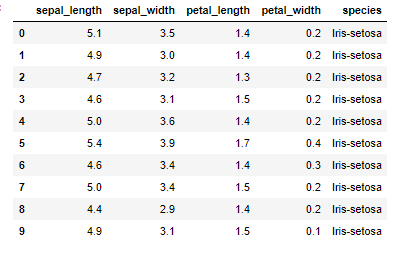
### **Classify Iris Plants into three species - KNN from Scratch**

**Purpose**

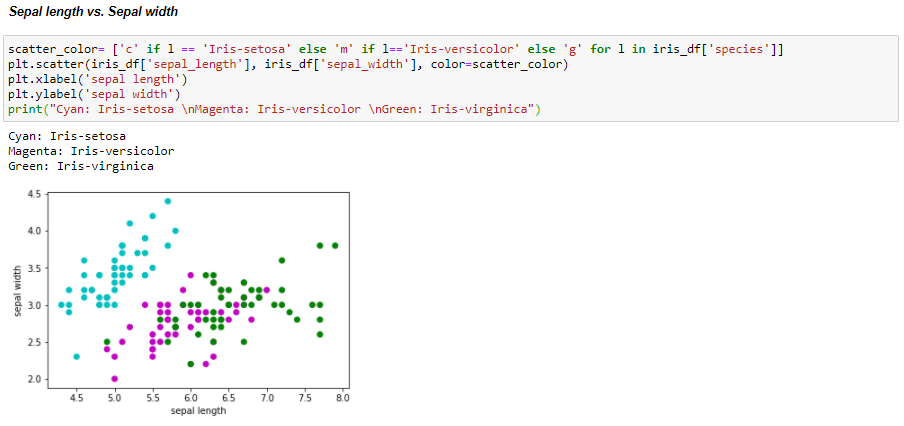
KNN algorithm makes predictions by calculating the similarity between the input sample and each training instance. This algorithm does not make strong assumptions about the form of mapping function hence it is Nonparametric. In simple words, by not making assumptions, the algorithm is free to learn any functional form from the training data. In K-NN algorithm output is a class membership. An object is assigned a class which is most common among its K nearest neighbours, K being the number of neighbours. Intuitively K is always a positive integer.

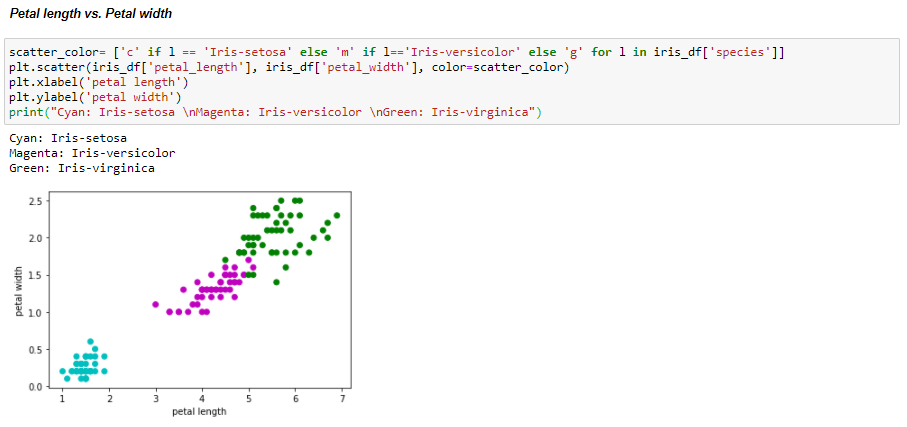
The main purpose of this project is to classify the iris plants into 3 kinds of species i.e., Iris-setosa, Iris-versicolor, Iris-virginica. These plants are classified on the basis of 4 features i.e., Sepal Length, Sepal Width, Petal Length, Petal Width.

