code

August 17, 2023

0.1 Assignment 1

0.1.1 Name: Ritesah M

0.1.2 Roll Number: 21CS30042

[1]:
```
# import all the necessary libraries here
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

[2]:
```
df = pd.read_excel('../../dataset/logistic-regression/Pumpkin_Seeds_Dataset.
↪xlsx')
print(df.shape)
```

```
(2500, 13)
```

[3]:
```
df["Class"].unique()
```

[3]:
```
array(['Çerçevelik', 'Ürgüp Sivrisi'], dtype=object)
```

[4]:
```
df.loc[df["Class"]=="Çerçevelik","Class"]=1
df.loc[df["Class"]=="Ürgüp Sivrisi","Class"]=0
```

[5]:
```
for column in df.columns:
    print(f"Max value {column}: {df[column].max()}")
    print(f"Min value {column}: {df[column].min()}")
    print("\n")
```

```
Max value Area: 136574
Min value Area: 47939


Max value Perimeter: 1559.45
Min value Perimeter: 868.485


Max value Major_Axis_Length: 661.9113
Min value Major_Axis_Length: 320.8446
```

```
Max value Minor_Axis_Length: 305.818
Min value Minor_Axis_Length: 152.1718


Max value Convex_Area: 138384
Min value Convex_Area: 48366


Max value Equiv_Diameter: 417.0029
Min value Equiv_Diameter: 247.0584


Max value Eccentricity: 0.9481
Min value Eccentricity: 0.4921


Max value Solidity: 0.9944
Min value Solidity: 0.9186


Max value Extent: 0.8296
Min value Extent: 0.468


Max value Roundness: 0.9396
Min value Roundness: 0.5546


Max value Aspect_Ration: 3.1444
Min value Aspect_Ration: 1.1487


Max value Compactness: 0.9049
Min value Compactness: 0.5608


Max value Class: 1
Min value Class: 0
```

[6]:

```
X= df.drop("Class",axis=1)
y= df["Class"]
# y
```

[7]:
```python
df.isna().any()
```

[7]:
```
Area                   False
Perimeter              False
Major_Axis_Length      False
Minor_Axis_Length      False
Convex_Area            False
Equiv_Diameter         False
Eccentricity           False
Solidity               False
Extent                 False
Roundness              False
Aspect_Ration          False
Compactness            False
Class                  False
dtype: bool
```

[8]:
```python
from sklearn.model_selection import train_test_split
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.5,␣
 ↪random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.4,␣
 ↪random_state=42)
```

[9]:
```python
X_train=X_train.to_numpy()
ones_column_train= np.ones((X_train.shape[0],1))
y_train=y_train.to_numpy(dtype="float64")
X_train=np.hstack((ones_column_train,X_train))

X_val=X_val.to_numpy()
ones_column_val= np.ones((X_val.shape[0],1))
y_val=y_val.to_numpy(dtype="float64")
X_val=np.hstack((ones_column_val,X_val))

X_test=X_test.to_numpy()
ones_column_test= np.ones((X_test.shape[0],1))
y_test=y_test.to_numpy(dtype="float64")
X_test=np.hstack((ones_column_test,X_test))
```

[10]:
```python
def calculate_rmse(y_actual, y_pred):
    squared_errors = (y_actual - y_pred) ** 2
    mean_squared_error = np.mean(squared_errors)
    rmse = np.sqrt(mean_squared_error)
    return rmse

def calculate_r_squared(y_actual, y_pred):
    total_variance = np.sum((y_actual - np.mean(y_actual)) ** 2)
    explained_variance = np.sum((y_pred - y_actual) ** 2)
```

```
    r_squared = 1 - (explained_variance / total_variance)
    return r_squared
```

[11]:
```
def normalize_matrix(X):
    columns_to_normalize = range(1, X.shape[1])
    first_column = X[:, 0]
    normalized_columns = (X[:, columns_to_normalize] - X[:,␣
 ↪columns_to_normalize].mean(axis=0)) / X[:, columns_to_normalize].std(axis=0)
    X_scaled = np.column_stack((first_column, normalized_columns))

    print("Original Matrix:")
    print(X)

    print("\nNormalized Matrix:")
    print(X_scaled)
    return X_scaled
X_train_scaled=normalize_matrix(X_train)
```

```
Original Matrix:
[[1.000000e+00 8.858100e+04 1.198677e+03 … 7.747000e-01 2.046500e+00
  6.966000e-01]
 [1.000000e+00 8.472400e+04 1.099715e+03 … 8.804000e-01 1.573100e+00
  7.944000e-01]
 [1.000000e+00 9.713200e+04 1.310223e+03 … 7.110000e-01 2.555400e+00
  6.236000e-01]
 …
 [1.000000e+00 9.413200e+04 1.182420e+03 … 8.461000e-01 1.739200e+00
  7.567000e-01]
 [1.000000e+00 8.119800e+04 1.118627e+03 … 8.154000e-01 1.906200e+00
  7.226000e-01]
 [1.000000e+00 7.875900e+04 1.115483e+03 … 7.954000e-01 2.023800e+00
  7.021000e-01]]

Normalized Matrix:
[[ 1.          0.60882926  0.6459801  … -0.30180065  0.01488933
  -0.14217799]
 [ 1.          0.32818862 -0.24722132 …  1.58513179 -1.47631136
   1.68881593]
 [ 1.          1.2310118   1.65276095 … -1.43895861  1.61791432
  -1.50887079]
 …
 [ 1.          1.01272767  0.49924928 …  0.97281596 -0.95309963
   0.98300334]
 [ 1.          0.07163201 -0.07652727 …  0.42476652 -0.42705291
   0.34458931]
 [ 1.         -0.105833   -0.10490407 …  0.06773106 -0.05661523
  -0.03920798]]
```

[12]:
```
X_train
```

[12]:
```
array([[1.000000e+00, 8.858100e+04, 1.198677e+03, …, 7.747000e-01,
         2.046500e+00, 6.966000e-01],
        [1.000000e+00, 8.472400e+04, 1.099715e+03, …, 8.804000e-01,
         1.573100e+00, 7.944000e-01],
        [1.000000e+00, 9.713200e+04, 1.310223e+03, …, 7.110000e-01,
         2.555400e+00, 6.236000e-01],
        …,
        [1.000000e+00, 9.413200e+04, 1.182420e+03, …, 8.461000e-01,
         1.739200e+00, 7.567000e-01],
        [1.000000e+00, 8.119800e+04, 1.118627e+03, …, 8.154000e-01,
         1.906200e+00, 7.226000e-01],
        [1.000000e+00, 7.875900e+04, 1.115483e+03, …, 7.954000e-01,
         2.023800e+00, 7.021000e-01]])
```

[13]:
```
X_train_scaled
print(X_train_scaled.shape)
```

```
(1250, 13)
```

[14]:
```
def mean_squared(y_train,y_pred):
    cost=0
    m= y_train.shape[0]
    for i in range(m):
        cost += (y_train[i]-y_pred[i])**2
    error = cost/(2*m)
    return error
```

[15]:
```
def sigmoid(x):
    ans= np.exp(-1*x)
    ans+=1
    ans= 1/ans
    return ans
```

[16]:
```
sigmoid(0)
```

[16]:
```
0.5
```

[17]:
```
train_losses=[]
val_losses=[]
def gradient_descent(X,y,X_val,y_val,w,alpha):
        m=X.shape[0]
        n=X.shape[1]
        pred=np.dot(X,w)
        pred=sigmoid(pred)
        error=pred-y
```

```
        gradient= np.dot(X.T,error)/m
        w-=alpha*gradient
        loss=mean_squared(y,pred)
        train_losses.append(loss)
        return w
```

[18]:
```
def logistic_regression(X,y,X_val,y_val,w,alpha,steps):
    for iter in range(steps):
        w=gradient_descent(X,y,X_val,y_val,w,alpha)
```

[19]:
```
w=np.zeros(X_train.shape[1])
print(w.shape)
```

```
(13,)
```

[20]:
```
logistic_regression(X_train_scaled,y_train,X_val,y_val,w,0.01,1000)
```

[21]:
```
print(train_losses)
```

```
[0.125, 0.12319409543362954, 0.12143812230465792, 0.11973125116126794,
0.11807258722608219, 0.11646117976665132, 0.11489603100053612,
0.11337610448560956, 0.11190033296056671, 0.11046762561370396,
0.10907687476960919, 0.10772696199349441, 0.10641676362142341,
0.105145155731761, 0.10391101857881958, 0.10271324051403485,
0.10155072142320916, 0.10042237571052016, 0.09932713486127183,
0.09826394961588152, 0.09723179178749444, 0.09622965575500238,
0.09525655966222522, 0.09431154635270873, 0.09339368406804723,
0.09250206693596022, 0.09163581527258173, 0.09079407572160665,
0.08997602125113113, 0.08918085102724749, 0.08840779018173277,
0.08765608948952344, 0.08692502497011374, 0.08621389742554518,
0.08552203192629775, 0.08484877725512614, 0.08419350531772521,
0.08355561052804339, 0.08293450917510091, 0.08232963877728054,
0.08174045742928217, 0.0811664431462039, 0.08060709320858933,
0.08006192351169901, 0.07953046792176553, 0.07901227764154017,
0.0785069205870403, 0.07801398077706305, 0.07753305773671673,
0.07706376591596224, 0.07660573412391537, 0.07615860497946421,
0.07572203437857855, 0.07529569097853805, 0.07487925569917686,
0.07447242124113439, 0.07407489162100842, 0.07368638172322935,
0.07330661686840745, 0.0729353323978587, 0.07257227327396482,
0.0722171936959904, 0.07186985673095582, 0.0715300339591438,
0.07119750513379934, 0.07087205785457783, 0.07055348725428628,
0.07024159569846544, 0.06993619249735462, 0.06963709362979224,
0.06934412147860412, 0.06905710457704006, 0.06877587736583322,
0.0685002799604581, 0.068230157928178, 0.06796536207448503, 0.06770574823854612,
0.06745117709727765, 0.06720151397768957, 0.06695662867714812,
0.06671639529121812, 0.06648069204876277, 0.06624940115398502,
0.06602240863511273, 0.06579960419943694, 0.06558088109442674,
```

0.06536613597465422, 0.06515526877427773, 0.0649481825848338,
0.0647447835381097, 0.06454498069387021, 0.06434868593222452,
0.06415581385042962, 0.06396628166393448, 0.06378000911147867,
0.06359691836406468, 0.06341693393763631, 0.06323998260929815,
0.0630659933369212, 0.0628948971819869 4, 0.06272662723552751,
0.06256111854702444, 0.06239830805614256, 0.06223813452716816,
0.06208053848604128, 0.06192546215986426, 0.06177284941878162,
0.061622645720130376, 0.06147479805475948, 0.061329254895430015,
0.061185966147203466, 0.061044883099733695, 0.060905958381385676,
0.06076914591509715, 0.06063440087591464, 0.06050167965013295,
0.06037093979596634, 0.0602421400056948, 0.060115240069215634,
0.05999020083894968, 0.05986698419604073, 0.059745553017797194,
0.05962587114632461, 0.05950790335830039, 0.059391615335843494,
0.05927697363843625, 0.05916394567585363, 0.05905249968206007,
0.05894260469003724, 0.05883423050750146, 0.058727347693478774,
0.05862192753570205, 0.05851794202879735, 0.058415363853230484,
0.058314166354980906, 0.0582143235259181 2, 0.05811580998484875,
0.05801860095921391, 0.057922672267406314, 0.057828000301687436,
0.05773456201167917, 0.05764233488840996, 0.057551296948891835,
0.05746142672121223, 0.05737270323011711, 0.057285105983069344,
0.05719861495676434, 0.05711321058408508, 0.057028873741481245,
0.05694558573675675, 0.05686332829724886, 0.05678208355838841,
0.05670183405262267, 0.056622562698691696, 0.056544252791242 22,
0.056466887990769404, 0.05639045231387372, 0.05631493012382072,
0.05624030612139526, 0.05616656533603 7, 0.05609369311725055,
0.05602167512627713, 0.055950497328020675, 0.0558801459832202,
0.055810607640857594, 0.055741869130794215, 0.055673917556628635,
0.05560674028876786, 0.05554032495770288, 0.05547465944748469,
0.055409731889392466, 0.055345530655785444, 0.05528204435413747,
0.055219261821242455, 0.055157172117588846, 0.05509576452189533,
0.055035028525805094, 0.05497495382872985, 0.05491553033284166,
0.054856748138206686, 0.054798597538055785, 0.0547410690141887 2,
0.05468415323250639, 0.054627841038667954, 0.05457212345386817,
0.05451699167073225, 0.05446243704932182, 0.0544084511132537,
0.05435502554592075, 0.05430215218681833, 0.054249823027969046,
0.05419803021044359, 0.05414676602097687, 0.05409602288867207,
0.05404579338179386, 0.0539960702046474, 0.05394684619453869,
0.05389811431881493, 0.0538498676719831, 0.053802099472903694,
0.05375480306205728, 0.05370797189888107, 0.05366159955917616,
0.05361567973257964, 0.053570206220101946, 0.0535251729317274,
0.053480573884075476, 0.053436403198120964, 0.05339265509697199,
0.05334932390370364, 0.05330640403924588, 0.053263890020324334,
0.05322177645745112, 0.05318005805296686, 0.05313872959913009,
0.05309778597625351, 0.05305722215088608, 0.053017033174039434,
0.05297721417945706, 0.05293776038192557, 0.052898667075626224,
0.052859929632526836, 0.05282154350081164, 0.052783504203347945,
0.052745807336190685, 0.05270844856712031, 0.05267142363421757,
0.05263472834446945, 0.05259835857240905, 0.052562310258787014,

0.05252657940927329, 0.052491162093189624, 0.05245605444227057,
0.05242125264945347, 0.05238675296769584, 0.052352551708820416,
0.052318645242385046, 0.05228502999457995, 0.052251702447148156,
0.052218659136331426, 0.05218589665183953, 0.05215341163584229,
0.05212120078198398, 0.05208926083441992, 0.05205758858687476,
0.052026180881720015, 0.05199503460907373, 0.05196414670591838,
0.05193351415523891, 0.05190313398517903, 0.0518730032682159,
0.051843119120352665, 0.05181347870032819, 0.05178407920884462,
0.051754917887809594, 0.05172599201959665, 0.05169729892631993,
0.05166883596912504, 0.051640600547493816, 0.05161259009856581,
0.0515848020964709, 0.051557234051679016, 0.05152988351036075,
0.05150274805376318, 0.05147582529759764, 0.05144911289144003,
0.05142260851814418, 0.051396309893266115, 0.05137021476450149,
0.05134432091113274, 0.05131862614348946, 0.05129312830241812,
0.05126782525876255, 0.051242714912856244, 0.051217795194023016,
0.05119306406008919, 0.0511685194969044, 0.05114415951787242,
0.051119982163491264, 0.05109598550090217, 0.051072167623447254,
0.051048526650236854, 0.051025060725724096, 0.051001768019287956,
0.05097864672482557, 0.05095569506035073, 0.05093291126760129,
0.05091029361165426, 0.05088784038054717, 0.050865549884907754,
0.050843420457589966, 0.05082145045331768, 0.05079963824833452,
0.050777982240060145, 0.05075648084675414, 0.05073513250718477,
0.05071393568030486, 0.05069288884493359, 0.05067199049944383,
0.05065123916145618, 0.050630633367537314, 0.05061017167290519,
0.050589852651138884, 0.050569674893894065, 0.05054963701062369,
0.05052973762830389, 0.050509975391164454, 0.050490348960424127,
0.05047085701403198, 0.05045149824641146, 0.05043227136821082,
0.0504131751060568, 0.05039420820231331, 0.050375369414844676,
0.050356657516782186, 0.050338071296295375, 0.05031960955636802,
0.05030127111457627, 0.0502830548028728, 0.05026495946737292,
0.05024698396814593, 0.050229127179008896, 0.05021138798732512,
0.05019376529380478, 0.050176258012310404, 0.050158865069664944,
0.050141585405462964, 0.05012441797188615, 0.0501073617335202,
0.05009041566717703, 0.050073578761717544, 0.050056850017880086,
0.05004022844810937, 0.050023713076389774, 0.05000730293808124,
0.04999099707975757, 0.04997479455904724, 0.04995869444447782,
0.049942695815321875, 0.0499267977614465, 0.0499109993831643,
0.049895299791087665, 0.04987969810598496, 0.04986419345863905,
0.0498487849897092, 0.04983347184959324, 0.049818253198293964,
0.04980312820528664, 0.04978809604938858, 0.04977315591863158,
0.04975830701013602, 0.04974354852998658, 0.04972887969311051,
0.04971429972315785, 0.049699807852383454, 0.04968540332153042,
0.049671085379716234, 0.049656853284319846, 0.04964270630087201,
0.049628643702944786, 0.04961466477204576, 0.04960076879751157,
0.04958695507640464, 0.049573222913410535, 0.04955957162073736,
0.049546000518016976, 0.04953250893220712, 0.049519096197495685,
0.049505761655205816, 0.049492504653702994, 0.04947932454830351,
0.04946622070118429, 0.04945319248129358, 0.04944023926426396,

0.0494273604323257, 0.04941455537422239, 0.049401823485127005,
0.0493891641665599, 0.04937657682630744, 0.04936406087834235,
0.049351615742744936, 0.04933924084562604, 0.049326935619050064,
0.04931469950096043, 0.04930253193510515, 0.0492904323709642,
0.04927840026367731, 0.04926643507397327, 0.04925453626810062,
0.049242703317758194, 0.04923093570002803, 0.04921923289730819,
0.04920759439724734, 0.04919601969267985, 0.0491845082815619,
0.04917305966690887, 0.04916167335673247, 0.04915034886398062,
0.04913908570647716, 0.04912788340686156, 0.049116741492531424,
0.04910565949558432, 0.049094636952761075, 0.04908367340538949,
0.04907276839932973, 0.04906192148491909, 0.04905113221691896,
0.049040400154461546, 0.049029724860997714, 0.049019105904245734,
0.049008542856140425, 0.048998035292783286, 0.048987582794392776,
0.048977184945256484, 0.04896684133368239, 0.048956551551951784,
0.048946315196273314, 0.04893613186673596, 0.048926001167264456,
0.04891592270557442, 0.04890589609312789, 0.04889592094509042,
0.0488859968802876, 0.04887612352116313, 0.04886630049373717,
0.048856527427564626, 0.04884680395569514, 0.0488371297146327,
0.04882750434429613, 0.048817927487980174, 0.04880839879231724,
0.04879891790723906, 0.04878948448593959, 0.04878009818483772,
0.04877075866354125, 0.04876146558481071, 0.04875221861452384,
0.048743017421640666, 0.04873386167816897, 0.048724751059130156,
0.04871568524252569, 0.04870666390930378, 0.04869768674332672,
0.04868875343133842, 0.04867986366293272, 0.04867101713052186,
0.048662213529305026, 0.048653452557238436, 0.048644733915004124,
0.04863605730598089, 0.04862742243621411, 0.048618829014387445,
0.04861027675179292, 0.04860176536230361, 0.0485932945623456,
0.04858486407086911, 0.048576473609322865, 0.048568122901626036,
0.04855981167414153, 0.04855153965565104, 0.048543306577327465,
0.048535112172710136, 0.04852695617767949, 0.048518838330432114,
0.04851075837145588, 0.04850271604350588, 0.04849471109157996,
0.04848674326289545, 0.04847881230686547, 0.04847091797507577,
0.048463060021262, 0.04845523820128708, 0.048447452273118606,
0.04843970199680732, 0.04843198713446513, 0.048424307450243516,
0.04841666271031235, 0.04840905268283908, 0.0484014771379678,
0.04839393584779923, 0.048386428586369684, 0.04837895512963224,
0.048371515255435885, 0.04836410874350682, 0.04835673537542912,
0.04834939493462523, 0.04834208720633766, 0.048334811977610366,
0.04832756903727038, 0.048320358175909586, 0.04831317918586673,
0.04830603186121017, 0.04829891599772049, 0.048291831392872, 0.0482847778458173,
0.04827775515736944, 0.04827076312998572, 0.048263801567750696,
0.04825687027636069, 0.04824996906310692, 0.04824309773686016,
0.04823625610805493, 0.048229443988673626, 0.04822266119223204,
0.048215907533763015, 0.04820918282980242, 0.048202486898374086,
0.0481958195589748, 0.048189180632560565, 0.048182569941531576,
0.04817598730971862, 0.04816943256236882, 0.04816290552613173 4,
0.04815640602904612, 0.04814993390052607, 0.048143488971348014,
0.048137071073637044, 0.04813068004085424, 0.048124315707783805,

0.048117977910520235, 0.04811166648645545, 0.048105381274267105,
0.0480991221139052, 0.048092888846580854, 0.048086681314753695,
0.04808049936212041, 0.04807434283360261, 0.04806821157533519,
0.04806210543465489, 0.04805602426008926, 0.04804996790134512,
0.04804393620929716, 0.0480379290359773, 0.048031946234563916,
0.04802598765937065, 0.04802005316583617, 0.048014142610513344,
0.04800825585105917, 0.04800239274622405, 0.04799655315584219,
0.04799073694082076, 0.047984943963130716, 0.04797917408579642,
0.047973427172886204, 0.047967703089502364, 0.0479620017017219,
0.04795632287683805, 0.047950666482848216, 0.0479450323889478,
0.04793942046526931, 0.047933830582923845, 0.047928262613992266,
0.04792271643151608, 0.04791719190948881, 0.047911688922847355,
0.04790620734746372, 0.04790074706013603, 0.04789530793858035,
0.047889889861422695, 0.047884492708190604, 0.047879116359304905,
0.04787376069607184, 0.04786842560067523, 0.04786311095616854,
0.04785781664646669, 0.0478525425563392, 0.04784728857140189,
0.04784205457810969, 0.04783684046374905, 0.04783164611643059,
0.047826471425081805, 0.04782131627944, 0.04781618057004479,
0.04781106418823145, 0.047805967026123636, 0.04780088897662646,
0.04779582993341989, 0.047790789790951735, 0.04778576844443078,
0.04778076578982068, 0.04777578172383292, 0.04777081614392036,
0.04776586894827093, 0.04776094003580138, 0.04775602930615053,
0.04775113665967334, 0.04774626199743504, 0.047741405221204056,
0.047736566233446964, 0.04773174493732195, 0.04772694123667289,
0.04772215503602374, 0.04771738624057261, 0.047712634756185666,
0.04770790048939201, 0.047703183347377526, 0.04769848323797972,
0.04769380006968198, 0.04768913375160776, 0.047684484193515786,
0.04767985130579444, 0.047675234999456285, 0.04767063518613307,
0.047666051778070255, 0.047661484688121876, 0.04765693382974589,
0.04765239911699826, 0.04764788046452903, 0.04764337778757622,
0.047638891001962204, 0.04763442002408734, 0.04762996477092662,
0.047625525160023816, 0.04762110110948736, 0.047616692537985436,
0.047612299364741574, 0.04760792150952984, 0.04760355889266992,
0.04759921143502394, 0.0475948790579904, 0.04759056168350075,
0.04758625923401498, 0.04758197163251682, 0.04757769880250998,
0.04757344066801338, 0.04756919715355741, 0.04756496818417985,
0.047560753685421064, 0.04755655358332053, 0.04755236780441268,
0.047548196275722904, 0.04754403892476314, 0.04753989567952902,
0.047535766468494424, 0.04753165122060944, 0.047527549865294554,
0.047523462332438815, 0.0475193885523948, 0.04751532845597531,
0.0475112819744495, 0.047507249039540005, 0.047503229583418,
0.04749922353870061, 0.0474952308384475, 0.04749125141615672,
0.04748728520576135, 0.04748333214162624, 0.04747939215854523,
0.047475465191736156, 0.04747155117683893, 0.04746765004991191,
0.047463761747428244, 0.047459886206272776, 0.04745602336373917,
0.04745217315752608, 0.047448335525734506, 0.047444510406864586,
0.04744069773981214, 0.04743689746386596, 0.047433109518704705,
0.047429333844393566, 0.04742557038138181, 0.047421819070499326,

0.04741807985295382, 0.04741435267032801, 0.04741063746457679,
0.047406934178024114, 0.047403242753360185, 0.047399563133639015,
0.04739589526227516, 0.04739223908304117, 0.047388594540065075,
0.047384961577827275, 0.04738134014115803, 0.04737773017523496,
0.04737413162557998, 0.047370544438057, 0.04736696855886972 4,
0.04736340393455805, 0.047359850511996225, 0.047356308238390474,
0.04735277706127623, 0.04734925692851517 4, 0.04734574778829374,
0.047342249589119985, 0.04733876227982151, 0.04733528580954294,
0.04733182012774337, 0.04732836518419408, 0.04732492092897702,
0.04732148731248099, 0.04731806428540047, 0.047314651798733294,
0.047311249803777614, 0.04730785825213076, 0.047304477095686054,
0.04730110628663121, 0.04729774577744591, 0.04729439552089976,
0.047291055470050164, 0.04728772557824003, 0.04728440579909595,
0.04728109608652577, 0.04727779639471689, 0.04727450667813406,
0.04727122689151729, 0.04726795698987963, 0.04726469692850601,
0.047261446662950145, 0.04725820614903355, 0.04725497534284271,
0.04725175420072816, 0.047248542679301515, 0.047245340735434564,
0.04724214832625637, 0.04723896540915247, 0.047235791941762244,
0.04723262788197741, 0.047229473187940074, 0.04722632781804123,
0.04722319173091865, 0.04722006488545519, 0.04721694724077684,
0.04721383875625179, 0.04721073939148742, 0.04720764910632989,
0.04720456786086147, 0.047201495615399644, 0.04719843233049446,
0.04719537796692791, 0.047192332485711586, 0.04718929584808541,
0.047186268015515884, 0.04718324894969437, 0.04718023861253598,
0.04717723696617733, 0.047174243972975316, 0.0471712595955059,
0.04716828379656183, 0.04716531653915174, 0.04716235778649847,
0.04715940750203739, 0.04715646564941494, 0.0471535321924877,
0.04715060709531991, 0.047147690322183085, 0.0471447818375538,
0.047141881606112705, 0.04713898959274279, 0.047136105762528295,
0.04713323008075333, 0.04713036251290004, 0.04712750302464755,
0.0471246515818707, 0.04712180815063869, 0.04711897269721346,
0.04711614518804878, 0.04711332558978827, 0.047110513869265035,
0.04710770999349934, 0.047104913929698344, 0.047102125645254095,
0.04709934510774246, 0.047096572284922006, 0.047093807144732464,
0.047091049655294225, 0.04708829978490596, 0.047085557502044385,
0.04708282277536243, 0.047080095573688745, 0.04707737586602581,
0.04707466362154891, 0.04707195880960536, 0.047069261399712986,
0.04706657136155922, 0.047063888664999345, 0.04706121328005635,
0.04705854517691894, 0.047055884325941065, 0.047053230697640266,
0.04705058426269706, 0.0470479449919534, 0.047045312856412044,
0.04704268782723513, 0.04704006987574326, 0.04703745897341448,
0.047034855091883127, 0.047032258202939096, 0.0470296682785263,
0.04702708529074173, 0.04702450921183534, 0.04702194001420771,
0.04701937767040981, 0.047016822153142, 0.047014273435252815,
0.04701173148973816, 0.0470091962897401, 0.047006667808546225,
0.0470041460195887, 0.047001630896442693, 0.04699912241282668,
0.04699662054259985, 0.04699412525976251, 0.046991636538454826,
0.04698915435295583, 0.04698667867768217, 0.046984209487187914,

0.046981746756163005, 0.04697929045943281, 0.046976840571957175,
0.046974397068829256, 0.046971959925275156, 0.04696952911665263,
0.046967104618450335, 0.046964686406287204, 0.04696227445591145,
0.046959868743199476, 0.046957469244155614, 0.046955075934910856,
0.046952688791722354, 0.04695030779097223, 0.04694793290916693,
0.04694556412293692, 0.04694320140903502, 0.046940844744336206,
0.04693849410583677, 0.04693614947065345, 0.04693381081602269,
0.046931478119299756, 0.04692915135795841, 0.046926830509589325,
0.04692451555190055, 0.046922206462715464, 0.04691990321997305,
0.046917605801726735, 0.04691531418614358, 0.04691302835150375,
0.04691074827619996, 0.04690847393873633, 0.04690620531772797,
0.04690394239190035, 0.04690168514008825, 0.04689943354123567,
0.04689718757439448, 0.04689494721872429, 0.04689271245349128,
0.04689048325806819, 0.04688825961193269, 0.04688604149466809,
0.046883828885961105, 0.046881621765602705, 0.046879420113486246,
0.046877223909607586, 0.04687503313406411, 0.04687284776705461,
0.04687066778887752, 0.04686849317993187, 0.046866323920715124,
0.04686415999182388, 0.04686200137395233, 0.04685984804789194,
0.04685769999453123, 0.04685555719485459, 0.046853419629941974,
0.04685128728096846, 0.0468491601292034, 0.04684703815600997,
0.04684492134284468, 0.046842809671256334, 0.04684070312288652,
0.04683860167946755, 0.0468365053228234, 0.046834414034868016,
0.0468323277976054, 0.04683024659312909, 0.04682817040362087,
0.04682609921135139, 0.04682403299867873, 0.04682197174804807,
0.04681991544199136, 0.04681786406312656, 0.04681581759415776,
0.04681377601787329, 0.04681173931714679, 0.04680970747493587,
0.04680768047428114, 0.04680565829830724, 0.04680364093022057,
0.046801628353309914, 0.046799620550945775, 0.046797617506579486,
0.04679561920374334, 0.046793625626049364, 0.04679163675718961,
0.046789652580935114, 0.04678767308113559, 0.046785698241719194,
0.0467837280466916, 0.04678176248013599, 0.046779801526212535,
0.04677784516915726, 0.04677589339328293, 0.04677394618297714,
0.04677200352270296, 0.04677006539699771, 0.046768131790473066,
0.04676620268781468, 0.0467642780737813, 0.046762357933204206,
0.04676044225098769, 0.04675853101210773, 0.046756624201612,
0.04675472180461925, 0.04675282380631912, 0.046750930191971594,
0.04674904094690632, 0.04674715605652288, 0.0467452755062896,
0.04674339928174367, 0.046741527368490694, 0.04673965975220417,
0.04673779641862484, 0.04673593735356097, 0.04673408254288722,
0.04673223197254475, 0.04673038562854079, 0.046728543496947925,
0.046726705563904083, 0.046724871815612044, 0.046723042238338876,
0.04672121681841609, 0.04671939554223835, 0.046717578396264246,
0.04671576536701483, 0.04671395644107407, 0.04671215160508815,
0.046710350845765185, 0.0467085541498746, 0.04670676150424739,
0.04670497289577498, 0.046703188311409634, 0.04670140773816356,
0.04669963116310875, 0.0466978585733768, 0.04669608995615812,
0.04669432529870252, 0.04669256458831754, 0.0466908078123692,
0.04668905495828128, 0.04668730601353507, 0.04668556096566887,

