1st Question:

From the above data:

1. Read the data with pandas and find features and target variables.

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error

file = pd.read_csv('/content/Salary_Data.csv')
file
```

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

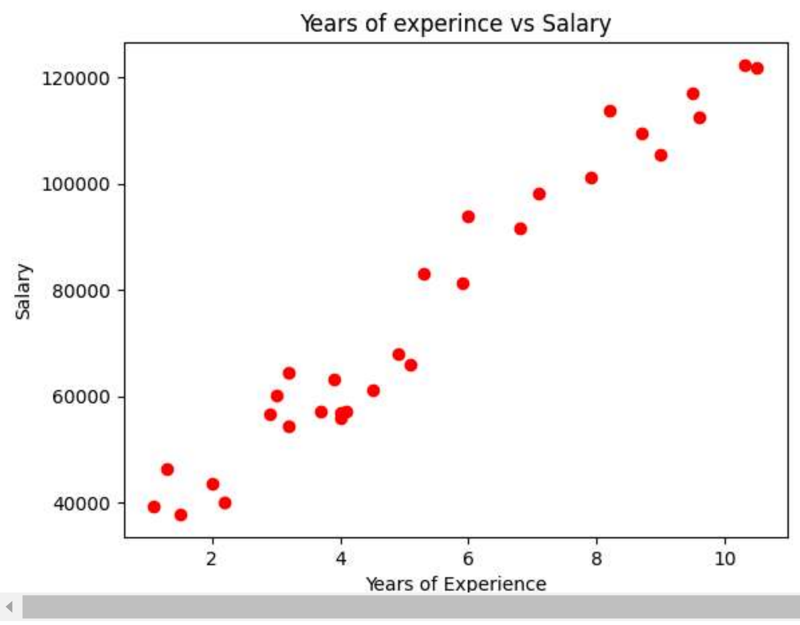
Next steps: Generate code with `file` View recommended plots New interactive sheet

Targets and features

```
X = file['YearsExperience']
y = file['Salary']
```

2. Plot a graph between features and target

```
plt.scatter(X, y, color='red')
plt.title('Years of experince vs Salary')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



3. Find Best fit line using linear regression.

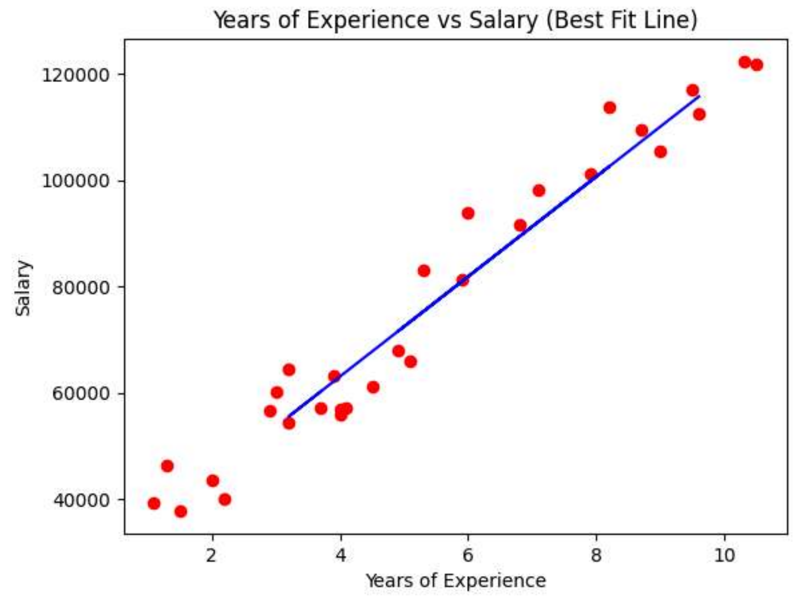
Suggested code may be subject to a license | AxeForward/TS-Project

```
X_train, X_test, y_train, y_test = train_test_split(X.values.reshape(-1, 1), y, test_size=0.2, random_state=42)
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

X_train_25, X_test_25, y_train_25, y_test_25 = train_test_split(X.values.reshape(-1, 1), y, test_size=0.25, random_state=42)
lr.fit(X_train_25, y_train_25)
y_pred_25 = lr.predict(X_test_25)

plt.scatter(X, y, color='red')
plt.plot(X_test, y_pred, color='blue')
plt.title('Years of Experience vs Salary (Best Fit Line)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()

print(f"Slope: {lr.coef_[0]}, Intercept: {lr.intercept_}")
```



```
Slope: 9371.016079696601, Intercept: 25478.12743600038
```

4. Find MSE, MAE, for test size (20,25)

MSE

```
mse_20 = mean_squared_error(y_test, y_pred)
mse_25 = mean_squared_error(y_test_25, y_pred_25)
print(f"MSE (20% test size): {mse_20}")
print(f"MSE (25% test size): {mse_25}")
```

```
MSE (20% test size): 49830096.85590839
MSE (25% test size): 38802588.99247065
```

MAE

```
mae_20 = mean_absolute_error(y_test, y_pred)
mae_25 = mean_absolute_error(y_test_25, y_pred_25)
print(f"MAE (20% test size): {mae_20}")
print(f"MAE (25% test size): {mae_25}")
```

```
MAE (20% test size): 6286.453830757749
MAE (25% test size): 5056.995466663592
```

2nd Question:

1. Read the data with pandas and find features and target variables

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

```
data = pd.DataFrame({
    'X1': [1.2, 2.4, 3.5, 4.1, 1.9, 3.7, 2.1, 3.3, 2.8, 4],
    'X2': [2.3, 1.9, 2.7, 3, 2.8, 2.5, 3.2, 2.4, 3.1, 3.3],
    'X3': [3.1, 2.8, 1.5, 3.6, 2.5, 1.9, 2.2, 3, 1.8, 2.7],
    'X4': [4.2, 3.5, 2.9, 4.8, 3.2, 4, 4.1, 4.5, 3.6, 4.9],
    'Y': [15.6, 13.1, 12.4, 18.2, 14.1, 16, 14.5, 17.3, 14.9, 19.2]
})
data
```

