MLFA MINI-PROJECT 3

SVM Kernel Functions

Name – Bhosale Ratnesh Sambhajirao
Roll No – 19MF10010

Instructor - Prof. Adway Mitra

1.1 Problem Statement

Try out different kinds of SVM Kernel Functions on non-linearly separable datasets. You may create your own datasets that are non-linearly separable, *i.e.*, can be separated by a non-linear structure, and see which all Kernels can he suitable to classify them.

1.2 Brief Introduction to SVM

1.2.1 Hyperplane

A hyperplane is a decision boundary that divides a set of data points with differing class labels into two groups. The SVM classifier uses a hyperplane with the most margin to separate data points. The maximum margin hyperplane and the linear classifier it specifies are known as the maximum margin hyperplane and maximum margin classifier, respectively.

1.2.2 Support Vectors

The sample data points nearest to the hyperplane are called support vectors. By calculating margins, these data points will better define the separation line or hyperplane.

1.2.3 Margin

A margin is the distance between the two lines on the data points that are closest to each other. It is determined as the perpendicular distance between the line and the nearest data points or support vectors. We strive to maximise this separation distance in SVMs to get the most margin.

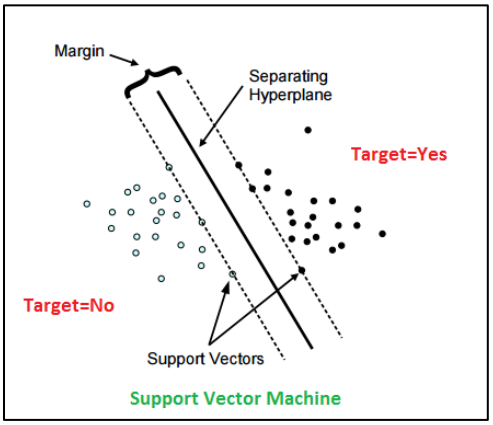

Fig. 1: SVM Margin

The diagram below clearly explains the concepts of maximum margin and maximum margin hyperplane.

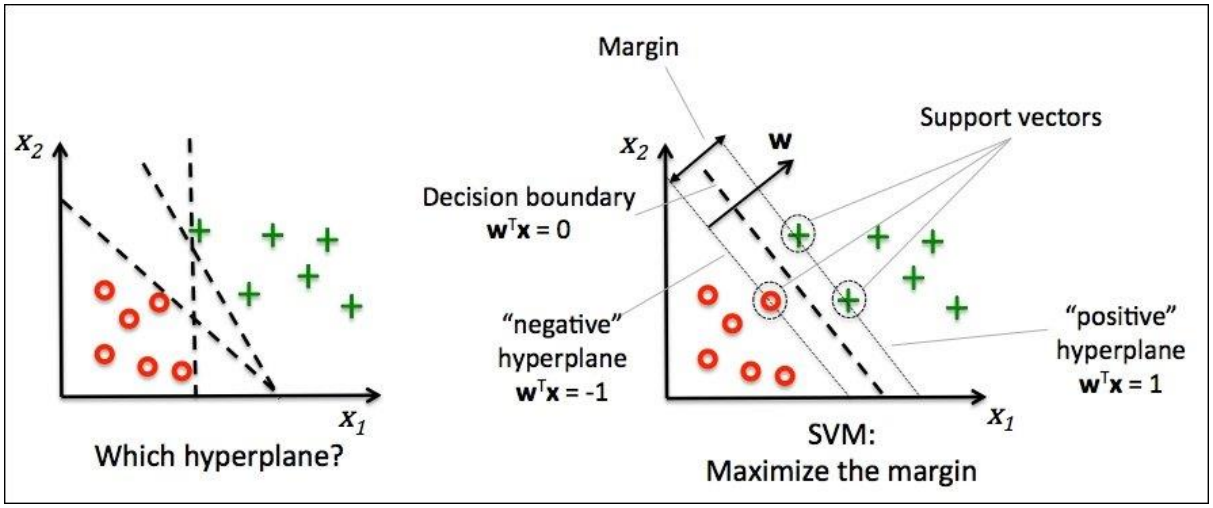

Fig. 2. Maximum Margin Hyperplane

The sample data points can sometimes be so spread that separating them with a linear hyperplane is impossible. SVMs use a kernel trick to shift the input space to a higher dimensional space in this circumstance, as demonstrated in the diagram below. It transforms the 2-D input space into the 3-D input space via a mapping function. Using linear separation, we can now easily separate the data points.

