Feature importance and weight determination

May 23, 2021

Author: Lakruwan

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction.

Accordigly we can assign a weight to each feature according to the importance score.

[6]:
```
import pandas as pd

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

[6]:

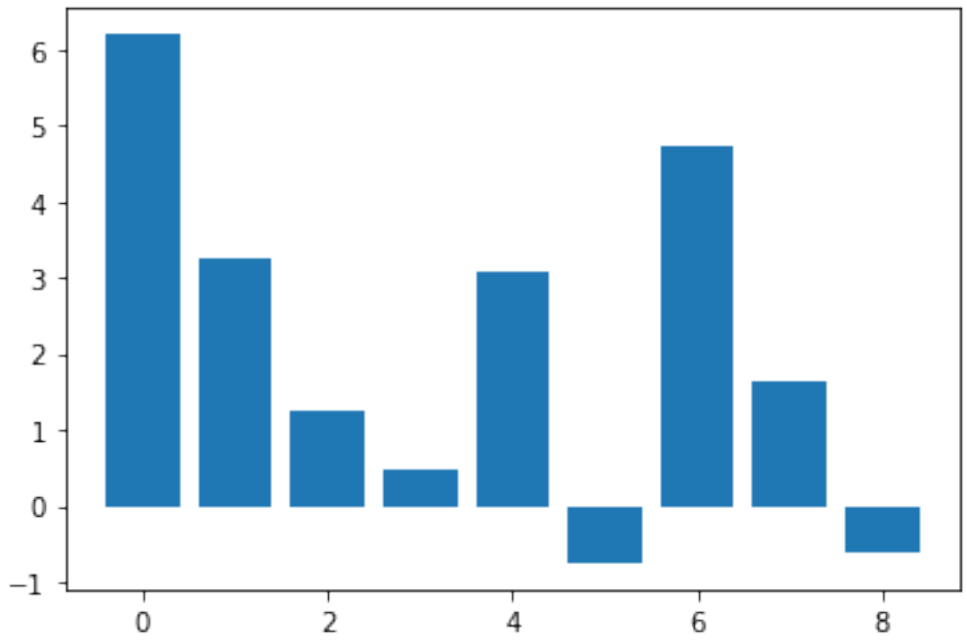
	Links_in_tags	Abnormal_URL	Submitting_to_email	SFH	Iframe	popUpWidnow	on_mouseover	RightClick	Redirect	Result
0	0	-1	1	1	-1	1	1	1	-1	1
1	1	-1	1	1	-1	1	1	1	-1	1
2	1	-1	1	1	-1	1	1	1	0	1
3	-1	-1	1	-1	1	-1	-1	-1	-1	-1
4	0	-1	1	-1	1	-1	-1	-1	-1	-1

[13]:
```
from sklearn.linear_model import LinearRegression
from matplotlib import pyplot

# 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*10))

# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance*10)
```

```
Feature: 0, Score: 6.19696
Feature: 1, Score: 3.23861
Feature: 2, Score: 1.26566
Feature: 3, Score: 0.47464
Feature: 4, Score: 3.07952
Feature: 5, Score: -0.75216
Feature: 6, Score: 4.72490
Feature: 7, Score: 1.64888
Feature: 8, Score: -0.59961
```

[13]: `<BarContainer object of 9 artists>`



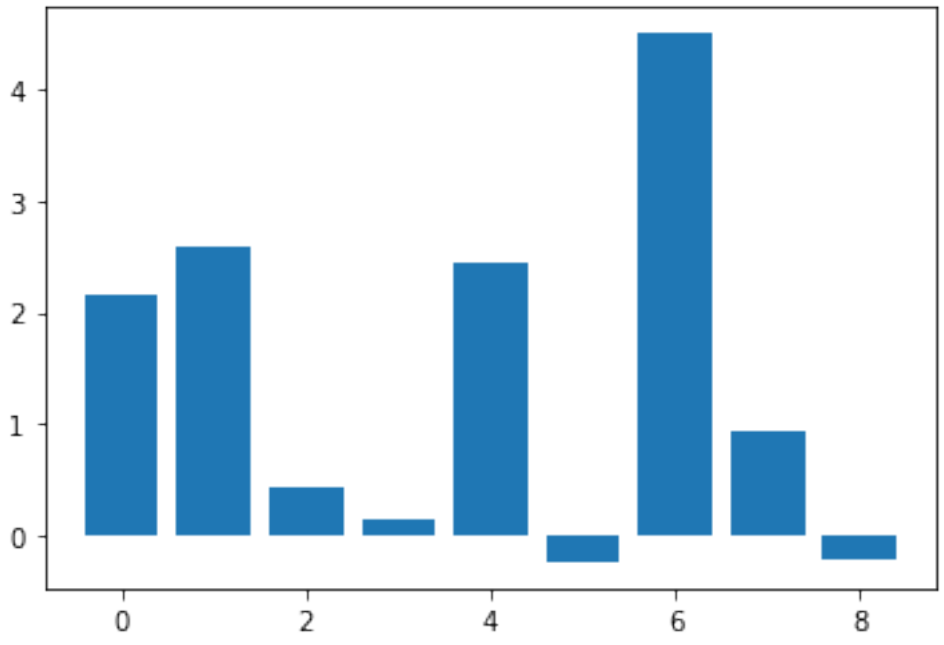
[ ]:

[ ]:

[15]:
```
from sklearn.linear_model import LogisticRegression
```

