Analysis of SVM

Soumyae Tyagi

Reuben Varghese Joseph

Anshuman Mohanty

Abstract

Support Vector Machine (SVMs) is one of the most powerful algorithms for problem solving in machine learning and is known to have deep roots in the fields of Statistical Learning Theory (SLT) and optimization problems along with numerous industrial applications. SVMs are known to reduce most of the problems in machine learning to optimization problems and the latter lies at the heart of SVMs. Due to its higher accuracy, support vector machine is a widely researched topic in the machine learning community. The algorithm promises a higher empirical performance. This project illustrates a detailed understanding of Support Vector Machines along with different parameters associated along with it. Besides, the project presents a comparative analysis of different SVM kernels on the 'Gender Recognition using voice' dataset. The accuracy scores were listed were displayed before and after hyperparameter tuning along with the overall percentage increase in the scores after performing tuning.

Keywords: Support Vector Machine (SVM), Classification, Hyperparameters, Optimization Algorithm, Kernel Functions

Introduction

SVM use supervised algorithms using support vector classifier as a base. The term support vector classifier is derived from the fact that all the observations on the edge and within the soft margins are called support vectors. It is one of the most important algorithms which are applicable for both classification as well as regression datasets. However, normally it is used for classification problems. The algorithm was first introduced in 1960 but it was Vapnik and his coworkers by the early 90s. This algorithm is extremely effective as it could handle numerous categorical and continuous variables.

Apart from machine learning, SVMs are also known to be effective in the domain of data mining. Numerous studies highlight the fact that SVMs are known to have significantly higher amount of accuracy as compared to other existing traditional classification algorithms. It is largely because they proceed with problem solving using finite training points and thus do not face the problems like overfitting, curse of dimensionality etc. It maps the training data in space to increase the separation between the categories. The newer data points would be mapped in the same space and would be predicted and placed in a category.

SVM supports both linear and non-linear classification. The latter is carried out by using kernel trick which simply means mapping the inputs in non-linear data into a higher-dimensional feature space.

The success of SVMs is mainly attributed to three fundamental reasons: Kernel trick, principle of maximal margin and dual theory. However, for some of the datasets, the values of cost and kernel parameters play a significant role in the accuracy of the model. Therefore, the user needs to continually perform considerable cross validation to obtain optimum parameter settings which in turn would result a higher accuracy score.

Kernels are referred to as a core of SVM since they support a set of mathematical functions which are used for manipulation of data.

In terms of real-time applications, SVMs are used in medical decision support, face authentication, image recognition, text categorization etc. SVMs are also applicable for all kinds of datasets be it, audio, video or text.

In the next section, we have made an effort to present a thorough analysis of Support Vector Machine ranging from the detailed explanation of core fundamentals of SVMs to implementation and analysis of SVM on different kernel functions using ‘Gender recognition by voice’ dataset.

Background Study

Durgesh K. Srivastava and Lekha Bhambhu [1] applied and analysed the comparative results of various kernel functions of SVM on four different applications data set such as diabetes data, heart data and satellite data which all have different features, classes, number of training data and different number of testing data. It was inferred that the choice of kernel function and best value of parameters for a particular kernel is critical for a given amount of data. Upon thorough analysis of the sample datasets, they deduced that the best kernel is RBF for infinite data and multi class.

In 2012, Himani Bhavsar and Mahesh H. Panchal [2] presented a review highlighting the advantages of SVM over existing data analysis techniques. This paper discusses about the various linear and non-linear functions and suggests that SVM uses statistical learning theory to search for a regularized hypothesis that fits the available data well without overfitting. All the kernel functions were compared and analysed on the basis of its respective parameters such as γ, r, c and d. It also brings forward disadvantages of SVM such as choice of the kernel as well as the speed and size required while training large datasets.

R.Saji Priya et al [3] provide insights on linear and non-linear classification and discusses about making linear models work in non-linear settings. It also presents an overview of Kernel selection chiefly describing about the RBF kernel function; and Model Selection in SVM for both linearly and non-linearly separable data by cataloguing models on the basis of regression and classification. It was observed that Non-linear classifiers produce only slightly better classification results whereas linear classifiers produce better results than non-linear classifier models.

Gidudu Anthony et al [4] compared and analysed two commonly used approaches namely the One-Against-One (1A1) and One-Against-All (1AA) techniques by evaluating them in accordance of their impact and implication for land cover mapping. It explores the above-mentioned approaches with a view of discussing their implications for the classification of remotely sensed images and establish their performance on the extraction of land cover information from satellite images. The prime inference from this paper suggested that whereas the 1AA technique is more predisposed to yielding unclassified and mixed pixels, the resulting classification accuracy is not significantly different from 1A1 approach; Thus, giving us the choice of approach to adopt on the basis of our preference and comfort.

Ajaj Khan et al [5] proposed a model to classify MRI images, differentiating between normal patients and a patient who has a tumor in his/her brain. The suggested method comprises of two stages: firstly, feature extraction where textural features are extracted from MRIs using GLCM followed by classification of the tumor into cancerous or benign category. The measurements obtained from the study of textural features are given as input to the SVM classifier for training in order to classify it. It was successfully concluded that the proposed approach proved to work efficiently by effectively detecting the tumor of human brain through the analysis of MRI images and produced accurate results as compared to other classifiers.

Damodara Krishna Kishore Galla et al [6] proposed a method to better the vision sensation of blind people by converting visualized data to audio data, thus helping in gender classification based on the basis of face recognition approach. The suggested model was processed with face extraction comprising of multi scale-invariant feature transform (MSIFT) which are converted to image frames and subsequently normalized by using multi-variant normalization. Feature optimization was applied using support vector machines followed by classification using LASSO classifier. After testing the scheme on 5 different databases, it was deduced that the proposed approach resulted in better accuracy with less time of execution.

In 2016, Kamil Aida-zade et al [7] applied support vector machines to acoustic model of speech recognition system based on Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC) features for Azerbaijani Dataset. The variety of results of SVM with different kernel functions is analysed in the training process. Lagrange Interpolation method was used to lead every utterance of speech signal having different lengths to the same scale. On close analysis, it was observed that that SVM with radial bases kernel on LPC features gives better result, whereas SVM with polynomial kernel on MFCC features indicate better accuracy. The experiments' results concluded that the overall performance of SVM based speech recognition is better than ANN on the same dataset with a marginal difference.

In 2020, J.P.Medlin Julia et al [8] proposed a paper to optimize the parameters and feature weighting in order to improve the strength of the SVM simultaneously. It puts forward the imperialist Competitive Algorithm based Support Vector Machine (ICA-SVM) classifier which is used in efficient weed detection and classification by selecting the appropriate input features and optimizing the parameters. The framework consists of pre-processing which uses histogram and Median filtering for improving the contrast and filtering unnecessary noise, colour segmentation using canny edge detector, Feature Extraction which makes use of Gabor Wavelet, and Imperialist competitive algorithm (ICA) to solve continuous-optimization problems effectively that improves the performance of SVM classifier by selecting an appropriate parameter. The model achieved high accuracy rate in classifying the weeds demonstrating greater effectiveness over existing weed detection systems.

Support Vector Machine

In a multidimensional space, SVMs depict different categories of hyperplane. The generation of hyperplane happens iteratively which contributes to the reduction of error. The aim of SVM is to select the separating hyperplane that is the most optimal and tends to increase the margin of training data.

The representation of SVM involves two parallel hyperplanes that is required to separate the data. The separating hyperplane tries to maximize the distance between two parallel hyperplanes.

Linearly and Non-Linearly separable data

Data can be considered as linearly separable when:

- In 1D, we can separate the data by a point.
- In 2D, we can separate the data by a line.
- In 3D, we can separate the data by a plane.

When we cannot find a point, line, or plane to separate the data points, it is considering to be non-linearly separable data.

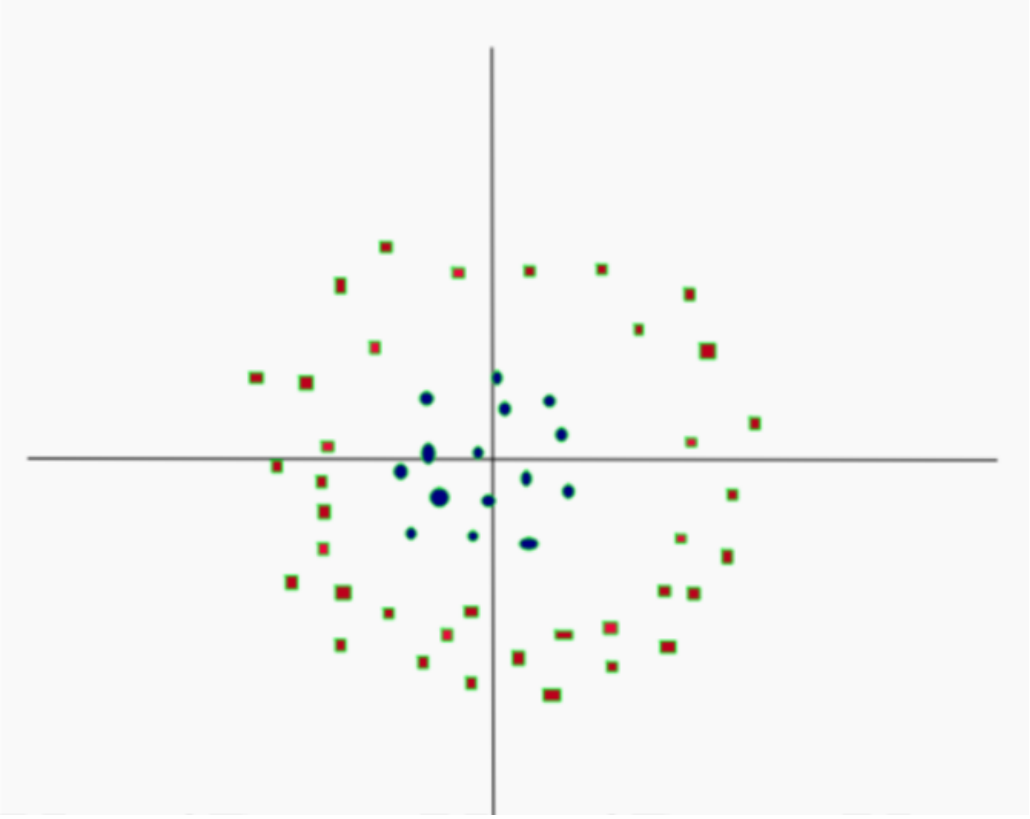


Figure 1: Non-Linearly Separable Data

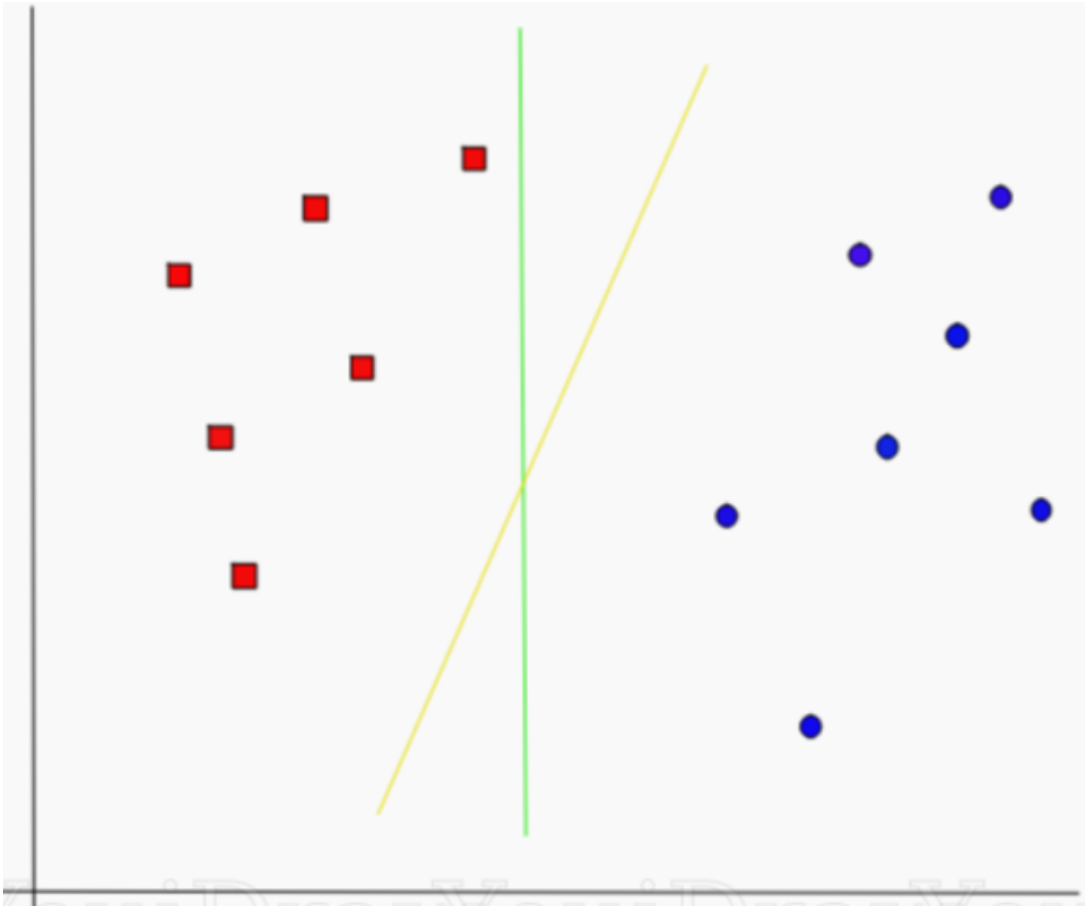


Figure 2: Linearly Separable Data

Differences between perceptron and SVM

Perceptron stops after it classifies all the data correctly whereas SVMs stop after finding the most optimum plane that possesses the best margin. There could be different hyperplane that a perceptron could generate which in turn depends upon initial weights which is not the case in SVM.

The following terms are the core concept of SVMs.

Support Vectors:

The data points which are nearest to the hyperplane that affects the position of the hyperplane is termed as support vector.

Margin:

The space between two lines in the nearest data points from two different classes. It is calculated as perpendicular distance from the line to the closest points. In SVM, it is one of the important criterions that is used for looking for a decision surface which is maximally far away from any given data point.

Hyperplane:

It is a plane/space which divides a set of objects as different classes. A SVM performs classification by finding a hyperplane that tends to maximize the margin between two classes. The vectors that define the hyperplane are called as support vectors. In an ideal case, SVM produces a hyperplane which completely separates the vectors into two non-overlapping classes. However, this is not always the case as data is rarely linearly separable. Generally, there are situations where a non-linear region could efficiently segregate the data into different groups.

Let's consider a case of two-dimensional linearly separable data. The data can be separated by a line. The function of the line y=ax+b can be modified by renaming x with x1 and y with x2. We get:-

$$ax_1 - x_2 + b = 0 \tag{1}$$

If we define x=(x1,x2) and w=(a,-1), we get:

$$w \cdot x + b = 0 \tag{2}$$

which is the equation of hyperplane. Here, w refers to the normal to a plane and b refers to the bias. The equation is derived from two-dimensional vectors but it works for any given number of dimensions.

Once we have the hyperplane, we can then use the hyperplane to make predictions. We define the hypothesis function h as:

$$h(x_i) = \begin{cases} +1 & if\ w \cdot x + b \geq 0 \\ -1 & if\ w \cdot x + b < 0 \end{cases} \tag{3}$$

The point above or on the hyperplane will be classified as class +1, and the point below the hyperplane will be classified as class -1.

So basically, the goal of the SVM learning algorithm is to find a hyperplane which could separate the data accurately. There might be many such hyperplanes. And we need to find the best one, which is often referred as the optimal hyperplane.

SVM optimization problem

One of the major goals of SVM is to find the most optimal hyperplane which could separate the data.

Three metrics have been presented to compare the hyperplanes and choose the most optimal one.

First Version:

Let's first consider the equation of the hyperplane $w \cdot x+b=0$. We know that if the point (**x**,y) is on the hyperplane, $w \cdot x+b=0$. If the point (**x**,y) is not on the hyperplane, the value of $w \cdot x+b$ could be positive or negative.

For all the training example points, we want to know the point which is closest to the hyperplane. We could calculate $\beta=|w \cdot x+b|$. To define the problem formally: -

Given a dataset,

$$D = \{(x_i, y_i) | x_i \in R^n, y_i \in \{-1, 1\}\}_{i=1}^{m} \quad (4)$$

where M is the number of training samples and n is the number of features in our dataset. We need to compute β for each training example, and B is the smallest β we get.

$$B = \min_{i=1...m} |w \cdot x + b| \quad (5)$$

If there are s hyperplanes, each of them would have a B_i value, and we would select the hyperplane with the maximum B_i value.

$$H = \max_{i=1...s} \{h_i | B_i\} \quad (6)$$

One of the major limitations of this metric is that it cannot distinguish between a good hyperplane and a bad one. Since, we are taking the absolute value of $w \cdot x+b$, we could get the same value for an incorrect hyperplane.

Second Version

Here, we use the information of the label y, We define the equation f=y(w·x+b) where the sign of f will always be positive if the point is correctly classified and will be negative if incorrectly classified. To define the problem formally:-

We are given a dataset D for which f is computed for each training data point, and F is the smallest value of f we get. In theory, F is called the functional margin of the dataset.

$$F = \min_{i=1...m} y_i(w \cdot x + b) \tag{7}$$

When comparing different hyperplanes, the hyperplane with the largest value of F will be selected.

But however, even this metric suffers from a limitation, let's consider an example: -

we have two vectors w1=(3,4) and w2=(30,40). Since they have the same unit vector u=(0.6,0.8), the two vectors w1 and w2 represent the same hyperplane. However, when we compute F, the one with w2 will return a larger number than the one with w1.

Hence, there needs to be a metric which needs to be scale invariant.

Third Version

We divide f by the length of the vector w. We define: -

$$\gamma = y(\frac{w}{\|w\|} \cdot x + \frac{b}{\|w\|}). \tag{8}$$

To define the problem formally: -

We are given a dataset D for which γ is computed for training example, and M is the smallest γ. In theory, M is referred to as the geometric margin of the dataset.

$$M = \min_{i=1...m} y_i(\frac{w}{||w||} \cdot x + \frac{b}{||w||}) \tag{9}$$

While comparing the hyperplane, the largest M will be favorably selected.

We now have a perfect metric for comparing different hyperplanes.

Now, the main objective is to find the optimal hyperplane which, in turn means that we need to find the values of w and b.

The problem of finding the values of w and b is called an optimization problem.

Deriving SVM optimization problem

We solve the optimization problem with the constraint that geometric margin of each training sample should be greater than or equal to M:

$$\max_{w,b} M$$
$$subject\ to\ \gamma_i \geq M, i = 1...m \tag{10}$$

$$M = \frac{F}{\|w\|}, \tag{11}$$

As,

the problem could be modified to as:-

$$\max_{w,b} M$$
$$subject\ to\ f_i \geq F, i = 1...m \tag{12}$$

While the values of w and b are still being rescaled, we are still maximizing M and the optimization result will not change. Let's rescale **w** and b and make **F=1**

$$\max_{w,b} \frac{1}{||w||}$$
$$subject\ to\ f_i \geq 1, i = 1...m \tag{13}$$

The maximization problem is equivalent to the following minimization problem:

$$\min_{w,b} ||w||$$
$$subject\ to\ f_i \geq 1, i = 1...m \tag{14}$$

This minimization problem is equivalent to the following minimization problem:

$$\min_{w,b} \frac{1}{2}||w||^2$$
$$subject\ to\ y_i(w \cdot x + b) - 1 \geq 0, i = 1...m \tag{15}$$

The statement is called as SVM optimization problem.

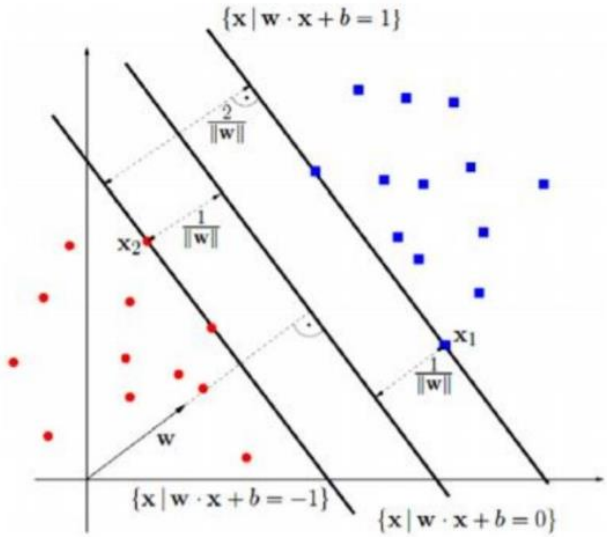


Figure 3: Support Vector Machine

Overfitting

It refers to a modelling error which occurs when a function is too closely fit to a limited set of data points.

Underfitting

It refers to a situation where a data model is not able to capture the relationship between the input and output variables accurately thereby, generating a higher error rate.

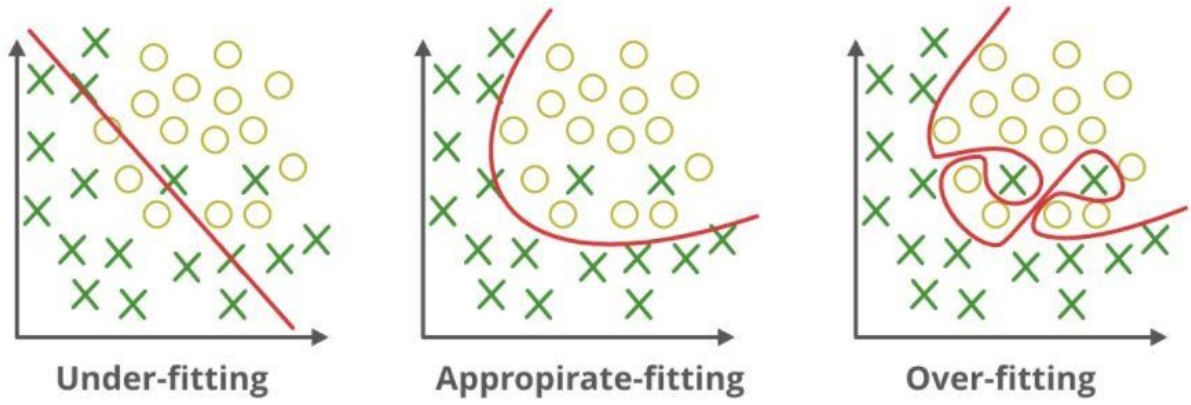


Figure 4: a) Underfitting of data points b) Appropriate fitting of data points c) Over Fitting of data points

Misclassifications:

Misclassified points are those data points which are classified into the wrong category. SVM aims to minimize the rate of misclassification errors.

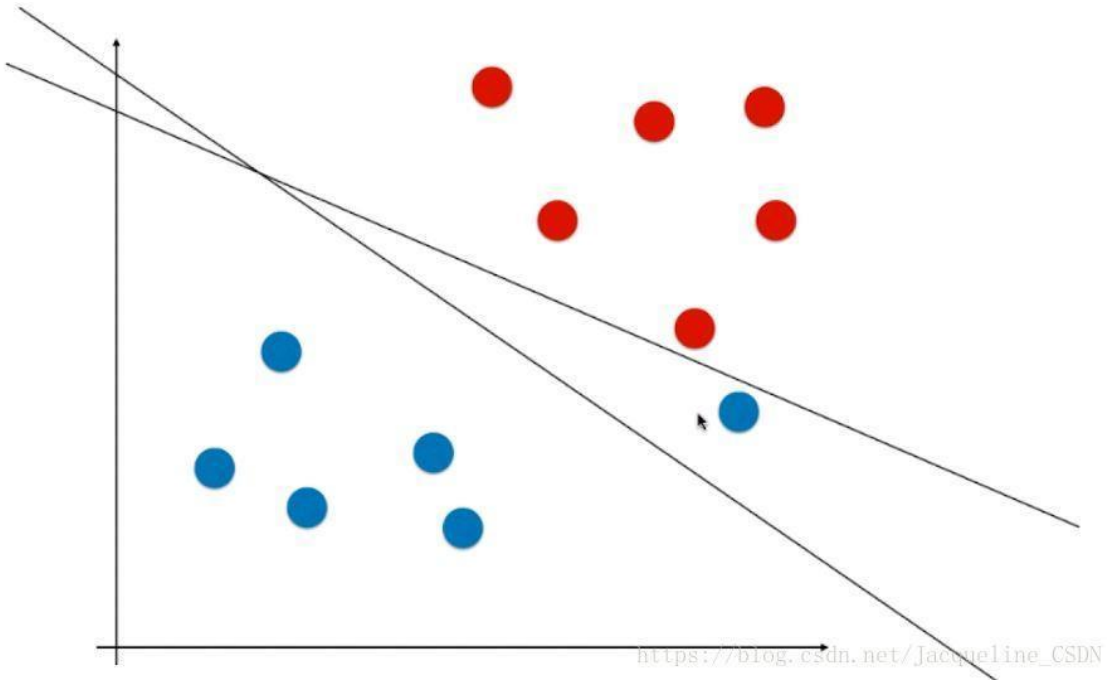


Figure 5: Missclassification of Data points

Hard Margin:

In hard margin, there is an effort to not have errors for all given data points as they have strict conditions to classify every data point. If the data points are linearly separable, then the hyperplane would be differentiating between the classes but if there is any misclassification, then it is not possible to separate them.

Solving SVM optimization prblem

The problem could be solved by lagrange multipliers.

$$\nabla f(x) - \alpha \nabla g(x) = 0 \tag{16}$$

Where alpha is called Lagrange Multiplier.

In terms of the SVM optimization problem, $f(w) = \frac{1}{2}\|w\|^2$, $g(w,b) = y_i(w \cdot x + b) - 1, i = 1...m$. The Lagrangian function is then $\mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(w \cdot x + b) - 1]$. (17)

We differentiate L w.r.t. w and b

$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0$$

$$\nabla_b \mathcal{L}(w,b,\alpha) = -\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{18}$$

On substituting, we get

$$W(\alpha, b) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{19}$$

The dual problem is stated as:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

$$subject\ to\ \alpha_i \geq 0, i = 1...m, \sum_{i=1}^{m} a_i y_i = 0 \tag{20}$$

By extending Lagrange multipliers to KKT (Karush-Kuhn-Tucker) conditions, we get the KKT condition that states:

$$\alpha_i [y_i (w \cdot x^\star + b) - 1] = 0 \tag{21}$$

Here, x* are the points when we reach the optimal. As alpha is positive the remaining part of equation

$$y_i (w \cdot x^\star + b) - 1 \tag{22}$$

must be zero.

