PRACTICAL-4

Use Naive bayes, K-nearest, and Decision tree classification algorithms to build classifiers on any two datasets. Pre-process the datasets using techniques specified in Q2. Compare the Accuracy, Precision, Recall and F1 measure reported for each dataset using the abovementioned classifiers under the following situations:

i. Using Holdout method (Random sampling):
  a) Training set = 80% Test set = 20%
  b) Training set = 66.6% (2/3rd of total), Test set = 33.3%

ii. Using Cross-Validation:
  a) 10-fold
  b) 5-fold

Output:

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import CategoricalNB
from sklearn.metrics import confusion_matrix,classification_report

fruits = pd.read_csv("/content/fruit_classification_dataset.csv")
#Data Pre-processing
le = LabelEncoder()
fruits['shape'] = le.fit_transform(fruits['shape'])
fruits['color'] = le.fit_transform(fruits['color'])
fruits['taste'] = le.fit_transform(fruits['taste'])
fruits['fruit_name'] = le.fit_transform(fruits['fruit_name'])
```

```
#KNN with test-size of 20%

x=fruits.drop('fruit_name',axis=1)
y=fruits['fruit_name']
X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size = 0.2)


k = 5
knn_classifier = KNeighborsClassifier(n_neighbors=k)
knn_classifier.fit(X_train,Y_train)

y_pred = knn_classifier.predict(X_test)
y_preddec = le.inverse_transform(y_pred)
Y_testdec = le.inverse_transform(Y_test)
print(y_pred)
print(classification_report(Y_testdec,y_preddec))
```

```
[ 1  1  6 ... 11 17 11]
               precision    recall  f1-score   support

        apple       1.00      1.00      1.00        83
       banana       1.00      1.00      1.00        97
    blueberry       1.00      1.00      1.00       110
       cherry       1.00      1.00      1.00        85
      coconut       0.93      0.88      0.90       107
 custard apple       1.00      1.00      1.00       103
  dragon fruit       1.00      1.00      1.00       106
        grape       1.00      1.00      1.00       112
        guava       0.98      0.99      0.99       106
         kiwi       1.00      1.00      1.00        97
       lychee       1.00      1.00      1.00        94
        mango       0.98      0.94      0.96       104
       orange       1.00      1.00      1.00       100
       papaya       1.00      1.00      1.00        95
         pear       0.99      0.98      0.99       114
    pineapple       0.87      0.93      0.90        95
         plum       1.00      1.00      1.00       102
  pomegranate       0.94      0.98      0.96       105
   strawberry       1.00      1.00      1.00        83
   watermelon       1.00      1.00      1.00       102

     accuracy                           0.98      2000
    macro avg       0.98      0.99      0.98      2000
 weighted avg       0.98      0.98      0.98      2000
```

```python
# Naive Bayes for test size 20%

nb = CategoricalNB()
categorical_features = ['shape', 'color', 'taste']
X_train_cat = X_train[categorical_features]
X_test_cat = X_test[categorical_features]

nb.fit(X_train_cat, Y_train)
y_nbpred = nb.predict(X_test_cat)
y_nbpred_dec = le.inverse_transform(y_nbpred)
Y_test_dec = le.inverse_transform(Y_test)
print(classification_report(Y_test_dec,y_nbpred_dec))
```

```
               precision    recall  f1-score   support

        apple       0.00      0.00      0.00        83
       banana       1.00      1.00      1.00        97
    blueberry       1.00      1.00      1.00       110
       cherry       0.51      1.00      0.67        85
      coconut       1.00      1.00      1.00       107
 custard apple       0.50      1.00      0.67       103
  dragon fruit       1.00      1.00      1.00       106
        grape       1.00      1.00      1.00       112
        guava       1.00      1.00      1.00       106
         kiwi       0.00      0.00      0.00        97
       lychee       1.00      1.00      1.00        94
        mango       1.00      1.00      1.00       104
       orange       1.00      1.00      1.00       100
       papaya       1.00      1.00      1.00        95
         pear       1.00      1.00      1.00       114
    pineapple       0.49      1.00      0.66        95
         plum       1.00      1.00      1.00       102
  pomegranate       0.56      1.00      0.72       105
   strawberry       0.00      0.00      0.00        83
   watermelon       0.00      0.00      0.00       102

     accuracy                           0.82      2000
    macro avg       0.70      0.80      0.74      2000
 weighted avg       0.72      0.82      0.76      2000
```