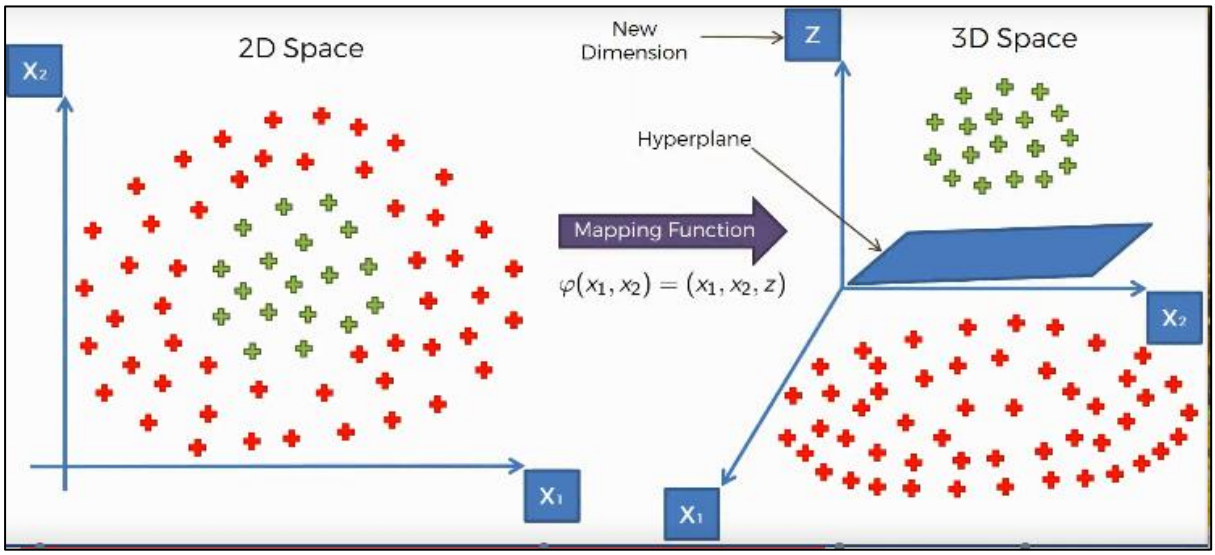

Fig. 3. Visualization of Kernel Trick

1.3 Kernel Trick

In practise, a kernel is used to implement the SVM algorithm. It employs a method known as the kernel trick. A kernel is just a function that translates data to a higher dimension where data can be separated. A kernel is a piece of software that converts a low-dimensional

input data space into a higher-dimensional data space. As a result, by adding more dimensions, it changes non-linear separable issues into linear separable problems. As a result, the kernel trick aids in the development of a more accurate classifier. As a result, it's useful in problems involving non-linear separation.

Kernel Function –

$$K\left(\overline{x}\right) = \begin{matrix} 1 & \text{if} \left\|\overline{x}\right\| \le 1 \\ 0 & \text{otherwise} \end{matrix}$$

In the context of SVMs, there are 4 popular kernels –

- Linear kernel
- Polynomial kernel
- Radial Basis Function (RBF) kernel (also called Gaussian kernel)
- Sigmoid kernel

1.3.1 Linear Kernel

The kernel function in a linear kernel takes the shape of a linear function as follows:

$$k(x, y) = x^T y + c$$

When the data is linearly separable, a linear kernel is utilised. It indicates that data can be split with only one line of code. It is one of the most often utilised kernels. It is most commonly utilised when a dataset contains a significant number of features. The linear kernel is frequently used for text classification.

Because we simply need to improve the C regularisation parameter, training with a linear kernel is usually faster. We must additionally optimise the parameter when training with alternative kernels. As a result, a grid search will normally take longer.

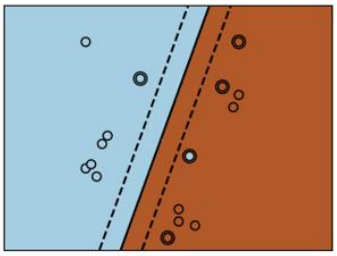

Fig. 4. Linear Kernel Visualization

1.3.2 Polynomial Kernel

The similarity of vectors (training samples) in a feature space over polynomials of the original variables is represented by a polynomial kernel. To estimate their similarity, the polynomial kernel examines not only the supplied attributes of input samples, but also combinations of input samples.

