Practical Activity 9 Classification using ensemble models

September 19, 2023

1 Practical Activity 9

1.1 Classification using Random Forest Ensemble

This notebook is an exercise for developing a Random Forest classifier for predicting the types of wine. We apply the concepts discussed in Week 9. We walk through RF Classifier in this practical. Note: this activity is unmarked. It develops your skills for predictive model development using Ensemble approaches.

1.2 Task

Our aim is to build a classification model to predict types of wine. We will be using the wine dataset which contains 178 observations and 13 variables: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline; and the Outcome - class_0, class_1 and class_2. The dataset is available at https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine

1.3 Evaluation Metric

We will evaluate the performance of the model using precision, recall and F1 score. See https://en.wikipedia.org/wiki/Precision_and_recall for more details.

1.3.1 Step 1 - Load libaries

[29]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from sklearn.datasets import load_wine
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
```

1.3.2 Step 2 - Load data

[17]:
```
# Load data
data = load_wine()
# Transform the data to dataframe format
df = pd.DataFrame(data=np.c_[data['data'], data['target']],
                  columns= data['feature_names'] + ['target'])
# Transform the outcome to categorical
df['target'] = df.target.astype('str')
df.head()
```

[17]:
```
   alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0    14.23        1.71  2.43               15.6      127.0           2.80
1    13.20        1.78  2.14               11.2      100.0           2.65
2    13.16        2.36  2.67               18.6      101.0           2.80
3    14.37        1.95  2.50               16.8      113.0           3.85
4    13.24        2.59  2.87               21.0      118.0           2.80

   flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0        3.06                  0.28             2.29             5.64  1.04
1        2.76                  0.26             1.28             4.38  1.05
2        3.24                  0.30             2.81             5.68  1.03
3        3.49                  0.24             2.18             7.80  0.86
4        2.69                  0.39             1.82             4.32  1.04

   od280/od315_of_diluted_wines  proline target
0                          3.92   1065.0    0.0
1                          3.40   1050.0    0.0
2                          3.17   1185.0    0.0
3                          3.45   1480.0    0.0
4                          2.93    735.0    0.0
```

[18]:
```
# Shape
df.shape
```

[18]:
```
(178, 14)
```

[19]:
```
# Statistical summary
df.describe()
```

[19]:
```
          alcohol  malic_acid         ash  alcalinity_of_ash   magnesium  \
count  178.000000  178.000000  178.000000         178.000000  178.000000
mean    13.000618    2.336348    2.366517          19.494944   99.741573
std      0.811827    1.117146    0.274344           3.339564   14.282484
min     11.030000    0.740000    1.360000          10.600000   70.000000
25%     12.362500    1.602500    2.210000          17.200000   88.000000
50%     13.050000    1.865000    2.360000          19.500000   98.000000
75%     13.677500    3.082500    2.557500          21.500000  107.000000
```

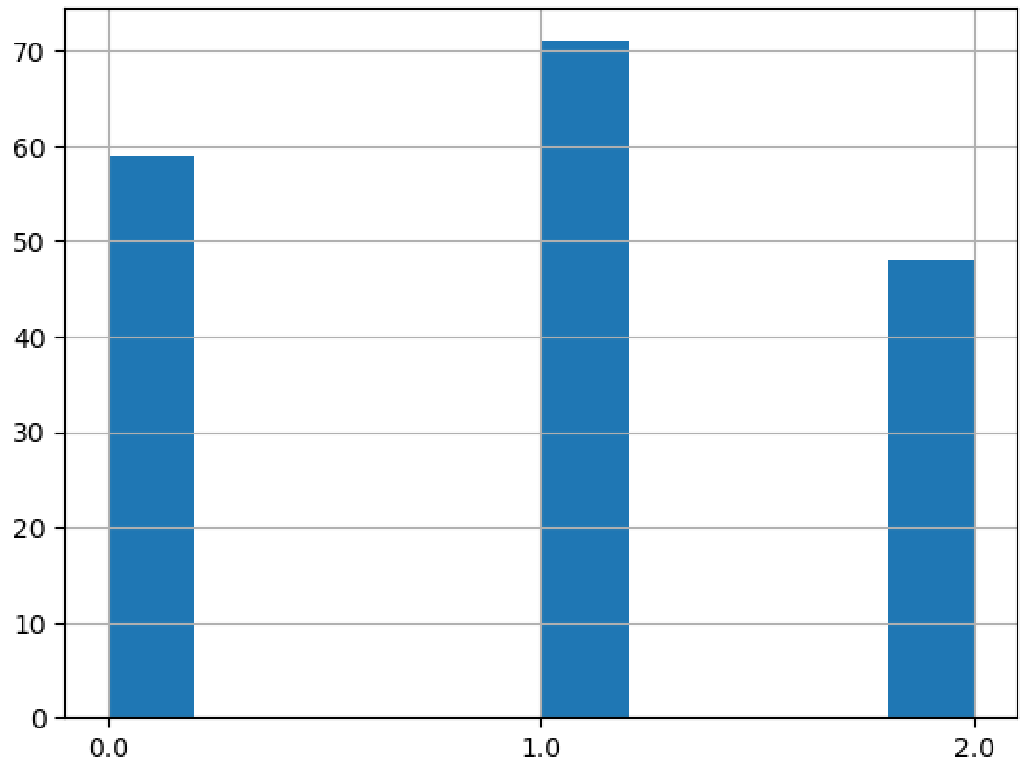
```
max     14.830000    5.800000    3.230000          30.000000  162.000000

       total_phenols  flavanoids  nonflavanoid_phenols  proanthocyanins  \
count     178.000000  178.000000            178.000000       178.000000
mean        2.295112    2.029270              0.361854         1.590899
std         0.625851    0.998859              0.124453         0.572359
min         0.980000    0.340000              0.130000         0.410000
25%         1.742500    1.205000              0.270000         1.250000
50%         2.355000    2.135000              0.340000         1.555000
75%         2.800000    2.875000              0.437500         1.950000
max         3.880000    5.080000              0.660000         3.580000

       color_intensity         hue  od280/od315_of_diluted_wines      proline
count       178.000000  178.000000                    178.000000   178.000000
mean          5.058090    0.957449                      2.611685   746.893258
std           2.318286    0.228572                      0.709990   314.907474
min           1.280000    0.480000                      1.270000   278.000000
25%           3.220000    0.782500                      1.937500   500.500000
50%           4.690000    0.965000                      2.780000   673.500000
75%           6.200000    1.120000                      3.170000   985.000000
max          13.000000    1.710000                      4.000000  1680.000000
```

[20]:
```
# Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  target                        178 non-null    object
dtypes: float64(13), object(1)
memory usage: 18.8+ KB
```

[23]:
```
df.target.hist()
```

[23]: <AxesSubplot:>



The above shows that the target variable is quite balanced.

1.3.3 Step 3 - Create train and test data

[26]:
```
# Split
train, test = train_test_split(df, test_size = 0.3, stratify = df['target'])

X_train = train.drop('target', axis=1)
y_train = train['target']

X_test = test.drop('target', axis = 1)
y_test = test['target']

# shapes
X_train.shape, X_test.shape
```

[26]: ((124, 13), (54, 13))

1.3.4 Step 4 - Build the random forest model

In this step, we will build the random forest model using sklearn, see https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html for details.

[28]:
```
# build model
clf = RandomForestClassifier(max_depth=5, random_state=0)
# fit to data
clf.fit(X_train, y_train)

clf
```

[28]:
```
RandomForestClassifier(max_depth=5, random_state=0)
```

1.3.5 Step 5 - Evaluate the performance

[30]:
```
# prediction on train
pred_train = clf.predict(X_train)

target_names = ['class 0', 'class 1', 'class 2']
print(classification_report(y_train, pred_train, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       1.00      1.00      1.00        41
     class 1       1.00      1.00      1.00        50
     class 2       1.00      1.00      1.00        33

    accuracy                           1.00       124
   macro avg       1.00      1.00      1.00       124
weighted avg       1.00      1.00      1.00       124
```

The F1 score on train data is 1.00.

[32]:
```
# prediction on test
pred_test = clf.predict(X_test)

target_names = ['class 0', 'class 1', 'class 2']
print(classification_report(y_test, pred_test, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       1.00      0.89      0.94        18
     class 1       0.90      0.90      0.90        21
     class 2       0.88      1.00      0.94        15

    accuracy                           0.93        54
   macro avg       0.93      0.93      0.93        54
```

```
weighted avg       0.93      0.93      0.93        54
```

The F1 score on test data is 0.93.

2 Task

Try to find the best set of parameters for the random forest model. Try to build another ensemble approaches, using different ensemble set of models, e.g. the combination of LR and SVM.