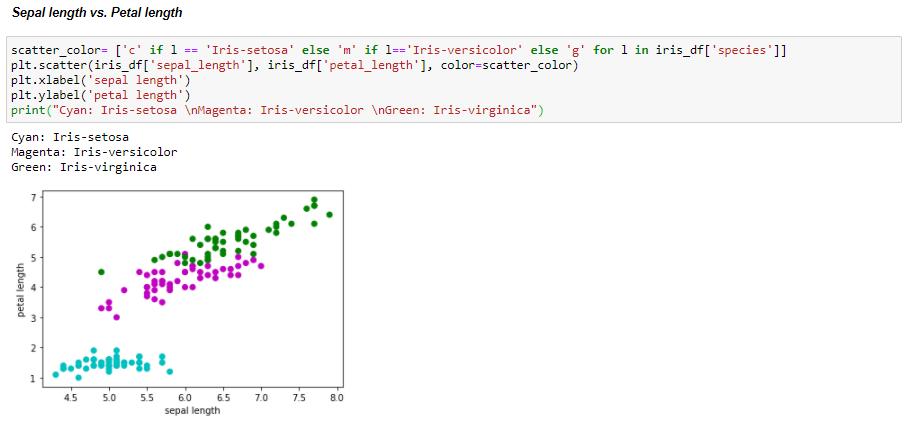
First few data points of the dataset are:

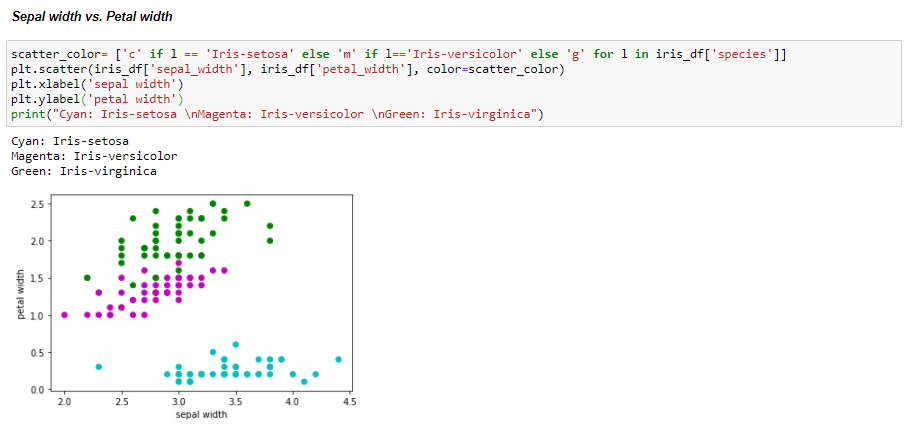


**Hypothesis**









**Algorithm used**

The classification in this project is implemented using the K Neighbours Classification algorithm.

The project is first implemented using the Sci-kit Learn libraries, and then without using the Sci-kit Learn libraries. The reason for choosing this algorithm is because the KNN algorithm is extremely easy to implement and it does not require any training to make real-time predictions. One of the useful benefits of the KNN algorithm is that there are only two parameters required to implement KNN i.e. the value of K and the distance function.

**Implementation and Results**

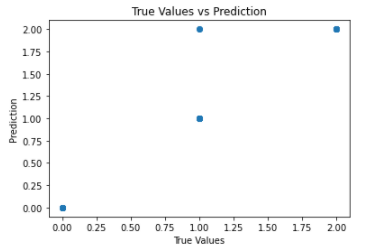
1. **Using Sci-kit Learn libraries**

The iris dataset consists of 4 features and 1 label with 3 classes. The features and labels are separated with features stored in x and labels in y. All the features are of float data type and labels are categorical, so there is a need to encode the labels as 0, 1, 2 representing the 3 classes - Iris-setosa, Iris-versicolor, Iris-virginica. These labels are encoded using the Label Encoder of Sci-kit Learn library.

Next, the dataset is splitted into train set and test set. The features are split into train set features and test set features. The labels are split into train set labels and test set labels.

