Name: Santiago, John Loyd C.

Course and Section: CPE 019 - CPE32S3

Date of Submission: April 3, 2024

Instructor: Engr. Roman Richard

LINK: https://colab.research.google.com/drive/1or1dHtML2IGYKl39NozyDEZ5LSIKdGdF?usp=sharing

PERFORM

- Task 1: Exploratory Data Analysis (Cleaning + Prepping the dataset)
- Task 2: Data modelling using ANN

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from keras.models  import Sequential
from keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
from keras.optimizers import Adam, SGD, RMSprop
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, precision_recall_curve, roc_auc_score, roc_cu
from sklearn.ensemble import RandomForestClassifier
```

```
pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages (0.
from ucimlrepo import fetch_ucirepo

# fetch dataset
phiusiil_phishing_url_website = fetch_ucirepo(id=967)

# data (as pandas dataframes)
X = phiusiil_phishing_url_website.data.features
y = phiusiil_phishing_url_website.data.targets

# metadata
print(phiusiil_phishing_url_website.metadata)

# variable information
print(phiusiil_phishing_url_website.variables)
```

```
38  None            no
39  None            no
40  None            no
41  None            no
42  None            no
43  None            no
44  None            no
45  None            no
46  None            no
47  None            no
48  None            no
49  None            no
50  None            no
51  None            no
52  None            no
53  None            no
54  None            no
55  None            no
```

```
X.head(235795)
```

	URL	URLLength	Domain	DomainLeng
0	https://www.southbankmosaics.com	31.0	www.southbankmosaics.com	2
1	https://www.uni-mainz.de	23.0	www.uni-mainz.de	1
2	https://www.voicefmradio.co.uk	29.0	www.voicefmradio.co.uk	2
3	https://www.sfnmjournal.com	26.0	www.sfnmjournal.com	1
4	https://www.rewildingargentina.org	33.0	www.rewildingargentina.org	2
...	...	...	...	
199156	https://www.logocross.com	24.0	www.logocross.com	1
199157	https://www.lipsum.com	21.0	www.lipsum.com	1
199158	https://www.thriveport.com	25.0	www.thriveport.com	1
199159	https://www.tepapa.govt.nz	25.0	www.tepapa.govt.nz	1
199160	NaN	NaN	NaN	N

199161 rows × 54 columns

```
y.head(235795)
```

	label
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
...	...
199156	1.0
199157	1.0
199158	1.0
199159	1.0
199160	NaN

199161 rows × 1 columns

```
X.dtypes
```

```
URL                           object
URLLength                    float64
Domain                        object
DomainLength                 float64
IsDomainIP                   float64
TLD                           object
URLSimilarityIndex           float64
CharContinuationRate         float64
TLDLegitimateProb            float64
URLCharProb                  float64
TLDLength                    float64
NoOfSubDomain                float64
HasObfuscation               float64
NoOfObfuscatedChar           float64
ObfuscationRatio             float64
NoOfLettersInURL             float64
LetterRatioInURL             float64
NoOfDegitsInURL              float64
DegitRatioInURL              float64
NoOfEqualsInURL              float64
NoOfQMarkInURL               float64
NoOfAmpersandInURL           float64
NoOfOtherSpecialCharsInURL   float64
SpacialCharRatioInURL        float64
IsHTTPS                      float64
LineOfCode                   float64
LargestLineLength            float64
HasTitle                     float64
```

```
Title                          object
DomainTitleMatchScore         float64
URLTitleMatchScore            float64
HasFavicon                    float64
Robots                        float64
IsResponsive                  float64
NoOfURLRedirect               float64
NoOfSelfRedirect              float64
HasDescription                float64
NoOfPopup                     float64
NoOfiFrame                    float64
HasExternalFormSubmit         float64
HasSocialNet                  float64
HasSubmitButton               float64
HasHiddenFields               float64
HasPasswordField              float64
Bank                          float64
Pay                           float64
Crypto                        float64
HasCopyrightInfo              float64
NoOfImage                     float64
NoOfCSS                       float64
NoOfJS                        float64
NoOfSelfRef                   float64
NoOfEmptyRef                  float64
NoOfExternalRef               float64
dtype: object
```

