Regression

About Regression

- What is regression?
- When is it used?

Let’s use an example of house rent again

Let’s see the data

- Mount the google drive
- Upload the dataset in your designated folder in your drive
- Access it using pandas
- Visualize it

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Class Contents/5. Summer 2024/AI Lab/Class Prep/Week 8 Prep/House_Rent_Dataset.csv')
print(df)
```

```
    Square Feet   House Rent
0           602  1358.738483
1           935  1881.890576
2          1360  2515.869060
3           770  1663.208241
4           606  1420.654762
5           571  1362.399731
6          1200  2373.118906
7           520  1356.935748
8          1114  2049.044709
9           621  1461.672061
10          966  1936.447802
11          714  1562.806644
12          830  1671.183515
13          958  2011.349048
14          587  1379.277241
```

Let’s see the data

Is there a better way to visualize this?

- Mount the google drive
- Upload the dataset in your designated folder in your drive
- Access it using pandas
- Visualize it

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Class Contents/5. Summer 2024/AI Lab/Class Prep/Week 8 Prep/House_Rent_Dataset.csv')
print(df)
```

```
    Square Feet   House Rent
0           602  1358.738483
1           935  1881.890576
2          1360  2515.869060
3           770  1663.208241
4           606  1420.654762
5           571  1362.399731
6          1200  2373.118906
7           520  1356.935748
8          1114  2049.044709
9           621  1461.672061
10          966  1936.447802
11          714  1562.806644
12          830  1671.183515
13          958  2011.349048
14          587  1379.277241
```

But first, let us transform it into a NumPy array

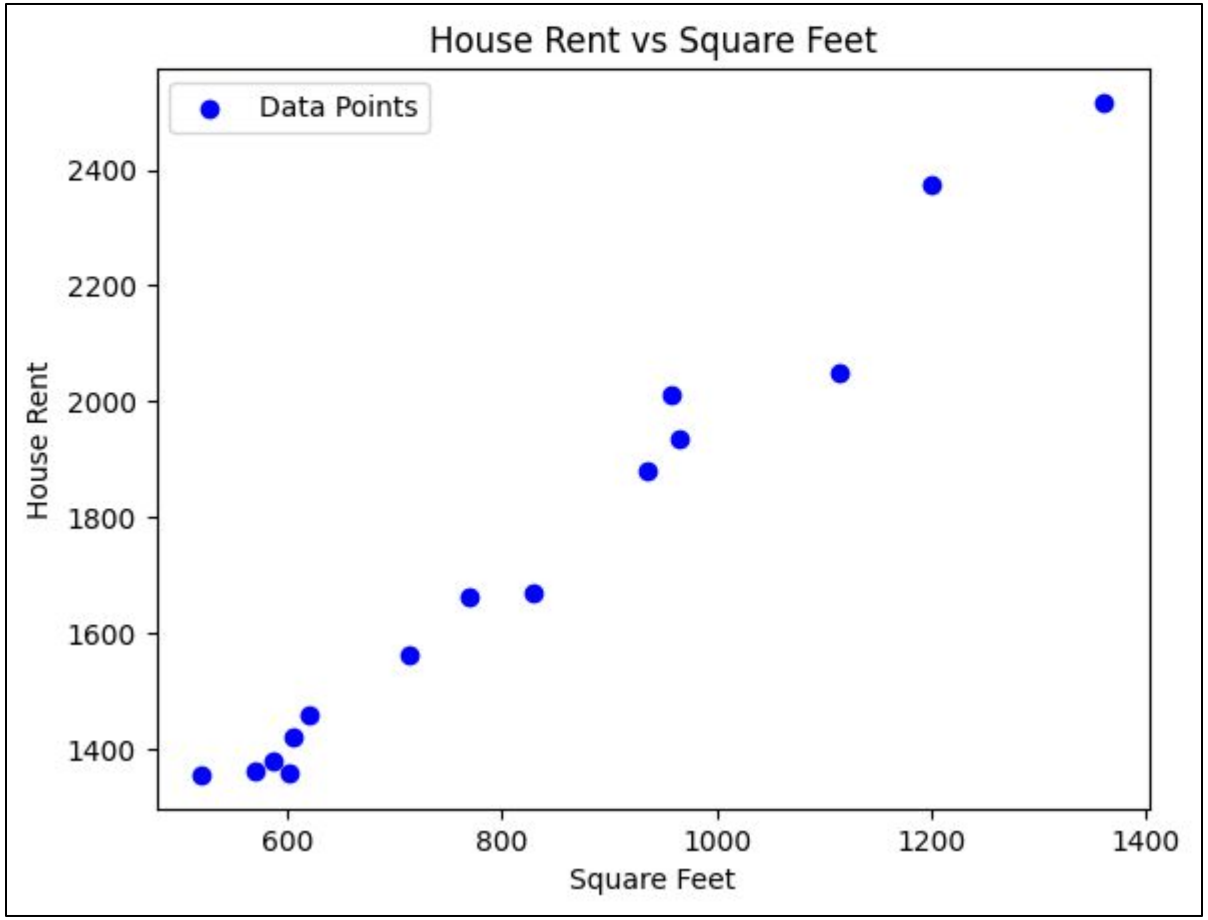
- We had done this before!

```
import numpy as np
# Convert the DataFrame to a NumPy array
data_array = df.to_numpy()
data_array=data_array.astype(float)
# Print the NumPy array
print(data_array)
```

Introducing matplotlib

- Does this plotting remind you of something?
- Y=mx+c

```
import matplotlib.pyplot as plt

xs= data_array[:, 0]
ys = data_array[:, 1]
plt.scatter(xs, ys, color='blue', label='Data Points')

# Adding labels and title
plt.xlabel('Square Feet')
plt.ylabel('House Rent')
plt.title('House Rent vs Square Feet')
plt.legend()

# Show the plot
plt.show()
```



Linear Regression – finding the best st. line

$$y = a_0 + a_1 x$$

$$a_1 = \frac{n\sum(x_i y_i) - \sum x_i \sum y_i}{n\sum(x_i^2) - (\sum x_i)^2}$$

$$a_0 = \frac{\sum y_i - a_1 \sum x_i}{n}$$

Python code for a1

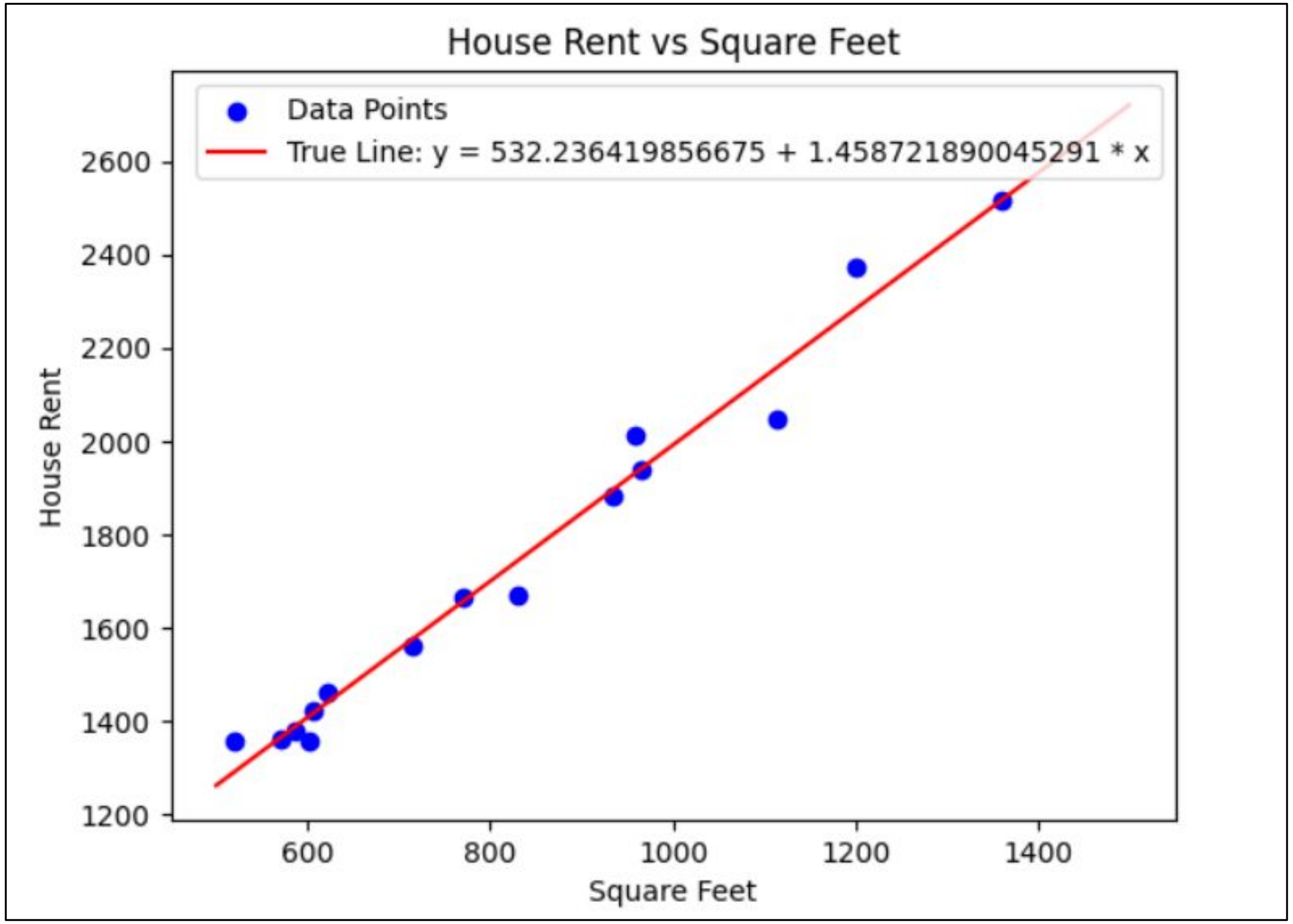
```
xs= data_array[:, 0]
ys = data_array[:, 1]


products_of_xs_and_ys = np.multiply(xs,ys)

sum_of_products_of_xs_and_ys = np.sum(products_of_xs_and_ys)

sum_of_xs = np.sum(xs)
sum_of_ys = np.sum(ys)

x_squares = xs*xs
sum_of_squared_xs = np.sum(x_squares)

sum_of_xs_squared = sum_of_xs**2

n=xs.shape[0]

a1 = (n*sum_of_products_of_xs_and_ys-sum_of_xs*sum_of_ys)/(n*sum_of_squared_xs-sum_of_xs_squared)
print(a1)
```

