01/11/22
Tuesday

Topic: 15

Support Vector Machine (SVM)

i) SVM :— It is a Supervised machine learning algorithm used for both classification & regression.

The Objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

Note- • (The dimensions of the hyperplane depends upon the number of features. If the number of inputs features is two, than the hyperplane is just a line.) (If the number of input feature is three, then hyperplane becomes a 2-D plane) (It becomes difficult to imagine when the number of features exceeds three.)

• The SVM algorithm has the characteristics to ignore the outliers and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

• A non-linear function that creates a new Variable is referred to as Kernel.

ii) SVM Kernel :- It is a function that takes low dimensional input space and transforms it into higher-dimensional space, i.e., it convert non-separable problems to separable problems.

[SVM Kernel are - Linear Kernel, Polynomial Kernel, Radial Basis]

- It is more mostly useful in non-linear separation problems.

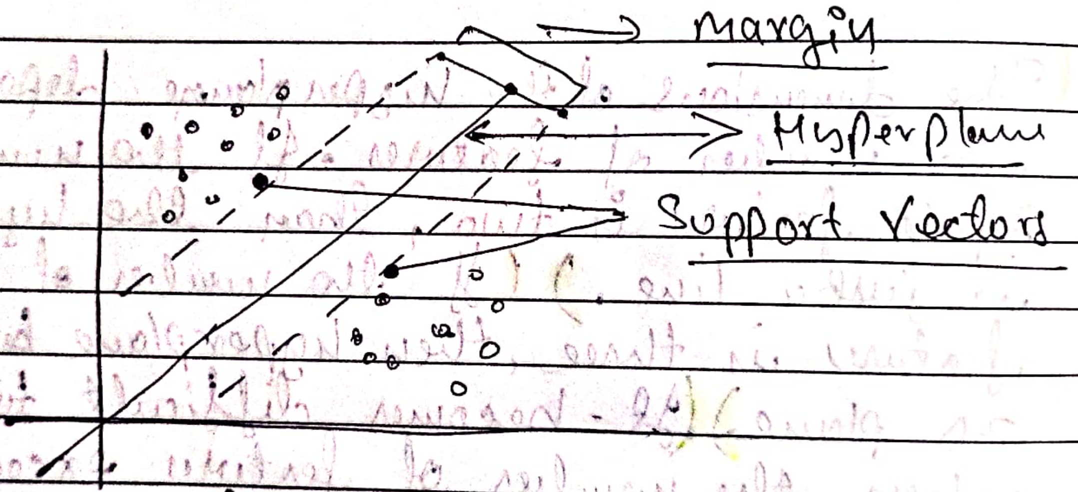


Diagram of SVM

iii) SVM use cases -
- Face detection
- Text & hypertext categorization
- Classification of images
- Bio informatics
- Remote Homology Detection
- Hand writing Detection
- Generalized Prediction Control.
- Health care.

SVM Kernel are linear Kernal, Polynomial Kernel & Radial basis function (rbf) are used model train as a parameter.

→ ξ_i (eita) is a greek symbol which means (means Submission of the distence of mis-classified point from hyper plane or marginal point)

Classifier:-

~~Cost function~~

Constraint = $y \times (w^T x_i + b) \geq 1$ ≥ 1 (T = Transpose)

$$\underset{(w,b)}{\text{Maximize}} = \frac{2}{\|w\|} \qquad \underset{(w,b)}{\text{Minimize}} = \frac{\|w\|}{2}$$

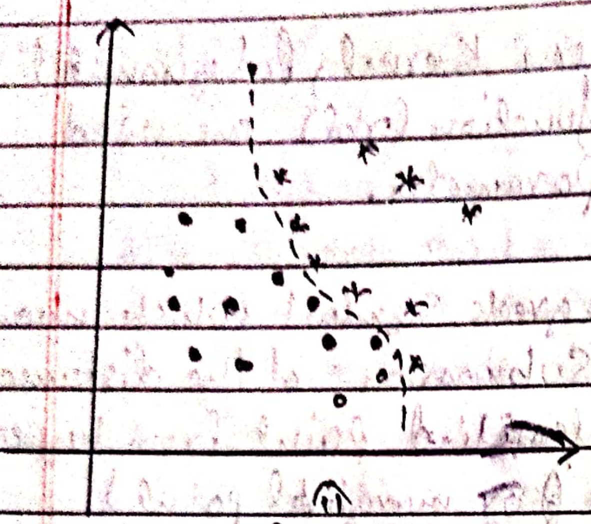
$$\text{Cost function} = \frac{\|w\|}{2} + C_i \sum_{i=1}^{w} \xi_i \text{ (eita)}$$

