

aSSIGNMENT 1

Submitted by Mohammad Tayyab Alam 24157



March 2, 2025

Financial data analytics

**Linear Regression Salary Questions**

# Questions

**1. How can we interpret the coefficient of "years of experience"? What does a positive coefficient indicate?**

The coefficient of years of experience shows us the exact amount by which the salary will increase for a given year(s) of experience. In this case, the coefficient of years of experience is 5044.76. Since it is positive, it means that the salary increases by $5044.76 for every unit increase in years of experience.

# Print all coefficients

coefficients = model.coef\_

print(f"Coefficients: {coefficients}")

# Print the intercept

intercept = model.intercept\_

print(f"Intercept: {intercept}")

Coefficients: [5044.76561066]

Intercept: 24836.27778369729

**2. Why is it important to check the distribution of salary and years of experience before applying regression?**

Checking the distribution helps identify skewness, outliers, and the relationship between variables. This ensures that the assumptions of linear regression (linearity, normality, homoscedasticity) are met, leading to more reliable results.

**3. What happens if we use all the data for training and none for testing? How does it impact model evaluation?**

We can see that the R^2 of the data with 100% being used for train is higher than that of the split data but we can not make any comments on the performance of the mocdel on unseen data. This is because being trained on a single piece of data leads to overfitting.

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

# Load the data

X = df[['Years Experience']]

y = df['Salary']

# Scenario 1: Train-Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train the model on the training set

model\_split = LinearRegression().fit(X\_train, y\_train)

# Evaluate on the training set

train\_score\_split = model\_split.score(X\_train, y\_train)

# Evaluate on the testing set

test\_score\_split = model\_split.score(X\_test, y\_test)

print("Train-Test Split Scenario:")

print(f"Training R²: {train\_score\_split}")

print(f"Testing R²: {test\_score\_split}")

print()

# Scenario 2: Use All Data for Training

model\_all = LinearRegression().fit(X, y)

# Evaluate on the entire dataset

train\_score\_all = model\_all.score(X, y)

print("All Data for Training Scenario:")

print(f"Training R² (using all data): {train\_score\_all}")

Train-Test Split Scenario:

Training R²: 0.8904357718490602

Testing R²: 0.8958649764604112

All Data for Training Scenario:

Training R² (using all data): 0.8916383204458667

**4. What does it mean if the model's predicted salaries are consistently lower or higher than actual salaries?**

This suggests that the model has a systematic bias. Possible reasons include missing important predictor variables, using an incorrect functional form (e.g., assuming a linear relationship when the true relationship is non-linear), or overfitting. Lower prediction leads to under fitting while higher prediction leads to overfitting.

**5. How does increasing or decreasing the training dataset size affect model performance?**

* Increasing the dataset size generally improves model performance by making it more generalizable.
* Decreasing the dataset, size can lead to overfitting, where the model captures noise instead of actual patterns.

**6. If another variable, such as "certifications," is added to the dataset, how might it impact the regression results?**

If certifications significantly influence salaries, adding them as a feature should improve the model. It could help explain salary variations that experience alone cannot.

**7. If the dataset contained outliers (e.g., a person with 50 years of experience but a very low salary), how would that affect the model?**

Such outliers could distort the model by pulling the regression line away from the general trend, reducing its predictive accuracy. Robust regression techniques or outlier handling methods like winsorization may be needed.

**8. If the dataset only contains a small number of observations, what problems might arise when training the model?**

* Higher variance in model estimates
* Less reliable coefficient estimates
* Greater risk of overfitting

**9. How does the LinearRegression().fit() function work in training the model?**

The function calculates the best-fit line by minimizing the sum of squared errors between actual and predicted salaries. It determines the optimal coefficients using Ordinary Least Squares (OLS).

**10. If the dataset had outliers (e.g., a CEO with 40 years of experience earning $1M), how would that affect the model?**

* High outliers can skew the regression line, leading to an overestimation of salaries for lower-experience employees.
* The Line of best fit would move above.
* This issue can be mitigated using transformations like log-salaries or robust regression.