The polynomial kernel for degree-d polynomials is defined as –

$$k(x, y) = (\alpha x^T y + c)^d$$

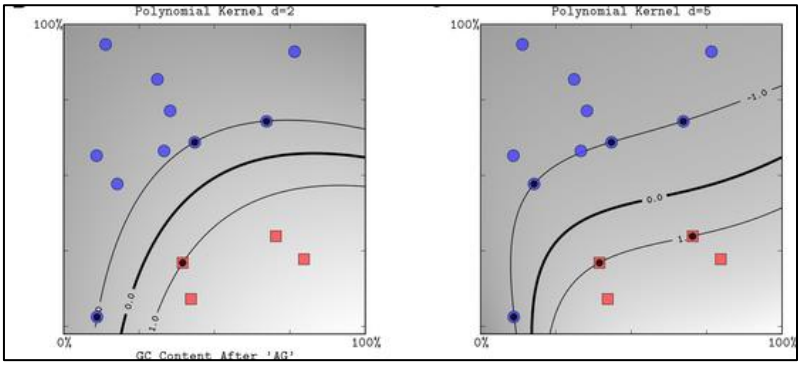

Fig. 5. Polynomial Kernel

1.3.3 Radial Basis Function Kernel (RBF)

Radial basis function kernel is a general-purpose kernel. It is used when we have no prior knowledge about the data. The RBF kernel on two samples x and y is defined by the following equation –

$$k\left(x, y\right) = \exp\left(-\frac{\left\|x - y\right\|^2}{2\sigma^2}\right)$$

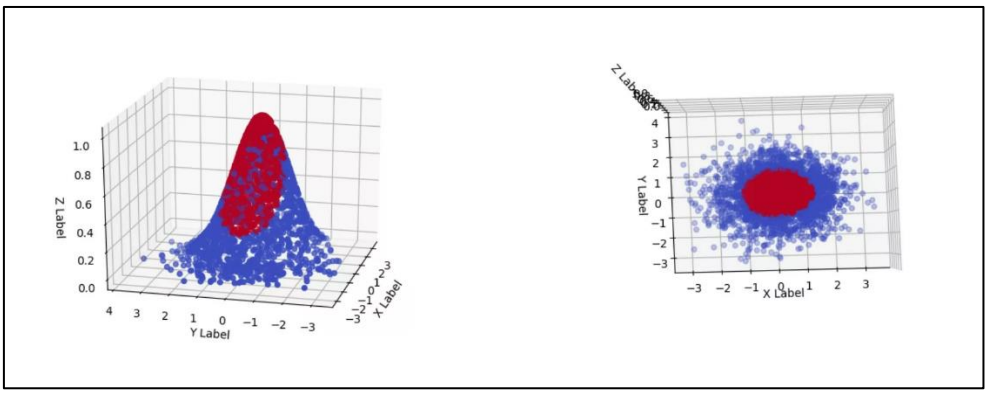

Fig. 6. RBF Kernel Visualization

1.3.4 Sigmoid Kernel

The origins of the sigmoid kernel can be traced back to neural networks. It can be used as a stand-in for neural networks. The following equation gives the sigmoid kernel –

$$k(x, y) = \tanh(\alpha x^T y + c)$$

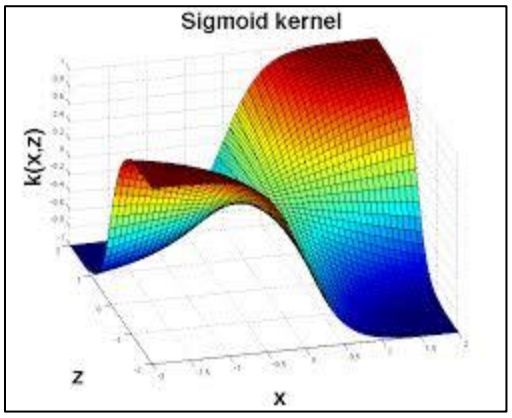

Fig. 7. Sigmoid Kernel

1.4 Dataset Description

Dataset Used - **Predicting a Pulsar Star** dataset

Attributes are summarised below:

1. Mean of the integrated profile.
2. Standard deviation of the integrated profile.
3. Excess kurtosis of the integrated profile.
4. Skewness of the integrated profile.
5. Mean of the DM-SNR curve.
6. Standard deviation of the DM-SNR curve.
7. Excess kurtosis of the DM-SNR curve.
8. Skewness of the DM-SNR curve.
9. Class

There are 12528 instances and 9 variables in the data set. 8 are continuous variables and 1 is discrete variable. The discrete variable is target_class variable. It is also the target variable.