$C_i \sum_{i=1}^{w} |\xi_i|$ → Hinge loss / Hyper parameter

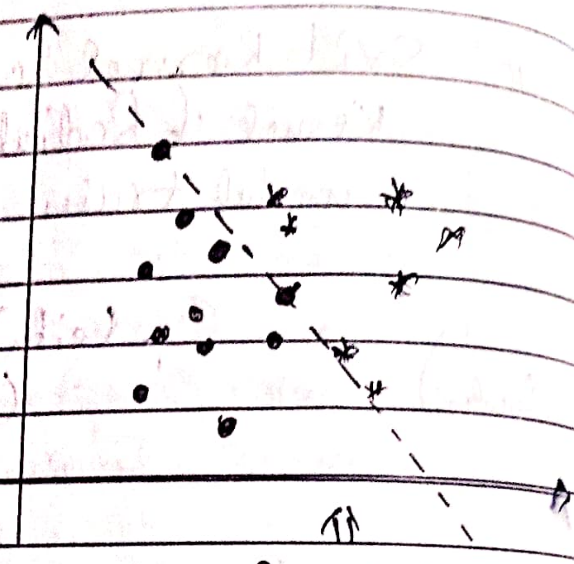
SVM

Classification   Regression

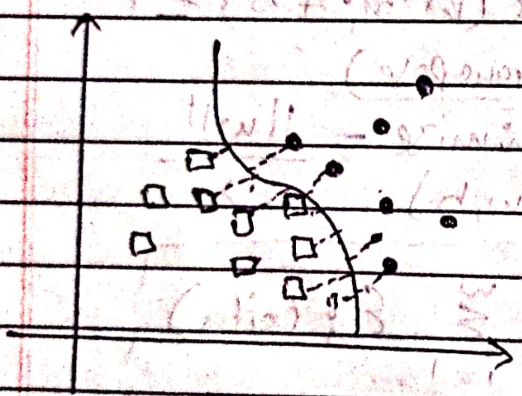
Linear & Non-linear data



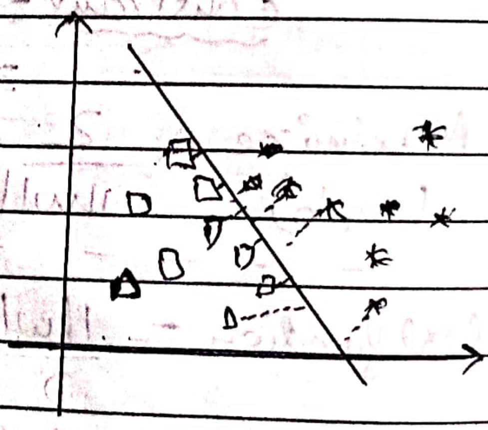
High Regularization (C)



Low Regularization (C)



High-Gamma



Low-Gamma

02/11/22
wednesday

Date
Page

Support Vector Classification

Code no ① >>> import numpy as np.
>>> import pandas as pd

Code no ② >>> from sklearn.dataset import load_breast_cancer
>>> data = load_bread_cancer ()

(∴ we have directly upload the data set from sklearn but remember it is in array & we have to convert it into Data frame)

Code no ③ >>> data.data
>>> data.features_names

Code no ④ ∴ as the dataset is in array convert in DataFrame.
>>> df = pd.Dataframe (np.c_[data.data, data.target],
columns = [list (data.features_name)+["target"]])
>>> df.head (3)

Code no ⑤ ∴ We have used Cancer dataset to predict in a model that patient has cancer or not.

>>> df.shape
(569, 31)

Date
Page

Code ⑥ >>> x = df.iloc [:, 0:-1] (# independence Variable)
>>> y = df.iloc [:, -1] (# dependence Variable)

Code ⑦ >>> from sklearn.model_selection import train_test_split
>>> x_train, x_test, y_train, y_test = train_test_split (x, y, test_size = 0.2, random_state = 2020)

⇒ ∴ Now Train Support Vector Classification Model —

Code ⑧ # import the Support vector classification Model :-
>>> from sklearn.svm import SVC

Code ⑨ ∴ train the model in one Variable

>>> classification_rbf = SVC (kernel = 'rbf')
>>> classification_rbf.fit (x_train, y_train)

∴ Hence our model train by using the (kernel = 'rbf')

Code ⑩ >>> classification_rbf.score (x_test, y_test) (# output = 0.91)
Output = 0.91

>>> classification_rbf.predict (x_test)
>>> y_test

→ to check the predict value with original output.

∴ Hence we ~~Prid~~ predict & check our Score that is only 0.91 or 91%. So, we will try feature Scaling & import Standard Scaler for more better prediction & good Score.

