lab5-wp871q

May 19, 2024

#

Kiss Dániel Márk

##

WP871Q

1 Library import

[1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

2 Loading train set

[2]:
```
df_verseny_public_train = pd.read_csv('data/verseny_public_train.csv', sep=',',␣
 ↪low_memory=False)
```

3 Remove missing values

[3]:
```
df_verseny_public_train = df_verseny_public_train.dropna()
```

4 Selecting columns with the highest variance in the training set

[4]:
```
df_verseny_public_train.var().sort_values(ascending=False)
```

[4]:
```
cookie_id       8.333417e+08
Topic63_ec      6.155122e+03
Topic52_ec      4.236199e+03
Topic42_ec      3.855324e+03
Topic33_ec      3.570435e+03
                    …
Topic173_ic     0.000000e+00
```

```
Topic171_ec    0.000000e+00
Topic171_ic    0.000000e+00
Topic170_ec    0.000000e+00
Topic170_ic    0.000000e+00
Length: 258, dtype: float64
```

[5]:

```
y = df_verseny_public_train['target']
var10 = df_verseny_public_train[df_verseny_public_train.var().
 ↪sort_values(ascending=False).index[:10]]
var20 = df_verseny_public_train[df_verseny_public_train.var().
 ↪sort_values(ascending=False).index[:20]]
var50 = df_verseny_public_train[df_verseny_public_train.var().
 ↪sort_values(ascending=False).index[:50]]
var100 = df_verseny_public_train[df_verseny_public_train.var().
 ↪sort_values(ascending=False).index[:100]]
```

[6]:

```
X10 = var10.drop(['cookie_id'], axis=1)
X20 = var20.drop(['cookie_id'], axis=1)
X50 = var50.drop(['cookie_id'], axis=1)
X100 = var100.drop(['cookie_id'], axis=1)
```

[10]:

```
X_train10, X_test10, y_train10, y_test10 = train_test_split(X10, y, test_size=0.
 ↪2, random_state=42)
X_train20, X_test20, y_train20, y_test20 = train_test_split(X20, y, test_size=0.
 ↪2, random_state=42)
X_train50, X_test50, y_train50, y_test50 = train_test_split(X50, y, test_size=0.
 ↪2, random_state=42)
X_train100, X_test100, y_train100, y_test100 = train_test_split(X100, y,␣
 ↪test_size=0.2, random_state=42)
```

5 Decision tree

[8]:

```
clf10 = DecisionTreeClassifier(random_state=42)
clf20 = DecisionTreeClassifier(random_state=42)
clf50 = DecisionTreeClassifier(random_state=42)
clf100 = DecisionTreeClassifier(random_state=42)

clf10.fit(X_train10, y_train10)
clf20.fit(X_train20, y_train20)
clf50.fit(X_train50, y_train50)
clf100.fit(X_train100, y_train100)

y_pred10 = clf10.predict(X_test10)
y_pred20 = clf20.predict(X_test20)
y_pred50 = clf50.predict(X_test50)
y_pred100 = clf100.predict(X_test100)
```

```
accuracy10 = accuracy_score(y_test10, y_pred10)
accuracy20 = accuracy_score(y_test20, y_pred20)
accuracy50 = accuracy_score(y_test50, y_pred50)
accuracy100 = accuracy_score(y_test100, y_pred100)

print('Decision Tree Classifier')
print('Accuracy for 10 features: ', accuracy10)
print('Accuracy for 20 features: ', accuracy20)
print('Accuracy for 50 features: ', accuracy50)
print('Accuracy for 100 features: ', accuracy100)
```

```
Decision Tree Classifier
Accuracy for 10 features:  0.97335
Accuracy for 20 features:  0.9715
Accuracy for 50 features:  0.96815
Accuracy for 100 features:  0.96775
```

```
[11]: column_names = list(X10.columns)
```

```
[12]: list(column_names)
```

```
[12]: ['Topic63_ec',
       'Topic52_ec',
       'Topic42_ec',
       'Topic33_ec',
       'Topic5_ec',
       'Topic8_ec',
       'Topic19_ec',
       'Topic4_ec',
       'Topic13_ec']
```

6 Loading test set

```
[15]: df_verseny_public_test = pd.read_csv('data/verseny_public_test.csv', sep=',',␣
        ↪low_memory=False)
```

```
[16]: X_test = df_verseny_public_test.drop(['cookie_id'], axis=1)
      X_test = X_test[column_names]
```

```
[17]: y_pred_df = clf10.predict_proba(X_test)[:,1]

      df_verseny_public_test['target'] = y_pred_df

      df_verseny_public_test = df_verseny_public_test[['cookie_id', 'target']]
```

```
df_verseny_public_test.to_csv('data/lab5.csv', index=False)
```

7 Public score: 0.38