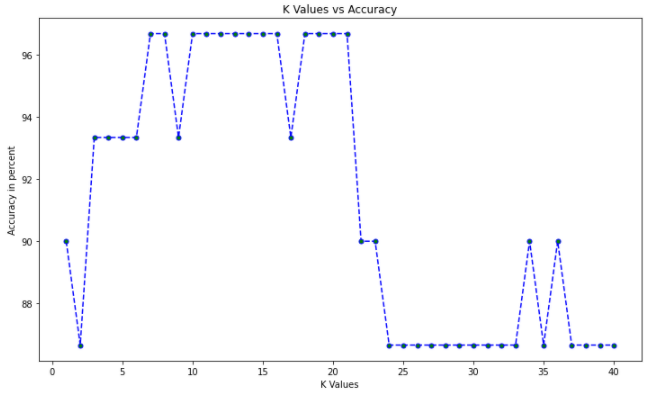
The KNeighborsClassifier method of the KNeighborsClassifier library is used to classify the test set into the 3 labels given that there will be 3 nearest neighbors into consideration and the classes for the test set are predicted.

When the true values are plotted against the prediction of the test set, the graph shows:

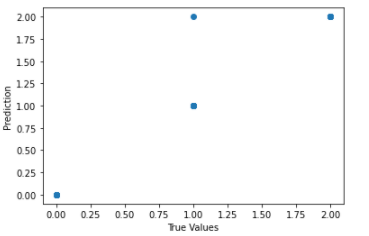


The model gives 93.33% accuracy. The confusion matrix and the classification report was calculated using the Sci-kit Learn libraries.

Next, we tried to predict the class of the plants considering the first 40 different nearest neighbors and found the accuracy score each time. The plot for K values vs. Accuracy:



It was observed that a K value equal to 7 gives a better accuracy than that equal to 3. So the same process was again repeated with K value equal to 7, which gives 96.67% accuracy. When the true values are plotted against the prediction of the test set, the graph shows:

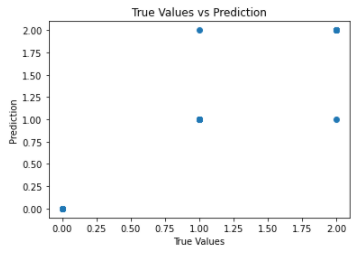


1. **From scratch (Without using Sci-kit Learn libraries)**

The same process of classifying the class of the plant species was repeated, however without the sci-kit Learn libraries.

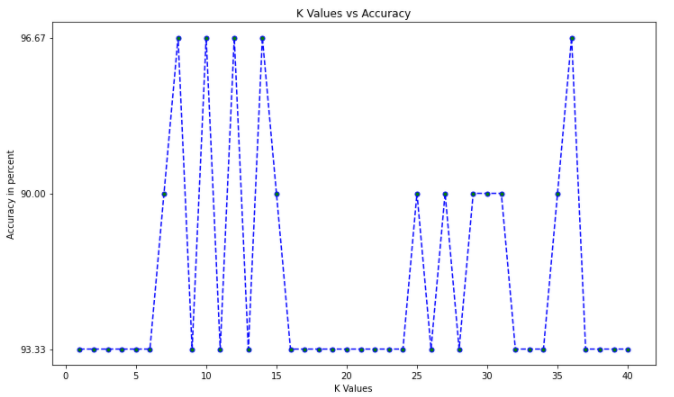
The test set was classified into the 3 labels given that there will be 3 nearest neighbors into consideration.

When the true values are plotted against the prediction of the test set, the graph shows:



The model gives 93.33% accuracy. The confusion matrix, precision and recall values were calculated.

Next, we tried to predict the class of the plants considering the first 40 different nearest neighbors and found the accuracy score each time. The plot for K values vs. Accuracy:



It was observed that a K value equal to 8 gives a better accuracy than that equal to 3.

**Conclusion**

KNN is a simple yet powerful classification algorithm. It requires no training for making predictions, which is typically one of the most difficult parts of a machine learning algorithm. However, it shows a long execution time.The KNN algorithm have been widely used to find document similarity and pattern recognition. It has also been employed for developing recommender systems and for dimensionality reduction and pre-processing steps for computer vision, particularly face recognition tasks.