Introduction :

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by biologist Ronald Fisher in his 1936 paper. This dataset contains samples of 3 species of iris flowers. Four key features – sepallength, sepalwidth, petallength, petalwidth were measured for the sample in the dataset. In this report we will use this dataset to apply supervised (K-nearest neighbors) and unsupervised learning techniques (K-means clustering and hierarchical clustering)

# Exploratory Data Analysis:

1. **Summary :**

The dataset contains 150 observations in total, of which 50 are from each species of Iris (*Iris setosa*, *Iris virginica* and *Iris versicolor*).

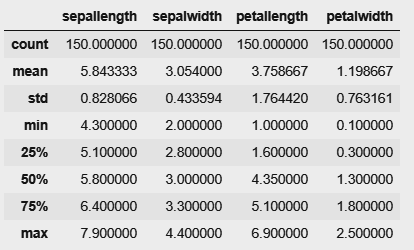


Table 1: Summary statistics of the variables in Iris dataset

From the summary table above, we can see that there are no major abnormal values in the variable columns and the values are within range for each column.

**BoxPlots :**

To check for outliers in the dataset, we plot box plots for each variable –

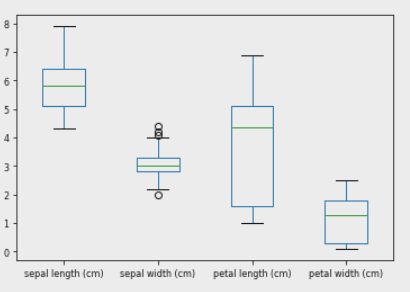


Figure 1: Box plots of the features

From the above box plots –

* We notice few outliers present in ‘sepal width’ column and no outliers in the other columns
* The mean/ median of ‘Petal length’ seem to be slightly on the higher end indicating that more number of observations have higher values

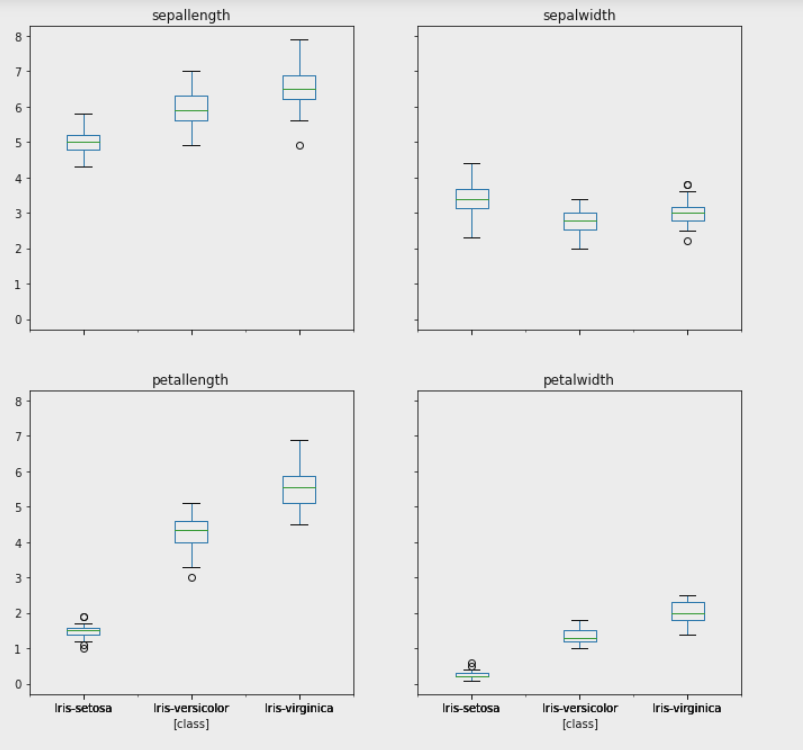


Figure 2: Box plots of the features by species

* In the feature petal length, we notice that setosa and versicolor has outliers
* The petal width of setosa has another outlier present in the sample
* The ‘Sepal width’ and ‘Sepal length’ of Virginica species has a few outliers

1. **Distributions :**

To understand the distribution of the variables further, we plot **histograms and density curves**. First we take the feature ‘sepallength’ and understand its distribution.

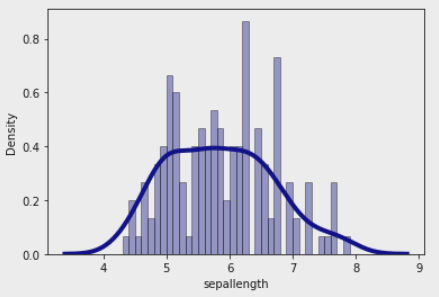
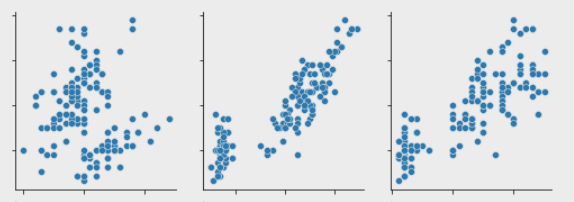


Figure 3: Histogram and density curve for ‘sepallength’

We observe that the distribution of ‘sepal length’ is not skewed and is slightly uniform. The other variables ‘Sepal width’ is normally distributed and the distribution of ‘ petal width’ and ‘petal length’ have a similar uneven distribution of the data.

**ScatterPlots –**

Here we plotted scatterplots wanted to learn the distribution of ‘sepal length’ with respect to other variables like ‘sepal width’, ‘petal length’ and ‘petal width’



Sepal width

Petal width

Petal length

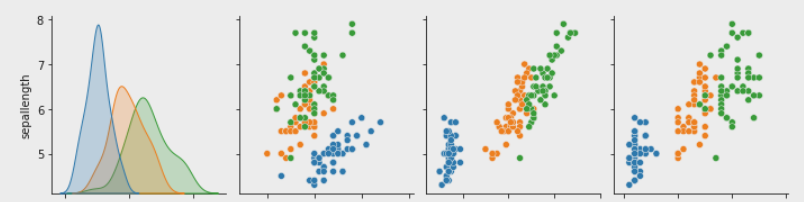
Sepal length

Figure 4: Scatter plots of ‘sepal length’

From the above scatterplots, we observe that ‘sepal length’ has some linear relationship with ‘petal length’ and ‘petal width’. The distribution of ‘sepal length’ and ‘sepal width’ seem to be distributed across.

**Distributions by species :**

To get a deeper understanding, we looked at the distribution of ‘sepal length’ by the 3 type of iris species, and got the below distribution



Petal width

Petal length

Sepal width



Figure 5: Distribution of ‘sepal length’ by species

We observe some clear patterns from the plots above –

* The ‘sepal length’ of the setosa species has a lower range when compared to other species, versicolor and virginica. The distribution of sepallength of ‘virginica’ is slightly right skewed
* Sepal length vs Sepal Width – Most of the sample from ‘Setosa’ seem to have large ‘Sepal width’ and lower ‘Sepal length’ when compared to other species
* Sepal length vs Petal length and Petal width - The distributions of the species appear to be similar in ‘Petal length’ and ‘Petal width’. When compared to other species ‘Setosa’ seem to have lower sepal length and lower Petal lengths and petal widths.

**Correlation Plots –**

We check for the correlation values to understand the linear relationship between the features

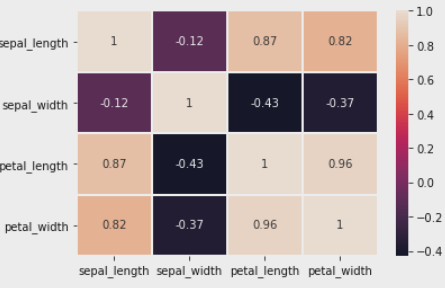


Figure 6: Correlation values between the features

* We observe that there is a high correlation (value of 0.96) between ‘petal width’ and ‘petal length’
* Sepal length is also has strong linear relationship with ‘petal width’ and ‘petal length’ and is in line with our observations from the scatter plots

**Basic statistics by species :**

We further compute the statistics by species to find any differences among the species –

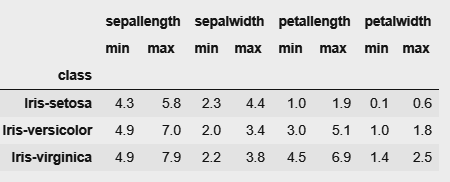


Table 2: Minimum and Maximum values of the features by species

Observations –

* Though the minimum of ‘sepal length’ is comparable among the species, the max of setosa is lower (5.8) than that of other species (7, 7.9)
* The maximum ‘sepal width’ of Setosa is higher among the other species and their minimum is comparable
* Among all the species, Virginica has the highest maximum ’sepal length’, ‘petal length’ and ‘petal width’ and slightly higher min

