Feature Importance

May 20, 2021

[2]:
```
# test classification dataset
from sklearn.datasets import make_classification
# define dataset
X, y = make_classification(n_samples=1000, n_features=9, n_informative=5,␣
 ↪n_redundant=4, random_state=1)
# summarize the dataset
print(X.shape, y.shape)
```

```
(1000, 9) (1000,)
```

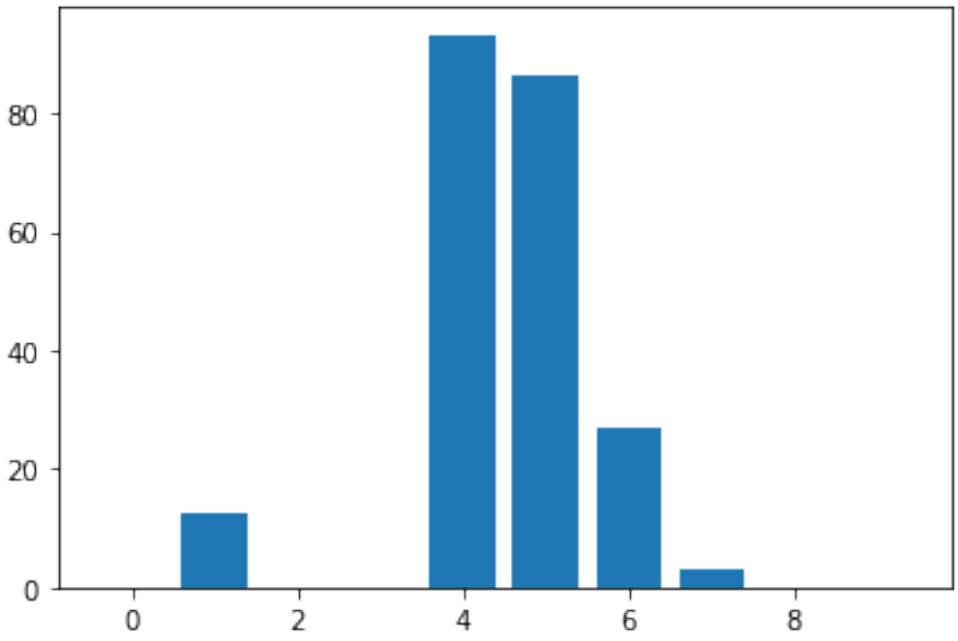
[5]:
```
# test regression dataset
from sklearn.datasets import make_regression
# define dataset
X, y = make_regression(n_samples=1000, n_features=9, n_informative=5,␣
 ↪random_state=1)
# summarize the dataset
print(X.shape, y.shape)
```

```
(1000, 9) (1000,)
```

[6]:
```
# linear regression feature importance
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot
# define dataset
X, y = make_regression(n_samples=1000, n_features=10, n_informative=5,␣
 ↪random_state=1)
# define the model
model = LinearRegression()
# fit the model
model.fit(X, y)
# get importance
importance = model.coef_
# summarize feature importance
for i,v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
```

```
pyplot.show()
```

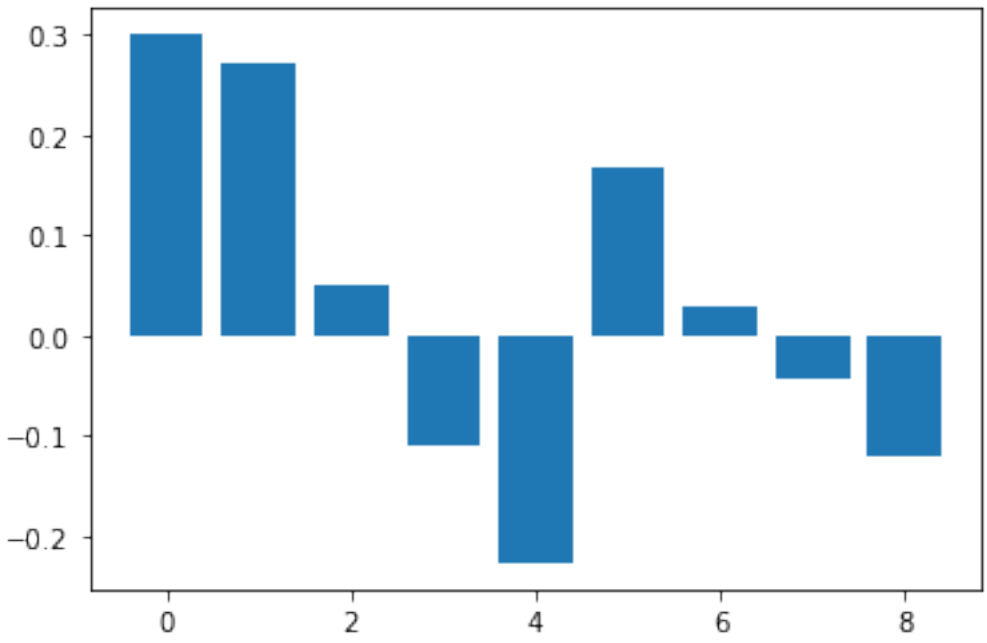
```
Feature: 0, Score: -0.00000
Feature: 1, Score: 12.44483
Feature: 2, Score: 0.00000
Feature: 3, Score: -0.00000
Feature: 4, Score: 93.32225
Feature: 5, Score: 86.50811
Feature: 6, Score: 26.74607
Feature: 7, Score: 3.28535
Feature: 8, Score: 0.00000
Feature: 9, Score: -0.00000
```



[7]:
```
# linear regression feature importance
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot
import pandas as pd
from sklearn.model_selection import train_test_split

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

# define dataset
```

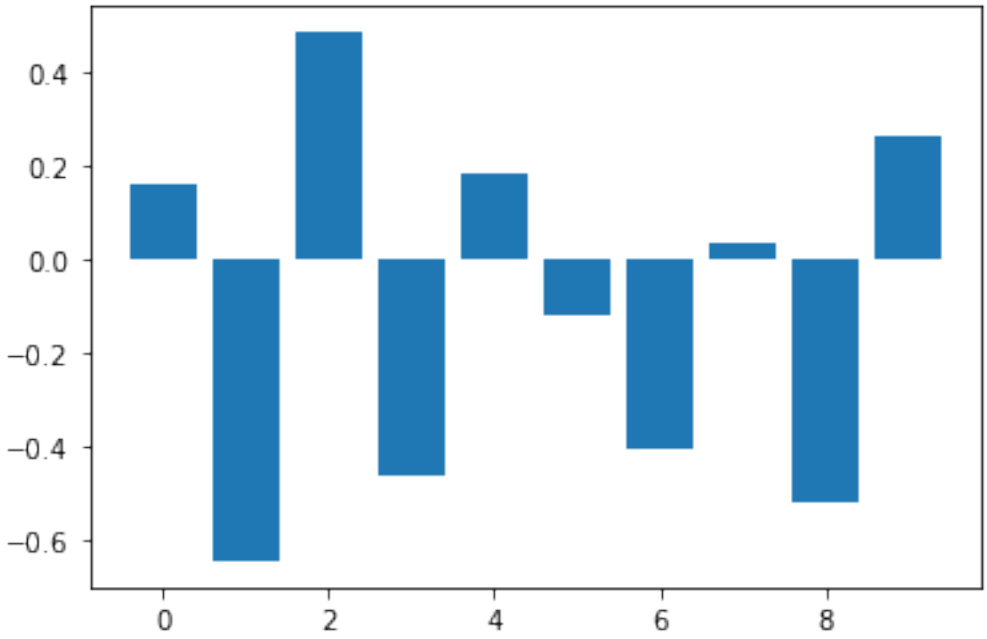
```
# X, y = make_regression(n_samples=1000, n_features=10, n_informative=5,␣
 ↪random_state=1)

# define the model
model = LinearRegression()
# fit the model
model.fit(X, Y)
# get importance
importance = model.coef_
# summarize feature importance
for i,v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```

