**ANALYSIS:**

* + multi-class classification using logistic regression:
    - The problem set we will work with is the famous iris dataset imported from the scikit learn library. After all the necessary imports, the design matrix is populated in the form of a dataframe consisting of the predictor data. The target data is stored in a separate dataframe.
    - A bit of feature engineering and data preparation is done by using the ‘split-apply-combine’ technique to assign the species names to the target set, which originally had it in the form of categorical variables.
    - A new dataframe that combines the predictor and target dataframes is created for initial exploratory analysis and data visualization.
    - The ‘pairplot’ function in seaborn is used to get a comprehensive visualization of the various feature dependencies and their distributions with the species set as hue.
    - Histograms, scatterplots, factorplots are used for further visualizations. it was interpreted that the setosa species in almost all respects show distinguishable characteristics , while the virginica and versicolor have similar characteristics.
    - Once the data is prepared, it is run through an algorithm that is similar to the binary classification using the logistic regression. Here, a technique called ‘one vs all’ is used, where a class is chosen from the class set and the usual logistic regression algorithm is run with the rest of the classes abstracted to the complimenting binary class. This is done for every class and the one with the maximum probability is the prediction.
  + Multi-class classification using K Nearest Neighbour algorithm:
    - We use the same training and testing set for this algorithm.
    - The K nearest Neighbour algorithm decides which class a particular instance of a feature set should fall under based on the classes of the neighbouring instances. Neighbours are defined as those instances that are near the instance of interest, which is decided using a distance metric based on the features. By affixing a user defined value ‘k’, the neighbours of the given problem are the ‘k’ instances nearest to the problem instance.
    - We know the classes of these neighbour instances as they are part of the training set, and among this class-set, the class most frequent is heuristically fixed as the most probable class of the problem instance .
    - The usual fit, predict and measure the score algorithm is carried out.
    - For a k value of 4 The score turns out to be 95%
    - Since k is determined by us, on finding the trend of how the value of k affcts the accuracy, the corresponding k versus accuracy plot gives a random rendering that is highly stochastically dependent on the dataset, the train test split etc.
    - It is usually advisable to go for a lukewarmly low value of k.