DSHBA - Model Building

Student: Gabriel Yugo Nascimento Kishida
SIT ID: Z122232

Code for Setup:

```
# Imports
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from imblearn.over_sampling import SMOTE

# Reading data and preparing arrays for model building
fruits = pd.read_csv('fruits.csv')
feature_names = ['mass', 'width', 'height', 'color_score']
X = fruits[feature_names]
Y = fruits['fruit_name']
```

Task

Build SVM models using "fruits.csv" dataset based on these conditions:

- **Training/Test Ratio: {90:10, 60:40}**
- **Resampling: {RUS, ROS, SMOTE}**

Definition of function for model building:

```
# Definition of function that builds models with different resampling
methods and test sizes
def test_model(resampling, test_size):
  print("Current resampling: " + resampling + " | Current test size: " +
str(test_size))
  x_prepared = X.copy()
  y_prepared = Y.copy()
  if resampling == 'RUS':
    rus = RandomUnderSampler()
    x_prepared, y_prepared =  rus.fit_resample(x_prepared, y_prepared)
  elif resampling == 'ROS':
    ros = RandomOverSampler()
    x_prepared, y_prepared = ros.fit_resample(x_prepared, y_prepared)
  elif resampling == 'SMOTE':
    sm = SMOTE(k_neighbors=4)
    x_prepared, y_prepared = sm.fit_resample(x_prepared, y_prepared)

  X_train, X_test, Y_train, Y_test = train_test_split(x_prepared,
y_prepared,
                                                      random_state=0,
test_size=test_size)
  scaler = MinMaxScaler()
  X_train = scaler.fit_transform(X_train)
  X_test = scaler.transform(X_test)
  svm = SVC()
  svm.fit(X_train, Y_train)
  pred = svm.predict(X_test)
  print('Accuracy of classifier on training set:
{:.2f}'.format(svm.score(X_train, Y_train)))
  print('Accuracy of classifier on test set:
{:.2f}'.format(svm.score(X_test, Y_test)))
  print(classification_report(Y_test, pred))
  print("")
```

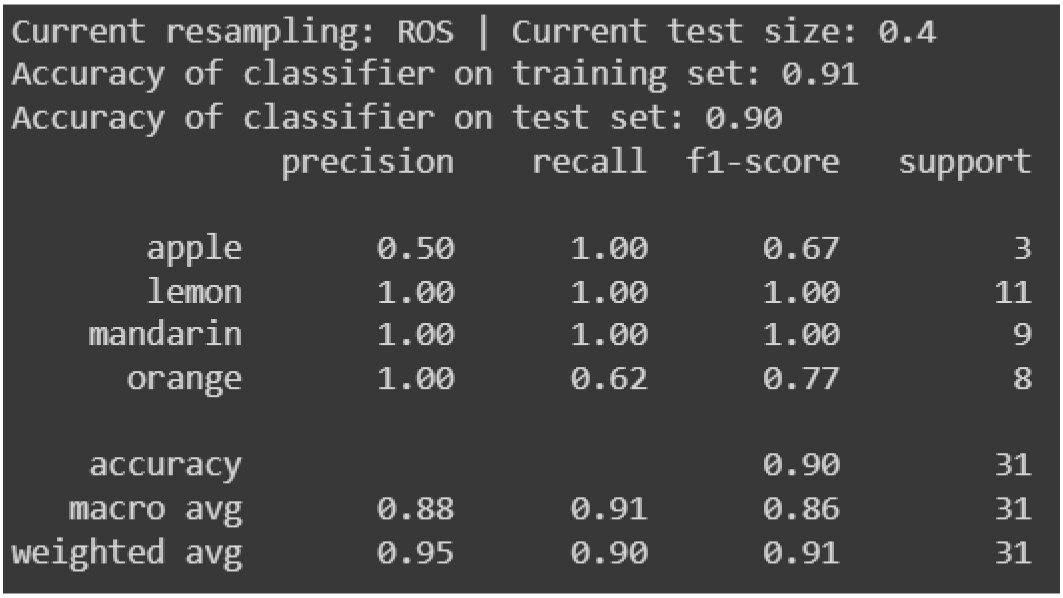
Calling the function for different sampling methods and test sizes:

```
# Begin building models
resampling_methods = {"RUS", "ROS", "SMOTE"}
test_sample_sizes = {0.4, 0.1}
for current_resampling_method in resampling_methods:
  for current_test_size in test_sample_sizes:
    test_model(current_resampling_method, current_test_size)
```