Feature Scaling —

Code no (11) ∴ do feature Scaling with import Standarr scaler

```
>>> from sklearn.preprocessing import StandardScaler
>>> sc = StandardScaler ()
```

Code no (12)
```
>>> sc.fit (x_train)
```
~~the analyze x_train these can use y_train & ...~~

Code no (13)
```
>>> x_train_sc = sc.transform (x_train)
>>> x_test_sc = sc.transform (x_test)
```

∴ we need transformation in Standard Scaler as we already split it into code no (7)

Code no (14)

Note :- Remember if we already ~~model~~ fit our model and after that we want to used Standar scaler we have to change Variable of x_test & x_train as we done in code no (13) but y_train & y_test will be remain Same. (not-nedded)

Coding (14) ∴ Now train the model

```
>>> classification_rbf_2 = SVC(kernel = 'rbf')
>>> classification_rbf_2.fit(x_train_sc, y_train)
```

Coding (15) # Check the Score

```
>>> classification_rbf_2.score(x_test_sc, y_test)
output = 0.95
```

∴ Hence we get the score coding (10) 0.91 before using Standard Scaler but after using it we get score of 0.95 in coding (15).

∴ So, we should use Standard Scaler sometime or according to data set or score.

- - - - - - - - - - - - - - - - - - - -

Now,
we will use different Kernel like polynomial & linear after the Standard Scaler used & we will also check the score & whose score is good enough to uses

∴ x_train_sc

Date ______
Page ______

— Using different Kernel —

Polynomial Kernel

Code no (16)
```
>>> classification_poly = SVC (Kernel = 'poly', degree = 2)
>>> classification_poly.fit (x_train_sc, y_train)
>>>
>>> classification_poly.score(x_test_sc, y_test)
```

~~see~~ output — Score = 0.83 (# very less)

Linear Kernel

Code no (17)
```
>>> classification_linear = SVC (Kernel = 'linear')
>>> classification_linear.fit (x_train_sc, y_train)
>>> classification_linear.score (x_test_sc, y_test)
```

Output — 0.96 (# it is the Higest score)

∴ SVM classification is mostly used than SVM Regression. And when we have to find True or false, Black or white, Yes/No then we use classification but when we have to find value eg:- predict the value of building, weight then we used Regression

Date
Page

03/11/22
Thursday

Support Vector Regression –

Code no ① >>> import pandas as pd
>>> import numpy as np

Code no ② >>> df = pd.read_csv(" Bangalore price")
>>> df.head ( )
>>> df.Shape
output = (7120, 108)

Split the dataset –

Code no ③ >>> x = df.drop ('price', axis = 1) # (independence Variable)
>>> y = df ['price'] # dependence variable

Code no ④ >>> from sklearn.model_selection import train_test_split
>>> x_train, x_test, y_train, y_test = train_test_split
(x, y, test_size = 0.2, random_state = 51)

Feature Scaling –

Code no ⑤ :- we are doing feature Scaling i.e, Standard Scaler before we train our data set eos

>>> from sklearn.preprocessing import Standard Scaler
>>> sc = Standard Scaler ( )
>>> sc.fit (x_train)

Date
Page

```
>>> x_train = sc.transform (x_train)
>>> x_test = sc.transform (x_test)
```

Support Vector Regression - ML Model Training

Coding (6)

```
>>> from sklearn.svm import SVR
>>> svr_rbf = SVR (kernel = 'rbf')
>>> svr_rbf.fit (x_train, y_train)

# to check the score
>>> svr_rbf.score (x_test, y_test)
#output - 0.20
```

∴ Hence the output is very less so we will change the kernel to 'linear' & we train it & check the score.
so output will be in
<u>0.79</u>

Predict the Value -

Coding (7)

```
>>> y_pred = svr_linear.predict (x_test)
                              (# prediction value)
>>> y_pred

                              (# original value)
>>> y_test
```

∴ from here compute both and check the difference.

Contd (8) Now using 'rmse' to find the

```
>>> from sklearn.metrics import mean_squared_error
>>> import numpy as np

>>> mse = mean_squared_error (y_test, y_pred)
>>> rmse = np. sqrt (mse)

>>> print ("mse:", mse)
>>> print ("rmse", rmse)
```

Output = MSE = 4093.451
RMSE = 63.98

Date ______
Page ______

Topic-16 **MSE & RMSE**

- in Machine learning the error is also Known as cost function.

Mean Square error = MSE = $\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}$

Root Mean Square error = RMSE = $\sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$

where,

$\hat{y}_i$ = predict value

y_i = actual value

n = no. of data point / Sample.

- Suppose one of my frd predict my value that is 40kg but after checking it or real actual value is 50kg.

