Name: Rhea Sharma
ID: 1001679252

# CSE 6363: Machine Learning

## Project 1 - Report

**About the dataset:**

The dataset being used is the **Iris Flower Dataset**, which is available for download for free at https://archive.ics.uci.edu/ml/datasets/iris. This dataset is one of the best known datasets used for statistical classification techniques.

The dataset consists of 50 samples from 3 species of Iris, namely: *Iris virginica*, *Iris setosa* and *Iris versicolor*. There are a total of 150 records in the dataset under 5 attributes. The attributes measured and collected from each species are the sepal length, sepal width, petal length, petal width in centimeters, and the class of the species.

**Method:**

The regression model being used is **Linear Regression**.

"*Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.*"

A linear regression line has an equation of the form:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

This model has been implemented in Python 3, which classifies the dataset into one of the three species. I have made use of the K-Folds Cross Validation technique in order to split the data into training and testing data.

First, I have created the design matrix, X, from the attributes of the dataset. I have also created the result matrix, Y, which consists of unique classes. These matrices were used to calculate the coefficient matrix B for the whole dataset. Further, the matrix B helps in fitting unseen data.

$$\mathbf{x} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$$

For cross validation, I have used the K-Folds Cross Validation technique, which depends heavily on the value of "K". The implementation of this technique is done in an iterative manner such that every fold has its own design matrix, result matrix and beta matrix. I am also calculating the error matrix, which consists of the difference between the actual and predicted values in every fold/iteration. This error matrix is used to calculate the average error of the model and further the accuracy of the same.

```
#Implementation of K-Fold Cross Validation
for i in range(0, len(K_Fold_X)):
    for j in range(0,len(K_Fold_X)):
        if(j == i):
            K_Fold_X_Test.append(K_Fold_X[j])
            K_Fold_Y_Test.append(K_Fold_Y[j])
        else:
            K_Fold_X_Train.append(K_Fold_X[j])
            K_Fold_Y_Train.append(K_Fold_Y[j])
```

**Note:**

Python Libraries used:

- Pandas
- NumPy

**Results:**

I ran the model for five different values of K to perform K-Fold Cross Validation and the accuracy for each of those values are as follows:

| Value of K | Accuracy |
|---|---|
| 2 | 49.33336% |
| 3 | 35.33336% |
| 5 | 92.66667% |
| 6 | 94.66667% |
| 10 | 95.33334% |

For this model, I think the best value for K would be '5' since it provides an accuracy of 92.66667%. This also has a good amount of equalisation between the training and the testing bins. If higher values of K are considered, since the dataset is not that large, the training and testing data would have very few values and hence the model would not be that accurate. Therefore, K = 5 is the optimal value amongst the pool of values for K.

**Conclusion:**

In this project I have implemented linear regression with cross validation using K-Fold Cross Validation technique and also fitted unseen data using the trained model.

**References:**

- https://archive.ics.uci.edu/ml/datasets/iris
- NumPy: https://numpy.org
- Pandas: https://pandas.pydata.org
- Python: https://www.python.org
- Stackoverflow
- http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm
- https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/13/lecture-13.pdf