Homework 6 - CS 5805

Name : Jyothi Sevakula

Q1.

1)

a) logistic regression model :-

$$\text{log-odds} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

given, $\beta_0 = -25.9382$

$\beta_1 = 0.1109$ (Income)

$\beta_2 = 0.9368$ ~~0.9638~~ (for Lot-size)

$$\text{log-odds} = -25.9382 + 0.1109 X_1 + 0.9638 X_2$$

b)

$$P(\text{Owner}) = \frac{1}{1 + e^{-(\text{log-odds})}}$$

for customer 1, (Income $X_1 = 60$ and Lot-size $= 18.4$)

$$\text{log-odds} = -25.9382 + 0.1109(60) + 0.9638(18.4) = -1.55028$$

$$P(\text{Owner})_{\text{customer 1}} = \frac{1}{1+e^{1.55}} = 0.175 < 0.75$$

classification :- Customer 1 → Non-owner

for customer 2, (Income $X_1 = 64.8$ and lot-size $= 21.6$)

$$\text{log-odds} = -25.9382 + 0.1109(64.8) + 0.9638(21.6) = 2.0662$$

$$P(\text{Owner})_{\text{customer 2}} = \frac{1}{1+e^{-2.0662}} = 0.887 > 0.75$$

classification :- Customer 2 → Owner ✓

for customer 3, (Income $X_1 = 84$ and lot-size $= 17.6$)

$$\text{log-odds} = -25.9382 + 0.1109(84) + 0.9638(17.6) = 0.34028$$

$$P(\text{Owner})_{\text{customer 3}} = \frac{1}{1+e^{-0.34028}} = 0.5842 < 0.75$$

(Non-owner)

for customer 4, (Income $X_1 = 59$ and lot-size $= 16$)

$$\text{log-odds} = -25.9382 + 0.1109(59) + 0.9638(16) = -3.9743$$

$$P(\text{Owner})_{\text{customer 4}} = \frac{1}{1+e^{3.9743}} = 0.018 < 0.75$$

(Non-owner)

for customer 5, (Income $X_1 = 108$ and lot-size $X_2 = 17.6$)

$$\text{log-odds} = -25.9382 + 0.1109(108) + 0.9638(17.6) = 3.0018$$

$$P(\text{Owner})_{\text{customer 5}} = \frac{1}{1+e^{-3.0018}} = 0.953 > 0.75$$

(owner) ✓

for customer 6, (Income $X_1 = 75$ and lot-size $X_2 = 19.6$)

$$\text{log-odds} = -25.9382 + 0.1109(75) + 0.9638(19.6) = 1.2697$$

$$P(\text{Owner})_{\text{customer 6}} = \frac{1}{1+e^{-1.2697}} = 0.7806 > 0.75$$

(owner) ✓

∴ customer 2, 5, 6 are classified as owners and customer 1, 3, 4 are classified as Non-owners.

c)

Customer #	Ownership	predicted ownership
1	Owner	Non-owner
2	Owner	owner
3	Non-owner	Non-owner
4	Non-owner	Non-owner
5	Owner	owner
6	Non-owner	owner

True positives (TP) = 2 (customer #2 and # 5)

False Negative (FN) = 1 (customer # 1)

False positive (FP) = 1 (customer #6)

True Negative (TN) = 2 (customer # 3 and # 4)

Confusion Matrix :-

	predicted owner	predicted non-owner
Actual owner	2 (TP)	1 (FN)
Actual Non-Owner	1 (FP)	2 (TN)

d)

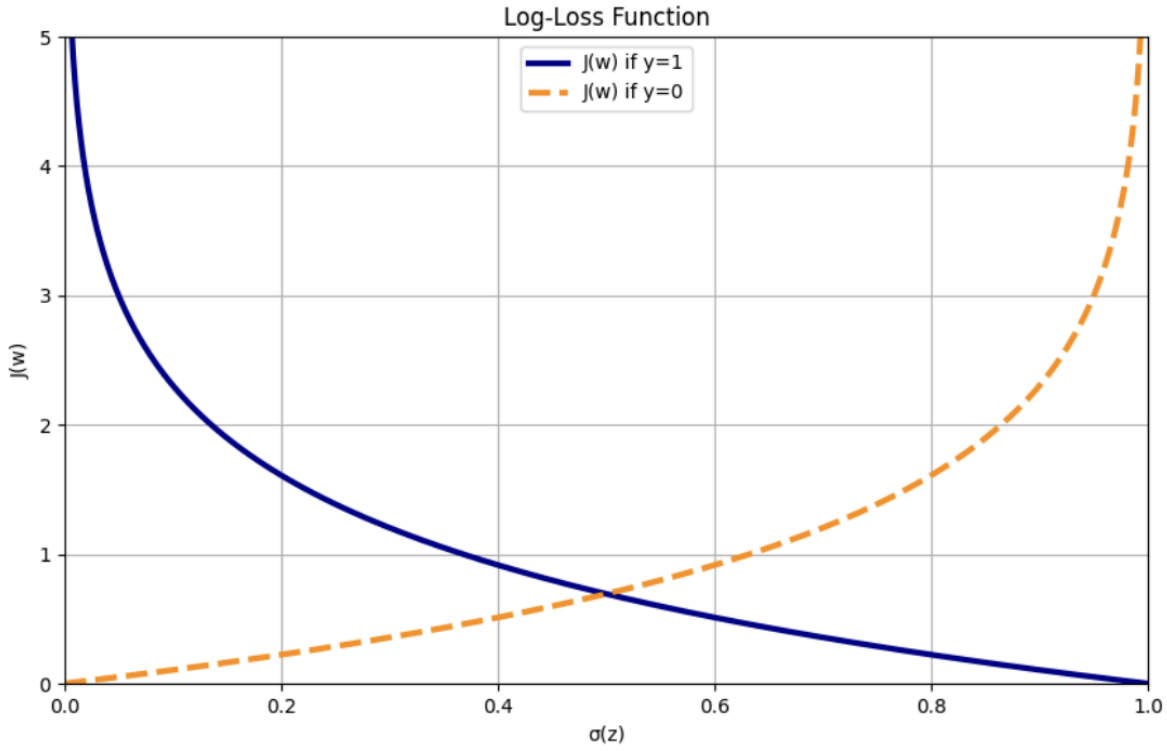
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{4}{6} = 2/3 = 0.667$$

$$\text{precision} = \frac{TP}{TP + FP} = \frac{2}{2+1} = \frac{2}{3} = 0.667$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{2}{2+1} = 2/3 = 0.667.$$

Accuracy, precision, recall are 66.7%

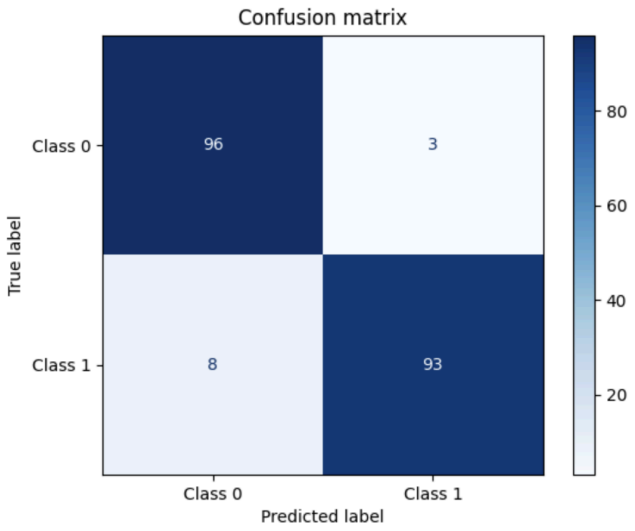
Q2.



Q3.

i.

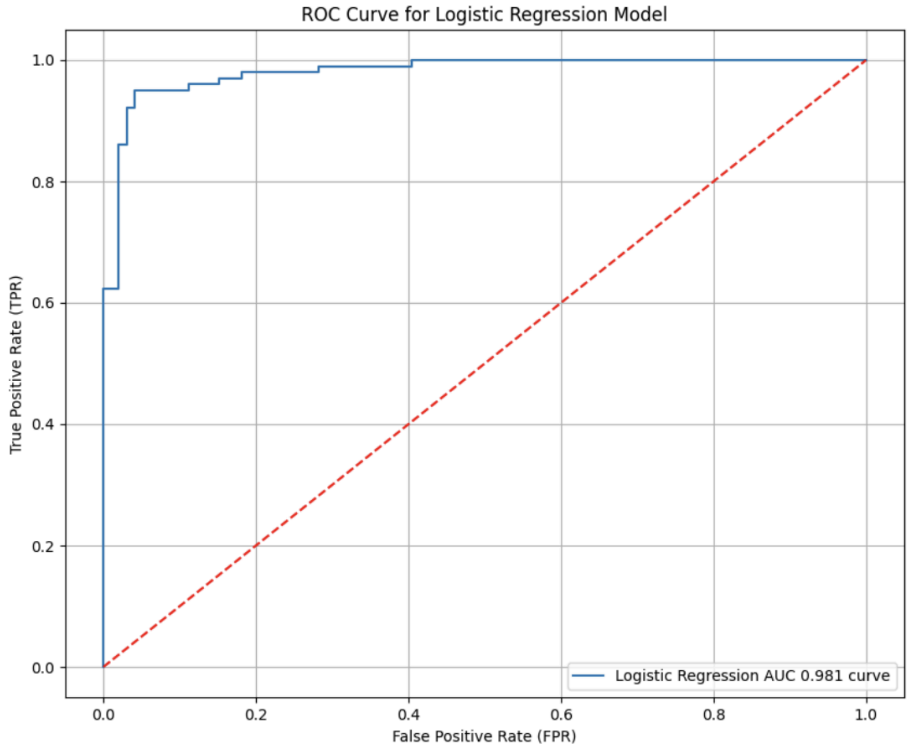
```
Confusion matrix:
[[96  3]
 [ 8 93]]
```



ii.

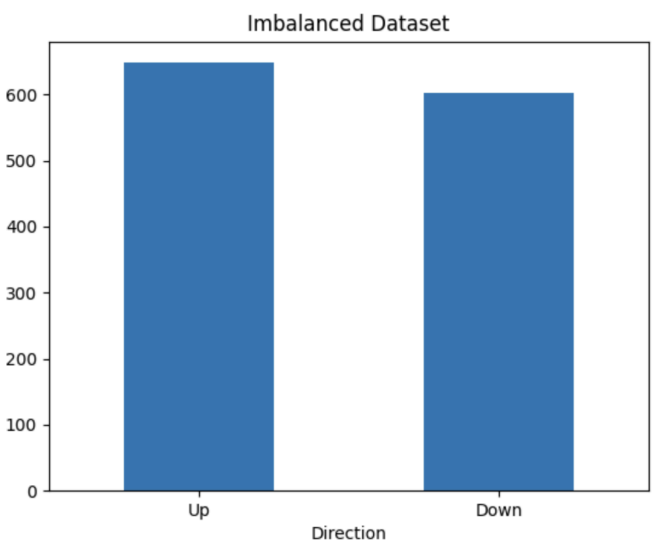
```
Classfification results:
Accuracy: 0.945
Precision: 0.969
Recall: 0.921
```

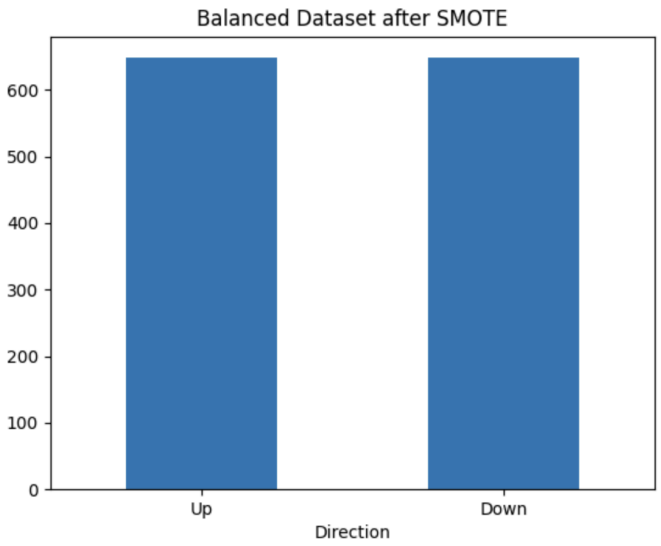
iii.



Q4.

a.






```
Initial distribution of 'Direction':
Direction
Up      648
Down    602
Name: count, dtype: int64
Final distribution of 'Direction' after SMOTE:
Direction
Up      648
Down    648
Name: count, dtype: int64
```

b.

```
Printing first 5 rows of the train dataset...
       Lag1      Lag2      Lag3      Lag4      Lag5    Volume
0 -0.050182  0.484540 -0.555386  0.783935  0.695138  0.250232
1 -1.533748 -1.170734 -1.437867  0.054327  2.503983  0.053777
2 -0.655856  0.766858  0.225759 -0.030037  1.201172 -0.057110
3 -0.274884 -0.562037 -0.741813 -1.081490  0.894284 -0.526171
4  0.126749 -0.333126  0.474321 -0.603802  0.422941  1.257128

Printing first 5 rows of the test dataset...
       Lag1      Lag2      Lag3      Lag4      Lag5    Volume
0 -0.278423  0.033653 -0.001331  1.539694  0.723836 -0.235371
1  0.133826  1.122718 -0.668636 -1.173890 -0.698891 -1.006707
2  0.110499 -0.686959 -0.636660  0.483230 -0.639539  0.270747
3  0.545190 -1.220159 -0.159010  0.079607  0.247274 -0.557289
4  0.554921 -1.448203  0.274826 -0.164468 -1.709409  0.875838
```

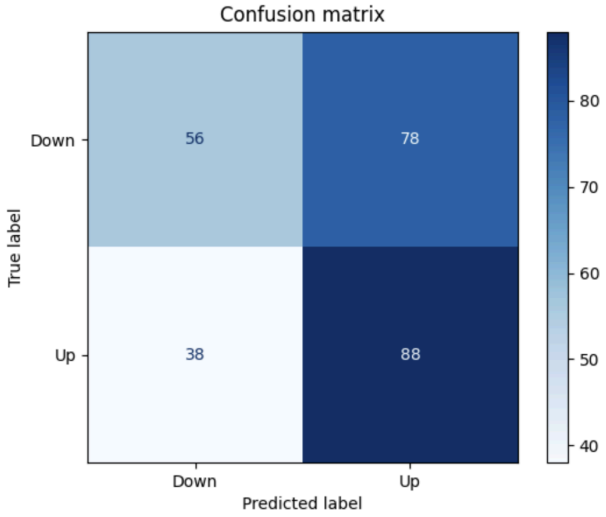
c.

i.

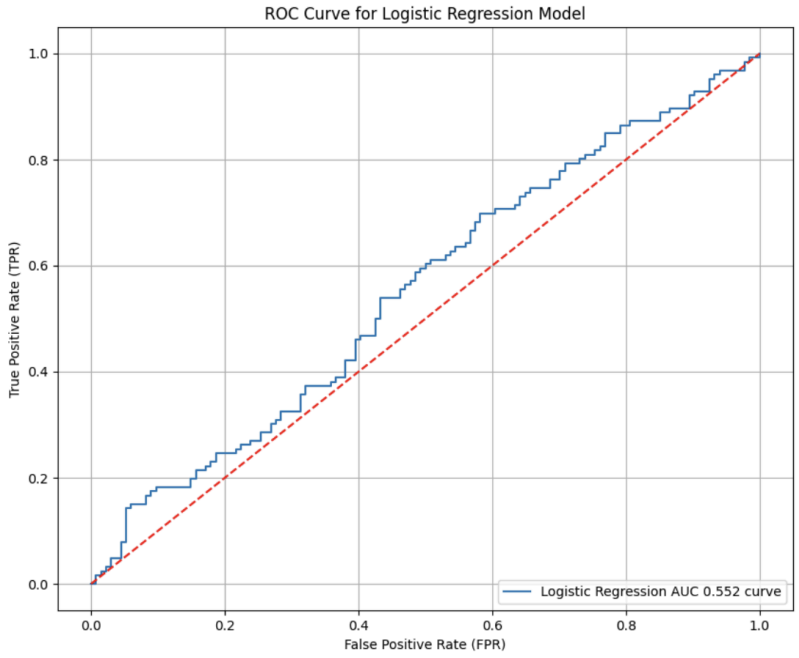
```
Train score: 0.529

Test score: 0.554
```

ii.



iii.



iv.

```
Classfification results:
Accuracy: 0.554
Precision: 0.530
Recall: 0.698
```

d.

```
Best parameters from Grid Search with CV:
{'C': np.float64(0.021544346900318832), 'l1_ratio': np.float64(0.3793103448275862), 'penalty': 'elasticnet'}

Train score (tuned model):, 0.504
Test score (tuned model):, 0.485

Classification Metrics (Tuned Model):
Accuracy: 0.48
Precision: 0.48
Recall: 1.00
F1 Score: 0.65

Did the grid search improve the performance? False
```