Assignment 2

Machine Learning as a Service

Monali Patil
Student ID: 14370946
10/10/2023

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

Table of Contents

Project Repo:	https://github.com/MonaliPatil19/adv_mla_assignment2.git
API Servicing Repo:	https://github.com/MonaliPatil19/assignment2_api.git
Heroku API URL:	https://obscure-escarpment-15993-549ea5837663.herokuapp.com/docs

1. Executive Summary

Project Overview:

The project involves working with an American retail chain operating across California (CA), Texas (TX), and Wisconsin (WI) offering a diverse range of products, including hobbies, foods, and household items, across ten stores. This project aimed to develop and deploy two distinct machine learning models as API services to assist in business decisions related to sales revenue to achieve overall business profitability.

1. Predictive Model: To accurately predict sales revenue for individual items in specific stores for any given date for managing inventory levels, optimizing pricing strategies, and ensuring product availability.

2. Forecasting Model: To forecast the total sales revenue across all stores and items for the upcoming 7 days, commencing from the provided input date using a time series analysis algorithm allowing for better resource allocation, staffing decisions, and financial planning.

Problem Statement and Context:

The challenge for predictive was to build robust machine learning models that could effectively predict sales revenue based on various factors such as historical sales data, calendar events, and item prices. While for forecasting was to develop a time series forecasting model that could predict sales revenue accurately for a specified future date, helping the business plan effectively.

This project addresses the critical challenge of sales revenue prediction and forecasting in the retail sector. In the predictive modelling phase, machine learning models were developed to enhance revenue prediction, while the forecasting phase focused on leveraging the Prophet algorithm for accurate forecasting of sales incomes. These initiatives aim to provide data driven insights and support informed decision making in the dynamic retail industry.

Achieved Outcomes and Results:

The project resulted in the development of predictive models using algorithms like XGBoost and Linear Regression. These models provided insights into revenue trends, contributing to better inventory management and pricing strategies. In addition, the project successfully implemented the Prophet algorithm to forecast sales revenue. It achieved low Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) scores, indicating accurate revenue predictions. This will aid the business in planning and decision-making.

In summary, the project involves predictive modelling for revenue prediction and Prophet algorithm-based forecasting to provide accurate future revenue estimates, contributing to improved business strategies in the retail industry.

2. Business Understanding

This project primarily focuses on two key business use cases:

Sales Revenue Prediction: The first use case involves predicting sales revenue for a retail business. Accurate sales revenue prediction is crucial for inventory management, pricing strategies, and overall business profitability. By predicting future sales revenue, the retail business can optimize stock levels, plan marketing campaigns effectively, and make informed pricing decisions.

Sales Revenue Forecasting: The second use case is revenue forecasting, specifically for a time horizon of 7 days. This short-term forecasting is essential for tactical decision-making, such as staffing and resource allocation. By accurately predicting revenue for the next 7 days, the retail business can optimize staffing levels, ensure product availability, and enhance customer service.

Challenges and Opportunities:

This project addresses critical business challenges in the retail sector by using machine learning to enhance sales revenue prediction and forecasting, ultimately driving better business outcomes.

- The challenges that motivated this project include the complexity of retail data, the need to handle large datasets efficiently, and the potential impact of inaccurate revenue predictions on business operations. Inaccurate predictions can lead to overstocking or understocking of products, resulting in financial losses, or missed sales opportunities.

- Opportunities lie in leveraging machine learning algorithms to harness the power of data for better decision-making. By applying regression and time-series forecasting techniques, the project aims to provide actionable insights that can lead to improved inventory management, pricing strategies, and overall business profitability.

Key Objectives:

- Accurate Predictions: Develop a revenue prediction and forecasting model that provides precise predictions.
- User-Friendly Interface: Develop an intuitive interface for end users and stakeholders to access the model through an API.

