import Necessary Library

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load & Data Acquisition

In [ ]:
```
df = pd.read_csv("Salary_dataset (1).csv")
df.head()
```

In [ ]:
```
# Here, we can see our dataset is sort of linear
plt.scatter(df['YearsExperience'],df['Salary'])
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt.show()
```

In [ ]:
```
# we are creating new dataframe
df1 = df[['YearsExperience','Salary']].head()
```

In [ ]:
```
# Import necessary libraries
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import tensorflow as tf

# Sample data: X (input), Y (output) for training
# Example: Let's assume we're predicting the output of a linear equation y = 2x + 1
X = df1['YearsExperience']
Y = df1['Salary']

# Build the ANN model
model = Sequential([
    Dense(units=1, input_shape=(1,), activation='linear')
    ])

# Compile the model with SGD optimizer
model.compile(optimizer='sgd', loss='mean_squared_error')

# Train the model without displaying the epochs
model.fit(X, Y, epochs=1000, verbose=0)

# Predict the output for a user input
user_input = float(input("Enter a value for prediction: "))
prediction = model.predict(np.array([[user_input]]))

# Display the prediction
print(f"Predicted output for input {user_input}: {prediction[0][0]}")
```

Predict salary for each YearsExperience in the dataset

predictions = model.predict(X)

Here, we can see clearly actual and predicted salary

df1 = pd.DataFrame({"Actual_Salary" : Y ,"Predicted_Salary": predictions.flatten()})

df1

In [ ]: