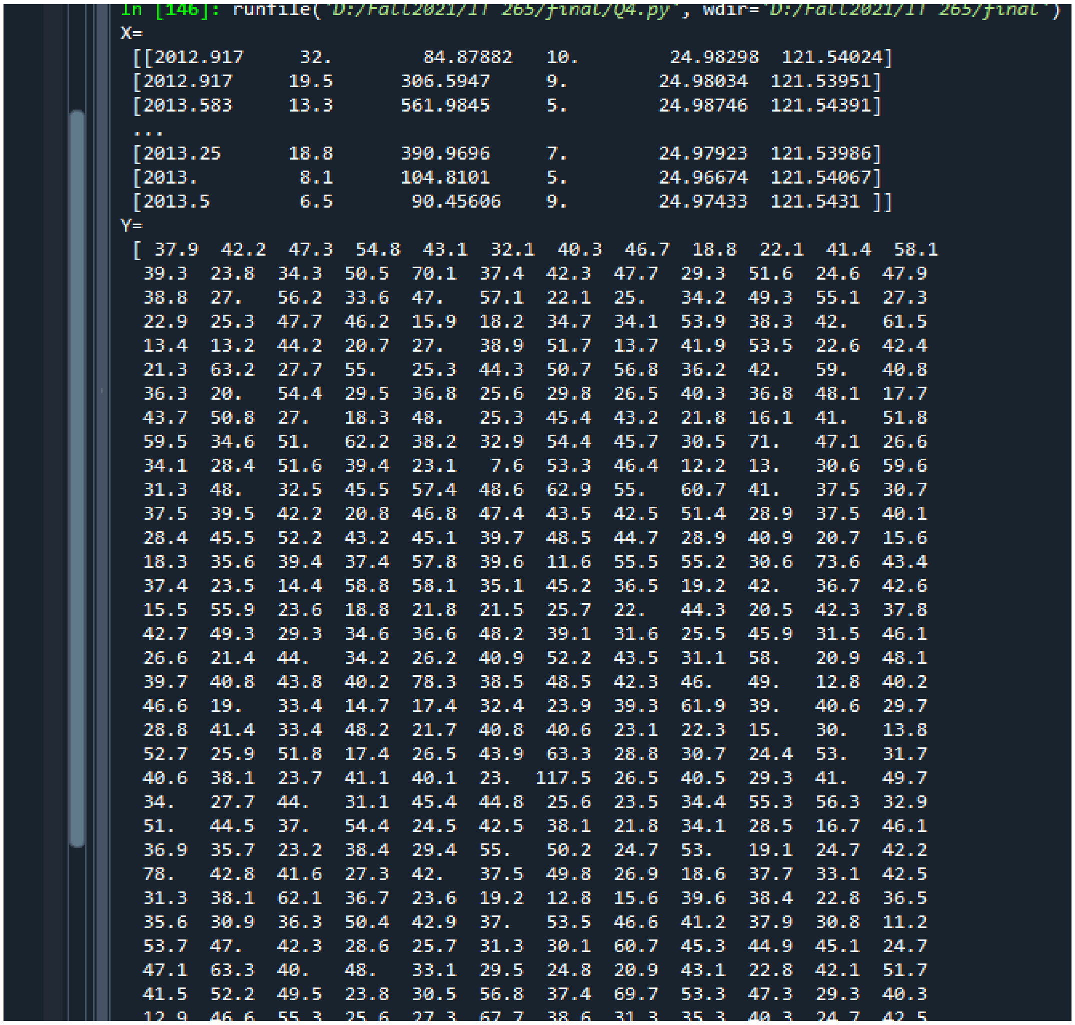
Final ITS256

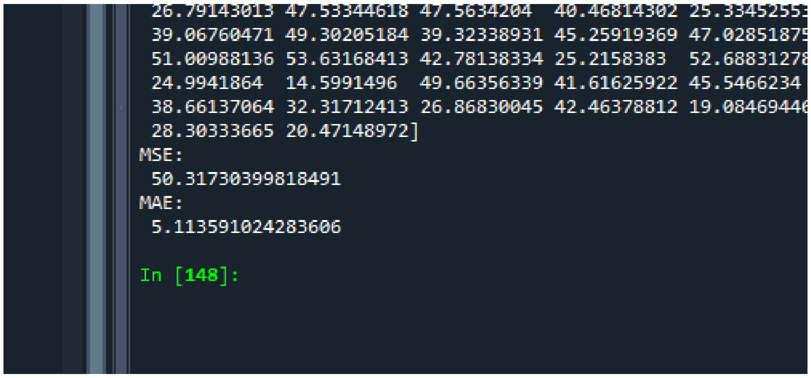
Ghadir Alfadhl

Question 4: Create a neural network in Python and SKLearn using MLPRegressor (produces continuous output). The purpose of the neural network program is to predict real estate value based on the input data. Use the provided real-estate dataset in csv format. Create an input dataset from columns 2 – 6 corresponding to the input feature set and an output dataset corresponding to the corresponding real estate prices (last column in the dataset). Use Sklearn train_test_split function to create the training and test datasets. Configure the MLPRegressor neural network as follows:

- Input layer having 5 inputs.
- 2 hidden layers each with 100 nodes (Hint: in the nn parameters use (100,100) to represent it).
- Use the "relu" activation function
- Use the "adam" solver
- Set a learning rate of 0.001
- Number of epochs = 300
- Loss/cost function = mean spared error (MSE) Train the neural network with the training set and then use then test it using the test input dataset. Print out the mean square error and absolute error



```
In [146]: runfile('D:/Fall2021/IT 265/final/Q4.py', wdir='D:/Fall2021/IT 265/final')
X=
 [[2012.917     32.        84.87882   10.        24.98298  121.54024]
 [2012.917     19.5      306.5947     9.        24.98034  121.53951]
 [2013.583     13.3      561.9845     5.        24.98746  121.54391]
 ...
 [2013.25      18.8      390.9696     7.        24.97923  121.53986]
 [2013.         8.1      104.8101     5.        24.96674  121.54067]
 [2013.5        6.5       90.45606    9.        24.97433  121.5431 ]]
Y=
 [ 37.9  42.2  47.3  54.8  43.1  32.1  40.3  46.7  18.8  22.1  41.4  58.1
  39.3  23.8  34.3  50.5  70.1  37.4  42.3  47.7  29.3  51.6  24.6  47.9
  38.8  27.   56.2  33.6  47.   57.1  22.1  25.   34.2  49.3  55.1  27.3
  22.9  25.3  47.7  46.2  15.9  18.2  34.7  34.1  53.9  38.3  42.   61.5
  13.4  13.2  44.2  20.7  27.   38.9  51.7  13.7  41.9  53.5  22.6  42.4
  21.3  63.2  27.7  55.   25.3  44.3  50.7  56.8  36.2  42.   59.   40.8
  36.3  20.   54.4  29.5  36.8  25.6  29.8  26.5  40.3  36.8  48.1  17.7
  43.7  50.8  27.   18.3  48.   25.3  45.4  43.2  21.8  16.1  41.   51.8
  59.5  34.6  51.   62.2  38.2  32.9  54.4  45.7  30.5  71.   47.1  26.6
  34.1  28.4  51.6  39.4  23.1   7.6  53.3  46.4  12.2  13.   30.6  59.6
  31.3  48.   32.5  45.5  57.4  48.6  62.9  55.   60.7  41.   37.5  30.7
  37.5  39.5  42.2  20.8  46.8  47.4  43.5  42.5  51.4  28.9  37.5  40.1
  28.4  45.5  52.2  43.2  45.1  39.7  48.5  44.7  28.9  40.9  20.7  15.6
  18.3  35.6  39.4  37.4  57.8  39.6  11.6  55.5  55.2  30.6  73.6  43.4
  37.4  23.5  14.4  58.8  58.1  35.1  45.2  36.5  19.2  42.   36.7  42.6
  15.5  55.9  23.6  18.8  21.8  21.5  25.7  22.   44.3  20.5  42.3  37.8
  42.7  49.3  29.3  34.6  36.6  48.2  39.1  31.6  25.5  45.9  31.5  46.1
  26.6  21.4  44.   34.2  26.2  40.9  52.2  43.5  31.1  58.   20.9  48.1
  39.7  40.8  43.8  40.2  78.3  38.5  48.5  42.3  46.   49.   12.8  40.2
  46.6  19.   33.4  14.7  17.4  32.4  23.9  39.3  61.9  39.   40.6  29.7
