar
[2] https://data-flair.training/blogs/applications-of-svm/

[3] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.
[4] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[5] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
[6] https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html

Code Documentation

Loading the packages

```
library(tidyverse)
library(mice)
library(ggcorrplot)
library(ggthemes)
library(e1071)
library(caret)
library(scales)
```

Reading the data

```
train<-read_csv("pulsar_data_train.csv")
test<-read_csv("pulsar_data_test.csv")
```

Renaming the columns of the dataset and joining both test and train data

```
colnames(train)<-c("Mean_IP","SD_IP","EK_IP","Skew_IP","Mean_DMSNR","SD_DMSNR",
                   "EK_DMSNR","Skew_DMSNR","class")

colnames(test)<-c("Mean_IP","SD_IP","EK_IP","Skew_IP","Mean_DMSNR","SD_DMSNR",
                  "EK_DMSNR","Skew_DMSNR","class")

full_data<-full_join(train,test)
```

Function to check number of NAs in the data

```
na_check<-function(dataset){
  sapply(dataset,function(x) sum(is.na(x)))
}
```

Basic summary of data

```
str(full_data)
summary(full_data)

na_check(full_data[,-9])

na_check(train)

na_check(test[,-9])
```

Correlation plot

```
correlation<-cor(full_data[,-9],use="na.or.complete")

ggcorrplot(correlation, hc.order = TRUE,lab = TRUE)
```

Imputation and data preprocessing

```
class<-full_data[,9]
full_dat_without_class<-full_data[-9]

imputed_Data <- mice(full_dat_without_class, m=5, maxit = 50, method = 'pmm', seed = 500)
completeData <- complete(imputed_Data,2)

full_data<-completeData
full_data$class<-class$class

full_data$class<-as.factor(full_data$class)
```

Setting up the theme for visualization

```
my_theme<-theme_fivethirtyeight()+theme(plot.title = element_text(hjust = 0.5,size=20),
                                        axis.title = element_text(size=20),
                                        axis.text = element_text(size=14),
                                        plot.subtitle = element_text(hjust=0.5),
                                        legend.position = "top",
                                        legend.title = element_text(size=15),
                                        legend.text = element_text(size=15))
```

Fig 4

```
full_data%>%filter(!is.na(class))%>%ggplot()+
  geom_point(aes(Mean_IP,Skew_IP,col=class),size=2)+
  scale_color_brewer(palette = "Set1")+
  labs(title="Mean Vs Skewness of Integrated Profile")+
  my_theme
```

Fig 5

```
full_data%>%filter(!is.na(class))%>%ggplot()+
  geom_point(aes(Skew_IP,EK_IP,col=class),size=2)+
  scale_color_brewer(palette = "Set1")+
  labs(title="Skewness Vs Excess kurtosis of Integrated Profile")+
  my_theme
```

Fig 6

```
full_data%>%filter(!is.na(class))%>%ggplot()+
  geom_point(aes(Mean_IP,SD_IP,col=class),size=2)+
  scale_color_brewer(palette = "Set1")+
  labs(title="Mean Vs Standard Deviation of Integrated Profile")+
  my_theme
```

Fig 7

```
full_data%>%filter(!is.na(class))%>%ggplot()+
  geom_point(aes(Mean_DMSNR,SD_DMSNR,col=class),size=2)+
  scale_color_brewer(palette = "Set1")+
  labs(title="Mean Vs Standard Deviation of DM-SNR curve")+
  my_theme
```

Fig 8

```
full_data%>%filter(!is.na(class))%>%ggplot()+
  geom_point(aes(Mean_DMSNR,EK_DMSNR,col=class),size=2)+
  scale_color_brewer(palette = "Set1")+
  labs(title="Mean Vs Excess kurtosis of DM-SNR curve")+
  my_theme
```

Fig 9

```
full_data%>%filter(!is.na(class))%>%ggplot()+
  geom_point(aes(Skew_DMSNR,EK_DMSNR,col=class),size=2)+
  scale_color_brewer(palette = "Set1")+
  labs(title="Skewness Vs Excess kurtosis of DM-SNR curve")+
  my_theme
```

Feature Scaling

```
full_data[,1:8]<-scale(full_data[,1:8])
```

Splitting the training, validation, test dataset

```
set.seed(1)

train<-full_data%>%filter(!is.na(class))

Index <- createDataPartition(train$class,p=0.8,list=FALSE)

train_data<-train[Index,]

validation_data<-train[-(Index),]

test<-full_data%>%filter(is.na(class))
```

Model

```
classifier_1<- svm(formula = class ~ .,
                   data = train_data,
                   type = 'C-classification',
                   kernel = 'linear')

val_pred<-predict(classifier_1,newdata = validation_data[,-9])

test_pred<- predict(classifier_1, newdata = test[,-9])

test$class<-test_pred
```

Confusion Matrix and Plot

```r
cm_model1<-confusionMatrix(val_pred,validation_data$class)



cm_dataframe<-as.data.frame(cm_model1$table)

ggplot(data =cm_dataframe ,
       aes(x = Reference, y = Prediction)) +
  geom_tile(aes(fill = log(Freq)), colour = "white") +
  scale_fill_gradient(low = "white", high = "steelblue") +
  geom_text(aes(x = Reference, y = Prediction, label = Freq),size=6) +
  labs(x="True Class",y="Predicted Class")+
  ggtitle(paste("Accuracy:",percent_format()(cm_model1$overall[1])))+
  theme_fivethirtyeight()+
  theme(legend.position = "none",
        axis.title = element_text(size=20),
        axis.text = element_text(size=14),
        plot.title = element_text(hjust=0.5,size=20))
```

Write predicted data csv

```r
write.csv(test,file = "predicted.csv",row.names = FALSE)
```