Assignment 2

ML as a Service

ANIKA CHAUHAN
Student ID: 14188775

08/10/2023

36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Executive Summary

The project's aim is to deploy two machine learning models as APIs for a U.S. retailer with ten stores across California, Texas, and Wisconsin. The first model predicts sales revenue for specific items in individual stores on given dates, while the second forecasts total sales revenue for the next 7 days across all stores and items. The models address the need for precise revenue predictions, essential for inventory management and business planning.

The project's significance lies in data-driven insights that help optimize operations and revenue. Predictive and forecasting models guide decisions on inventory, store-specific sales strategies, and resource allocation.

1. Predictive Model: We've built a predictive model using machine learning to accurately forecast item-specific store sales on specific dates. It's been rigorously trained and evaluated, ensuring its reliability.

2. Forecasting Model: For total sales revenue predictions over the next 7 days, we've implemented a time-series forecasting model. This model supports proactive decision-making by providing insights into revenue trends.

3. API Deployment: Both models were deployed through Heroku-based APIs, offering the ability to retrieve predictions for specific items, stores, dates, and 7-day national revenue forecasts.

In conclusion, these models can equip the retailer with data-driven tools to enhance sales predictions, optimize resources, and make better inventory decisions. Thus contributing to improved business operations, leading to revenue maximization.

For in-depth information and access to the APIs, please refer to the provided GitHub repository and API URLs.

2. Business Understanding

a. Business Use Cases

The Business addresses the following business use cases:

1. **Sales Forecasting:** Accurately predicting future sales is critical for inventory management, staffing, and financial planning. Machine learning models can help retailers make data-driven decisions by forecasting sales for individual products in different store locations.
2. **Revenue Optimization:** Optimizing revenue involves pricing strategies, promotion planning, and product assortment. Machine learning models assist in identifying the best pricing and promotion strategies to maximize revenue while considering factors like seasonality, product demand, and competition.
3. **Inventory Management:** Efficient inventory management is essential for reducing costs and ensuring products are in stock when customers demand them. Machine learning models help retailers understand demand patterns and reduce overstock or understock situations.
4. **Customer Behavior Analysis:** Understanding customer behavior, such as preferences and purchase patterns, is essential for targeted marketing and improving customer experience. Machine learning can provide insights into customer segmentation and product recommendations.

Challenges and Opportunities:

The challenges in retail include volatile demand, seasonality, competition, and rapidly changing market conditions. Opportunities lie in making informed decisions that lead to increased sales, reduced costs, and improved customer satisfaction.

b. Key Objectives

The key objectives according to each of the aforementioned use-cases of this project are: To develop accurate models to forecast sales at the store and product level. This information can then be stored and used for optimal pricing and promotion strategies. The retail chain can also look into minimizing overstock and understock situations by predicting demand.

The potential stakeholders and their requirements are as follows:

1. **Retailers and store managers** who want to optimize inventory and pricing.

2. **Marketing and sales teams** looking for insights into customer behavior.

3. **Financial planners** who need accurate sales and revenue forecasts.
4. **Supply chain managers** aiming to streamline operations.
5. **Customers** seeking improved shopping experiences.

3. Data Understanding

In this project, we had several datasets that provided information about the American retailer's sales and related data. These datasets included:

1. Training Data (sales_train.csv): This dataset contained historical sales data for various items in different stores across three states (CA, TX, WI).
2. Calendar Data (calendar.csv): This dataset provided information about dates, weeks, and events.
3. Calendar Events Data (calendar_events.csv): This dataset contained additional information about calendar events, such as special events and holidays.
4. Item Weekly Sell Prices Data (items_weekly_sell_prices.csv): This dataset contained information about item prices in different stores.

Data Limitations:

While the data provided was rich and comprehensive, there were several potential limitations to consider:

- The large number of items, stores, and dates lead to **model complexity**, leading it to require advanced techniques and computational resources.
- The project **assumed** that historical sales data and pricing information were representative of future sales patterns.
- The dataset did not consider **external factors** that could potentially influence sales, such as economic conditions or competitor actions.

3.Data Preparation

Common data preparation steps were fundamental in establishing a robust foundation for both predictive and forecasting models, ensuring data consistency and reliability. The process began with **data loading**, importing sales, price, and calendar data. Subsequently, **data transformation** converted the sales data from a wide to long format, yielding a new dataframe with vital columns: 'item_id,' 'store_id,' 'day,' and 'value' (quantity sold).