Computation of w and b

From the above equation, the value of w:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{23}$$

To get value of b we use,

$$y_i (w \cdot x^\star + b) - 1 = 0 \tag{24}$$

Multiply by y_i^2, as we know $y_i^2=1$, we get:

$$b = y_i - w \cdot x^\star \tag{25}$$

The formula could be modified as:

$$b = \frac{1}{S} \sum_{i=1}^{S} (y_i - w \cdot x) \tag{26}$$

where S is the number of support vectors

Soft Margin:

In soft margin, there is an effort to maximize the margin between the support vectors. Here, we allow our model to have some relaxation to few points. If we choose to consider these points, then our margin would significantly reduce, and our decision boundary becomes poorer. In short, we allow misclassifications on our data.

Soft margin leads to underfitting whereas hard margin leads to overfitting.

Solving SVM optimization problem

The soft-margin enables addition of slack variables ζi to the constraints of optimization.

The constraints become:

$$y_i(w \cdot x_i + b) \geq 1 - \zeta_i, i = 1...m \tag{27}$$

By adding the slack variables, when minimizing the objective function, it is possible to satisfy the constraint even if the example does not meet the original constraint. The problem is we can always choose a large enough value of ζ so that all the examples will satisfy the constraints.

One technique to handle this is to use regularization. For example, we could use L1 regularization to penalize large values of ζ. The regularized optimization problem becomes:

$$\min_{w,b,\zeta} \frac{1}{2}||w||^2 + \sum_{i=1}^{m} \zeta_i$$
$$subject\ to\ y_i(w \cdot x_i + b) \geq 1 - \zeta_i, i = 1...m \tag{28}$$

We have to make sure that we do not minimize the objective function by choosing -ve values of ζ. We also add a regularization parameter C to determine how important ζ should be, which means how much we want to avoid misclassifying each training data points.

The problem becomes:

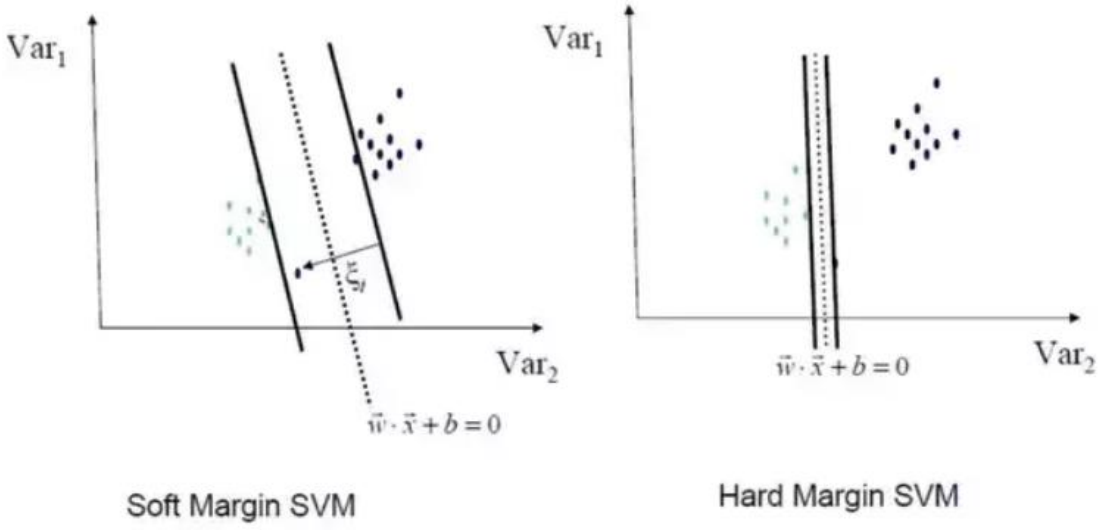
$$\min_{w,b,\zeta} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \zeta_i$$
$$subject\ to\ y_i(w \cdot x_i + b) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1...m \tag{29}$$

The optimization problem is changed to the dual problem as: -

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

$$subject\ to\ 0 \leq \alpha_i \leq C, i = 1...m, \sum_{i=1}^{m} a_i y_i = 0 \quad (30)$$

Here the constraint αi≥0 has been changed to 0≤αi≤C.



Figur 6: Hard Margin and Soft Margin

Maximal Margin Classifier:

It is the most suitable (or hypothetical) hyperplane that is defined such that two classes are linearly separable. In other words, it is the hyperplane which has the maximum margin i.e. it has largest distance between the hyperplane and training observations. It can be a point, a line or a plane depending on the data points.

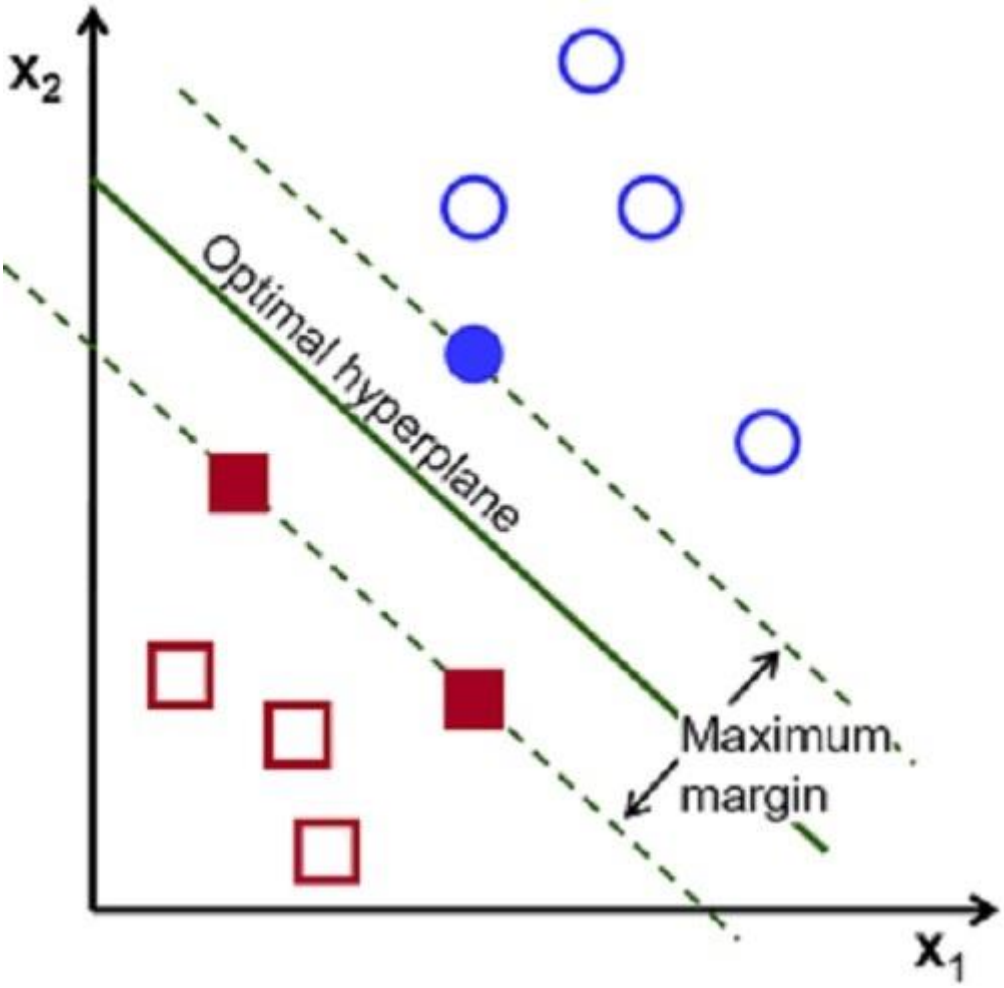


Figure 7: Maximal Margin Classifier

Bias Variance tradeoff:

Bias refers to the assumptions for simplification made by a model to make the target function easier to learn. Lower Bias indicates that less assumptions are made about the form of target function where high bias indicates more assumptions about the form of target function. SVMs fall under the category of low-bias machine learning algorithm. Variance refers to the amount that the estimate of the target function would change when the training data is altered.

Estimation of target function is performed from the training data, so the algorithm will have some variance. In an ideal case, if the algorithm does not change much even after changing from one dataset to another, it signifies that the algorithm is flexible and picks out the hidden underlying mapping between the inputs and their corresponding outputs. Lower variance means that minor modifications are suggested to estimate the target function with changes to training dataset whereas Higher variance means that large modifications are suggested to estimate the target function with changes to dataset. SVMs are the class of machine learning algorithms which fall under the high variance category.

Ideally, one of the major goals in any supervised algorithm is to achieve a lower bias and low variance.

Thus, the relation between bias and variance is as follows:

- Increase in bias would result in decrease in variance.
- Decrease in bias would result in increase in variance.

SVM is considered as an algorithm which is said to have low bias and high variance, but the trade-off can be modified by changing some parameters in the algorithm.

However, in real life situations, it is difficult to compute the real bias and variance error as we do not know the actual underlying target function. But they play a major role in understanding the behavior of algorithms for increasing the accuracy.

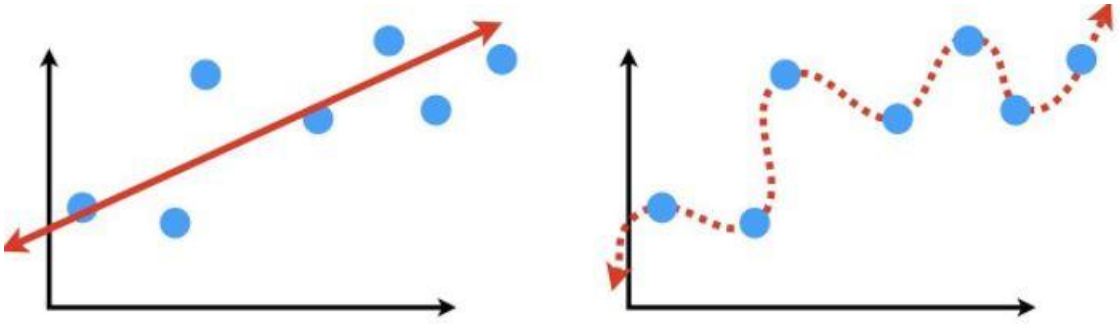


Figure 8: a) Model with high bias and low variance b) Model with low bias and high variance

Hyperparameter Tuning

A machine learning model consists of several parameters which need to be learned from the data. Hyperparameters are those parameters of a model which cannot be directly learned. The parameters need to be chosen based on intuition or hit and trial. These parameters play a crucial role in improving the performance of the model such as complexity, learning rate etc.

The hyperparameters of SVM include C and gamma and finding the optimal hyper-parameter would lead us to giving higher accuracy of the model. One of the most popular methods of Hyperparameter tuning is done using Grid Search. It takes a dictionary which describes the parameters which could be tried on a model to train it. The grid is defined as a dictionary in which the keys are the parameters, and the values are the settings that are to be tested.

Parameter C

C parameter in SVM is also called as regularization parameter. It defines the extent to which SVM works on the errors. In other words, it refers to the extent to which we weight the slack variables in SVM classifier. It basically refers to the tradeoff between misclassification of training examples against simplicity of decision surface.

When the C value is large, slacks penalize the objective function of SVM. As C approaches infinity, any slack variable having non-zero value would be having infinite penalty and thus all the slack variables would be set to zero and we will end with a hard-margin classifier. As hard margins are more sensitive to outliers, they are more likely to face the problem of overfitting. Thereby due to the nature of training data, it would not be able to find a margin at all. In this case, the accuracy of the model is quite high as compared to soft margin but the accuracy score will be minimized on testing.

Minimizing the C value leads to shift in margin that could be much closer to certain datapoints of a particular category. When the value of C approaches 0, the slack variables for all data points are set free and could be as large as possible and thus would end up with soft margins which would lead to underfitting of data.

Therefore, it can be understood that smaller C would give a wider margin, but at the cost of some misclassifications whereas a larger C would give a hard margin which is strict and tolerates zero constraint violation. Therefore, it is key to find an optimum value of C such that noisy data does not have too much of an impact on the solution. However, soft-margin SVM formulation is a nice improvement over the hard-margin classifier which allows us to classify data correctly even when there would be noisy data that would break linear separability. A few misclassifications on training data are not a bad thing if the model can generalize well in the end. In this case, the accuracy of the model is lower than in hard margin but the accuracy score will be maintained even during testing data.

To find the best value of C, we must use grid search or cross validation.

