linear_regression_v1_1

October 5, 2022

[ ]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import r2_score
from statsmodels.tools.eval_measures import stde
```

[ ]:
```
df_info = pd.read_csv('../dataset_clean/options_csv_v1_etl.csv')
df_info
```

[ ]:
```
   generic_features  remove_atypical_values  feature_combination  \
0             False                   False                False

   remove_feature_selection  remove_time_features  \
0                     False                  True

   remove_invalid_correlated_features
0                               False
```

[ ]:
```
df = pd.read_csv('../dataset_clean/PlatteRiverWeir_features_v1_clean.csv')
df
```

[ ]:
```
        Stage  Discharge  exposure  fNumber  isoSpeed  shutterSpeed  \
0        2.99      916.0  0.000250      4.0       200          -1.0
1        2.99      916.0  0.000312      4.0       200          -1.0
2        2.96      873.0  0.000312      4.0       200          -1.0
3        2.94      846.0  0.000312      4.0       200          -1.0
4        2.94      846.0  0.000312      4.0       200          -1.0
…        …         …      …         …        …         …
42054    2.54      434.0  0.000312      4.0       200          -1.0
42055    2.54      434.0  0.000250      4.0       200          -1.0
```

```
42056   2.54      434.0  0.000250      4.0       200          -1.0
42057   2.54      434.0  0.000312      4.0       200          -1.0
42058   2.54      434.0  0.000400      4.0       200          -1.0

         grayMean  graySigma  entropyMean  entropySigma  …  WeirPt2X  \
0       97.405096  39.623303     0.203417      0.979825  …        -1
1      104.066757  40.179745     0.206835      1.002624  …        -1
2      105.636831  40.533218     0.204756      0.994246  …        -1
3      104.418949  41.752678     0.202428      0.983170  …        -1
4      106.763541  44.442097     0.202661      0.989625  …        -1
…             …         …            …             …  …         …
42054   82.872720  57.702652     0.221708      1.076393  …      2446
42055   89.028383  55.840861     0.233168      1.124774  …      2440
42056   94.722097  54.355753     0.240722      1.151833  …      2447
42057   96.693270  52.787629     0.244789      1.171987  …      2443
42058   98.738399  52.025453     0.252812      1.213278  …      2436

       WeirPt2Y  WwRawLineMin  WwRawLineMax  WwRawLineMean  WwRawLineSigma  \
0            -1           0.0           0.0       0.000000        0.000000
1            -1           0.0           0.0       0.000000        0.000000
2            -1           0.0           0.0       0.000000        0.000000
3            -1           0.0           0.0       0.000000        0.000000
4            -1           0.0           0.0       0.000000        0.000000
…             …             …             …              …               …
42054      1900        9284.0       77521.0   38385.370066    15952.029728
42055      1900       10092.0       74614.0   40162.989292    15467.708856
42056      1900        7067.0       83260.0   42095.946590    16770.357949
42057      1900        6283.0       83045.0   45345.490954    17498.432849
42058      1900        7375.0       89813.0   47877.870782    19963.166359

       WwCurveLineMin  WwCurveLineMax  WwCurveLineMean  WwCurveLineSigma
0                 0.0             0.0         0.000000          0.000000
1                 0.0             0.0         0.000000          0.000000
2                 0.0             0.0         0.000000          0.000000
3                 0.0             0.0         0.000000          0.000000
4                 0.0             0.0         0.000000          0.000000
…                   …               …              …                 …
42054             0.0         70085.0     37550.894823      16444.401209
42055             0.0         70061.0     39397.339095      16009.008049
42056             0.0         76335.0     41350.006568      17489.374617
42057             0.0         78882.0     44553.920296      18268.294896
42058             0.0         82630.0     47280.270559      20559.358767

[42059 rows x 50 columns]
```

[ ]:
```
y = df[["Stage", "Discharge"]]
X = df.drop(columns=["Stage", "Discharge"])
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,␣
↪random_state=0)
```

[ ]:
```
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('model', LinearRegression())
])

folds = KFold(n_splits = 5, shuffle = True, random_state = 100)
clf = cross_val_score(pipeline, X_train, y_train, scoring='r2', cv=folds)
```

[ ]:
```
clf
```

[ ]:
```
array([0.62272775, 0.6248884 , 0.62869041, 0.6165668 , 0.62798439])
```

[ ]:
```
pipeline.fit(X_train, y_train)
```

[ ]:
```
Pipeline(steps=[('scaler', StandardScaler()), ('model', LinearRegression())])
```

[ ]:
```
y_pred = pipeline.predict(X_test)
```

[ ]:
```
print("R^2: ", r2_score(y_test, y_pred))
print("Error estandar: ", stde(y_test.squeeze(), y_pred.squeeze(), ddof = len(X.
↪columns) + 1))
```

