1.Title
How to Estimate Salary with Linear Regression

2.Abstract :

- Does our salary really grow with our years of working experience? While this can be answered based on our general understandings of the job market, we will use a simple data-driven approach to verify the fact. We will use **linear regression** to model the relationship between the amount of salary with the years of working experience.
- Linear regression is a model that assumes a linear relationship between an **explanatory variable (X) and a response variable (y)**. We can predict the value of y based on the value of X. In our context here, the estimated salary will be our response variable (y) since it is our target predicted value and the years of working experience will be our explanatory variable (X).

3.Dataset description:

- Linear regression is a model that assumes a linear relationship between an **explanatory variable (X) and a response variable (y)**. We can predict the value of y based on the value of X. In our context here, the estimated salary will be our response variable (y) since it is our target predicted value and the years of working experience will be our explanatory variable (X).
- Once we have defined the explanatory variable and response variable in our case, we will use **Python** to build a linear regression model to address our question.

Linear Regression

1. Loading data

Firstly, we will use the *Python Pandas* library to read our CSV data.

- Import all the required libraries.
  Use the *Pandas read_csv* function to read the CSV file. This function will return the data in a dataframe format.

Index	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2	43525
4	2.2	39891
5	2.9	56642
6	3	60150
7	3.2	54445

Line 9–10: Extract the column of *YearsExperience* and *Salary* and assign them to the variables X and y, respectively.

4. modules description :

2. Splitting data into a training set and a test set

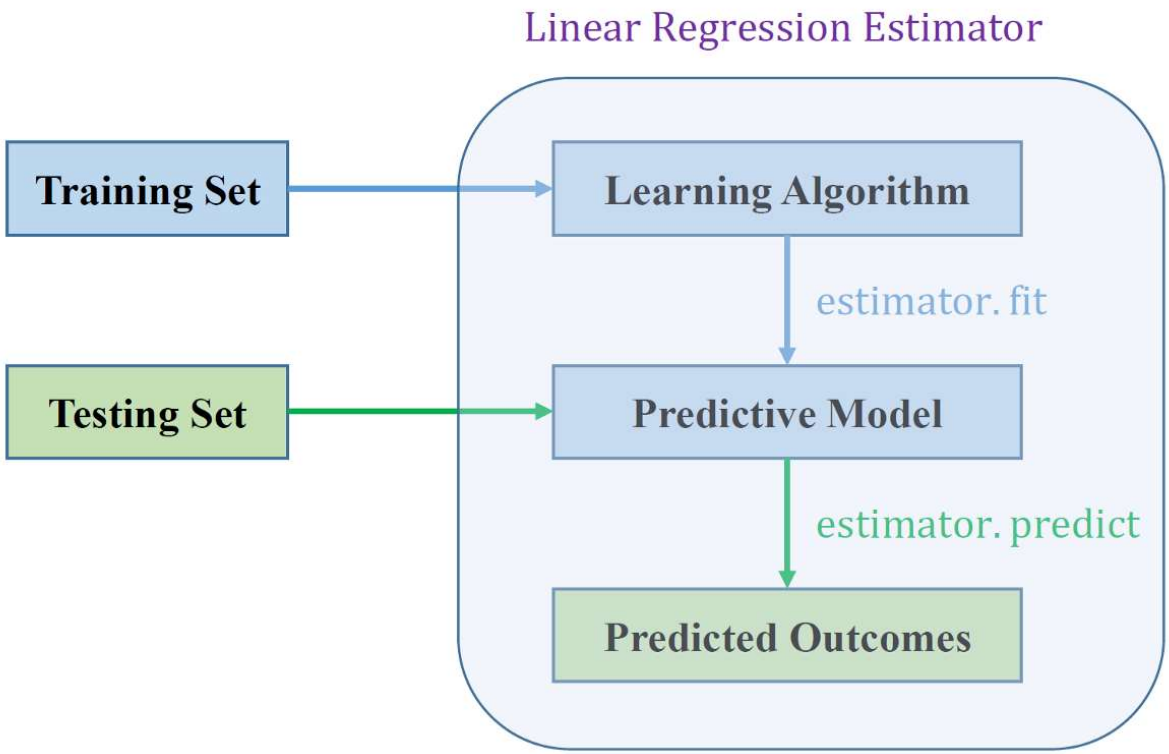
- Prior to building a linear model, we need to prepare a training set and a test set (part of standard procedure in a machine learning workflow). The training set will be used to train the model whereas the test set will be used to assess the performance of the trained model in predicting the result from unseen data.
- **3. Data Transformation**
- *Python scikit-learn* only accepts the training and test data in a 2-dimensional array format. We have to perform data transformation on our training set and test set.
- **4. Training Model**
- Now we are ready to train our linear model.

