

Name: Gullas, Rainer L.

Section: BSCPE32S3

Date Performed: 03/24/2024

Date Submitted: 03/26/2024

Professor: Engr. Roman Richard

```
1 from google.colab import drive
2 drive.mount('/content/MyDrive')
```

Mounted at /content/MyDrive

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import LabelEncoder, StandardScaler
5 from sklearn.neural_network import MLPClassifier
6 from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
1 data = pd.read_csv("/content/MyDrive/MyDrive/name_gender_dataset.csv")
```

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

|   | Name    | Gender | Count   | Probability |
|---|---------|--------|---------|-------------|
| 0 | James   | M      | 5304407 | 0.014517    |
| 1 | John    | M      | 5260831 | 0.014398    |
| 2 | Robert  | M      | 4970386 | 0.013603    |
| 3 | Michael | M      | 4579950 | 0.012534    |
| 4 | William | M      | 4226608 | 0.011567    |
| 5 | Mary    | F      | 4169663 | 0.011411    |
| 6 | David   | M      | 3787547 | 0.010366    |
| 7 | Joseph  | M      | 2695970 | 0.007378    |
| 8 | Richard | M      | 2638187 | 0.007220    |
| 9 | Charles | M      | 2433540 | 0.006660    |

```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 147269 entries, 0 to 147268
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    Name      147269 non-null  object
1    Gender     147269 non-null  object
2    Count      147269 non-null  int64
3    Probability 147269 non-null  float64
dtypes: float64(1), int64(1), object(2)
memory usage: 4.5+ MB
```

```

1
2 X = data[['Name']]
3 y = data['Gender']
4
5 #Converting the Name column to a numerical format
6 le = LabelEncoder()
7 X = X.apply(le.fit_transform)
8
9 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
10
11
12 scaler = StandardScaler()
13 X_train = scaler.fit_transform(X_train)
14 X_test = scaler.transform(X_test)
15
16
17 mlp = MLPClassifier(hidden_layer_sizes=(100,), random_state=42)
18
19
20 mlp.fit(X_train, y_train)
21
22 y_pred = mlp.predict(X_test)
23
24
25 print("Accuracy: {:.2f}".format(accuracy_score(y_test, y_pred)))
26 print(classification_report(y_test, y_pred))
27 print(confusion_matrix(y_test, y_pred))

Accuracy: 0.61
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
_warn_prf(average, modifier, msg_start, len(result))
      precision    recall  f1-score   support

      F         0.61      1.00      0.76      26962
      M         0.00      0.00      0.00      17219

 accuracy
macro avg      0.31      0.50      0.38      44181
weighted avg      0.37      0.61      0.46      44181

[[26962    0]
 [17219    0]]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
_warn_prf(average, modifier, msg_start, len(result))

```

```

1 #Testing the MLP
2 user_input_name = 'Alice'
3
4 user_input_name = user_input_name.strip().title()
5 user_input_name_encoded = le.transform([user_input_name])
6 user_input_name_encoded = user_input_name_encoded.reshape(1, -1)
7 user_input_name_encoded = scaler.transform(user_input_name_encoded)
8 user_input_name_gender = mlp.predict(user_input_name_encoded)
9
10 if user_input_name_gender[0] == 0:
11     print(f'Based on the MLP model, the name "{user_input_name}" is predicted to be female.')
12 else:
13     print(f'Based on the MLP model, the name "{user_input_name}" is predicted to be male.')

Based on the MLP model, the name "Alice" is predicted to be male.
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler was fi
warnings.warn(

```

bases on the accuracy given by the training which is at 61% I expected this kind of output that even if the given name is considered as Female the MLP recognized it as Male

In conclusion in this activity the use of Multilayer Perceptron(MLP) can be used in many things such as classification, regression, and etc.. my MLP only outputted 61% accuracy which is not desirable and as expected of the output when I inputted a random person name I know for my self that the name Alice should be classified as Female but the algo recognized it as Male due to the lack of samples in the dataset and the algorithm itself

