Assignment 2 for Capstone Project.

Dear Ladies and Gentlemen!
Good Luck!

1) **(35 points)**

Suppose that you're working in NU admssion, namely in the Department of Digitalization. You have the results of 1000 applicants (with their GPA score and SAT score) and output variable as 0 or 1 (admitted or not admitted). [Weblink to Dataset: https://drive.google.com/file/d/1klV2KsfzzMpcC-D3aikddZzwds0KVOkD/view?usp=sharing]

Output variable was implemented by using some traditional methods. GPA score can be between 0 – 4, while SAT score between 0 – 1600. You have a target to automate students' filtration for the next stage and at the same time you would like to check the efficiency of addmission comittee. You are going to use Logistic Regression algorithm to do above task. Now let's start everything step-by-step:

1 a) Firstly, it is necessary to do some feature normalization, because the range of GPA and SAT scores are different. You need to use the following Z-normalization:

$$Z = \frac{X - \mu}{\sigma}$$

where μ – average mean; σ – std. deviation.

If you did everything correctly, what is the sum of first row+ last row of GPA score in your normalized dataset? ???

What is the sum of first row+ last row of SAT score in your normalized dataset? …

Hint: here you need to drag-and-drop digit-by-digit. Moreover, if your anwer is negative one, then you should first drag-and-drop negative sign. For example: if your answer is -0.15 then firstlly you should drag-and-drop -, then 0, then 1, then 5. Please, round up to 3 digits after floating point.

1 b) So, as you understood, you are going to use the following hypothesis:

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Let's initialize theta parameters with zeros. Now, tell me please cost function value: ???

Now, you are going to use minimumcost function and tell me what your new cost function is: ???

1 c) What are your new theta parameters after calling minimum cost function ???

1 d) What is the number of admitted stundets (1s) in predicted list (with theta parameters from part 1 c)- ??

What is the number of not admitted stundets (0s) (with theta parameters from part 1 c) - ??

1 e) Compute reg.score ??? Please round up to 3 digits

Compute admission probability: for 0.35 GPA Score and 0.15 of SAT Score in normalized ones. ??? Please round up to 3 digits

1 f) Now it's time to draw a decision boundary. For that please include the following piece of code:

plot_x = np.array([normalized[1].min(),normalized[1].max()])

plot_y = -(theta[0]+theta[1]*plot_x)/theta[2]

Please find correct picture of decision boundary for 0.35 GPA score and 0.15 SAT score.

1 g) Please find real values of 0.35 GPA Score and 0.15 SAT score.

That's the end of 1st task.

2) **(39 points)**

Now we will start with a new dataset. [Weblink to dataset: https://drive.google.com/file/d/1ouAsiNuW6qNwf97rai_YlSyePn8Wnnky/view?usp=sharing]

It contains 2 input features, 1 output variable (0 or 1). In this case we will try to apply Logistic Regression with regularization.

2 a) Let's apply polynomial feature with 5 degrees:

log_reg = logistic_regression_reg()

X = data[[0,1]]

y = data[2].values

X1 = log_reg.mapfeature(X,5)

What is the shape of X1 ??rows and ?? columns

Let's initialize theta parameters with zero. What is the cost function ??

2 b) In this step it is necessary to implement mincost optimization. What is the maximum value of optimal theta: ??

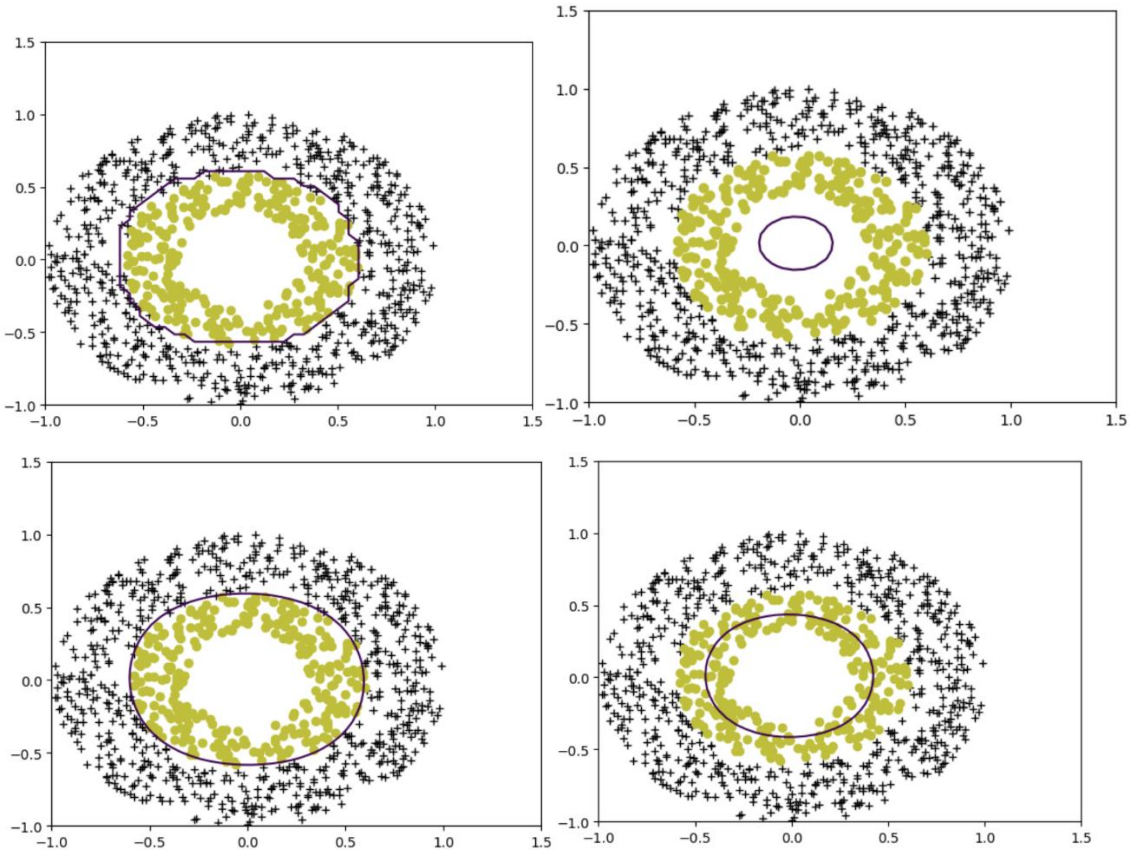
What is the minimum value of optimal theta: ??

