Interview Presentation for PyQ1

- **Overview**: “PyQ1 required merging sales, TV, and video advertising data to prepare for analysis. I used pandas for merging and wrote an equivalent SQL query.”
- **Key Steps**:
  - “I created three DataFrames: sales_data (13 weeks), tv_data (5 weeks), and video_data (7 weeks).”
  - “I merged them using a left join on Timestamp to retain all sales records, resulting in NaN for weeks without marketing data.”
  - “The SQL query replicates this, showing my ability to translate Python operations to SQL.”
- **Key Output**: “The merged DataFrame shows sales peaking at 431,400 on September 15, 2021, with sparse TV and video data, indicating selective marketing campaigns.”
- **Business Insight**: “The merged data can be used to analyze how TV GRPs and video impressions drive sales. I’d impute missing values or focus on matched weeks for modeling.”
- **Potential Improvements**: “I could impute NaN values using averages or interpolate based on trends. Adding visualizations, like sales vs. TV GRPs, would clarify marketing impact.”

Anticipated Questions

- **Q**: Why use a left join?
  A: “To keep all sales records, as sales is the primary dataset, and marketing data is supplementary. This ensures no sales data is lost.”
- **Q**: How would you handle NaN values?
  A: “I’d impute with mean/median for continuous variables like TV GRPs or use time-based interpolation to reflect trends.”
- **Q**: Why is the merge incomplete (only tv_data)?
  A: “The code has a typo (sales_dataRange), but I executed a second merge with video_data, as shown in the output. I’d ensure both merges are explicit in the code.”

Interview Presentation for PyQ2

“I used pandas and numpy for data handling, and seaborn/matplotlib for visualizations to explore sales drivers.”

“I loaded the dataset with 122 weeks of data for Brand A, including sales, marketing channels, discounts, and external factors like gas prices. The preview helped verify data integrity.”

“I dropped missing values to ensure model stability, but I’d explore imputation (e.g., mean or interpolation) if the dataset lost significant data.”

“The heatmap identified key sales drivers like Organic Search Impressions (0.56 correlation), guiding my feature engineering to focus on high-impact variables.”

“Organic and Paid Search Impressions had the highest correlations (~0.55), indicating they strongly influence sales, which informed my modeling focus.”

“Surprisingly, sales dropped 30% during holidays, likely due to consumer behavior shifts, prompting me to include holiday dummies in the model.”

“Scatter plots confirmed search impressions drive sales, while discounts and email clicks showed weaker relationships, guiding feature prioritization.”

“EDA revealed sales peaks tied to discounts and gas price changes, with no holiday effects in early 2022. This informed my feature engineering, like adstock and seasonality.”

“I imported sklearn for Ridge Regression and metrics, and statsmodels for p-values and VIF to assess feature significance and multicollinearity.”

“I applied log transformations to sales and marketing variables to handle skewness, improving model performance.”

“I used adstock with a 0.3 decay rate to capture the delayed effects of Paid Social, Paid Search, and Modular Video ads, reflecting real-world advertising dynamics.”

“I added lagged variables to model the delayed impact of marketing, filling initial missing values with 0 to avoid data loss.”

“I used a sine function based on week numbers to model yearly seasonality, as sales may vary cyclically, especially around holidays.”

“I standardized gas prices to ensure fair comparison with other features, as its scale differs significantly.”

"I added an interaction term to model the synergy between discounts and Paid Social, as their combined effect may amplify sales."

"I selected features based on EDA, including log-transformed variables, adstock for ads, and interaction terms to capture complex relationships."

"I used an 80-20 split to balance training data and test evaluation, with a random state for reproducibility."

"I standardized features to prevent variables with larger scales, like impressions, from dominating the model."

: "I chose Ridge Regression to address multicollinearity, common in marketing data, with alpha=1.0 for balanced regularization."

"I generated predictions to evaluate model performance on both training and unseen test data."

"The model fits training data well (R^2=0.78), but the negative test R^2 indicates overfitting, likely due to holiday effects or multicollinearity. The low MAPE suggests predictions are relatively accurate in percentage terms."

"The OLS summary confirmed Paid Search and Organic Search as significant drivers ($p<0.001$), while holidays negatively impact sales. High VIF for discounts suggests multicollinearity issues."

"High VIF for discount features confirms multicollinearity, which Ridge Regression mitigates, but I'd consider feature selection to improve the model."

"This plot shows the model struggles to predict sales during holiday weeks, contributing to the negative test R^2. I'd add holiday-specific features to improve fit."

"The feature importance plot highlights Paid Search and Organic Search as top drivers, guiding my recommendation to prioritize these channels."

"I recommended allocating more budget to search ads, pairing discounts with social media, and boosting pre-holiday campaigns to offset sales dips, using tools like Google Ads for efficiency."

- **Overview**: "PyQ2 involved building a Marketing Mix Model to identify sales drivers and optimize marketing spend over 122 weeks."
- **Key Steps**:

    - "I conducted EDA, finding strong correlations with search impressions and a 30% sales drop during holidays."
    - "Feature engineering included log transformations, adstock for ad carryover, and seasonality to capture trends."
    - "I used Ridge Regression to handle multicollinearity, achieving a training R^2 of 0.78 but a negative test R^2 due to holiday effects."
    - "Charts like the actual vs. predicted plot highlighted overfitting issues."
- **Key Output**: "Paid Search and Organic Search were top drivers, with holidays reducing sales by 30%. The model suggests focusing on search ads and pre-holiday promotions."
- **Business Insight**: "Allocate 40-50% of the budget to search ads, pair discounts with social media, and simplify product offerings to boost sales."
- **Potential Improvements**: "I'd add holiday-specific features, use cross-validation, and remove high-VIF features like Discount1 to improve test performance."

Anticipated Questions

- **Q**: Why the negative test R^2?
  A: "Overfitting due to holiday effects and multicollinearity in discount features. I'd add holiday flags and use feature selection to improve generalization."
- **Q**: Why Ridge Regression?
  A: "It handles multicollinearity, common in marketing data, as shown by high VIF for discounts."
- **Q**: How would you optimize the budget?
  A: "Prioritize Paid and Organic Search, pair discounts with Paid Social, and use tools like Google Ads for real-time optimization."

Interview Presentation for PyQ3

"I bucketed ages to analyze salary trends by age group, using pd.cut for clear categorization."

"I confirmed age bucketing worked, with Employee_1 correctly assigned to '40-50 years'."

"I merged department codes to standardize functions, dropping redundant columns for clarity."

Explanation: Uses the IQR method to flag salaries outside 1.5*IQR from Q1/Q3 as outliers (e.g., 120,000 is an outlier).

Interview Talking Points: "I used the IQR method to identify salary outliers, flagging high salaries like 120,000 for further review."

- **Overview**: "PyQ3 involved processing employee data to standardize department codes, bucket ages, and identify salary outliers."
- **Key Steps**:
  - "I created a DataFrame with 30 employees' salaries, ages, and functions."
  - "I mapped departments to codes (e.g., HR → HR001) and bucketed ages into five groups."
  - "I used the IQR method to flag outliers, identifying high salaries like 120,000."
- **Key Output**: "The final DataFrame includes function codes, age buckets, and outlier flags, with Employee_3's 120,000 salary marked as an outlier."
- **Business Insight**: "This data can help HR analyze salary distributions, identify anomalies, and standardize department reporting."
- **Potential Improvements**: "I could add visualizations, like a boxplot for salaries, or analyze outliers by department."

Anticipated Questions

- **Q**: Why use IQR for outliers?
  A: "IQR is robust for detecting outliers in non-normal data, identifying extreme salaries like 120,000."

- **Q**: Why bucket ages?
  A: “Bucketing simplifies age analysis, allowing HR to compare salary trends across age groups.”
- **Q**: How would you handle outliers?
  A: “I’d investigate outliers to check for data errors or justify high salaries (e.g., Manager roles).”