```
Feature: 0, Score: 0.29977
Feature: 1, Score: 0.26998
Feature: 2, Score: 0.05019
Feature: 3, Score: -0.10827
Feature: 4, Score: -0.22632
Feature: 5, Score: 0.16645
Feature: 6, Score: 0.02907
Feature: 7, Score: -0.04217
Feature: 8, Score: -0.11890
```



[8]:
```
# logistic regression for feature importance
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from matplotlib import pyplot
# define dataset
X, y = make_classification(n_samples=1000, n_features=10, n_informative=5,␣
 ↪n_redundant=5, random_state=1)
# define the model
model = LogisticRegression()
# fit the model
model.fit(X, y)
# get importance
importance = model.coef_[0]
# summarize feature importance
for i,v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```

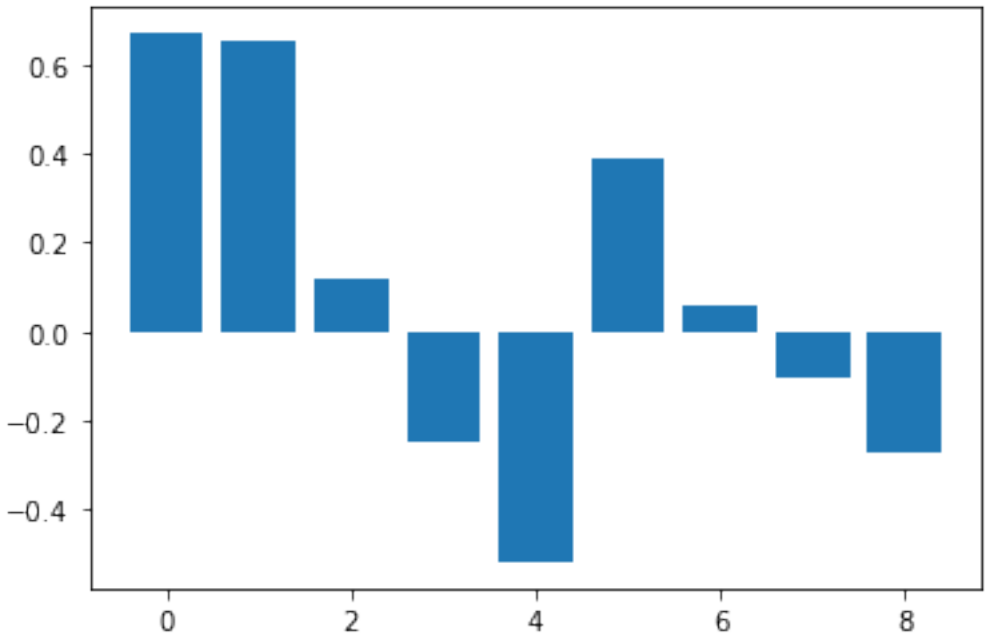
```
Feature: 0, Score: 0.16320
Feature: 1, Score: -0.64301
Feature: 2, Score: 0.48497
Feature: 3, Score: -0.46190
Feature: 4, Score: 0.18432
Feature: 5, Score: -0.11978
Feature: 6, Score: -0.40602
Feature: 7, Score: 0.03772
Feature: 8, Score: -0.51785
Feature: 9, Score: 0.26540
```



[9]:
```
# logistic regression for feature importance
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from matplotlib import pyplot
import pandas as pd
from sklearn.model_selection import train_test_split

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


# define dataset
# X, y = make_classification(n_samples=1000, n_features=10, n_informative=5,␣
 ↪n_redundant=5, random_state=1)
# define the model
model = LogisticRegression()
# fit the model
model.fit(X, Y)
# get importance
importance = model.coef_[0]
# summarize feature importance
for i,v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i,v))
```

```
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```

```
Feature: 0, Score: 0.67041
Feature: 1, Score: 0.65376
Feature: 2, Score: 0.11908
Feature: 3, Score: -0.24866
Feature: 4, Score: -0.51892
Feature: 5, Score: 0.39163
Feature: 6, Score: 0.05648
Feature: 7, Score: -0.10398
Feature: 8, Score: -0.27460
```



[ ]: