CSE343/CSE543/ECE363/ECE563: Machine Learning (PG)
Monsoon 2022
Assignment-2

Sehban Fazili
MTech CSE
MT21143

Q1. Data's Objective Identification:

(a) DataSet used:
https://data.gov.in/resource/percentage-schools-drinking-water-facility-2013-14-2015-16

- The percentage of Schools with drinking water facilities in Delhi is given here.
- The data included is from the year 2014 to 2016.
- Around 95% of all schools in Delhi have drinking water facilities.
- This data can be used to help improve the drinking water facilities in Delhi schools over the next few years.

(b) Row 10th before Normalization:

```
Primary_Only                        78.32
Primary_with_U_Primary              91.18
Primary_with_U_Primary_Sec_HrSec     95.9
U_Primary_Only                      79.55
U_Primary_With_Sec_HrSec            97.01
Primary_with_U_Primary_Sec          93.25
U_Primary_With_Sec                  93.46
Sec_Only                             88.3
Sec_with_HrSec.                      88.6
HrSec_Only                          96.22
All Schools                          80.3
Name: 9, dtype: object
```

Row 10th after Normalization:

```
array([0.44353183, 0.78275862, 0.959     , 0.7955    , 0.9701    ,
       0.9325    , 0.9346    , 0.883     , 0.886     , 0.9622    ,
       0.48483264])
```

(c) Row 10th before Standardization:

```
Primary_Only                        78.32
Primary_with_U_Primary              91.18
Primary_with_U_Primary_Sec_HrSec     95.9
U_Primary_Only                      79.55
U_Primary_With_Sec_HrSec            97.01
Primary_with_U_Primary_Sec          93.25
U_Primary_With_Sec                  93.46
Sec_Only                             88.3
Sec_with_HrSec.                      88.6
HrSec_Only                          96.22
All Schools                          80.3
Name: 9, dtype: object
```

Row 10th before Standardization:

```
array([-1.70155198, -0.75611258, -0.09560039, -0.38123488,  0.10528095,
        0.0413757 ,  0.34573378,  0.50351322,  0.22834341,  0.45276508,
       -1.8703309 ])
```

Q2. Data Augmentation:

Used the **PIL** library to implement this question.

The steps involved are as follows:

(i) To pick three different random images from the dataset for each augmentation task, I used the function:

```
def random_images(x):
    random_imgs = random.sample(path,x)
    return random_imgs
```

(ii) Apart from the pad operation rest were done using the **PIL** library, and pad was done using transformers from **torchvision** library.

Q3. Logistic Regression:

(b) Steps involved are as follows:

(i) Loaded the data into train and test sets.

(ii) Scaled the data by dividing it by 255.

(iii) Performed OVO and OVR over the dataset.

(iv) Result:

OVR:

Classification Report

	precision	recall	f1-score	support
0	0.95	0.98	0.96	980
1	0.96	0.98	0.97	1135
2	0.94	0.89	0.91	1032
3	0.89	0.91	0.90	1010
4	0.92	0.93	0.93	982
5	0.89	0.86	0.88	892
6	0.94	0.95	0.94	958
7	0.93	0.93	0.93	1028
8	0.87	0.87	0.87	974
9	0.90	0.89	0.90	1009
accuracy			0.92	10000
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000

Class-wise Accuracy:

```
Class 0 accuracy: 97.86 %
Class 1 accuracy: 97.97 %
Class 2 accuracy: 89.15 %
Class 3 accuracy: 90.89 %
Class 4 accuracy: 93.18 %
Class 5 accuracy: 86.21 %
Class 6 accuracy: 94.68 %
Class 7 accuracy: 92.51 %
Class 8 accuracy: 87.27 %
Class 9 accuracy: 88.90 %
```

OVO:

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	980
1	0.98	0.99	0.98	1135
2	0.94	0.94	0.94	1032
3	0.92	0.93	0.93	1010
4	0.95	0.96	0.95	982
5	0.92	0.90	0.91	892
6	0.95	0.96	0.96	958
7	0.95	0.94	0.95	1028
8	0.92	0.91	0.91	974
9	0.93	0.92	0.93	1009
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000