1.4.1 Distribution of target_class

11375 observations belong to target_class 0, while 1153 observations belong to target_class 0.

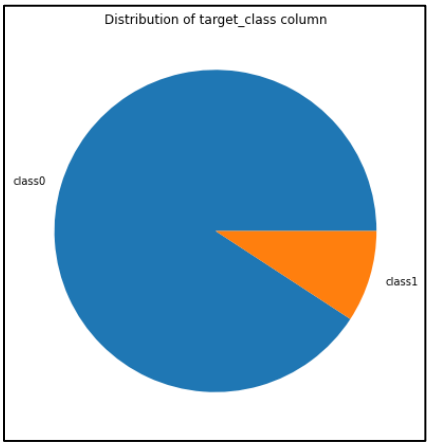

Fig. 8. Distribution of target_class

1.4.2 Outliers

To visualise outliers in the aforementioned variables, boxplots were used.

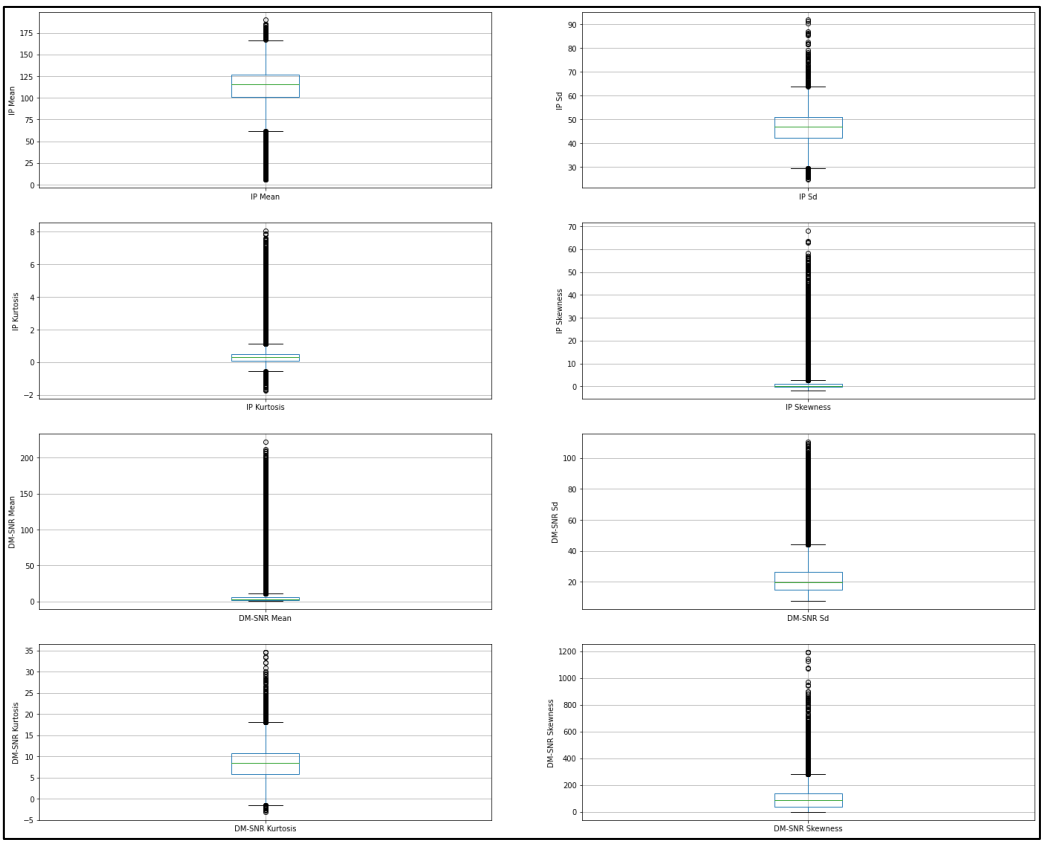

Fig. 9. Box plots

The boxplots above show that these variables have a lot of outliers. Because the dataset contains outliers, C should be set to a high value while training the model.

1.4.3 Distribution of variables

Check distributions with histograms to see if they are normal or skewed.

