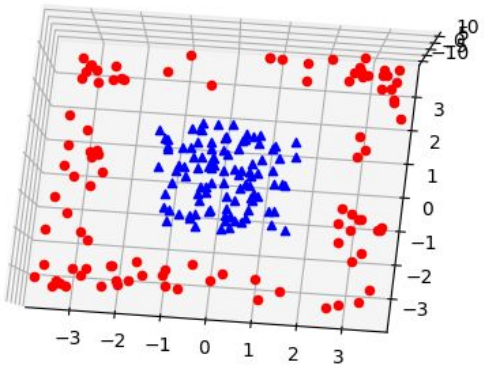
Kernel-Trick:

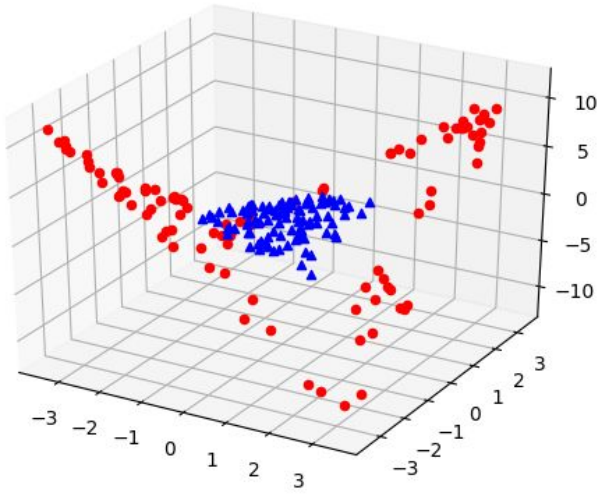
Name:B.Vishal Reddy
Rollo:201501173

Initial Data:

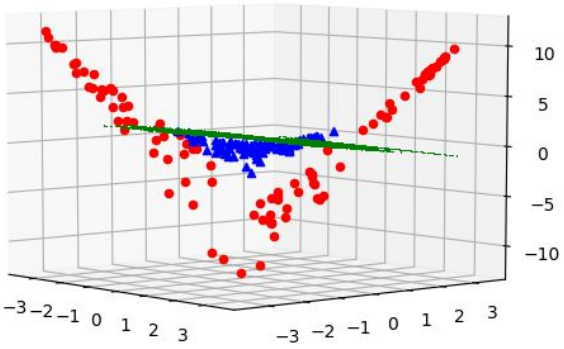


Kernel-1:

Kernel = [X1 X2 X1*X2]

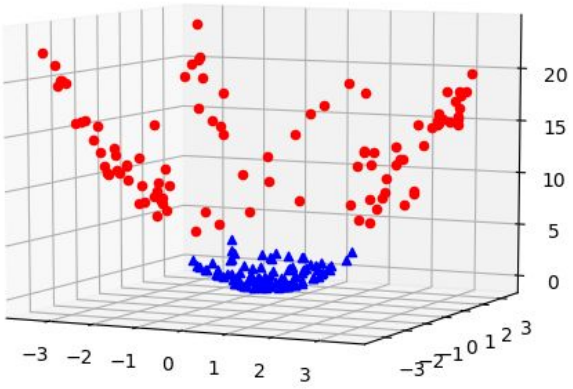


Using perceptron with max_iteration equal to 10000,
Getting an accuracy of 0.7 on training data
So the data is not linearly seperable using this kernel.
Image with Hyperplane:



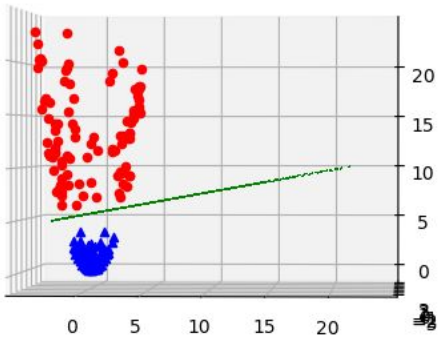
Kernel-2:

Kernel = [X1 X2 (X1*X1 + X2*X2)]



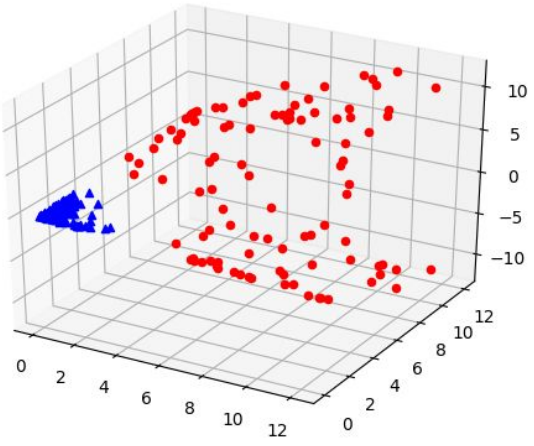
Using perceptron with max_iteration equal to 10000 and tolerance as 0 ,
Getting an accuracy of 1.0 on training data.
So the data is linearly separable using this kernel.