Class-wise Accuracy:

```
Class 0 accuracy: 97.96 %
Class 1 accuracy: 98.94 %
Class 2 accuracy: 93.51 %
Class 3 accuracy: 93.37 %
Class 4 accuracy: 96.13 %
Class 5 accuracy: 90.25 %
Class 6 accuracy: 96.14 %
Class 7 accuracy: 94.26 %
Class 8 accuracy: 90.86 %
Class 9 accuracy: 92.17 %
```

Q5. Unsupervised Learning:

(i) Hierarchical Clustering:

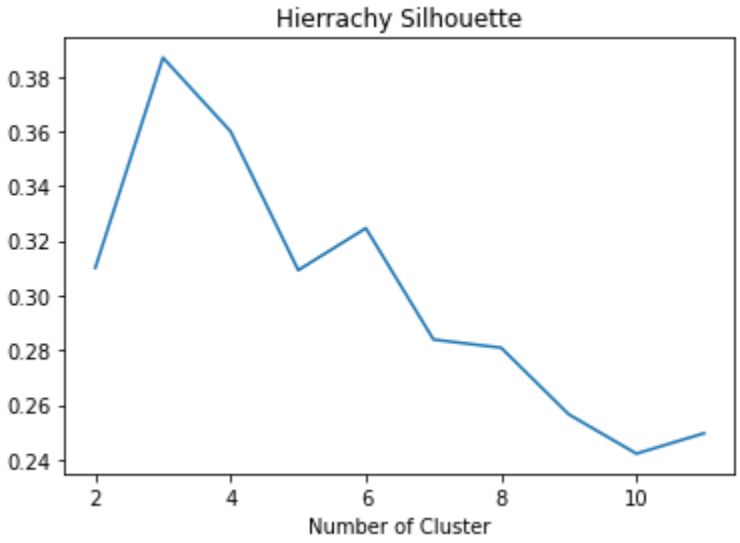
Results:

Train Data:

Silhouette Scores:

```
[0.31014285 0.38692481 0.35996439 0.30925818 0.32459993 0.28389621
 0.28090785 0.25655897 0.24220513 0.24965192]
```

Silhouette Score Plot:

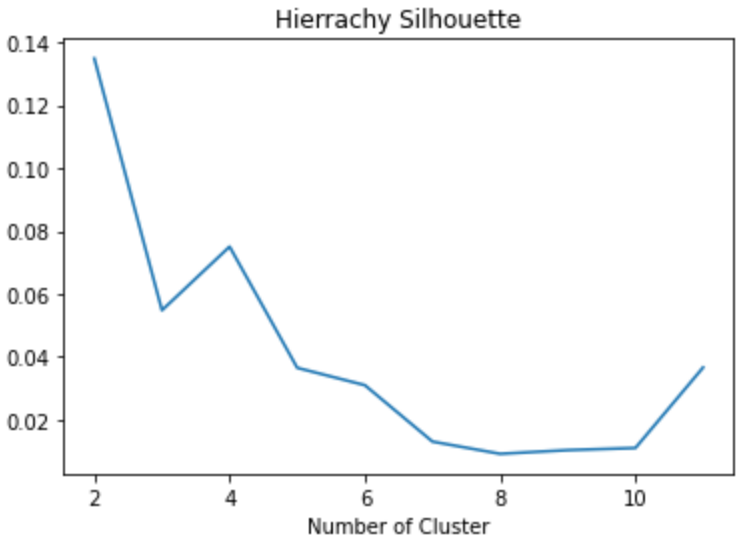


Validation Data:

Silhouette Scores:

```
[0.13489496 0.05480511 0.07505127 0.03650255 0.03098671 0.01304108
 0.00915171 0.01032926 0.01100885 0.03660229]
```

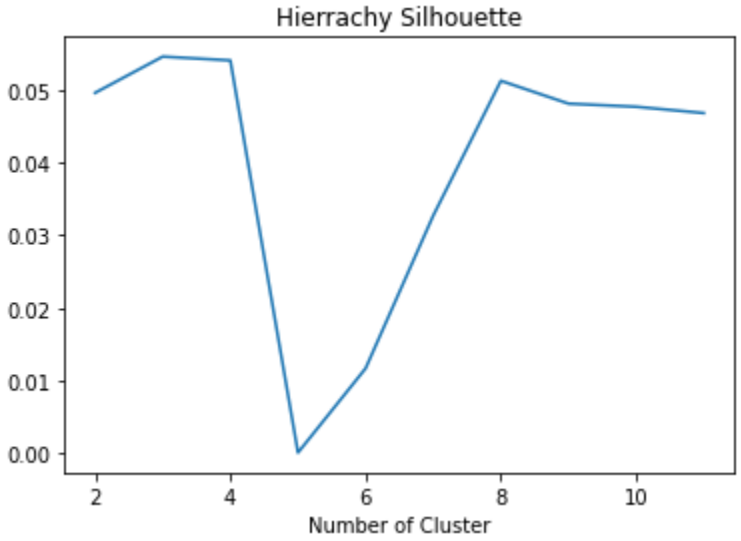
Silhouette Score Plot:



Test Data:
Silhouette Scores:

```
[0.04958959 0.05456964 0.05404873 0.00015569 0.01173942 0.03273271
 0.05124872 0.04810848 0.04767871 0.04681162]
```

Silhouette Scores Plot:

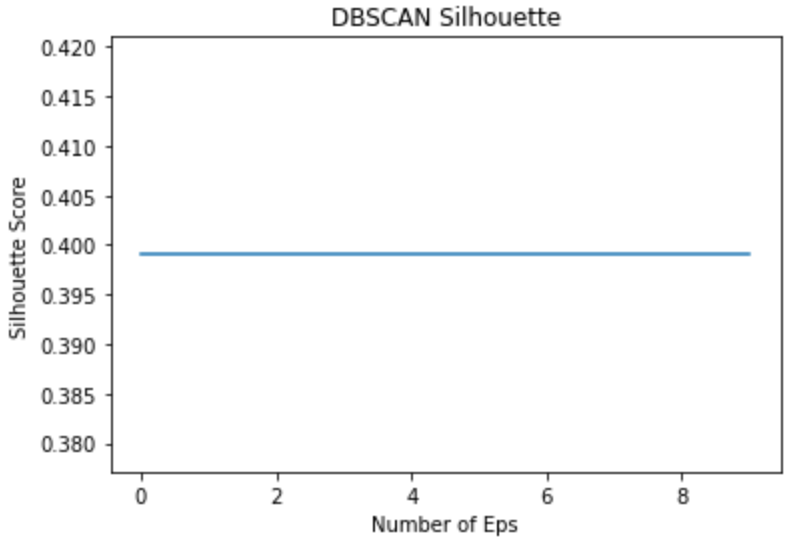


(ii) Density-based Clustering:

Silhouette Scores:

```
[0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534, 0.3990678143095534]
```

Silhouette Scores Plot:



Q6. Classification Metrics:

(a) Data preprocessing was done.

(b) Tried two models which drop columns. One model used **information gain** to select the most important features, and the other model used a **correlation matrix** to select important features.

(c) Used Logistic Regressor as the classifier.

(d) Results:

Best Model:

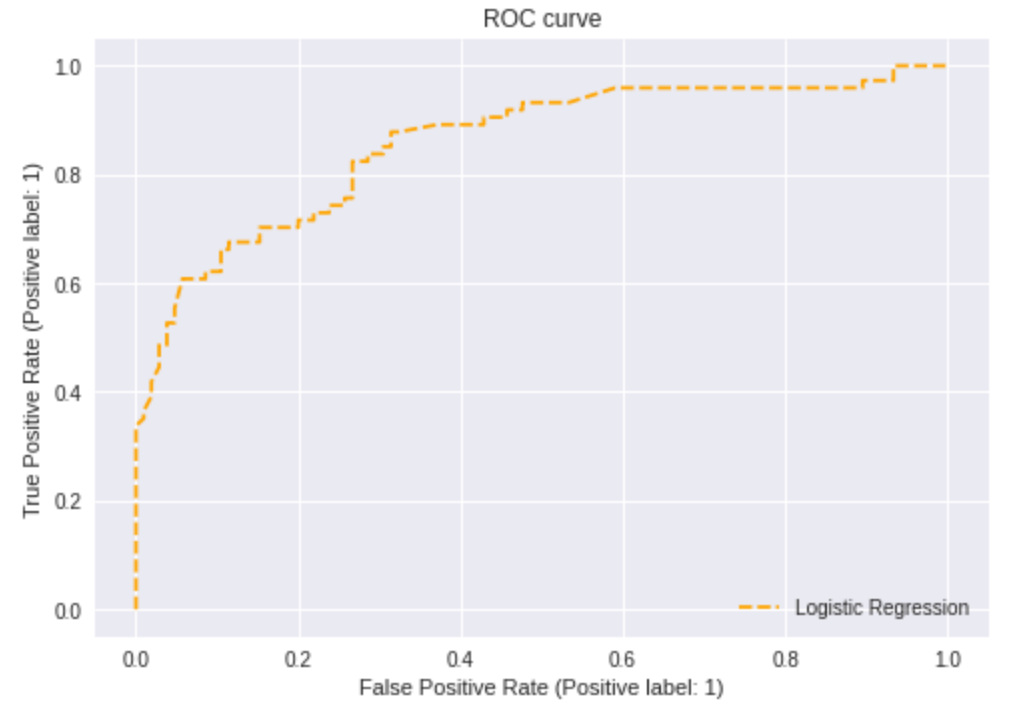
Classification Report:

	precision	recall	f1-score	support
0	0.80	0.85	0.82	105
1	0.76	0.70	0.73	74
accuracy			0.79	179
macro avg	0.78	0.78	0.78	179
weighted avg	0.79	0.79	0.79	179

ROC-AUC Score:

```
0.7751608751608752
```

ROC-AUC Curve:



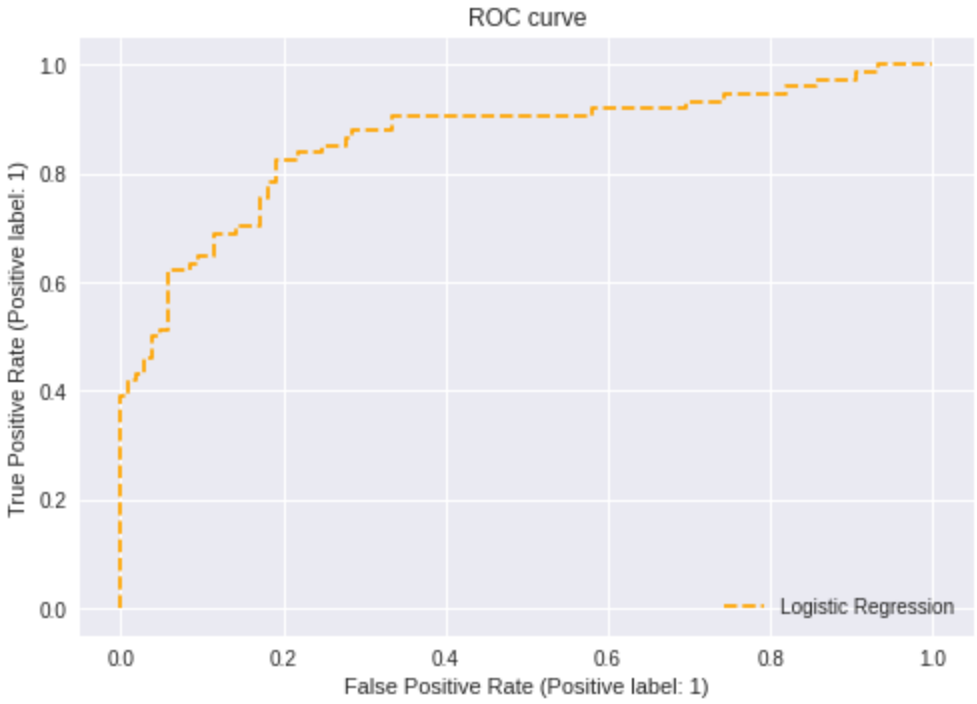
Other Model:
Classification Report:

	precision	recall	f1-score	support
0	0.80	0.84	0.82	105
1	0.75	0.70	0.73	74
accuracy			0.78	179
macro avg	0.78	0.77	0.77	179
weighted avg	0.78	0.78	0.78	179

ROC-AUC Score:

```
0.7703989703989704
```

ROC-AUC Curve:



Q7. Thinking beyond what is written:

(c) Calculated from part (a).

Results:

	Metrics	Score
0	Accuracy	0.500000
1	Precision	0.500000
2	Recall	1.000000
3	F0.5 Score	0.666667
4	F1 Score	0.666667
5	F5 Score	0.962963

Q8. Cross Validation:

All cross-validation techniques were implemented using the sklearn library. The result is as follows:

Cross-Validation Technique	CV-Score
Monte Carlo Cross Validation	0.971111
Leave P Out Cross-Validation	0.965414
Stratified 3-fold Cross Validation	0.973333
Hold Out Cross-Validation	0.933333