  28.8  41.4  33.4  48.2  21.7  40.8  40.6  23.1  22.3  15.   30.   13.8
  52.7  25.9  51.8  17.4  26.5  43.9  63.3  28.8  30.7  24.4  53.   31.7
  40.6  38.1  23.7  41.1  40.1  23.  117.5  26.5  40.5  29.3  41.   49.7
  34.   27.7  44.   31.1  45.4  44.8  25.6  23.5  34.4  55.3  56.3  32.9
  51.   44.5  37.   54.4  24.5  42.5  38.1  21.8  34.1  28.5  16.7  46.1
  36.9  35.7  23.2  38.4  29.4  55.   50.2  24.7  53.   19.1  24.7  42.2
  78.   42.8  41.6  27.3  42.   37.5  49.8  26.9  18.6  37.7  33.1  42.5
  31.3  38.1  62.1  36.7  23.6  19.2  12.8  15.6  39.6  38.4  22.8  36.5
  35.6  30.9  36.3  50.4  42.9  37.   53.5  46.6  41.2  37.9  30.8  11.2
  53.7  47.   42.3  28.6  25.7  31.3  30.1  60.7  45.3  44.9  45.1  24.7
  47.1  63.3  40.   48.   33.1  29.5  24.8  20.9  43.1  22.8  42.1  51.7
  41.5  52.2  49.5  23.8  30.5  56.8  37.4  69.7  53.3  47.3  29.3  40.3
  12.9  46.6  55.3  25.6  27.3  67.7  38.6  31.3  35.3  40.3  24.7  42.5
```



```
  26.79143013 47.53344618 47.5634204  40.46814302 25.3345255
  39.06760471 49.30205184 39.32338931 45.25919369 47.02851875
  51.00988136 53.63168413 42.78138334 25.2158383  52.6883127
  24.9941864  14.5991496  49.66356339 41.61625922 45.5466234
  38.66137064 32.31712413 26.86830045 42.46378812 19.0846944
  28.30333665 20.47148972]
 MSE:
  50.31730399818491
 MAE:
  5.113591024283606

 In [148]:
```

Code:

import pandas as pd

```
import numpy as np
from sklearn import datasets
from sklearn.neural_network import MLPRegressor
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split



data= pd.read_csv("Real estate valuation data set.csv")
#print(data)

x = data.drop(['No', 'Y house price of unit area'], axis=1).values
y = data['Y house price of unit area'].values

# x = data.iloc[2:, :7]
# y = data.iloc[:,7]

print("X=\n",x)
print("Y=\n",y)


x_train, x_test, y_train, y_test = train_test_split(x, y)

regr = MLPRegressor(hidden_layer_sizes=(100, 100),
                    activation='relu',
                     alpha=1e-4,
                    solver='adam',
                     tol=1e-4,
                    learning_rate_init=0.001,
                   max_iter=300,
                   random_state=1,
                   verbose=True)

regr.fit(x_train, y_train)
pred = regr.predict(x_test)
print(pred)

from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,pred)
print("MSE:\n", mse)


from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_test,pred)
print("MAE:\n", mae)
```