Name: Jimenez, Jerviz Mico A. Course and Section: CPE32S3 Date Submitted: 03/26/2024
Instructor: Engr. Roman Richard

In this assignment, you are task to build a multilayer perceptron model. The following are the requirements:

1. Choose any dataset
2. Explain the problem you are trying to solve
3. Create your own model
4. Evaluate the accuracy of your model

1. Choosing the dataset

The dataset used for this activity was acquired at https://archive.ics.uci.edu/dataset/379/website+phishing. This dataset includes features that shows if a website is a phishing website. There are 19 different features that is used to compare with the URL to determine if it is a fake.

Abdelhamid,Neda. (2016). Website Phishing. UCI Machine Learning Repository. https://doi.org/10.24432/C5B301.

1. Problem

This assignment has tasked us to build and apply Multilaayer Perceptron on our dataset. The model created would be used to predict if a URL is a phishing website or not. The accuracy of the model is then evaluated to determine if the model fits.

1. Creating the model

```
# --- Importing Libraries ---
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix

# --- Loading the Data ---
data = pd.read_csv('PhishingData.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1353 entries, 0 to 1352
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  1353 non-null   int64
```

```
 1   SFH                 1353 non-null   int64
 2   popUpWidnow         1353 non-null   int64
 3   SSLfinal_State      1353 non-null   int64
 4   Request_URL         1353 non-null   int64
 5   URL_of_Anchor       1353 non-null   int64
 6   web_traffic         1353 non-null   int64
 7   URL_Length          1353 non-null   int64
 8   age_of_domain       1353 non-null   int64
 9   having_IP_Address   1353 non-null   int64
 10  Result              1353 non-null   int64
dtypes: int64(11)
memory usage: 116.4 KB
```

```
data.head(10)
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 1353,\n  \"fields\":
[\n    {\n      \"column\": \"id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 390,\n        \"min\": 1,\n
\"max\": 1353,\n        \"num_unique_values\": 1353,\n
\"samples\": [\n          50,\n          639,\n          1034\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"SFH\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\":
-1,\n        \"max\": 1,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1,\n          -1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"popUpWidnow\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": -1,\n        \"max\": 1,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          -1,\n
0,\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"SSLfinal_State\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": -1,\n
\"max\": 1,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          1,\n          -1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Request_URL\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": -1,\n        \"max\": 1,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          -1,\n
0,\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"URL_of_Anchor\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": -1,\n
\"max\": 1,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          -1,\n          0,\n          1\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"web_traffic\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
0,\n        \"min\": -1,\n        \"max\": 1,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          1,\n
0,\n          -1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"URL_Length\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": -1,\n
\"max\": 1,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          1,\n          -1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"age_of_domain\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": -1,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          -1,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"having_IP_Address\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Result\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": -1,\n
\"max\": 1,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```


Observation:

The results show 3 different which are -1, 0 and 1. The dataset included URL that are fake(-1), Legitimate(1), and Unknown(0) meaning the URL have such features that makes it either legitimate or fake.

```
X = data[['SFH', 'popUpWidnow', 'SSLfinal_State', 'Request_URL',
'URL_of_Anchor', 'web_traffic', 'URL_Length',
          'age_of_domain', 'having_IP_Address']]

y = data['Result']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

clf = MLPClassifier(hidden_layer_sizes=(9, 7), random_state=5,
verbose=True, learning_rate_init=0.01)

clf.fit(X_train,y_train)
```

```
Iteration 1, loss = 0.98668412
Iteration 2, loss = 0.87110015
Iteration 3, loss = 0.76651652
```

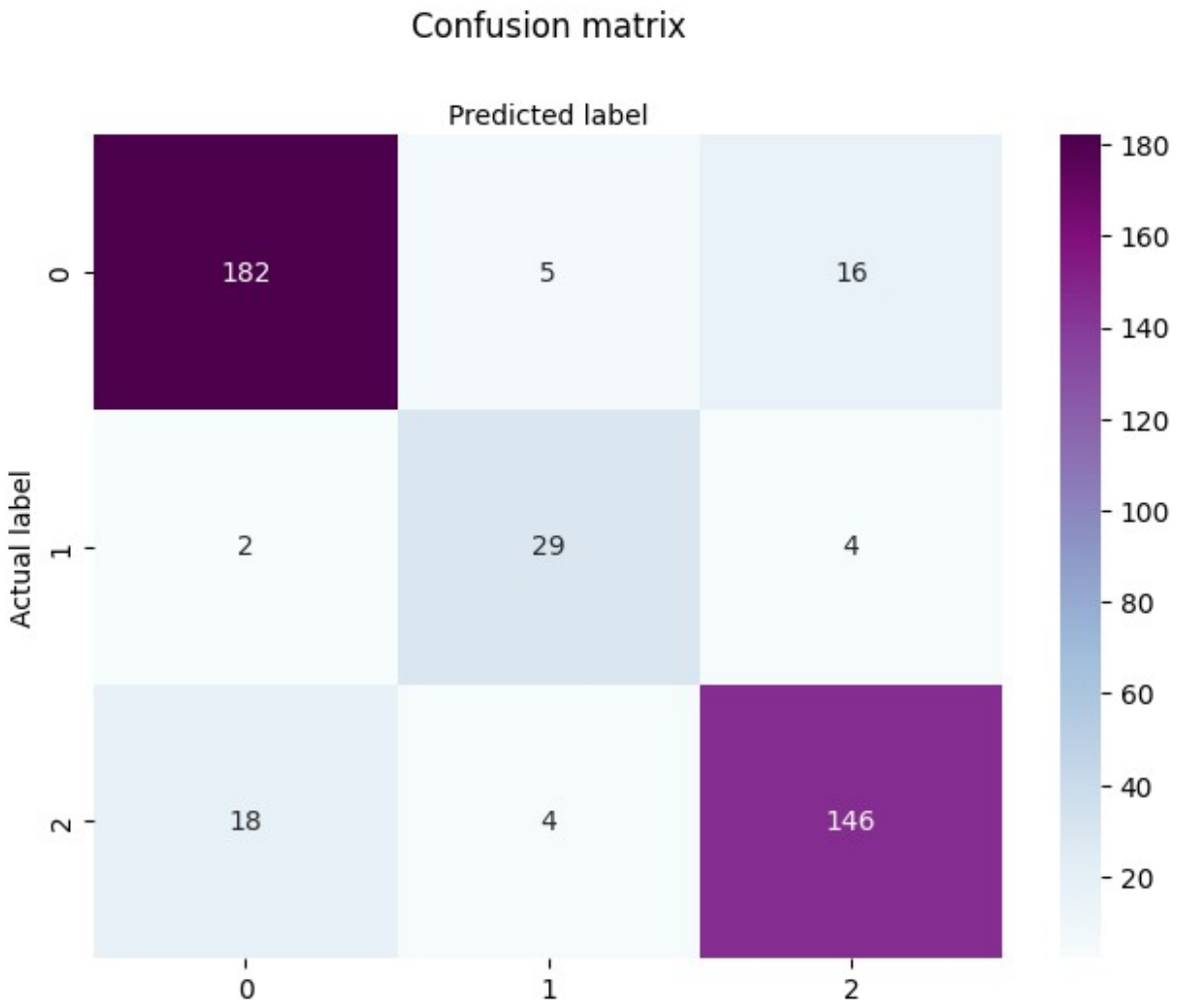
```
Iteration 4, loss = 0.68946969
Iteration 5, loss = 0.62582113
Iteration 6, loss = 0.57298219
Iteration 7, loss = 0.53035188
Iteration 8, loss = 0.49566307
Iteration 9, loss = 0.47107904
Iteration 10, loss = 0.45412725
Iteration 11, loss = 0.44346742
Iteration 12, loss = 0.43567415
Iteration 13, loss = 0.42803444
Iteration 14, loss = 0.42098146
Iteration 15, loss = 0.41474247
Iteration 16, loss = 0.40891269
Iteration 17, loss = 0.40272601
Iteration 18, loss = 0.39713820
Iteration 19, loss = 0.39162139
Iteration 20, loss = 0.38557598
Iteration 21, loss = 0.37938986
Iteration 22, loss = 0.37462380
Iteration 23, loss = 0.36676262
Iteration 24, loss = 0.36094799
Iteration 25, loss = 0.35628559
Iteration 26, loss = 0.35135829
Iteration 27, loss = 0.34472937
Iteration 28, loss = 0.33992640
Iteration 29, loss = 0.33548629
Iteration 30, loss = 0.32957845
Iteration 31, loss = 0.32725410
Iteration 32, loss = 0.32220715
Iteration 33, loss = 0.31699386
Iteration 34, loss = 0.31419584
Iteration 35, loss = 0.31056824
Iteration 36, loss = 0.30675891
Iteration 37, loss = 0.30483743
Iteration 38, loss = 0.30132889
Iteration 39, loss = 0.29908388
Iteration 40, loss = 0.29669183
Iteration 41, loss = 0.29509058
Iteration 42, loss = 0.29178628
Iteration 43, loss = 0.29027154
Iteration 44, loss = 0.29034000
Iteration 45, loss = 0.28569200
Iteration 46, loss = 0.28599611
Iteration 47, loss = 0.28446202
Iteration 48, loss = 0.28041762
Iteration 49, loss = 0.28207106
Iteration 50, loss = 0.27689134
Iteration 51, loss = 0.27578642
Iteration 52, loss = 0.27135520
```

```
Iteration 53, loss = 0.27131181
Iteration 54, loss = 0.26792908
Iteration 55, loss = 0.26570769
Iteration 56, loss = 0.26506263
Iteration 57, loss = 0.26381687
Iteration 58, loss = 0.26307433
Iteration 59, loss = 0.26173122
Iteration 60, loss = 0.25958345
Iteration 61, loss = 0.25891992
Iteration 62, loss = 0.26029230
Iteration 63, loss = 0.25622343
Iteration 64, loss = 0.25582789
Iteration 65, loss = 0.25511661
Iteration 66, loss = 0.25281200
Iteration 67, loss = 0.25168740
Iteration 68, loss = 0.25044904
Iteration 69, loss = 0.25030094
Iteration 70, loss = 0.25018630
Iteration 71, loss = 0.24997127
Iteration 72, loss = 0.24857006
Iteration 73, loss = 0.24663173
Iteration 74, loss = 0.24524792
Iteration 75, loss = 0.24438270
Iteration 76, loss = 0.24602899
Iteration 77, loss = 0.24411354
Iteration 78, loss = 0.24235860
Iteration 79, loss = 0.24165118
Iteration 80, loss = 0.24058787
Iteration 81, loss = 0.24066338
Iteration 82, loss = 0.23895289
Iteration 83, loss = 0.23841259
Iteration 84, loss = 0.23770104
Iteration 85, loss = 0.23517898
Iteration 86, loss = 0.23490997
Iteration 87, loss = 0.23363080
Iteration 88, loss = 0.23265949
Iteration 89, loss = 0.23355502
Iteration 90, loss = 0.23231737
Iteration 91, loss = 0.23066850
Iteration 92, loss = 0.23003258
Iteration 93, loss = 0.23361098
Iteration 94, loss = 0.22871528
Iteration 95, loss = 0.23058712
Iteration 96, loss = 0.23139331
Iteration 97, loss = 0.23232869
Iteration 98, loss = 0.22664673
Iteration 99, loss = 0.22816127
Iteration 100, loss = 0.22510521
Iteration 101, loss = 0.22416310
```