Please round your answers up to 2 digits after floating point

Compute reg_score value ??

2 c) Let's plot data with 3 different lambda values: 0, 1, 25

Now, please match the following figure with correct lambda values.



2 d) In this step, you need to define true or false statements after your detailed analysis.

The case with lambda = 0 related to underfitting problem. ??

Accuracy of model with lambda = 0 is 0.98 (rounded up to 2 digits after floating point) ??

The case with lambda = 1 related to OK case. ??

Accuracy of model with lambda = 1 is 0.991 (rounded up to 3 digits after floating point) ??

The case with lambda = 25 related to overfitting problem. ??

Accuracy of model with lambda = 25 is 0.712 (rounded up to 2 digits after floating point) ??

That's the end of 2nd task.

3) **(26 points)**

Suppose that you're going to build NN model for detecting English alphabets.[Weblink to dataset:

https://drive.google.com/file/d/1VnS1430Jo-AFdXXbjJpn29Xzn2SeWwlx/view?usp=sharing]

You have different piece of codes. Your main task is to collect all codes into ones by drag-and-dropping.

3a) import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

import tensorflow as tf

from keras.optimizers import Adam, SGD

import warnings

import sys

if not sys.warnoptions:

warnings.simplefilter("ignore")

3 b) data = pd.read_csv("C:/Users/Ruslan/Downloads/archive/A_Z Handwritten Data.csv")

3 c) data.head()

3) d) pixel_data = data.iloc[0].drop('0').values

image_array = pixel_data.reshape(28, 28)

plt.figure(figsize=(5,5))

plt.imshow(image_array, cmap='cool')

plt.show()

3 e) inputs = data.drop('0' ,axis =1)

targets = data['0']

3 f) seed = 333

(train_inputs, train_validate_inputs, train_targets, train_validate_targets) = train_test_split(inputs, targets, test_size=0.70, random_state=seed)

(test_inputs, validation_inputs, test_targets, validation_targets) = train_test_split(train_validate_inputs, train_validate_targets,

test_size=0.70, random_state=seed)

3 g)

input_size = inputs.shape[1]

output_size = 26

layer_1 = 150

layer_2 = 100

layer_3 = 50

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(layer_1, activation='relu'),
    tf.keras.layers.Dense(layer_2, activation='relu'),
    tf.keras.layers.Dense(layer_3, activation='relu'),
    tf.keras.layers.Dense(output_size, activation='softmax')
])
model.compile(optimizer=Adam(learning_rate=0.001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

3 h) batch_size = 100

max_epochs = 40

3 j) early_stopping = tf.keras.callbacks.EarlyStopping(patience=2)

3 k) modelmetrics = model.fit(train_inputs,

train_targets,

batch_size=batch_size,

epochs=max_epochs,

callbacks=[early_stopping],

validation_data=(validation_inputs, validation_targets),

verbose = 2

)

3 l) test_loss, test_accuracy = model.evaluate(test_inputs, test_targets)

print('The model accuracy is ' + f"{test_accuracy*100:.1f}" + '%')

3 m) model.summary()

Now, if you successfully collect all pieces of code into ones, and if you don't have any problem with running then let me check some technical information from you. Let's start:

3 m) We have ???rows and ??? columns. Hint: you need to drag-and-drop digit-by-digit

What English letter is on the 150,000th line of the dataset? ???

3 n) We divided our dataset into 75% of training set and 25% of test set. ??? (true, false, not given)

How many hidden layers do we have? ???

We are using 'relu' activation function in hidden layers ??? (true, false, not given)

3 o) In the code we are using maximum 30 epochs initially. ??? (true, false, not given)

How many epochs of non-decreasing validation loss are tolerated in our early stopping setup? ???

We are using Adaboost optimizer. ??? (true, false, not given)

3 p) What was the final model accuracy? ???

Total number of parameters: ???

Total number of trainable parameters: ???

Total number of optimizer parameters: ???

That's the end of 3rd task.

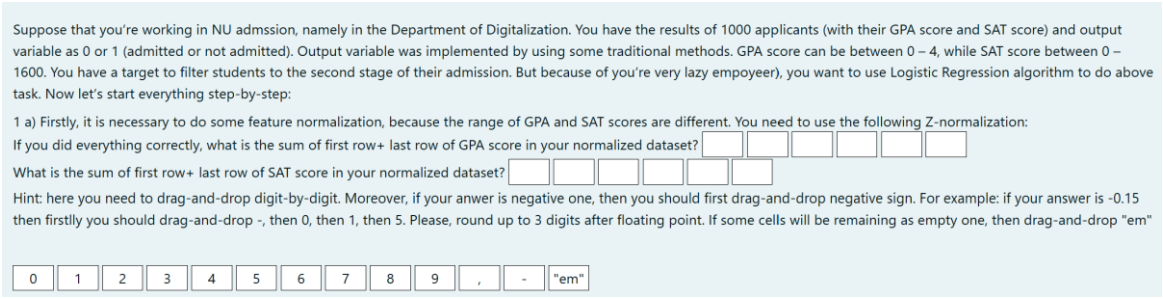
Useful References:

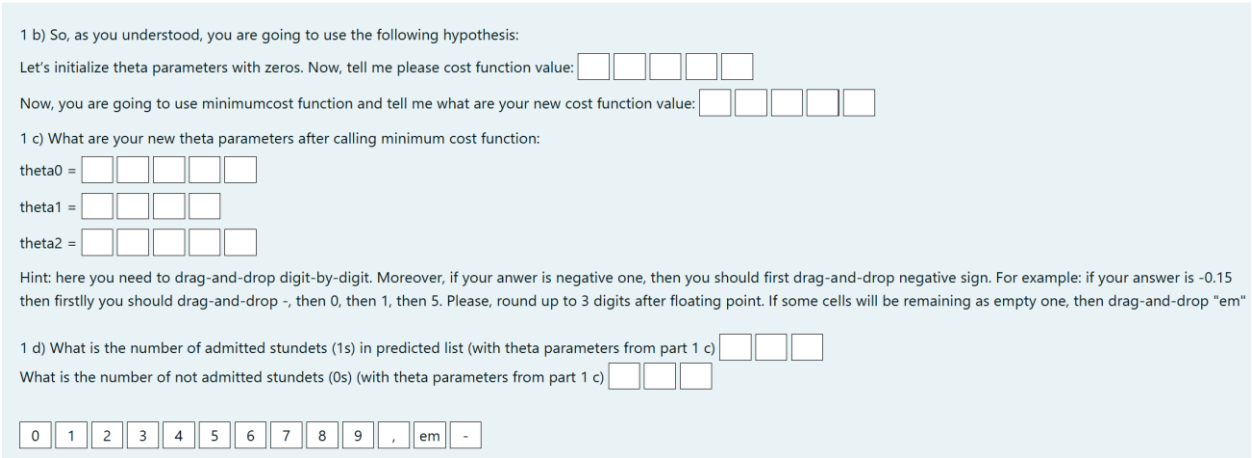
[1] https://github.com/akchaudhary57/Andrew-Ng-Coursera-Machine-learning-in-Python/blob/main/Logistic%20Regression/LogisticRegression.ipynb

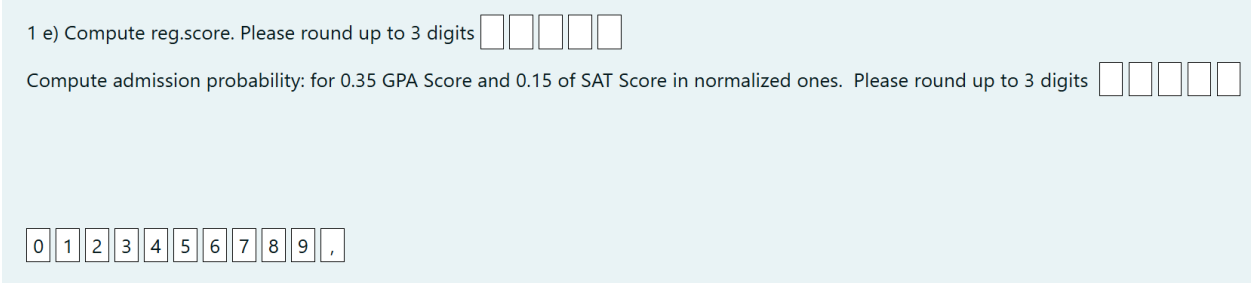
[2] https://github.com/akchaudhary57/Andrew-Ng-Coursera-Machine-learning-in-Python/blob/main/Logistic%20Regression/Logistic%20Regression%20with%20Regularization.ipynb

[3] https://github.com/akchaudhary57/Andrew-Ng-Coursera-Machine-learning-in-Python/tree/main/Neural%20Network

Screenshoots:







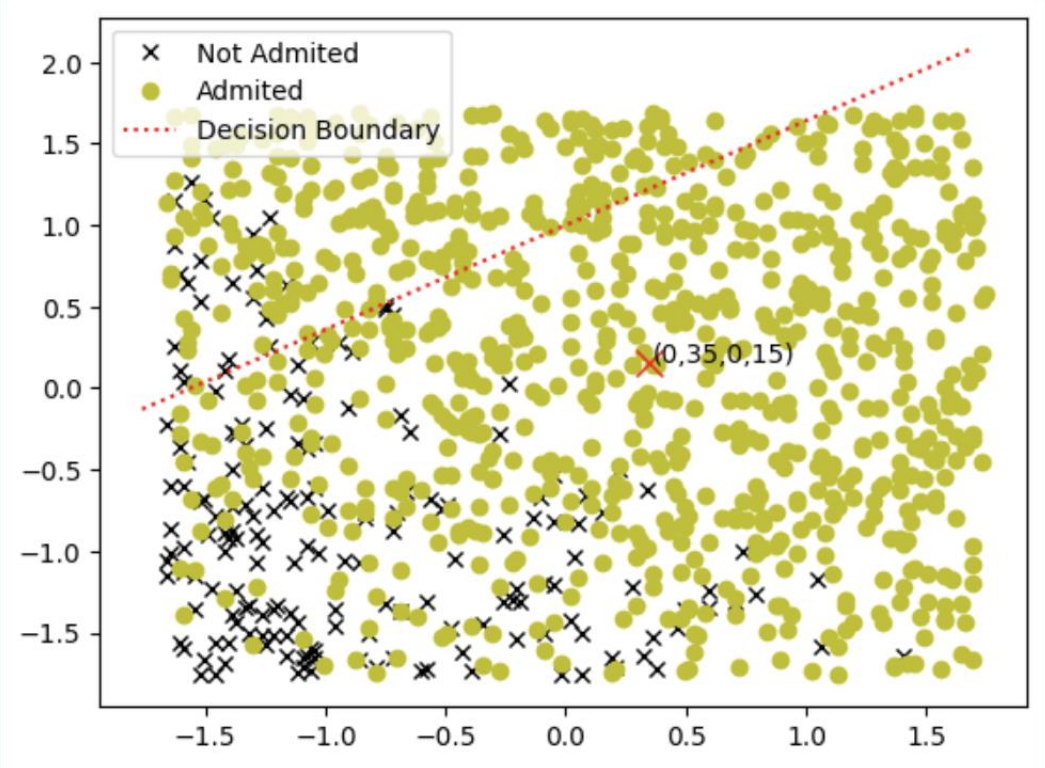
1 f) Now it's time to draw a decision boundary. For that please include the following piece of code:

plot_x = np.array([normalized[1].min(),normalized[1].max()])

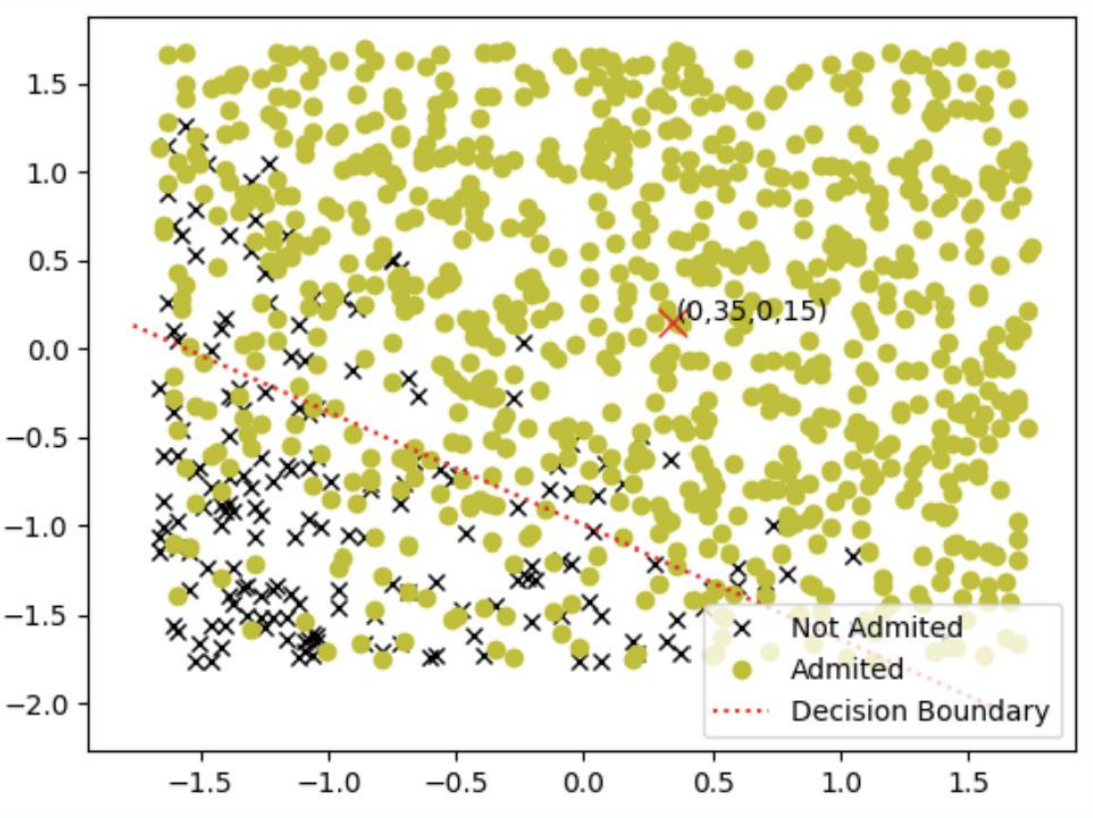
plot_y = -(theta[0]+theta[1]*plot_x)/theta[2]

Please find correct picture of decision boundary for 0,35 GPA score and 0,15 SAT score.