**Mean and Median by species -**

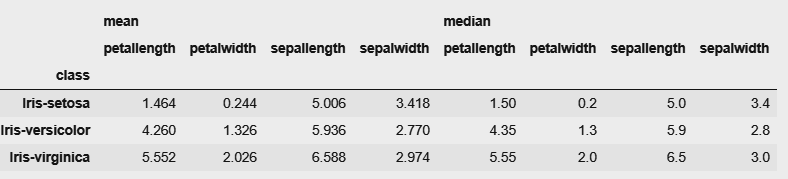


Table 3: Mean and median of the features by species

* Setosa has lower mean/ median ‘petallength’ and ‘petalwidth’ and slightly higher ‘sepal width’ as compared to other species
* Virginica has a higher mean/ median ‘sepal length’, ‘petal length’ and ‘petal width’

1. Nearest Neighbors Learning Models

The first determining property of machine learning algorithms is the split between supervised and unsupervised models. The difference between supervised and unsupervised models is the problem statement.

In supervised models, you have two types of variables at the same time:

* A target variable is the variable that you want to predict. It depends on the independent variables and it is not something that you know ahead of time.
* The independent variables are variables that you do know ahead of time. You can plug them into an equation to predict the target variable.

In unsupervised models, you don’t have a split between target variables and independent variables. Unsupervised learning tries to group data points by evaluating their similarity.

KNN(supervised) and K Means (Unsupervised) are some of the most famous Nearest Neighbors learning algorithms, and we are covering those using Iris dataset.

K Nearest Neighbors

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. It is a nonlinear model ( use any approach other than a line to separate their cases). KNN is a model that classifies data points based on the nearest points that are most similar to it. It uses train data to make an “educated guess” on what an unclassified point should be classified as.

|  |  |
| --- | --- |
| Pros: | Cons |
| Easy to use. | Accuracy depends on the quality of the data. |
| Quick calculation time. | Poor at classifying data points in a boundary where they can be classified one way or another. |
| Does not make assumptions about the data. |

Table 4: Advantages and disadvantages of KNN

KNN is  **non-parametric**, it means that it does not make any assumptions on the underlying data distribution. KNN is also a lazy algorithm, which means it does not use the training data points to do any generalization. In other words, there is no explicit training phase, or it is very minimal.

Problem

* If the training data available contains categories of iris flowers for classification, conduct a k-nearest neighbor classification.
* Try different k for KNN. Compare the out-of-sample prediction accuracy on the 20% testing sample that was not used for model building.

Implementation

* Split the data into 2 parts train, & test (80-20)
* Build KNN model with GridSearchCV, to find optimal K value.
* Plot the test-train scores.
* With the best model predict the category of the test dataset and get the performance metrics.

Results

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | | --- | --- | --- | --- | | Actual  Predicted | Setosa | Versicolour | Virginica | | Setosa | 11 | 0 | 0 | | Versicolour | 0 | 11 | 0 | | Virginica | 0 | 1 | 7 | |
| Figure 7: f\_score variation with k | Figure 8: Confusion Matrix |

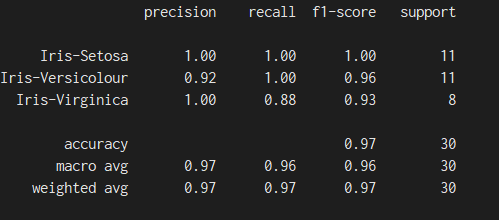


Figure 9: Test Data metric with KNN(K=11)

We see we get best model for K = 11.

K-Means Clustering

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are:

1. The centroids of the K clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster)

Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

Error Sum of Squares (**SSE**) is the sum of the squared differences between each observation and its group's mean. It can be used as a measure of variation within a **cluster**. The SSE vs k plot is used to identify an optimal value for the no. of clusters

# Problem

* Fit a k-means clustering to the iris data
* Compare and contrast different values of k
* Find an optimal k-value for clustering

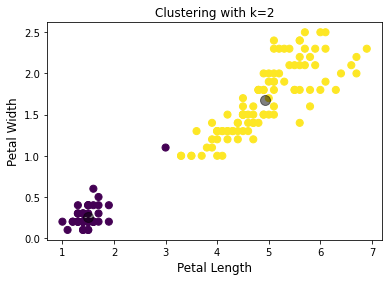
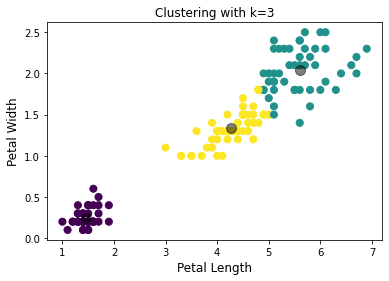
# Approach

