ASSIGNMENT 2 : LEAST SQUARE FITTING

Computer vision

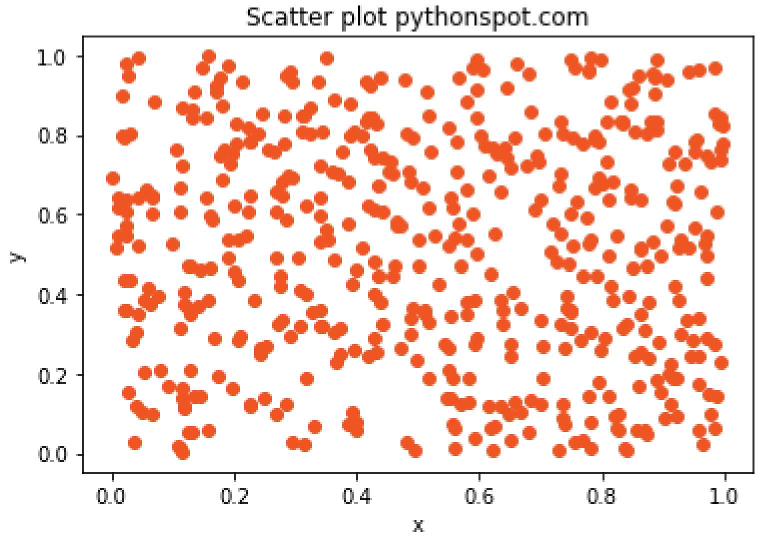
NAME: Reem Mahmoud ELsayed
ID: 4525

Assignment 2: least square method

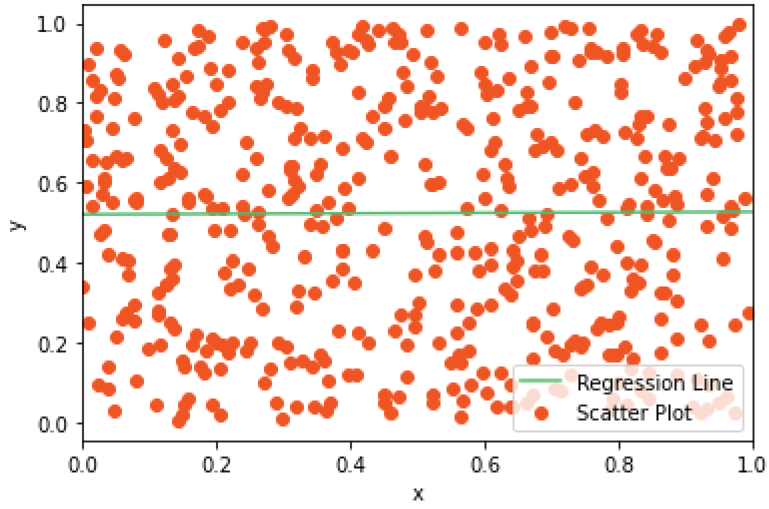
Q1) Implement Least Square Fitting algorithm for line detection (single line) from binary images. Once again, the output would be a binary image with fitted line.

Answer:

Input image:



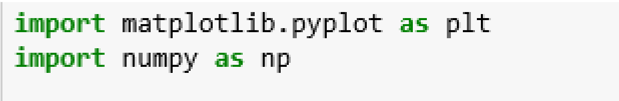
Output image:



The code using python language:

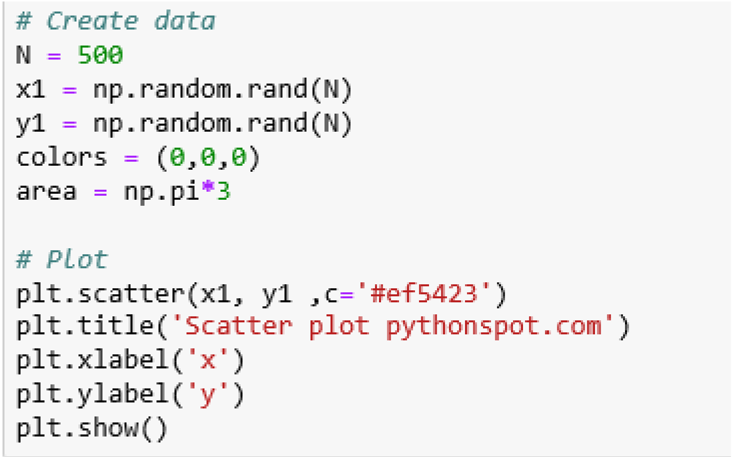
Using jupyter notebook

1)first we import the libraries:



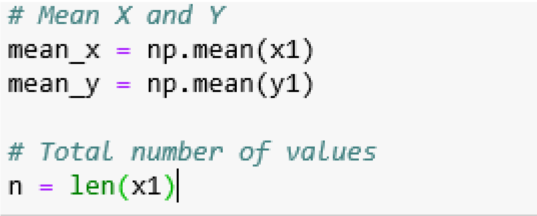
```python
import matplotlib.pyplot as plt
import numpy as np
```

2)we create our data and plotted as scattered and this will be our binary image

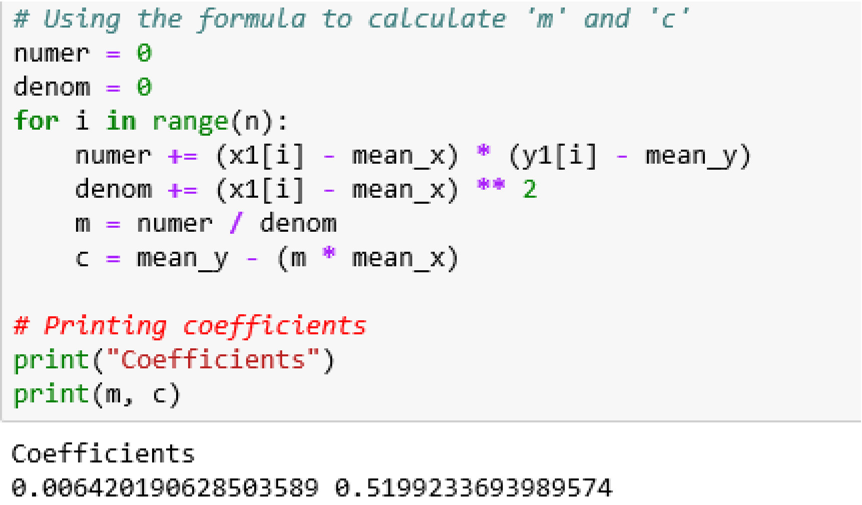


```python
# Create data
N = 500
x1 = np.random.rand(N)
y1 = np.random.rand(N)
colors = (0,0,0)
area = np.pi*3

# Plot
plt.scatter(x1, y1 ,c='#ef5423')
plt.title('Scatter plot pythonspot.com')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

3)the implementation of least square fitting to detect single line algorithm



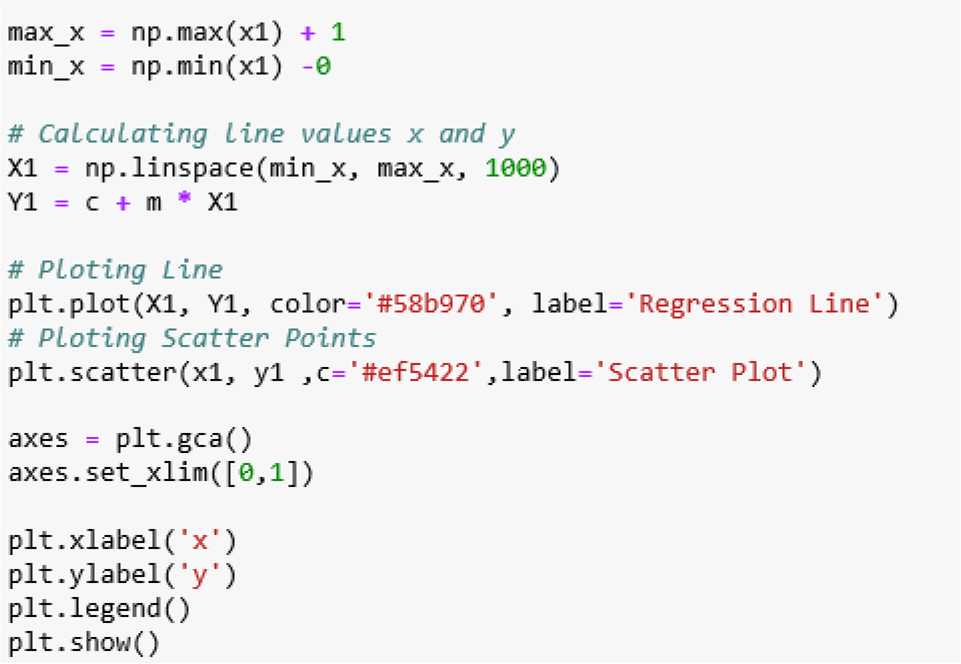
```python
# Mean X and Y
mean_x = np.mean(x1)
mean_y = np.mean(y1)

# Total number of values
n = len(x1)
```



```python
# Using the formula to calculate 'm' and 'c'
numer = 0
denom = 0
for i in range(n):
    numer += (x1[i] - mean_x) * (y1[i] - mean_y)
    denom += (x1[i] - mean_x) ** 2
    m = numer / denom
    c = mean_y - (m * mean_x)

# Printing coefficients
print("Coefficients")
print(m, c)
```

```
Coefficients
0.006420190628503589 0.5199233693989574
```

4)plotting the line detected on the image



```
max_x = np.max(x1) + 1
min_x = np.min(x1) -0

# Calculating line values x and y
X1 = np.linspace(min_x, max_x, 1000)
Y1 = c + m * X1

# Ploting Line
plt.plot(X1, Y1, color='#58b970', label='Regression Line')
# Ploting Scatter Points
plt.scatter(x1, y1 ,c='#ef5422',label='Scatter Plot')

axes = plt.gca()
axes.set_xlim([0,1])

plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```