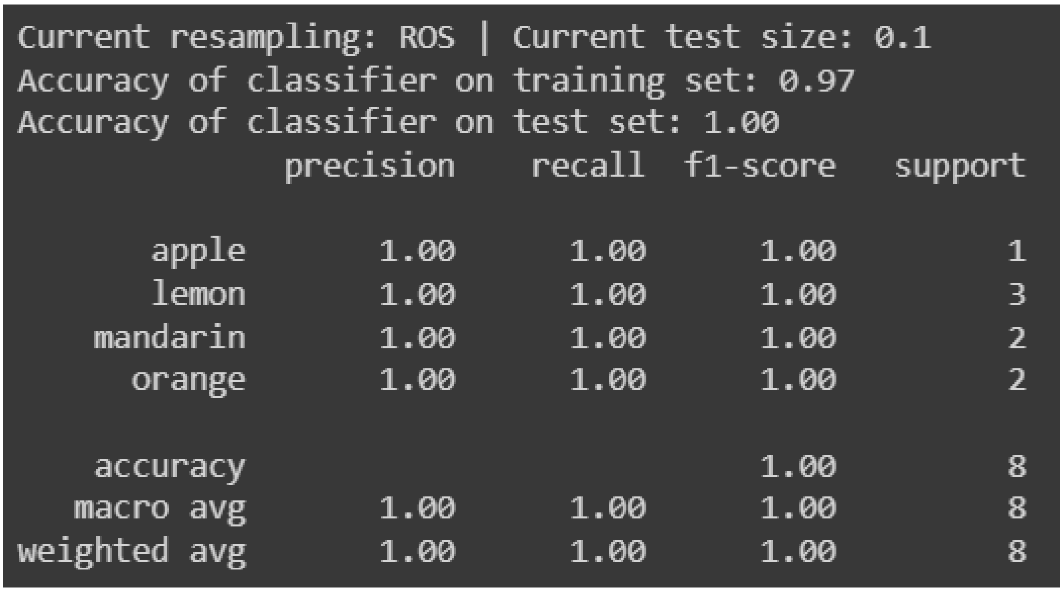
Result Table:

Conditions	Accuracy on Training Set	Accuracy on Test Set	F1-Score (Macro average)	F1-Score (Weighted average)
90:10, RUS	1.00	0.50	0.33	0.33
90:10, ROS	0.97	1.00	1.00	1.00
90:10, SMOTE	0.97	1.00	1.00	1.00
60:40, RUS	0.75	0.62	0.60	0.55
60:40, ROS	0.91	0.90	0.86	0.91
60:40, SMOTE	0.91	0.90	0.86	0.91

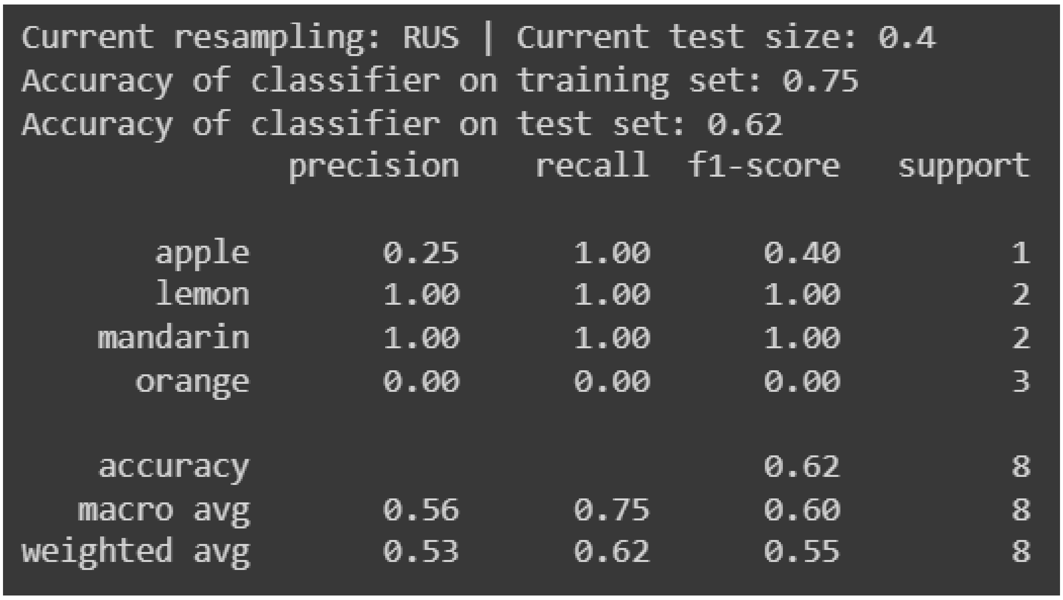
Output obtained



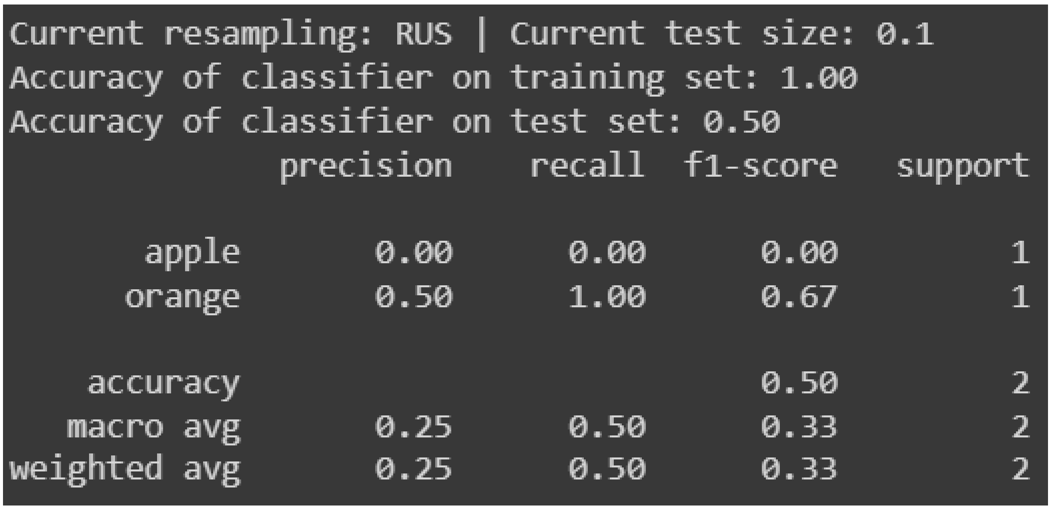
```
Current resampling: ROS | Current test size: 0.4
Accuracy of classifier on training set: 0.91
Accuracy of classifier on test set: 0.90
              precision    recall  f1-score   support

       apple       0.50      1.00      0.67         3
       lemon       1.00      1.00      1.00        11
    mandarin       1.00      1.00      1.00         9
      orange       1.00      0.62      0.77         8

    accuracy                           0.90        31
   macro avg       0.88      0.91      0.86        31
weighted avg       0.95      0.90      0.91        31
```



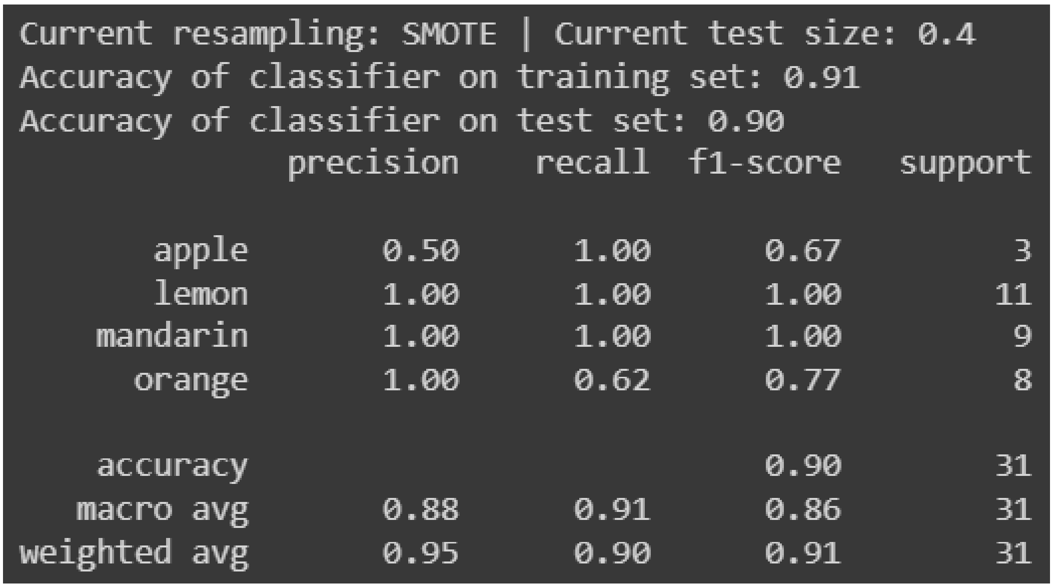
```
Current resampling: ROS | Current test size: 0.1
Accuracy of classifier on training set: 0.97
Accuracy of classifier on test set: 1.00
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00         1
       lemon       1.00      1.00      1.00         3
    mandarin       1.00      1.00      1.00         2
      orange       1.00      1.00      1.00         2

    accuracy                           1.00         8
   macro avg       1.00      1.00      1.00         8
weighted avg       1.00      1.00      1.00         8
```



