lab-7-22mcb1002

April 22, 2023

1 Numerical Feature Selection

[ ]:
```
# Package Import
import pandas as pd
import numpy as np
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.feature_selection import mutual_info_classif
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

[ ]:
```
# Function definition for dataset import and preprocessing
def load_dataset(fname):
  df = pd.read_csv(fname, header = None)
  df = df.fillna(df.mean())
  df = df.dropna()
  dataset = df.values
  X = dataset[:,:-1]
  y = dataset[:,-1]
  return X,y
```

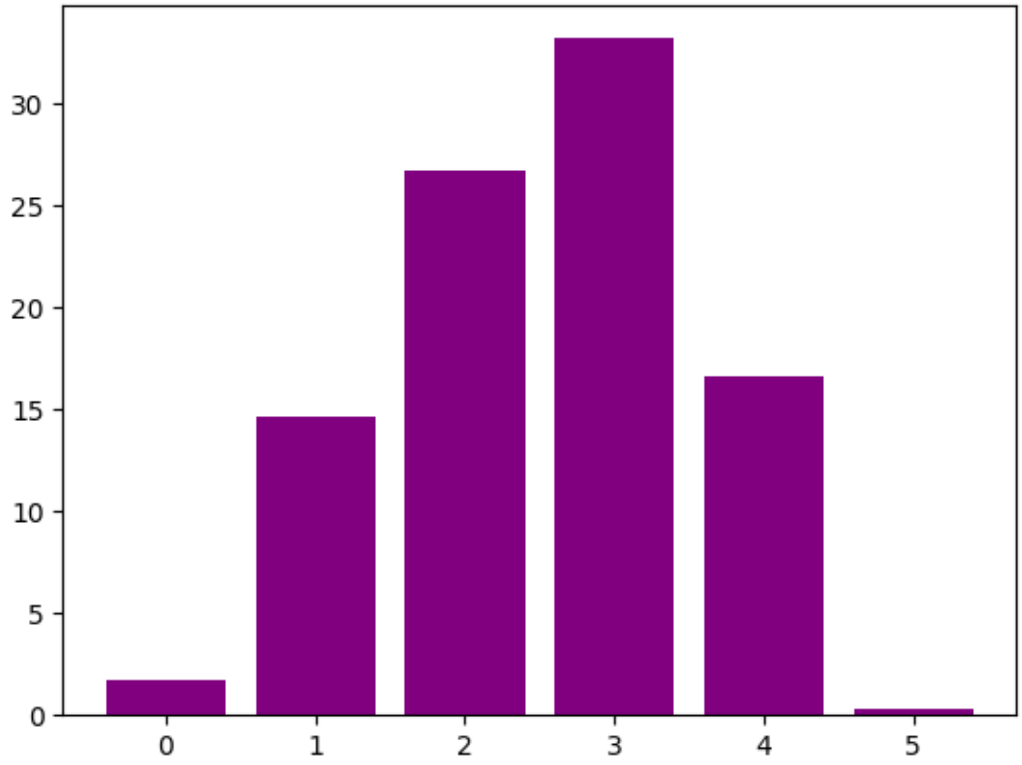
[ ]:
```
# Loading heart attack dataset
X,y = load_dataset('/content/drive/MyDrive/Data Analytics/heartdataset.csv')
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.
 ↪33,random_state = 1)
print('Train', X_train.shape, y_train.shape)
print('Test', X_test.shape, y_test.shape)
```

```
Train (616, 6) (616,)
Test (304, 6) (304,)
```

[ ]:
```
# Training
def select_features(X_train, y_train,X_test):
  fs=SelectKBest(score_func = f_classif, k=4)
  fs.fit(X_train, y_train)
```