* Use the columns petal length and petal width to perform clustering
* Use k-values from 2 to 5 and plot the k-means clustering results
* Compare with original labelled data to visually obtain an optimal k-value
* Verify with the *elbow plot* showing sum of squared errors (SSE) vs k

# Implementation

To implement k-means clustering visually, we are only considering the columns *petal length* and *petal width* to perform the clustering.

First, we try different values of k, namely k=2 through k=5 and see the results of clustering as below. The black dots in the graphs show the positions of the centroids in the clusters.

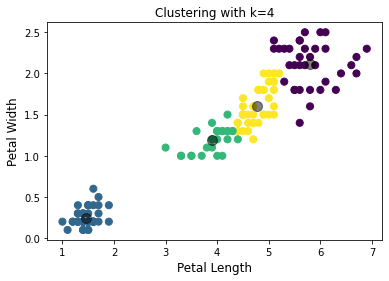
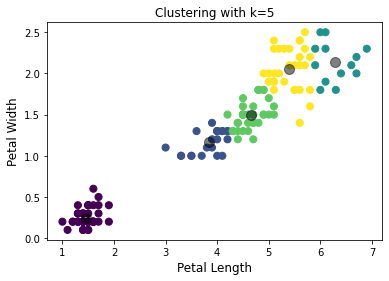
 

Figure 10: Cluster variations from k =2 to 5

Based on the above visualizations, we can see that from k=4 onwards, further clustering is not helpful and could be overfitting the data. We can further verify the same using the *elbow plot* below showing SSE vs k.

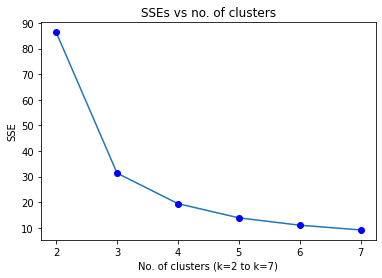


Figure 11: Distribution of Sum of squared errors w.r.t k-values

The above plot shows a huge drop in SSE from k=2 to k=3 and then flattens out from k=4. Thus, we can have an optimal k value of 3. This is also in-line with the labelled data which has 3 clusters.

Hierarchical Clustering

Hierarchical clustering is an alternative class of clustering algorithms that produce 1 to n clusters, where n is the number of observations in the data set. Here, we perform agglomerative hierarchical clustering where clustering starts with each observation as its own cluster. The two closest clusters are joined into one cluster. The next closest clusters are grouped together and this process continues until there is only one cluster containing the entire data set

# Implementation on Iris dataset

Here, we produce a dendrogram representing the hierarchical clustering on the iris dataset as below

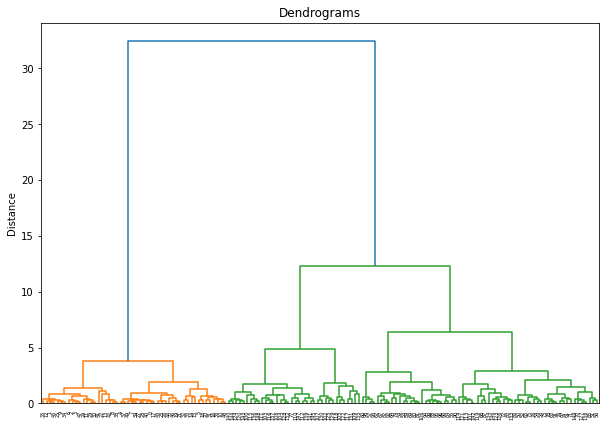


Figure 12: Dendogram for the Iris dataset

Since, the data has 3 clusters, we cut the dendrogram to obtain 3 clusters. Below is the *truncated dendrogram* to obtain 3 clusters

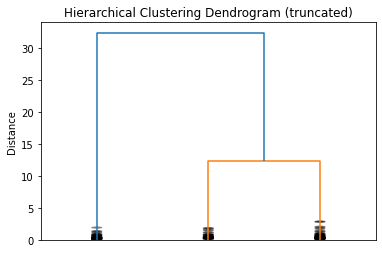


Figure 13: Truncated Dendogram