

```
In [ ]: !jupyter-nbconvert --to PDFviaHTML example.ipynb
```

```
In [12]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt
```

```
In [19]: df = pd.read_csv('C:\\Users\\sahil\\OneDrive\\Desktop\\Fall 2020\\Security Analytics\\wcl.csv')
pd.set_option('display.max_columns', None)
df.head()
```

```
Out[19]:
```

	inline_count	external_count	onclick_count	onload_count	onchange_count	avg_inline_script_block	avg_external_script_block	avg_onc
0	21.0	23.0	1	131.0	0.0	0.0	662.062500	
1	13.0	30.0	2	4.0	1.0	0.0	55.777778	
2	0.0	3.0	1	0.0	0.0	0.0	207.000000	
3	21.0	11.0	1	10.0	1.0	0.0	104.800000	
4	10.0	5.0	1	0.0	0.0	0.0	473.000000	

```
In [20]: X_train, X_test, y_train, y_test = train_test_split(
df.drop('type', axis=1), df['type'],
test_size=0.33, random_state=133)
```

```
In [23]: # Initialize and train classifier model
clf = LogisticRegression().fit(X_train, y_train)

# Make predictions on test set
y_pred = clf.predict(X_test)
y_score2 = clf.predict_proba(X_test)[:,1]
```

```
C:\Users\sahil\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning:
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

```
In [24]: # Compare test set predictions with ground truth labels
print(accuracy_score(y_pred, y_test))
print(confusion_matrix(y_test, y_pred))
```

```
0.8105539577341503
```

```
[[5532  273]
```

```
 [1242  950]]
```

```
In [25]: df.columns
print(clf.coef_[0])
```

```
[-9.37131471e-03 -2.87062772e-02  5.23397237e-02 -7.16068137e-03
 3.94834331e-02  3.49383416e-04 -8.35559335e-04  1.08404788e-05
 3.71928654e-02 -9.66807556e-02 -1.64175196e-02 -1.06295850e-01]
```