```
  X_train_fs = fs.transform(X_train)
  X_test_fs = fs.transform(X_test)
  return X_train_fs, X_test_fs, fs
```

[ ]:
```
# ANOVA F_statistic score
X_train_fs, X_test_fs, fs = select_features(X_train, y_train, X_test)
pyplot.bar([i for i in range(len(fs.scores_))], fs.scores_, color = 'purple')
pyplot.show()
```

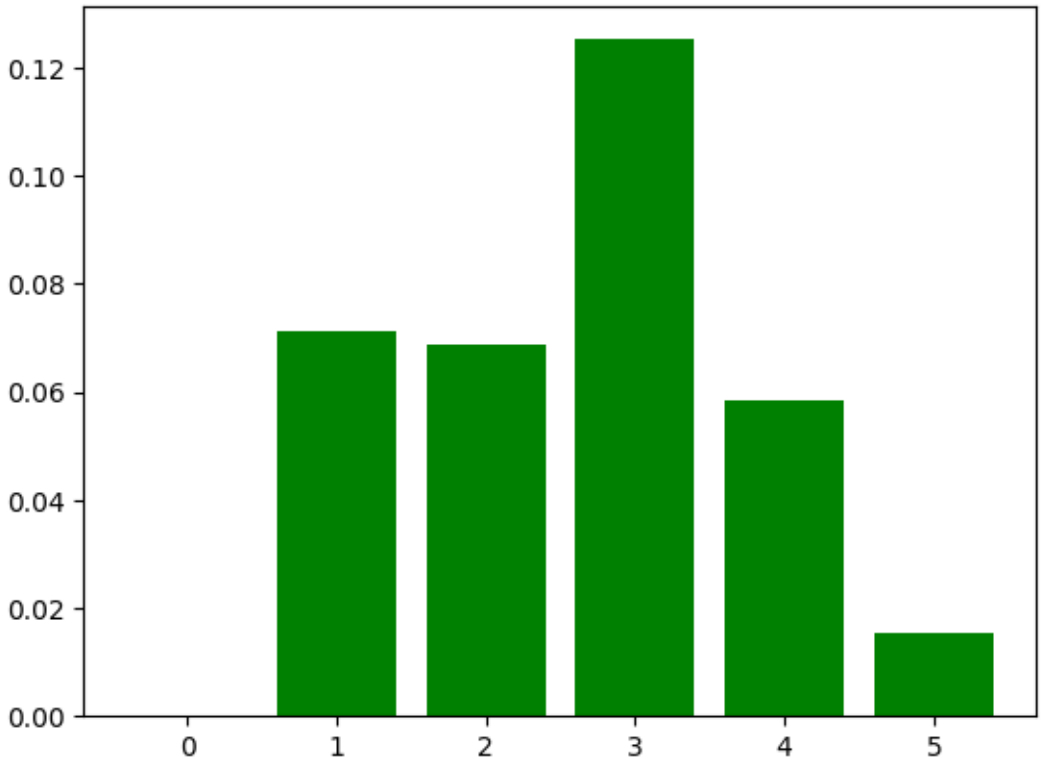


[ ]:
```
fs.scores_
```

[ ]:
```
array([ 1.74318675, 14.66752737, 26.72634966, 33.18661485, 16.65146106,
        0.26876886])
```

[ ]:
```
# Mutual information based feature selection
def select_features_2(X_train,y_train,X_test):
  fs = SelectKBest(score_func = mutual_info_classif, k=4)
  fs.fit(X_train, y_train)
  X_train_fs = fs.transform(X_train)
  X_test_fs = fs.transform(X_test)
  return X_train_fs, X_test_fs,fs
```

[ ]:
```
# Selecting best features using Mutual information
X_train_fs2, X_test_fs2, fs2 = select_features_2(X_train,y_train,X_test)
pyplot.bar([i for i in range(len(fs2.scores_))],fs2.scores_, color = 'green')
pyplot.show()
```



[ ]:
```
fs2.scores_
```

[ ]:
```
array([0.        , 0.08369714, 0.07055355, 0.08481058, 0.04439527,
       0.01274103])
```

[ ]:
```
model1 = LogisticRegression(solver = 'liblinear')
model1.fit(X_train, y_train)
yhat = model1.predict(X_test)
accuracy = accuracy_score(y_test, yhat)
print("Accuracy: %.2f" %(accuracy*100))
```

```
Accuracy: 52.30
```

[ ]:
```
#Model built using features chosen with ANOVA F-statistic
model2 = LogisticRegression(solver = 'liblinear')
model2.fit(X_train_fs, y_train)
yhat = model2.predict(X_test_fs)
```

```
accuracy = accuracy_score(y_test, yhat)
print("Accuracy: %.2f" %(accuracy*100))
```

```
Accuracy: 51.64
```

[ ]:
```
#Model built using features chosen with Mutual information
model3 = LogisticRegression(solver = 'liblinear')
model3.fit(X_train_fs2, y_train)
yhat = model3.predict(X_test_fs2)
accuracy = accuracy_score(y_test, yhat)
print("Accuracy: %.2f" %(accuracy*100))
```

```
Accuracy: 51.64
```

[ ]:
```
from sklearn.pipeline import Pipeline
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV
```

[ ]:
```
#Tuning the number of selected features grid search
cv = RepeatedStratifiedKFold(n_splits = 10, n_repeats = 3, random_state = 1)
```

[ ]:
```
# Defining pipeline
model = LogisticRegression(solver = 'liblinear')
fs = SelectKBest(score_func = f_classif)
pipeline = Pipeline(steps = [('anova',fs),('lr',model)])
```

[ ]:
```
#defining grid
grid = dict()
grid['anova__k'] = [i+1 for i in range(X.shape[1])]
```

[ ]:
```
# Applying grid search
search = GridSearchCV(pipeline, grid, scoring = 'accuracy', n_jobs = -1, cv =␣
↪cv)
results = search.fit(X, y)
print('Best Mean Accuracy: %.3f' % results.best_score_)
print('Best config %s' %results.best_params_)