**11. If the company wanted to predict salary for interns (0 years experience), would this model still be reliable? Why or why not?**

The model may not be reliable because it is trained on data where all employees have some experience. Extrapolating to zero years may produce inaccurate predictions if the relationship between experience and salary changes at the lower end. To resolve this a salary pool of interns should be used as they have no experience, the y intercept of experienced people is not reliable method.

MLR Startup Profit Questions

# Questions

**1. How can this model help the company decide how much to invest in R&D, Administration, and Marketing?**

* The model provides coefficients for R&D Spend, Administration, and Marketing Spend, indicating how much each feature contributes to the profit.
* Spending 1 dollar on R&D raises profit by dollar 0.84, while administration raises by 0.02 dollar and Marketing spend raises by 0.01 dollar.
* The company can use these coefficients to optimize spending:
* Increase investment in areas with higher positive coefficients (e.g., R&D).
* Reduce spending in areas with lower or negative coefficients (e.g., Administration, as its coefficient is low).
* The coefficients also point out that we should minimize expenses on Marketing spend, as coefficient is small.
* The interactive slider allows the company to simulate different spending scenarios and predict the resulting profit.
* # Print the model coefficients
* print("Model Coefficients:")
* coefficients = pd.DataFrame(model.coef\_, X.columns, columns=['Coefficient'])
* print(coefficients)
* # Print the intercept
* intercept = model.intercept\_
* print(f"Intercept (β0): {intercept}")

Model Coefficients:

Coefficient

R&D Spend 0.845081

Administration 0.021114

Marketing Spend 0.015938

Intercept (β0): 15530.085654485796

**2. What does the mean and standard deviation of R&D Spend suggest about the dataset?**

* The mean R&D Spend of 70,543.76 indicates the average investment in R&D across all startups in the dataset. The standard deviation of 43,042.35 measures the variability in R&D Spend.
* A high standard deviation suggests significant differences in R&D investment among startups, with some investing heavily while others allocate considerably less.
* This variability highlights diverse investment strategies and potential differences in innovation priorities.
* # Calculate the mean and standard deviation of R&D Spend
* mean\_rd\_spend = df['R&D Spend'].mean()
* std\_rd\_spend = df['R&D Spend'].std()
* # Print the results
* print(f"Mean of R&D Spend: {mean\_rd\_spend:.2f}")
* print(f"Standard Deviation of R&D Spend: {std\_rd\_spend:.2f}")
* Mean of R&D Spend: 70543.76
* Standard Deviation of R&D Spend: 43042.35

**3. The Marketing Spend has a large range (difference between min and max values). How might this impact our model?**

* A large range in Marketing Spend can lead to scaling issues in the model, as features with larger ranges can dominate the model's learning process.
* It may also introduce outliers, which can skew the model's predictions. Moreover, higher range can lead to more variance reducing reliability of predictions.
* To test this, we can normalize or standardize the Marketing Spend feature and compare the model's performance.

**4. If the dataset had categorical features like Startup Industry, how would we handle them in this model?**

If the dataset includes categorical features like "Startup Industry," they can be handled using one-hot encoding or label encoding to make them suitable for the model:

One-hot encoding creates separate binary columns for each category, ensuring no ordinal relationships are introduced. This is useful for models that struggle with numerical category rankings. Label encoding assigns a unique integer to each category, which can be efficient but may introduce unintended ordinal relationships.

**5. What does the correlation matrix tell us about which spending category influences profit the most?**

The correlation matrix shows us that R&D spend has the strongest positive correlation with profit of 0.87. This means this is the most important variable to focus on to improve profitability. Marketing spend is the second most important variable has it has a light positive correlation of 0.37 with Profit. Meanwhile, administrative spend has almost no influence on profit as the correlation is very weak of 0.005 only.

**6. Why do we use train\_test\_split() before fitting the model? What is the default split ratio?**

* train\_test\_split() is used to evaluate the model's performance on unseen data.
* It splits the dataset into training and testing sets, ensuring that the model is not overfitting to the training data.
* The default split ratio being used is 80% training and 20% testing but the generally accepted default split is 75% training and 25% testing.

**7. What happens if we remove Administration spending from the model? How would it affect predictions?**

As administration has almost zero correlation with profit, removing it will not significantly affect the predictive performance of the model. It could improve the model by reducing overfitting or complexity.

To test this, we can train a model without the Administration feature and compare its performance to the original model.

We can see that by removing administration the R^2 did not change significantly and rather it improved from 0.77331 to 0.77384 as complexity was reduced.

# Split the data without Administration

X\_no\_admin = df[['R&D Spend', 'Marketing Spend']]

y\_no\_admin = df['Profit']

X\_train\_no\_admin, X\_test\_no\_admin, y\_train\_no\_admin, y\_test\_no\_admin = train\_test\_split(X\_no\_admin, y\_no\_admin, test\_size=0.2, random\_state=42)

# Train the model

model\_no\_admin = LinearRegression()

model\_no\_admin.fit(X\_train\_no\_admin, y\_train\_no\_admin)

# Evaluate the model

y\_pred\_no\_admin = model\_no\_admin.predict(X\_test\_no\_admin)

r2\_no\_admin = r2\_score(y\_test\_no\_admin, y\_pred\_no\_admin)

print(f"R-squared without Administration: {r2\_no\_admin}")

R-squared without Administration: 0.7738467615014745

# Making predictions on the test set

y\_pred = model.predict(X\_test)

# Evaluating the model

mae = mean\_absolute\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Absolute Error: {mae}")

print(f"R-squared: {r2}")

Mean Absolute Error: 20039.45507172663

R-squared: 0.7733120881784735

**8. If the test set accuracy is much lower than the training set accuracy, what might be the reason?**

This suggests overfitting, where the model excels on the training data but struggles with unseen data, leading to poor generalization.

**Potential causes:**

The model is too complex, possibly due to an excessive number of features or high-degree polynomial terms, capturing noise rather than true patterns. The training data is not representative of the test data, meaning the model has learned dataset-specific trends that do not generalize well. To address this, techniques like feature selection, regularization (e.g., Lasso or Ridge regression), and increasing training data diversity can help improve model performance on unseen data.

**9. If a new startup has zero spending in Marketing but high R&D and Administration spending, would this model still make accurate predictions?**

Yes, but with some limitations. Marketing Spend has a moderate correlation of 0.37 with Profit, meaning it has some influence but is not the strongest predictor. Since its impact is relatively weak compared to R&D Spend, having zero Marketing Spend may not drastically affect the prediction. However, if the dataset shows a pattern where all highly profitable startups have at least some marketing investment, the model might struggle to generalize accurately for such cases, leading to potential prediction errors.

**10. The company wants to maximize profit. Should they focus more on increasing R&D spending or Marketing spending based on the model’s results?**

The company should focus on R&D spending as it has the higher positive correlation and the strongest positive coefficient with respect to a unit increase contribution to profit. (0.87 correlation and 0.845 coefficient).

**11. Could this model be used to predict the profit of a startup in a different country? Why or why not?**

No the predictions will not be reliable. The model is trained on data from a specific market, and factors such as cost structures, economic conditions, industry trends, and regulations can vary significantly across countries. These differences may cause the model to make inaccurate predictions if applied to a different region.

Limited generalization: The model may not adapt well to new environments where business dynamics, consumer behavior, and government policies differ. For instance, labor costs, tax structures, and marketing effectiveness can vary widely, affecting profitability.

Solution: To improve accuracy, additional data from the target country would be needed to retrain or fine-tune the model, ensuring it captures local market conditions and economic realities.

**12. If you were advising this startup, what additional variables would you suggest adding to improve predictions?**

* Additional variables could include:
* Geographic location (e.g., country, city).
* Industry sector (e.g., tech, healthcare).
* Company size (e.g., number of employees).
* Economic indicators (e.g., GDP, inflation rate).

Student Performance Regression Questions

# Questions

**1. What role do the coefficients (slopes) play in Linear Regression? How would you interpret them in the context of our dataset?**

**Role of Coefficients in Linear Regression**

* Coefficients indicate how much the Performance Index changes with a one-unit increase in each predictor while keeping others constant:
* Hours Studied: +2.853 → each additional hour studied increases the performance index by 2.853 points.
* Previous Scores: +1.018 → A one-point increase in previous scores increases performance index by 1.018.
* Extracurricular Activities: +0.613 → Students engaged in extracurricular score 0.613 points higher on average.
* Sleep Hours: +0.481 → each additional sleep hour increases performance index by 0.481.
* Sample Question Papers Practiced: +0.194 → Practicing one extra sample paper increases the index by 0.194.