```
J = X.copy()
```

```
columns_to_delete = ['URL','Domain', 'TLD', 'Title' ]
existing_columns = [col for col in columns_to_delete if col in X.columns]

if existing_columns:
    J.drop(columns=existing_columns, inplace=True, axis=1)
```

```
J.head(235795)
```

	URLLength	DomainLength	IsDomainIP	URLSimilarityIndex	CharContinuationRate
0	31.0	24.0	0.0	100.0	1.000000
1	23.0	16.0	0.0	100.0	0.666667
2	29.0	22.0	0.0	100.0	0.866667
3	26.0	19.0	0.0	100.0	1.000000
4	33.0	26.0	0.0	100.0	1.000000
...	...	...	...	...	...
199156	24.0	17.0	0.0	100.0	1.000000
199157	21.0	14.0	0.0	100.0	1.000000
199158	25.0	18.0	0.0	100.0	1.000000
199159	25.0	18.0	0.0	100.0	0.636364
199160	NaN	NaN	NaN	NaN	NaN

199161 rows × 50 columns

```
J.corr()
```

NoOfOtherSpecialCharsInURL	0.779924	0.263729	0.276359	-0.524191
SpacialCharRatioInURL	0.193159	0.182372	0.116209	-0.603776
IsHTTPS	0.013106	-0.020287	-0.012815	0.349383
LineOfCode	-0.058631	-0.075290	-0.016350	0.232190
LargestLineLength	0.044705	0.066946	0.001270	-0.080328
HasTitle	-0.072196	-0.107225	-0.003601	0.352920
DomainTitleMatchScore	-0.208399	-0.295763	-0.052176	0.602361
URLTitleMatchScore	-0.186066	-0.327518	-0.054562	0.542251
HasFavicon	-0.091325	-0.148825	-0.030895	0.403341
Robots	-0.071299	-0.081853	-0.028343	0.313730
IsResponsive	-0.080284	-0.120499	-0.006774	0.435351
NoOfURLRedirect	0.029019	0.020231	0.022050	-0.061979
NoOfSelfRedirect	-0.005140	-0.046955	-0.010228	-0.051655
HasDescription	-0.143505	-0.201117	-0.022250	0.589530
NoOfPopup	-0.010322	-0.012462	-0.002666	0.039628
NoOfiFrame	-0.041859	-0.049273	-0.009593	0.185412
HasExternalFormSubmit	-0.034032	-0.038250	-0.007846	0.139315
HasSocialNet	-0.174526	-0.215104	-0.046428	0.673360
HasSubmitButton	-0.065267	-0.111196	-0.012760	0.449141
HasHiddenFields	-0.070277	-0.111878	-0.013025	0.406833
HasPasswordField	0.021533	0.008827	-0.008591	0.062244
Bank	-0.030758	-0.041239	-0.016915	0.154575
Pay	-0.055422	-0.066682	-0.014710	0.292792
Crypto	-0.024504	-0.034977	-0.006179	0.087517
HasCopyrightInfo	-0.121959	-0.195758	-0.023804	0.613655
NoOfImage	-0.061817	-0.082908	-0.016371	0.233435
NoOfCSS	-0.012848	-0.016034	-0.003812	0.053291
NoOfJS	-0.072201	-0.097177	-0.022110	0.301477
NoOfSelfRef	-0.078368	-0.104941	-0.020684	0.295120
NoOfEmptyRef	-0.022238	-0.031409	-0.006664	0.090480
NoOfExternalRef	-0.064490	-0.079109	-0.017326	0.251421

50 rows × 50 columns

```
X_train, X_test, y_train, y_test = train_test_split(J, y, test_size=0.25, random_state=1111
```

```
normalizer = StandardScaler()
X_train_norm = normalizer.fit_transform(X_train)
X_test_norm = normalizer.transform(X_test)
```

```
np.mean(y), np.mean(1-y)
```

```
/usr/local/lib/python3.10/dist-packages/numpy/core/fromnumeric.py:3502: FutureWarning:
  return mean(axis=axis, dtype=dtype, out=out, **kwargs)
/usr/local/lib/python3.10/dist-packages/numpy/core/fromnumeric.py:3502: FutureWarning:
  return mean(axis=axis, dtype=dtype, out=out, **kwargs)
```

```
(label    0.572334
 dtype: float64,
 label    0.427666
 dtype: float64)
```

```
J.head(235795)
```

	URLLength	DomainLength	IsDomainIP	URLSimilarityIndex	CharContinuationRate
0	31.0	24.0	0.0	100.0	1.000000
1	23.0	16.0	0.0	100.0	0.666667
2	29.0	22.0	0.0	100.0	0.866667
3	26.0	19.0	0.0	100.0	1.000000
4	33.0	26.0	0.0	100.0	1.000000
...	...	...	...	...	...
199156	24.0	17.0	0.0	100.0	1.000000
199157	21.0	14.0	0.0	100.0	1.000000
199158	25.0	18.0	0.0	100.0	1.000000
199159	25.0	18.0	0.0	100.0	0.636364
199160	NaN	NaN	NaN	NaN	NaN

199161 rows × 50 columns

```
model  = Sequential([
    Dense(100, input_shape=(50,), activation="relu"),
    Dense(42, activation="relu"),
    Dropout(0,7),
    Dense(34, activation="relu"),
    Dense(1, activation="sigmoid")
])
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 100)               5100

 dense_1 (Dense)             (None, 42)                4242

 dropout (Dropout)           (None, 42)                0
 dense_2 (Dense)             (None, 34)                1462

 dense_3 (Dense)             (None, 1)                 35

=================================================================
Total params: 10839 (42.34 KB)
Trainable params: 10839 (42.34 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
model.compile(SGD(learning_rate = 0.001),"binary_crossentropy", metrics = ['accuracy'])
model_fit = model.fit(X_train_norm, y_train, validation_data = (X_test_norm, y_test),epochs
```

```
Epoch 1/5
4668/4668 [==============================] - 38s 8ms/step - loss: nan - accuracy: 0.722
Epoch 2/5
4668/4668 [==============================] - 32s 7ms/step - loss: nan - accuracy: 0.427
Epoch 3/5
4668/4668 [==============================] - 16s 3ms/step - loss: nan - accuracy: 0.427
Epoch 4/5
4668/4668 [==============================] - 16s 3ms/step - loss: nan - accuracy: 0.427
Epoch 5/5
4668/4668 [==============================] - 14s 3ms/step - loss: nan - accuracy: 0.427
```

```
y_pred_class_nn_1 = (model.predict(X_test_norm)> 0.5).astype('int32')
y_pred_prob_nn_1 = model.predict(X_test_norm)
```

```
1556/1556 [==============================] - 2s 2ms/step
1393/1556 [=========================>....] - ETA: 0s
```

```
accuracies = accuracy_score(y_test,y_pred_class)
accuracies
```

```
fig, ax = plt.subplots()
ax.plot(model_fit.history["accuracy"],'r', label="Accuracy")
ax.plot(model_fit.history["val_accuracy"],'b',label="Validation Accuracy")
ax.plot(model_fit.history["loss"],'g', label="Train Loss")
ax.plot(model_fit.history["val_loss"],'y', label="Validation Loss")
ax.legend()
```

```
def plot_roc(y_test, y_pred, model_name):
    fpr, tpr, thr = roc_curve(y_test, y_pred)
    fig, ax = plt.subplots(figsize=(8, 8))
    ax.plot(fpr, tpr, 'k-')
    ax.plot([0, 1], [0, 1], 'k--', linewidth=.5)  # roc curve for random model
    ax.grid(True)
    ax.set(title='ROC Curve for {} on PIMA diabetes problem'.format(model_name),
           xlim=[-0.01, 1.01], ylim=[-0.01, 1.01])
    print('accuracy is {:.3f}'.format(accuracy_score(y_test,y_pred_class)))
    print('roc-auc is {:.3f}'.format(roc_auc_score(y_test,y_pred_prob)))
plot_roc(y_test, y_pred_prob, 'NN')
```