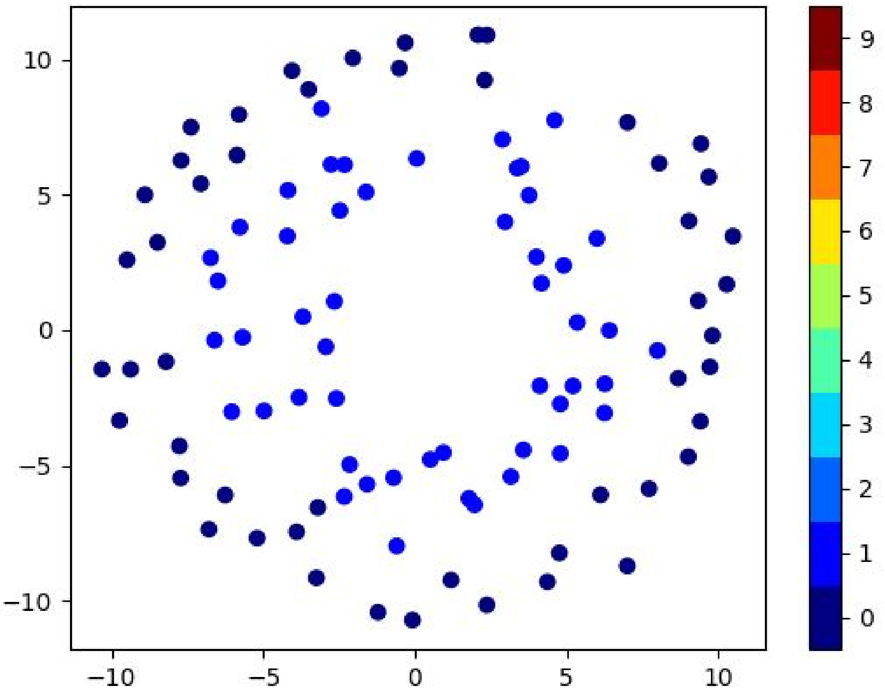
ML Assignment 2
Neha Jhamb
MT16037

Exploring data sets and kernels:

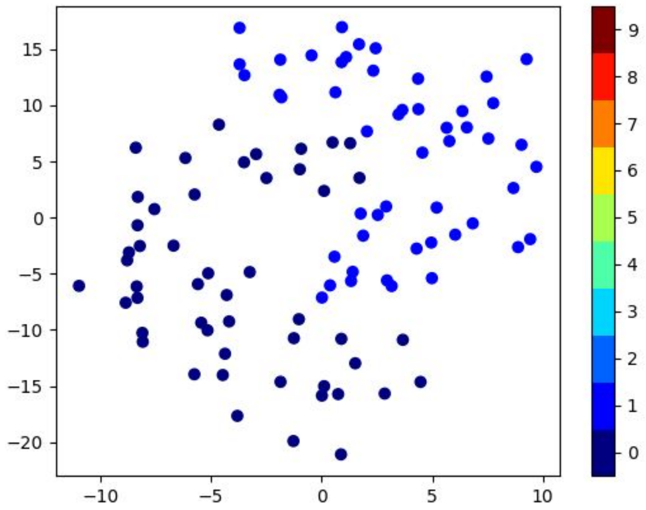
1. **Plots are shown below:**

data_1



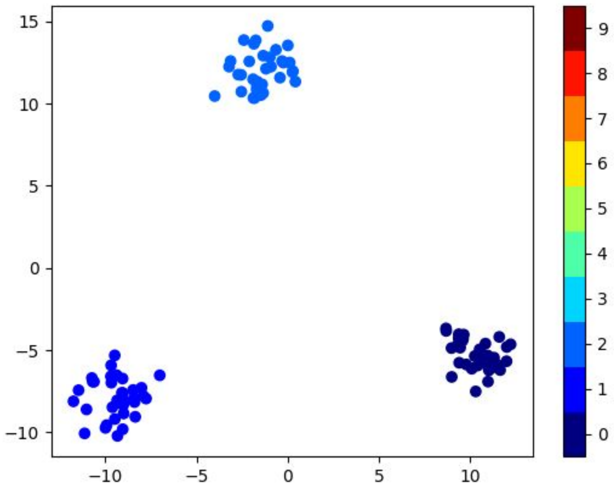
data_1 is a small dataset containing data points from 2 classes. These classes lie around a concentric circle. It is clear that this dataset is linearly separable if we plot its polar coordinates. The feature space is 2-D with classes 0 and 1.

data_2



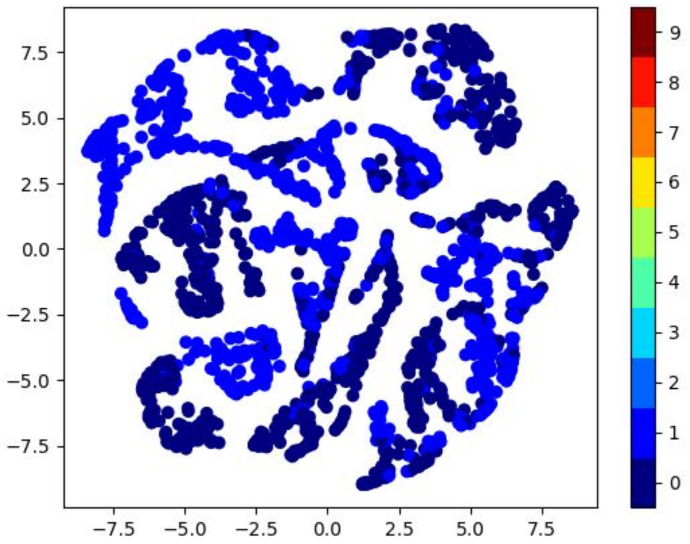
data_2 is again a small data set which has 2 classes. The data points lie in the shape of a bean for both the classes. This data set can be linearly separable in an infinite dimensional space. The feature space is 2-D with classes 0 and 1.