```
0.046683819802277655, 0.04668208251101328, 0.04668034907958338,
0.046678619495751715, 0.04667689374733725, 0.04667517182221457,
0.04667345370831277, 0.046671739393615845, 0.0466700288661619,
0.04666832211404294, 0.04666661912540505, 0.046664919888447044,
0.046663224391421405, 0.04666153262263312, 0.046659844570439715,
0.04665816022325078, 0.046656479569528016, 0.04665480259778448,
0.04665312929658488, 0.046651459654544425, 0.04664979366032965,
0.04664813130265699, 0.046646472570293585, 0.04664481745205612,
0.04664316593681111, 0.046641518013474055, 0.04663987367101006,
0.04663823289843259, 0.04663659568480377, 0.04663496201923405,
0.04663333189088172, 0.046631705288953086, 0.04663008220270135,
0.046628462621427655, 0.04662684653447947, 0.04662523393125128,
0.0466236248011838, 0.04662201913376397, 0.04662041691852468,
0.046618818145044455, 0.04661722280294707, 0.04661563088190174,
0.04661404237162246, 0.046612457261867704, 0.04661087554244065,
0.04660929720318851, 0.04660772223400233, 0.04660615062481716,
0.04660458236561108]
```

[22]:
```
w
```

[22]:
```
array([ 0.03870637, -0.06542164, -0.19583812, -0.37865278,  0.40473166,
       -0.05864629, -0.02669884, -0.54653277, -0.27165641,  0.05987412,
        0.48379414, -0.59092967,  0.58835679])
```

[23]:
```
X_test_scaled=normalize_matrix(X_test)

