feature_selection

February 8, 2021

[189]:
```
import pandas as pd
import numpy as np
import warnings
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import mutual_info_classif
from sklearn.ensemble import RandomForestClassifier
from scipy import stats

warnings.filterwarnings('ignore')
```

[190]:
```
data = pd.read_csv("../Dataset/leaf.csv", delimiter=",")
X_, y_ = data.iloc[:, :-1], data.iloc[:, -1:],
x_train, x_test, y_train, y_test = train_test_split(X_, y_, test_size=0.2,␣
↪random_state= 0)
num_features = len(X_.columns)
```

1 Full Dataset Accuracy

[193]:
```
clf = KNeighborsClassifier(n_neighbors=5)
cv_strat = RepeatedStratifiedKFold(n_splits=5, n_repeats=4, random_state=42)
```

[194]:
```
cv_results_full = cross_val_score(estimator = clf,  X = X_, y = y_, cv =␣
↪cv_strat, scoring = 'accuracy')
```

[195]:
```
cv_results_full.mean()
```

[195]:
```
0.5477941176470588
```

2 FScore accuracy

[197]:
```
fscore = SelectKBest(k = 'all')
fscore.fit_transform(X_, y_)
indices_fscore = np.argsort(fscore.scores_)[::-1][0:num_features]
print(X.columns[indices_fscore].values)
print(fscore.scores_[indices_fscore])
fscore = SelectKBest(k = 'all')
fscore.fit_transform(X_, y_)
indices_fscore = np.argsort(fscore.scores_)[::-1][0:num_features]
print(X.columns[indices_fscore].values)
print(fscore.scores_[indices_fscore])# FScore accuracy
```

```
['Aspect Ratio' 'Isoperimetric Factor' 'Solidity' 'Elongation'
 'Stochastic Convexity' 'Eccentricity' 'Maximal Indentation Depth'
 'Average Contrast' 'Smoothness' 'Average Intensity' 'Entropy'
 'Third moment' 'Lobedness' 'Uniformity']
[177.36851327 150.62706584 143.51971656 120.46404269  85.31945942
  66.08024337  44.42536511  37.38593165  33.2840698   29.12326416
  27.17029832  26.67072643  23.63650093  11.45661752]
['Aspect Ratio' 'Isoperimetric Factor' 'Solidity' 'Elongation'
 'Stochastic Convexity' 'Eccentricity' 'Maximal Indentation Depth'
 'Average Contrast' 'Smoothness' 'Average Intensity' 'Entropy'
 'Third moment' 'Lobedness' 'Uniformity']
[177.36851327 150.62706584 143.51971656 120.46404269  85.31945942
  66.08024337  44.42536511  37.38593165  33.2840698   29.12326416
  27.17029832  26.67072643  23.63650093  11.45661752]
```

[198]:
```
cv_results_fscore = cross_val_score(estimator = clf,  X = X_.iloc[:,␣
↪indices_fscore[:10]], y = y_, cv = cv_strat, scoring = 'accuracy')
cv_results_fscore.mean()
```

[198]:
```
0.5691176470588235
```

3 Mutual Info Accuracy

[199]:
```
mutual_info = SelectKBest(mutual_info_classif, k = 'all')
mutual_info.fit_transform(X_, y_)
indices_mutual_info = np.argsort(mutual_info.scores_)[::-1][0:num_features]
print(X.columns[indices_mutual_info].values)
print(mutual_info.scores_[indices_mutual_info])
```

```
['Aspect Ratio' 'Solidity' 'Elongation' 'Eccentricity'
 'Isoperimetric Factor' 'Maximal Indentation Depth' 'Lobedness'
 'Stochastic Convexity' 'Third moment' 'Average Contrast' 'Smoothness'
```

```
 'Average Intensity' 'Entropy' 'Uniformity']
[1.42371091 1.36483519 1.33016315 1.31395589 1.27636629 1.11460545
 1.09612128 0.79660176 0.79423131 0.78663465 0.77598898 0.77019057
 0.75231111 0.70088635]
```

[200]:
```
cv_results_mutual_info = cross_val_score(estimator = clf,  X = X_.iloc[:,␣
↪indices_mutual_info[:12]], y = y_, cv = cv_strat, scoring = 'accuracy')
cv_results_mutual_info.mean()
```

[200]:
```
0.5647058823529412
```

4 Random Forest Importance Accuracy

[209]:
```
rfi = RandomForestClassifier(n_estimators = 200)
rfi.fit(X_, y_)
indices_rfi = np.argsort(rfi.feature_importances_)[::-1][0:num_features]
print(X.columns[indices_rfi].values)
print(rfi.feature_importances_[indices_rfi])
```

```
['Solidity' 'Elongation' 'Isoperimetric Factor' 'Aspect Ratio'
 'Eccentricity' 'Maximal Indentation Depth' 'Lobedness' 'Entropy'
 'Uniformity' 'Stochastic Convexity' 'Average Intensity' 'Third moment'
 'Smoothness' 'Average Contrast']
[0.11393757 0.08906468 0.08836677 0.08810554 0.08799477 0.07425238
 0.0703576  0.06909254 0.05912546 0.05410451 0.0535216  0.05154372
 0.05141333 0.04911953]
```

[210]:
```
cv_results_rfi = cross_val_score(estimator = clf,  X = X_.iloc[:, indices_rfi[:
↪8]], y = y_, cv = cv_strat, scoring = 'accuracy')
cv_results_rfi.mean()
```

[210]:
```
0.5463235294117648
```

5 Paired T-Test

[211]:
```
print(stats.ttest_rel(cv_results_fscore, cv_results_rfi))
print(stats.ttest_rel(cv_results_mutual_info, cv_results_rfi))
print(stats.ttest_rel(cv_results_mutual_info, cv_results_fscore))
```

```
Ttest_relResult(statistic=2.014561350514188, pvalue=0.05832635473073461)
Ttest_relResult(statistic=1.8276667316931274, pvalue=0.08334819806466919)
Ttest_relResult(statistic=-0.4175067982745951, pvalue=0.6809877848731882)
```

[212]:
```
print(stats.ttest_rel(cv_results_fscore, cv_results_full))
```

```
Ttest_relResult(statistic=1.7772251837139819, pvalue=0.0915437103875662)
```

6 Convert to CSV

[188]:
```
data.iloc[:, list(indices_fscore[:10]) + [num_features]].to_csv(path_or_buf="../
↪Dataset/fe_leaf.csv", index= False)
```