```
# Step 1: Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Step 2: Load Dataset
# You can replace this with your own dataset
df = pd.read_csv("https://raw.githubusercontent.com/selva86/datasets/master/Salary_Data.csv")  # Example dataset

# Step 3: Explore Data
print(df.head())
print(df.info())
print(df.describe())

# Step 4: Data Visualization (EDA)
plt.figure(figsize=(6,4))
sns.scatterplot(x='YearsExperience', y='Salary', data=df)
plt.title("Experience vs Salary")
plt.show()
```

```
# Step 5: Preprocessing (if any categorical columns present)
# If you have categorical columns like Job Title, use LabelEncoder:
# df['JobTitle'] = LabelEncoder().fit_transform(df['JobTitle'])

# Step 6: Define Features and Target
X = df[['YearsExperience']]  # Change/add more features if available
y = df['Salary']

# Step 7: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 8: Train Models

# Linear Regression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
lr_pred = lr_model.predict(X_test)

# Random Forest
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)

# Step 9: Evaluation Function
def evaluate_model(y_true, y_pred, model_name):
    print(f"\n🔍 Evaluation for {model_name}:")
    print("MAE:", mean_absolute_error(y_true, y_pred))
    print("MSE:", mean_squared_error(y_true, y_pred))

    print("RMSE:", np.sqrt(mean_squared_error(y_true, y_pred)))

    print("R² Score:", r2_score(y_true, y_pred))


evaluate_model(y_test, lr_pred, "Linear Regression")

evaluate_model(y_test, rf_pred, "Random Forest Regressor")


# Step 10: Prediction Example

experience = [[5]]  # Example: Predict salary for 5 years of experience

predicted_salary = rf_model.predict(experience)

print(f"\n💼 Predicted Salary for 5 years experience: ₹{predicted_salary[0]:,.2f}")
```