BIRZEIT UNIVERSITY

MACHINE LEARNING AND DATA SCIENCE

ENCS5341.

ASSIGNMENT#2.

PREPARED BY:

SALEH KHATIB – 1200991.

SECTION 1.

DR: YAZAN ABU FARHA.

DATE: 22/12/2023.

Table of Contents

Model Selection and Hyper-parameters Tunning

1-)

First of all, we must check our data set, so we print the data set information.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x1      200 non-null    float64
 1   x2      200 non-null    float64
 2   y       200 non-null    float64
dtypes: float64(3)
memory usage: 4.8 KB
None
```

Fig1: data set information.

As we see, there are 2 features, one label, and there is no missing values, or errors.

Now we will split this data set into 3 sets: train, validation, and test set, by:

data_set.iloc[:120], which will take 120 rows for the train set, and we do this for validation and test sets.

Now, we will plot these sets in a 3d scatter as shown below:

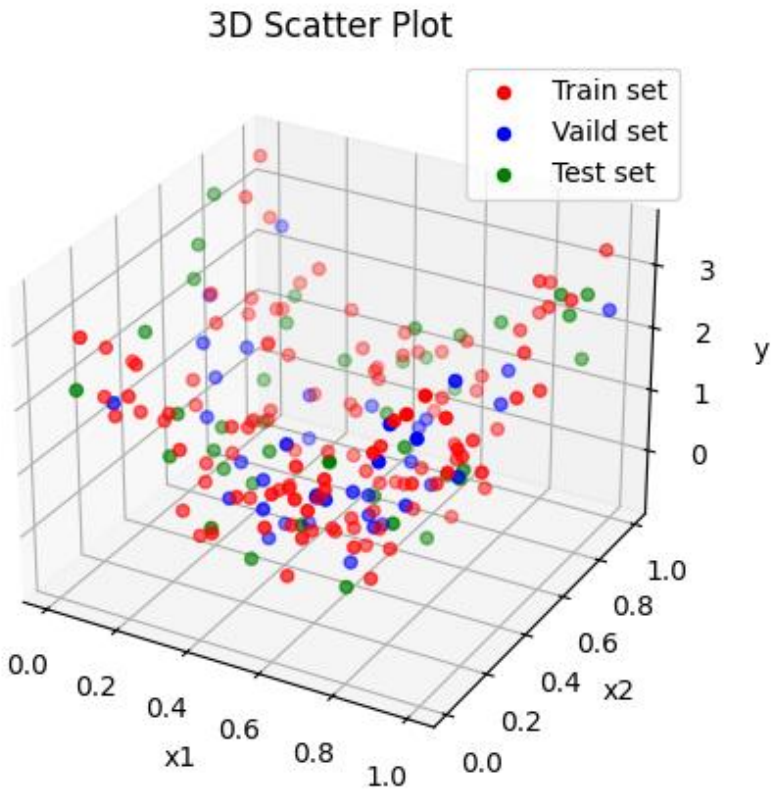

Fig2: 3d Scatter Plot.

As we note, the VS and TS are spread will over all the data set.

2-)

Now we will apply polynomial regression on the training set with different degrees from 1 to 10, and check which is the best degree.

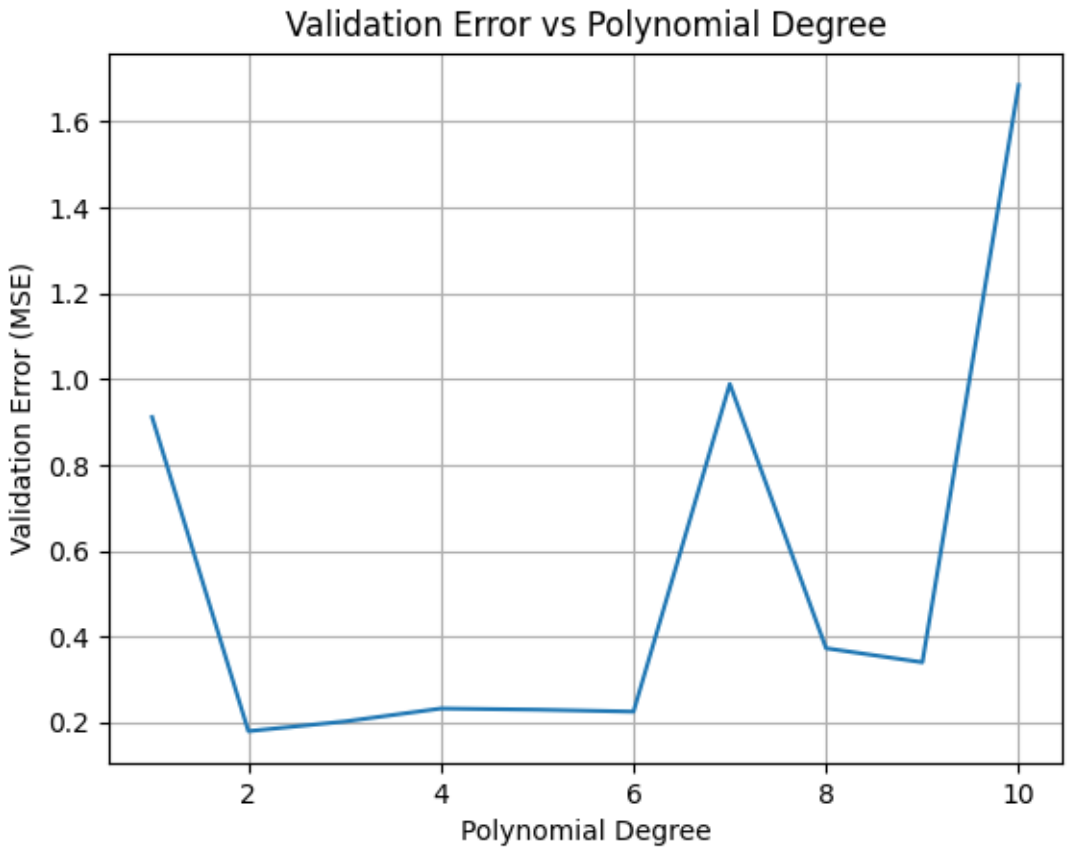

Fig3: Validation Error vs Polynomial Degree.

As we see, the least Validation Error 0.18 at degree 2, which is the best degree.

Now we will see t the surface of the learned function alongside with the training examples.

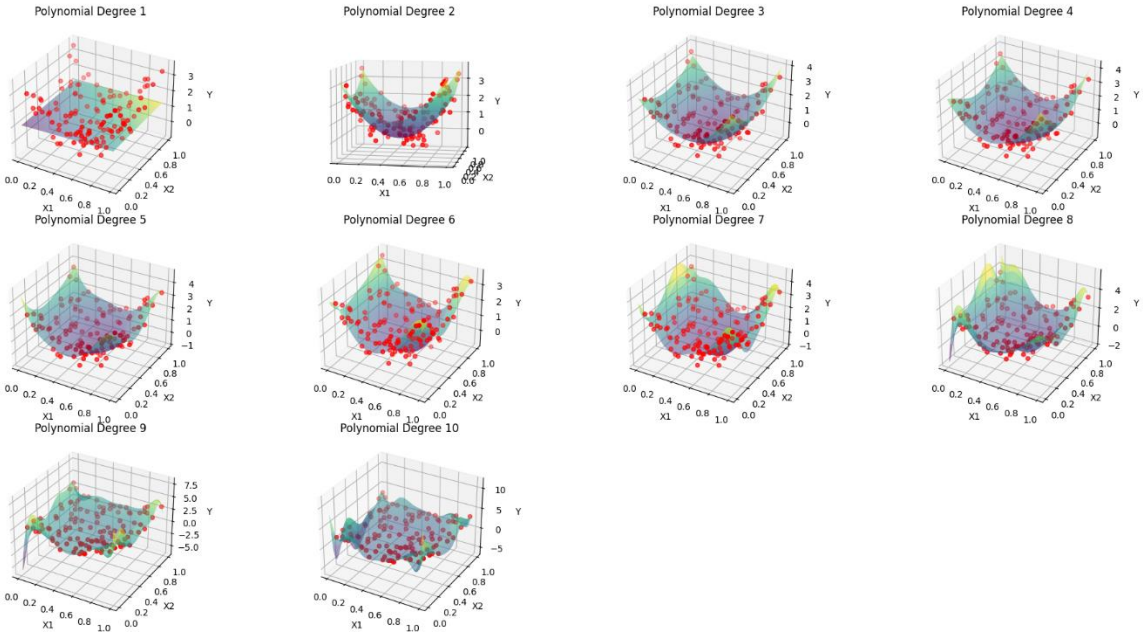

Fig4: surface of the learned function alongside with the training examples.

As we note, the Fig4 explain Fig3.

3-)

In this we will use ridge regression and take the best regularization parameter.

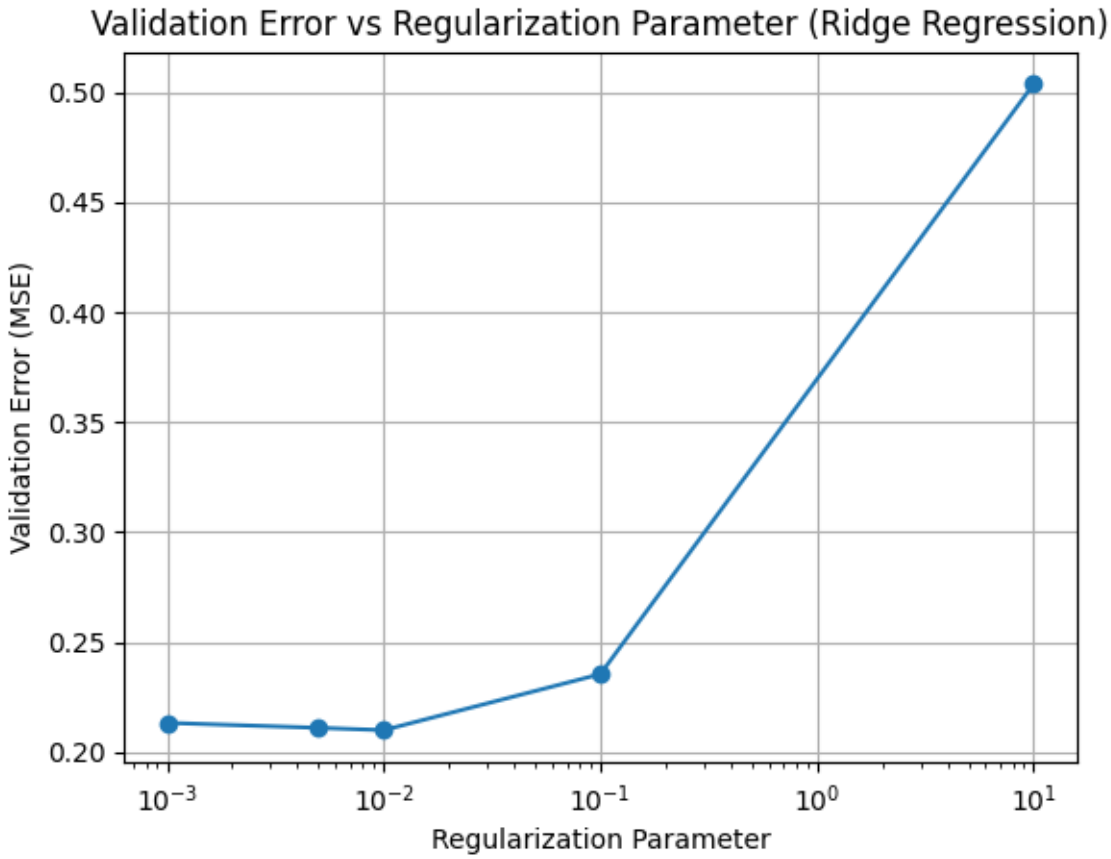

Fig5: Validation Error vs regularization parameter.

As we see, the best regularization parameter is 0.01, which give the least VE.

Logistic Regression

1-)

In this we will apply logistic regression in liner mode (degree=1).

So, we get these results:

```
RangeIndex: 62 entries, 0 to 61
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x1      62 non-null     float64
 1   x2      62 non-null     float64
 2   class   62 non-null     object
dtypes: float64(2), object(1)
memory usage: 1.6+ KB
None
Classification Report test:
              precision    recall  f1-score   support

           0       0.75      0.55      0.63        11
           1       0.64      0.82      0.72        11

    accuracy                           0.68        22
   macro avg       0.70      0.68      0.68        22
weighted avg       0.70      0.68      0.68        22

Classification Report train:
              precision    recall  f1-score   support

           0       0.67      0.65      0.66        31
           1       0.66      0.68      0.67        31

    accuracy                           0.66        62
   macro avg       0.66      0.66      0.66        62
weighted avg       0.66      0.66      0.66        62
```

Fig6: data set information and Classification Report.

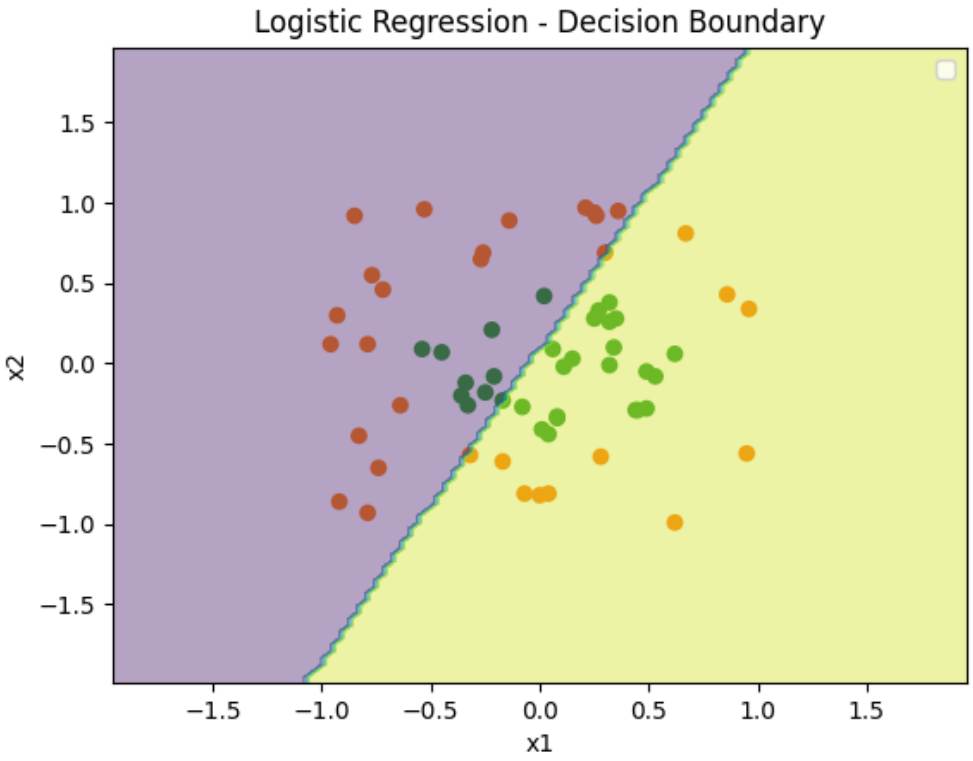

Fig7: linear decision boundary.

As we see, from the report, where accuracy for test set = 0.68, train test = 0.66, and other metrics like Precision, Recall, f1-score are all low in the two sets (0.66 – 0.70).

So, from the metrics and even from the Fig7, this module has the underfitting problem and the module is not performing well.

2-)

In this we will apply logistic regression in quadratic mode (degree=2).

So, we get these results:

```
RangeIndex: 62 entries, 0 to 61
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x1      62 non-null     float64
 1   x2      62 non-null     float64
 2   class   62 non-null     object
dtypes: float64(2), object(1)
memory usage: 1.6+ KB
None
Classification Report test:
              precision    recall  f1-score   support