```
Iteration 102, loss = 0.22794814
Iteration 103, loss = 0.22503276
Iteration 104, loss = 0.22460136
Iteration 105, loss = 0.22186996
Iteration 106, loss = 0.21852359
Iteration 107, loss = 0.21945772
Iteration 108, loss = 0.21735414
Iteration 109, loss = 0.21496802
Iteration 110, loss = 0.21397279
Iteration 111, loss = 0.21381717
Iteration 112, loss = 0.21147494
Iteration 113, loss = 0.21061266
Iteration 114, loss = 0.20883220
Iteration 115, loss = 0.20751553
Iteration 116, loss = 0.21004156
Iteration 117, loss = 0.21142989
Iteration 118, loss = 0.20838927
Iteration 119, loss = 0.20404426
Iteration 120, loss = 0.20614670
Iteration 121, loss = 0.20369456
Iteration 122, loss = 0.20399976
Iteration 123, loss = 0.20468695
Iteration 124, loss = 0.20136543
Iteration 125, loss = 0.20312970
Iteration 126, loss = 0.19907483
Iteration 127, loss = 0.19920479
Iteration 128, loss = 0.19748568
Iteration 129, loss = 0.19648946
Iteration 130, loss = 0.19584651
Iteration 131, loss = 0.19438440
Iteration 132, loss = 0.19484385
Iteration 133, loss = 0.19319753
Iteration 134, loss = 0.19329179
Iteration 135, loss = 0.19096529
Iteration 136, loss = 0.19277791
Iteration 137, loss = 0.19371166
Iteration 138, loss = 0.19199827
Iteration 139, loss = 0.19011876
Iteration 140, loss = 0.18989963
Iteration 141, loss = 0.19428270
Iteration 142, loss = 0.19817212
Iteration 143, loss = 0.19020543
Iteration 144, loss = 0.18864406
Iteration 145, loss = 0.18836971
Iteration 146, loss = 0.18512328
Iteration 147, loss = 0.18660557
Iteration 148, loss = 0.18603036
Iteration 149, loss = 0.18365072
Iteration 150, loss = 0.18424002
```

```
Iteration 151, loss = 0.18319671
Iteration 152, loss = 0.18241403
Iteration 153, loss = 0.18544027
Iteration 154, loss = 0.18559454
Iteration 155, loss = 0.18312538
Iteration 156, loss = 0.18337011
Iteration 157, loss = 0.18304189
Iteration 158, loss = 0.18283224
Iteration 159, loss = 0.18598919
Iteration 160, loss = 0.18109933
Iteration 161, loss = 0.18281583
Iteration 162, loss = 0.18175361
Iteration 163, loss = 0.18506248
Iteration 164, loss = 0.18212425
Iteration 165, loss = 0.17762361
Iteration 166, loss = 0.17953253
Iteration 167, loss = 0.17718006
Iteration 168, loss = 0.17789837
Iteration 169, loss = 0.17557623
Iteration 170, loss = 0.17441991
Iteration 171, loss = 0.17454663
Iteration 172, loss = 0.17417425
Iteration 173, loss = 0.17876095
Iteration 174, loss = 0.18022054
Iteration 175, loss = 0.17475405
Iteration 176, loss = 0.17591235
Iteration 177, loss = 0.17597158
Iteration 178, loss = 0.17207142
Iteration 179, loss = 0.17272135
Iteration 180, loss = 0.17158745
Iteration 181, loss = 0.17199999
Iteration 182, loss = 0.17163541
Iteration 183, loss = 0.17041075
Iteration 184, loss = 0.17213985
Iteration 185, loss = 0.17153146
Iteration 186, loss = 0.16906917
Iteration 187, loss = 0.17054172
Iteration 188, loss = 0.17057459
Iteration 189, loss = 0.16776673
Iteration 190, loss = 0.16921500
Iteration 191, loss = 0.17017304
Iteration 192, loss = 0.17065933
Iteration 193, loss = 0.16699130
Iteration 194, loss = 0.16658608
Iteration 195, loss = 0.16612448
Iteration 196, loss = 0.16551648
Iteration 197, loss = 0.16648660
Iteration 198, loss = 0.16314596
```

```
Iteration 199, loss = 0.16440214
Iteration 200, loss = 0.16635320

/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/
_multilayer_perceptron.py:686: ConvergenceWarning: Stochastic
Optimizer: Maximum iterations (200) reached and the optimization
hasn't converged yet.
  warnings.warn(

MLPClassifier(hidden_layer_sizes=(9, 7), learning_rate_init=0.01,
              random_state=5, verbose=True)

matrix = confusion_matrix(y_test, ypred)
print(matrix)
names=[-1,0,1]
fig, ax = plt.subplots()
tick_marks = np.arange(len(names))
plt.xticks(tick_marks,names)
plt.yticks(tick_marks,names)
sns.heatmap(pd.DataFrame(matrix), annot=True, cmap="BuPu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

[[182   5  16]
 [  2  29   4]
 [ 18   4 146]]

Text(0.5, 427.9555555555555, 'Predicted label')
```




```
ypred = clf.predict(X_test)
accuracy_score(y_test, ypred)
```

```
0.8793103448275862
```