Algorithm:
linear regression

- Linear regression is **a statistical method for modeling relationships between a dependent variable with a given set of independent variables**. Note: In this article, we refer to dependent variables as responses and independent variables as features for simplicity.

1. Initialize the parameters.
2. Predict the value of a dependent variable by given an independent variable.
3. Calculate the error in prediction for all data points.
4. Calculate partial derivative w.r.t a0 and a1.
5. Calculate the cost for each number and add them.
6. Update the values of a0 and a1.

Archtecture diagram:

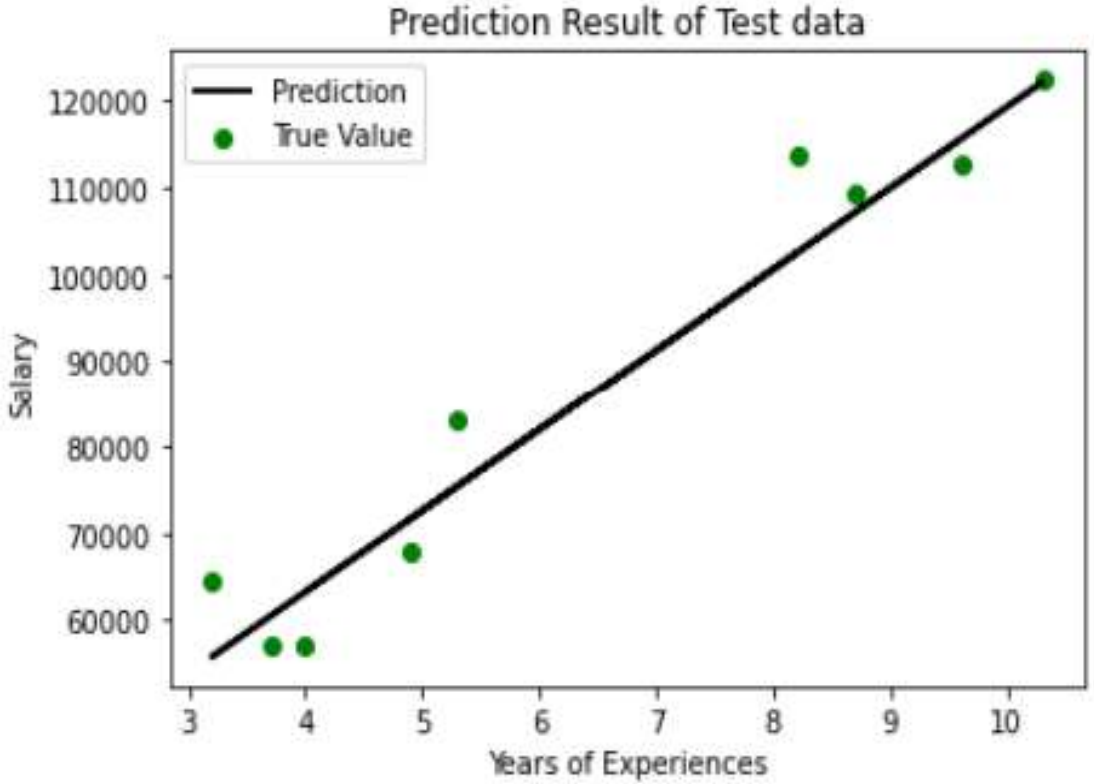


5.Result and discussion:

- **5. Predicting Salary using Linear Model**
- At this stage, we have trained a linear model and we first use it to predict the salary on our training set to see how well it fit on the data.
- Use the Matplotlib to create a plot to visualize the predicted results. The “true values” are plotted as the blue dots on the chart and the predicted values are plotted as a black color straight line.

Now, we need to check if the linear model can perform well on our test set (unknown data).

