Wrapper Methods

May 23, 2021

1 1 - Forward Feature Selection

```python
[2]: # Forward Feature Selection
     from mlxtend.feature_selection import SequentialFeatureSelector
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression


     dataset = pd.read_csv('dataset1.csv')
     X= dataset.drop(columns='Result')
     Y= dataset['Result']
     dataset.head()
```

```
[2]:    Links_in_tags  Abnormal_URL  Submitting_to_email  SFH  …  on_mouseover
     RightClick  Redirect  Result
     0              0            -1                   1    1  …             1
     1        -1        1
     1              1            -1                   1    1  …             1
     1        -1        1
     2              1            -1                   1    1  …             1
     1         0        1
     3             -1            -1                   1   -1  …            -1
     -1        -1       -1
     4              0            -1                   1   -1  …            -1
     -1        -1       -1

     [5 rows x 10 columns]
```

```python
[12]: X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2)

      lr = LogisticRegression(class_weight='balanced', solver='lbfgs',␣
       ↪random_state=42, n_jobs=-1, max_iter=500)
      ffs = SequentialFeatureSelector(lr, k_features='best', forward=True, n_jobs=-1)
      ffs.fit(X,Y)
      features = list(ffs.k_feature_names_)
      features = list(map(str, features))
```

```
print(features)
```

```
['Links_in_tags', 'Abnormal_URL', 'SFH', 'Iframe', 'on_mouseover', 'RightClick',
'Redirect']
```

Above algorithm have drop two features - Submitting_to_email - popUpWidnow

[ ]:

[ ]:

2 2 - Backward Feature Selection

```
[15]: # Backward Feature Selection
      from mlxtend.feature_selection import SequentialFeatureSelector
      import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.linear_model import LogisticRegression


      dataset = pd.read_csv('dataset1.csv')
      X= dataset.drop(columns='Result')
      Y= dataset['Result']
      # X.head()

      lr = LogisticRegression(class_weight='balanced', solver='lbfgs',␣
       ↪random_state=42, n_jobs=-1, max_iter=500)
      bfs = SequentialFeatureSelector(lr, k_features='best', forward=False, n_jobs=-1)
      bfs.fit(X,Y)
      features = list(bfs.k_feature_names_)
      features = list(map(str, features))

      print(features)

      # lr.fit(X_train[features], Y_train)
      # y_pred = lr.predict(X_train[features])
      # print(y_pred)
```

```
['Links_in_tags', 'Abnormal_URL', 'SFH', 'on_mouseover', 'RightClick',
'Redirect']
```

Above algorithm have drop 3 features - Submitting_to_email - Iframe - popUpWidnow

[ ]:

[ ]:

3 3 - Exhaustive Feature Selection

```
[17]:  # Exhaustive Feature Selection
       from mlxtend.feature_selection import ExhaustiveFeatureSelector
       from sklearn.model_selection import train_test_split
       from sklearn.ensemble import RandomForestClassifier
       import pandas as pd

       dataset = pd.read_csv('dataset1.csv')
       X= dataset.drop(columns='Result')
       Y= dataset['Result']

       X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2)


       # create the exhaustive feature selector object
       efs = ExhaustiveFeatureSelector(RandomForestClassifier(),
                                       min_features=4,
                                       max_features=8,
                                       scoring='roc_auc',
                                       cv=2)

       efs = efs.fit(X,Y)

       selected_features = X_train.columns[list(efs.best_idx_)]
       print(selected_features)
```

Features: 381/381

```
Index(['Links_in_tags', 'Abnormal_URL', 'SFH', 'on_mouseover', 'RightClick',
       'Redirect'],
      dtype='object')
```

Above algorithm have drop 3 features - Submitting_to_email - Iframe - popUpWidnow

```
[ ]:
```

```
[ ]:
```

4 4 - Recursive Feature Elimination

```
[19]:  #Recursive Feature Elimination
       import pandas as pd
       from sklearn.feature_selection import RFE

       dataset = pd.read_csv('dataset1.csv')
       X= dataset.drop(columns='Result')
       Y= dataset['Result']
```

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2)

rfe = RFE(lr, n_features_to_select=7) #choose number of columns to be selected
rfe = rfe.fit(X_train,Y_train)

# summarize all features
for i in range(X.shape[1]):
        print('Column: %d, Selected %s, Rank: %.3f col_name: %s' % (i, rfe.
 ↪support_[i], rfe.ranking_[i], X_train.columns[i]))
```

```
Column: 0, Selected True, Rank: 1.000 col_name: Links_in_tags
Column: 1, Selected True, Rank: 1.000 col_name: Abnormal_URL
Column: 2, Selected True, Rank: 1.000 col_name: Submitting_to_email
Column: 3, Selected False, Rank: 3.000 col_name: SFH
Column: 4, Selected True, Rank: 1.000 col_name: Iframe
Column: 5, Selected True, Rank: 1.000 col_name: popUpWidnow
Column: 6, Selected True, Rank: 1.000 col_name: on_mouseover
Column: 7, Selected True, Rank: 1.000 col_name: RightClick
Column: 8, Selected False, Rank: 2.000 col_name: Redirect
```

Above algorithm have drop 2 features - SFH - Redirect