ASSIGNMENT 1

Christos Eleftheriou

1009537

Part A – Data prep & visualization | Classification

Using python code, answer the following questions

1. How many men from the United States are represented in this dataset?

```
Males in adults.csv:  19488
```

2. Is it true that adults with at least a Bachelors degree are guaranteed to receive more than 50K per year?

```
Adults with at least a Bachelors degree are not guaranteed to receive more than 50K per year
```

3. What is the minimum, maximum, average and standard deviation of the hours-per-week for each race-gender pair?

```
Minimum, maximum, average and standard deviation of the hours-per week for pair  White Male  :
1
99
42.6688223636174
12.194632884770783
Minimum, maximum, average and standard deviation of the hours-per week for pair  White Female  :
1
99
36.296690580884054
12.190951154577998
Minimum, maximum, average and standard deviation of the hours-per week for pair  Black Male  :
1
99
39.9974506054812
10.909413348979392
Minimum, maximum, average and standard deviation of the hours-per week for pair  Black Female  :
2
99
36.834083601286174
9.419959967928806
Minimum, maximum, average and standard deviation of the hours-per week for pair  Asian-Pac-Islander Male  :
1
99
41.46897546897547
12.387562694006963
```

```
Minimum, maximum, average and standard deviation of the hours-per week for pair  Asian-Pac-Islander Female  :
1
99
37.4393063583815
12.479458534510538
Minimum, maximum, average and standard deviation of the hours-per week for pair  Amer-Indian-Eskimo Male  :
3
84
42.197916666666664
11.596280132632396
Minimum, maximum, average and standard deviation of the hours-per week for pair  Amer-Indian-Eskimo Female  :
4
84
36.57983193277311
11.046508611950053
Minimum, maximum, average and standard deviation of the hours-per week for pair  Other Male  :
5
98
41.851851851851855
11.084779011865367
Minimum, maximum, average and standard deviation of the hours-per week for pair  Other Female  :
6
65
35.92660550458716
10.300760869072795
```

Using python code : For both adults.csv and adults_test.csv

1. Remove any missing values or duplicates from the dataset.

  a. How many missing values are there for each feature?

  b. What is the size of the dataset before/after cleaning the data?

```
adults.csv
----------------------------------- BEFORE -----------------------------------
Number of adults before:  32561
Unnamed: 0           0
Age                  0
Work Class        1836
Education            0
Marital Status       0
Occupation        1843
Relationship         0
Race                 0
Sex                  0
Hours Per Week       0
Native Country     583
Salary               0
dtype: int64
----------------------------------- AFTER -----------------------------------
Number of adults after:  30162
Unnamed: 0        0
Age               0
Work Class        0
Education         0
Marital Status    0
Occupation        0
Relationship      0
Race              0
Sex               0
Hours Per Week    0
Native Country    0
Salary            0
dtype: int64
```

```
adults_test.csv
----------------------------------- BEFORE -----------------------------------
Number of adults before:  16281
Unnamed: 0          0
Age                 0
Work Class        963
Education           0
Marital Status      0
Occupation        966
Relationship        0
Race                0
Sex                 0
Hours Per Week      0
Native Country    274
Salary              0
dtype: int64
----------------------------------- AFTER -----------------------------------
Number of adults after:  15060
Unnamed: 0        0
Age               0
Work Class        0
Education         0
Marital Status    0
Occupation        0
Relationship      0
Race              0
Sex               0
Hours Per Week    0
Native Country    0
Salary            0
dtype: int64
```

2. Which raw feature in the dataset is suitable for categorical/ordinal encoding? Add a new column to the dataset with the ordinal encoding of that feature.

a. Use “print(data[["Feature", "feature_encoded"]].head(20))” to show the results

The raw feature that is suitable for ordinal encoding is education.

adults.csv

```
       Education  education_encoded
0      Bachelors               12.0
1      Bachelors               12.0
2        HS-grad                8.0
3           11th                6.0
4      Bachelors               12.0
5        Masters               13.0
6            9th                4.0
7        HS-grad                8.0
8        Masters               13.0
9      Bachelors               12.0
10  Some-college                9.0
11     Bachelors               12.0
12     Bachelors               12.0
13    Assoc-acdm               11.0
15       7th-8th                3.0
16       HS-grad                8.0
17       HS-grad                8.0
18          11th                6.0
19       Masters               13.0
20     Doctorate               15.0
```

adults_test.csv

```
       Education  education_encoded
0           11th                6.0
1        HS-grad                8.0
2     Assoc-acdm               11.0
3   Some-college                9.0
5           10th                5.0
7    Prof-school               14.0
8   Some-college                9.0
9        7th-8th                3.0
10       HS-grad                8.0
11     Bachelors               12.0
12       HS-grad                8.0
14       HS-grad                8.0
15       Masters               13.0
16  Some-college                9.0
17       HS-grad                8.0
18       HS-grad                8.0
20     Bachelors               12.0
21  Some-college                9.0
23     Bachelors               12.0
24     Bachelors               12.0
```

3. Scale the "Age" feature with the appropriate scaling method and add the result as a new column to the dataset named "age_scaled".

a. Use "print(data[["Age", "age_scaled"]].head(20))" to show the results

adults.csv

	Age	age_scaled
0	39	0.042796
1	50	0.880288
2	38	-0.033340
3	53	1.108695
4	28	-0.794697
5	37	-0.109476
6	49	0.804152
7	52	1.032559
8	31	-0.566290
9	42	0.271203
10	37	-0.109476
11	30	-0.642425
12	23	-1.175375
13	32	-0.490154
15	34	-0.337883
16	25	-1.023104
17	32	-0.490154
18	38	-0.033340
19	43	0.347338
20	40	0.118931

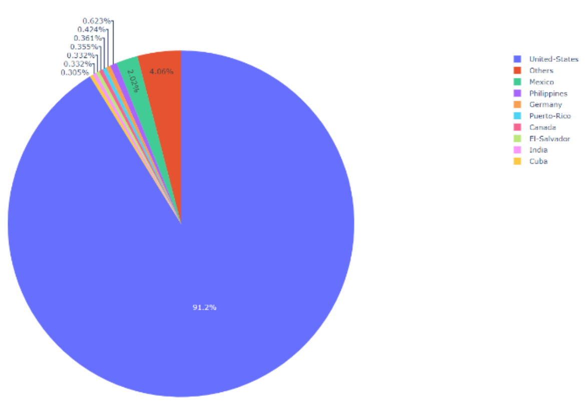
adults_test.csv

	Age	age_scaled
0	25	-1.029005
1	38	-0.057423
2	28	-0.804794
3	44	0.391000
5	34	-0.356371
7	63	1.811006
8	24	-1.103742
9	55	1.213109
10	65	1.960480
11	36	-0.206897
12	26	-0.954268
14	48	0.689949
15	43	0.316263
16	20	-1.402691
17	43	0.316263
18	37	-0.132160
20	34	-0.356371
21	34	-0.356371
23	25	-1.029005
24	25	-1.029005

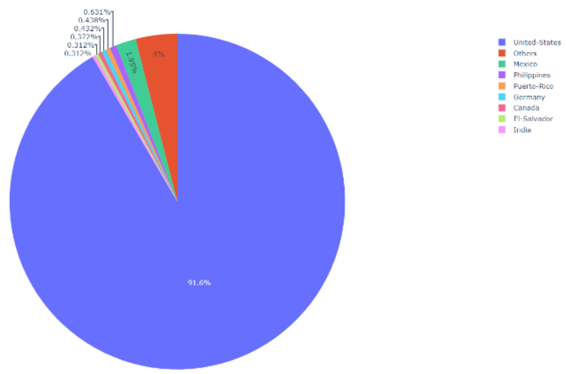
4. Visualization

a. Using a Plotly Pie chart visualize the distribution of adults in the dataset based on their native country.