```python
#KNN with test-size of 33%

x=fruits.drop('fruit_name',axis=1)
y=fruits['fruit_name']
X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size = 1/3)


k = 5
knn_classifier = KNeighborsClassifier(n_neighbors=k)
knn_classifier.fit(X_train,Y_train)

y_pred = knn_classifier.predict(X_test)
y_preddec = le.inverse_transform(y_pred)
Y_testdec = le.inverse_transform(Y_test)
print(y_pred)
print(classification_report(Y_testdec,y_preddec))
```

```
               precision    recall  f1-score   support

        apple       1.00      1.00      1.00       173
       banana       1.00      1.00      1.00       179
    blueberry       1.00      1.00      1.00       162
       cherry       1.00      1.00      1.00       177
      coconut       0.93      0.87      0.90       177
custard apple       1.00      1.00      1.00       185
 dragon fruit       1.00      1.00      1.00       184
        grape       1.00      1.00      1.00       156
        guava       0.99      0.99      0.99       143
         kiwi       1.00      1.00      1.00       146
       lychee       1.00      1.00      1.00       164
        mango       0.94      0.96      0.95       155
       orange       1.00      1.00      1.00       157
       papaya       1.00      1.00      1.00       185
         pear       0.99      0.99      0.99       174
    pineapple       0.87      0.93      0.90       167
         plum       1.00      1.00      1.00       166
  pomegranate       0.96      0.94      0.95       165
   strawberry       1.00      1.00      1.00       158
   watermelon       1.00      1.00      1.00       161

     accuracy                           0.98      3334
    macro avg       0.98      0.98      0.98      3334
 weighted avg       0.98      0.98      0.98      3334
```

.0]
0s

```python
# Naive Bayes for test size 33%

nb = CategoricalNB()
categorical_features = ['shape', 'color', 'taste']
X_train_cat = X_train[categorical_features]
X_test_cat = X_test[categorical_features]

nb.fit(X_train_cat, Y_train)
y_nbpred = nb.predict(X_test_cat)
y_nbpred_dec = le.inverse_transform(y_nbpred)
Y_test_dec = le.inverse_transform(Y_test)
print(classification_report(Y_test_dec,y_nbpred_dec))
```

```
                 precision    recall  f1-score   support

          apple       0.00      0.00      0.00       173
         banana       1.00      1.00      1.00       179
      blueberry       1.00      1.00      1.00       162
         cherry       0.53      1.00      0.69       177
        coconut       1.00      1.00      1.00       177
  custard apple       0.00      0.00      0.00       185
   dragon fruit       1.00      1.00      1.00       184
          grape       1.00      1.00      1.00       156
          guava       1.00      1.00      1.00       143
           kiwi       0.00      0.00      0.00       146
         lychee       1.00      1.00      1.00       164
          mango       1.00      1.00      1.00       155
         orange       1.00      1.00      1.00       157
         papaya       1.00      1.00      1.00       185
           pear       1.00      1.00      1.00       174
      pineapple       0.53      1.00      0.70       167
           plum       1.00      1.00      1.00       166
    pomegranate       0.49      1.00      0.66       165
     strawberry       0.00      0.00      0.00       158
     watermelon       0.47      1.00      0.64       161

       accuracy                           0.80      3334
      macro avg       0.70      0.80      0.73      3334
   weighted avg       0.70      0.80      0.74      3334
```