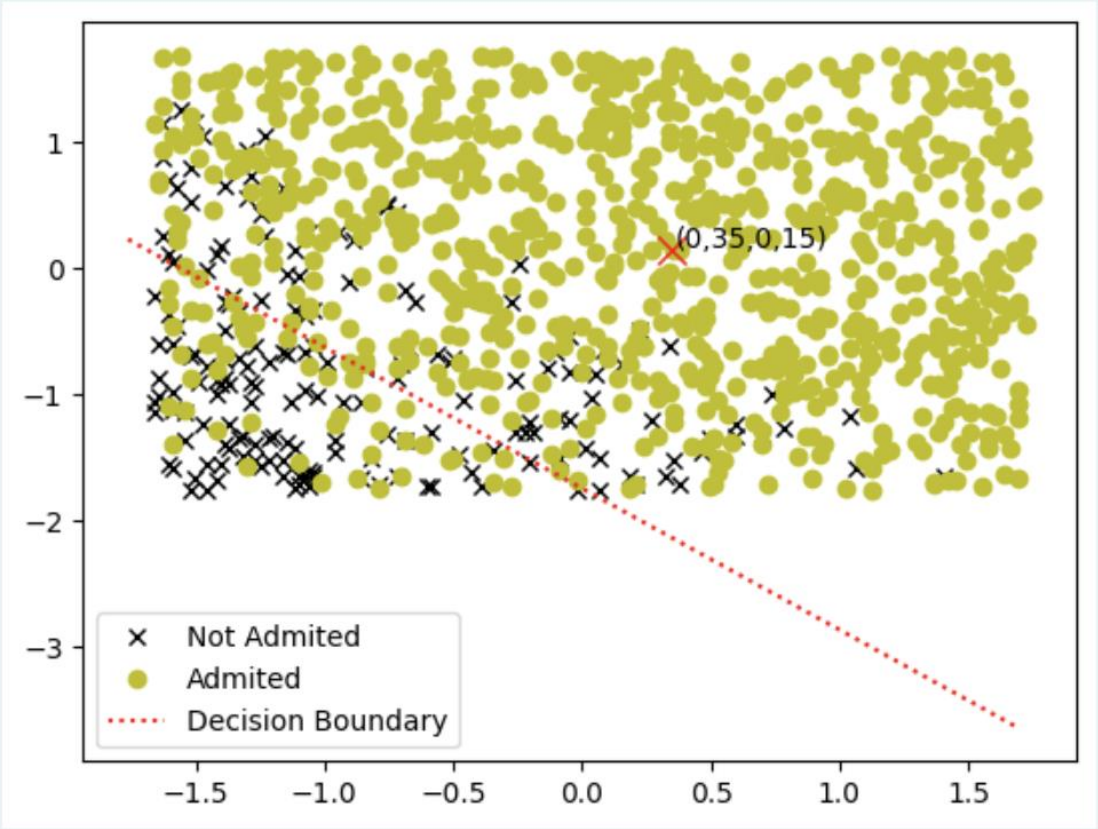
○ a.



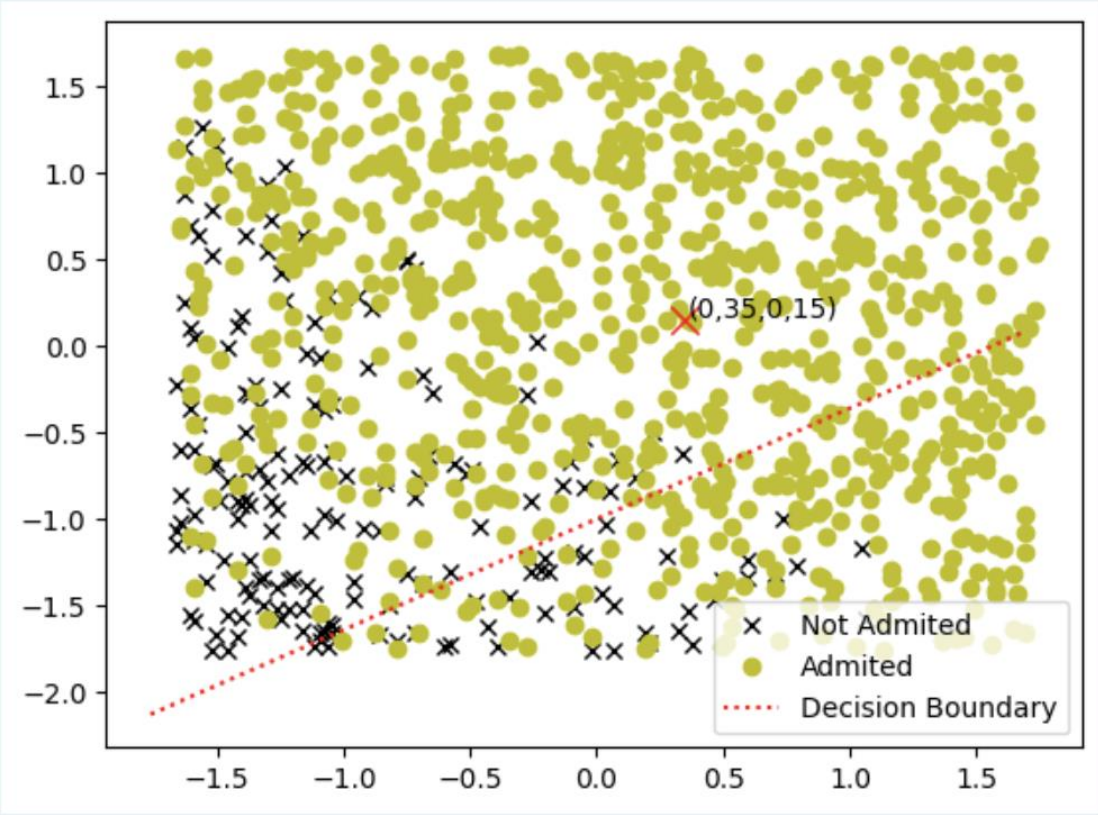
○ b.