- Scalability: Ensure the model's ability to handle real-time data for both prediction and forecasting.
- Privacy and Fairness: Define principles and guidelines for privacy and fairness.
- Ethical Data Handling and Bias Prevention: Ensure ethical treatment of the retailer's data and prevent bias.
- Stakeholder Collaboration: Collaborate closely with retail managers, pricing analysts, and domain experts.
- Ethical Guidelines: Define principles for responsible prediction use.
- Continuous Enhancement: Utilize new data to improve model performance.

3. Data Understanding

Dataset:

The dataset consists of historical sales and revenue data from an American based retail business. It includes information on various aspects, such as dates, store locations, item categories, and daily sales spanning across separate CSV files. This dataset serves as the foundation for building predictive and forecasting models to aid in inventory management and pricing strategies.

Data Source and Collection Methods:

The dataset is provided in CSV format and was accessed through the canvas portal. It included separate files for training, testing, and various other below purposes. As it was acquired via the University portal as part of student resources, there were no copyright or privacy concerns associated with the dataset.

Dataset Names	Description
1. sales_train.csv	The training dataset includes the retailer's transactional records spanning the period from 2011 to 2016.
2. sales_test.csv	The testing dataset comprises the retailer's transactional records for the years 2015 to 2016.
3. calendar.csv	A dataset containing date-related information and its associated reference columns.
4. calendar_events.csv	A dataset containing event names and their corresponding types, for instance, an event name is 'LaborDay' with an event type of 'National'.
5. items_weekly_sell_prices.csv	A dataset containing the weekly selling prices of items of various types from different stores on specific dates.

Data Limitations:

- Lack of Metadata: The dataset lacks metadata or detailed descriptions of variables and their meanings. This limitation required relying on domain knowledge or making anticipations, which can introduce uncertainty into the analysis.
- Limited Store and State Coverage: The dataset only includes weekly sell prices for items from two stores, CA_1 and CA_2, out of a total of 10 stores spanning three different states. This limited coverage may raise concerns about data completeness and generalizability to the broader retail environment.
- Unique Identifiers: The presence of unique identifiers, such as product IDs, in the data can pose a challenge. These identifiers may lead to overfitting if not handled carefully. Overfitting occurs when a model learns to memorize specific data points rather than identifying general patterns, which can result in poor predictive performance on new data.
- Ethical Considerations: It is essential to consider ethical data limitations, particularly when handling transactional data. This includes ensuring that customer privacy is respected, and that data usage complies with relevant regulations and guidelines.

Addressing these limitations through appropriate data preprocessing, feature selection, and model training techniques is crucial for building a robust and reliable revenue prediction model.

4. Data Preparation

To ensure the quality of the data for modelling, conducted the below activities.

- **Reshaping the training dataset and renaming the attributes.**

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	...	d_1532	d_1533	d_1534	d_1535	d_1536	d_1537	d_1538
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	1	1	1	0	1	0	1
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	0	0	0	0	0	0	0
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	0	0	1	0	0	0	0
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	8	2	0	8	2	3	1
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...	2	0	1	3	2	1	1

The above training dataset had columns named from d_1 to d_1541 representing the sold item volume for each day for distinct items from 10 stores across 3 separate states. So, melt functionality was utilized to reshape the data to improve attribute organization and for more convenient and suitable data analysis and modelling task.
Additionally, the 'd_*' column was relabeled as 'dayofsale,' and the 'sold item numbers' were renamed as 'volume.

- **Combining the individual datasets into a single dataset.**

id	item_id	dept_id	cat_id	store_id	state_id	dayofsale	volume	date	wm_yr_wk	d	event_name	event_type	sell_price
BBIES_2_050_CA_4_evaluation	HOBBIES_2_050	HOBBIES_2	HOBBIES	CA_4	CA	d_741	0	2013-02-07	11302	d_741	NaN	NaN	1.97
OODS_3_782_WI_2_evaluation	FOODS_3_782	FOODS_3	FOODS	WI_2	WI	d_114	0	2011-05-22	11117	d_114	NaN	NaN	2.48
EHOLD_2_505_CA_1_evaluation	HOUSEHOLD_2_505	HOUSEHOLD_2	HOUSEHOLD	CA_1	CA	d_686	1	2012-12-14	11246	d_686	NaN	NaN	4.97

Combining data from various sources into a single data frame is a fundamental step in data preprocessing for modelling. It ensures data consistency, enables feature engineering, and simplifies the overall analysis process, ultimately contributing to better decision-making and model performance.

