1. Write a Python program to view basic statistical details like percentile, mean, std etc. of iris data.
2. Write a Python program to split the iris dataset into its attributes (X) and labels (y). The X variable contains the first four columns (i.e. attributes) and y contains the labels of the dataset.
3. Write a Python program using Scikit-learn to split the iris dataset into 70% train data and 30% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Print both datasets.
4. Write a Python program using Scikit-learn to convert Species columns in a numerical column of the iris dataframe. To encode this data map convert each value to a number. e.g. Iris-setosa:0, Iris-versicolor:1, and Iris-virginica:2. Now print the iris dataset into 80% train data and 20% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Print both datasets.
5. Write a Python program using Scikit-learn to split the iris dataset into 70% train data and 30% test data. Out of total 150 records, the training set will contain 105 records and the test set contains 45 of those records. Predict the response for test dataset (SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm) using the K Nearest Neighbor Algorithm. Use 5 as number of neighbors.
6. Write a Python program using Scikit-learn to split the iris dataset into 80% train data and 20% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Train or fit the data into the model and calculate the accuracy of the model using the K Nearest Neighbor Algorithm.

|  |  |
| --- | --- |
|  | # Basic packages  import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  # Sklearn modules & classes  from sklearn.linear\_model import Perceptron, LogisticRegression  from sklearn.svm import SVC  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from sklearn import datasets  from sklearn import metrics |

|  |  |
| --- | --- |
|  | # Load the data set; In this example, the breast cancer dataset is loaded.  bc = datasets.load\_breast\_cancer()  X = bc.data  y = bc.target  # Create training and test split |
|  | sc = StandardScaler()  sc.fit(X\_train)  X\_train\_std = sc.transform(X\_train)  X\_test\_std = sc.transform(X\_test) |
|  | # Instantiate the Support Vector Classifier (SVC)  # Fit the model |
|  | # Make the predictions  # Measure the performance |

1. Import the libraries below

import numpy as np

import pandas as pd

import statsmodels.api as sm

import matplotlib.pyplot as plt

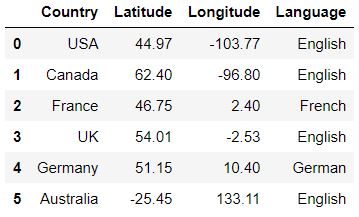
import seaborn as sns

sns.set()

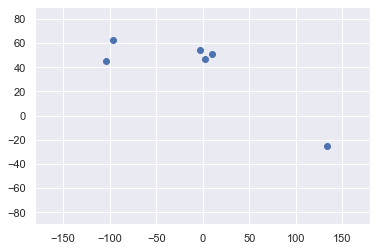
from sklearn.cluster import KMeans

Make a csv file name ‘Countryclusters.csv’ from table below and load it.

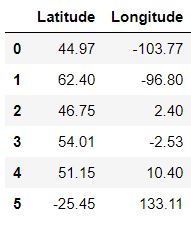
Print the data.



# Plot the data as shown below



Selecting the features below



Fit the features into KMeans Clustering

Predict the clustering results

identified\_clusters = kmeans.fit\_predict(x)

#Print identified clusters

#Scatter plot the Longitude and Latitude

data\_with\_clusters = data.copy()

data\_with\_clusters['Clusters'] = identified\_clusters

plt.scatter(data\_with\_clusters['Longitude'],data\_with\_clusters['Latitude'],c=data\_with\_clusters['Clusters'],cmap='rainbow')