The transformed sales data was merged with the calendar information to align the temporal aspects in the dataset. Categorical variables, 'item_id' and 'store_id,' were **label-encoded** for consistency. This encoding was extended to the pricing data ('data_price_df') to create a comprehensive dataset. Weekly sell prices were incorporated via merging based on 'store_id,' 'item_id,' and 'wm_yr_wk.'

Data cleaning addressed missing values in the 'value' column. **Feature engineering** enhanced the dataset by converting the 'date' column into datetime format and representing it as a timestamp in seconds since the epoch. The original 'date' column was dropped to reduce redundancy.

Data splitting segregated the dataset into feature variables (X) and the target variable, 'cya_target' (for the predictive model). These steps collectively established a reliable and insightful foundation for both predictive and forecasting models, ensuring data integrity and consistency.

4. Modeling

For Predictive model:

a. Baseline - Mean Prediction Model

The baseline model for this project was a simple mean prediction model that provided a baseline MAE score of approximately 5.2976. The goal of the advanced predictive models (XGBoost and LightGBM) was to outperform this baseline by achieving lower MAE scores, thereby demonstrating their predictive capabilities.

b. XGBoost (Extreme Gradient Boosting)

Preprocessing and Feature Engineering:

The XGBoost model implementation commenced with preprocessing steps, which included label encoding to convert categorical variables 'item_id' and 'store_id' into numeric format.

No further feature engineering was introduced, and the model utilized the original features, namely 'item_id,' 'store_id,' and 'date_timestamp.'

Training Process:

The XGBoost model was trained on the preprocessed training data, and it was fitted to the cleaned optimized dataset using specified hyperparameters. MAE was calculated for reference to judge its performance against the baseline.

Rationale:

XGBoost was chosen as the predictive modeling tool due to its widespread popularity and its prowess in handling tabular data problems. Renowned for its ability to capture intricate relationships between features, XGBoost was a reasonable choice to optimize predictive performance. The selected hyperparameters represented typical starting values, striking a balance between model complexity and performance.

c. LightGBM (Light Gradient Boosting Machine)

Preprocessing and Feature Engineering:

Similar to the XGBoost model, LightGBM's preprocessing phase included label encoding for 'item_id' and 'store_id' to facilitate numeric conversion.

Training Process:

The LightGBM model was trained on the preprocessed training data, with the training process mirroring that of the XGBoost model.

Rationale:

LightGBM, another gradient boosting algorithm, was selected for its efficiency and speed, making it an apt choice for predictive modeling. This algorithm often excels when dealing with tabular data featuring a substantial number of categorical features.

We used two gradient boosting algorithms to determine how well could both of them mold according to the same data

For Forecasting model:

a. SARIMA Model (Seasonal Autoregressive Integrated Moving Average - Forecasting):

Preprocessing and Feature Engineering:

In the case of the SARIMA model, several preprocessing steps were taken:

1. Merging 'sales_train.csv' with 'calendar.csv' to incorporate crucial date-related information into the dataset.

2. Applying label encoding to categorical variables 'item_id' and 'store_id' for numerical conversion.

The model relied on the core features 'item_id,' 'store_id,' 'date_timestamp,' and 'cya_target' (the target variable).

Training Process:

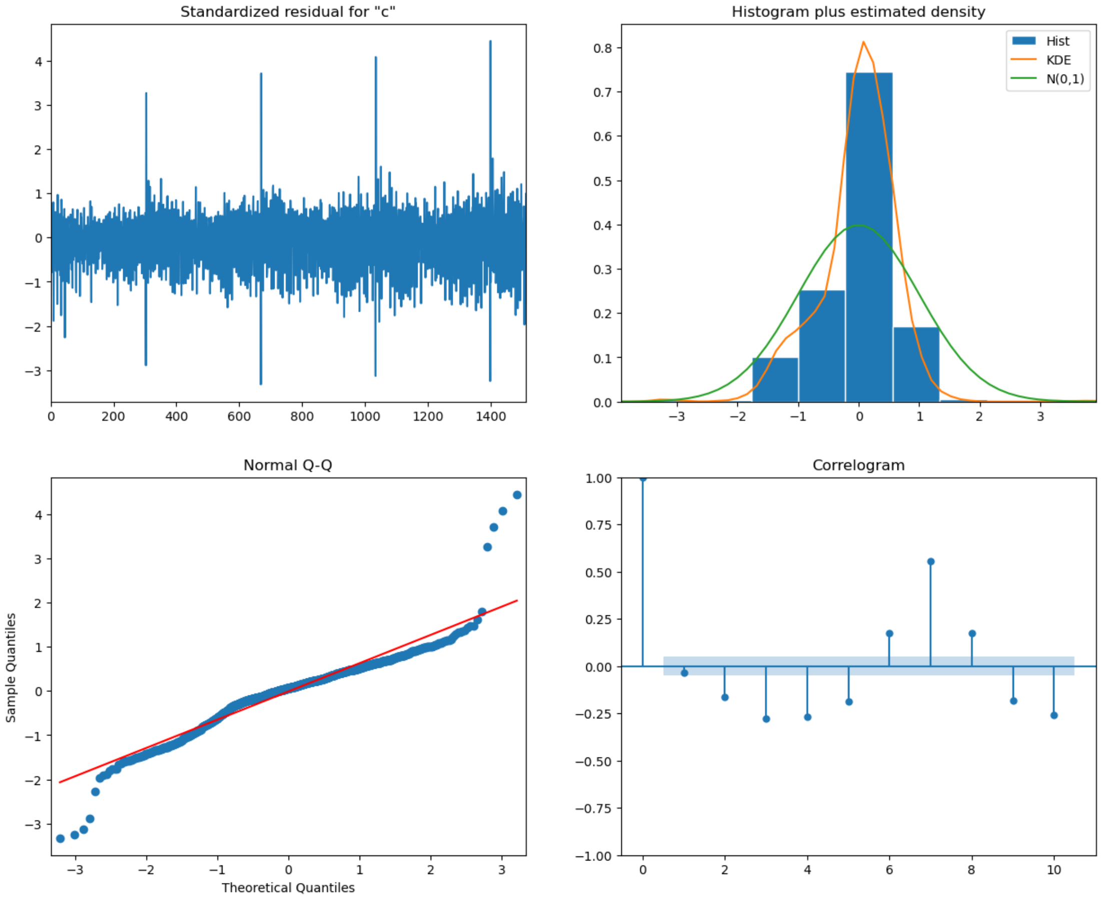
The SARIMA (Seasonal Autoregressive Integrated Moving Average) model was defined and trained on the historical dataset. To capture seasonality and trends, specific model orders and parameters were set:

Order: (1, 1, 1) for the non-seasonal components.

Seasonal Order: (1, 1, 1, 12) with a seasonal period of 12, indicating monthly seasonality. The training process revolved around fitting the SARIMA model to the historical time series data.

Rationale:

SARIMA models are well-suited for time series forecasting tasks as they are tailored to capture seasonality, trends, and autocorrelation patterns in time series data, especially when data exhibits distinct seasonality, trends, and autocorrelation. In this context, the choice of SARIMA is justified, as it is tailored to effectively capture monthly patterns within the data. The specified SARIMA orders (1, 1, 1) for non-seasonal components and (1, 1, 1, 12) for seasonal components are typical starting values for SARIMA modeling and are known to yield reliable forecasting results.



b. Exponential Smoothening Model (Extreme Gradient Boosting)

Preprocessing and Feature Engineering:

In the Exponential Smoothing model designed for forecasting, the preprocessing and feature engineering steps followed this structure:

1. The dataset was initially prepared by merging the 'sales_train.csv' data with 'calendar.csv' to introduce date-related information.

2. Categorical variables 'item_id' and 'store_id' were subjected to label encoding to convert them into a numeric format.

Training Process:

The Exponential Smoothing model, specifically employing the Simple Exponential Smoothing method, was trained on the preprocessed dataset. The model was fit to historical data using a specified smoothing parameter ('smoothing_level') with a value of 0.2. The training process centered on capturing the underlying patterns and trends in the time series data.

Rationale:

Exponential Smoothing is a time series forecasting method known for its simplicity and effectiveness in capturing time-dependent patterns within data. The choice of Exponential Smoothing is reasonable for time series forecasting tasks, where the primary objective is to predict future values based on historical data. The smoothing parameter (alpha) value of 0.2 represents a typical starting point and governs the emphasis given to recent observations when making predictions. Next steps will include experimenting with different alpha values to get a better result.