- **Feature Engineering.s**

- Computing the total revenue based on the items sold and the weekly price of an item:
    - Calculating the overall revenue by considering the number of items sold and their corresponding weekly prices was essential for developing the machine learning model to predict target variable revenue.

- Processing the 'date' feature to derive additional date related information
    - To enhance the dataset's date related information by extracting additional date related features from the 'date' attribute such as day of a week, month, year, and week of the year for a model to learn generalized features.
    - Therefore, set the datetime type to the date feature and utilized the datetime functionality to extract the day of the week, month, year, and week of the year information.

- **Selecting appropriate features.**

Features selected: 'item_id', 'store_id', 'date', 'event_name', 'event_type', 'revenue'

To build a machine learning predictive model, above specific features were chosen. Due to the presence of duplicate columns after aggregating the datasets, it resulted in the large dataset leading to memory related issues. For instance, 'store_id' field values (CA_1, WI_2, TX_9) already incorporates the 'state_id' attribute values (CA, WI, TX). Therefore, it's logical to choose pertinent features that have generalized unique values and avoid duplicates.

Additionally, to address the memory challenges associated with analyzing the extensive dataset, the above mentioned features were chosen based on their relevance from a business perspective.

- **Identifying and eliminating identifiers.**

The 'id' attributes serve as unique identifiers for individual items sold within a specific category, store, and state on a particular date. As a result, during the feature selection process, this identifier was omitted.

Including it in the analysis could potentially result in overfitting, where the model becomes too focused to these specific values, rather than capturing the underlying, generalized patterns present in the retailer's data.

- **Splitting data into different datasets.**

To split the dataset into training and validation below strategy was employed.

- Training data: From 2011-01-01 to 2014-12-31
- Validation data: from 2015-01-01 to 2016-12-31

The retail data covers multiple years, and splitting it into training (historical) and validation (future) sets, would maintain the temporal consistency of the dataset. This ensures that the model is trained on past data and tested on more recent data, mimicking a real-world scenario.

Furthermore, this approach aided in handling the relatively compact dataset, mitigating memory-related challenges.

- **Handling and imputing missing values.**

After combining the datasets, missing values were identified in the attributes.

Attribute Name	Total missing values
event_name	43143350
event_type	43143350
revenue	12291876

To handle missing values in the 'event_name' and 'event_type' features, 'None' was used as a replacement for the missing values.

For address missing values in the 'revenue' target feature, 0 was used for imputation. This signifies days with no items sold and, consequently, zero income.

- **Aggregating the revenues based on the date across all items, stores, and states.**

	date	revenue
0	2011-01-29	81650.61
1	2011-01-30	78970.57
2	2011-01-31	57706.91
3	2011-02-01	60761.20
4	2011-02-02	46959.95

To construct the revenue forecasting model using the Prophet time-series model, aggregated the revenue figures based on the date, encompassing all items, stores, and states.

5. Modelling

As a part of the learning process, developed and trained the following regression and time-series models.

