Assignment 2
ML as a Service

—

Vishal Raj
Student ID: 14227627

06th October 2023

Table of Contents

1. Executive Summary

- **Overview:** SalesMaster-ML is a cutting-edge revenue prediction and forecasting system designed meticulously to revolutionise business sales revenue forecasting. It aims to provide reliable predictions at both national and granular levels (specific store-item combinations), serving as an essential tool for efficient inventory management. With accurate predictions, it helps companies efficiently navigate inventory planning, avoiding overstock and stockout scenarios and ensuring customer satisfaction while optimising profitability.

- **Problem Statement and Context:** Companies continually seek accurate sales revenue prediction mechanisms in today's business environment. Reliable forecasts are crucial for effective inventory management strategies. This project was initiated to integrate diverse datasets, including historical sales, calendar information, and event data, to create a robust system for sales forecasting.

- **Objectives and Significance:** The primary goal was to develop, train, and evaluate models that accurately forecast sales revenues with minimal error. The project explored the intricacies of the 'XGBoost' and 'Prophet' models, assessing their performance and ensuring user accessibility. It was imperative to develop sophisticated models and deploy them through a user-friendly and accessible application, democratising access to advanced forecasting tools.

- **Achieved Outcomes:** The project successfully navigated the development and evaluation phases of the 'XGBoost' and 'Prophet' models. Both models exhibited promising performance metrics in a controlled environment, with potential effectiveness in real-world scenarios. A significant milestone was deploying these models through a Fast API application on Heroku, making advanced sales forecasting accessible and user-friendly for all.

2. Business Understanding

a. Business Use Cases

SalesMaster-ML is designed for crucial business use cases and is valuable in various commercial scenarios.

- **Inventory Management:** SalesMaster-ML significantly eases inventory management challenges by providing accurate sales predictions and helping maintain optimal inventory levels. This is vital, especially for retail businesses where inventory management directly impacts customer satisfaction and revenue. Precise predictions prevent stockouts and excess stock, avoiding lost sales, increased holding costs, and wastage for perishable goods.

- **Financial Planning:** Accurate sales predictions, like those provided by SalesMaster-ML, are foundational for effective financial planning and forecasting, enabling businesses to allocate budgets, invest, and distribute resources informedly. These forecasts support the development of solid financial strategies, anticipating revenues and highlighting profitable and underperforming products.

b. Key Objectives

- **Develop Accurate Models:** SalesMaster-ML's main objective is to develop and evaluate accurate and reliable predictive models. A precise sales prediction model is essential in the face of fluctuating market dynamics. The aim is not just to develop a working model but to create one that excels in prediction, serving as a reliable tool for strategic business planning.

- **User-Friendly Deployment:** Given the need for accessibility and ease of use in today's business environment, SalesMaster-ML ensures the models are theoretically sound and practically applicable. Deployed via a Fast API application on Heroku and hosted on Github, the models are accessible without complex interfaces, offering an intuitive application that allows quick data input and prediction retrieval.

- **Stakeholder Engagement:** The project considered stakeholder requirements, developing SalesMaster-ML to meet the needs of business professionals requiring accurate sales predictions for planning and strategy. By meeting these objectives, SalesMaster-ML provides reliable sales forecasting solutions, helping businesses optimise operations,

enhance customer satisfaction, and maximise profitability. Each goal aligns with practical business needs, offering a valuable and accessible tool.

■ ■ ■

3. Data Understanding

Understanding the data is crucial for SalesMaster-ML's success, with the dataset being a careful compilation from various sources, each offering unique insights essential for forecasting.

- **Data Sources and Collection**

The primary dataset combines four sources: 'sales_train', 'calendar', 'items_weekly_sell_prices', and 'calendar_events'. The 'sales_train' dataset is the project's foundation, providing detailed historical sales data at the store-item level. The 'calendar' dataset gives contextual information, mapping each date to its day, week, month, and year and marking special events or holidays, helping understand seasonal sales trends. 'items_weekly_sell_prices' gives weekly selling prices of items at each store, linking sales volume to revenue. The 'calendar_events' dataset lists special events or promotions impacting sales, helping the models account for sales variations during these periods.

- **Data Features and Significance**

'sales_train' data features include 'item_id', 'dept_id', 'cat_id', 'store_id', 'state_id', and columns 'd_1 to d_1541' representing daily item sales in stores. The 'calendar' dataset has columns like 'date', 'wm_yr_wk', and 'd'. 'calendar_events' dataset features are 'date', 'event_name', and 'event_type'. The 'items_weekly_sell_prices' dataset includes 'store_id', 'item_id', 'wm_yr_wk', and 'sell_price'.

- **Data Limitations and Challenges**

The dataset, while comprehensive, has challenges like the presence of null values and the need for additional features for more in-depth model training. The null values, created during data merging and analysis, needed careful handling. The extensive size and complexity of the dataset also posed computational and processing challenges, requiring efficient algorithms and optimisation techniques. These challenges were addressed through careful data understanding, preparation, and engineering, resulting in a dataset suitable for training and evaluating predictive models.

4. Data Preparation

Data preparation was vital, transforming raw data into a format suitable for modelling.

- **Initial Data Processing:** Data from 'sales_train', 'calendar', 'items_weekly_sell_prices', and 'calendar_events' were combined to form a comprehensive dataset. This merge aligned sales data with price, calendar, and event information, providing a context-rich dataset for forecasting.

- **Feature Engineering:** Feature engineering enhanced the dataset's predictive power. Temporal features like 'day_of_week', 'month', and 'year' were derived to capture sales data seasonality. Categorical variables like 'event_type' were encoded to be algorithm-friendly. This process introduced null values, which were not errors but represented a lack of events or specific characteristics for some entries.

- **Handling Null Values:** Null values were carefully handled. For example, nulls in 'event_type_encoded' were filled with a placeholder value indicating 'no event', allowing the models to distinguish between days with and without events.

- **Data Transformation:** Categorical features were label encoded for the 'XGBoost' model, transforming them into numerical format while maintaining their categorical nature. The dataset was then split into training and testing sets, ensuring a balanced representation of different stores, items, and periods, facilitating reliable model evaluation.

This careful data preparation laid a strong foundation for model development and evaluation, providing accurate, relevant, and organised data for effective learning and forecasting.

5. Modelling

The modelling stage of the SalesMaster-ML project was meticulously planned and executed, with a clear focus on developing accurate and reliable predictive models for sales revenue predicting and forecasting. Two distinct modelling approaches were employed, 'XGBoost' and 'Prophet', each leveraging a different machine learning algorithm designed to optimise the predictive accuracy and reliability of the sales revenue forecasts. 'XGBoost', or Extreme Gradient Boosting, is an advanced implementation of gradient boosting known for its speed and performance. For sales

revenue prediction at the store-item level, the 'XGBoost' algorithm was chosen due to its capability to handle large datasets efficiently and its flexibility in modelling complex relationships within the data.

Also, the 'Prophet' algorithm was employed for national-level sales revenue forecasting. Developed by Facebook, Prophet is designed for forecasting time series data with seasonal solid patterns and multiple seasonality. The algorithm suits sales data, which exhibits daily, weekly, and yearly seasonal patterns.

a. Approach 1

The process began with data pre-processing, where the dataset was transformed into a format suitable for training the 'XGBoost' model. Features such as 'day_of_week', 'month', 'year', 'event_type_encoded', 'item_id', 'dept_id', 'cat_id', 'store_id', and 'state_id' were carefully engineered to encapsulate the temporal dynamics, item characteristics, and store information essential for accurate sales forecasting. These features provided the 'XGBoost' model with the necessary contextual information to effectively learn the underlying patterns and trends in the sales data.

Hyperparameter tuning was a crucial aspect of the 'XGBoost' modelling process. The model's hyperparameters, including learning rate, maximum depth of the trees, and number of trees, were optimised using 'HyperOpt', a Python library for optimising the hyperparameters of machine learning models. This optimisation process enhanced the model's predictive accuracy while preventing overfitting, resulting in a robust model capable of generalising well to unseen data.

b. Approach 2

The Prophet model was trained on a dataset comprising daily national sales data, with features engineered to capture the influence of various events on sales revenues. An additional feature, 'num_events', was created to quantify the number of events occurring each day on an average for the next 7 days, providing the model with contextual information necessary for understanding the impact of events on daily national sales. It can be kept at "0" for standard forecasting if one doesn't want to include its effect in forecasting.

The Prophet algorithm automatically detects and accounts for seasonality in the data, making it an excellent tool for forecasting sales revenues with high seasonal variations. Its intuitive parameters, such as seasonality and holidays, allow for easy customisation and fine-tuning of the model to accurately capture the seasonal trends and holiday effects observed in the sales data.

6. Evaluation

a. Evaluation Metrics

Various metrics assessed the models' performance in the SalesMaster-ML project.

- MAE: The Mean Absolute Error (MAE) gauges the average absolute errors between predicted and actual values, offering accuracy insights.

- MSE & RMSE: Mean Squared Error (MSE) and its square root, Root MSE (RMSE), highlight more significant errors, with RMSE providing error magnitude in data units.

- MAPE & SMAPE: Mean Absolute Percentage Error (MAPE) and Symmetric MAPE (SMAPE) express errors as percentages, with SMAPE normalising errors for different series scales.

b. Results and Analysis

- **XGBoost Model Results:** With an MAE of 4.068, MSE of 77.637, and RMSE of 8.811, the 'XGBoost' model reasonably predicts item-store-level sales, managing intricate data relationships effectively.

- **Prophet Model Results:** Despite a high MAPE of 779.59%, the Prophet model's 7.80% SMAPE indicates acceptable national sales forecasting accuracy, capturing complex seasonal sales data patterns.

Comparatively, 'XGBoost' excels in granular predictions, while Prophet is adept at national forecasting, with both models complementing each other's strengths for diverse business forecasting needs.

c. Business Impact and Benefits

From a business perspective, the predictive models developed offer substantial value. Accurate sales forecasts empower businesses to optimise inventory levels efficiently, reducing the risks of overstock and stockouts, which directly influence customer satisfaction and profitability. Strategic decision-making processes at various organisational levels are enhanced, with accurate forecasts providing a solid foundation for planning and execution, ultimately contributing to increased business efficiency and profitability.

d. Data Privacy and Ethical Concerns

Considering the sensitive nature of sales data, privacy concerns were diligently addressed. The dataset was anonymised to protect individual identities and sensitive information. Ethical considerations were at the forefront of the project, ensuring that the data was used responsibly and that the models developed would not inadvertently introduce bias or unfairness in their predictions.

7. Deployment

Deploying the developed machine learning models into a production environment is crucial for realising their practical value and impact. This section discusses the deployment process, integration considerations, and challenges encountered for the SalesMaster-ML project.

- **Deployment Process**

The deployment process for SalesMaster-ML involved several crucial steps:

- ○ Fast API Application: The 'FastAPI' framework was chosen for deploying the models due to its fast performance, ease of use, and automatic generation of 'OpenAPI' documentation. It allows creating a 'RESTful API' that can serve predictions in real-time, making the models accessible and usable for various applications and devices.

- ○ Docker Containerization: Docker was used to containerise the application, creating a portable and consistent environment for running the models. Containerisation ensures that the application behaves the same regardless of where it is deployed, simplifying the deployment process and mitigating potential issues related to differences in operating environments.

- ○ Heroku Deployment: Because of Heroku container size limitations, the GitHub repository was used to deploy the application on Heroku, a cloud platform that enables easy scaling and management of applications. Heroku was chosen for its simplicity and ease of use, providing a platform that allows for quick deployment and management of the application without requiring extensive infrastructure management skills.

- GitHub Repository: All relevant code, including the trained models, pre-processing scripts, and deployment configurations, were organised and stored in a GitHub repository. This repository acts as a central location for the deployment, managing, and sharing of the project's code, making it accessible for collaboration, review, and future development efforts.

- **Integration and Real-world Implementation**

The deployed models must integrate seamlessly with existing business systems and processes for real-world implementation. Integration considerations included:

- API Integration: The 'FastAPI' application provides endpoints that can be accessed through HTTP requests, allowing for easy integration with various systems and applications. The API's design considered the potential needs and technical capabilities of the systems that would be interacting with it, ensuring compatibility and ease of use.

- Data Consistency: Ensuring consistency between the data used for training the models and the data input for predictions is vital. Data pre-processing and transformation scripts used during the modelling phase were incorporated into the deployment pipeline, guaranteeing that incoming data is appropriately prepared for predictions.

- Scalability: The deployment platform and architecture were designed with scalability in mind to handle varying loads and usage patterns. The application can be scaled horizontally to accommodate increased demand, ensuring consistent performance even under heavy loads.

- **Deployment Challenges**

Despite careful planning and execution, the deployment phase encountered challenges:

- Size Limitations: Deployment platforms often impose limitations on applications' size and associated files. Managing the size of the trained models and ensuring they fit within these limits while maintaining their predictive performance was a challenge that required careful consideration and optimisation.

- Data Pre-processing Consistency: Maintaining consistency in data pre-processing between the modelling and deployment phases was non-trivial. Ensuring that the deployed models received data in the expected format and range required careful management of pre-processing scripts and vigilant testing.

8. Conclusion

● **Key Findings and Insights**

The SalesMaster-ML project aimed to develop robust forecasting models to predict sales revenues effectively. The project yielded several significant findings and insights:

○ Effective Forecasting: The developed XGBoost and Prophet models demonstrated a commendable capability in forecasting sales revenues accurately at both granular (store-item) and national levels. The models captured complex patterns within the data, proving the viability of machine learning in sales forecasting tasks.

○ Feature Importance: Through meticulous feature engineering and selection, the project revealed the significant impact of specific features (like day_of_week, month, year, event_type_encoded) on the forecasting performance. Understanding these influential features is crucial for interpreting model predictions and making informed business decisions.

○ Deployment Challenges: The deployment process, while successful, presented challenges related to size limitations, data pre-processing consistency, and platform compatibility. Addressing these challenges required careful planning and optimisation, highlighting the complexities of deploying machine learning models into production environments.

● **Achievement of Goals and Stakeholder Satisfaction**

SalesMaster-ML achieved its primary objectives:

○ Accurate Forecasts: The project delivered models that provide accurate sales revenue forecasts, supporting effective inventory management and business strategic planning.

○ User-Friendly Application: With the deployment of a FastAPI application, the models are accessible through a user-friendly interface, making advanced forecasting tools available to non-technical users.

○ Stakeholder Requirements: The project's outcomes align with the requirements and expectations of stakeholders, offering a practical solution to the challenges of sales revenue forecasting. The delivered system supports the stakeholders in

making data-driven decisions, ultimately contributing to enhanced business efficiency and profitability.

- **Recommendations for Future Work**

Reflecting on the project's outcomes, several recommendations emerge for future work:

- ○ Model Improvement: Continued exploration and implementation of advanced modelling techniques and algorithms are essential. Experimenting with ensemble methods and deep learning approaches may yield models with improved accuracy and reliability.

- ○ Feature Engineering: Further refinement and expansion of feature engineering techniques can enhance model performance. Investigating additional features and their relationships with sales revenues can provide deeper insights into sales dynamics.

- ○ Deployment Optimisation: Consider alternative deployment platforms and strategies to overcome size limitations and enhance scalability. Implementing Continuous Integration and Continuous Deployment (CI/CD) pipelines can automate deployment processes, ensuring the application is always up-to-date and stable.

- ○ User Interface Enhancement: Investing in improving the application's user interface can provide users with a more intuitive and engaging experience, encouraging widespread adoption and use of forecasting tools.

- **Final Reflections**

The SalesMaster-ML project was a comprehensive endeavour, successfully navigating through complex data analysis, modelling, evaluation, and deployment phases. It achieved its set objectives and provided valuable insights and lessons for future work in sales revenue forecasting. The journey from understanding the business problem to deploying a functional application underscored the importance of each step in the data science process, each contributing to the project's ultimate success. The project stands as a testament to the power and potential of machine learning in transforming business processes and delivering tangible value to stakeholders. Future efforts in this domain will undoubtedly build upon the foundations laid by SalesMaster-ML, pushing the boundaries of what is possible with machine learning and data science in sales forecasting.

9. References

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://dl.acm.org/doi/10.1145/2939672.2939785

Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. *The American Statistician*, 72(1), 37–45. https://peerj.com/preprints/3190/

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51 - 56). https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning* (Vol. 28, pp. I-115–I-123). http://proceedings.mlr.press/v28/bergstra13.html

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array Programming with NumPy. *Nature*, 585(7825), 357–362. https://www.nature.com/articles/s41586-020-2649-2

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261–272. https://www.nature.com/articles/s41592-019-0686-2

FastAPI. (2021). FastAPI: A modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. https://fastapi.tiangolo.com/

Docker. (2021). Empowering App Development for Developers. https://www.docker.com/

Heroku. (2021). Cloud Application Platform. https://www.heroku.com/