

Write a program to implement Support Vector Machine (LSVM/Kernel/Soft Margin SVM)

1) Data Description:-

A) Name :- UniversalBank.csv

B) Description :- This is a data of a bank, of the people who has applied for a creditcard and based on the situation ~~whether~~ whether they have got the creditcard or not. This dataset is available in kaggle website.

C) Attributes :- The dataset has 12 attributes -

- a) Age :- numeric data
- b) Experience :- numeric data
- c) Income :- numeric data
- d) Zip code :- numeric data
- e) Family :- numeric data
- f) CCAvg :- Continuous data
- g) Education :- Categorical data
- h) Mortgage :- Categorical data
- i) Personal loan :- Categorical data
- j) Securities Account :- Categorical data
- k) CID Account :- Categorical data
- l) Online :- Categorical data

D) Label :- The target data is CreditCard which is a categorical binary data. 0 indicates the client didn't get credit-card and 1 indicates client has got a credit card.

E) Size: The dataset consist of 5000 rows and 13 columns.

2) Model Evaluation:

A) Name :- Support Vector classifier

B) Type :- This is a classifier type supervised model which classify the data points based on hyperplane.

C) Algorithm:-

Input: $D = [X, Y]$, X is array of features, Y is label
function $\text{train_svm}(X, Y, \text{epochs})$

Step 1. Initialize learning_rate

Step 2: for learning_rate in epochs
 error = 0

Step 3. for i in X

 if $(Y[i] * X[i] * w) < 1$ then

 update $w = w + \text{learning_rate} * ((X[i] * Y[i]) * (-2 * (1/\text{epochs}) * w))$

 else

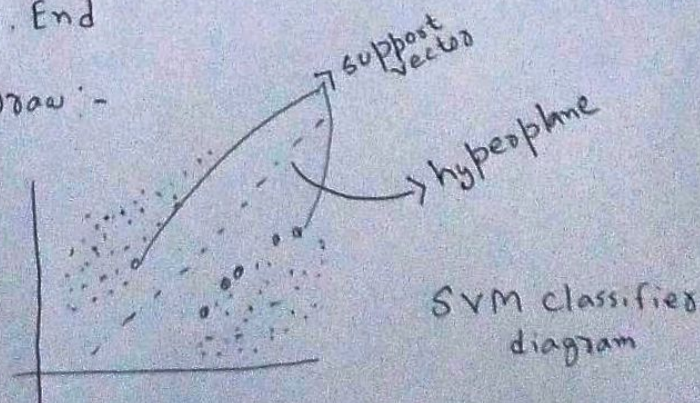
 update $w = w + \text{learning_rate} * (-2 * (1/\text{epochs}) * w)$

 endif

end

Step 4. End

D) Draw :-



3) Result Analysis:-

In this experiment I have used support vector classifier with kernel rbf and soft margin 10 as these shows the best result after cross-validation.

The accuracy of the model for this dataset is 74.5% which is moderate but if we deep dive and analyze that the model is capable for identifying '0' output very well as the precision is 0.99 and recall is 0.74 but for output '1' the model fails badly the precision is only 0.15 and recall is 0.88.

Finally by looking into F-Score for output '0' which is 0.85 and for output '1' which is 0.26 we can conclude the model fails for classifying '1' but perform well for classifying '0's.