- Predictive Model

Regression Algorithm: The target variable 'revenue' contains continuous values that represent sales income. Since our goal is to predict these continuous incomes, using a regression algorithm is the appropriate choice, aligning with the objective of learning and prediction.	
Algorithm Name	Rationale
1. XGBoost Regression	* XGBoost algorithm is an ensemble learning method, which means it combines the predictions of multiple weak learners (typically decision trees) to create a stronger, more accurate model. It builds trees sequentially, each one correcting the errors of the previous trees, leading to improved predictive performance. * The algorithm has demonstrated exceptional predictive performance in various machine learning competitions and real-world applications. It is known for its ability to handle complex, non-linear relationships in the data, making it suitable for regression tasks. * Additionally, it is optimized for speed and efficiency, making it suitable for large datasets. It can handle a substantial number of features and samples efficiently.
2. Linear Regression	* Linear regression algorithm is computationally efficient and does not require the extensive computational resources that some complex machine learning algorithms demand. This efficiency makes it suitable for large datasets and real-time applications. * It serves as a valuable starting point for regression tasks, allowing data scientists to establish a base model before exploring more complex alternatives.

Pipeline

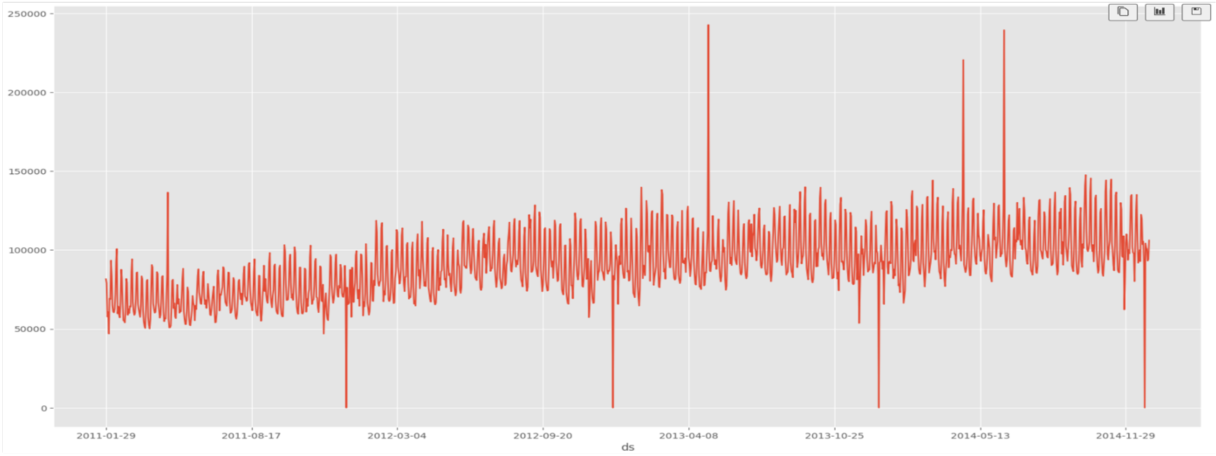
The pipeline automates the steps required to preprocess input data, and make predictions as the same preprocessing and prediction steps are consistently applied to incoming data, ensuring that results are reproducible and dependable.

