Linear Regression on Salary Dataset

The dataset consists of two columns: "YearsExperience" and "Salary".

1. YearsExperience: This column represents the number of years of experience an individual has in a particular field. It is a continuous numerical variable.

2. Salary: This column represents the corresponding salary of an individual based on their years of experience. It is also a continuous numerical variable.

The dataset is used for performing linear regression, where the goal is to predict the salary based on the number of years of experience. The "YearsExperience" column serves as the input feature, and the "Salary" column serves as the target variable.

By analyzing this dataset and applying regression techniques, we can explore the relationship between years of experience and salary and build a model that can predict the salary for individuals based on their experience.

1 Setup

Download Salary.zip from ed. Use the following command to read the data.

```
import pandas as pd
df = pd.read_csv("Salary_dataset.csv")
X = df["YearsExperience"].values
X = X.reshape(-1,1)
Y = df['Salary']
```

2 Data Exploration and Visualization

Split the dataset into train and test sets: The train_test_split function from scikit-learn is used to split the dataset into training and testing sets. This step is essential to evaluate the performance of the model on unseen data.

```
from sklearn.model_selection import train_test_split
# Extract features and target variable
# Split the dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=1)
```

The arguments passed to train_test_split are X and Y, representing the features and target variable respectively. The test_size parameter is set to 0.2, indicating that 20% of the data will be used for testing, and the remaining 80% will be used for training. The random_state parameter is set to 1 to ensure reproducibility of the split.

By splitting the dataset into training and testing sets, we can train the model on the training data and evaluate its performance on the testing data. This allows us to estimate how well the

model will generalize to new, unseen data. It helps us detect overfitting (when the model performs well on the training data but poorly on the testing data) and assess the model's ability to make accurate predictions on real-world data.

Use the following code to visualize your training and testing data.

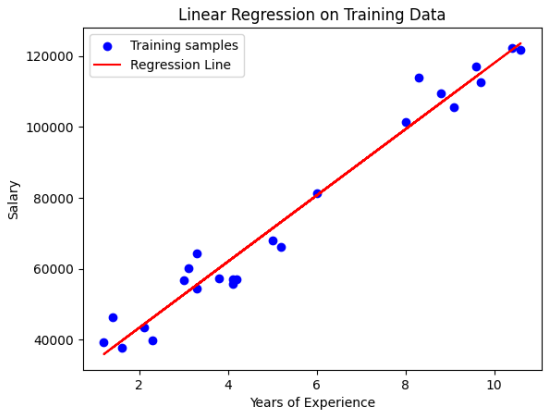
```
import matplotlib.pyplot as plt
# Visualize the data
plt.figure(figsize=(10,8))
plt.scatter(X_train, y_train, c='blue', label='Training')
plt.scatter(X_test, y_test, c='red', label='Testing')
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.legend()
plt.show()
```

3 Main Task: Model Training and Evaluation

In this section, we will focus on training a linear regression model to predict the salary based on the number of years of experience. We will utilize the LinearRegression class from the sklearn library to implement linear regression (example). Follow the steps below:

1. Train a linear regression model using the training data (X_train and y_train).
2. Print out the intercept and coefficient of your model.
3. Use the trained model to make predictions on the training set (X_train) and calculate the mean squared error (MSE) between the predicted and actual values.
4. Use the trained model to make predictions on the test set (X_test) and calculate the mean squared error (MSE) between the predicted and actual values.
5. Visualize your training model by plotting all the training samples (as shown in Section 2) and then plot the regression line on top of them.

Checkpoint: If everything goes correctly, you should see the following graph:



4 Advanced Task: Dealing with an outlier

Mark Zuckerberg is an American technology entrepreneur and philanthropist. In 2024, his annual salary is reported to be 2.7 milllion dollars (I made this number up for this exercise).

Now, let's merge his data into our dataset: (years of experience: 19, salary: 27e5).

1. Repeat all the steps in Section 3 to train a linear regression model.
2. Train a ridge regression model instead of linear regression.

   Sklearn Ridge Regression Model: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ridge_regression.html
3. Try the following **alpha** values for ridge regression:
   - 1
   - 10
   - 100
   - 1000
   - 10000

5 Analytic Questions

1. `LinearRegression` has one argument named `fit_intercept` that is set to `True` by default. Comment on the effect of this option on your model. In other words, what would happen if you change it to `False`?
2. What happened to your model when we add Mark's record into the training data? Describe your model with MSE and plots.
3. What could you do to improve the model with Mark's record?