data_3



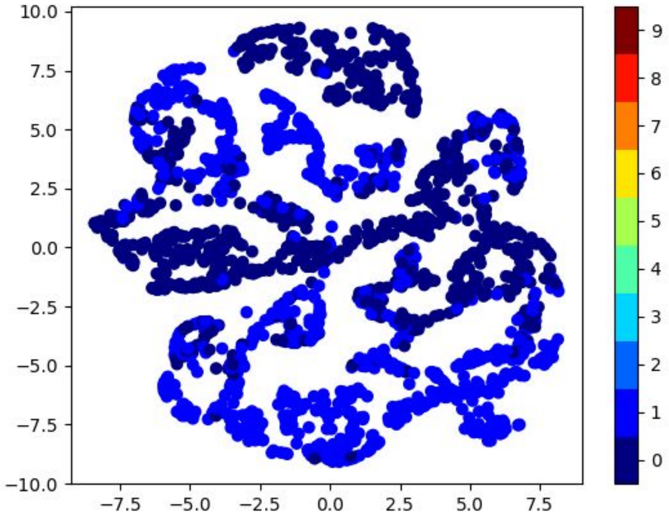
data_3 is a moderate sized dataset with 3 linearly separable classes.
The feature space is 2-D with classes 0, 1 and 2.

data_4



data_4 is a large data set which is clearly non separable in 2-D space. The feature space is 2-D with classes 0 and 1. We need a kernel to transform the data to a higher dimensional space to get a better view.

data_5



data_5 is a large data set which is clearly non separable in 2-D space. The feature space is 2-D with classes 0 and 1.

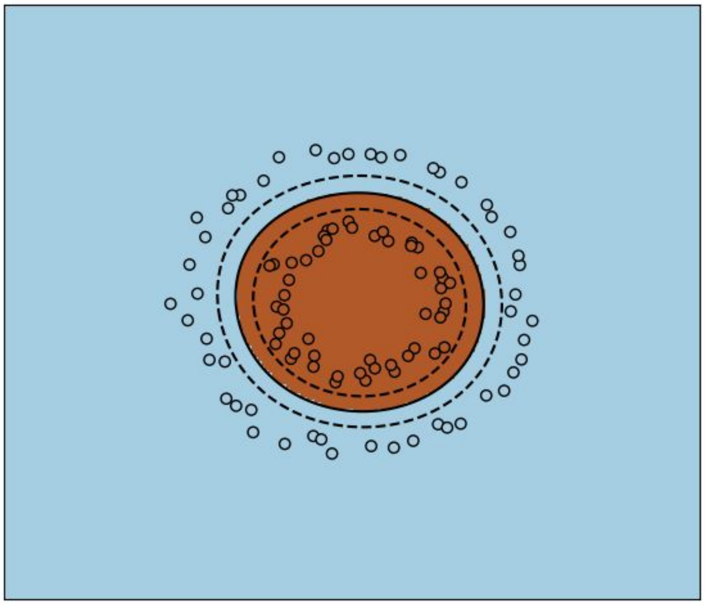
We need a kernel to transform the data to a higher dimensional space to get a better view.

----

2. Explain the choice of kernels.

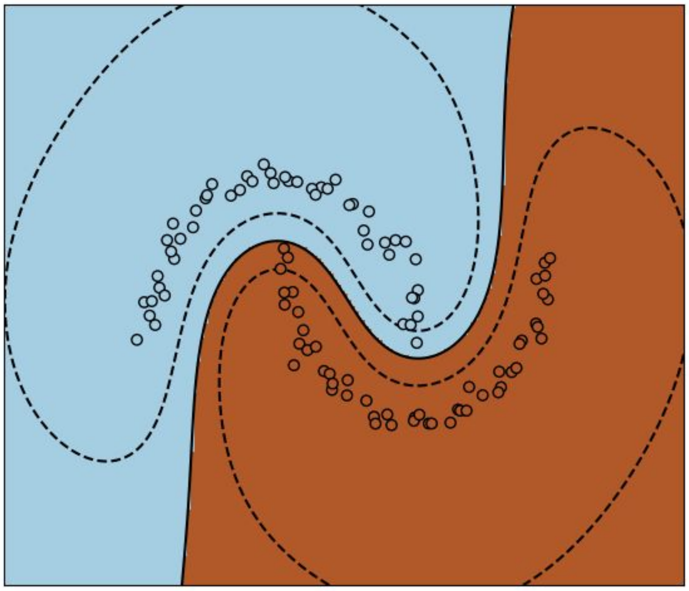
- **Data_1**

Used RBF kernel as the dataset is clearly in the form of concentric circles which can be easily separated using the radial basis function as seen in the below plot.



