Assignment 1

Name: Gautam Kumar

Roll Number: 21CS30020

```
# import all the necessary libraries here
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from numpy.linalg import inv
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay

df =
pd.read_excel('../../dataset/logistic-regression/Pumpkin_Seeds_Dataset
.xlsx')
print(df.shape)
```

```
(2500, 13)
```

```
df.head()
```

```
    Area  Perimeter  Major_Axis_Length  Minor_Axis_Length  Convex_Area
\
0  56276    888.242           326.1485           220.2388        56831

1  76631   1068.146           417.1932           234.2289        77280

2  71623   1082.987           435.8328           211.0457        72663

3  66458    992.051           381.5638           222.5322        67118

4  66107    998.146           383.8883           220.4545        67117


   Equiv_Diameter  Eccentricity  Solidity  Extent  Roundness
Aspect_Ration  \
0        267.6805        0.7376    0.9902  0.7453     0.8963
1.4809
1        312.3614        0.8275    0.9916  0.7151     0.8440
1.7811
2        301.9822        0.8749    0.9857  0.7400     0.7674
2.0651
3        290.8899        0.8123    0.9902  0.7396     0.8486
1.7146
4        290.1207        0.8187    0.9850  0.6752     0.8338
1.7413

   Compactness       Class
```

```
0       0.8207  Çerçevelik
1       0.7487  Çerçevelik
2       0.6929  Çerçevelik
3       0.7624  Çerçevelik
4       0.7557  Çerçevelik
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Area               2500 non-null   int64  
 1   Perimeter          2500 non-null   float64
 2   Major_Axis_Length  2500 non-null   float64
 3   Minor_Axis_Length  2500 non-null   float64
 4   Convex_Area        2500 non-null   int64  
 5   Equiv_Diameter     2500 non-null   float64
 6   Eccentricity       2500 non-null   float64
 7   Solidity           2500 non-null   float64
 8   Extent             2500 non-null   float64
 9   Roundness          2500 non-null   float64
 10  Aspect_Ration      2500 non-null   float64
 11  Compactness        2500 non-null   float64
 12  Class              2500 non-null   object 
dtypes: float64(10), int64(2), object(1)
memory usage: 254.0+ KB
```

```
df.describe()
```

```
                Area    Perimeter  Major_Axis_Length  
Minor_Axis_Length  \
count    2500.000000  2500.000000        2500.000000  
2500.000000
mean    80658.220800  1130.279015         456.601840  
225.794921
std     13664.510228   109.256418          56.235704  
23.297245
min     47939.000000   868.485000         320.844600  
152.171800
25%     70765.000000  1048.829750         414.957850  
211.245925
50%     79076.000000  1123.672000         449.496600  
224.703100
75%     89757.500000  1203.340500         492.737650  
240.672875
max    136574.000000  1559.450000         661.911300  
305.818000
```

```
         Convex_Area  Equiv_Diameter  Eccentricity     Solidity
Extent  \
count    2500.000000     2500.000000   2500.000000  2500.000000
2500.000000
mean    81508.084400      319.334230      0.860879     0.989492
0.693205
std     13764.092788       26.891920      0.045167     0.003494
0.060914
min     48366.000000      247.058400      0.492100     0.918600
0.468000
25%     71512.000000      300.167975      0.831700     0.988300
0.658900
50%     79872.000000      317.305350      0.863700     0.990300
0.713050
75%     90797.750000      338.057375      0.897025     0.991500
0.740225
max    138384.000000      417.002900      0.948100     0.994400
0.829600

         Roundness  Aspect_Ration  Compactness
count  2500.000000    2500.000000  2500.000000
mean      0.791533       2.041702     0.704121
std       0.055924       0.315997     0.053067
min       0.554600       1.148700     0.560800
25%       0.751900       1.801050     0.663475
50%       0.797750       1.984200     0.707700
75%       0.834325       2.262075     0.743500
max       0.939600       3.144400     0.904900
```

```
print(df["Class"].unique())
```

```
['Çerçevelik' 'Ürgüp Sivrisi']
```

```
mapping = {'Çerçevelik': 0 , 'Ürgüp Sivrisi' : 1}
df.replace({'Class': mapping} , inplace=True)
df.head()
```

```
    Area  Perimeter  Major_Axis_Length  Minor_Axis_Length  Convex_Area
\
0  56276    888.242           326.1485           220.2388        56831

1  76631   1068.146           417.1932           234.2289        77280

2  71623   1082.987           435.8328           211.0457        72663

3  66458    992.051           381.5638           222.5322        67118

4  66107    998.146           383.8883           220.4545        67117


   Equiv_Diameter  Eccentricity  Solidity  Extent  Roundness
```

```
Aspect_Ration  \
0        267.6805        0.7376    0.9902  0.7453     0.8963
1.4809
1        312.3614        0.8275    0.9916  0.7151     0.8440
1.7811
2        301.9822        0.8749    0.9857  0.7400     0.7674
2.0651
3        290.8899        0.8123    0.9902  0.7396     0.8486
1.7146
4        290.1207        0.8187    0.9850  0.6752     0.8338
1.7413

   Compactness  Class
0       0.8207      0
1       0.7487      0
2       0.6929      0
3       0.7624      0
4       0.7557      0
```

```
X = df.iloc[:,:-1].values
Y = df.iloc[:,-1].values
X_train , X,Y_train,Y =
train_test_split(X,Y,test_size=0.5,random_state=0)
X_val,X_test,Y_val,Y_test = train_test_split(X,Y,test_size =
0.4,random_state = 0)


from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
X_train= st_x.fit_transform(X_train)
X_test= st_x.transform(X_test)

print(X_train.shape, Y_train.shape)
print(X_val.shape, Y_val.shape)
print(X_test.shape, Y_test.shape)
```

```
(1250, 12) (1250,)
(750, 12) (750,)
(500, 12) (500,)
```

```
class logistic_regression():
    def __init__(self, epoch= 15000, learning_rate = 0.001 ):
        self.epoch = epoch
        self.learning_rate = learning_rate
        self.cost = []
        self.init_weight = None
        self.final_weight = None

    def initialize_weight(self,n_feature):
        limit = np.sqrt(1/n_feature)
        weight = np.random.uniform(-limit,limit,(n_feature,1))
```

```
        b = 0
        self.init_weight = np.insert(weight,0,b,axis = 0)


    def train(self, X,Y,X_val,Y_val):
        n_sample ,n_feature = X.shape
        X = np.insert(X,0,1,axis = 1)
        Y = np.reshape(Y,(n_sample,1))
        nv_sample = X_val.shape[0];
        X_val = np.insert(X_val,0,1,axis = 1);
        Y_val = np.reshape(Y_val,(nv_sample,1));
        self.initialize_weight(n_feature)
        self.fit(X,Y,X_val,Y_val)


    def fit(self,X,Y,X_val,Y_val):
        _weight = self.init_weight.copy()
        y_pred = self.sigmoid(np.dot(X,_weight))
        self.cost.append(self.gradient_cost(X,Y,_weight))

        for iter in range(self.epoch):
            y_pred = self.sigmoid(np.dot(X,_weight))
            grad = np.dot(X.T, y_pred - Y)
            _weight = _weight - self.learning_rate*grad
            self.cost.append(self.gradient_cost(X,Y,_weight))
            if iter%100 ==0:
                print(f"The training cost for iteration ::{iter} is
_____________________________________________{np.squeeze(self.cost[
-1])}","\n......")
        self.final_weight = _weight
        return


    def predict(self,X):
        out = np.dot(X,self.final_weight)
        out = self.sigmoid(out)
        out = (out >= 0.5)*1
        return out

    def sigmoid(self,Y):
        sig = 1 + np.exp(-1*Y)
        sig = 1/sig
        return sig

    def gradient_cost(self,X,Y,_weight):
        y_pred = self.sigmoid(np.dot(X,_weight))
        return np.mean(-1*(Y*np.log(y_pred) + (1-Y)*np.log(1 -
y_pred)))
```

```
    def viswalize_loss(self):
        figure, ax = plt.subplots()
        nums = np.arange(len(self.cost))
        ax.plot(nums, np.array(self.cost).reshape((len(self.cost,))))
        ax.set_xlabel('Epoch')
        ax.set_ylabel('Cost')
        plt.show()

    def metrics_loss(self,X,Y):
        n_sample,n_feature = X.shape
        X = np.insert(X,0,1,axis = 1)
        Y = np.reshape(Y,(n_sample,1))
        y_pred = self.sigmoid(np.dot(X,self.final_weight))
        y_pred = (y_pred >= 0.5)
        con_matrix = confusion_matrix(Y,y_pred)
        cm_display = ConfusionMatrixDisplay(confusion_matrix =
con_matrix, display_labels = [False, True])
        cm_display.plot()
        plt.show()
        recall = con_matrix[1][1]/(con_matrix[1][0] + con_matrix[1]
[1])
        precison = con_matrix[1][1] /(con_matrix[1][1] + con_matrix[0]
[1])
        accuracy = (con_matrix[0][0] + con_matrix[1]
[1])/(con_matrix[0][0] + con_matrix[0][1] + con_matrix[1][0] +
con_matrix[1][1])
        df = pd.DataFrame([[recall, precison, accuracy]],
columns=['Recall','Precision', 'Mean Accuracy'])
        return df

    def print_loss(self):
        print(self.cost)



logistic_regressor = logistic_regression()

logistic_regressor.train(X_train,Y_train,X_val,Y_val)

The training cost for iteration ::0 is
_____________________________________________0.33958563353374965
......
The training cost for iteration ::100 is
_____________________________________________0.3143659029205259
......
The training cost for iteration ::200 is
_____________________________________________0.31298912246293287
......
The training cost for iteration ::300 is
_____________________________________________0.3117200765150373
```

```
......
The training cost for iteration ::400 is
_____________________________________________________0.3105483882430749
......
The training cost for iteration ::500 is
_____________________________________________________0.30946511667783855
......
The training cost for iteration ::600 is
_____________________________________________________0.3084623582810819
......
The training cost for iteration ::700 is
_____________________________________________________0.30753310998426475
......
The training cost for iteration ::800 is
_____________________________________________________0.30667124084556024
......
The training cost for iteration ::900 is
_____________________________________________________0.3058715389732966
......
The training cost for iteration ::1000 is
_____________________________________________________0.30512982070815686
......
The training cost for iteration ::1100 is
_____________________________________________________0.3044430559703882
......
The training cost for iteration ::1200 is
_____________________________________________________0.3038093738368832
......
The training cost for iteration ::1300 is
_____________________________________________________0.3032277022295546
......
The training cost for iteration ::1400 is
_____________________________________________________0.3026968552707426
......
The training cost for iteration ::1500 is
_____________________________________________________0.3022143535503374
......
The training cost for iteration ::1600 is
_____________________________________________________0.3017758178904056
......
The training cost for iteration ::1700 is
_____________________________________________________0.3013754791752596
......
The training cost for iteration ::1800 is
_____________________________________________________0.3010073253986448
......
The training cost for iteration ::1900 is
_____________________________________________________0.30066603092866834
......
```

```
The training cost for iteration ::2000 is
_____________________________________________________0.300347326031182
......
The training cost for iteration ::2100 is
_____________________________________________________0.30004795802148
......
The training cost for iteration ::2200 is
_____________________________________________________0.299765488233756
......
The training cost for iteration ::2300 is
_____________________________________________________0.29949807315037835
......
The training cost for iteration ::2400 is
_____________________________________________________0.29924428565791966
......
The training cost for iteration ::2500 is
_____________________________________________________0.2990029852625272
......
The training cost for iteration ::2600 is
_____________________________________________________0.298773229330753
......
The training cost for iteration ::2700 is
_____________________________________________________0.2985542142813959
......
The training cost for iteration ::2800 is
_____________________________________________________0.29834523719020983
......
The training cost for iteration ::2900 is
_____________________________________________________0.2981456707769319
......
The training cost for iteration ::3000 is
_____________________________________________________0.29795494696814223
......
The training cost for iteration ::3100 is
_____________________________________________________0.2977725458877133
......
The training cost for iteration ::3200 is
_____________________________________________________0.2975979882664914
......
The training cost for iteration ::3300 is
_____________________________________________________0.29743083001129883
......
The training cost for iteration ::3400 is
_____________________________________________________0.2972706581509636
......
The training cost for iteration ::3500 is
_____________________________________________________0.2971170876761652
......
The training cost for iteration ::3600 is
```

```
_____0.2969697589748651
......
The training cost for iteration ::3700 is
_____0.29682833567848427
......
The training cost for iteration ::3800 is
_____0.2966925028032118
......
The training cost for iteration ::3900 is
_____0.296561965112941
......
The training cost for iteration ::4000 is
_____0.29643644565600474
......
The training cost for iteration ::4100 is
_____0.29631568444361
......
The training cost for iteration ::4200 is
_____0.29619943724755177
......
The training cost for iteration ::4300 is
_____0.29608747450089967
......
The training cost for iteration ::4400 is
_____0.29597958028922067
......
The training cost for iteration ::4500 is
_____0.295875551422466
......
The training cost for iteration ::4600 is
_____0.2957751965794106
......
The training cost for iteration ::4700 is
_____0.2956783355177691
......
The training cost for iteration ::4800 is
_____0.2955847983440341
......
The training cost for iteration ::4900 is
_____0.29549442483781174
......
The training cost for iteration ::5000 is
_____0.2954070638260047
......
The training cost for iteration ::5100 is
_____0.2953225726026603
......
The training cost for iteration ::5200 is
_____0.295240816390739
```

```
......
The training cost for iteration ::5300 is
_____________________________________________________0.29516166784239484
......
The training cost for iteration ::5400 is
_____________________________________________________0.2950850065746761
......
The training cost for iteration ::5500 is
_____________________________________________________0.2950107187378407
......
The training cost for iteration ::5600 is
_____________________________________________________0.29493869661372574
......
The training cost for iteration ::5700 is
_____________________________________________________0.2948688382418357
......
The training cost for iteration ::5800 is
_____________________________________________________0.29480104707101035
......
The training cost for iteration ::5900 is
_____________________________________________________0.2947352316347255
......
The training cost for iteration ::6000 is
_____________________________________________________0.29467130524823393
......
The training cost for iteration ::6100 is
_____________________________________________________0.2946091857259151
......
The training cost for iteration ::6200 is
_____________________________________________________0.29454879511732585
......
The training cost for iteration ::6300 is
_____________________________________________________0.2944900594605785
......
The training cost for iteration ::6400 is
_____________________________________________________0.29443290855177545
......
The training cost for iteration ::6500 is
_____________________________________________________0.294377275729341
......
The training cost for iteration ::6600 is
_____________________________________________________0.29432309767217923
......
The training cost for iteration ::6700 is
_____________________________________________________0.294270314210668
......
The training cost for iteration ::6800 is
_____________________________________________________0.2942188681495869
......
```

```
The training cost for iteration ::6900 is
_____________________________________________________0.29416870510213905
......
The training cost for iteration ::7000 is
_____________________________________________________0.2941197733342938
......
The training cost for iteration ::7100 is
_____________________________________________________0.29407202361874263
......
The training cost for iteration ::7200 is
_____________________________________________________0.2940254090978021
......
The training cost for iteration ::7300 is
_____________________________________________________0.2939798851546635
......
The training cost for iteration ::7400 is
_____________________________________________________0.2939354092924178
......

The training cost for iteration ::7500 is
_____________________________________________________0.2938919410203448
......
The training cost for iteration ::7600 is
_____________________________________________________0.29384944174697386
......
The training cost for iteration ::7700 is
_____________________________________________________0.29380787467947783
......
The training cost for iteration ::7800 is
_____________________________________________________0.29376720472897894
......
The training cost for iteration ::7900 is
_____________________________________________________0.2937273984213862
......
The training cost for iteration ::8000 is
_____________________________________________________0.29368842381340626
......
The training cost for iteration ::8100 is
_____________________________________________________0.2936502504133959
......
The training cost for iteration ::8200 is
_____________________________________________________0.293612849106746
......
The training cost for iteration ::8300 is
_____________________________________________________0.2935761920855168
......
The training cost for iteration ::8400 is
_____________________________________________________0.2935402527820469
......
The training cost for iteration ::8500 is
```

```
_____0.2935050058063021
......
The training cost for iteration ::8600 is
_____0.29347042688671754
......
The training cost for iteration ::8700 is
_____0.29343649281433154
......
The training cost for iteration ::8800 is
_____0.29340318138999955
......
The training cost for iteration ::8900 is
_____0.2933704713745082
......
The training cost for iteration ::9000 is
_____0.2933383424414113
......
The training cost for iteration ::9100 is
_____0.2933067751324231
......
The training cost for iteration ::9200 is
_____0.2932757508152218
......
The training cost for iteration ::9300 is
_____0.293245251643514
......
The training cost for iteration ::9400 is
_____0.29321526051923247
......
The training cost for iteration ::9500 is
_____0.2931857610567382
......
The training cost for iteration ::9600 is
_____0.29315673754891314
......
The training cost for iteration ::9700 is
_____0.29312817493503435
......
The training cost for iteration ::9800 is
_____0.29310005877032264
......
The training cost for iteration ::9900 is
_____0.29307237519707824
......
The training cost for iteration ::10000 is
_____0.2930451109173031
......
The training cost for iteration ::10100 is
_____0.29301825316673746
```

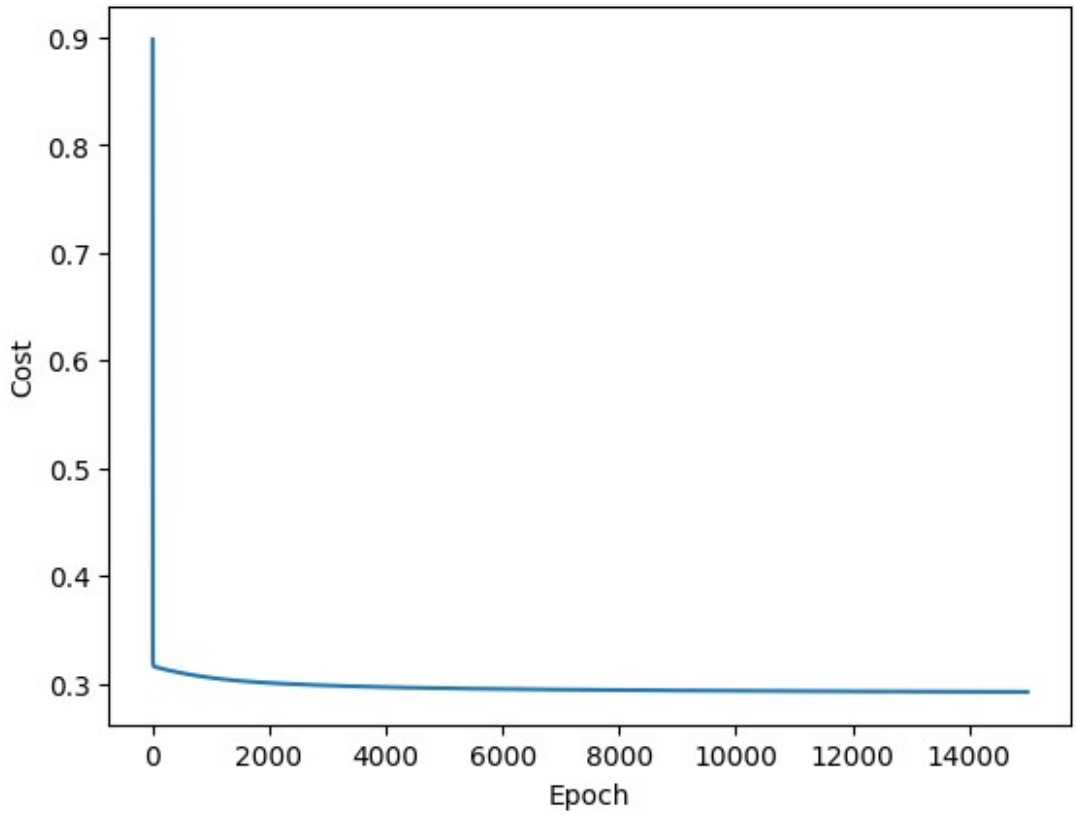
```
......
The training cost for iteration ::10200 is
_____________________________________________________0.2929917896902186
......
The training cost for iteration ::10300 is
_____________________________________________________0.29296570871829963
......
The training cost for iteration ::10400 is
_____________________________________________________0.2929399989450493
......
The training cost for iteration ::10500 is
_____________________________________________________0.29291464950697427
......
The training cost for iteration ::10600 is
_____________________________________________________0.29288964996299516
......