Python code for a0

```
a0 = (sum_of_ys-a1*sum_of_xs)/n
print(a0)
```

Now plot this line

- y=a0 + a1x



Metrics for evaluating the obtained equation

$$S_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

$$r = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{\left(n\sum x_i^2 - \left(\sum x_i\right)^2\right)\left(n\sum y_i^2 - \left(\sum y_i\right)^2\right)}}$$

$$r^2$$

Metrics for evaluating the obtained equation

1. Sr (Residual Sum of Squares):

- This is the total amount of error in your regression model.
- It tells you how far off your predicted values are from the actual values.
- A smaller Sr means your model predicts the data more accurately.
- **Example**: If Sr=500, this means the total error in the predictions (squared differences between actual and predicted values) adds up to 500.

$$S_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

$$r = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{\left(n\sum x_i^2 - \left(\sum x_i\right)^2\right)\left(n\sum y_i^2 - \left(\sum y_i\right)^2\right)}}$$

$$r^2$$

Metrics for evaluating the obtained equation

2. Sy/x (Standard Error of the Estimate):

- This tells you, on average, how much your predictions deviate from the actual values.
- It’s like the average "error" per prediction.

Example: If Sy/x=5, this means, on average, your model's predictions are 5 units off from the actual values.

$$S_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

$$r = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{\left(n\sum x_i^2 - \left(\sum x_i\right)^2\right)\left(n\sum y_i^2 - \left(\sum y_i\right)^2\right)}}$$

$$r^2$$

Metrics for evaluating the obtained equation

3. r (Pearson Correlation Coefficient):

- This measures the strength and direction of the relationship between your independent variable and dependent variable.
- It ranges from -1 to 1:
  - r=1: Perfect positive linear relationship.
  - r=−1: Perfect negative linear relationship.
  - r=0: No linear relationship.
- **Example**: If r=0.85, this means a strong positive relationship between x and y, meaning as x increases, y also increases.

$$S_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

$$r = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{\left(n\sum x_i^2 - \left(\sum x_i\right)^2\right)\left(n\sum y_i^2 - \left(\sum y_i\right)^2\right)}}$$

$$r^2$$

Metrics for evaluating the obtained equation

4. r^2 (Coefficient of Determination):

- This is the square of the correlation coefficient, and it tells you how much of the variation in y is explained by the model.
- r_2 ranges from 0 to 1:
  - $r^2=1$: Your model perfectly explains the data.
  - $r^2=0$: Your model does not explain any of the variation in the data.
- **Example**: If $r^2=0.72$, this means 72% of the variation in y (e.g., house rent) is explained by the independent variable x (e.g., square footage).

$$S_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

$$r = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{\left(n\sum x_i^2 - \left(\sum x_i\right)^2\right)\left(n\sum y_i^2 - \left(\sum y_i\right)^2\right)}}$$

$$r^2$$

Assignment 3 – Task 1

- Calculate the S_r , $S_{x/y}$, r and r^2 for the problem solved in class.

Polynomial Regression

$$y = a_0 + a_1 x + a_2 x^2$$

$$(n)a_0 + \left(\sum x_i\right)a_1 + \left(\sum x_i^2\right)a_2 = \sum y_i$$

$$\left(\sum x_i\right)a_0 + \left(\sum x_i^2\right)a_1 + \left(\sum x_i^3\right)a_2 = \sum x_i y_i$$

$$\left(\sum x_i^2\right)a_0 + \left(\sum x_i^3\right)a_1 + \left(\sum x_i^4\right)a_2 = \sum x_i^2 y_i$$

$$s_{y/x} = \sqrt{\frac{S_r}{n - (m + 1)}}$$

Multiple Linear Regression

$$y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_mx_m$$

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{Bmatrix}$$

$$s_{y/x} = \sqrt{\frac{S_r}{n - (m + 1)}}$$

Assignment 3 – Task 2

- A dataset for carrying out multiple linear regression will be given. You need to build a model using the formula given in the previous slide.

- Note: There will be a quiz based on today’s lecture, codes and your assignment.