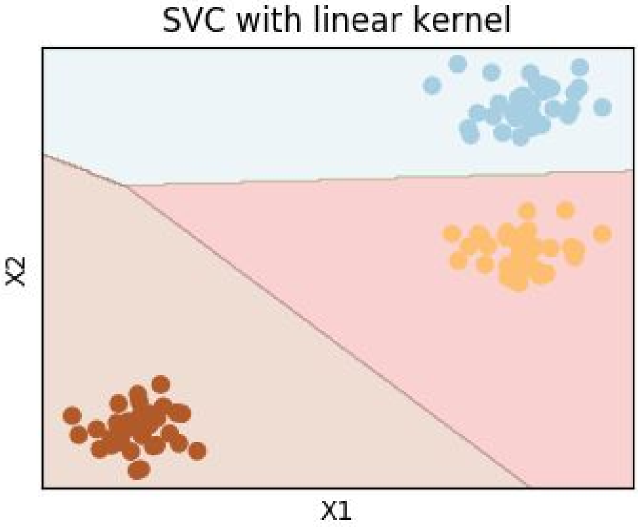
- **Data_2**

For the bean shaped data set also, RBF kernel worked well. The reason being, this kernel transforms the data into an infinite dimensional space and thus it is easy to visualize the data set.



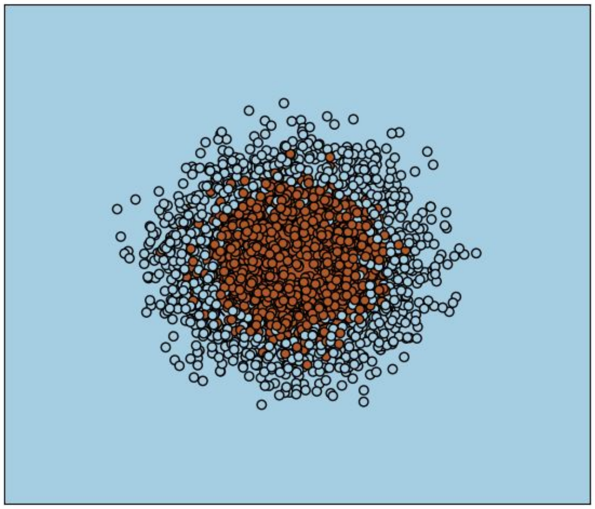
- **Data_3**

This data set consists of 3 classes which are linearly separable as we figured out while visualizing the data set. Thus linear kernel works pretty well for it.



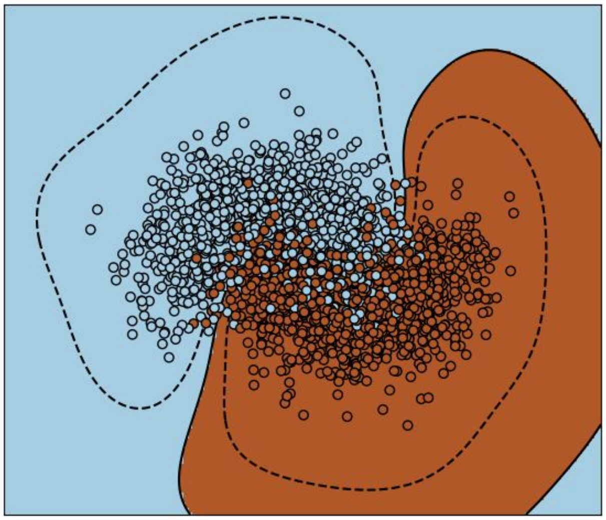
- **Data_4**

This is a large data set which is not separable even in an infinite dimensional space due to the presence of outliers. So, the decision boundaries are not visible. This looks similar to the data set 1.



- **data_5**

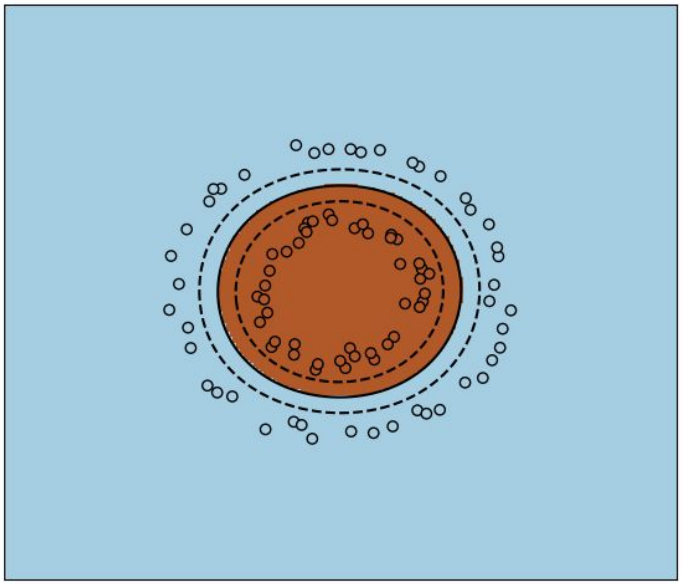
This also is a large data set which is not separable in an infinite dimensional space due to the presence of outliers. So, the decision boundaries are not visible but the shape is clearly similar to the data_2 data set.