```
# 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: 2.15227
Feature: 1, Score: 2.59256
Feature: 2, Score: 0.43173
Feature: 3, Score: 0.14246
Feature: 4, Score: 2.45339
Feature: 5, Score: -0.24258
Feature: 6, Score: 4.50383
Feature: 7, Score: 0.93503
Feature: 8, Score: -0.22967
```

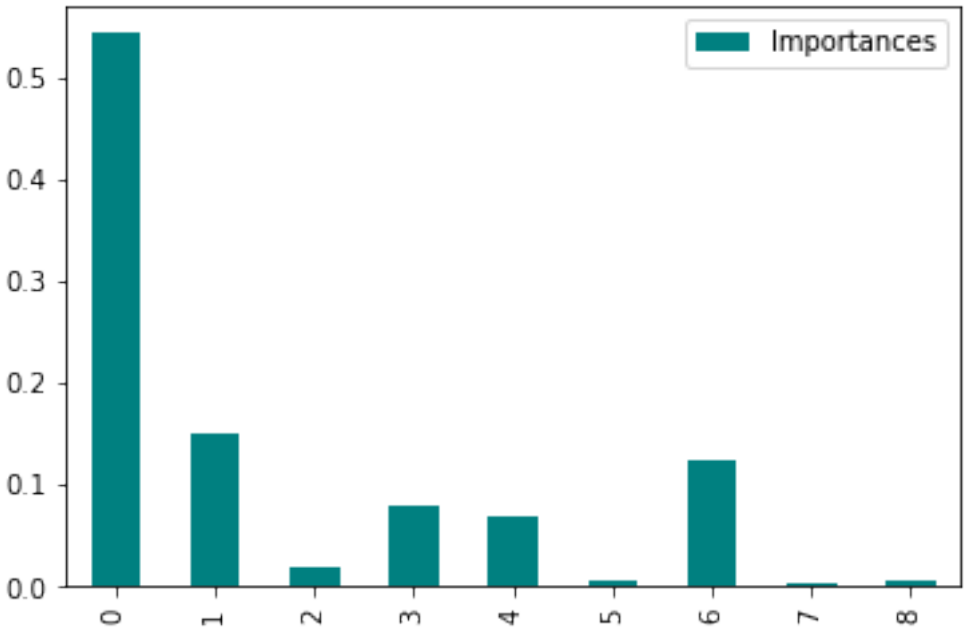


[ ]:

[ ]:

[3]:
```
#Random Forest Importance
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
from sklearn.model_selection import train_test_split
import pandas as pd

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

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

model = RandomForestClassifier(n_estimators=340)
model.fit(X_train,Y_train)

# get the importance of resulting features
importances = model.feature_importances_

# create a dataframe for vissualization
final_df = pd.DataFrame({"Features": pd.DataFrame(X).columns, "Importances":
 ↪importances})
final_df.set_index('Importances')

# sort in ascending order for better vissualization
# final_df = final_df.sort_values('Importances')

final_df.plot.bar(color='teal')
```

[3]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d3ba8a6cc8>
```



[4]:
```
# summarize feature importance
for i,v in enumerate(importances):
    print('Feature: %0d, Score: %.5f' % (i,v))
```

```
Feature: 0, Score: 0.54267
Feature: 1, Score: 0.15002
Feature: 2, Score: 0.02037
Feature: 3, Score: 0.07912
Feature: 4, Score: 0.06850
Feature: 5, Score: 0.00628
Feature: 6, Score: 0.12435
Feature: 7, Score: 0.00248
Feature: 8, Score: 0.00620
```