Image with Hyperplane:

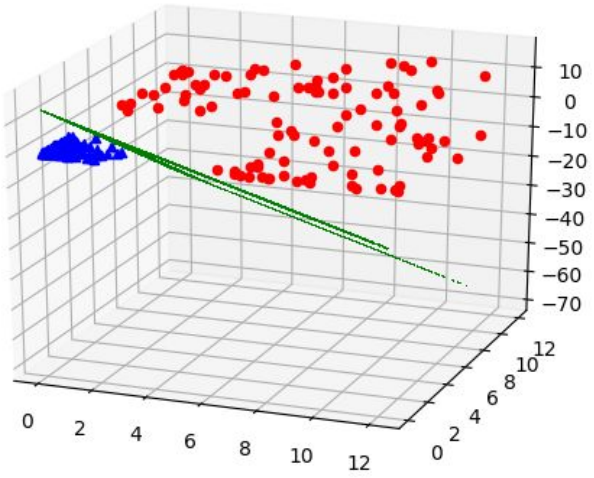


Kernel-3:

Kernel = [(X1*X1) (X2*X2) (X1*X2)]



Using perceptron with max_iteration equal to 10000 and tolerance as 0 ,
Getting an accuracy of 1.0 on training data.
So the data is linearly separable using this kernel.
Image with Hyperplane:



Letter_Classification:

1) ***Kernel -- RBF Kernel***:

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

Here, gamma and C are hyperparameters

When gamma = 1/no.of samples, ie, default value,
we get the values of Accuracy , precision, recall and f1 as 0.9413125, 0.943756004877 , 0.940793793641 , 0.942272479485.

When gamma = 0.001,
We get the values of Accuracy, precision, recall and f1 as 0.6997 , 0.71834147703 , 0.696687188202 , 0.707342985672

When gamma = 0.1
We get the values of Accuracy, precision, recall and f1 as 0.956 , 0.957586789849 , 0.95558095486 , 0.956582775763

When gamma = 0.5
We get the values of Accuracy, precision, recall and f1 as 0.9639 , 0.964639073674 , 0.963673805809 , 0.964156172394

When gamma = 1
We get the values of Accuracy, precision, recall and f1 as 0.9364 , 0.950964444749 , 0.936097008012 , 0.943469547225

When gamma = 10
We get the values of Accuracy, precision, recall and f1 as 0.2678 , 0.962033324436 , 0.265660978928 , 0.416296676166

Here, we observe that as gamma increases accuracy increases until gamma becomes 0.5 and then as gamma increases accuracy decreases. Reason is as the gamma value is very less, the radius is very large and the model becomes constrained and for the higher values of gamma(ie,>0.5) accuracy decreases as the model becomes too loose in restricting, since radius becomes smaller.

C parameter controls the extent to which we tolerate classifications. High C values will try to classify max data and hence will take help of many support vectors. Low C values will prefer a smooth surface and ignore mis-classifications to an extent.

This implies the best set of hyperparameters for ‘RBF’ kernel is when gamma is 0.5 and C is very high and C should not be too high as it may over-classify the data.

So, best set of hyperparameters for “RBF” kernel is when gamma = 0.5 and C = 10000
We get the accuracy, precision, recall and f1 as 0.9677 , 0.96849 , 0.967552158541 , 0.968023636034

2) ***Kernel -- Linear Kernel***:

$$\langle x, x' \rangle$$

Here, C is the only hyperparameter.

For large values of C, the optimisation will choose a small-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane mis-classifies more points.

When C = 1, We get the values of Accuracy, precision, recall and f1 as 0.8465 , 0.848401955596 , 0.845055245813 , 0.846725072098

When C = 100, We get the values of Accuracy , precision, recall and f1 as 0.8545 , 0.856475837146 , 0.853038068663 , 0.854753411887

When C = 1000, We get the values of Accuracy, precision, recall and f1 as 0.861 , 0.861981567269 , 0.859444679713 , 0.859653495745

Here, as C increases accuracy increases, but it takes huge time to train.

3) ***Kernel -- Polynomial Kernel***:

$$(\gamma\langle x, x'\rangle + r)^d$$

Here, gamma, C and Coef are hyperparameters.

For default values:
We get accuracy, precision, recall and f1 as 0.881 , 0.913095561745 , 0.880205504684 , 0.889664856891

For gamma=0.5, C = 100 and Coef = 0,
We get accuracy, precision, recall and f1 as 0.9375 , 0.939397290246 , 0.937068598276 , 0.937606959379

For gamma = 0.5, C = 10000 and Coef = 10,
We get accuracy, precision, recall and f1 as 0.9625 , 0.962999317169 , 0.962186400159 , 0.962366533728

For gamma = 10, C = 1000 and Coef = 10,
We get accuracy, precision, recall and f1 as 0.955 , 0.955549449683 , 0.954823737106 , 0.954848086065

So, the optimal values of gamma = 0.5, C = 10000 for polynomial kernel to give maximum accuracy.

Overall, the best accuracy is given by RBF kernel when gamma is 0.5 and C = 10000.