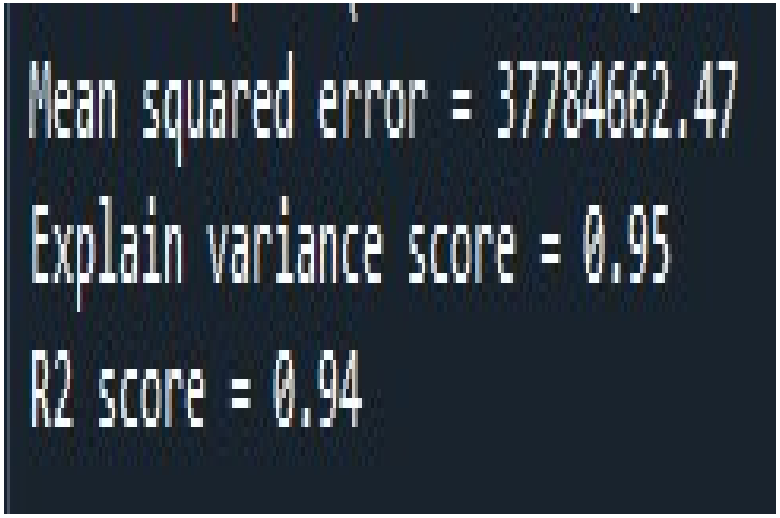
- Use the linear model to predict the salary based on the test set.
- Use the Matplotlib to create a plot to visualize the predicted results. The “true values” are plotted as the green dots on the chart and the predicted values are plotted as a black color straight line.
-



The graph shows that our linear model can fit quite well on the test set. We can observe a linear pattern of how the amount of salary is increased by the years of experience.

- **Model Evaluation**
- The previous section uses a graphical approach to evaluate the performance of our linear model.
- **Evalution parameters :**
- Here we will use some quantitative methods to obtain a more precise performance evaluation of our linear model.
- **Mean Square Error** — The average of the squares of the difference between the true values and the predicted values. The lower the difference the better the performance of the model. This is a common metric used for regression analysis.
- **Explained Variance Score** — A measurement to examine how well a model can handle the variation of values in the dataset. A score of 1.0 is the perfect score.
- **R2 Score** — A measurement to examine how well our model can predict values based on the test set (unknown samples). The perfect score is 1.0.

Use the *Scikit-learn Metrics* functionalities to calculate the mean square error, explained variance, and R2 score for our linear model. We feed the functions with the true values (test set) and the predicted values.



If we perform a root square calculation on our mean squared error, we will gain an average discrepancy of around $6146.92 which is quite a low error. Besides, the explain variance score and the R2 score hit 0.9 above. This shows our linear model is not overfitted and can work nicely to predict the salary based on new data.

➢ Conclusion :

We have managed to build a simple linear model to predict salary based on years of working experience. Based on our linear model, **we can conclude that our salary is grown with** salary**our years of working experience and there is a linear relationship between them.** We can use our linear model to predict the by giving input of years of experience.

Source code:

- import pandas as pd
- import numpy as np
- from sklearn.model_selection import train_test_split
- from sklearn import linear_model
- import sklearn.metrics as sm
- import matplotlib.pyplot as plt
- data = pd.read_csv("Data/Salary_Data.csv")
- X = data['YearsExperience']
- y = data['Salary’]
- X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, random_state=42)

- X_train = np.array(X_train).reshape((len(X_train),1))
- y_train = np.array(y_train).reshape((len(y_train),1))
- X_test = np.array(X_test).reshape(len(X_test), 1)
- y_test = np.array(y_test).reshape(len(y_test), 1)

- model = linear_model.LinearRegression()
- model.fit(X_train, y_train)
- y_train_pred = model.predict(X_train)

- plt.figure()
- plt.scatter(X_train, y_train, color='blue', label="True Value")
- plt.plot(X_train, y_train_pred, color='black', linewidth=2, label="Prediction")
- plt.xlabel("Years of Experiences")
- plt.ylabel("Salary")
- plt.title('Prediction Result of Training Data')
- plt.legend()
- plt.show()

- y_test_pred = model.predict(X_test)

- plt.figure()
- plt.scatter(X_test, y_test, color='green', label='True Value')
- plt.plot(X_test, y_test_pred, color='black', linewidth=2, label='Prediction')
- plt.xlabel("Years of Experiences")
- plt.ylabel("Salary")
- plt.title('Prediction Result of Test data')
- plt.legend()
- plt.show()
- print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_pred), 2))
- print("Explain variance score =", round(sm.explained_variance_score(y_test, y_test_pred), 2))
- print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))