

## Lab3: Naïve Bayes using Python Scikit-learn: Multiple Labels

1	<pre>#Naive Bayes with Multiple Labels #Import scikit-learn dataset library from sklearn import datasets  #Loading Data #Load dataset wine = datasets.load_wine()</pre>
2	<pre>#Exploring Data # print the names of the 13 features print("Features: ", wine.feature_names)  # print the label type of wine(class_0, class_1, class_2) print("Labels: ", wine.target_names)</pre>
3	<pre># print data(feature)shape wine.data.shape</pre>
4	<pre># print the wine data features (top 5 records) print(wine.data[0:5])</pre>
5	<pre># Let's check records of the target set. # print the wine labels (0:Class_0, 1:class_2, 2:class_2) print(wine.target)</pre>
6	<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, random_state=109) # 70% training and 30% test</pre>
7	<pre># Model Generation # Import Gaussian Naive Bayes model from sklearn.naive_bayes import GaussianNB  #Create a Gaussian Classifier gnb = GaussianNB()  #Train the model using the training sets gnb.fit(X_train, y_train)  #Predict the response for test dataset y_pred = gnb.predict(X_test)</pre>
8	<pre>#Evaluating Model #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>
9	<pre># Getting Accuracy and Statistics print(metrics.classification_report(y_test, y_pred)) print(metrics.confusion_matrix(y_test, y_pred))</pre>