adults.csv

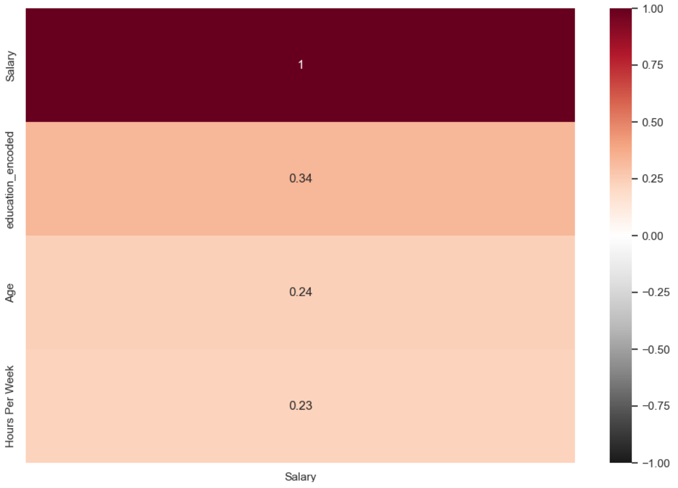


adults_test.csv

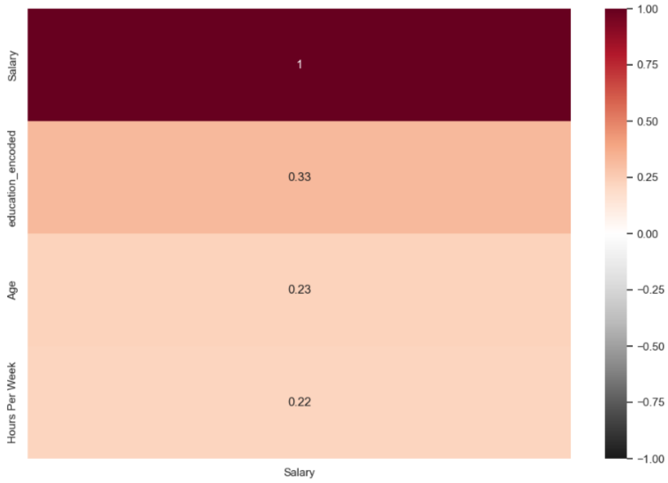


b. Using Seaborn Heat Map visualize the correlation of Age, feature_encoded and Hours Per Week with the Salary. Based on correlation alone, which single feature would you select as the most important for classifying Salary?

adults.csv



adults_test.csv



Using the heatmaps and based on correlation alone, the feature I would select as the most important for classifying salary is education_encoded.

Classification

1. Using Logistic Regression predict the salary classes of the adults in the test set.

2. Create the confusion matrix. Explain what TP, TN, FP, FN mean in this case (for this salary data)

TP means that the salary was predicted >50K and was >50K

TN means that the salary was predicted <=50K and was <=50K

FP means that the salary was predicted >50K but was <=50K

FN means that the salary was predicted <=50K but was >50K

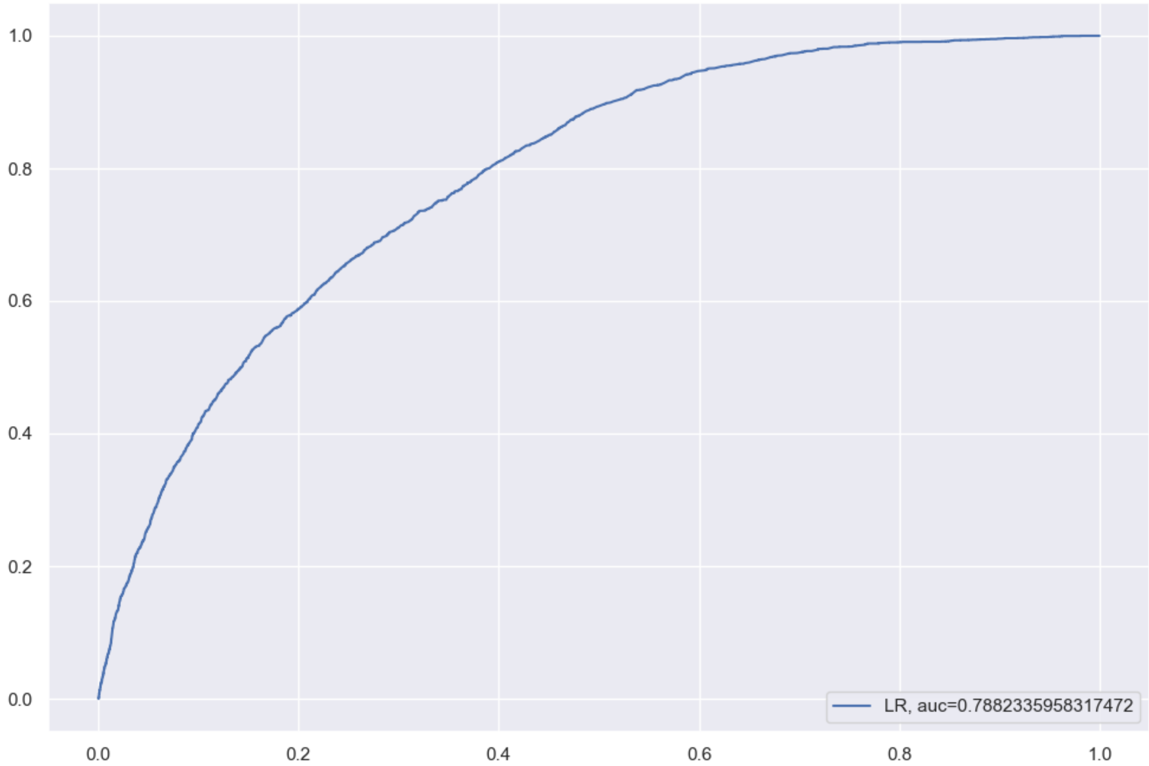
```
LR Confusion Matrix
TN: 10565 FP: 795 FN: 2463 TP: 1237
[[10565   795]
 [ 2463  1237]]
```

3. Calculate the accuracy, precision, recall and f1 score of the classifier when evaluated on the test set. Do the metrics show any issue with the model?

```
Accuracy:  0.7836653386454183
Precision:  0.6087598425196851
Recall:  0.33432432432432435
F1:  0.43161200279134687
```

The metrics show that the model is not very accurate.

4. Plot the ROC curve and calculate the area-under-the-curve (AUC) of the classifier when evaluated on the test set.



Part B – Feature Engineering | Regression

1. Feature engineering: Create a Bivariate, Polynomial and Custom feature of your choice, and add them as columns to the dataset

```
Bivariate Feature: RAD x tmdb_DIS
     RAD     DIS       RD
0      1  4.0900   4.0900
1      2  4.9671   9.9342
2      2  4.9671   9.9342
3      3  6.0622  18.1866
4      3  6.0622  18.1866
..   ...     ...      ...
501    1  2.4786   2.4786
502    1  2.2875   2.2875
503    1  2.1675   2.1675
504    1  2.3889   2.3889
505    1  2.5050   2.5050
```

```
Polynomial feature RAD^2
     RAD  strong_RAD
0      1           1
1      2           4
2      2           4
3      3           9
4      3           9
..   ...         ...
501    1           1
502    1           1
503    1           1
504    1           1
505    1           1
```

```
Custom feature: DIS to RAD ratio-> DIS/RAD
        DIS  RAD  DIS_to_RAD_ratio
0    4.0900    1          4.090000
1    4.9671    2          2.483550
2    4.9671    2          2.483550
3    6.0622    3          2.020733
4    6.0622    3          2.020733
..      ...  ...               ...
501  2.4786    1          2.478600
502  2.2875    1          2.287500
503  2.1675    1          2.167500
504  2.3889    1          2.388900
505  2.5050    1          2.505000
```

2. Split the dataset into train/test sets (90 train/10 test split, random_state=0)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.10, random_state=0)
```

3. Using Linear Regression predict the house prices of the test set

```
lr = LinearRegression()
lr.fit(X_train,y_train)
lr_pred = lr.predict(X_test)
```

4. Calculate the Mean Squared Error of the regression model.

```
MSE of LR: 38.35449995956134
```

5. Plot a line chart of the real VS predicted house prices of the test set