**🎯 Objective:**

Develop a machine learning solution to predict the **Estimated\_Annual\_Salary** of a customer using structured data.

**🪜 Step-by-Step Instructions**

**1. Study the Data Dictionary**

* Read the file: ./synthetic\_regression\_data\_dictionary.docx
* Understand the role, type, and description of each variable
* Identify the target variable: Estimated\_Annual\_Salary
* Log summary of variables (numerical vs. categorical) into ./logs/data\_overview.txt

**2. Import and Inspect the Dataset**

* Load dataset from ./synthetic\_regression\_dataset.csv
* Report:
  + Shape (rows, columns)
  + Data types
  + Missing values
  + Unique values per column
* Save this report to: ./eda/data\_inspection.docx and data\_inspection.pdf

**3. Exploratory Data Analysis (EDA)**

Perform both **univariate and bivariate analysis**:

* **Univariate** (each variable alone): Histograms, boxplots, value counts
* **Bivariate** (vs. target): Scatterplots, boxplots by category, correlation heatmap

For each type:

* Include both descriptive statistics and visual plots
* Use labeled titles and legends

Save output to:  
📄 ./eda/univariate\_analysis.docx, ./eda/bivariate\_analysis.docx  
📄 ./eda/univariate\_analysis.pdf, ./eda/bivariate\_analysis.pdf

**4. Feature Engineering**

Handle features based on type:

* **Categorical Variables:**
  + Apply One-Hot or Ordinal Encoding as appropriate
  + Document encoding logic
* **Continuous Variables:**
  + Scale features using StandardScaler or MinMaxScaler
  + Create meaningful transformations if needed (e.g., log, binning)

Document all transformations in:  
📄 ./feature-engineering/feature\_transformation.docx and feature\_transformation.pdf  
Include tables showing original vs. transformed features.

**5. Data Splitting**

* Split data into **train (80%)** and **test (20%)** using stratification if necessary
* Save datasets as:
  + ./model-data/train.csv
  + ./model-data/test.csv

**6. Build and Evaluate Linear Regression Model**

* Train a linear regression model using training data
* Evaluate on test data using:
  + MAE, MSE, RMSE, R² Score
* Save:
  + Residual plots
  + Actual vs. Predicted plots
  + Model coefficients table

Save report to:  
📄 ./model-performance/linear\_regression\_analysis.docx and .pdf

**7. Build and Evaluate XGBoost Regression Model**

* Train an XGBoost model on same train/test split
* Use the same evaluation metrics and visualizations
* Include feature importance chart

Save report to:  
📄 ./model-performance/xgboost\_analysis.docx and .pdf

**8. Model Comparison**

Compare both models across:

| **Metric** | **Linear Regression** | **XGBoost** |
| --- | --- | --- |
| MAE, MSE, RMSE | ✅ | ✅ |
| R² Score | ✅ | ✅ |
| MAPE | ✅ | ✅ |
| Fairness Insights | Bias in prediction across Education\_Level, City\_Tier |  |
| Explainability | Coefficients (Linear) vs Feature Importance (XGBoost) |  |

Summarize findings in:  
📄 ./model-performance/model\_comparison.docx and .pdf

**9. Save Best Model and Scoring Script**

* Based on accuracy and interpretability, select the best model
* Save model as .pkl to: ./model/best\_model.pkl
* Write a score.py script that:
  + Accepts new input as a CSV
  + Loads model and preprocessing pipeline
  + Outputs predictions to a CSV

Save scoring script in:  
📄 ./model/score.py