--

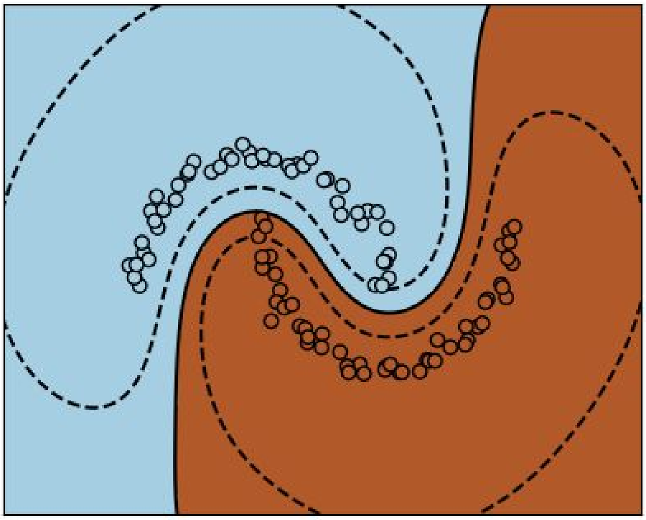
3. Outlier removed data sets.

- **Data_1**



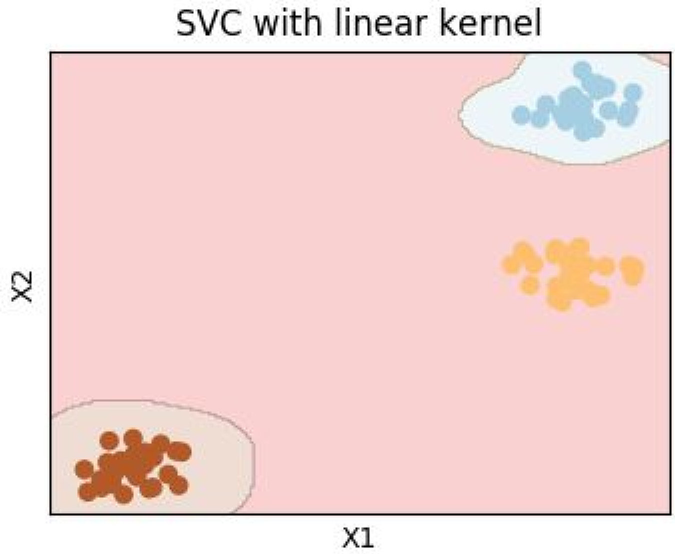
Used L1 norm distance from mean to remove the outliers.

- **Data_2**



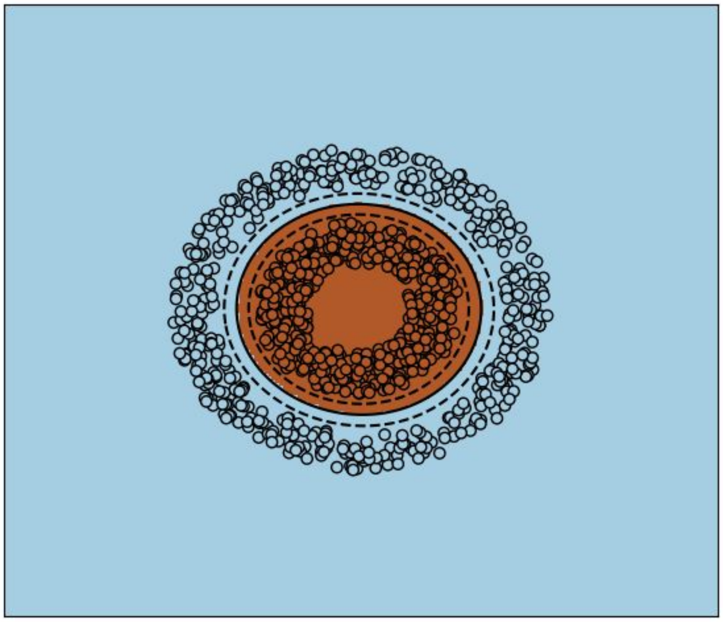
Used the density method (k nearest neighbours) to find the outliers and remove them.

- **Data_3**



Used the distance (l1 norm) from mean method to remove the outliers as the data is in the form of clusters.

