EXPERIMENT REPORT

Student Name	Vishal Raj
Project Name	SalesMaster-ML: Advanced Sales Revenue Forecasting System
Date	6th October 2023
Deliverables	• raj_vishal-14227627-predictive_xgboost.ipynb • raj_vishal-14227627-forecasting_prophet.ipynb • XGBoost Sales Forecasting Model (final_xgboost_model.joblib) • Prophet National Sales Forecasting Model (final_prophet_model.joblib) • Fast API Application – **Heroku Link** (https://sales-master-app-031d89e0c0e1.herokuapp.com/docs#/default/read_root__get) • Dockerfile & .dockerignore for containerisation • Procfile for Heroku Deployment • SalesMaster-ML **Github Repository** (https://github.com/vishalraj247/SalesMaster-ML.git)

1. EXPERIMENT BACKGROUND

Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.

1.a. Business Objective	The overarching goal of the Sales Master App is to offer a sophisticated, reliable predicting and forecasting system that can predict sales revenues meticulously at both national and granular levels (specific store-item combinations). An accurate predictive and forecast system is paramount for businesses implementing efficient inventory management strategies. With precise predictions, companies can avoid overstock and stockout scenarios, which not only ensures customer satisfaction but also maximises profitability by optimising inventory costs and enhancing sales efficiency.
1.b. Hypothesis	The experiment is grounded in the hypothesis that sales revenues can be accurately predicted by effectively utilising a combination of historical sales data, calendar information, and data regarding specific events and pricing. It is anticipated that the implementation of advanced machine learning models, like ‘XGBoost’ and ‘Prophet’, will significantly enhance the accuracy and reliability of the sales revenue forecasts by learning complex patterns and trends within the data.

1.c. Experiment Objective	The primary objective of this experiment is to meticulously develop, train, and evaluate predictive models capable of forecasting sales revenues with a minimal margin of error. The experiment aims to assess and compare the performance of the ‘XGBoost’ and ‘Prophet’ models under various scenarios and conditions, ultimately identifying the model that offers the most accurate and reliable forecasts for different forecasting tasks. Additionally, part of the objective is to deploy the developed models in a user-friendly application, ensuring accessibility and ease of use for end-users.

2. EXPERIMENT DETAILS	
Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.	
2.a. Data Preparation	A comprehensive dataset was compiled by merging data from different sources, including ‘sales_train’, ‘calendar’, ‘sell_prices’, and ‘calendar_events’. Each data source provided unique insights, and their combination resulted in a rich dataset that captured the various factors influencing sales. The data underwent a rigorous cleaning, pre-processing and transformation process to address missing values, outliers, and format inconsistencies, ensuring it was well-suited for training the predictive and forecasting models.
2.b. Feature Engineering	Feature engineering was meticulously conducted to enhance the models’ predictive capabilities. For the XGBoost model, features encapsulating temporal dynamics, item characteristics, and store information were created and transformed. These included ‘day_of_week’, ‘month’, ‘year’, ‘event_type_encoded’, ‘item_id’, ‘dept_id’, ‘cat_id’, ‘store_id’, and ‘state_id’. For the Prophet model, an additional feature, ‘num_events’, was crafted to capture the influence of the number of events on sales revenues. These engineered features provided the models with the necessary contextual information to effectively learn and predict sales trends and patterns.
2.c. Modelling	Two distinct predictive models, ‘XGBoost’ and ‘Prophet’, were developed and fine-tuned. ‘XGBoost’, known for its high performance and efficiency with structured data, was utilised for its ability to handle complex relationships within the data. On the other hand, ‘Prophet’ was chosen for its proficiency in time series forecasting, especially with data exhibiting strong seasonal components. The hyperparameters of the ‘XGBoost’ model were optimised using ‘HyperOpt’ to enhance its predictive accuracy.

3. EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance	Both developed models exhibited satisfactory performance in preliminary tests. The XGBoost model yielded an MAE of 4.068, MSE of 77.637, and RMSE of 8.811, indicating a reasonable level of accuracy in predicting sales at the item-store level. The Prophet model achieved a MAPE of 779.59% and a SMAPE of 7.80%, demonstrating its capability to forecast national sales with acceptable accuracy. These metrics indicate reasonable accuracy in predicting sales revenues at both the item-store and national levels.
3.b. Business Impact	The accurate forecasting provided by the models is invaluable from a business standpoint. With precise sales predictions, businesses can efficiently plan their inventory levels, which is crucial for preventing revenue loss due to overstock and stockouts. Furthermore, accurate forecasts are instrumental for financial planning and can significantly contribute to strategic decision-making processes at various levels within a business organisation, thus, contributing to improving business efficiency and profitability.
3.c. Encountered Issues	During the course of the experiment, challenges related to data inconsistency and quality were encountered. The extensive data preparation and cleaning process was necessary to address these issues, highlighting the importance of investing time and resources in data preprocessing to ensure the development of robust and reliable predictive and forecasting models. The pre-processing of input data points through manual input for prediction to bring them to the same transformed format as trained data using the same preprocessor and encoder saved as joblib was a little exhausting and included lots of trial and error. The complexity of the size limit of the app deploying on Heroku also created a few issues.

4. FUTURE EXPERIMENT

Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.

4.a. Key Learning	The experiment provided valuable insights and learning experiences related to sales revenue forecasting. It highlighted the importance of careful data preparation, rigorous feature engineering, and thoughtful selection and tuning of forecasting models. The hands-on experience with the ‘XGBoost’ and ‘Prophet’ models provided a deeper understanding of their strengths, weaknesses, and areas where they require further optimisation and fine-tuning. Also, the process of developing a user-friendly Fast API application elucidated the importance of creating an intuitive user interface that effectively communicates with the backend models, ensuring a seamless user experience. Furthermore, the deployment process, involving Docker and Heroku, highlighted the necessity for a scalable, maintainable, and accessible platform that allows end-users to utilise the forecasting capabilities conveniently.

4.b. Suggestions / Recommendations

For future work, it is recommended to explore additional modelling approaches and advanced feature engineering techniques to further improve forecasting accuracy. There is also a need to investigate ensemble methods that might leverage the strengths of both ‘XGBoost’ and ‘Prophet’ models. Continuous model validation and performance monitoring are essential to ensure the models' robustness and reliability in a live, production environment. Once the models are further validated and fine-tuned, steps should be initiated to deploy them into a production environment where they can provide real-time forecasting insights for business users.

In terms of application development, future iterations could focus on improving the user interface's intuitiveness and responsiveness, possibly by implementing more advanced front-end technologies and frameworks. Additionally, considering alternative deployment strategies and platforms could also be beneficial, taking into account factors like cost, scalability, and ease of use. Continuous model validation and performance monitoring are essential to ensure the models' robustness and reliability in a live, production environment.

Further, future endeavours should consider implementing continuous integration and continuous deployment (CI/CD) pipelines to automate the testing and deployment processes, thereby ensuring that the application is always running the most updated and stable version of the code. Implementing logging and monitoring within the application will also be crucial for tracking the application’s performance and identifying and resolving issues promptly.