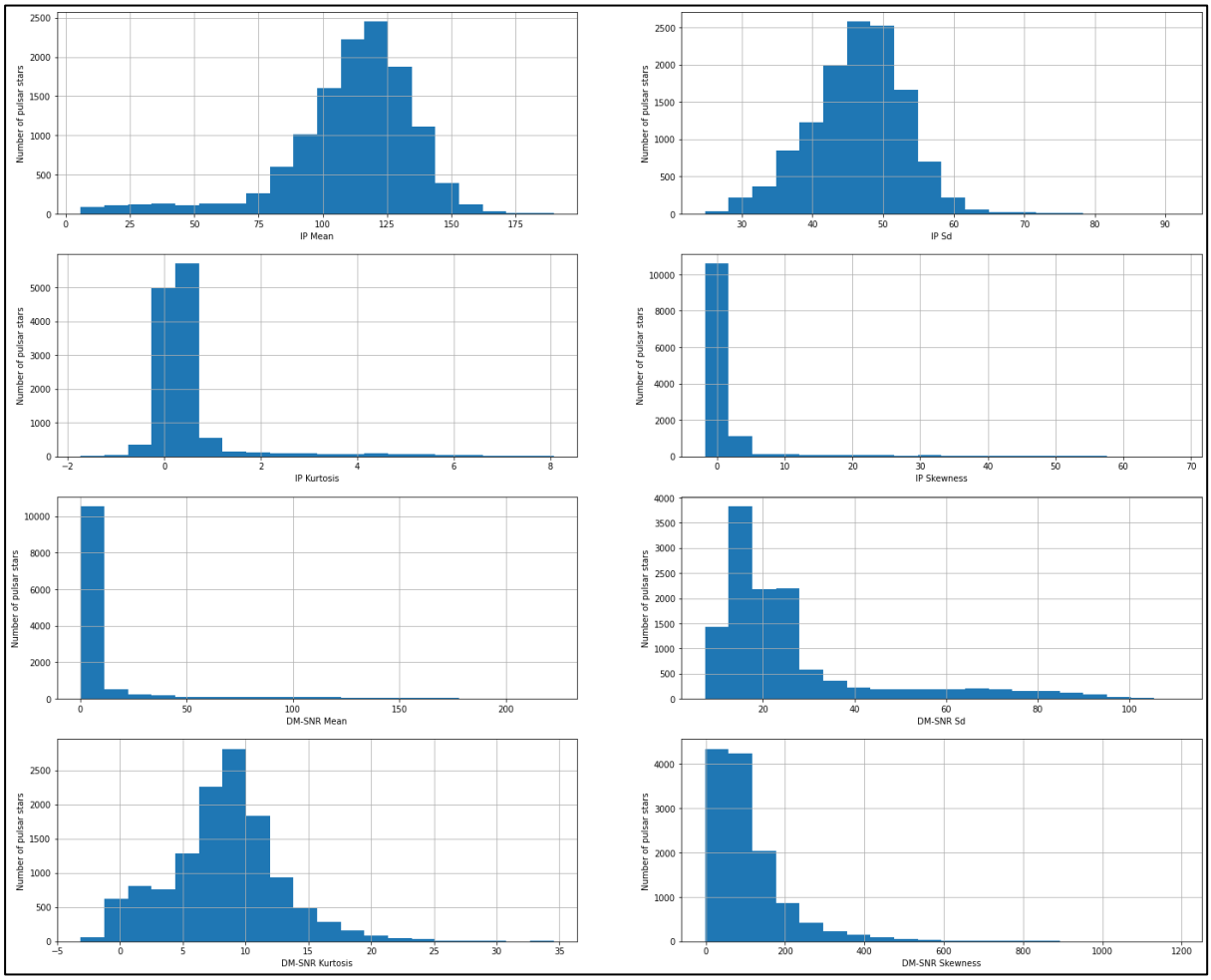

Fig. 10. Distribution of variables

We can see that all the 8 continuous variables are skewed.

1.5 Model Training

Model is trained with different kernels *e.g.*, linear kernel, RBF kernel, polynomial kernel and sigmoid kernel.

1.5.1 Confusion Matrix

A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give us a clear picture of classification model performance

and the types of errors produced by the model. It gives us a summary of correct and incorrect predictions broken down by each category. The summary is represented in a tabular form.

1.5.2 ROC AUC

ROC AUC stands for Receiver Operating Characteristic - Area Under Curve. It is a technique to compare classifier performance. In this technique, we measure the area under the curve (AUC). A perfect classifier will have a ROC AUC equal to 1, whereas a purely random classifier will have a ROC AUC equal to 0.5. So, ROC AUC is the percentage of the ROC plot that is underneath the curve.