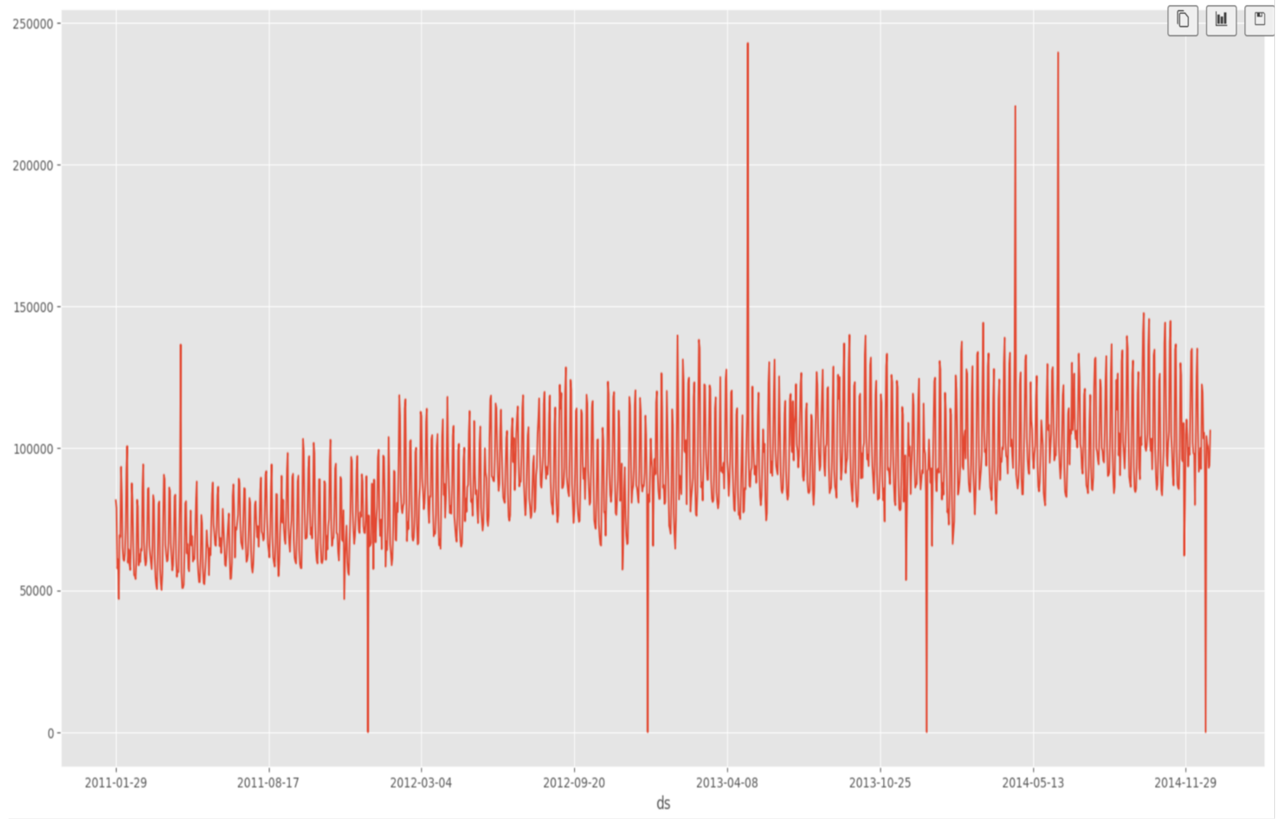
Building a pipeline for model deployment for servicing with an API enhances the model's usability and reliability in using the predictive model in an operational environment.

- Forecasting Model

Regression Algorithm: The target variable 'revenue' contains continuous values that represent sales income. Since our goal is to predict these continuous incomes, using a regression algorithm is the appropriate choice, aligning with the objective of learning and prediction.	
Algorithm Name	Rationale
1. Phophet time series algorithm	* The Prophet time series model for revenue forecasting has the ability to handle time-dependent data effectively. * Prophet is specifically designed for forecasting tasks and can capture the various patterns and seasonality often present in time series data, which is common in revenue forecasting. * Additionally, it provides the flexibility to incorporate external factors, such as holidays and events, which can significantly impact sales. * Moreover, Prophet is user-friendly and requires minimal tuning, making it a suitable choice for both beginners and experts. Overall, its predictive accuracy and ease of use make it a strong candidate for revenue forecasting in this learning task.

Employing Holiday Construct of the Phophet algorithm.





The charts above demonstrate periodic spikes in retail revenue in both directions, with noticeable increases and declines occurring during specific months of the year, following a recurring pattern at certain intervals.

While these spikes could be attributed to various factors, we can hypothesize that they may be related to long weekends, holidays, or significant promotions that the company runs approximately every quarter, potentially impacting the sales revenue.

To leverage this information for more accurate forecasting during those months, utilising Prophet's holiday feature which is a Pandas dataframe containing the holidays and their respective dates.

6. Evaluation

Evaluation Metrics:

Mean Absolute Error (MAE): MAE measures the average absolute difference between the predicted and actual values. It provides a clear understanding of the model's prediction accuracy in terms of the absolute deviation from the actual values. A lower MAE indicates more accurate predictions.

MAE is relevant because it offers a straightforward assessment of how much the model's revenue predictions deviate from actual values, without considering the direction of errors (overestimation or underestimation). Like RMSE, it helps in understanding the magnitude of prediction errors, which is essential for effective inventory management and pricing decisions.

Root Mean Square Error (RMSE): RMSE calculates the square root of the average of the squared differences between predicted and actual values. It provides a measure of prediction accuracy in the same units as the target variable 'revenue'.

RMSE is a suitable metric for understanding the overall model performance and is relevant to the project's objective of minimizing prediction and forecast errors. It assesses the magnitude of errors in revenue predictions giving higher weights to larger errors and helps in understanding how close the model's predictions are to the actual revenue values.

Mean Absolute Percentage Error (MAPE): MAPE measures the percentage difference between the predicted and actual values. It provides insights into the accuracy of revenue predictions in terms of percentage deviation from the actual values. A lower MAPE indicates more accurate predictions, which align with the project's goal of accurate revenue forecasting.

It's valuable for understanding how well the model performs in forecasting revenue, which is a critical aspect of the project's goal to improve inventory management and pricing strategies. A lower MAPE indicates more accurate predictions, which can lead to better inventory management decisions and pricing strategies.

Results and Analysis:

- Predictive Model

Algorithm	Datasets	MAS Score	RMSE Score
	Baseline Performance	4.3372	9.0484
Experiment 1: XGBoost Regressor	Training Dataset	4.0605	8.7152
	Validation Dataset	4.3389	10.4643
Experiment 2: Linear Regression	Training Dataset	4.2788	9.0114
	Validation Dataset	4.4691	10.6893

The baseline model has a MAS score of 4.3372 and an RMSE score of 9.0484. This serves as a reference point for evaluating the other models.

Experiment 1: XGBoost Regressor

- The XGBoost Regressor algorithm achieves a lower MAS score (4.0605) and RMSE score (8.7152) compared to the baseline on the training data. This indicates that the model provides a better fit to the training dataset.

- However, the model's performance on the validation dataset, deteriorates slightly, with a slightly higher MAS score (4.3389) and considerably higher RMSE score (10.4643), suggesting that the model is overfitting to the training dataset, as it struggles to generalize to unseen data.

Experiment 2: Linear Regression

- The Linear Regression model has a MAS score of 4.2788 and an RMSE score of 9.0114 on the training dataset. It performs slightly worse than the XGBoost Regressor but is still better than the baseline.

- On the validation dataset, the model has a higher MAS score (4.4691) and RMSE score (10.6893) compared to the XGBoost Regressor. Similar to XGBoost model, it also overfitts on the training dataset.

- Forecasting Model

Algorithm	Datasets	MAPE Score	RMSE Score
	Baseline Performance	898.0839	21278.1704
Experiment 1: Phophet algorithm	Training Dataset	821.0444	10442.7834
	Validation Dataset	7.0549	9861.6278

The baseline model's MAPE score of 898.0839 indicates that, on average, it predicted sales revenue values with an error of approximately 898%.

The RMSE score of 21278.1704 represents the root mean square error, which measures the square root of the average squared differences between the predicted and actual revenue values. This score is exceptionally elevated, suggesting a significant spread of errors in the baseline model's revenue forecasts.

Experiment 1: Phophet algorithm

- The Prophet algorithm's, MAPE score on the training dataset reduced to 821.0444, indicating that the Prophet algorithm reduced the average prediction error by a significant margin.

- The RMSE score in the training dataset decreased to 10442.7834, signifying a substantial reduction in prediction errors' dispersion.

- The validation dataset showed remarkable results, with an exceptionally low MAPE score of 7.0549. This indicates that, on average, the Prophet algorithm's predictions deviated by only about 7% from the actual sales revenue values, demonstrating high accuracy.

- The RMSE score in the validation dataset further decreased to 9861.6278, highlighting the effectiveness of the Prophet algorithm in reducing errors in revenue predictions.

Therefore, when evaluating the performance metrics, the predictive model using the XGBoost algorithm, from Experiment 1 indicates slightly superior performance compared to linear regression. It not only outperforms the other models but also exceeds the baseline performance.

As a result, the XGBoost model is selected for model deployment, and the pipeline is constructed using the XGBoost model, while for forecasting, the trained Prophet algorithm was employed as part of the forecasting service via an API.

Business Impact and Benefits:

- Predictive Model

The XGBoost Regressor model from experiment 1 outperforms the Linear Regression model although it also exhibits overfitting issues. This suggests that the models may have learned specific patterns in the training dataset that do not generalize well to new, unseen data. The impact of overfitting is that the models may make less accurate predictions while servicing in real-world scenarios.

The root causes of model performance issues, such as overfitting, need further investigation. It could be due to insufficient data, or data transformation choices like ordinal encoder. Identifying and addressing these root causes is critical for improving model performance and business outcomes.

Additionally, incorrect predictions can have various impacts on the business such as overstocking or understocking of products, affecting inventory management efficiency, suboptimal pricing decisions, affecting profitability and inefficiencies in staffing and supply chain management.

- Forecasting Model

The experiments using the Prophet algorithm showed significant improvements in terms of both MAPE and RMSE scores. The MAPE score for the validation dataset, in particular, dropped significantly, indicating that the Prophet model was able to provide more accurate revenue forecasts. This can result in improved inventory management, better pricing decisions, and ultimately higher profitability for the business.

The model's ability to capture time series patterns contributes to strategic decision-making by providing insights into revenue patterns. This information allows the retailer to allocate resources effectively, plan marketing campaigns, and optimize staffing levels based on anticipated sales.

Additionally, the results indicate that employing the Prophet time series algorithm is a suitable choice forecasting model for revenue forecasting in this context, with the potential for further fine-tuning and optimization which can significantly benefit the business by providing more accurate revenue predictions, which can inform better decision-making processes.

Below are a few risks from a business point of view.

- Overfitting: One of the primary business risks is overfitting, where the model performs exceptionally well on the training data but struggles to generalize to new, unseen data. In the context of revenue prediction, overfitting could lead to inaccurate forecasts when applied to real-world sales data.
- Model Robustness: Business conditions can change over time, and the model should continue to deliver accurate forecasts as market dynamics evolve.
- Data Quality: Another significant business risk is the quality of the data used for training and prediction. Inaccurate, incomplete, or biased data can adversely affect the model's predictions.
- Privacy Concerns: Businesses must handle customer and sales data ethically and in compliance with privacy regulations. Failure to do so can result in legal and reputational issues.

Below are some essential recommendations for the retail business.

- Invest in Data Quality Assurance: Address any data anomalies or inconsistencies promptly to prevent them from affecting predictive models.

- Enhance Data Privacy and Compliance: Implement data encryption, access controls, and auditing to protect sensitive information.

- Continuous Model Monitoring and Improvement: Establish a system for ongoing model monitoring and performance evaluation.

- Incorporate External Data Sources: Incorporate relevant data that can provide insights into market trends and potential sales drivers.

- Ethical AI and Fairness: Establish guidelines for the responsible use of predictive models, including ethical considerations.

Data Privacy and Ethical Concerns:

- As the dataset was procured from a university portal for academic purposes, there were no apprehensions related to copyright or privacy infringements.

- Lack of transparency in pricing algorithms can lead to hidden markups or discriminatory pricing. So, it is essential to make pricing algorithms transparent, ensuring customers understand how prices are determined.

