Brazilian E-commerce Data Mining Project - Team Action Plan

Project Timeline

April 18 - May 2, 2025 (14 days)

Team Members

- Tanishk Singh
- Kritvirya Singh
- Kanit Mann
- Umesh Kumar Siyak

Project Goals

1. Analyze factors influencing delivery performance
2. Segment customers based on purchasing behaviors
3. Develop time series models for sales forecasting

Phase 1: Setup and Data Preparation (4 days: April 18-21)

Day 1-2: Environment Setup and Dataset Understanding

Team Member Assignment: ALL

1. Set up a shared GitHub repository for version control
2. Create a project environment with necessary Python libraries:

```python
# Required libraries
# - pandas, numpy (data manipulation)
# - matplotlib, seaborn (visualization)
# - scikit-learn (machine learning)
# - statsmodels (time series analysis)
# - mlxtend (for association rule mining)
```

3. Download and understand all datasets:
   - Map out how the 9 CSV files relate to each other
   - Create an entity-relationship diagram
   - Document the meaning of each column in each dataset

4. Divide the datasets among team members for initial exploration:
    - **Tanishk**: orders, order_items
    - **Kritvirya**: customers, geolocation
    - **Kanit**: products, category_name translation
    - **Umesh**: sellers, payments, order_reviews

Day 3-4: Data Cleaning and Integration

Team Member Assignment: ALL

1. Clean individual datasets:

    ```
    python

    # Common data cleaning operations
    # - Handle missing values
    # - Convert date columns to datetime
    # - Check for and handle outliers
    # - Validate data types
    ```

2. Integrate datasets:

    ```
    python

    # Example of joining datasets (actual implementation will be more complex)
    # orders_customers = pd.merge(orders_df, customers_df, on='customer_id')
    # orders_items = pd.merge(orders_df, order_items_df, on='order_id')
    ```

3. Create a unified dataset for each research question

Phase 2: Exploratory Data Analysis (3 days: April 22-24)

Day 5: Basic Statistical Analysis

Team Member Assignment: Tanishk & Kritvirya

1. Calculate basic statistics for key variables
2. Visualize distributions of important features
3. Identify patterns in order volumes, delivery times, and customer behavior

Day 6: Correlation Analysis

Team Member Assignment: Kanit & Umesh

1. Examine relationships between variables:

```python
# Example correlation analysis
# correlation_matrix = df.corr()
# sns.heatmap(correlation_matrix, annot=True)
```

2. Perform chi-square tests for categorical variables
3. Document initial findings that address research questions

Day 7: Advanced EDA and Feature Engineering

Team Member Assignment: ALL

1. Engineer features for each research question:

```python
# Example feature engineering for delivery performance
# orders_df['delivery_time'] = (pd.to_datetime(orders_df['order_delivered_customer_date'
#                               pd.to_datetime(orders_df['order_purchase_timestamp'])).dt
#
# orders_df['delay'] = (pd.to_datetime(orders_df['order_delivered_customer_date']) -
#                       pd.to_datetime(orders_df['order_estimated_delivery_date'])).dt.d
```

2. Create new datasets with engineered features
3. Team meeting to review findings and adjust strategies if needed

Phase 3: Modeling and Analysis (5 days: April 25-29)

Day 8-9: Delivery Performance Analysis

Team Member Assignment: Tanishk & Kritvirya

1. Prepare features that might influence delivery time:
   - Product attributes (weight, dimensions, category)
   - Seller location
   - Customer location
   - Order attributes (payment type, order value)
2. Build predictive models:

python

```
# Example modeling approach (multiple models should be tested)
# from sklearn.ensemble import RandomForestRegressor
# X = delivery_df[['distance_km', 'product_weight_g', 'payment_value', ...]]
# y = delivery_df['delivery_time']
# model = RandomForestRegressor()
# model.fit(X, y)
```

3. Evaluate model performance and identify key factors

Day 10-11: Customer Segmentation

Team Member Assignment: Kanit & Umesh

1. Prepare customer features:

python

```
# Calculate customer metrics
# customer_metrics = orders_df.groupby('customer_id').agg({
#     'order_id': 'count',  # Purchase frequency
#     'payment_value': ['sum', 'mean'],  # Total spend, average order value
#     'delivery_time': 'mean'  # Average delivery time
# })
```

2. Apply K-means clustering:

python

```
# Example K-means implementation
# from sklearn.cluster import KMeans
# from sklearn.preprocessing import StandardScaler
#
# scaler = StandardScaler()
# scaled_features = scaler.fit_transform(customer_features)
#
# kmeans = KMeans(n_clusters=4, random_state=42)
# clusters = kmeans.fit_predict(scaled_features)
```

3. Apply the Apriori algorithm for product association analysis within segments:

```python
# Example using mlxtend
# from mlxtend.frequent_patterns import apriori, association_rules
#
# # For each cluster, find frequent itemsets
# frequent_itemsets = apriori(transaction_data, min_support=0.01, use_colnames=True)
# rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)
```

4. Characterize the identified customer segments

Day 12: Time Series Analysis

Team Member Assignment: ALL

1. Prepare time series data:

```python
# Aggregate orders by day/week/month
# daily_orders = orders_df.resample('D', on='order_purchase_timestamp').agg({'order_id':
# monthly_orders = orders_df.resample('M', on='order_purchase_timestamp').agg({'order_id
```

2. Analyze seasonality and trends
3. Build ARIMA/SARIMA models:

```python
# Example time series modeling with statsmodels
# from statsmodels.tsa.statespace.sarimax import SARIMAX
#
# model = SARIMAX(time_series_data, order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
# results = model.fit()
```

4. Evaluate forecast accuracy

Phase 4: Results Compilation and Report Writing (3 days: April 30-May 2)

Day 13: Finalizing Analyses and Visualizations

Team Member Assignment: ALL

1. Create final visualizations for each research question:
   - Delivery performance factors visualization
   - Customer segment profiles and geographical distribution

- Time series forecasts and seasonal patterns

2. Prepare result tables and summary statistics

Day 14: Report Writing and Submission

Team Member Assignment: ALL

1. Write the final report with the following structure:
   - Introduction and problem statement
   - Dataset description
   - Methodology
   - Results and discussion for each research question
   - Conclusions and business recommendations
   - References
   - Appendix with code snippets and additional visualizations
2. Review report for clarity and completeness
3. Submit final project by May 2nd deadline

Task Division Strategy

Tanishk

- Lead on delivery performance analysis
- Contribute to time series forecasting
- Coordinate GitHub repository maintenance

Kritvirya

- Support delivery performance analysis
- Lead data cleaning and integration
- Contribute to report writing

Kanit

- Lead customer segmentation
- Support product association analysis
- Create customer segment visualizations

Umesh

- Lead product association analysis
- Support customer segmentation
- Contribute to time series forecasting

Recommended Tools and Libraries

Data Handling

- pandas (data manipulation)
- numpy (numerical operations)

Visualization

- matplotlib (basic plotting)
- seaborn (statistical visualizations)
- plotly (interactive visualizations)

Machine Learning

- scikit-learn (general ML algorithms)
- mlxtend (association rules mining)

Time Series Analysis

- statsmodels (ARIMA/SARIMA models)
- prophet (alternative forecasting tool)

Development

- Jupyter Notebooks (exploratory analysis)
- GitHub (version control)
- Google Colab or AWS (if larger computing resources needed)

Best Practices

1. **Code Documentation**: Comment code thoroughly and maintain a consistent style
2. **Regular Commits**: Commit changes to GitHub frequently with clear messages
3. **Peer Review**: Review each other's code and analysis
4. **Daily Check-ins**: Brief team meetings to discuss progress and challenges
5. **Backup Data**: Maintain backup copies of processed datasets

6. **Modular Code**: Create reusable functions for common operations
7. **Progress Tracking**: Maintain a shared document to track progress on tasks

Expected Challenges and Solutions

1. **Data Quality Issues**:
   - Solution: Implement robust data validation and cleaning procedures
2. **Computational Limitations**:
   - Solution: Use sampling techniques or cloud computing resources
3. **Feature Selection**:
   - Solution: Use domain knowledge and statistical tests to identify relevant features
4. **Model Selection**:
   - Solution: Test multiple models and use cross-validation
5. **Time Management**:
   - Solution: Stick to the timeline and adjust scope if necessary

Good luck with your project!