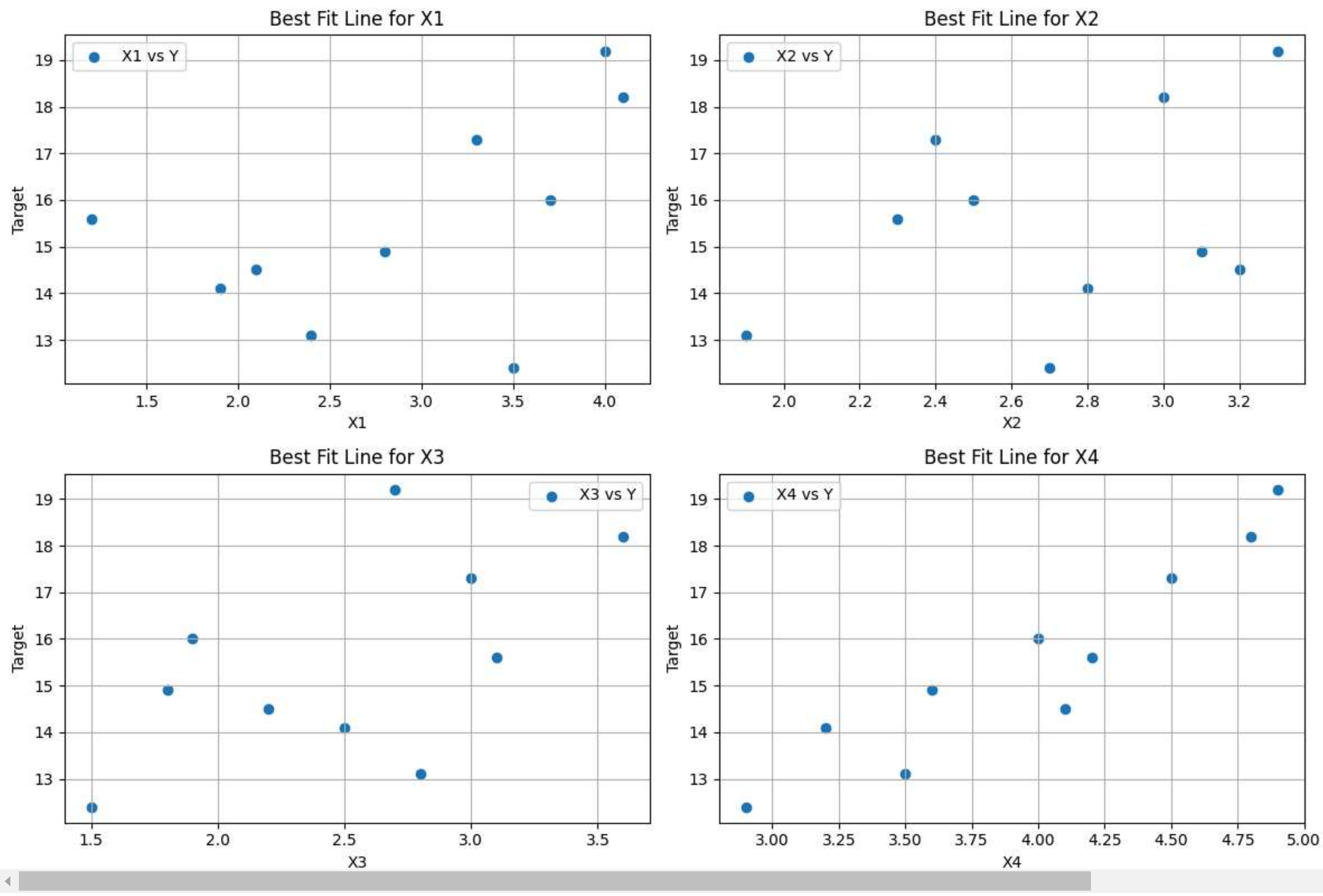
	X1	X2	X3	X4	Y
0	1.2	2.3	3.1	4.2	15.6
1	2.4	1.9	2.8	3.5	13.1
2	3.5	2.7	1.5	2.9	12.4
3	4.1	3.0	3.6	4.8	18.2
4	1.9	2.8	2.5	3.2	14.1
5	3.7	2.5	1.9	4.0	16.0
6	2.1	3.2	2.2	4.1	14.5
7	3.3	2.4	3.0	4.5	17.3
8	2.8	3.1	1.8	3.6	14.9
9	4.0	3.3	2.7	4.9	19.2

Next steps: Generate code with data View recommended plots New interactive sheet

```
df = pd.DataFrame(data)
y = df['Y']
X = df.drop('Y', axis=1)
```

2. Plot a graph between features and target

```
plt.figure(figsize=(12, 8))
for i, feature in enumerate(X.columns):
    plt.subplot(2, 2, i+1)
    plt.scatter(df[feature], y, label=f'{feature} vs Y')
    plt.xlabel(feature)
    plt.ylabel('Target')
    plt.grid(True)
    plt.legend()
    plt.title(f'Best Fit Line for {feature}')
plt.tight_layout()
plt.show()
```

