ml-assignment-3

May 17, 2023

[76]:
```
# Heart Disease dataset
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('heart_disease_uci.csv')
df
```

[76]:

	id	age	sex	dataset	cp	trestbps	chol	fbs	\
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True	
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False	
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False	
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False	
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False	
..	…	…	…	…	…	…	…	…	
915	916	54	Female	VA Long Beach	asymptomatic	127.0	333.0	True	
916	917	62	Male	VA Long Beach	typical angina	NaN	139.0	False	
917	918	55	Male	VA Long Beach	asymptomatic	122.0	223.0	True	
918	919	58	Male	VA Long Beach	asymptomatic	NaN	385.0	True	
919	920	62	Male	VA Long Beach	atypical angina	120.0	254.0	False	

	restecg	thalch	exang	oldpeak	slope	ca	\
0	lv hypertrophy	150.0	False	2.3	downsloping	0.0	
1	lv hypertrophy	108.0	True	1.5	flat	3.0	
2	lv hypertrophy	129.0	True	2.6	flat	2.0	
3	normal	187.0	False	3.5	downsloping	0.0	
4	lv hypertrophy	172.0	False	1.4	upsloping	0.0	
..	…	…	…	…	…	…	
915	st-t abnormality	154.0	False	0.0	NaN	NaN	
916	st-t abnormality	NaN	NaN	NaN	NaN	NaN	
917	st-t abnormality	100.0	False	0.0	NaN	NaN	

```
918    lv hypertrophy     NaN    NaN      NaN          NaN  NaN
919    lv hypertrophy    93.0   True      0.0          NaN  NaN

                  thal  num
0         fixed defect    0
1               normal    2
2    reversable defect    1
3               normal    0
4               normal    0
..                 …  …
915             NaN    1
916             NaN    0
917       fixed defect    2
918             NaN    0
919             NaN    1

[920 rows x 16 columns]
```

[77]:
```
df.shape
```

[77]:
```
(920, 16)
```

[78]:
```
df.drop_duplicates
```

[78]:
```
<bound method DataFrame.drop_duplicates of       id  age     sex        dataset
cp  trestbps   chol    fbs  \
0      1   63    Male      Cleveland   typical angina     145.0  233.0   True
1      2   67    Male      Cleveland     asymptomatic     160.0  286.0  False
2      3   67    Male      Cleveland     asymptomatic     120.0  229.0  False
3      4   37    Male      Cleveland      non-anginal     130.0  250.0  False
4      5   41  Female      Cleveland  atypical angina     130.0  204.0  False
..   …  …     …            …               …       …    …    …
915  916   54  Female  VA Long Beach     asymptomatic     127.0  333.0   True
916  917   62    Male  VA Long Beach   typical angina       NaN  139.0  False
917  918   55    Male  VA Long Beach     asymptomatic     122.0  223.0   True
918  919   58    Male  VA Long Beach     asymptomatic       NaN  385.0   True
919  920   62    Male  VA Long Beach  atypical angina     120.0  254.0  False

              restecg  thalch  exang  oldpeak        slope   ca  \
0      lv hypertrophy   150.0  False      2.3  downsloping  0.0
1      lv hypertrophy   108.0   True      1.5         flat  3.0
2      lv hypertrophy   129.0   True      2.6         flat  2.0
3              normal   187.0  False      3.5  downsloping  0.0
4      lv hypertrophy   172.0  False      1.4    upsloping  0.0
..                 …     …    …       …           …  …
915  st-t abnormality   154.0  False      0.0          NaN  NaN
916  st-t abnormality     NaN    NaN      NaN          NaN  NaN
```

```
917  st-t abnormality   100.0  False      0.0          NaN  NaN
918    lv hypertrophy     NaN    NaN      NaN          NaN  NaN
919    lv hypertrophy    93.0   True      0.0          NaN  NaN

                  thal  num
0         fixed defect    0
1               normal    2
2    reversable defect    1
3               normal    0
4               normal    0
..                 …   …
915             NaN       1
916             NaN       0
917       fixed defect    2
918             NaN       0
919             NaN       1

[920 rows x 16 columns]>
```

[79]:
```
df = df.drop(['dataset'], axis=1)
```

[80]:
```
df["num"].value_counts()
```

[80]:
```
0    411
1    265
2    109
3    107
4     28
Name: num, dtype: int64
```

[81]:
```
df = df.dropna()
```

[82]:
```
categorical_cols = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca',␣
 ↪'thal']
df_encoded = pd.get_dummies(df, columns=categorical_cols)
```

[83]:
```
# Split dataset into features (X) and target variable (y)
X = df_encoded.drop('num', axis=1)
y = df_encoded['num']

# Splitting the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)

# Data balancing
oversampler = SMOTE(random_state=42)
X_train_scaled_balanced, y_train_balanced = oversampler.
 ↪fit_resample(X_train_scaled, y_train)

# Model selection and hyperparameter tuning
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 4, 5],
    'learning_rate': [0.1, 0.01, 0.001]
}

grid_search = GridSearchCV(XGBClassifier(random_state=42), param_grid, cv=5)
grid_search.fit(X_train_scaled_balanced, y_train_balanced)
best_model = grid_search.best_estimator_

# Model evaluation
y_pred = best_model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.6333333333333333
              precision    recall  f1-score   support

           0       0.91      0.89      0.90        35
           1       0.31      0.31      0.31        13
           2       0.33      0.40      0.36         5
           3       0.20      0.25      0.22         4
           4       0.00      0.00      0.00         3

    accuracy                           0.63        60
   macro avg       0.35      0.37      0.36        60
weighted avg       0.64      0.63      0.64        60
```

[84]:
```
# Make a correlation matrix
df.corr()
```

[84]:

	id	age	trestbps	chol	thalch	oldpeak	\
id	1.000000	0.001379	-0.021051	-0.138639	-0.159716	-0.091294	
age	0.001379	1.000000	0.286149	0.199258	-0.384176	0.195929	
trestbps	-0.021051	0.286149	1.000000	0.134240	-0.053320	0.191144	
chol	-0.138639	0.199258	0.134240	1.000000	0.014894	0.033964	
thalch	-0.159716	-0.384176	-0.053320	0.014894	1.000000	-0.348089	

```
oldpeak  -0.091294  0.195929  0.191144  0.033964 -0.348089  1.000000
ca        0.020103  0.362764  0.096641  0.121907 -0.256831  0.291958
num       0.031397  0.221787  0.159272  0.065081 -0.416480  0.501325

                ca       num
id        0.020103  0.031397
age       0.362764  0.221787
trestbps  0.096641  0.159272
chol      0.121907  0.065081
thalch   -0.256831 -0.416480
oldpeak   0.291958  0.501325
ca        1.000000  0.520058
num       0.520058  1.000000
```

[85]:
```
corr_matrix=df.corr()

fig, ax = plt.subplots(figsize=(15,10))
ax = sns.heatmap(corr_matrix,
                 annot=True,
                 linewidths=0.5,
                 fmt=".2f",
                 cmap="YlGnBu");
```