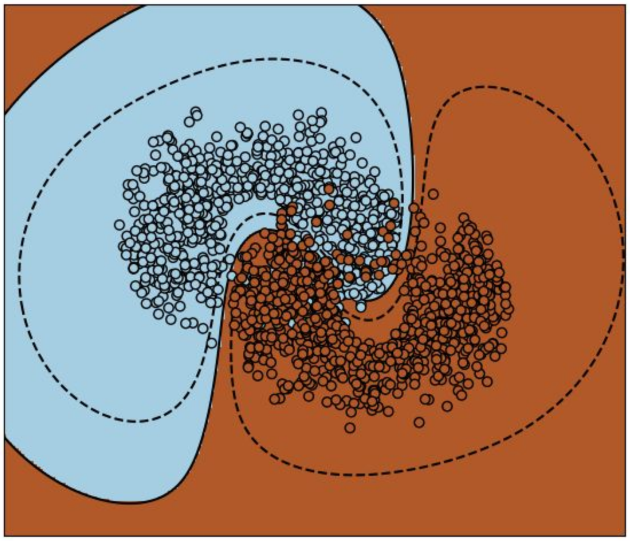
- **Data_4**



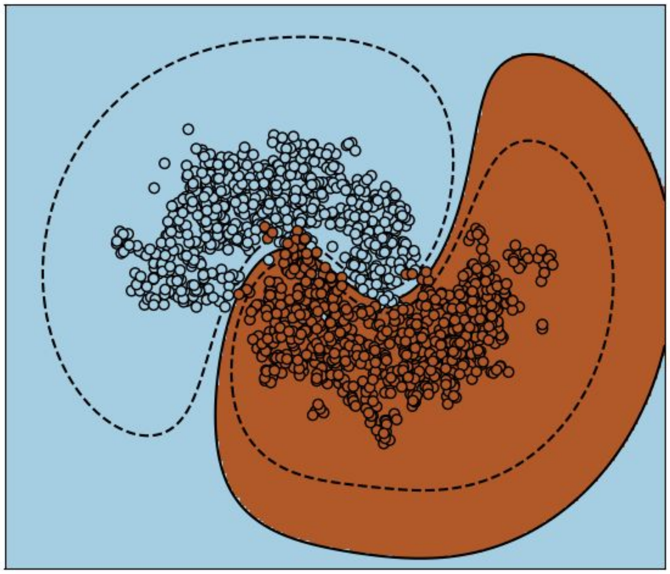
Used the same technique as used for data set 1 i.e. distance from mean.

- **Data_5**

Using simple distance from mean technique: It fails for some outliers which are actually close to mean but are not present in the dense areas of the data set.



Using the density based technique similar to DBSCAN: Considering a data point only if more than 6 neighbours are present in a distance (l1 norm) of 0.1 from it.



==

SVM

1. **Choice of hyperparameters**

OVR and OVO classifiers with 'linear' and 'rbf' kernel:

The hyper parameters I tested upon **(for linear kernel)** are: **C (penalty) and max_iters**. By default the maximum number of iterations is not defined as the iterations go on till the algorithm converges. But, we can hard code the max_iters to stop the iterations even if the solution has converged.
I tried using other parameters like shrinkage and random see etc., but they don't result in any significant difference in the accuracy. I used **grid search** to find the best parameter for all the datasets.

For RBF kernel, the hyper parameters used are: **C and gamma (kernel coefficient for rbf). Gamma = 1/n where n=no. of features when set to auto**

K-fold validation is also applied to make sure the model is trained well and it is generalized and does not overfit.

- **data_1**

There is no change in the accuracy by changing the values of the hyper parameters as the data is clearly linearly separable.

LINEAR KERNEL

C	max_iters	Accuracy
1	-1 (default)	0.390
1	100	0.390
1	1000	0.390
500	-1 (default)	0.409
500	100	0.409
500	**1000**	**0.530**

The accuracy for such a data set is quite low as the data is not linearly separable. Rbf kernel must give better accuracy for such datasets which are linearly separable in polar coordinate system.
It is clear that the large C gives better accuracy because large C means the constraints have to be taken care of which results into narrow margins but better accuracy. However, the small value of C means we are trying to implement soft margin SVM which will try to get larger margins and

would try to ignore the outliers or would lead to poorer accuracy. In this case, we get poorer accuracy because our data set does not contain any outliers as such.

RBF KERNEL:

C	Gamma	Accuracy
1	**1**	**1.0**
1	0.001	0.53
1	auto	1.0
500	1	1.0
500	0.001	0.5
500	auto	1.0

When the rbf kernel is used, we find out that the data is completely separable and hence the accuracy is 1, no matter what the parameters are. This is self explanatory from the plot shown above with decision boundary. Gamma = 0.001 gives poor accuracy and hence we should choose the parameter values carefully.

- **data_2**

There is no change in the accuracy by changing the values of the hyper parameters as the data is clearly linearly separable.

C	max_iters	Accuracy
1	-1 (default)	0.849
1	100	0.849
1	1000	0.849
500	**-1 (default)**	**0.859**
500	100	0.429
500	1000	0.63

The accuracy is better compared to the data_1 as this dataset can be linearly separated by having a margin pass through the two bean shaped classes.

As C is increased, accuracy increases from 0.849 to .859 unless the number of iterations are reduced to 100 and 1000 which are less than the iterations required for the algo to converge. This is the reason the accuracy increases as we increase the number of iterations. **This also means that as we increase the value of C, the number of iterations required for the algo to converge increases.** Large C means the constraints have to be taken care of which results into narrow margins but better accuracy. However, the small value of C means we are trying to implement soft margin SVM which will try to get larger margins and would try to ignore the outliers or would lead to poorer accuracy.

RBF KERNEL:

C	**Gamma**	**Accuracy**
1	**1**	**1.0**
1	0.001	0.859
1	auto	1.0
500	1	1.0
500	0.001	0.53
500	auto	0.65

When the rbf kernel is used, we find out that the data is completely separable and hence the accuracy is 1 when C = 1 or C=500. As gamma is decreased the accuracy falls. Even gamma = auto gives bad accuracy for C=500 (hard margin svm)

- **data_3**

C	**max_iters**	**Accuracy**
1	**-1 (default)**	**1.0**
1	100	1.0
1	1000	1.0
500	-1 (default)	1.0
500	100	1.0
500	1000	1.0

The accuracy is 1 as the data is linearly separable with large margins.
Since the data is linearly separable with large gaps for wider margins, so, changing the parameters does not change the accuracy. So, the data is separated by the SVM in 3 classes in less than 100 iterations both for C=1 and C=500.