1.5.3 Comparisons of kernels

Result after training is shown in the table –

Table 1. Comparison of kernels

Kernel	Accuracy	F1-score (weighted avg)	ROC AUC
rbf	0.9796	0.98	0.9015
linear	0.9777	0.98	0.8905
polynomial	0.9757	0.97	0.8736
sigmoid	0.8747	0.87	0.6158

Source: Notebook

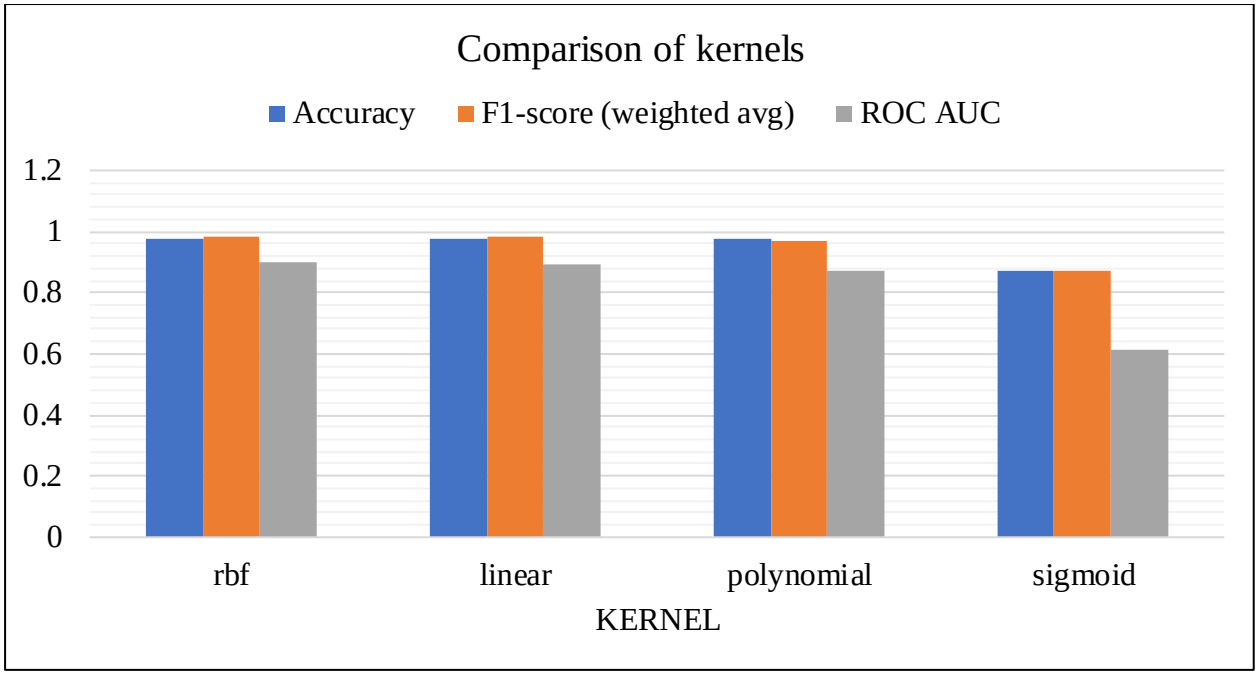

Fig. 11. Comparison of kernels

1.5.4 Comments

- Maximum accuracy achieved with rbf kernel with C=10.0. and the accuracy is 0.9796. Based on the above analysis we can conclude that our classification model accuracy is very good. Our model is doing a very good job in terms of predicting the class labels.
- ROC AUC is a single number summary of classifier performance. The higher the value, the better the classifier. ROC AUC of our model approaches towards 1. So, we can conclude that our classifier does a good job in classifying the pulsar star.
- 'rbf' kernel achieved highest ROC AUC score (0.9015) and weighted F1-score (0.98), so rbf is better kernel for this dataset.

1.6 Results and Conclusions

- There are outliers in our dataset. So, as kernel changed accuracy changed.
- We get maximum accuracy with rbf and linear kernel with C=10.0 and the accuracy is 0.9796. So, we can conclude that our model is doing a very good job in terms of predicting the class labels. Here, we have an imbalanced dataset. Accuracy is an inadequate measure for quantifying predictive performance in the imbalanced dataset problem. So, we explored confusion matrix that provide better guidance in selecting models.