○ c.


(0,35,0,15)
Not Admited
Admited
Decision Boundary

○ d.


(0,35,0,15)
Not Admited
Admited
Decision Boundary

1 g) Please find real values of 0.35 GPA Score and 0.15 SAT score. Round up to 2 digits after floating point

- a. 3.18 1064.38
- b. 3.33 845.26
- c. 2.58 812.36
- d. 2.66 1005.31
- e. 1.98 556.29
- f. 3.56 1500.14

Now we will start with a new dataset. It contains 2 input features, 1 output variable (0 or 1). In this case we will try to apply Logistic Regression with regularization.

2 a) Let's apply polynomial feature with 5 degrees:

```
log_reg = logistic_regression_reg()
X = data[0,1]
y = data[2].values
X1 = log_reg.mapfeature(X,5)
```

What is the shape of X1?? ☐☐☐☐ rows and ☐☐ columns

Let's initialize theta parameters with zero. What is the cost function? ☐☐☐☐

Round up to 2 digits after floating point

0 1 2 3 4 5 6 7 8 9 ,

2 b) In this step it is necessary to implement mincost optimization. What is the maximum value of optimal theta: ☐☐☐☐

What is the minimum value of optimal theta: ☐☐☐☐☐. Please round your answers up to 2 digits after floating point

Compute reg_score value? ☐☐☐☐☐ Please round your answers up to 3 digits after floating point

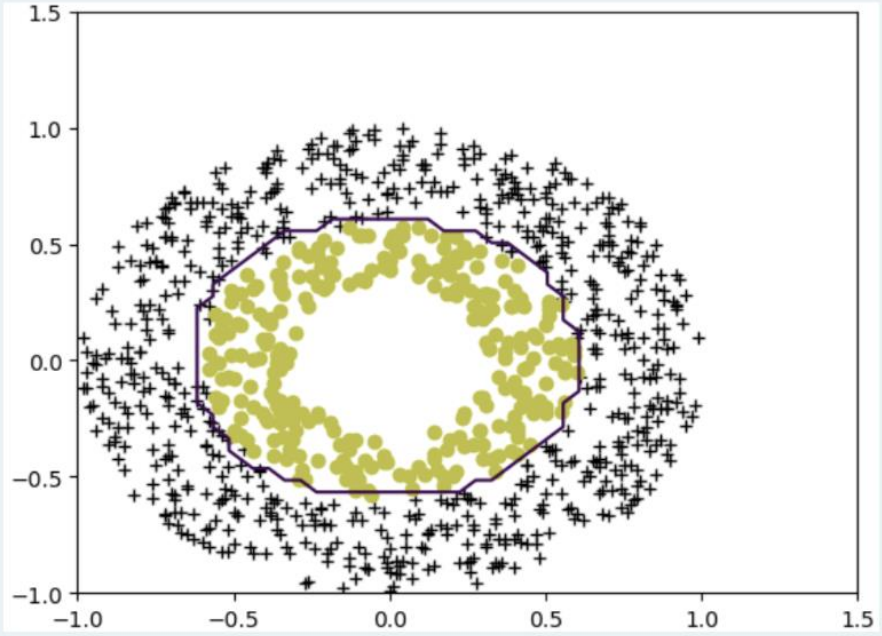
Hint: here you need to drag-and-drop digit-by-digit. Moreover, if your anwer is negative one, then you should first drag-and-drop negative sign. For example: if your answer is -0.15 then firstlly you should drag-and-drop -, then 0, then 1, then 5. Please, round up to 3 digits after floating point.

