15ME31001

Piyush Khushlani

Assignment 8

Problem Statement

A group of researchers have identified different features related to legitimate and phishy websites and collected 1353 different websites from difference sources. Phishing websites were collected from Phishtank data archive (www.phishtank.com). We have to Classify the 1353 websites with 702 as phishing websites, 103 as suspicious URLs and 548 as legitimate websites. The goal of this assignment is to build a classifier to detect phising/malicious web pages.

Methodology

Preprocessing

No preprocessing has been done any more as the attributes in the data are already in their normalized or standardised form.
Moreover, we shuffled and divided the data into 81% train and 9% validation and 10% test data.

Learning Algorithm and Evaluation Strategy

We used `MLPClassifier` from `sklearn.neural_network` package in which was initialized by the following arguments:

```
mlp = MLPClassifier(hidden_layer_sizes=(100, ), activation
='relu', solver='adam', alpha=0.0001, batch_size='auto', l
earning_rate='constant', learning_rate_init=0.1, power_t=0
.5, max_iter=200, shuffle=True, random_state=None, tol=0.0
001, verbose=False, warm_start=False, momentum=0.9, nester
ovs_momentum=True, early_stopping=False, validation_fracti
on=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no
_change=10)
```

We then trained and evaluated the model by:

```
mlp.fit(X_train, Y_train)
mlp.score(X_test,Y_test)
```

and `score` function the mean accuracy on the given test data and labels.
Here, `X_train` and `Y_train` are training inputs and outputs respectively. Similarly for testing Inputs and Outputs.

Experimental Results

The following is the output when tested on different Number of Iterations for the Neural Network w.r.t. the Testing Accuracy:

> *Note: The results can be different for different runs of the Input training/testing Data.*
> *Results are for just a random 1 run of the Program.*

Num_iterations	Testing_Accuracy
1	0.867
2	0.845
3	0.845
4	0.912
5	0.882
6	0.831
7	0.889
8	0.831
9	0.868

Discussion of the Results

With each run, as the data gets shuffled, it shuffles the legitimate, suspicious and phishing URLs which get in the input for training and validation and the test results are thus different. Moreover we can use `random.seed()` if we wish to remove the difference in the results with each run.