Challenge Overview

Demand forecasting is essential for effective supply chain management, ensuring optimal inventory levels while minimizing waste and stockouts. However, predicting future demand is complex due to factors like seasonality, pricing changes, promotions, and store-specific trends. The M5 dataset presents an additional challenge with its high-dimensional structure—over 30,000 unique products sold across multiple stores in different states over five years. Given this complexity, a structured, data-driven approach is crucial to developing an accurate and scalable forecasting solution.

Approach

To address this challenge, I followed a systematic methodology:

1. **Exploratory Data Analysis (EDA):** I examined historical sales trends, missing values, seasonality effects, and external influences (e.g., holidays and promotions) to gain insights into demand patterns.

2. **Feature Engineering:** I generated additional features such as rolling sales averages, lag-based features, price elasticity indicators, and event-based variables to improve predictive accuracy.

3. **Model Selection & Testing:** I experimented with multiple models, including statistical methods (Prophet), machine learning models (XGBoost, LightGBM), and ensemble approaches to compare performance.

4. **Forecast Generation:** The best-performing model was used to generate 28-day forecasts for each product at each store.

5. **Evaluation & Optimization:** Forecast accuracy was assessed using **MAE (Mean Absolute Error) and R^2 (Coefficient of Determination).** These metrics provided insights into the precision, stability, and business impact of the model.

Results and Discussion

- **Sales Growth & Trends:**
    - California has the highest sales volume, followed by Texas and Wisconsin.

    - FOODS category dominates, followed by HOUSEHOLD and then HOBBIES.
    - Sales patterns are consistent across all states.

- **Exploratory Insights:**
    - Correlation matrix helped identify key relationships between features.
    - Weekly sales show stability despite occasional outliers.
    - Clear seasonality detected in monthly sales trends.

- **Data Processing & Feature Engineering:**
    - Applied **lag features** and **rolling window aggregations** to capture historical trends.
    - Performed **data transformations** to improve model performance.
    - Merged datasets, including **calendar data**, to incorporate external factors.
    - Detected **price changes**, which significantly impact demand.

- **Model Selection & Performance:**
    - Tested **ARIMA, Exponential Smoothing, LightGBM, Prophet-XGBoost ensemble, and XGBoost**.
    - **XGBoost selected** due to high accuracy, efficiency, and robustness to outliers.
    - **Final Model Metrics:**
        - **$R^2 = 0.67$** (explains 67% of demand variability).
        - **MAE = 1.02** (average forecast error of 1.02 units).

- **Deployment & Monitoring:**
    - Model deployed using **FastAPI** in a **Docker container** for real-time predictions.
    - **Airflow pipeline** can automate weekly retraining and model drift detection.

- **Streamlit dashboard** could provide business teams with forecast monitoring tools.
- **Business alignment is crucial** before full production rollout to validate expectations.

Project Workflow Overview

The demand forecasting project followed a structured approach, progressing through multiple stages, from data preparation to model deployment. Below is a breakdown of each step:

1. Data Cleaning & Processing (`data_cleaning_processing.ipynb`)

- Applied **lag features** and **rolling window aggregations** to capture temporal patterns.
- Performed **data transformations** to normalize and enhance predictive signals.
- Merged different datasets, including the **calendar data**, to incorporate external factors.
- Detected **price changes**, which can significantly impact demand and need to be considered in the model.

2. Exploratory Data Analysis (EDA) (`EDA.ipynb`)

- Conducted in-depth **Exploratory Data Analysis (EDA)** to understand sales trends, seasonality, and anomalies.
- Analyzed missing data, demand patterns, and external influences on sales.

3. Baseline Model Testing with PyCaret (`experiments_pycarot.ipynb`)

- Used **PyCaret (AutoML)** to quickly test **10 different models** on **10% of the dataset** to establish a baseline performance.

4. Initial Model Exploration (`exploration_models_1.ipynb`)

- Tested various forecasting models, including:
    - **XGBoost** (best performance)
    - **LightGBM** (similar accuracy to XGBoost but slower training)
    - **Exponential Smoothing**
    - **ARIMA**

5. Hyperparameter Tuning & Feature Importance (`exploration_models_2.ipynb`)

- Fine-tuned **XGBoost** using different configurations to optimize performance.
- Applied **SHAP (SHapley Additive Explanations)** to interpret feature importance and understand which variables drive model predictions.

6. Final Model Training & Predictions (`final_model.ipynb`)

- Trained the final **XGBoost model** using optimized hyperparameters.
- Generated new **predictions** and saved results in an efficient format:
    - **Predictions stored as Parquet** for better performance and storage efficiency.
    - **Model saved as Pickle (`.pkl`)** for easy loading and deployment with FastAPI.