**2. What are the independent (predictor) and dependent (target) variables in this dataset?**

* Predictors (Independent) : Hours Studied, Previous Scores, Extracurricular Activities, Sleep Hours, Sample Question Papers Practiced
* Target Variable (Dependent) : Performance Index

**3. What happens when we increase the number of features (independent variables) in a Linear Regression model? How does it affect accuracy?**

* More features generally improve accuracy if they are relevant.
* The R-squared value (0.989) suggests that our model already explains 98.9% of the variance, meaning additional features may not improve accuracy significantly.
* Adding irrelevant features can cause overfitting.

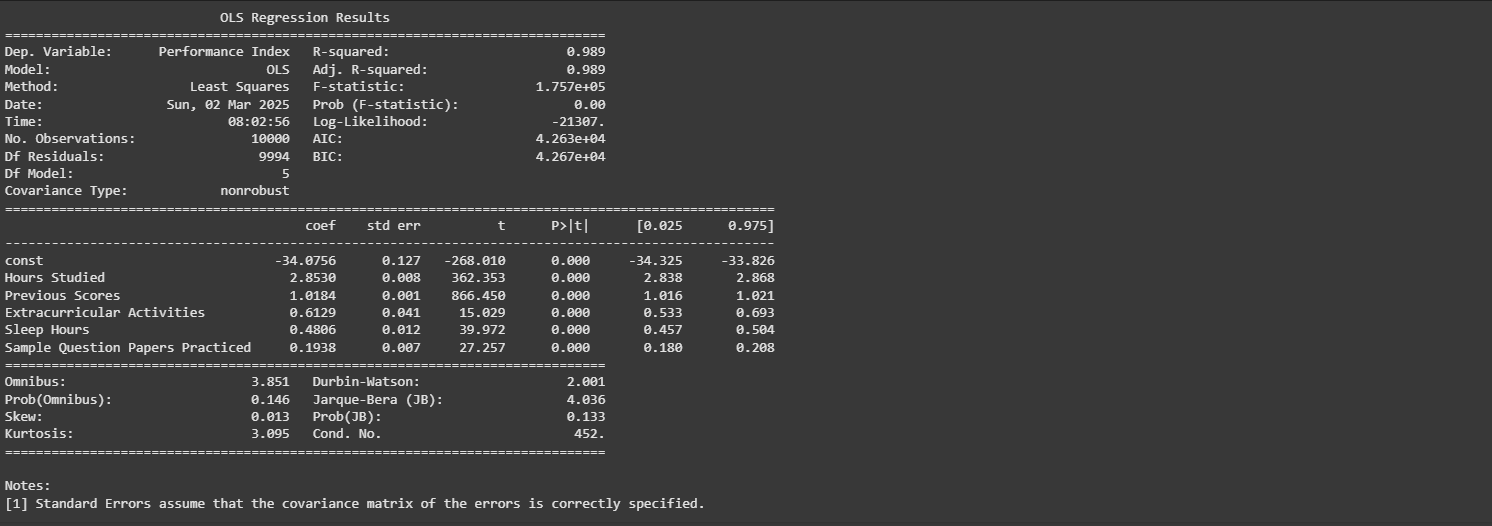
**4. What does the output summary of the model tell us about the statistical significance of each feature?**

From the model summary code below we can see that:

* The p-values of all predictors are <0.05, meaning they are statistically significant and contribute meaningfully to the model.
* F-statistic (1.757e+05) and p-value (0.00) confirm that the model is a good fit.

# Print the model summary

print(model.summary())



**5. If we observe that the model has high training accuracy but low testing accuracy, what does that indicate?**

If the model has high training accuracy but low testing accuracy, it indicates that the model is overfitting to the training data.

Overfitting occurs when the model learns the noise or specific patterns in the training data too well, to the point where it performs poorly on unseen data (the testing set).

Possible fixes:

* Reduce the number of predictors.
* Use regularization techniques like Ridge/Lasso Regression.
* Increase the size of the training dataset.