$\hat{y}_i = 40$, $y_i = 50$

MSE — It measure the amount of error in statistical model. It calculated the average squared difference between the observed & predicted value.

SVM

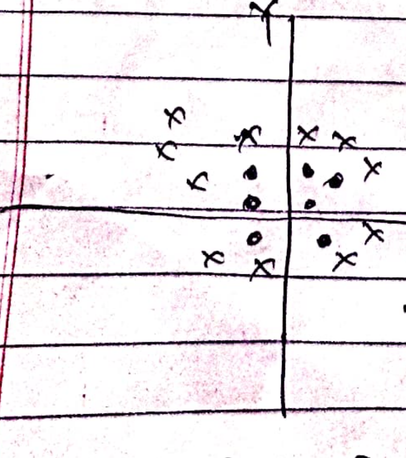
1. SVM – Support Vector Machine is a Supervised Machine learning algorithm that can be used for both classification or Regression problem.

⇒ But, it is mostly used in classification problem, such as text classification.

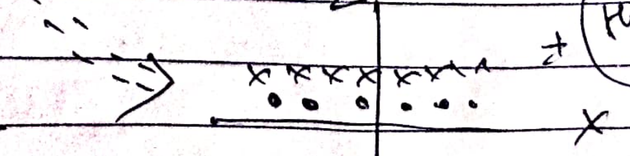
⇒ In the SVM we plot each data item as a point in n-dimension space where n is the number of feature we have.

The SVM algo. has a feature to ignore outliers and find the hyper-plane that has the maximum-margin. Hence we can say SVM classification is robust to outliers.

Q. How SVM hyper-plane segregate to this scenario



→ SVM will solve this problem easily by additional new feature where $z = x^2 + y^2$ and it will generate new scenario



(This is called Kernel trick)

$z = x^2 + y^2$

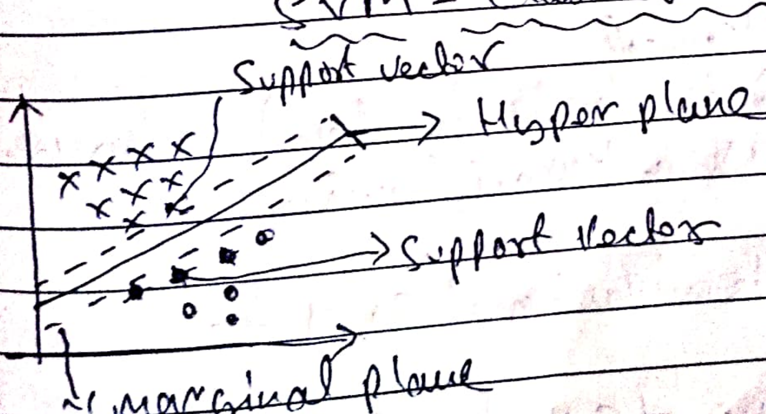
2. Kernel trick = Whatever we have done in Q, it is a Kernel trick
The SVM Kernel is a function that take low dimensional input space & transform it to a higher dimension space.

Q2 What is the use of Kernel in the SVM algorithm?

Ans Kernel can be used in SVM to transform the data to the higher dimension, to find the optimal hyperplane.

SVM — Classification
— Regression

SVM - classification (SVC)



Constrain of SVC/Support Vector Classifier = $y_i(w^T x_i + b) \geq 1$

SVC (Support Vector Classification)

Note → The maximum distance of marginal plane is consider to be best fit line or Hyper plane.

$$\text{Cost function of SVC} = \frac{\|w\|}{2} + C_i \sum_{i=1}^{n} \underbrace{\xi_i}_{\text{Sumation of the distance of the misclassified point from marginal plane}}$$

eta

Note → when it is almost difficult to separate non-linear classes, we then apply another trick called "Kernel Trick" that help handle the data.

3 types of Kernel function are -

i) Polynomial Kernal
ii) RBF (Radial basis function) also Know as Gaussian Kernal function.

3 — advantage & disadvantage of SVM —

advantage are - (SVM)

i) They are flexible in unstructured, structured and Semi structured data.

ii) Kernel function eases the complexities in almost any data type.
iii) Overfitting is less observed compared to other model.

- Disadvantages — (SVM)
i) Training time is more while computing large dataset.
ii) Hyperparameter are often challenging.
iii) Overall interpretation is difficult because of some black box approaches.

- Popular application of SVM —
i) Health care, | iii) Social Networking
ii) Banking

Note :— Kernel trick fn function is used when we can't draw a hyperplane separation. See in Q1 diagram.

Q) When to use Kernel trick.

Ans) Kernel trick function is used when we can't not make or draw separation or the hyperplane. See in Q1 diagram