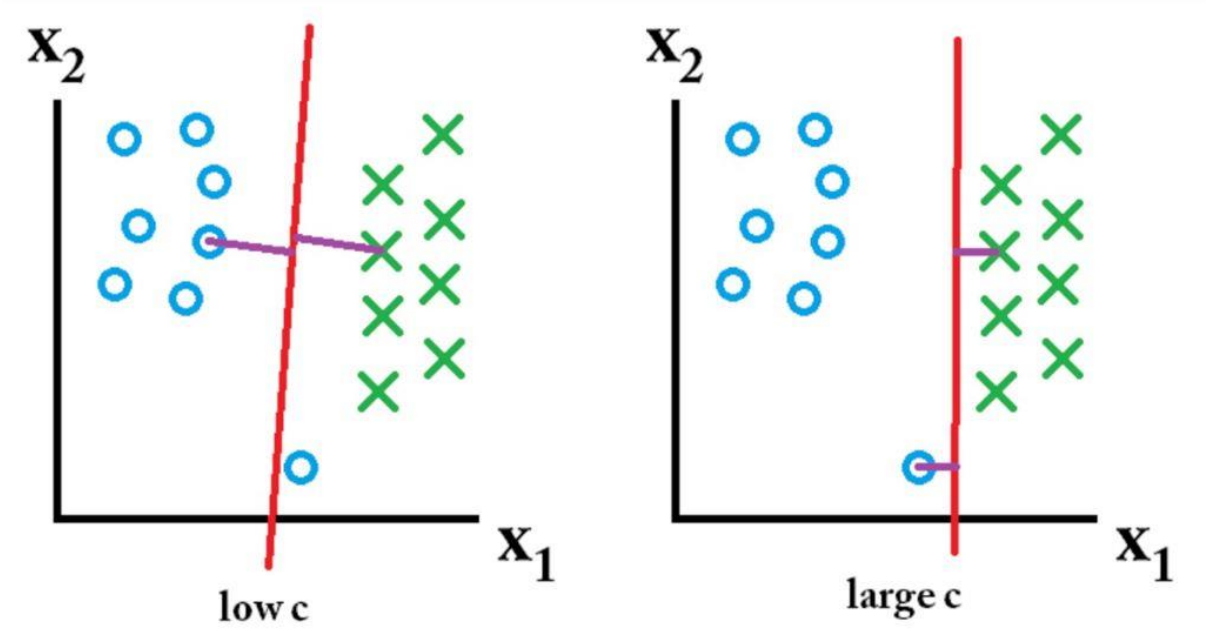


Figure 9: Model with a) Lower C value b) Higher C value

Gamma

It controls the influence of new features on the decision boundary. Higher the value of gamma, higher is the influence that the features will have on the decision boundary. i.e. the boundary will be more wiggly. In other words, it controls the distance of influence of a single training point. Low values of gamma indicate a large similarity radius which results in more points being grouped together. For high values of gamma, the points need to be very close to each other in order to be considered in the same group (or class). Therefore, models with very large gamma values tend to overfit.

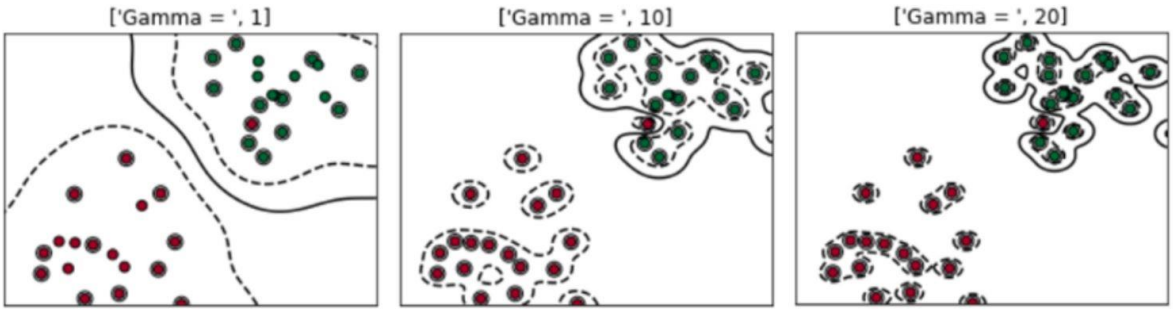


Figure 10: Plots with gamma value of a) 1 b) 10 c) 20

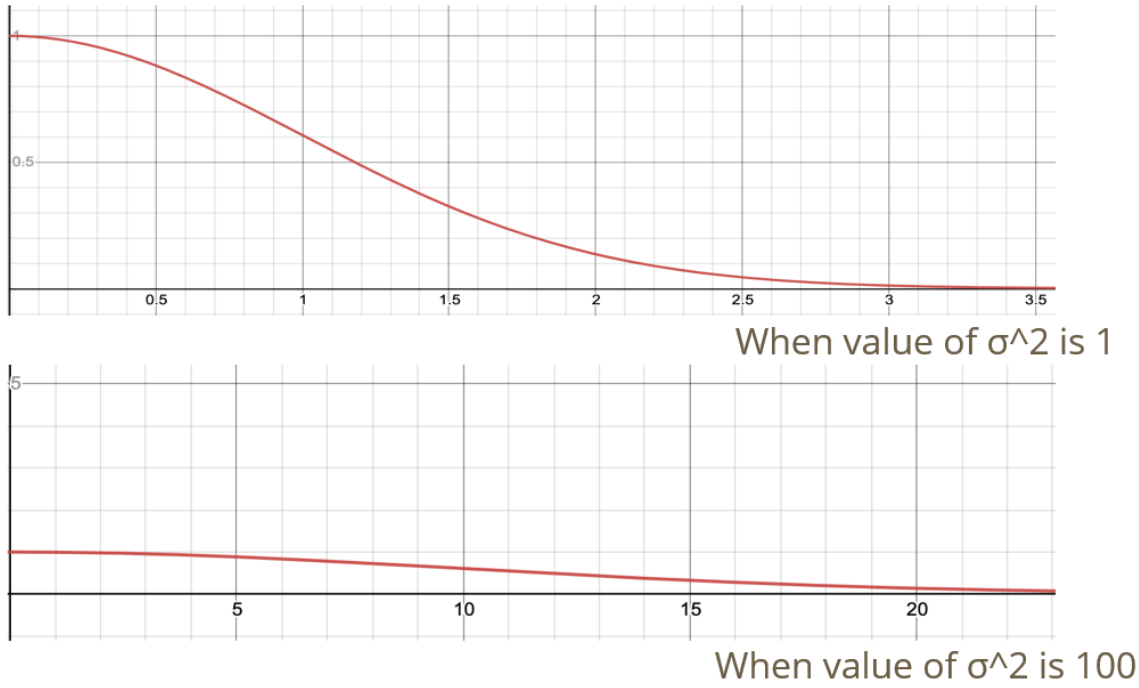


Figure 11: a) Plot with lower value of square value gives lower region of similarity b) Plot with higher value of square of sigma gives higher region of similarity.

As gamma is inversely proportional to σ^2, the value of σ^2 increases and hence more points lie in the region of similarity and vice-versa. It can be observed from the graph that the region of similarity is higher when the value of σ^2 is 100 when as compared to the value of σ^2 is 1.

we can conclude that as γ increases, i.e. σ reduces, the model tends to overfit for a given value of C.

Non-Linear SVMs

Non-Linear SVMs refer to the projection of data that cannot be linearly separable into a higher dimensional space can make it linearly separable. The general idea involves mapping original feature space to some higher dimensional space where the training set could be distinguished such that we can preserve relevant dimensions of relatedness between data points, so that the resultant classifier could be generalized.

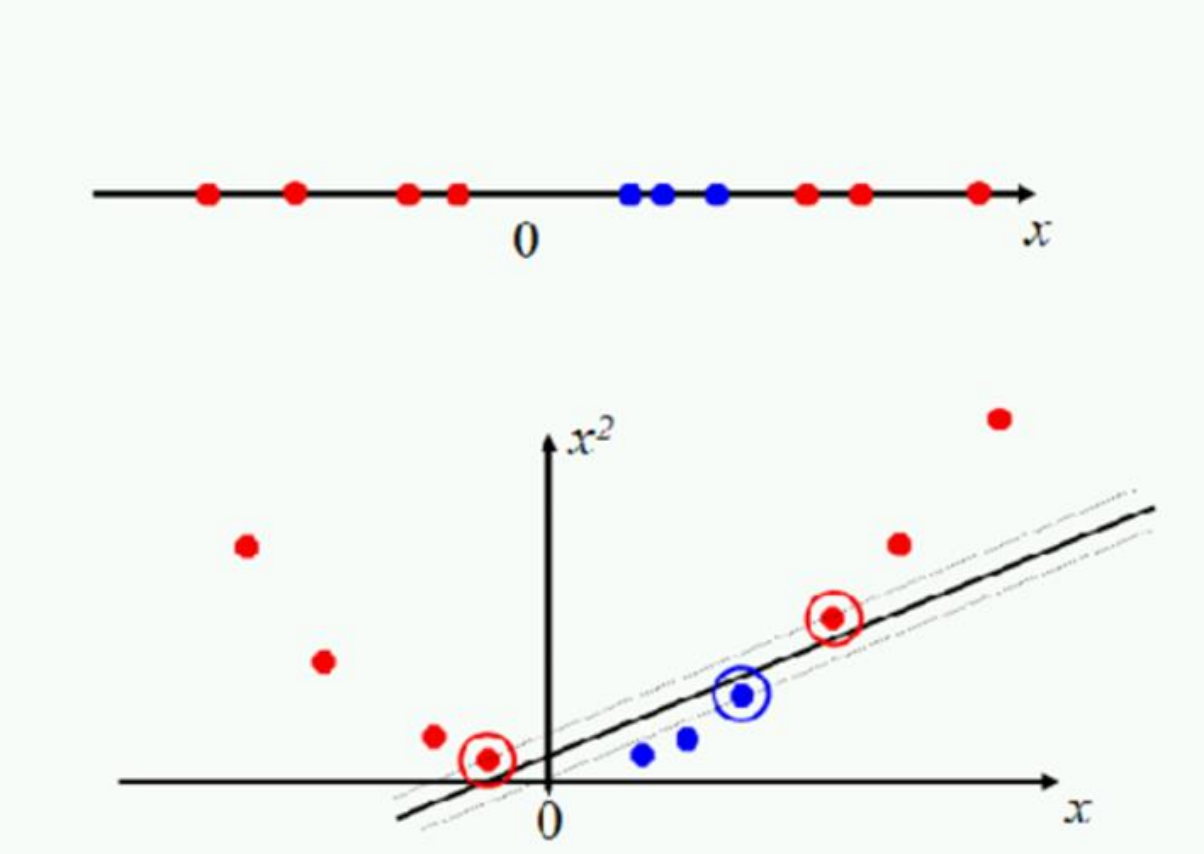


Figure 12: Projection of non linear data points

Linear SVMs

Linear SVMs refers to the projection of data that is linearly separable without changing the dimensions to draw a hyperplane. The given SVM could be separated by dot in 1D, a line in 2D and by a plane in 3D.

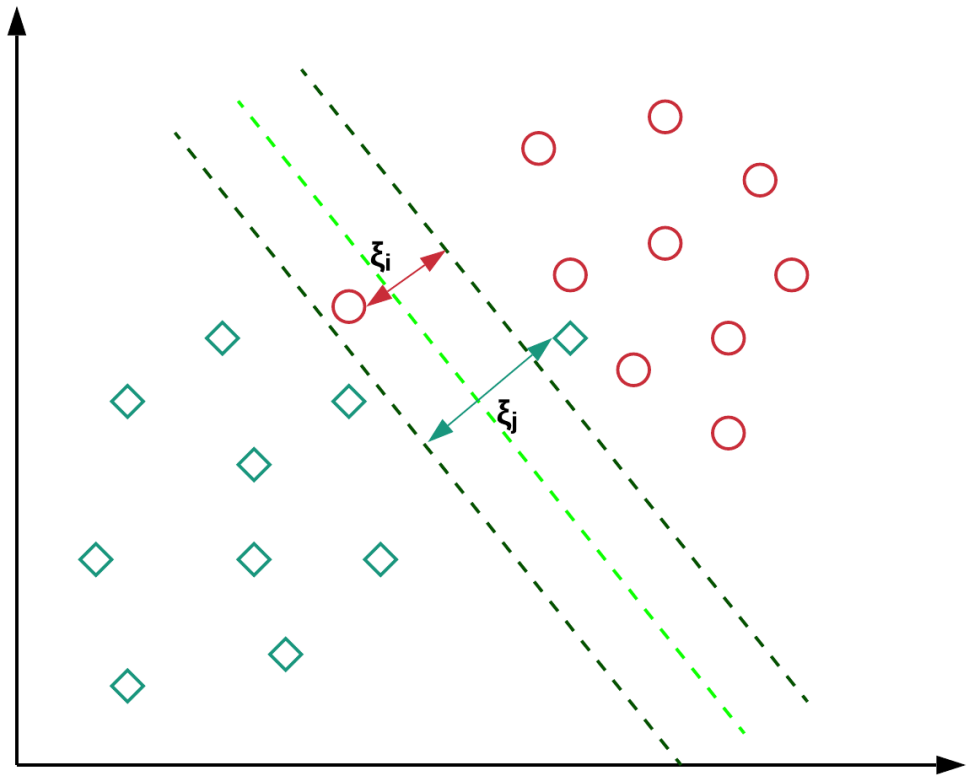


Figure 12: Projection of linear data as linear SVM

Kernel Trick

In real life applications, a dataset might have multiple features and one of the major functionalities of kernel is the transformation of a non-linear decision surface to a linear equation in a higher number of dimension space. One of the primary methods to do so are called as kernel tricks. In other words, it is a way through which nonlinear data could be projected onto a higher dimension to make it easier to classify the data where it could be linearly divided by a hyperplane.

Applying the Kernel trick would simply means replacing the dot product of two examples by a kernel function.

From Wolfe Dual Problem,

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

$$subject\ to\ \alpha_i \geq 0, i = 1...m, \sum_{i=1}^{m} a_i y_i = 0 \quad (31)$$

To solve this problem, we only care about the result of the dot product xi·xj. If there is a function which could calculate the dot product and the result is the same as when we transform the data into higher dimension, it would be useful. The function is called kernel function.

So, for example, in case of linear kernel, we would rewrite the above problem as:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \quad (32)$$

$$subject\ to\ \alpha_i \geq 0, i = 1...m, \sum_{i=1}^{m} a_i y_i = 0$$

The trick is quite powerful as the result of the dot product is performed in another space and hence provides us with the ability to change the kernel function in order to classify non-linearly separable data.

Kernel Functions

SVM makes use of kernels which is a group of mathematical functions. The main task of a kernel is taking the data as input and transform it into desired form.

Different types of SVM algorithms make use of various kernel functions.

They include: -

- Linear kernel
- rbf Kernel
- Polynomial
- Sigmoid
- Precomputed

Linear Kernel

Linear Kernel is one of the most basic type of kernel which is one-dimensional in nature. It is used when data is linearly separable, i.e. it can be separated by a line. It is mostly preferred in text-classification scenarios and various other classification problems as it involves data that can be linearly separated. Besides, linear kernel is faster than any other kernel as training a SVM in linear kernel is faster.

$$K_s(\chi, \chi_i) = \chi \cdot \chi_i \quad (33)$$

Where x and x_i are two support vectors

Rbf Kernel

When SVMs make use of rbf as a kernel function, it's referred to as rbf SVM or simple SVM. It is also called as gaussian kernel.

Rbf or radial basis function computes the resemblance between two points from the dataset. It does so by calculating exponential function of negation of division of square of Euclidean Distance between those two points with twice the variance σ, with base e.

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (34)$$

where σ works as a Hyperparameter for Rbf SVM and $||x-x'||^2$ is the squared Euclidean distance between two data points x and x'.

Explaining K in rbf kernel

Case 1: - When the value of x and x' are same

In this case as Euclidean Distance becomes 0 so the the function changes to exp (0) which equals 1, which is the value of K(x,x').

Case 2: - When value of x and x' are very far away.

In this case more the distance between two points more smaller will be the value of K(x,x'), although value can never become negative.

It is one of the most popular kernels due to the following reasons:

- It non-linearly maps the samples in a higher dimensional space as compared to linear kernel.
- It has lesser hyperparameters than polynomial kernel.
- It has lesser numerical difficulties.

Polynomial Kernel

In polynomial kernel, we simply perform the calculation of dot product by increasing the power of kernel.

$$K(X_1, X_2) = (a + X_1^T X_2)^b \quad (35)$$

b = degree of kernel & a = constant term.

The kernel contains only two parameters: a constant term a and b is degree of kernel. A d value with 1 is just the linear kernel. A larger value of b will make the decision boundary more complex and might result in overfitting.

Sigmoid Kernel

The kernel was very popular for support vector machines due to their origin from neural network theory. SVM model using sigmoid kernel function is equivalent to that of a two-layer perceptron neural network.

$$\tilde{K}(x_i , x_j) = \tanh(\gamma\, x_i^T\, x_j + r) \tag{36}$$

Explaining σ parameter

Gamma parameter is considered as the 'spread' of the kernel and thereby the decision region. When value of gamma is low, the 'curve' of decision boundary is lower too and hence the decision region is very broad. When the value of gamma is high, the 'curve' of the decision boundary is high which tends to create an island of decision-boundaries around the data points.

If gamma='scale' then 1/(n_features* X.var()) as value of gamma.

If 'auto' then it uses 1/n_features.

Explaining coef0 parameter

Coef0 is a parameter of kernel projection which is used to overcome an important issue in polynomial kernel. On using coef0=0, it works fine but when p -> infinity, it separates the pair of points more and more. The gap is widened when (x,y) is smaller than one and (a,b) is greater than one as taking powers of (x,y) makes it go closer to 0 and (a,b) value grows towards infinity. Therefore, coef0 is set to scale the data so that there would not be any such distinction.

How to choose the best possible kernel function?

Selection of kernel for any given dataset depends on which problem we are solving. If our data is linearly separable then we choose linear kernel as it takes lesser training time.

- Linear Kernels is the most preferable choice for kernel functions when it is used for text classification problems. It performs well when the size of the dataset is large.
- Rbf is a type of kernel which gives good results when there is no information regarding the data and projects the data in higher dimensions and keeps searching for a linear separation.
- Polynomial kernels provide good results where all the training data is normalized.

Model Selection of SVM

One of the most important aspects of SVM is that of model selection. The success of SVM heavily relies on the tuning of several parameters. The parameter tuning process is referred to as model selection.

In the case of linear SVM, we need to tune only the cost parameter C. However, it is very rare to get linearly separable data which is often applied to the linearly separable problems. Most of the real-world data is non-linearly separable. Thereby, we often use nonlinear kernel to solve the cost parameter (C) and kernel parameters (γ, d).

To find the best values of the parameter set, we use grid-search method in cross validation. Then the acquired values of different parameters are applied to the training dataset. After which, the classifier will be used to classify the testing dataset and produce an accuracy score.

Preparing data for SVM

1. Numerical Conversion

SVM requires the inputs to be numerical instead of categorical values. So, there is a need for conversion using some encoding techniques like one hot encoding, label encoding etc.

2. Binary Conversion

As SVM could be used to classify binary data, thereby we need to convert the multi-dimensional dataset into binary form by using one v/s rest method and one v/s one method.

Advantage and disadvantage of SVMs.

Pros

1) SVM is advantageous for higher dimensions.

In real world application, it is quite common to have data with infinite dimensions. For e.g.: - applications using image data, medical data, genetic data etc. have higher dimensions and SVM is very useful in these cases. In other words, when the number of features/columns are higher, SVM is known to perform well.

2) It is the best algorithm when we consider the cases when the classes are separable.

3) They make use of a subset of training points in the decision function and hence, it is highly memory efficient.

4) SVM models are relatively stable and a small modification to the data does not affect the hyperplane.

5) SVM handles non-linear data efficiently using kernel trick.

Cons

1) While processing larger datasets, the algorithm is quite slow.

Training larger datasets in SVM is a problem because the quadratic form is completely dense and the memory requirements grow with the square of the number of data points.

2) Selection appropriate kernel functions and hyperparameters is quite important and tricky. If this step is not done properly, then it could affect our final accuracy score.

3) SVM does not perform well when the dataset has more noise i.e. the target classes are overlapping.

4) SVM tends to underperform when there are higher number of features for each corresponding data points as compared to the number of training data samples.

5) Memory Requirements of SVM is quite high as we need lot of memory to store all of the support vectors and this number increases with increase in training dataset size.

Applications of svm

Face Detection- SVMs are used to classify sections of image as face and other sections as non-face and create a square boundary around the face.

Image Classification – SVMs provide great accuracy for image classification.

Text Categorization – SVMs are widely used for text categorization. They utilize the training data to classify (segregate) the documents in different categories. It performs categorization based on the generated score and is compared with threshold value.

Speech Recognition – Speech recognition is put in use for segregation of individual words from speech. Features of each individual word is extracted and trained successfully. One of the most common forms of speech recognition is making an interface for communication with deaf people. In this case, the acoustics is data and SVM uses it to train its models.

Implementation

Dataset

Analysis of Support Vector Machine is done using the ‘Gender Recognition by Voice and Speech analysis’. This dataset comprises of 3,168 instances of recorded voice samples which is collected from both male and female speakers. The samples were pre-processed by using acoustic analysis in R using seewave and tuner packages, with an analyzed frequency range of 0hz-280hz which is the human focal range.

- meanfreq: mean frequency (in kHz)
- sd: standard deviation of frequency
- median: median frequency (in kHz)
- Q25: first quantile (in kHz)

- Q75: third quantile (in kHz)
- IQR: interquantile range (in kHz)
- skew: skewness
- kurt: kurtosis
- sp.ent: spectral entropy
- sfm: spectral flatness
- mode: mode frequency
- centroid: frequency centroid
- meanfun: average of fundamental frequency measured across acoustic signal
- minfun: minimum fundamental frequency measured across acoustic signal
- maxfun: maximum fundamental frequency measured across acoustic signal
- meandom: average of dominant frequency measured across acoustic signal
- mindom: minimum of dominant frequency measured across acoustic signal
- maxdom: maximum of dominant frequency measured across acoustic signal
- dfrange: range of dominant frequency measured across acoustic signal
- modindx: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
- label: male or female

The dataset contains 20 input attributes and 1 output attribute. All the input values are continuous whereas the values in output are categorical. The categorical values in the output column ‘label’ are ‘male’ and ‘female’. The dataset consists of equal number of males and females in the output section. Also, the dataset has no missing values.

In data preprocessing, we have used Label Encoder to encode the categorical values of the output ‘Label’ column to numerical value. In addition, normalization of data is performed using Standard Scaler which follows Standard Normal Distribution.

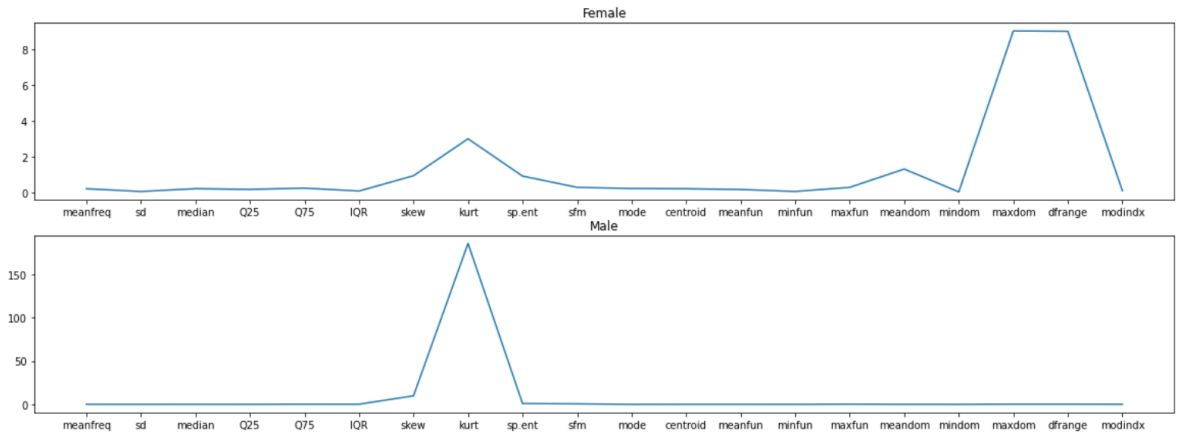


Figure 13: a) Plot for output label as Female b) Plot with output label as Male

The graphs clearly depict that Kurt is higher for males as compared to females whereas maxdom and dfrange value are quite low. These attributes are essential for classification of data.

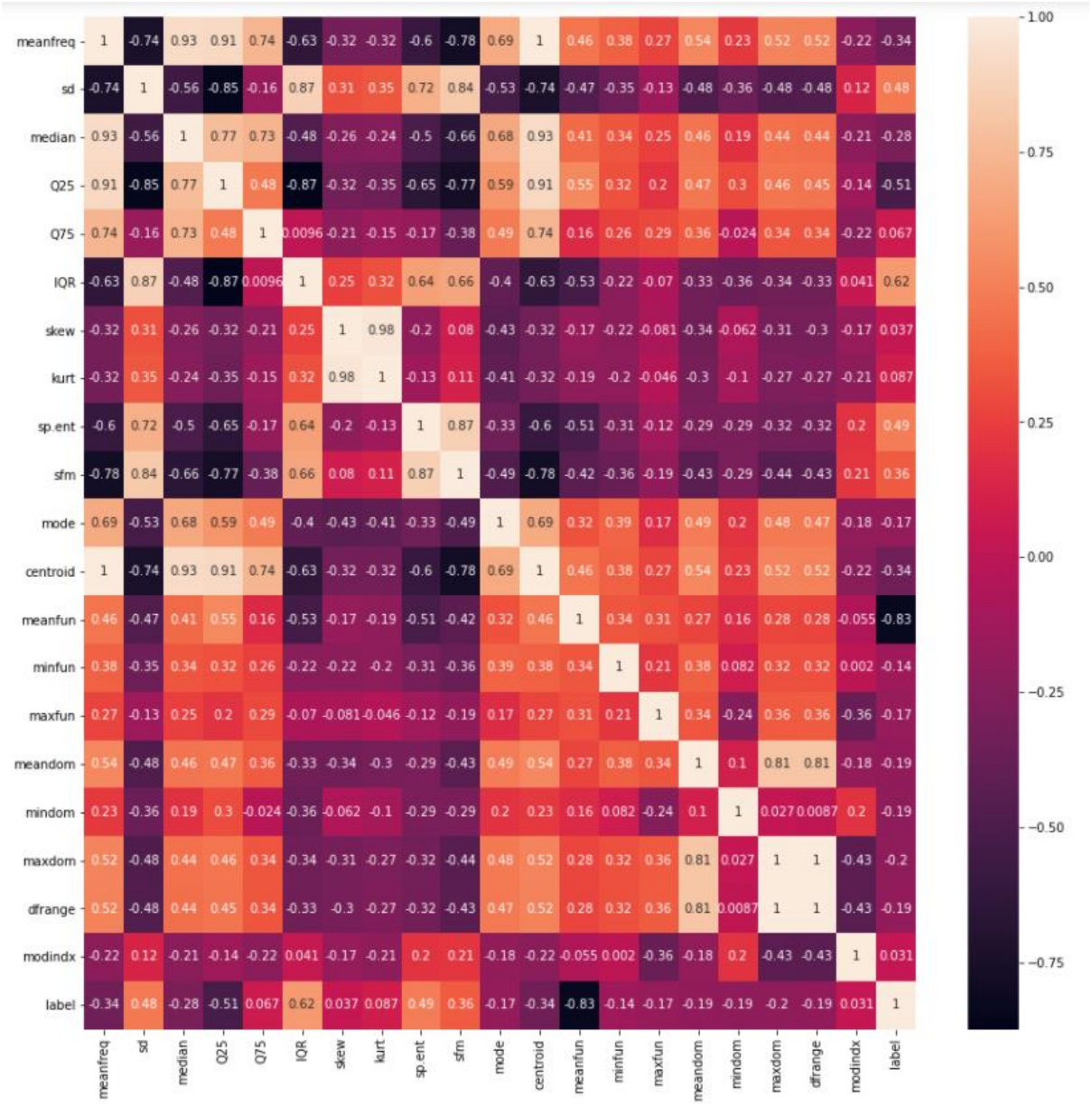


Figure 14: Linearity check using HeatMap

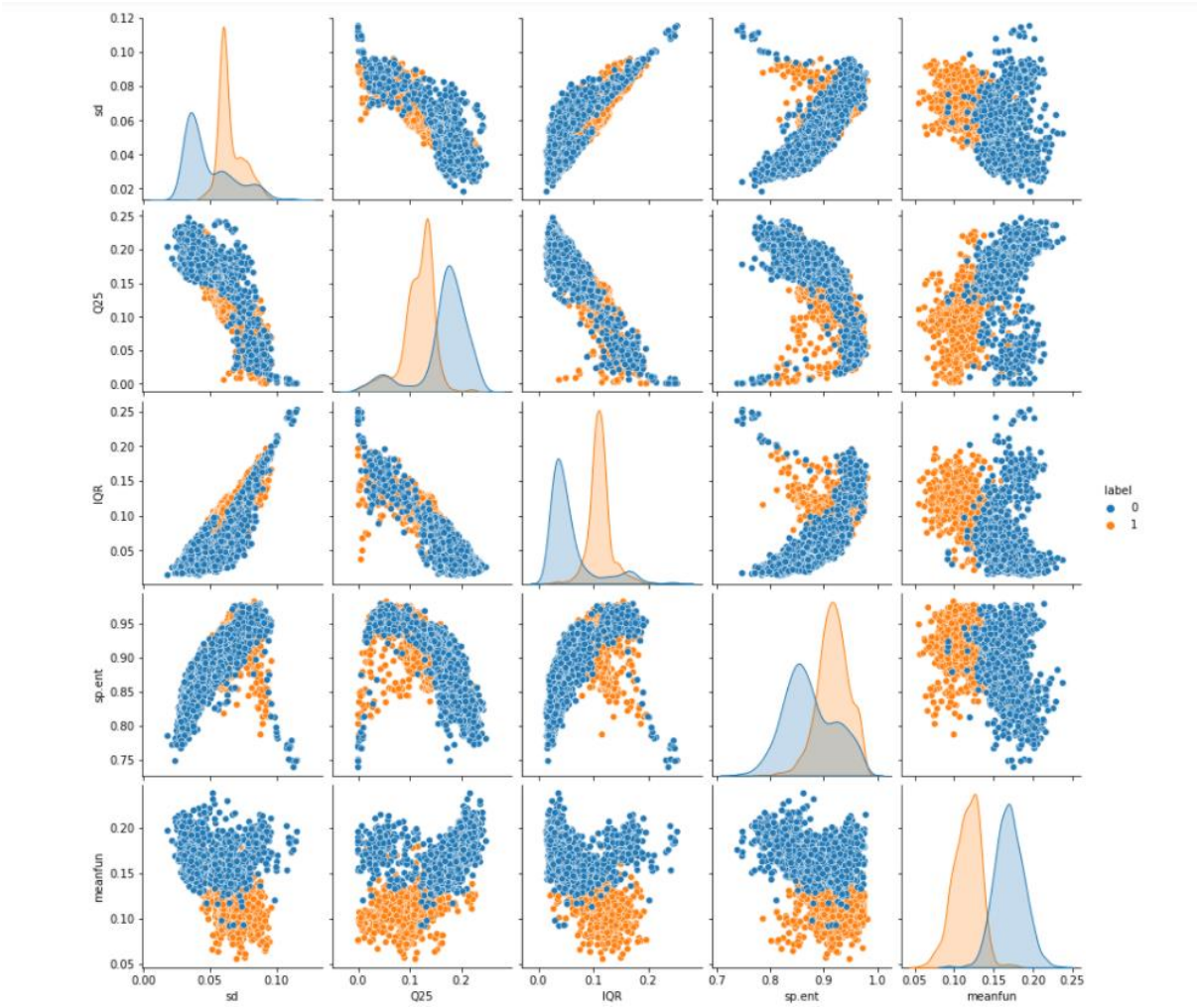


Figure 15: Pairplot with output labels and corresponding features having correlation greater than 0.4 (+ve/-ve)

This pair plot is made from the attributes of the dataset having a correlation greater than 0.4 (both positive and negative).

Hyperparameter C calculation for non-linear kernels

Tuning for Hyperparameter C is performed to increase the overall accuracy of the model.

a) A range of values for C is considered. The list is as follows:

C_values=[0.0001,0.001,0.01,0.1,1,10,100]

Using, the list C_values, accuracy score was computed for each individual value in the list.

Accuracy score = [0.5582318412330791, 0.5582318412330791, 0.8879207762648245, 0.9583316695284111, 0.9665325639899376, 0.9602164277442797, 0.9491714251487442]

The plot is made with Value of C v/s Cross Validated Score and shown below:

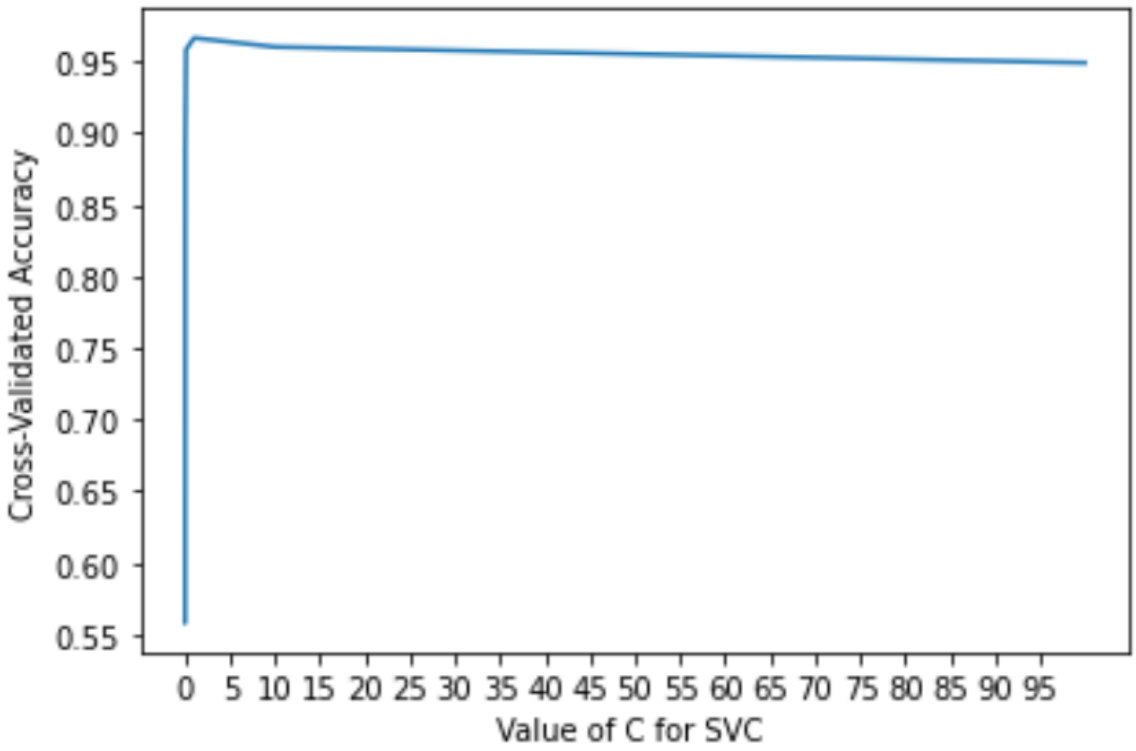


Figure 16: Cross Validated Accuracy v/s C value

From the given plot, it can be inferred that the accuracy score tends to decrease when we move from 1 towards 100 in the C_values list. So, the values 10,100 could be removed from the C_values list.

The process is repeated with the following values of C:

C_values=[0.0001,0.001,0.01,0.1,1]

The accuracy score for the values C values is calculated:

Accuracy Score= [0.5582318412330791, 0.5582318412330791, 0.8879207762648245, 0.958 3316695284111, 0.9665325639899376]

The plot is made with Value of C v/s Cross Validated Score and shown below:

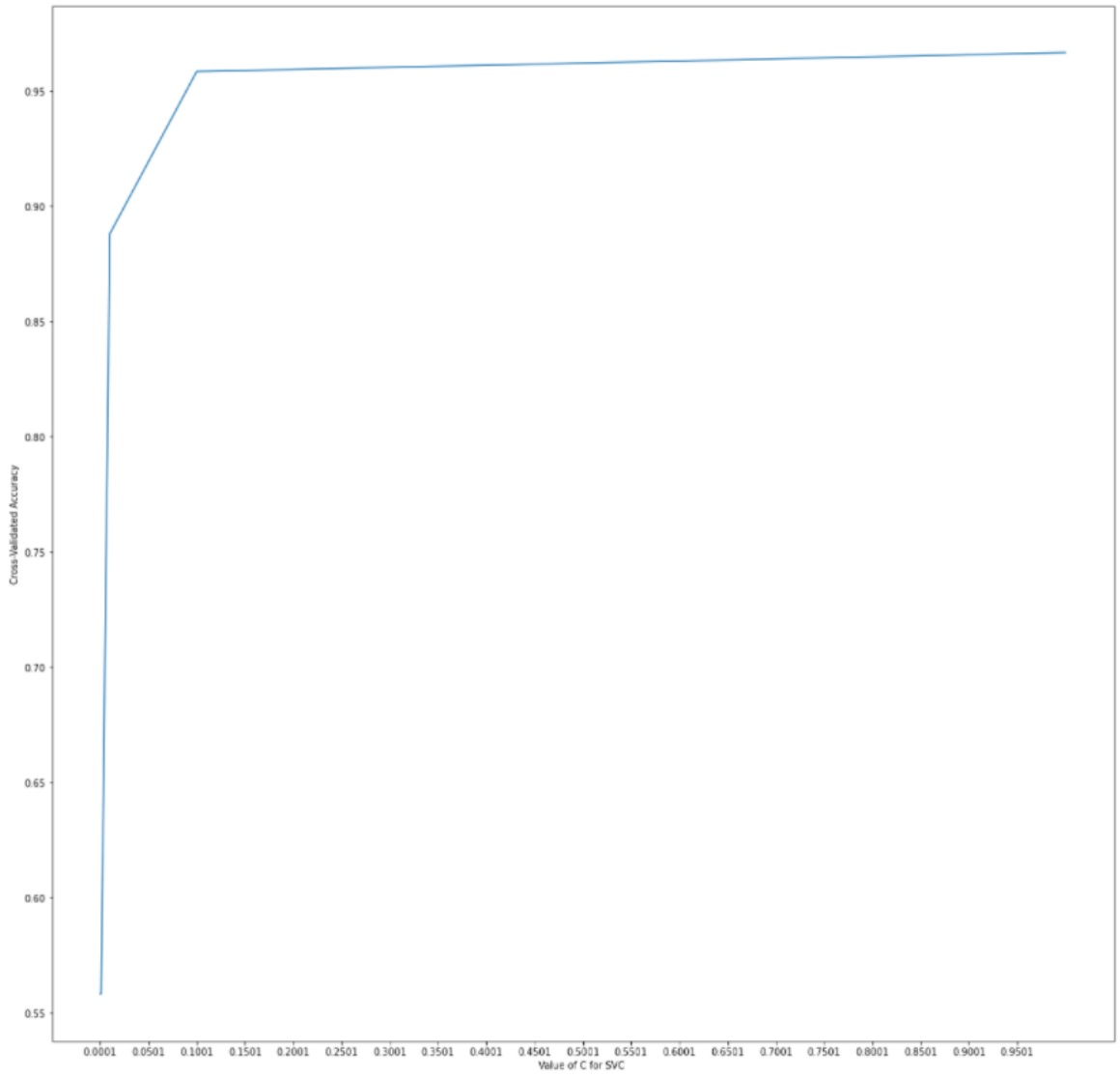


Figure 17: Cross Validated Accuracy v/s C value

The accuracy score was found to be very low for the values 0.0001 and 0.001. So, we eliminate them from the list.

The process is repeated with the following values of C:

C_values = [0.1,1,10]

The accuracy score for the values C values is calculated:

Accuracy Score= [0.9583316695284111, 0.9665325639899376, 0.9602164277442797]

The plot is made with Value of C v/s Cross Validated Score and shown below

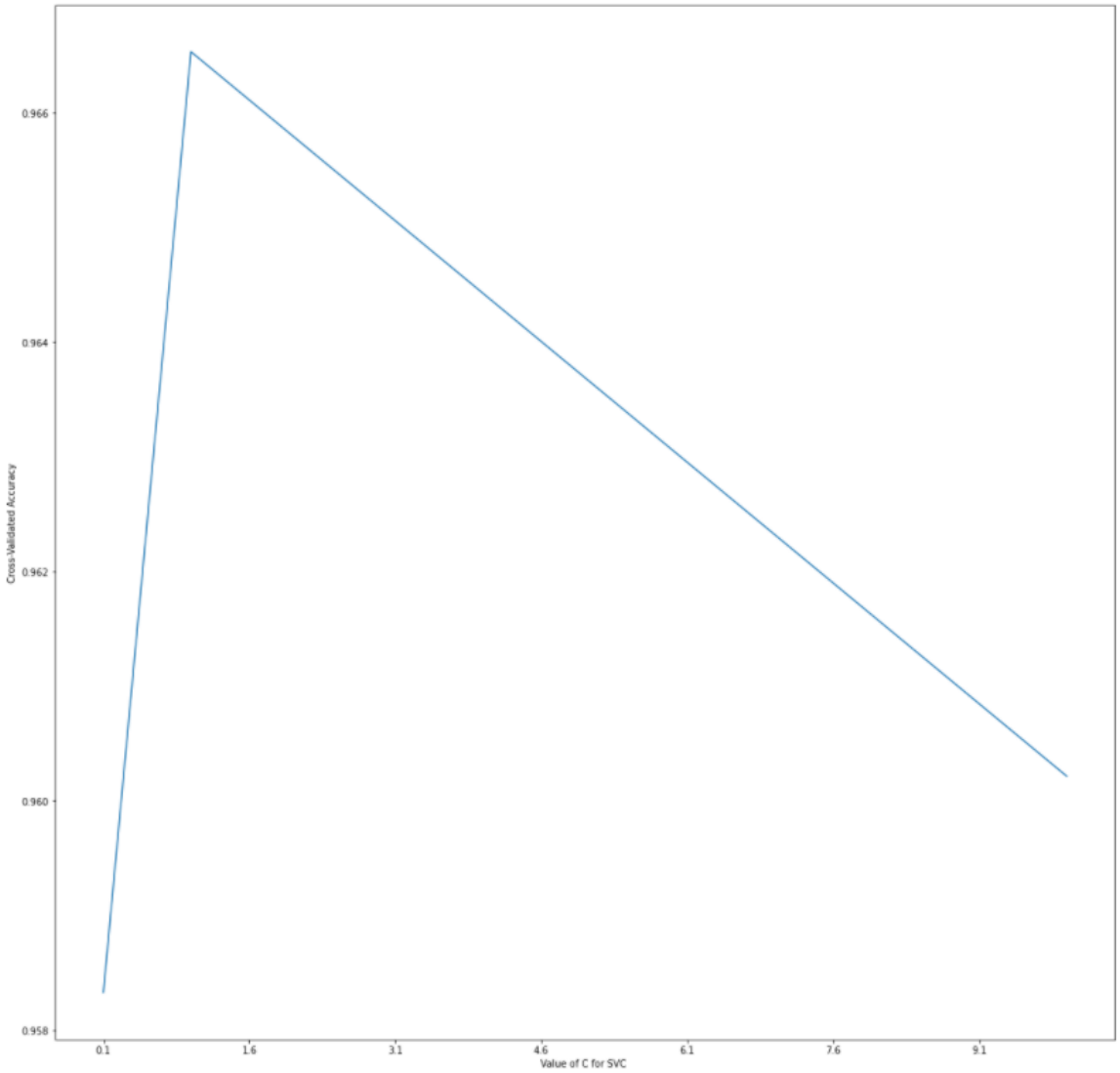


Figure 18: Cross Validated Accuracy v/s C value

From the plot, we could observe that the cross validated accuracy is the highest at approximately 1.5. So we take, newer values of C in the next step.

The process is repeated with the following values of C:

C_values = [1.5,1.6,1.7]

The accuracy score for the values C values is calculated:

Accuracy Score= [0.965268737771034, 0.9646378229445354, 0.9643223655312863]

The plot is made with Value of C v/s Cross Validated Score and shown below

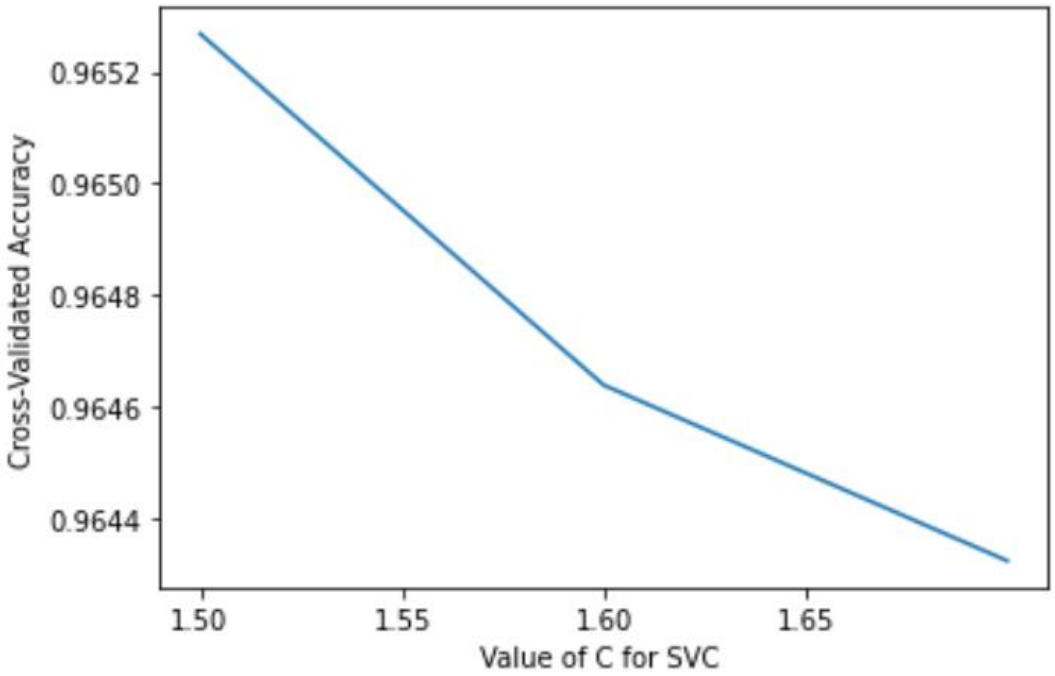


Figure 19: Cross Validated Accuracy v/s C value

The plot shows that the Cross Validated accuracy is the highest when C assumes the value of 1.5. So, we fixed the value of C to be 1.5.

The dataset was split into two parts where 80% of the data was used for training and 20% was used for testing.

Kernel Parameters

The table given below describes the values of parameters considered for different kernels in order to achieve a higher accuracy score. These parameters include C, gamma, degree, dual, penalty, loss etc. which are written against the different kernels.

Tabel 1: Kernel functions and their corresponding Parameter values

	C	Gamma	Degree	Dual	Penalty	Loss
Rbf	1.5	0.01	N.A.	N.A.	N.A.	N.A.
Poly	1.5	auto	1	N.A.	N.A.	N.A.
Sigmoid	1.5	0.004	N.A.	N.A.	N.A.	N.A.
Linear	1(default)	N.A.	N.A.	True(default)	L2(default)	Hinge

Results and Discussion

The following table shows a comparative analysis of the accuracy scores from different kernels before Hyperparameter tuning is applied and after Hyperparameter tuning is applied.