```
R^2:  0.6315035804849779
Error estandar:  [4.68950004e-01 7.66236100e+02]
```

[ ]:
```
residuals = y_test - y_pred
residuals
```

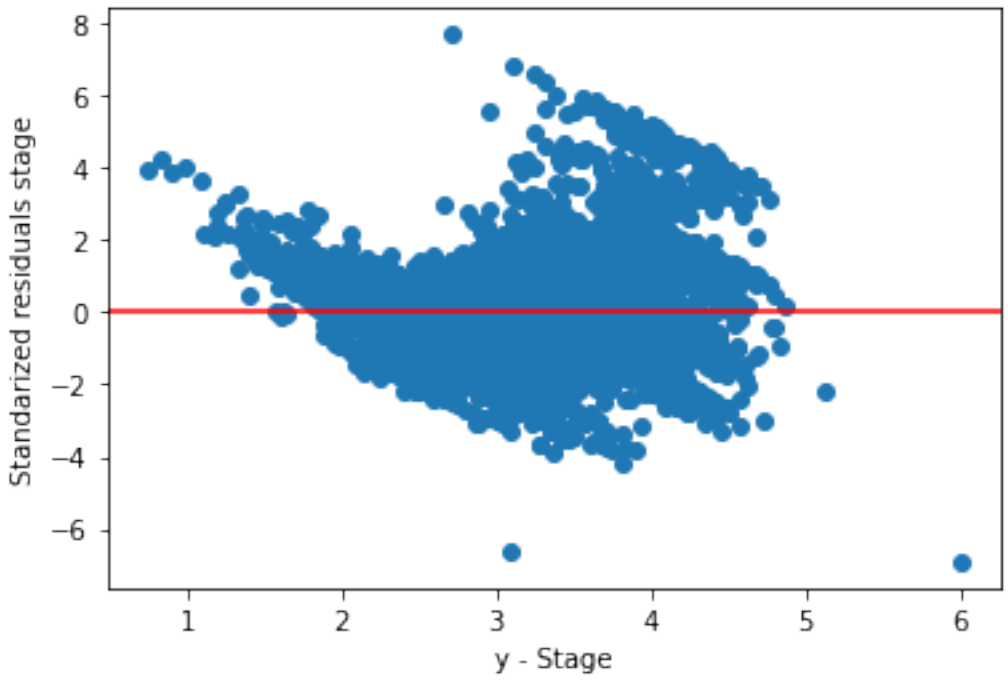
[ ]:
```
           Stage    Discharge
2714    0.163924   362.236901
6409   -0.077983    22.431331
23395   0.597619   771.641748
3335   -1.806253 -2416.449595
31874 -0.391807  -515.053372
…          …           …
11619   0.047276    29.679963
4541   -0.115972  -232.900402
37056   0.093463   110.414510
34059   0.108842    45.911475
29120   0.249453   502.533604

[13880 rows x 2 columns]
```

[ ]:
```
resid = np.array(residuals["Stage"])
norm_resid = resid / resid.std()

plt.scatter([i[0] for i in y_pred], norm_resid)
plt.axhline(y = 0.0, color = 'r', linestyle = '-')
plt.xlabel("y - Stage")
plt.ylabel("Standarized residuals stage")
```

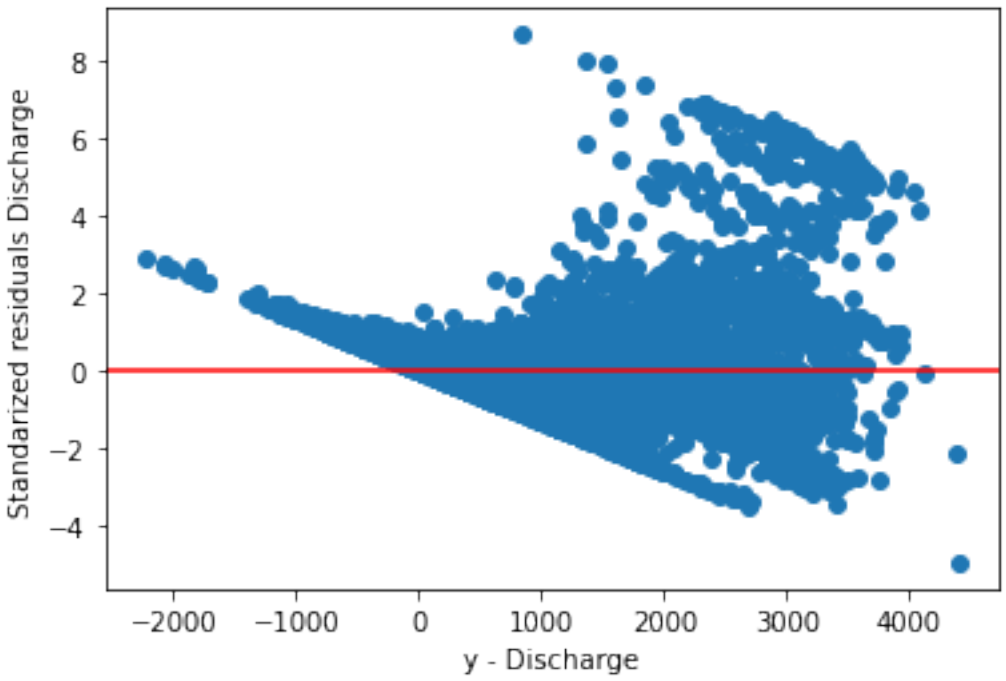
[ ]:
```
Text(0, 0.5, 'Standarized residuals stage')
```



[ ]:
```
resid = np.array(residuals["Discharge"])
norm_resid = resid / resid.std()

plt.scatter([i[1] for i in y_pred], norm_resid)
plt.axhline(y = 0.0, color = 'r', linestyle = '-')
plt.xlabel("y - Discharge")
plt.ylabel("Standarized residuals Discharge")
```

[ ]:
```
Text(0, 0.5, 'Standarized residuals Discharge')
```


Standarized residuals Discharge
8
6
4
2
0
−2
−4
−2000
−1000
0
1000
2000
3000
4000
y - Discharge

[ ]: