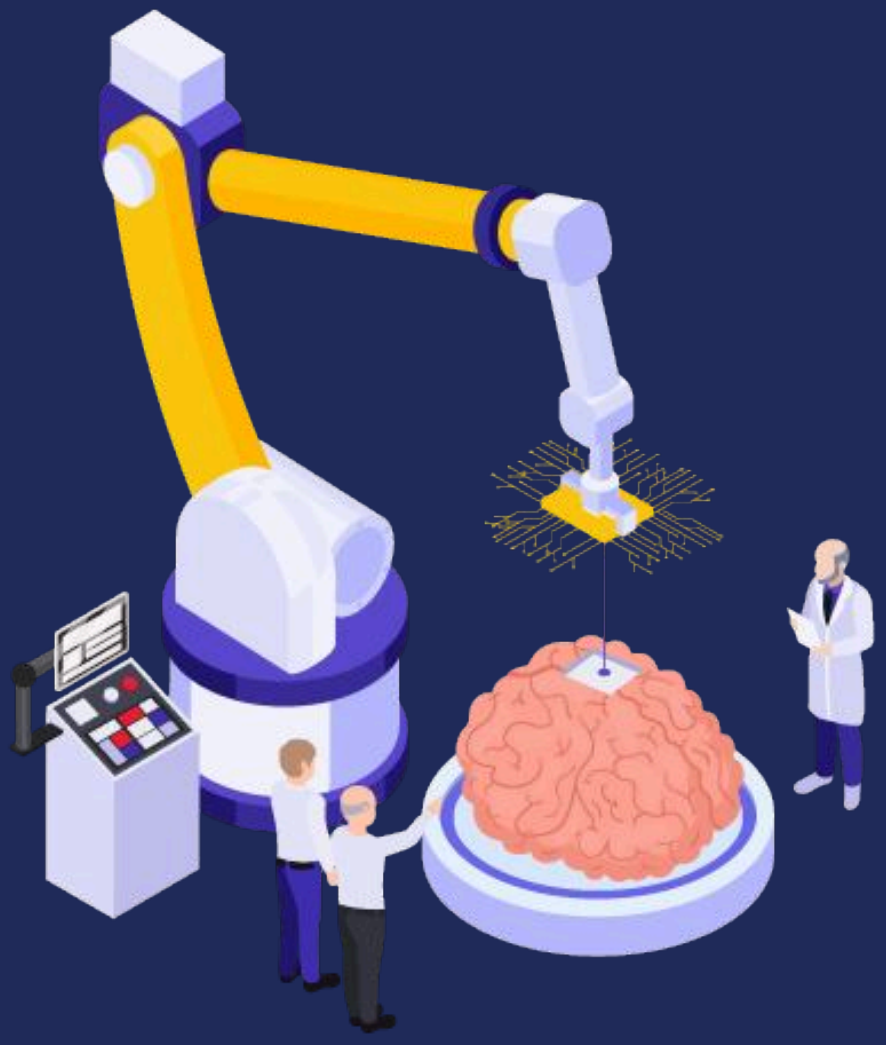
Evaluation Metrics and Regression Implementation

Assignment Questions



PW SKILLS

Theoretical:

1. What does R-squared represent in a regression model?
2. What are the assumptions of linear regression?
3. What is the difference between R-squared and Adjusted R-squared?
4. Why do we use Mean Squared Error (MSE)?
5. What does an Adjusted R-squared value of 0.85 indicate?
6. How do we check for normality of residuals in linear regression?
7. What is multicollinearity, and how does it impact regression?
8. What is Mean Absolute Error (MAE)?
9. What are the benefits of using an ML pipeline?
10. Why is RMSE considered more interpretable than MSE?
11. What is pickling in Python, and how is it useful in ML?
12. What does a high R-squared value mean?
13. What happens if linear regression assumptions are violated?
14. How can we address multicollinearity in regression?
15. How can feature selection improve model performance in regression analysis?
16. How is Adjusted R-squared calculated?
17. Why is MSE sensitive to outliers?
18. What is the role of homoscedasticity in linear regression?
19. What is Root Mean Squared Error (RMSE)?
20. Why is pickling considered risky?
21. What alternatives exist to pickling for saving ML models?
22. What is heteroscedasticity, and why is it a problem?
23. How can interaction terms enhance a regression model's predictive power?

Practical:

1. Write a Python script to visualize the distribution of errors (residuals) for a multiple linear regression model using Seaborn's "diamonds" dataset.

2. Write a Python script to calculate and print Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) for a linear regression model.

3. Write a Python script to check if the assumptions of linear regression are met. Use a scatter plot to check linearity, residuals plot for homoscedasticity, and correlation matrix for multicollinearity.

4. Write a Python script that creates a machine learning pipeline with feature scaling and evaluates the performance of different regression models

5. Implement a simple linear regression model on a dataset and print the model's coefficients, intercept, and R-squared score.

6. Write a Python script that analyzes the relationship between total bill and tip in the 'tips' dataset using simple linear regression and visualizes the results.

7. Write a Python script that fits a linear regression model to a synthetic dataset with one feature. Use the model to predict new values and plot the data points along with the regression line.

8. Write a Python script that pickles a trained linear regression model and saves it to a file.

9. Write a Python script that fits a polynomial regression model (degree 2) to a dataset and plots the regression curve.

10. Generate synthetic data for simple linear regression (use random values for X and y) and fit a linear regression model to the data. Print the model's coefficient and intercept.

11. Write a Python script that fits polynomial regression models of different degrees to a synthetic dataset and compares their performance.

12. Write a Python script that fits a simple linear regression model with two features and prints the model's coefficients, intercept, and R-squared score.

13. Write a Python script that generates synthetic data, fits a linear regression model, and visualizes the regression line along with the data points.

14. Write a Python script that uses the Variance Inflation Factor (VIF) to check for multicollinearity in a dataset with multiple features.

15. Write a Python script that generates synthetic data for a polynomial relationship (degree 4), fits a polynomial regression model, and plots the regression curve.

16. Write a Python script that creates a machine learning pipeline with data standardization and a multiple linear regression model, and prints the R-squared score.

17. Write a Python script that performs polynomial regression (degree 3) on a synthetic dataset and plots the regression curve.

18. Write a Python script that performs multiple linear regression on a synthetic dataset with 5 features. Print the R-squared score and model coefficients.

19. Write a Python script that generates synthetic data for linear regression, fits a model, and visualizes the data points along with the regression line.

20. Create a synthetic dataset with 3 features and perform multiple linear regression. Print the model's R-squared score and coefficients.

21. Write a Python script that demonstrates how to serialize and deserialize machine learning models using joblib instead of pickling.

22. Write a Python script to perform linear regression with categorical features using one-hot encoding. Use the Seaborn 'tips' dataset.

23. Compare Ridge Regression with Linear Regression on a synthetic dataset and print the coefficients and R-squared score.

24. Write a Python script that uses cross-validation to evaluate a Linear Regression model on a synthetic dataset.

25. Write a Python script that compares polynomial regression models of different degrees and prints the R-squared score for each.