0 1 2 3 4 5 6 7 8 9 , -

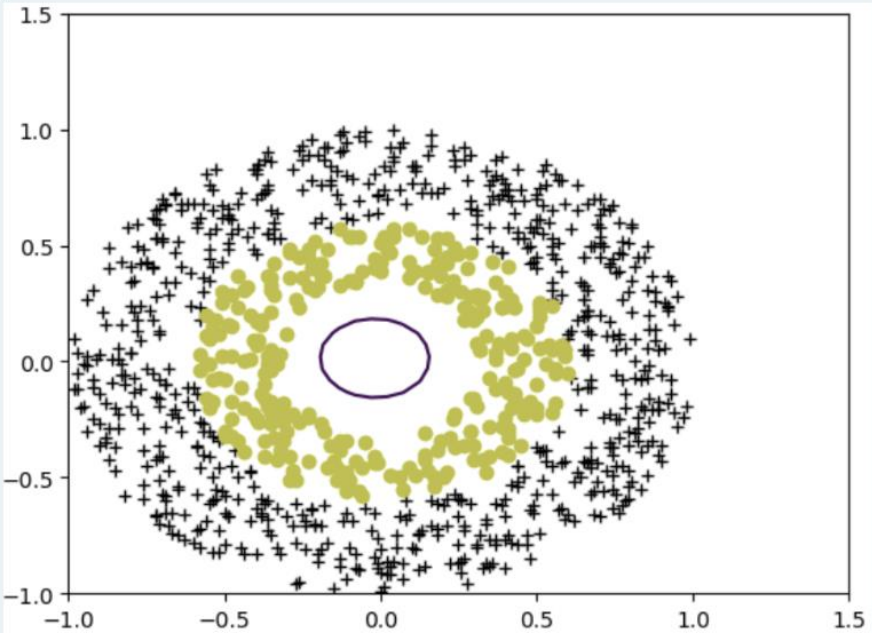
2 c) Let's plot data with 3 different lambda values: 0, 1, 25

Now, please match the following figure with correct lambda values.

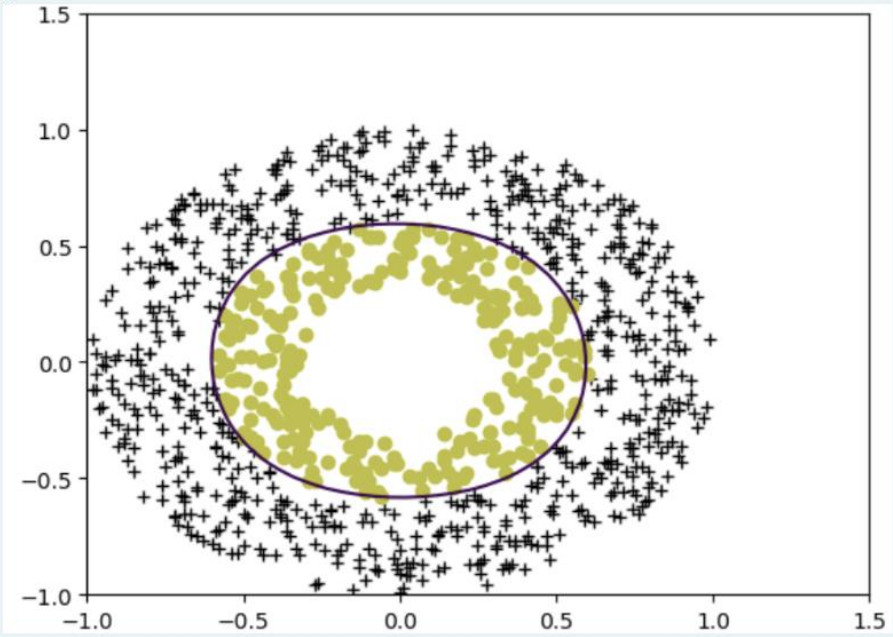
A:



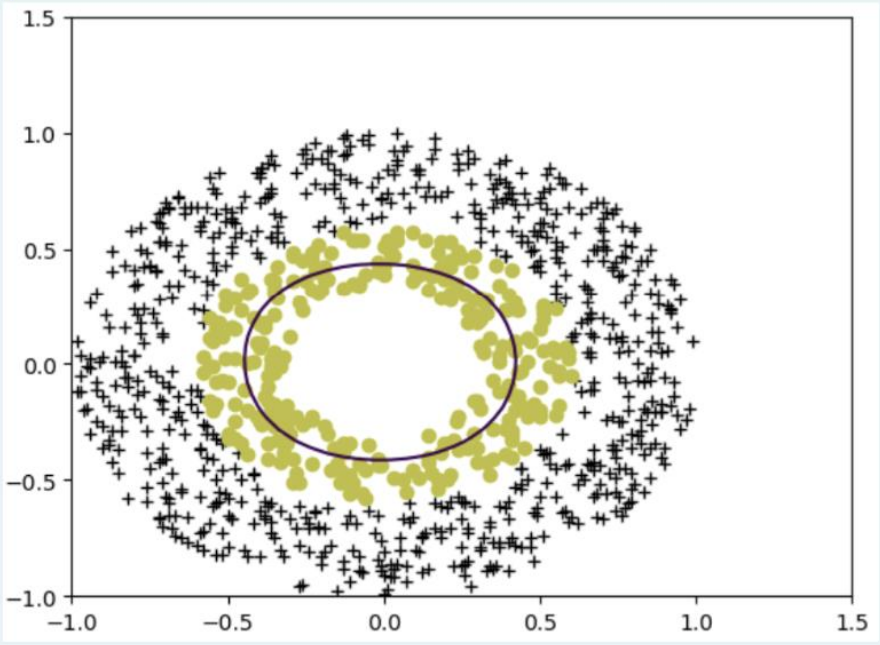
B:



C:



D:



Lambda = 0 Choose...

lambda = 25 Choose...

lambda = 1 Choose...