- Biased data can lead to unfair predictions, such as discriminatory pricing or targeting. This can harm certain customer groups. So regularly audit data for bias, correct imbalances, and employ fairness-aware algorithms to ensure equitable outcomes.

- Retail businesses collect customer data for various purposes, including sales forecasting. There's a risk of privacy breaches and data misuse. Therefore, implement strong data encryption, access controls, and anonymization techniques.

7. Deployment

Deploying a machine learning model involves several below steps to ensure it can be used in the operational environment.

1. Ensure that the data preprocessing and feature engineering steps used during model development are consistent with those used during deployment.

2. Save the trained machine learning model to a file or format that can be easily loaded in a production environment.

3. Create an API (Application Programming Interface) to expose your model to external systems.

4. To ensure consistency and portability, package your API and model in a container.

5. Choose a deployment environment that suits your needs (AWS, Azure, or on-premises)

6. Consider how your application will handle increased load.

7. Implement monitoring and logging to keep track of the model's performance in production.

8. Implement security measures to protect your API and data.

9. Thoroughly test your deployed model to ensure it performs as expected in the production environment.

10. Create clear and comprehensive documentation for your API.

11. Implement backup and recovery procedures to handle unexpected failures.

12. Ensure that your deployment complies with relevant regulations and privacy laws, especially if you're handling sensitive data.

13. After deployment, continue monitoring your model's performance and retrain it periodically with new data to keep it up to date.

8. Conclusion

In conclusion, this project aimed to develop predictive (XGBoost) and forecasting (Phophet) models for sales revenue in a retail setting. Models were successfully deployed as APIs, providing real-time predictions and forecasts to enhance the overall retail business profits and success.

The project achieved its primary goal by delivering predictive and forecasting models for sales revenue. Stakeholder requirements, including those of retail managers and pricing analysts, were met with user-friendly interfaces providing real-time predictions and forecasts to enhance the overall retail business profits and success.

Further, I would experiment with the XGBoost regressor with hyperparameters like max_depth, subsample, max_leaves etc. to improve their predictive performance for better model generalization. And explore other time series models namely SARIMA.

9. References

So, A., & Tith, R. (2023, September 7). *Model Optimisation* [Lab]. https://colab.research.google.com/drive/1pq9ncvqjzI0GAr830b5kWZsT9Q4GuBvB

So, A., & Tith, R. (2023, September 7). *Model Optimisation* [Lecture]. https://docs.google.com/presentation/d/1QHwfDMJVDTolb7-TGx46Pt738SunE6W6RvFz5xidz-4/edit#slide=id.gb753713d85_0_643

So, A., & Tith, R. (2023, September 14). *Model Deployment* [Lab]. https://docs.google.com/presentation/d/1VZHvfQ3zBBRsfA90-oEjFwhZ6sSJrm-Hqrf_7_6r6BE/edit#slide=id.gb753713d85_0_643

So, A., & Tith, R. (2023, September 14). *Model Deployment* [Lecture]. https://colab.research.google.com/drive/1Qv7MzKCzDMskH2kR_qletXwZLLyzFrts#scrollTo=CcBG6EMNkNjr

(n.d.). *XGBoost Tutorials*. XGBoost. Retrieved October 7, 2023, from https://xgboost.readthedocs.io/en/stable/tutorials/index.html

Brownlee, J. (2021, March 7). *XGBoost for Regression*. Machine Learning Mastery. Retrieved October 6, 2023, from https://machinelearningmastery.com/xgboost-for-regression/

Brownlee, J. (2020, May 8). *Time Series Forecasting With Prophet in Python*. Machine Learning Mastery. Retrieved October 4, 2023, from https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/

(n.d.). *FastAPI*. FastAPI. Retrieved October 8, 2023, from https://fastapi.tiangolo.com

Shah, A. (2023, July 25). *FastAPI Handbook – How to Develop, Test, and Deploy APIs*. FastAPI. Retrieved October 9, 2023, from https://www.freecodecamp.org/news/fastapi-quickstart/