y_testpred=np.dot(X_test_scaled,w)
y_testpred=sigmoid(y_testpred)
print(X_test_scaled.shape)
print(w.shape)
y_testpred.shape
```

```
Original Matrix:
[[1.000000e+00 7.427900e+04 1.078300e+03 … 8.028000e-01 1.984700e+00
  7.087000e-01]
 [1.000000e+00 9.441100e+04 1.257530e+03 … 7.502000e-01 2.289100e+00
  6.600000e-01]
 [1.000000e+00 6.743700e+04 1.014144e+03 … 8.240000e-01 1.895600e+00
  7.235000e-01]
 …
 [1.000000e+00 7.802500e+04 1.198898e+03 … 6.821000e-01 1.950500e+00
  7.051000e-01]
 [1.000000e+00 7.607300e+04 1.064233e+03 … 8.440000e-01 1.911700e+00
  7.225000e-01]
 [1.000000e+00 7.309200e+04 1.126401e+03 … 7.239000e-01 2.506400e+00
  6.302000e-01]]

Normalized Matrix:
```

```
[[ 1.         -0.51628977 -0.52878068 …  0.20422438 -0.17122072
   0.07742287]
 [ 1.          0.97880743  1.15238809 … -0.71877211  0.80009411
  -0.84400762]
 [ 1.         -1.02440893 -1.13056087 …  0.57623057 -0.45553134
   0.35744692]
 …
 [ 1.         -0.23809415  0.60242275 … -1.91375427 -0.28035005
   0.00930891]
 [ 1.         -0.38305887 -0.66072846 …  0.92717981 -0.40415759
   0.33852637]
 [ 1.         -0.60444198 -0.07759561 … -1.18027036  1.49348016
  -1.40783983]]
(500, 13)
(13,)
```

[23]: (500,)

[24]:
```
for i in range(y_testpred.shape[0]):
    if y_testpred[i]>=0.5:
        y_testpred[i]=1
    else:
        y_testpred[i]=0
y_testpred.shape
```

[24]: (500,)

[25]:
```
y_test.shape
```

[25]: (500,)

[26]:
```
# precision
def print_precision_recall(y_test,y_testpred):
    true_positives=0
    true_negatives=0
    false_positives=0
    false_negatives=0

    for i in range(y_test.shape[0]):
        if y_test[i]==1 and y_testpred[i]==1:
            true_positives+=1
        elif y_test[i]==1 and y_testpred[i]==0:
            false_negatives+=1
        elif y_test[i]==0 and y_testpred[i]==0:
            true_negatives+=1
        elif y_test[i]==0 and y_testpred[i]==1:
            false_positives+=1
```

```python
    precision = true_positives/(true_positives+false_positives)
    recall= true_positives/(true_positives+false_negatives)
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    return true_positives,true_negatives,false_positives,false_negatives
```

[27]:
```python
tp,tn,fp,fn=print_precision_recall(y_test,y_testpred)
```

```
Precision: 0.8712121212121212
Recall: 0.8679245283018868
```

[28]:
```python
print(f"mean accuracy: {(tp+tn)/y_test.shape[0]}")
```

```
mean accuracy: 0.862
```