RBF KERNEL:

C	Gamma	Accuracy
1	**1**	**1**
1	0.001	0.869
1	auto	1
500	1	1
500	0.001	1
500	auto	1

Even when the rbf kernel is used, we find out that the data is completely separable and hence the accuracy is 1, no matter what the parameters are. The only exception is gamma=0.001 which is the worst case of accuracy.
This can be seen from the plot above with the decision boundaries that separate the 3 classes completely.

- **data_4**

C	max_iters	Accuracy
1	**-1 (default)**	**0.525**
1	100	0.4315
1	1000	0.525
500	-1 (default)	0.522
500	100	0.4745
500	1000	0.4955

The accuracy is quite low for such dataset because the data is not linearly separable at all.

data_4 is quite similar to data_1 but the number of data points are way larger. The accuracy is not much compromised by having small value of C. The accuracy however decreases as the number of iterations are reduced. When C=1, the algo has converged within 1000 iterations but it is not the case when C=500.

RBF KERNEL:

C	**Gamma**	**Accuracy**
1	1	0.8824
1	0.001	0.592
1	auto	0.881
500	**1**	**0.8859**
500	0.001	0.592
500	auto	0.667

When the rbf kernel is used, we find out that the data is separable with an accuracy of 0.88. This is self explanatory from the plot shown above with decision boundary. The best parameters are C=500 and gamma=1. As the value of gamma is made 0.001 the accuracy is reduced quite a lot. But the accuracy is good at gamma = auto

- **data_5**

C	**max_iters**	**Accuracy**
1	**-1 (default)**	**0.8439**
1	100	0.7615
1	1000	0.8439
500	-1 (default)	0.8439
500	100	0.686
500	1000	0.622

data_5 is similar to data_2 except that the number of data points is very large. The accuracy is not that bad because the the data can be separated linearly to some extent. However the accuracy remains same even when C is increased to 500 meaning that the hard margin and the soft margins do not make any much difference here. However, as we increase the value of C to 500, the algo takes larger no. of iterations to converge. Therefore the accuracy is low at 100 and increases at 1000 but is more if the algo is allowed to converge.

RBF KERNEL:

C	Gamma	Accuracy
1	**1**	**0.880**
1	0.001	0.856
1	auto	0.845
500	1	0.881
500	0.001	0.458
500	auto	0.654

When the rbf kernel is used, we find out that the data is separable to a great extent and hence the accuracy is close to 0.8. However, the accuracy decreases as we decrease the value of gamma. Notice that the accuracy increases a little (by 0.001) when we increase the value of C. But there is a weird trend in the accuracy which falls when the value of gamma is 0.001 while increases as the value of gamma is made auto.

- **part_A_train**

C	max_iters	Accuracy
1	-1 (default)	0.1638
1	100	0.1395
1	1000	0.1678
500	-1 (default)	0.1638
500	100	0.1395
500	1000	0.1678

This data set gives very poor accuracy when SVM with linear kernel is run on it. The accuracy does not improve at all by increasing the value of C (using hard margin classifier) thereby proving that SVM is not a good classifier for modelling this data set.

Logistic Regression is the most suitable ML algorithm for modelling this dataset. Gaussian and Decision Trees give accuracy of around 0.6 which is less than the accuracy given by Logistic Regression i.e. 0.83

RBF KERNEL:

C	**Gamma**	**Accuracy**
1	**1**	**0.1176**
1	0.001	0.1176
1	auto	0.1176
500	1	0.1176
500	0.001	0.1176
500	auto	0.1176

Very low accuracy because the 10 classes are not separable. This shows that these classes are not separable even in very high dimensions.

- **part_B_train**

C	**max_iters**	**Accuracy**
1	-1 (default)	0.5552
500	-1 (default)	0.56

This is a huge data set which gives similar accuracy using all the modelling algorithms: Gaussian, Logistic Regression, Decision Trees, SVM. The accuracy increases a little with increase in the value of C.

RBF KERNEL:

C	Gamma	Accuracy
1	auto	0.4935
500	auto	0.4935

When the rbf kernel is used, we find out that the data is not separated well as the accuracy is 0.4. The parameter C does not play a major role here.

- **part_C_train**

C	max_iters	Accuracy
1	-1 (default)	0.1997
1	100	0.1952
1	1000	0.1994
500	-1 (default)	0.1997
500	100	0.1952
500	1000	0.1994

We can see that the accuracy is very poor for the data set C. However, the accuracy was better for the NB, Logistic Regression and Decision Trees. Each of these gave an accuracy of above 0.9.

Therefore, we can conclude that linear SVM gives very poor accuracy for these models and hence is not at all suitable for modelling the old datasets.