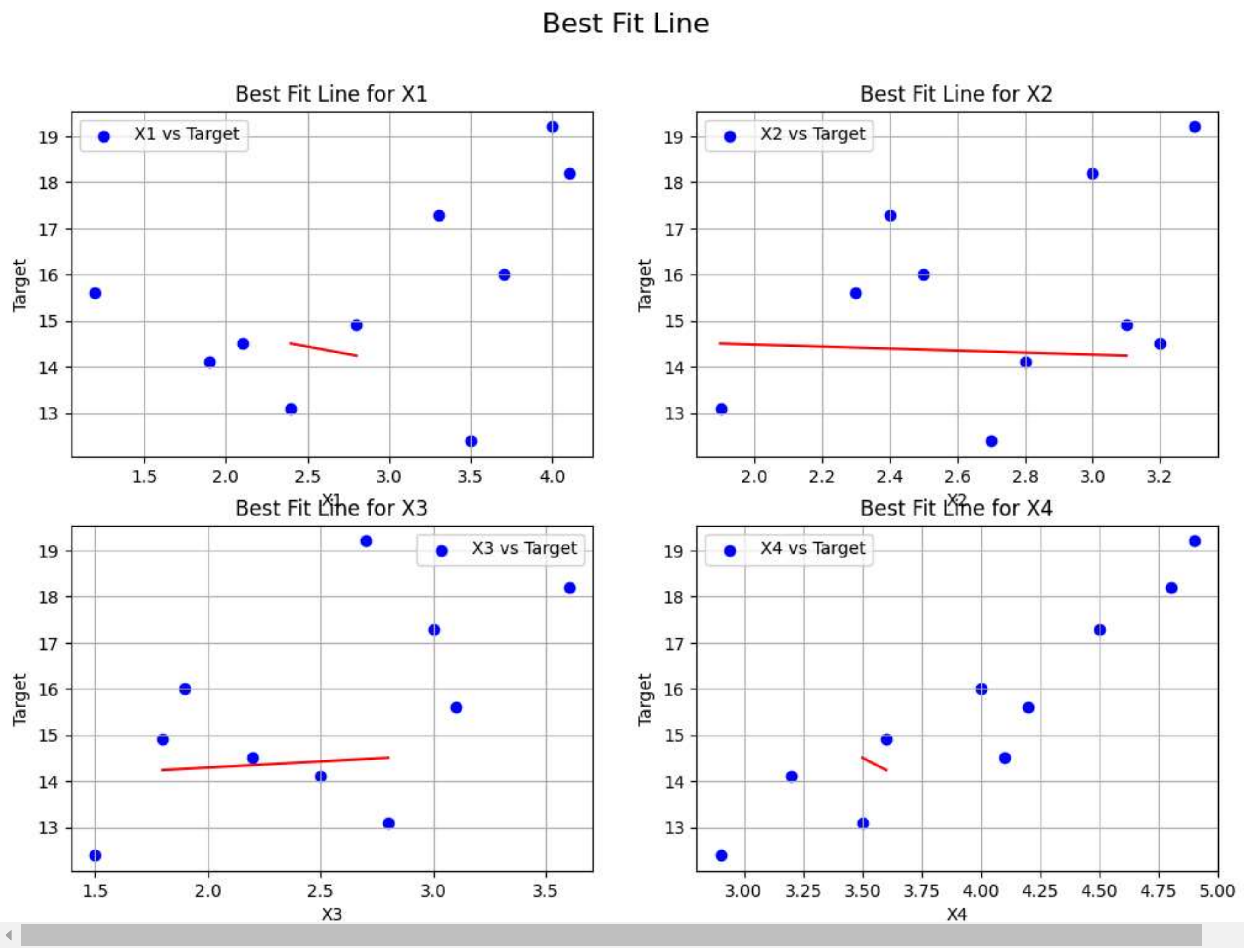


3. Find Best fit line using linear regression.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

X_train_30, X_test_30, y_train_30, y_test_30 = train_test_split(X, y, test_size=0.3, random_state=42)
lr.fit(X_train_30, y_train_30)
y_pred_30 = lr.predict(X_test_30)

plt.figure(figsize=(12, 8))
plt.suptitle('Best Fit Line', fontsize=16)
for i, feature in enumerate(X.columns):
    plt.subplot(2, 2, i+1)
    plt.scatter(X[feature], y, label=f'{feature} vs Target', color='blue')
    plt.plot(X_test[feature], y_pred, color='red')
    plt.xlabel(feature)
    plt.ylabel('Target')
    plt.grid(True)
    plt.legend()
    plt.title(f'Best Fit Line for {feature}')
plt.show()
```



4. Find MSE , MAE for test size (20,30)

MSE

```
mse_20 = mean_squared_error(y_test, y_pred)
mse_30 = mean_squared_error(y_test_30, y_pred_30)
print(f"MSE (20% test size): {mse_20}")
print(f"MSE (30% test size): {mse_30}")
```

```
MSE (20% test size): 1.2020186561889346
MSE (30% test size): 0.8283376869765737
```

MAE

```
mae_20 = mean_absolute_error(y_test, y_pred)
mae_30 = mean_absolute_error(y_test_30, y_pred_30)
print(f"MAE (20% test size): {mae_20}")
print(f"MAE (30% test size): {mae_30}")
```

```
MAE (20% test size): 1.0313248712002014
MAE (30% test size): 0.8651731636820733
```