7. Model Deployment (`serv_model/`)

- Set up a deployment pipeline using **FastAPI** within a **Docker container**.
- This allows the model to be **served as an API**, making it accessible for real-time applications.

Results and Discussion

Results

Sales Growth & Trends:

- California has the highest sales volume, followed by Texas and Wisconsin.
- FOODS category dominates, followed by HOUSEHOLD and then HOBBIES.
- Sales patterns are consistent across all states.

- **Exploratory Insights:**
  - Correlation matrix helped identify key relationships between features.
  - Weekly sales show stability despite occasional outliers.
  - Clear seasonality detected in monthly sales trends.
- **Data Processing & Feature Engineering:**
  - Applied **lag features** and **rolling window aggregations** to capture historical trends.
  - Performed **data transformations** to improve model performance.
  - Merged datasets, including **calendar data**, to incorporate external factors.
  - Detected **price changes**, which significantly impact demand.
- **Model Selection & Performance:**
  - Tested **ARIMA, Exponential Smoothing, LightGBM, and XGBoost**.
  - **XGBoost selected** due to high accuracy, efficiency, and robustness to outliers.
  - **Final Model Metrics:**
    - **$R^2 = 0.67$** (explains 67% of demand variability).
    - **MAE = 1.02** (average forecast error of 1.02 units).
- **Deployment & Monitoring:**
  - Model deployed using **FastAPI** in a **Docker container** for real-time predictions.

- **Business alignment is crucial** before full production rollout to validate expectations.

Discussion

During the exploratory analysis, we identified important patterns in the data that shaped our modeling decisions. Over time, sales have grown, with California leading in volume, followed by Texas and Wisconsin. This pattern suggests that different states contribute differently to overall demand, likely influenced by regional customer preferences or store-specific factors. In terms of product categories, FOODS had the highest sales, followed by HOUSEHOLD items, with HOBBIES coming in last. Interestingly, this ranking remained consistent across all three states, highlighting a general trend rather than a location-specific behavior.

A closer look at the correlation matrix revealed relationships between key variables, offering insights into which features might be most valuable for predicting demand. We also examined sales patterns across the week and found that while outliers were present, the overall demand remained relatively stable. However, when analyzing demand across months, a clear seasonal trend emerged, indicating that certain periods of the year consistently see higher sales. This insight was crucial in designing our forecasting approach.

Based on these findings, we structured our data processing pipeline to capture essential patterns. We applied lag features and rolling window aggregations to account for historical demand trends, ensuring that our model could recognize past sales behavior. Data transformations were implemented to standardize key features, improving the model's ability to generalize across different time periods. Additionally, we merged multiple datasets, including calendar data, to provide context for external events like holidays or promotions that might impact sales. Another critical step was detecting price changes, as fluctuations in pricing could significantly influence demand, and capturing these shifts helped improve forecast accuracy.

When selecting a forecasting model, we prioritized both performance and practical considerations such as training efficiency and robustness to outliers. After testing multiple approaches—including traditional time series models like ARIMA and Exponential Smoothing, as well as machine learning methods like XGBoost and LightGBM—we found that XGBoost provided the best balance between accuracy and efficiency. LightGBM delivered similar accuracy but took slightly longer to train, making XGBoost the preferred choice.

To evaluate the model's performance, we focused on metrics that are both meaningful for forecasting accuracy and easy to communicate with stakeholders. The model achieved an R^2 of 0.67, meaning it explains 67% of the variability in demand, which is a strong result given the complexity of the dataset. The MAE of 1.02 indicates that, on average, our predictions deviate by approximately one unit from actual sales, providing a tangible reference for business teams when interpreting forecast reliability.

With a working model in place, the next challenge was ensuring its deployment and ongoing monitoring. We developed an API using FastAPI, packaged within a Docker container, allowing easy integration with external applications that may require demand predictions. However, model performance can degrade over time due to data drift, where changes in consumer behavior, pricing, or external factors make past patterns less relevant. To address this, we propose automating retraining with Airflow, scheduling weekly model updates and generating reports to detect performance shifts. If significant degradation occurs, the system could trigger either automatic retraining or a manual review for further investigation.

In addition to automated monitoring, a Streamlit dashboard could provide a user-friendly way for business teams to visualize the model's performance, tracking key metrics and forecast accuracy over time. However, before moving to full production, it is essential to align with business stakeholders to ensure the model meets their expectations. This final step allows for fine-tuning the approach based on real-world feedback and ensures that the model delivers meaningful insights for decision-making.