RBF KERNEL:

C	Gamma	Accuracy
1	1	0.49
1	**0.001**	**0.511**
1	auto	0.49
500	1	0.49

500	0.001	0.511
500	auto	0.49

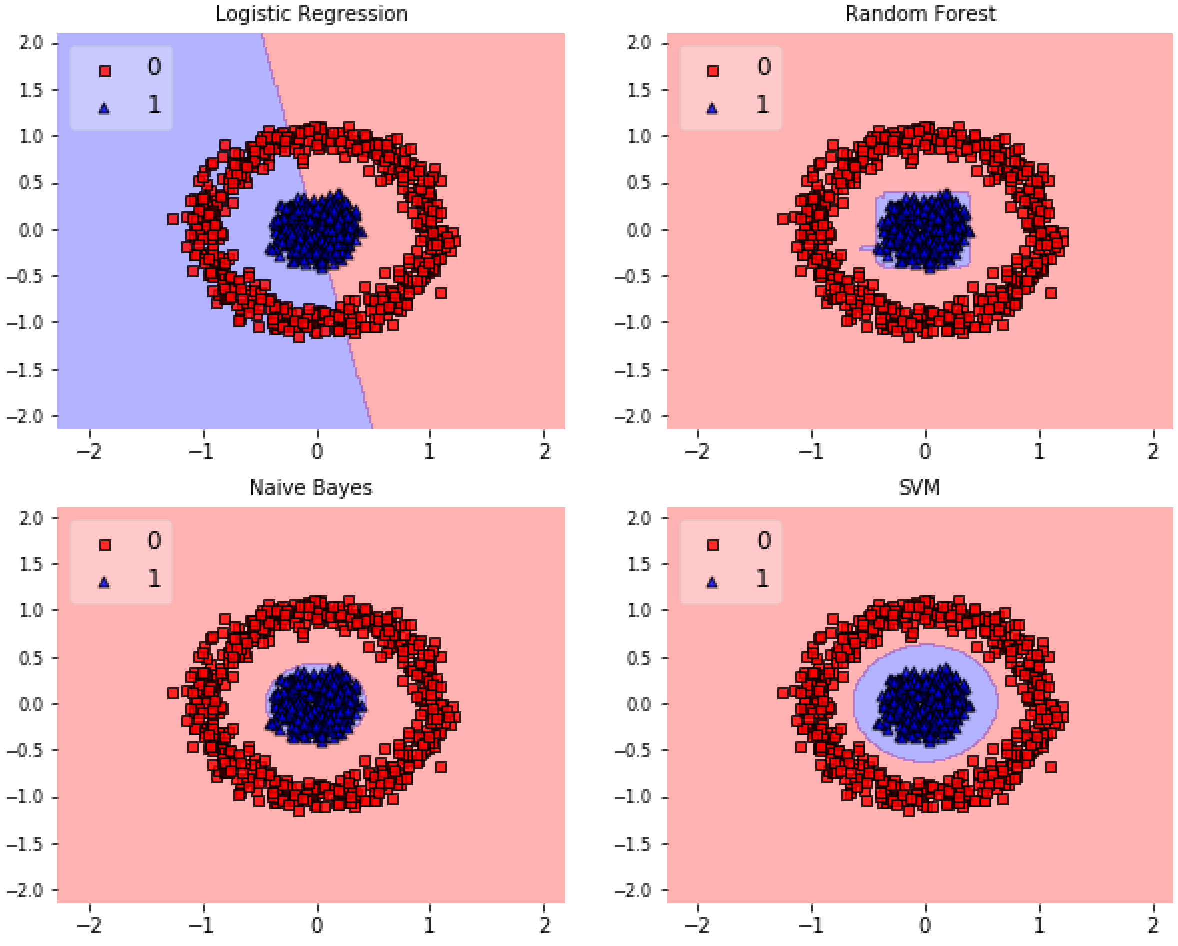
When the rbf kernel is used, we find out that the data is completely separable and hence the accuracy is 1, no matter what the parameters are. This is self explanatory from the plot shown above with decision boundary.