5.Evaluation

For Predictive models:

a. Evaluation Metrics

For predictive models, the Mean Absolute Error (MAE) was used as the evaluation metric. MAE measures the average absolute difference between the predicted and actual values. It provides a straightforward understanding of how well the model's predictions match the true target values. Lower MAE indicates better performance. MAE was chosen as the evaluation metric in the context of predicting revenue as it directly quantifies the model's ability to provide accurate forecasts, aligning with the project's goal of improving revenue predictions.

b. Results and Analysis

- Baseline Model (Mean Prediction): The baseline model, which predicts the mean revenue, serves as a reference point for evaluation, yielding an MAE of 5.2976.
- XGBoost Model: XGBoost outperformed the baseline, achieving an MAE of 4.9325.
- LightGBM Model: The LightGBM model also outperformed the baseline, with an MAE of 4.9719.
- Analysis: Both advanced predictive models, XGBoost and LightGBM, demonstrated superior performance by outperforming the baseline. XGBoost, in particular, achieved the lowest MAE, indicating the highest predictive accuracy among the tested models.

c. Business Impact and Benefits

The predictive models, especially XGBoost, contribute to more accurate revenue estimations, which are crucial for inventory management, pricing strategies, and revenue optimization. By reducing prediction errors, the models can help businesses make informed decisions, allocate resources efficiently, and maximize revenue. The potential value generated includes cost savings from better inventory management, increased revenue from optimized pricing, and improved customer satisfaction through product availability.

The forecasting models, including SARIMA and SES (Simple Exponential Smoothing), play a pivotal role in improving the accuracy of revenue predictions, a critical aspect for effective inventory management, pricing strategies, and revenue optimization. By minimizing prediction errors, these models empower businesses to make data-driven decisions, allocate resources more efficiently, and ultimately maximize their revenue potential. The tangible benefits encompass substantial cost savings achieved through enhanced inventory management, augmented revenue resulting from optimized pricing strategies, and the heightened satisfaction of customers who can rely on improved product availability.

d. Data Privacy and Ethical Concerns

- The data used in predictive modeling may contain sensitive information related to sales and pricing.
- Ethical concerns related to data collection and usage include ensuring data security, privacy compliance, and fair pricing practices.
- To address these concerns, steps were taken to anonymize and protect sensitive customer and sales data. Data handling practices adhered to relevant privacy regulations and ethical guidelines.

6. Deployment

We created an API using python FastAPI. In the application we created end points which would be accessible by the user allowing them to provide their custom inputs which would result in a structured response generated through the use of our trained model which was loaded using the joblib library. This application was further stored in a docker container so that it could be deployed as an app through Heroku.

As the model is trained more its size is bound to increase and as github has a 100 mb limit, if the file size increases more than that, then the deployment through heroku will not be possible. For that we may need to upload the models to an external cloud based system like AWS or Microsoft Azure.

The large size of the models, did not allow the app to be launched successfully. It crashed due to the size restrictions put in place by Heroku and Git.

We recommend that instead of directly uploading the model, a link should be created to external cloud sources which would host the models, leading to less memory stress on deployment platforms such as Heroku.

7.Conclusion

This project was successfully in producing models which could accurately predict and forecast sales revenue for specific items in individual stores on given datesand the total sales revenue for the next 7 days across all stores and items. The models addressed the need for precise revenue predictions, essential for inventory management and business planning.

The project was successful in combining provided data, and creating classifiers for predictions and forecasting. However, the API which was to be deployed, had challenges due to the unforeseen size restrictions which were encountered during the deployment process leading to it crashing.

In the future, we may create a link to an external cloud source which would host the models, leading to less memory stress on deployment platforms such as Heroku.

8. References

- Include a list of references used throughout the project report.

- What Is a SARIMAX Model?- https://365datascience.com/tutorials/python-tutorials/sarimax/

- LightGBM - https://stackoverflow.com/questions/60360750/lightgbm-classifier-with-gpu

- A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python- https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/

- Heroku deployment- https://www.freecodecamp.org/news/how-to-deploy-an-application-to-heroku/

- Code reference – Some snippet of codes were incorporated from Adv_MLA lab exercises

9.Appendix

1. Planned structure of the API:
   ‘/’ (GET): Displaying a brief description of the project objectives, list of endpoints, expected input parameters and output format of the model, link to the Github repo related to this project
   ‘/health/’ (GET): Returning a welcome message
   ‘/sales/national/’ (GET): Returning next 7 days sales volume forecast for an input date
   ‘/sales/stores/items/’ (GET): Returning predicted sales volume for an input item, store and date
2. GitHub Repo: https://github.com/anikachauhan30/amla2
3. Heroku API: https://shrouded-scrubland-09364-50393435d911.herokuapp.com/