The training cost for iteration ::10700 is
_____________________________________________________0.292864990275427
......
The training cost for iteration ::10800 is
_____________________________________________________0.29284066079190163
......
The training cost for iteration ::10900 is
_____________________________________________________0.29281665222818704
......
The training cost for iteration ::11000 is
_____________________________________________________0.29279295565184976
......
The training cost for iteration ::11100 is
_____________________________________________________0.29276956246672164
......
The training cost for iteration ::11200 is
_____________________________________________________0.29274646439812174
......
The training cost for iteration ::11300 is
_____________________________________________________0.29272365347879814
......
The training cost for iteration ::11400 is
_____________________________________________________0.29270112203554716
......
The training cost for iteration ::11500 is
_____________________________________________________0.29267886267648047
......
The training cost for iteration ::11600 is
_____________________________________________________0.2926568682788987
......
The training cost for iteration ::11700 is
_____________________________________________________0.29263513197774627
......
```

```
The training cost for iteration ::11800 is
_____________________________________________________0.292613647154614
......
The training cost for iteration ::11900 is
_____________________________________________________0.29259240742726295
......
The training cost for iteration ::12000 is
_____________________________________________________0.29257140663964065
......
The training cost for iteration ::12100 is
_____________________________________________________0.2925506388523657
......
The training cost for iteration ::12200 is
_____________________________________________________0.29253009833365834
......
The training cost for iteration ::12300 is
_____________________________________________________0.2925097795506923
......
The training cost for iteration ::12400 is
_____________________________________________________0.29248967716134705
......
The training cost for iteration ::12500 is
_____________________________________________________0.29246978600634
......
The training cost for iteration ::12600 is
_____________________________________________________0.2924501011017225
......
The training cost for iteration ::12700 is
_____________________________________________________0.2924306176317169
......
The training cost for iteration ::12800 is
_____________________________________________________0.2924113309418809
......
The training cost for iteration ::12900 is
_____________________________________________________0.2923922365325834
......
The training cost for iteration ::13000 is
_____________________________________________________0.29237333005277316
......
The training cost for iteration ::13100 is
_____________________________________________________0.29235460729402946
......
The training cost for iteration ::13200 is
_____________________________________________________0.2923360641848809
......
The training cost for iteration ::13300 is
_____________________________________________________0.2923176967853741
......
The training cost for iteration ::13400 is
```

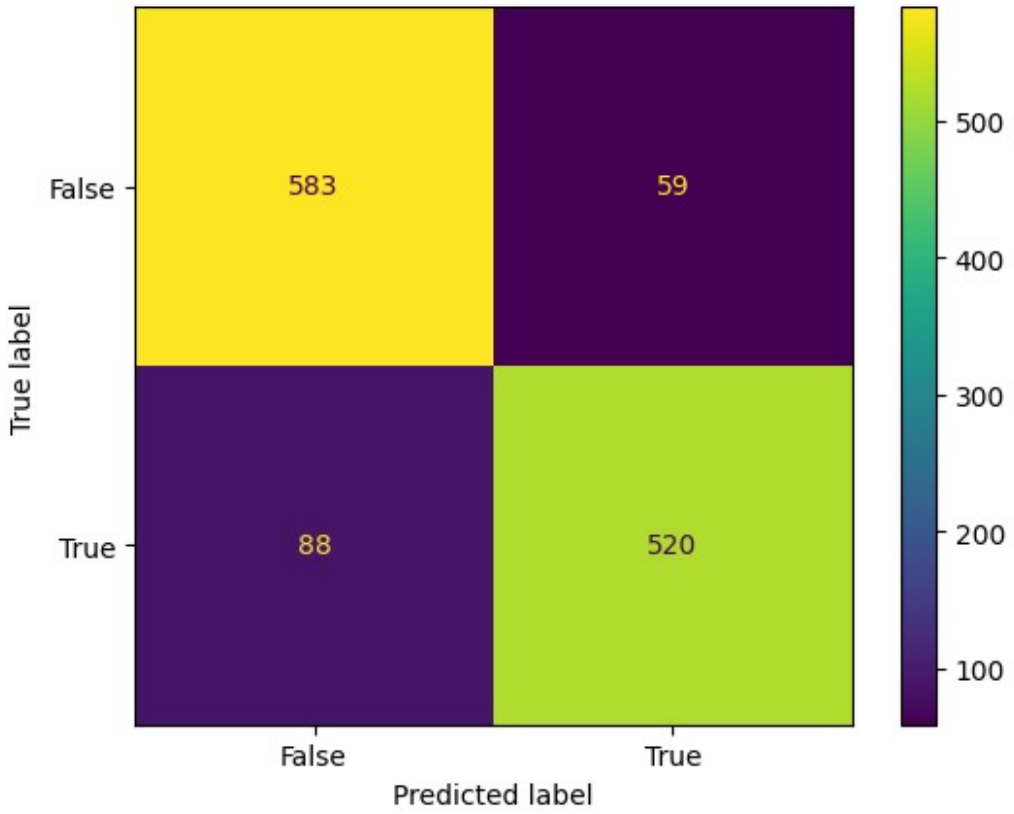
```
_____0.29229950128188786
......
The training cost for iteration ::13500 is
_____0.29228147398217613
......
The training cost for iteration ::13600 is
_____0.29226361131062856
......
The training cost for iteration ::13700 is
_____0.292245909803741
......
The training cost for iteration ::13800 is
_____0.29222836610578373
......
The training cost for iteration ::13900 is
_____0.2922109769646587
......
The training cost for iteration ::14000 is
_____0.2921937392279365
......
The training cost for iteration ::14100 is
_____0.29217664983906677
......
The training cost for iteration ::14200 is
_____0.2921597058337487
......
The training cost for iteration ::14300 is
_____0.29214290433645945
......
The training cost for iteration ::14400 is
_____0.2921262425571334
......
The training cost for iteration ::14500 is
_____0.29210971778797695
......

The training cost for iteration ::14600 is
_____0.29209332740042393
......
The training cost for iteration ::14700 is
_____0.29207706884221796
......
The training cost for iteration ::14800 is
_____0.2920609396346169
......
The training cost for iteration ::14900 is
_____0.29204493736971576
......

logistic_regressor.viswalize_loss()
```



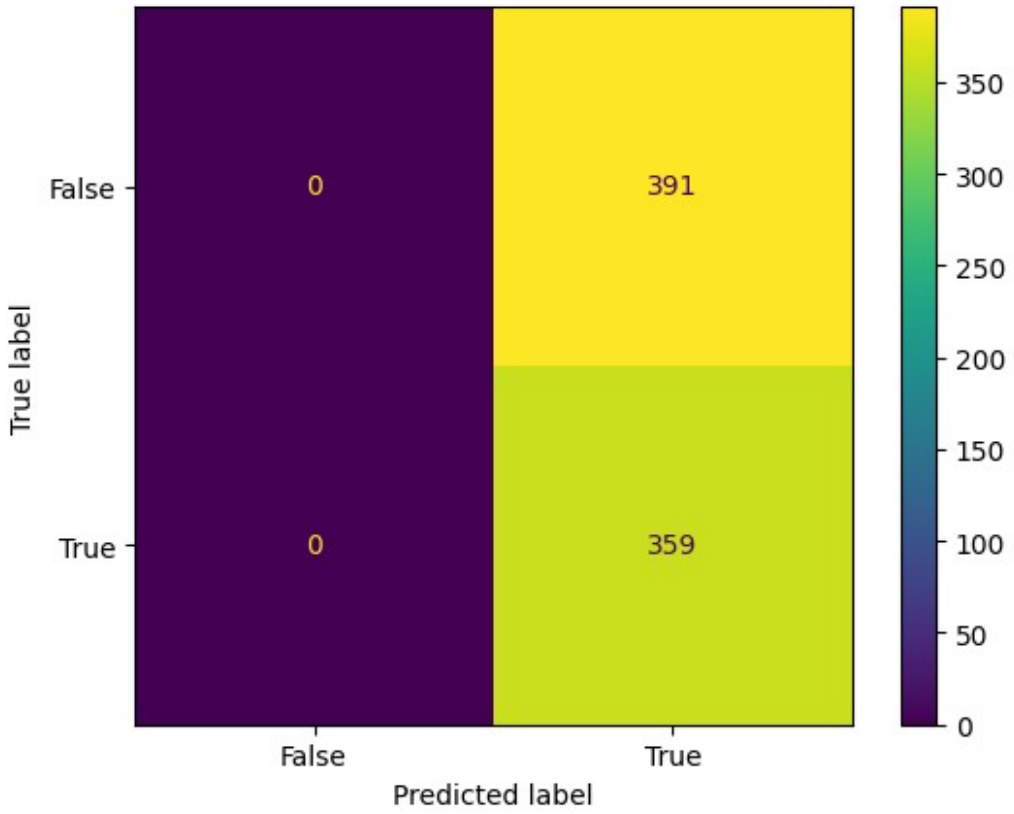

```
mat_loss_df = logistic_regressor.metrics_loss(X_train,Y_train)
mat_loss_df.rename(index={0:'Train_data'},inplace=True)
```




```
mat_loss_df
```

	Recall	Precision	Mean Accuracy
Train_data	0.855263	0.8981	0.8824

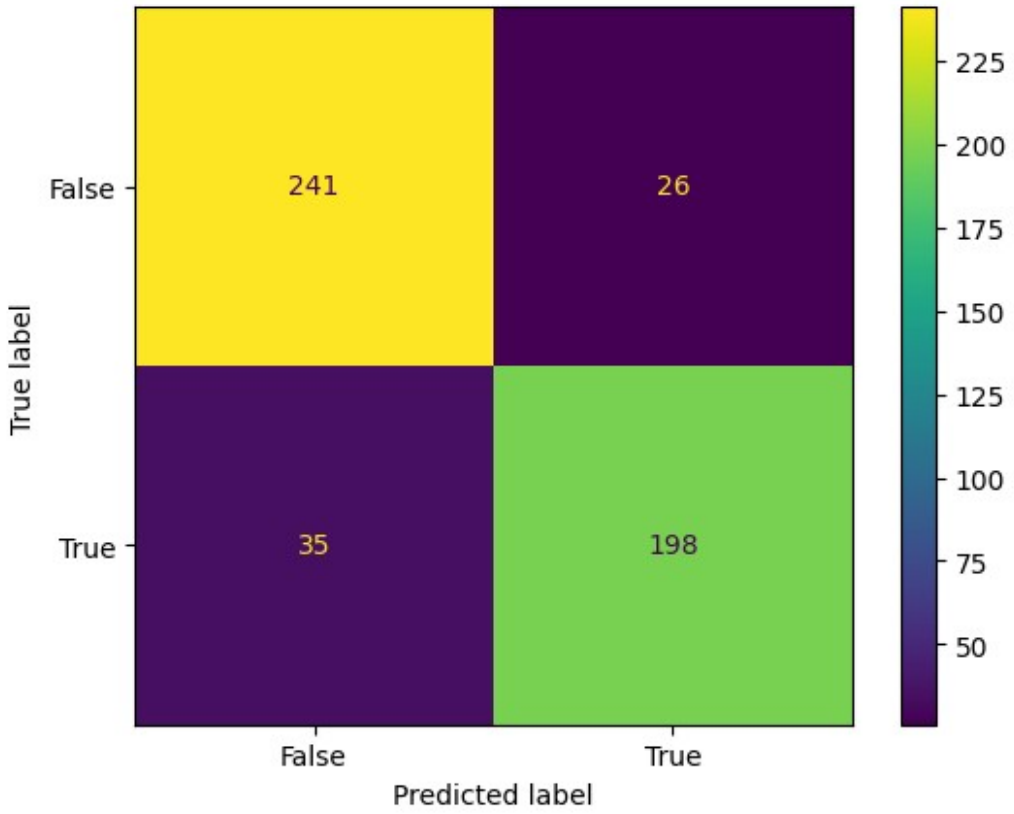
```
mat_loss_df = logistic_regressor.metrics_loss(X_val,Y_val)
mat_loss_df.rename(index={0:'Validation_data'},inplace=True)
```




```
mat_loss_df
```

	Recall	Precision	Mean Accuracy
Validation_data	1.0	0.478667	0.478667

```
mat_loss_df = logistic_regressor.metrics_loss(X_test,Y_test)
mat_loss_df.rename(index={0:'Test_data'},inplace=True)
```




```
mat_loss_df
```

	Recall	Precision	Mean Accuracy
Test_data	0.849785	0.883929	0.878