

Lab1: Simple Nearest Neighbor using Scikit-learn with Multiple Labels

1	<pre># KNN with Multiple Labels # Loading Data #Import scikit-learn dataset library from sklearn import datasets #Load dataset wine = datasets.load_wine()</pre>
2	<pre># Exploring Data # print the names of the features print(wine.feature_names)</pre>
3	<pre># print the label species(class_0, class_1, class_2) print(wine.target_names)</pre>
4	<pre># print the wine data (top 5 records) print(wine.data[0:5])</pre>
5	<pre># print the wine labels (0:Class_0, 1:Class_1, 2:Class_3) print(wine.target)</pre>
6	<pre># print data(feature) shape print(wine.data.shape)</pre>
7	<pre># print target(or label) shape print(wine.target.shape)</pre>
8	<pre># Splitting Data # Import train_test_split function from sklearn.model_selection import train_test_split # Split dataset into training set and test set X_train, X_test, y_train, y_test = train_test_split(wine.data, wine.target, test_size=0.3)</pre>
9	<pre># Generating Model for K=5 #Import knearest neighbors Classifier model from sklearn.neighbors import KNeighborsClassifier #Create KNN Classifier knn = KNeighborsClassifier(n_neighbors=5) #Train the model using the training sets knn.fit(X_train, y_train) #Predict the response for test dataset y_pred = knn.predict(X_test)</pre>
10	<pre># Model Evaluation for k=5 #Import scikit-learn metrics module for accuracy calculation from sklearn import metrics # Model Accuracy, how often is the classifier correct? print("Accuracy:",metrics.accuracy_score(y_test, y_pred))</pre>
11	<pre># Re-generating Model for K=7 #Import knearest neighbors Classifier model from sklearn.neighbors import KNeighborsClassifier</pre>

	<pre>#Create KNN Classifier knn = KNeighborsClassifier(n_neighbors=7) #Train the model using the training sets knn.fit(X_train, y_train) #Predict the response for test dataset y_pred = knn.predict(X_test)</pre>
12	<pre>#Import scikit-learn metrics module for accuracy calculation from sklearn import metrics # Model Accuracy, how often is the classifier correct? print("Accuracy:",metrics.accuracy_score(y_test, y_pred))</pre>