**Python project workflow**

1. Data Preparation and Initial Analysis:
   * Loading and Inspection: Import necessary libraries such as pandas, numpy, seaborn, etc., and load the product dataset. Utilize info() and head() functions to gain an initial understanding of the dataset's structure and content.
   * Handling Missing Values: Identify and address missing values within the dataset, ensuring the data's integrity and usability for further analysis.
   * Date Indexing: Convert the 'date' column to datetime format and set it as the index of the DataFrame. This step is crucial for time series analysis, enabling easier slicing and manipulation of data based on time.
2. Extensive Exploratory Data Analysis (EDA):
   * Sales Trends Analysis: Aggregate sales data and plot overall sales trends over time to understand the general sales behavior and identify any obvious patterns or anomalies.
   * Sales Distribution and Comparison: Analyze the distribution of sales using histograms and compare sales across different stores and product types with box plots. This helps in identifying outliers, variations, and other key insights.
   * Statistical Summary: Provide a statistical summary of sales, such as mean, median, and standard deviation, to understand the central tendency and dispersion of sales data.
3. Segmentation and Time Series Analysis:
   * Creating Segments: Segment the data by unique combinations of store numbers and product types, storing each segment in a dictionary for easy access and individual analysis.
   * Visual Time Series Analysis: Plot individual time series data for selected store-product combinations to visually inspect trends, seasonality, and other time series characteristics.
   * Statistical Testing for Stationarity: Perform the Augmented Dickey-Fuller test on each time series segment to check for stationarity. Segments are categorized based on whether they exhibit zero sales, non-stationary, or stationary characteristics. The purpose of implement this strategy aims to build appropriate forecasting model due to the investigation of time series feature from our data set.

\* Eventually, our group decide to only split zero-sales and non-zero-sales sub-data set.

* + **Grouping by Both Store and Product**

**Perks**:

* **Tailored Insights**: This approach provides explicit insights into sales patterns for each product within each store. It allows for a more accurate understanding of consumer preferences and product performance at a specific location, which is tailored to the main task.
* **Localized Strategy Development**: By analyzing sales data at the store-product level, businesses can develop localized marketing and inventory strategies, catering to the unique demands of customers at each store in the future.

**Drawbacks**:

* **Complexity and Resource Intensity**: Analyzing data at this specific level can be complex and computationally intensive, especially for businesses with a large number of stores and products.
* **Risk of Overfitting in Models**: When creating predictive models, there's a higher risk of overfitting due to the highly specific nature of the data.
* **Data Sparsity Issues**: Some store-product combinations might have limited data, leading to challenges in making reliable forecasts.
* **Grouping Solely By Store**

**Perks**:

* **Simplified Analysis**: Analyzing data at the store level simplifies the analysis process, making it less resource-intensive.
* **Store Performance Evaluation**: This approach is effective for assessing the overall performance of a store and understanding region-based trends.
* **Resource Allocation**: Helps in strategic decision-making regarding resource allocation and management at the store level.

**Drawbacks**:

* **Loss of Product-Specific Insights**: It does not provide insights into how individual products are performing within each store.
* **Inability to Tailor Product Strategies**: Lacks the granularity needed for product-specific strategies like inventory management for particular products.
* **Grouping Solely by Product**

**Perks**:

* **Product Performance Analysis**: Enables a comprehensive analysis of a product's performance across all stores, useful for overall product strategy.
* **Standardization of Strategies**: Facilitates the development of standardized marketing and production strategies for each product.
* **Demand Forecasting**: Useful for forecasting overall demand for products irrespective of store locations.

**Drawbacks**:

* **Ignoring Regional Variations**: This method overlooks regional variations in sales, which can be critical in a retail setting.
* **Lack of Localized Insights**: Does not provide insights into local consumer preferences or store-specific challenges.

1. Implementation of a Forecasting Class with Model Optimization and Prediction:
   * Class Definition and Feature Engineering: Define the StationarySalesForecaster class, capable of creating lag features essential for time series forecasting models.
   * Model Training and Optimization: Implement functions within the class to train and optimize various machine learning models (Linear Regression, Random Forest, LightGBM, XGBoost) for sales forecasting. This includes hyperparameter tuning using techniques like grid search.
   * Forecasting and Performance Evaluation: Develop functions for making sales predictions and evaluating the models' performance using metrics like Mean Squared Error, R2 Score, and Mean Absolute Error. The class also explores rolling forecasts with XGBoost for sequential day-by-day prediction.
2. Building the model
3. Detect the scale of the sales unit
4. Using different categories of models to do the forecasting
5. Store the result form the prediction, including the actual values, predicted values, score , and graphs.

Train-Validation splitting portion.

2013-01-01 = 365

2014-01-01 = 365

2015-01-01 = 365

2016-07-31 = 180

2016-08-01 ~ 2017-07-30 = 365