Capstone - Design

Data Wrangling

* Ensure that the data is clean and free from any missing or incorrect entries.
* Inspect the data manually to identify missing or incorrect information using the functions isna() and notna().
* Based on your knowledge of data analytics, include your recommendations for treating missing and incorrect data (dropping the null values or filling them).
* Choose a suitable data wrangling technique—either data standardization or normalization. Execute the preferred normalization method and present the resulting data. (Normalization is the preferred approach for this problem.)
* Share your insights regarding the application of the GroupBy() function for either data chunking or merging, and offer a recommendation based on your analysis.

Data Analysis

* Perform descriptive statistical analysis on the data in the Sales and Unit columns. Utilize techniques such as mean, median, mode, and standard deviation for this analysis.
* Identify the group with the highest sales and the group with the lowest sales based on the data provided.
* Generate weekly, monthly, and quarterly reports to document and present the results of the analysis conducted.  
  (Use suitable libraries such as NumPy, Pandas, and SciPy for performing the analysis.)

Data Visualization

* Use suitable data visualization libraries to construct a dashboard for the head of sales and marketing. The dashboard should encompass key parameters:
* State-wise sales analysis for different demographic groups (kids, women, men, and seniors).
* Group-wise sales analysis (Kids, Women, Men, and Seniors) across various states.
* Time-of-the-day analysis: Identify peak and off-peak sales periods to facilitate strategic planning for S&M teams. This information aids in designing programs like hyper-personalization and Next Best Offers to enhance sales.
* Ensure the visualization is clear and accessible for effective decision-making by the head of sales and marketing (S&M).
* The dashboard must contain daily, weekly, monthly, and quarterly charts.
* (Any visualization library can be used for this purpose. However, since statistical analysis is being done, Seaborn is preferred.)
* Include your recommendation and indicate why you are choosing the recommended visualization package.

Handle the left Class Imbalance using the SMOTE technique.

* Pre-process the data by converting categorical columns to numerical columns by:
* Separating categorical variables and numeric variables
* Applying get\_dummies() to the categorical variables
* Combining categorical variables and numeric variables
* Do the stratified split of the dataset to train and test in the ratio 80:20 with random\_state=123.
* Upsample the train dataset using the SMOTE technique from the imblearn module.

Perform 5-fold cross-validation model training and evaluate performance.

* Train a logistic regression model, apply a 5-fold CV, and plot the classification report.
* Train a Random Forest Classifier model, apply the 5-fold CV, and plot the classification report.
* Train a Gradient Boosting Classifier model, apply the 5-fold CV, and plot the classification report.

Identify the best model and justify the evaluation metrics used.

* Find the ROC/AUC for each model and plot the ROC curve.
* Find the confusion matrix for each of the models.
* Explain which metric needs to be used from the confusion matrix: Recall or Precision?

Deep Learning models

* CNN
* LSTM

RAG model

Prompt Engineering

Upload csv

Sending email

Agentic workflow for LLM

sales\_dashboard/

├── backend/

│ ├── main.py # FastAPI app

│ ├── routers/

│ │ ├── prediction.py # ML/DL endpoints

│ │ ├── rag.py # Q&A from CSV

│ │ └── charts.py # Plotly data

│ ├── rag\_engine.py

│ ├── ml\_models/

│ └── data/

│ ├── train.csv

│ └── test.csv

├── preprocessing/

│ ├── feature\_engineering.py

│ └── data\_loader.py

├── training/

│ ├── train\_lgbm.py

│ └── train\_nn.py

├── frontend/

│ └── index.html # The HTML file above