# The results shows that the no. of best features is 5 and the mean accuracy
# with 5 features is 0.54
```

```
Best Mean Accuracy: 0.545
Best config {'anova__k': 5}
```

Comparing different no. of features selected using ANOVA f-test

[ ]:
```
from numpy import mean
from numpy import std
from sklearn.model_selection import cross_val_score
```

[ ]:
```
def evaluate_model(model):
  cv = RepeatedStratifiedKFold(n_repeats = 3, n_splits = 10, random_state = 1)
  scores = cross_val_score(model, X, y, scoring='accuracy',cv = cv,n_jobs = 1)
  return scores
```

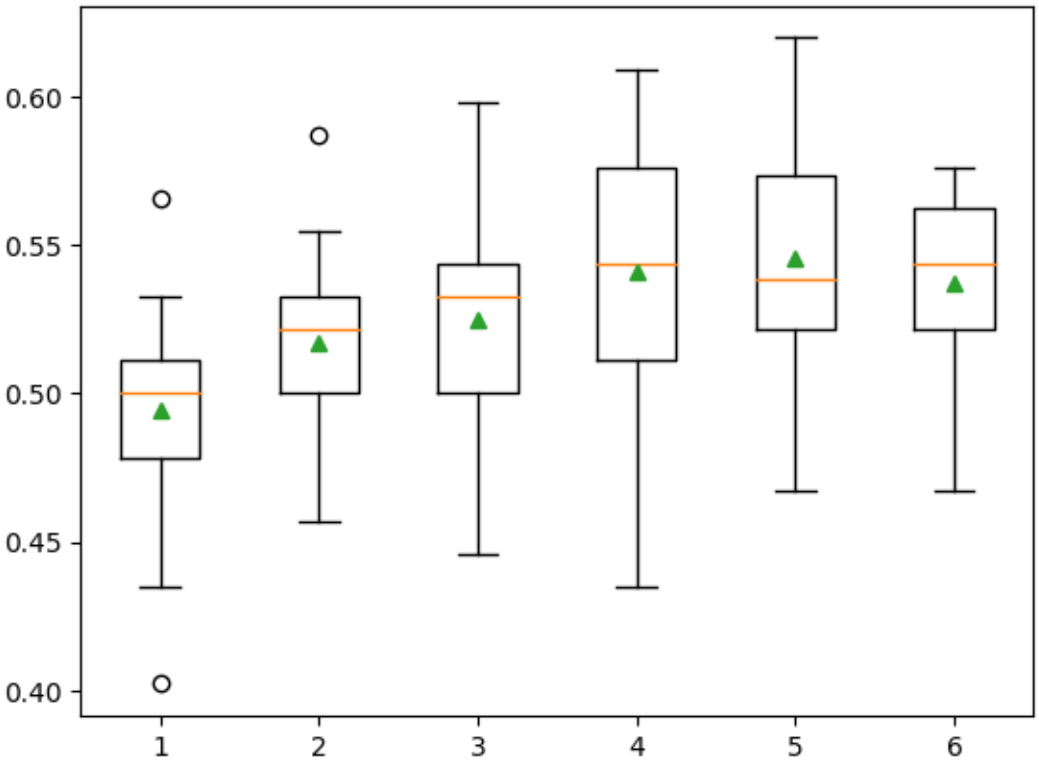
[ ]:
```
X,y = load_dataset('/content/drive/MyDrive/Data Analytics/heartdataset.csv')
```

[ ]:
```
num_features = [i+1 for i in range(X.shape[1])]
```

[ ]:
```
# Enumerating each feature
results = list()
for k in num_features:
#create pipeline
  model = LogisticRegression(solver = 'liblinear')
  fs = SelectKBest(score_func = f_classif,k = k)
  pipeline = Pipeline(steps = [('anova',fs),('lr',model)])
  #evaluate the model
  scores = evaluate_model(pipeline)
  results.append(scores)
  print('#%d %.3f (%.3f)' %(k,mean(scores),std(scores)))
```

```
#1 0.495 (0.030)
#2 0.517 (0.027)
#3 0.525 (0.030)
#4 0.541 (0.038)
#5 0.545 (0.036)
#6 0.537 (0.030)
```

[ ]:
```
pyplot.boxplot(results, labels = num_features, showmeans = True)
pyplot.show()
```


0.60
0.55
0.50
0.45
0.40
1
2
3
4
5
6