

Structure of Code:

- The code is designed using Python and contains a function for classification problem
- First, the input file "iris.data" is read as a dataframe using numpy library.
- Then the three iris species names are changed to '0','1','2' labels in float format and the entire data is manually split into training and testing data(80/20)
- With training data as X and training labels as Y we find Beta cap $\hat{\beta}$ using linear regression model formula,

$$\hat{\beta} = (A^T A)^{-1} A^T Y$$

- Now, using the estimator values we classify the model and also predict label values over the testing data,

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots w_d x_d = w_0 + \sum_{j=1}^d w_j x_j$$

- In order to cross validate we fit the data again into predefined regression model and Compute the predicted values from N-fold cross validation and pass it in regression score function to get the accuracy of predicted output which comes out to be 1.0 mostly.
- Since there is no discrepancy in no. of folds, the N value is chosen to be **10** which is most widely used as the more folds we have, we will be reducing the error due to the bias but increasing the error due to variance which nullifies each other.

Screenshot of output:

The screenshot shows the Spyder Python IDE with a file named 'my_iris.py' open. The code in the editor performs the following steps:

- Imports necessary libraries (numpy, pandas, sklearn).
- Loads the 'iris.data' file into a DataFrame.
- Converts the species names ('setosa', 'versicolour', 'virginica') to numerical labels (0, 1, 2).
- Splits the data into training (80%) and testing (20%) sets.
- Calculates the Beta cap $\hat{\beta}$ using the formula $\hat{\beta} = (A^T A)^{-1} A^T Y$.
- Predicts the labels for the testing data.
- Calculates the accuracy of the predictions.
- Performs 10-fold cross-validation and prints the scores.

The Variable explorer on the right shows the following variables:

Name	Type	Size	Value
s	int	1	0
scores	float64	(120,)	Min: -4.44892098500626e-16 Max: 2.0000000000000004
split_train	int	1	120
t1	float64	(5, 5)	Min: -0.18553579265087347 Max: 1.114796993614246
t2	float64	(5, 120)	Min: -0.2617944122802749 Max: 0.25757435562884896
test	float64	(30, 5)	Min: 1.0 Max: 7.9
testLabel	float64	(30,)	Min: 1.0 Max: 1.0
test_index	int32	(60,)	Min: 0 Max: 59
train	float64	(120, 5)	Min: 0.1 Max: 7.7

The console output shows the following results:

```

[-0.04558273 -0.63329621 0.52108015 -0.45601488 1.91854987]
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
Cross-validated scores: 1.0
In [49]: runfile('C:/Users/Balaji/Desktop/my_iris.py', wdir='C:/Users/Balaji/Desktop')
[-0.04558273 -0.63329621 0.52108015 -0.45601488 1.91854987]
2 -fold Cross-validation scores: 1.0
3 -fold Cross-validation scores: 1.0
4 -fold Cross-validation scores: 1.0
5 -fold Cross-validation scores: 1.0
6 -fold Cross-validation scores: 1.0
7 -fold Cross-validation scores: 1.0
8 -fold Cross-validation scores: 1.0
9 -fold Cross-validation scores: 1.0
In [50]:

```