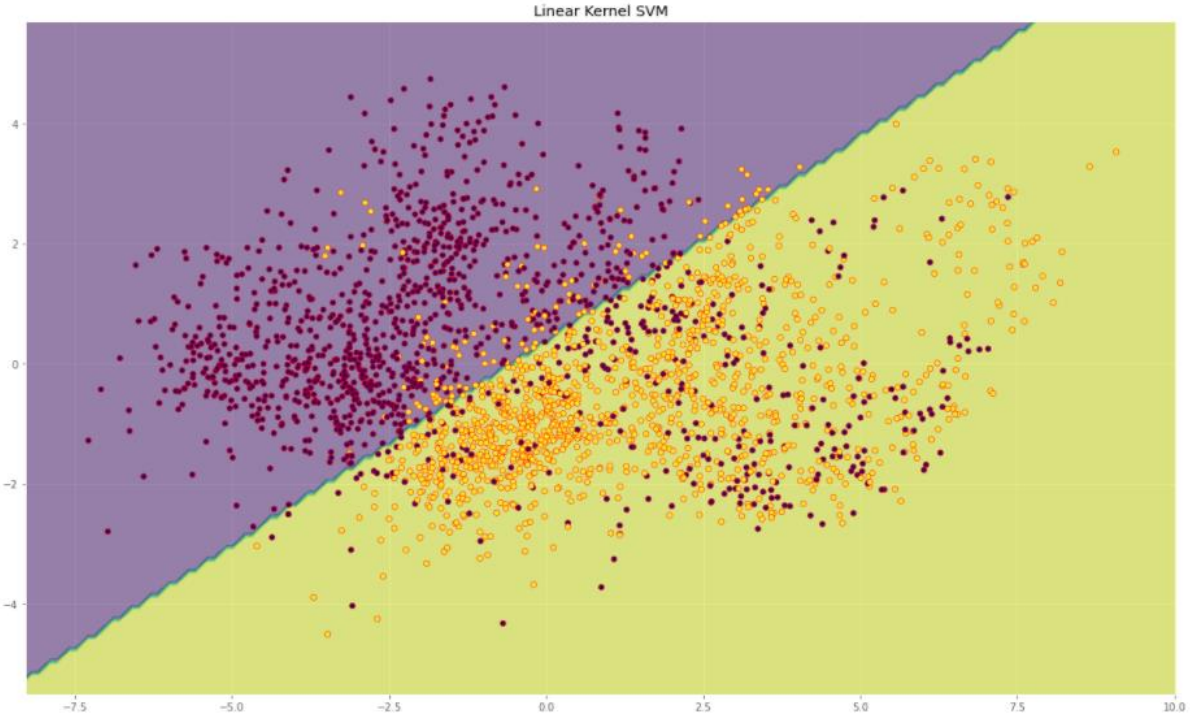


Figure 20: Plot for Linear Kernel SVM

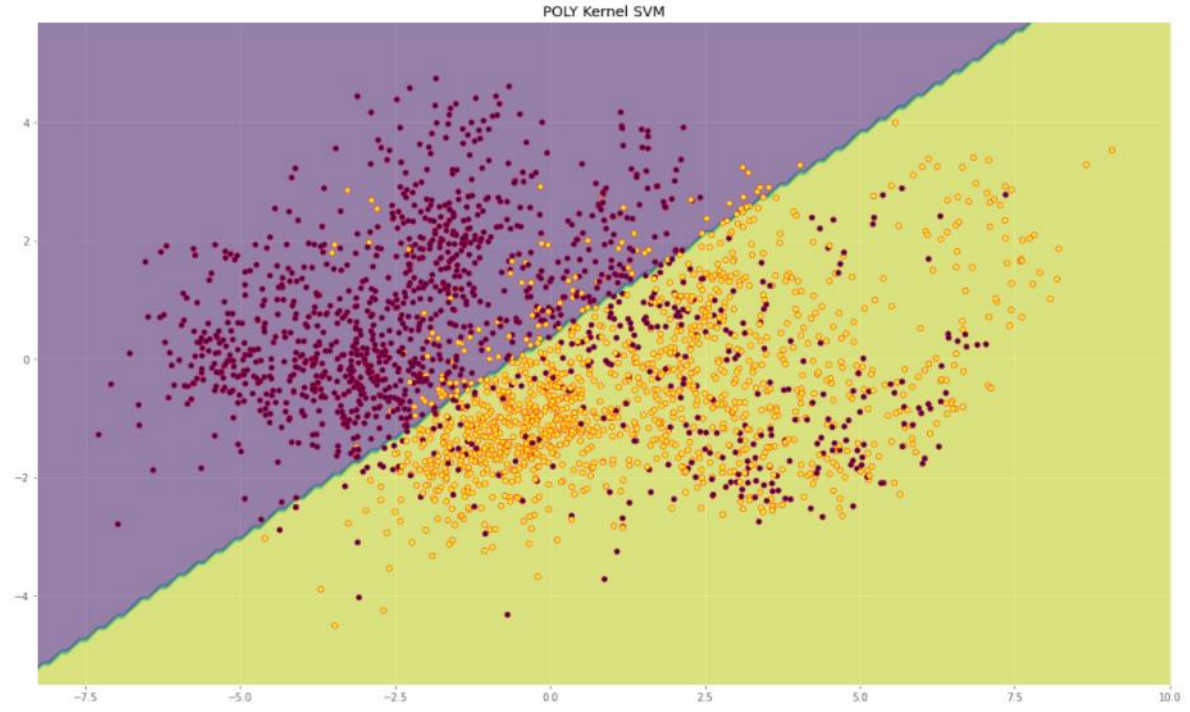


Figure 21: Plot for Polynomial Kernel SVM

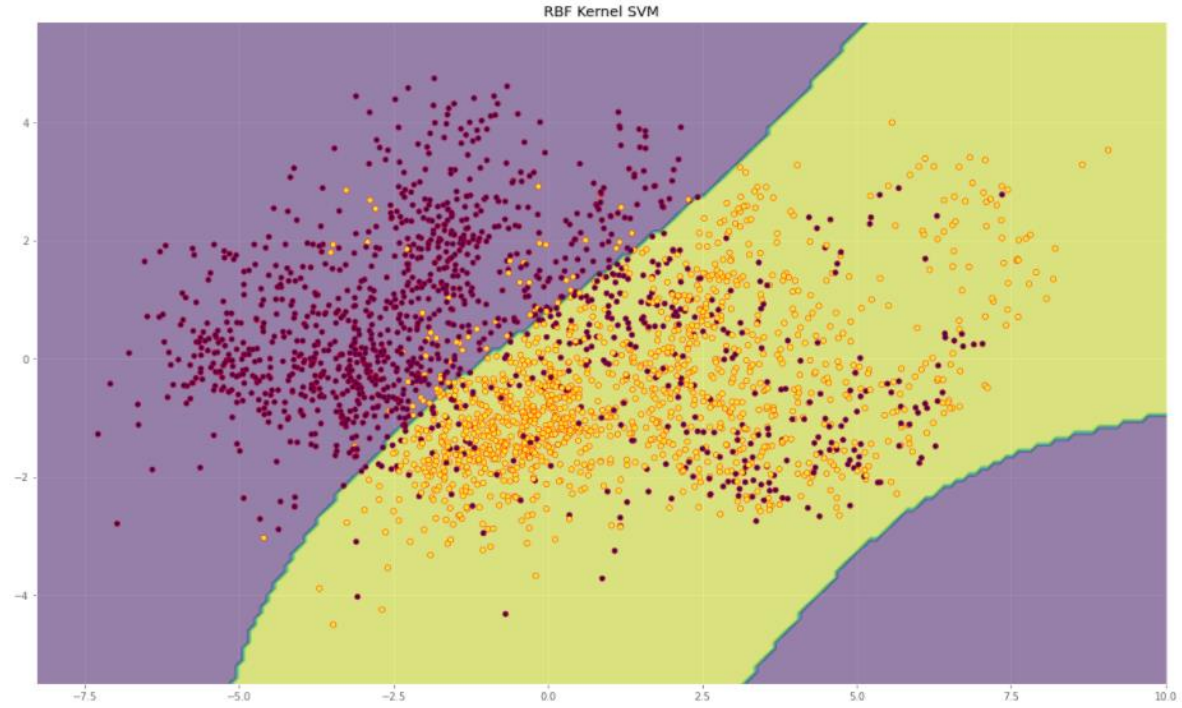


Figure 20: Plot for Rbf Kernel SVM

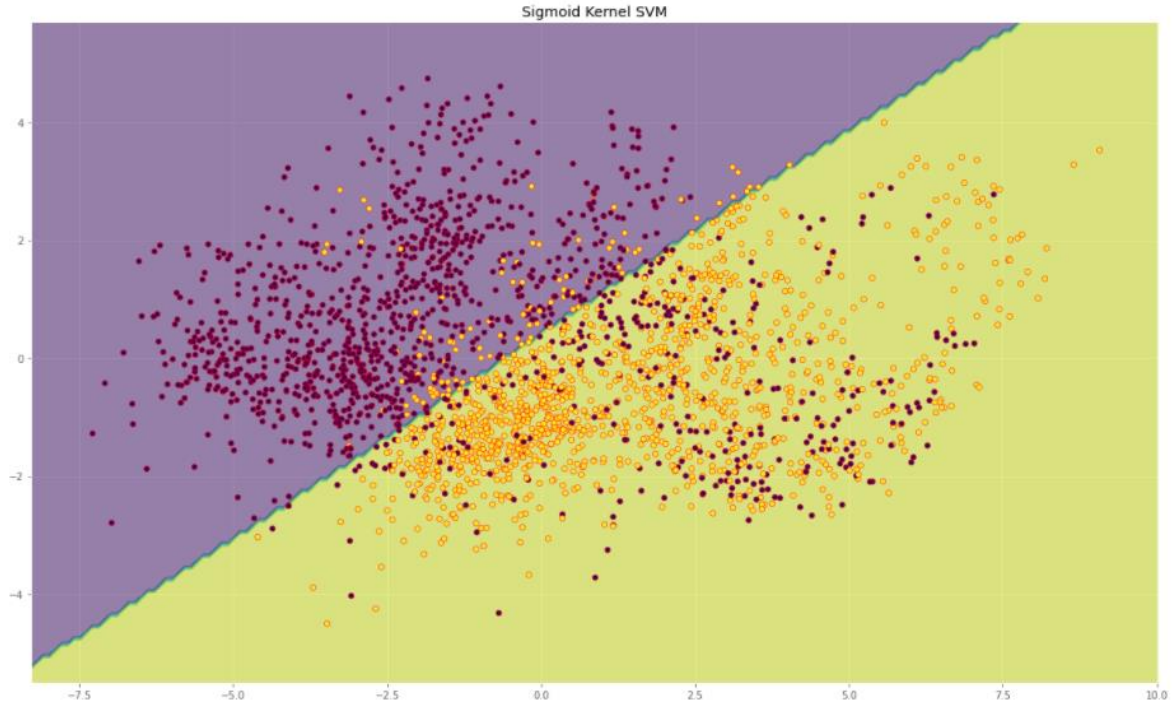


Figure 21: Plot for Sigmoid Kernel SVM

SVM Classification Score for Linear is : 0.9779179810725552

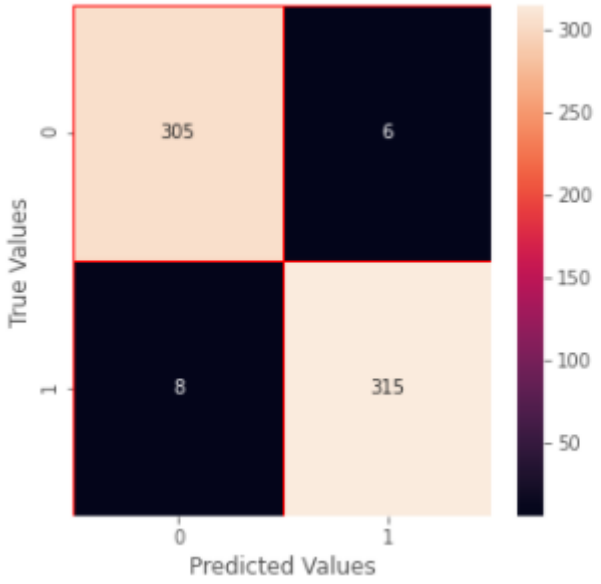


Figure 22: Confusion matrix for Linear Kernel SVM

SVM Classification Score for poly is : 0.9747634069400631

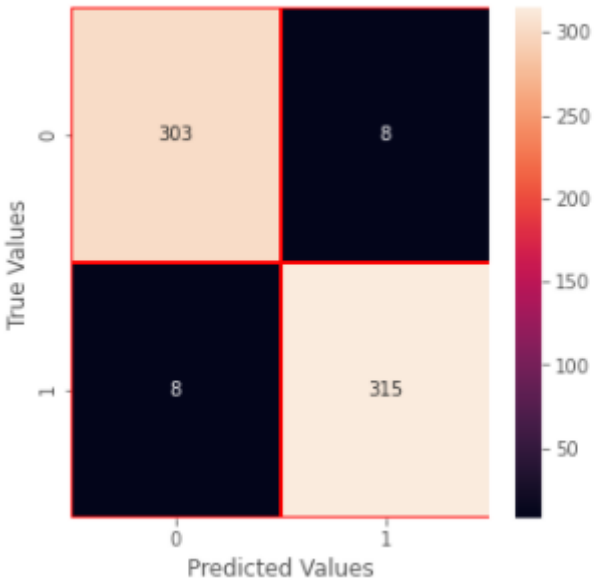


Figure 23: Confusion matrix for polynomial Kernel SVM

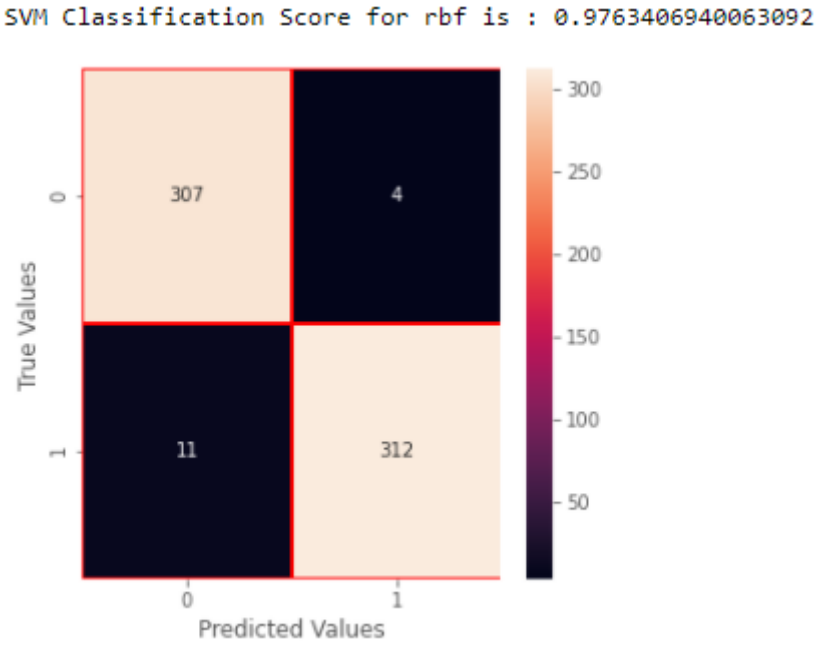


Figure 24: Confusion matrix for Rbf Kernel SVM

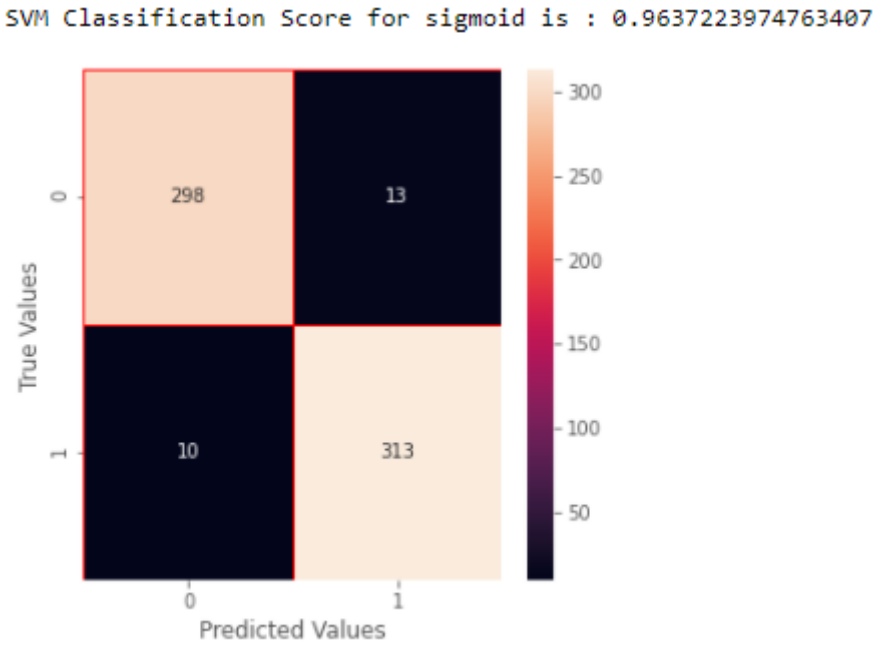


Figure 25: Confusion matrix for Sigmoid Kernel SVM

Table 2: Accuracy score comparison before and after hyperparameter tuning

Kernel	Before Hyperparameter Tuning	After Hyperparameter Tuning	%age increase in accuracy score
Rbf	0.9763406940063092	0.9763406940063092	0
Poly	0.9589905362776026	0.9747634069400631	1.61812
Sigmoid	0.7933753943217665	0.9637223974763407	17.6759
Linear	0.9700315457413249	0.9779179810725552	0.806452

It can be inferred from the above table that the best possible result is achieved in Linear Kernel which is closely followed by rbf and poly kernel. However, on performing Hyperparameter tuning, the highest increase in accuracy score is observed in the case of Sigmoid Kernel with a percentage increase of 17.67% whereas Linear Kernel which has the best result after tunning, shows the least percentage increase of 0.80%.

Conclusion and Future Works

SVM analysis was performed on the dataset "Gender Recognition By voice" and was observed that the linear kernel displayed the highest accuracy score whereas Sigmoid provided the highest variation in accuracy score after Hyperparameter tuning was performed.

SVM is considered to be one of the most powerful algorithms for its robust classification techniques. Research in some fields where SVMs do not perform well has spurred development of other applications such as SVM for large data sets, SVM for multi classification and SVM for unbalanced data sets. Further, SVM has been integrated with other advanced methods such as evolve algorithms, to enhance the ability of classification and optimize parameters. SVM algorithms have gained recognition in research and applications in several scientific and engineering areas.

References

[1] Durgesh, K. S., & Lekha, B. (2010). Data classification using support vector machine. Journal of theoretical and applied information technology, 12(1), 1-7.

[2] Bhavsar, H., & Panchal, M. H. (2012). A review on support vector machine for data classification. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(10), 185-189.

[3] Priya, R. S., & Soms, N. A Review on Support Vector Machine: a Classification Method.

[4] Anthony, G., Greg, H., & Tshilidzi, M. (2007). Classification of images using support vector machines. arXiv preprint arXiv:0709.3967.

[5] Khan, M. A., & Syed, N. A. (2015). Image processing techniques for automatic detection of tumor in human brain using SVM. Int J Adv Res Comput Commun Eng, 4(4).

[6] Galla, D. K. K., Mukamalla, B. R., & Chegireddy, R. P. R. (2020). Support vector machine-based feature extraction for gender recognition from objects using lasso classifier. Journal of Big Data, 7(1), 1-16.

[7] Aida-zade, K., Xocayev, A., & Rustamov, S. (2016, October). Speech recognition using support vector machines. In 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-4). IEEE.

[8] J.P.Medlin Julia, D.Bennet (2020). Weed Detection and Classification using ICA Based SVM Classifier. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020