2 d) In this step, you need to define true or false statements after your detailed analysis.

The case with lambda = 0 related to underfitting problem. [ ]

Accuracy of model with lambda = 0 is 0.98 (rounded up to 2 digits after floating point) [ ]

The case with lambda = 1 related to OK case. [ ]

Accuracy of model with lambda = 1 is 0.991 (rounded up to 3 digits after floating point) [ ]

The case with lambda = 25 related to overfitting problem. [ ]

Accuracy of model with lambda = 25 is 0.712 (rounded up to 3 digits after floating point) [ ]

True | False

3) Suppose that you're going to build NN model for detecting English alphabets. You have different piece of codes. Your main task is to collect all codes into ones by drag-and-dropping. And also please answer the following question

3 a)
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import tensorflow as tf
from keras.optimizers import Adam, SGD
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
```

3 b)
```
data = pd.read_csv("C:/Users/Ruslan/Downloads/archive/A_Z Handwritten Data.csv")
```

3 c)
```
data.head()
```

3) d)
```
pixel_data =  data.iloc[0].drop('0').values
      image_array = pixel_data.reshape(28, 28)
      plt.figure(figsize=(5,5))
      plt.imshow(image_array, cmap='cool')
      plt.show()
```

3 e)
```
inputs = data.drop('0' ,axis =1)
targets = data['0']
```

3 f)
```
seed = 333
(train_inputs, train_validate_inputs, train_targets, train_validate_targets) = train_test_split(inputs, targets, test_size=0.70, random_state=seed)
(test_inputs, validation_inputs, test_targets, validation_targets) = train_test_split(train_validate_inputs, train_validate_targets,
                                                                                      test_size=0.70, random_state=seed)
```

```
3 g)
input_size = inputs.shape[1]
output_size = 26
layer_1 = 150
layer_2 = 100
layer_3 = 50
model = tf.keras.Sequential([
    tf.keras.layers.Dense(layer_1, activation='relu'),
    tf.keras.layers.Dense(layer_2, activation='relu'),
    tf.keras.layers.Dense(layer_3, activation='relu'),
    tf.keras.layers.Dense(output_size, activation='softmax')
])
model.compile(optimizer=Adam(learning_rate=0.001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

3 h) batch_size = 100
     max_epochs = 40

3 j) early_stopping = tf.keras.callbacks.EarlyStopping(patience=2)

3 k) modelmetrics = model.fit(train_inputs,
          train_targets,
          batch_size=batch_size,
          epochs=max_epochs,
          callbacks=[early_stopping],
          validation_data=(validation_inputs, validation_targets),
          verbose = 2
          )

3 l) test_loss, test_accuracy = model.evaluate(test_inputs, test_targets)
print('The model accuracy is ' + f"{test_accuracy*100:.1f}" + '%')

3 m) model.summary()
```

Number of samples which is propagated through the network	Identification of input and output features
Rebuilding of a dataset	Importing necessary libraries
Display first 5 rows of dataset	Display first image from dataset
Table of parameters related to model	Assess Model Accuracy
Load data	Data preprocessing step
Training and learning of a model	Convergence-Based Termination
Model Building	Dividing dataset into 2 different sets

Now, if you successfully collect all pieces of code into ones, and if you don't have any problem with running then let me check some technical information from you. Let's start:

3 m) We have ☐☐☐☐☐☐ rows and ☐☐☐ columns. Hint: you need to drag-and-drop digit-by-digit

What English letter is on the 150,000th line of the dataset? ☐

3 n) We divided our dataset into 75% of training set and 25% of test set. ☐

How many hidden layers do we have? ☐

We are using 'relu' activation function in hidden layers ☐

3 o) In the code we are using maximum 30 epochs initially. ☐

How many epochs of non-decreasing validation loss are tolerated in our early stopping setup? ☐

We are using Adaboost optimizer. ☐

3 p) What was the final model accuracy? ☐

Total number of parameters: ☐☐☐☐☐☐

Total number of trainable parameters: ☐☐☐☐☐☐

Total number of optimizer parameters: ☐☐☐☐☐☐

That's the end of 3rd task

1 2 3 4 5 6 7 8 9 0

A B O P N R

True False Not Given

~93% ~85% ~80% ~99% ~75%

If you successfully finished all 3 tasks, now you can transfer your anwers on Moodle. But again, please be sure that you're starting when it is convenient for you. Because you have only one attempt. Assignment2 Password: 172410482

That's all about 2nd Assignment. I hope that you really enjoyed during solving these interesting problems and derived some useful information for yourself. Thank you for your accurate reading and for your attention until the end. If you have any questions, please do not hesitate to contact me via MS Teams.

Kind Regards,
Ruslan Omirgaliyev
Senior-Lecturer of Department of Computer Engineering
Astana IT University