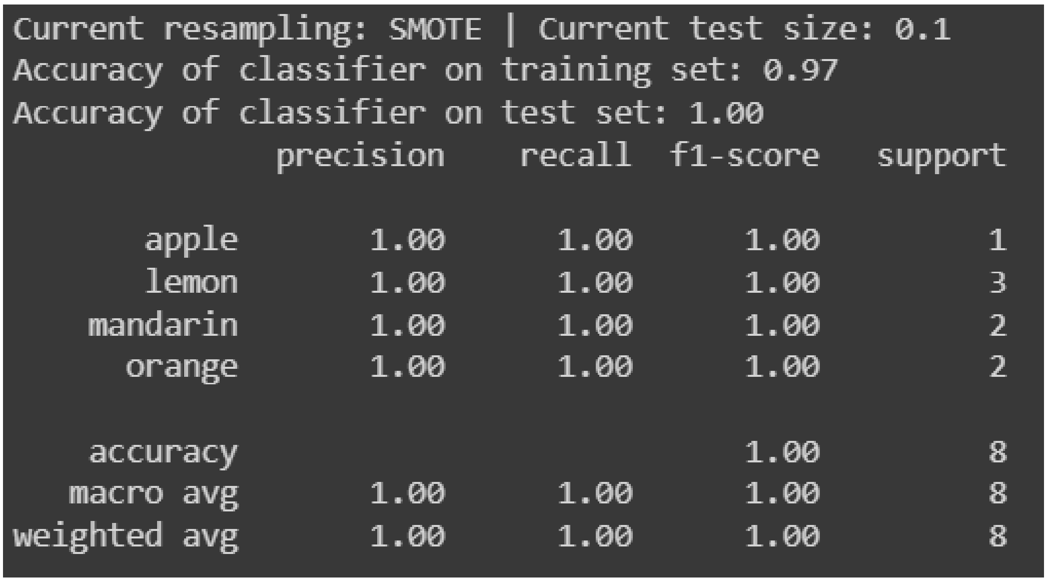
```
Current resampling: RUS | Current test size: 0.4
Accuracy of classifier on training set: 0.75
Accuracy of classifier on test set: 0.62
              precision    recall  f1-score   support

       apple       0.25      1.00      0.40         1
       lemon       1.00      1.00      1.00         2
    mandarin       1.00      1.00      1.00         2
      orange       0.00      0.00      0.00         3

    accuracy                           0.62         8
   macro avg       0.56      0.75      0.60         8
weighted avg       0.53      0.62      0.55         8
```



```
Current resampling: RUS | Current test size: 0.1
Accuracy of classifier on training set: 1.00
Accuracy of classifier on test set: 0.50
              precision    recall  f1-score   support

       apple       0.00      0.00      0.00         1
      orange       0.50      1.00      0.67         1

    accuracy                           0.50         2
   macro avg       0.25      0.50      0.33         2
weighted avg       0.25      0.50      0.33         2
```



```
Current resampling: SMOTE | Current test size: 0.4
Accuracy of classifier on training set: 0.91
Accuracy of classifier on test set: 0.90
              precision    recall  f1-score   support

       apple       0.50      1.00      0.67         3
       lemon       1.00      1.00      1.00        11
    mandarin       1.00      1.00      1.00         9
      orange       1.00      0.62      0.77         8

    accuracy                           0.90        31
   macro avg       0.88      0.91      0.86        31
weighted avg       0.95      0.90      0.91        31
```



```
Current resampling: SMOTE | Current test size: 0.1
Accuracy of classifier on training set: 0.97
Accuracy of classifier on test set: 1.00
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00         1
       lemon       1.00      1.00      1.00         3
    mandarin       1.00      1.00      1.00         2
      orange       1.00      1.00      1.00         2

    accuracy                           1.00         8
   macro avg       1.00      1.00      1.00         8
weighted avg       1.00      1.00      1.00         8
```