Kavya Shah SUID: 330126500 PIN: 68

Introduction to ML HW-06

Problem Statement:

The problem statement of this problem is to do

1. Linear Regression in a way that it minimises Mean Squared Error and fits the line near citations 2017 - 2021. We need to predict the 2022 citation numbers based on that line.
2. Logistic Regression in a way such that it categorises individuals into one of three categories based on the ratio of citations in 2022 to citations in 2021, approximated to two decimals.

1. Low (ratio < 1.05)
2. Medium (ratio: 1.06 - 1.15)
3. High (ratio > 1.15)

This aims to compare the results of all the models including HW-04 (5-3-1 Neural network Architecture), HW-5 (6-6-3 Neural Network Architecture), Linear Regression, and Logistic Regression.
The dataset is split into two parts: 80% for training and 20% for testing.

Approach:

1. Imported the important and useful libraries.
2. Loaded the data set into the python notebook using pandas.
3. Explored and analysed the data.
4. Linear Regression Approach:
    a. Splitting the data into training (80 %) and testing (20 %) set using sklearn. This division allows for model training and evaluation on separate databases
    b. I created a Linear Regression Model using sklearn library and applied it to the training data and fit it.
    c. I used test data to predict the values of citation_2022
    d. Calculated Mean Squared Error to evaluate the accuracy of the model

5. Logistic Regression Approach:
    a. I calculated the citation ration of citations in 2022 to citations in 2021 of each individual.
    b. Divided the cit_2022 according to the low, medium and high category
    c. Splitting the data into training (80 %) and testing (20 %) set using sklearn. This division allows for model training and evaluation on separate databases
    d. After which I applied the Logistic Regression Model using sklearn library and applied it to the training data and fit it.
    e. I used test data to predict the correct category of test dataset.
    f. Atlast, I calculated the accuracy and Mean Squared Error to evaluate the accuracy of the model.

9. Evaluation of Results:

- After training, I evaluated the model's performance using the remaining 20% of the test data.
    1. Mean Square Error: To find the mean error between the predicted and the actual values.
    2. R2 Score: To calculate the accuracy of the model.
    3. Accuracy: To find how correct model predictions are.
    4. Loss: To measure the error between model predictions and actual labels.
- The evaluation metrics provided insights into how well the model has learned to classify individuals based on the data set and citation ratios (in logistic regression).

Result:

Linear Regression:

In [21]:
```
# Calculating MSE for accuracy

mse_lr = mean_squared_error(y_test,y_pred)
print(f'Linear Regression Mean Squared Error : {mse_lr}')

```

```
Linear Regression Mean Squared Error : 14687.2157143493
```

HW-04:

```
# calculate MAE

from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.metrics import r2_score
y_pred = model.predict(x_test)
mean_squared_error = mean_squared_error(y_test, y_pred)
mean_absolute_error = mean_absolute_error(y_test, y_pred)
r2_score = r2_score(y_test, y_pred)
print(f"Mean squared Error (MSE): {mean_squared_error}")
# print(f"Mean Absolute Error (MAE): {mean_absolute_error}")
# print(f"r2_score: {r2_score}")
```

```
1/1 [==============================] - 0s 56ms/step
Mean squared Error (MSE): 2425.074829658851
```

Logistic Regression:

In [22]:
```
# calculating MSE
mse = mean_squared_error(y_test,y_pred)
print(f'Logistic Regression Mean Squared Error: {mse}')

```

```
Logistic Regression Mean Squared Error: 0.15
```

HW-05:

:
```
max_values = y_pred.max(axis=1, keepdims=True)

# Create a binary array by comparing each element with the maximum value
y_pred = (y_pred == max_values).astype(int)

# calculating MSE
mse = mean_squared_error(y_test,y_pred)
print(f'Mean Squared Error: {mse}')

```

```
Mean Squared Error: 0.09999999999999999
```

Conclusion:

	MSE
Linear Regression	14687.21
Logistic Regression	0.15
HW-04	2425.07
HW-05	0.09

The MSE indicates, on average, the squared difference between the predicted and actual values.

Comparison between Linear Regression and HW-04 and HW-05:
The Neural Network model with 6-6-3 architecture from HW-05 outperforms both the Multiple Linear Regression model and the Neural network with 5-3-1 architecture in terms of MSE.
A lower MSE generally indicates a better performing model.
Hence, based on the results the Neural Network model with a 6-6-3 architecture appears to be a more powerful model for my specific 61-70 dataset task, providing a better fit to the data compared to the Multiple Linear Regression Model and HW-04.

Comparison between Logistic Regression and HW-04 and HW-05:
The neural network model from HW-05 exhibits a lower MSE (0.09) compared to the Logistic Regression (0.15) and HW-04. A lower MSE indicates better performance in regression tasks , suggesting neural networks with 6-6-3 architecture may have outperformed both Logistic Regression and Neural network with 5-3-1 architecture.

Hence, The Neural network models (hw-04 and HW-05) both have lower MSE values compared to the Linear and Logistic Regression.
Among the models compared, HW-05 (Neural Network with 6-6-3 architecture) has the lowest MSE, indicating the superior performance in minimising MSE i.e. prediction errors.