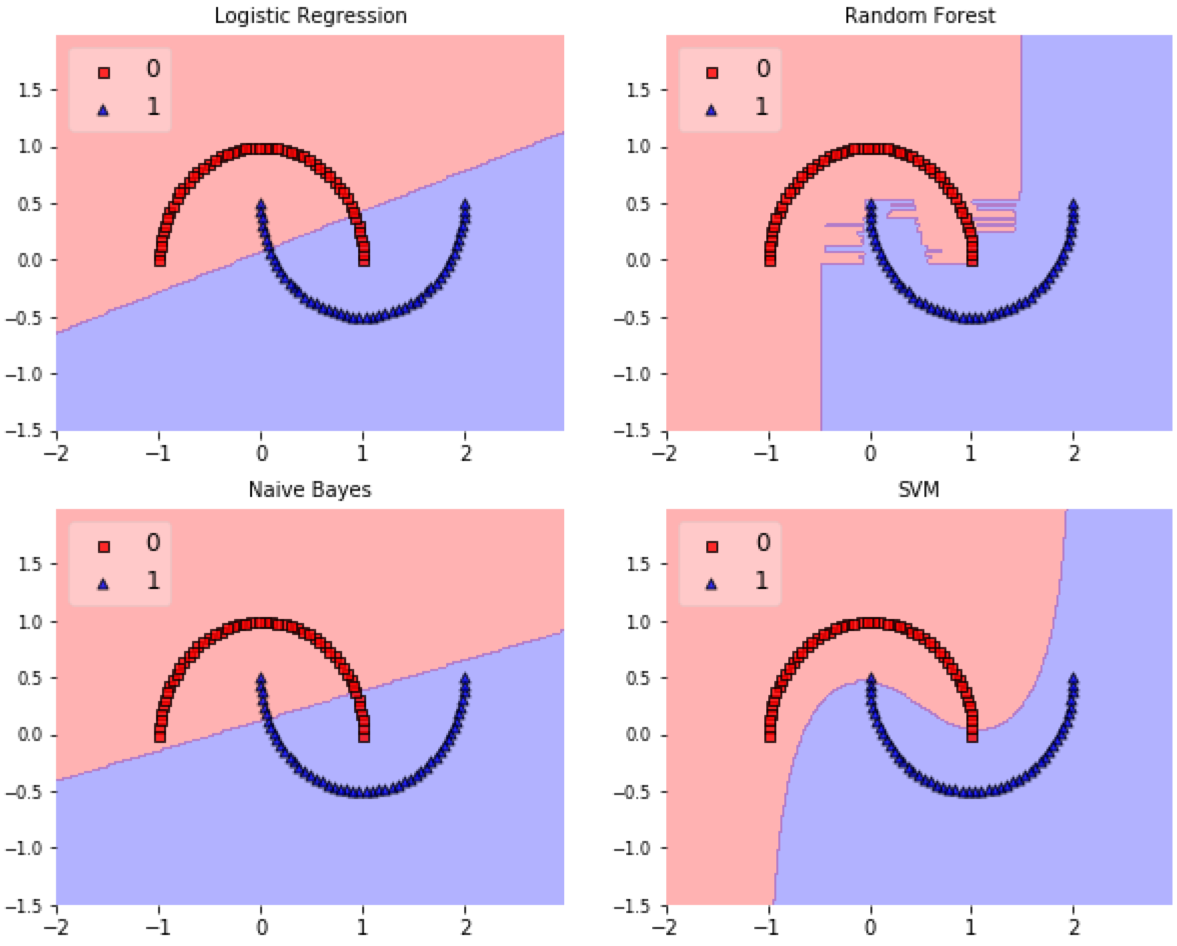
2. Comparison of models:

The comparison can be done using the accuracy achieved using SVM models on the previous assignment datasets. Refer to the plots given below the table.

	SVM	**Gaussian NB**	**Logistic Regression**	**Decision Trees**
data_1	RBF kernelized SVM works best. Linear kernel fails on this data set.	It will be able to draw a boundary but this boundary won’t be largest margin decision boundary.	Works well only for data that is linearly separable.So, won’t give give good accuracy.	It will be able to draw a boundary but this boundary won’t be largest margin decision boundary.
data_2	Linear and rbf kernels both work equally well on this mesh shaped data	It will be able to draw a linear boundary which will give little error as some data will be misclassified.	LR and NB will form a similar boundary in this case.	Again, DTs will form a boundary but it won’t be largest margin.
data_3	Linear and rbf kernels both work equally well on this data in which the classes are linearly separable.	The 3 classes can be linearly separated by NB easily.	The 3 classes can be linearly separated by LR easily.	DTs will be able to separate the data easily into 3 classes.
data_4	RBF kernelized SVM works best. Linear kernel fails on this data set.	It will be able to draw a boundary but this boundary won’t be largest margin decision boundary.	Works well only for data that is linearly separable.So, won’t give give good accuracy.	It will be able to draw a boundary but this boundary won’t be largest margin decision boundary.
data_5	Linear and rbf kernels both work equally well on this data in which the classes are linearly separable.	It will be able to draw a linear boundary which will give little error as some data will be misclassified.	LR and NB will form a similar boundary in this case.	Again, DTs will form a boundary but it won’t be largest margin.

part_A_train	SVM	Not able to classify the data well as the data is not linearly separable by NB.	LR gives somewhat better accuracy compared to Gaussian NB.	Not able to form a good model for this dataset with multiple classes.
part_B_train	None of the models gives an accuracy above 0.6 for this dataset.			
part_C_train	Linear kernel gives very very poor accuracy ~0.11. Accuracy for rbf kernel is improved but is not at par with other models.	Works very well for this data set.	Works very well for this data set.	Works very well for this data set.





Que) In which cases which models should be preferred?
Ans) When the data is linearly separable, we should prefer either logistic regression or linear SVM.
Naive Bayes model size is low and quite constant with respect to the data. The NB models cannot represent complex behavior so it won't get into overfitting. Therefore, NB is generally used when the data keeps changing and the model required is not very complex.
For the data which cannot be linearly separable, we can use decision trees or SVM with RBF or polynomial or sigmoid kernel.

Que) Which evaluation metric would you use to compare?
Ans) Confusion matrix is a good metric to compare the models. We can see the following from a confusion matrix:

- Accuracy : the proportion of the total number of predictions that were correct.
- Positive Predictive Value or Precision : the proportion of positive cases that were correctly identified.

- Negative Predictive Value : the proportion of negative cases that were correctly identified.
- Sensitivity or Recall : the proportion of actual positive cases which are correctly identified.
- Specificity : the proportion of actual negative cases which are correctly identified.

KAGGLE SUBMISSION

Data Preprocessing:

1) The redundant data is removed from the dataset.
2) Number of features are calculated and the frequency of each feature is computed.
3) The features which are occurring less number of times compared to a threshold are excluded from the dataset to reduce the dimensions of the feature space.
4) LinearSVC is used to train the model and thus accuracy is computed.