           0       1.00      0.91      0.95        11
           1       0.92      1.00      0.96        11

    accuracy                           0.95        22
   macro avg       0.96      0.95      0.95        22
weighted avg       0.96      0.95      0.95        22

Classification Report train:
              precision    recall  f1-score   support

           0       1.00      0.94      0.97        31
           1       0.94      1.00      0.97        31

    accuracy                           0.97        62
   macro avg       0.97      0.97      0.97        62
weighted avg       0.97      0.97      0.97        62
```

Fig8: data set information and Classification Report.

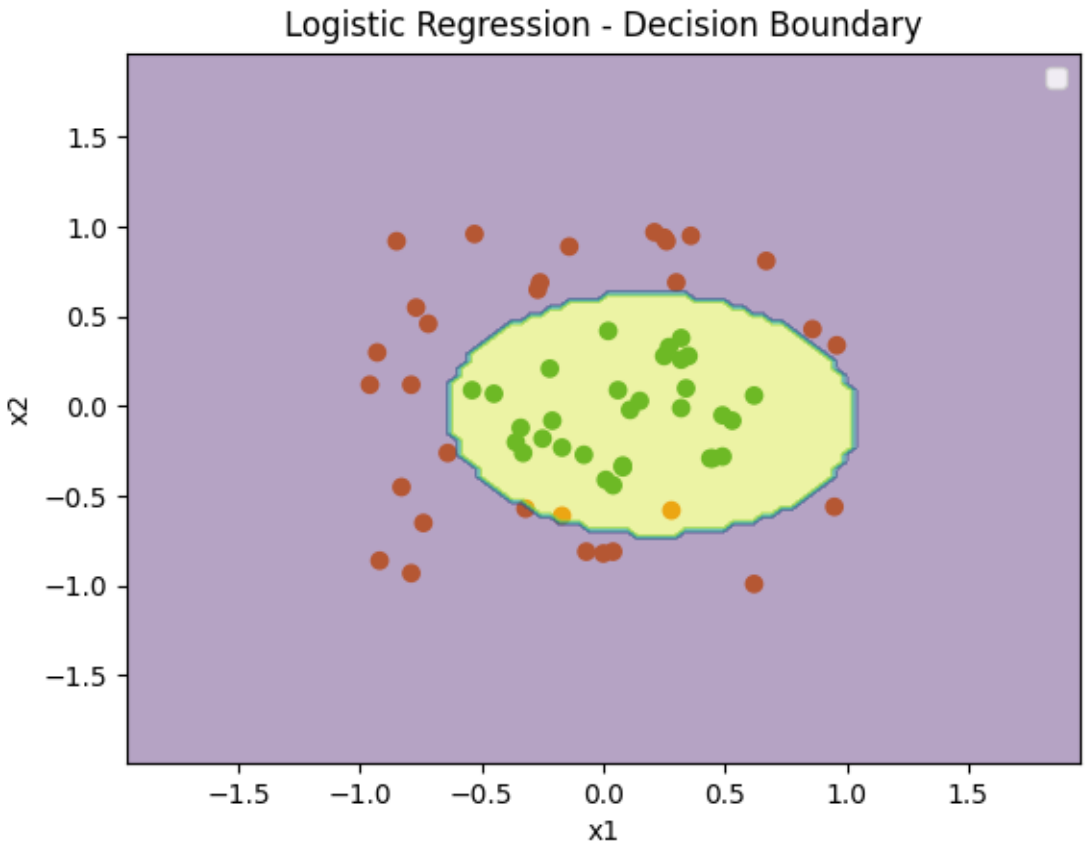

Fig7: quadratic decision boundary.

As we see, from the report, where accuracy for test set = 0.95, train test = 0.97, and other metrics like Precision, Recall, f1-score are all vary high in the two sets (0.95 – 0.97).

So, from the metrics and even from the Fig7, this module may has no problem in overfitting or under fitting, because accuracy on both the training and test sets is high, indicating that the model is making the correct predictions, and all other metrics are in the same levels and values, and the module is performing well.

(3) is already answered in 1&2.