

Assignment 2

ML as a Service

Kritika Dhawale

Student ID: 24587661

October 8, 2023

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



Table of Contents

1. Executive Summary	2
2. Business Understanding	3
a. Business Use Cases	3
3. Data Understanding	5
4. Data Preparation	8
5. Modeling	10
a. Predictive	10
b. Forecasting	10
6. Evaluation	12
a. Evaluation Metrics	12
b. Results and Analysis	12
c. Business Impact and Benefits	14
d. Data Privacy and Ethical Concerns	15
7. Deployment	16
8. Conclusion	18
9. References	19



1. Executive Summary

The project's goal was to create and use two separate predictive & forecasting models to meet important business needs:

1. **Predictive Sales Revenue Model:** This model uses machine learning to predict the revenue made from sales for a specific item, in a specific store on the given date. It makes it possible to manage inventory, plan sales, and make the most money possible.
2. **Forecasting Sales Revenue Model:** This model uses time series analysis to forecast how much revenue will be made from sales of all items and stores in the next seven days. It gives us useful information for making quick decisions and allocating resources.

The goal was to develop precise forecasting and prediction models, which would ultimately aid in better business outcomes and well-informed decision-making. The importance is in improving sales planning, guaranteeing customer satisfaction, streamlining inventory control, and eventually increasing profitability.

Github Link: [adv_mla_at2](#)

Heroku Link: [at2_api](#)

Attained Goals and Outcomes:

- Two models with reachable endpoints were created and successfully implemented on Heroku.
- The predictive sales revenue model, which focused on item-specific predictions for specific stores, showed encouraging results.
- Operational decisions were aided by the forecasting model's useful insights into short-term sales revenue trends.
- Stakeholders can now use both models for data-driven decision-making because they are available via user-friendly endpoints.

The project's outcomes provide the company with the means to raise profitability, sales forecasting, and inventory control—all of which will help it become more competitive in the marketplace.



2. Business Understanding

a. Business Use Cases

The project provides useful answers to particular problems by addressing important business use cases and scenarios.

- **Inventory Optimisation:** The predictive sales revenue model helps individual stores optimise their inventory levels for particular items. By guaranteeing product availability when needed, it lowers carrying costs and avoids stockouts.
- **Sales Planning:** The business can set realistic sales targets and allocate resources appropriately by using accurate sales revenue predictions to facilitate effective sales planning. As a result, sales performance is enhanced.
- **Operational Efficiency:** Effective sales planning and inventory control have a positive influence on operational efficiency. Streamlining supply chain operations involves lowering excess inventory and stockouts.
- **Customer Satisfaction:** Preventing disappointments from out-of-stock items by making sure products are consistently available enhances customer satisfaction.
- **Profitability:** The project directly affects profitability by reducing the costs associated with excess inventory and increasing sales revenue. It improves the bottom line by raising revenue and cutting costs.

Opportunities and Difficulties:

- **Data Complexity:** The company works with enormous datasets that include a wide range of products, retailers, and dates. Accurate sales prediction necessitated managing this complexity.
- **Seasonality:** Temporal dependencies and seasonality are evident in sales patterns. The forecasting model sought to identify these patterns and offer practical advice.
- **Competitive Edge:** By facilitating better decision-making, enhanced inventory management, and increased customer satisfaction, accurate predictions give businesses a competitive edge.
- **Assignment:** This project was given as a part of the assessment for the course AMLA.

The need to use data-driven insights to overcome obstacles and seize opportunities ultimately led to the project's motivation—raising the company's profitability and competitiveness.



b. Key Objectives

Precise Sales Revenue Forecasting: One of the two main objectives is to create a predictive model that can precisely project sales revenue for particular products in particular stores on specified dates.

Short-term Sales Forecasting: Developing a time-series forecasting model that projects the total sales revenue for all items and stores over the next seven days is another important goal.

Stakeholders and What They Need:

1. **Retail Management:** Teams in charge of sales forecasting, inventory control, and profitability are among the stakeholders in the retail industry. Accurate sales forecasts at the item-store-date and short-term total sales levels are among their requirements.
2. **Operations Teams:** To effectively optimise supply chain operations, operations teams need insights into inventory levels, demand patterns, and short-term trends.

Addressing Stakeholder Requirements:

1. The predictive sales revenue model satisfies the demands of retail management by offering precise sales projections for individual items, facilitating well-informed choices concerning inventory and sales scheduling.
2. By providing insights into total sales revenue trends for the next week, the forecasting model helps operations teams with short-term forecasting, which helps with resource allocation and operational efficiency.

The project guarantees inventory availability, improves decision-making processes, and maximises profitability by fulfilling stakeholder requirements and objectives.



3. Data Understanding

a. Data Overview

The project's dataset, which includes details on sales revenue, items, stores, and dates, is derived from retail sales data. It consists of unit sales of different products sold in the United States arranged as grouped time series. To be more precise, the dataset includes the unit sales of 3,049 products that are divided into three product categories (Foods, Household, and Hobbies) and seven product departments that contain the breakdown of the aforementioned categories. Ten stores spread across three states—Wisconsin, Texas, and California—sell the products. Below is a hierarchy map relating to the data attributes:

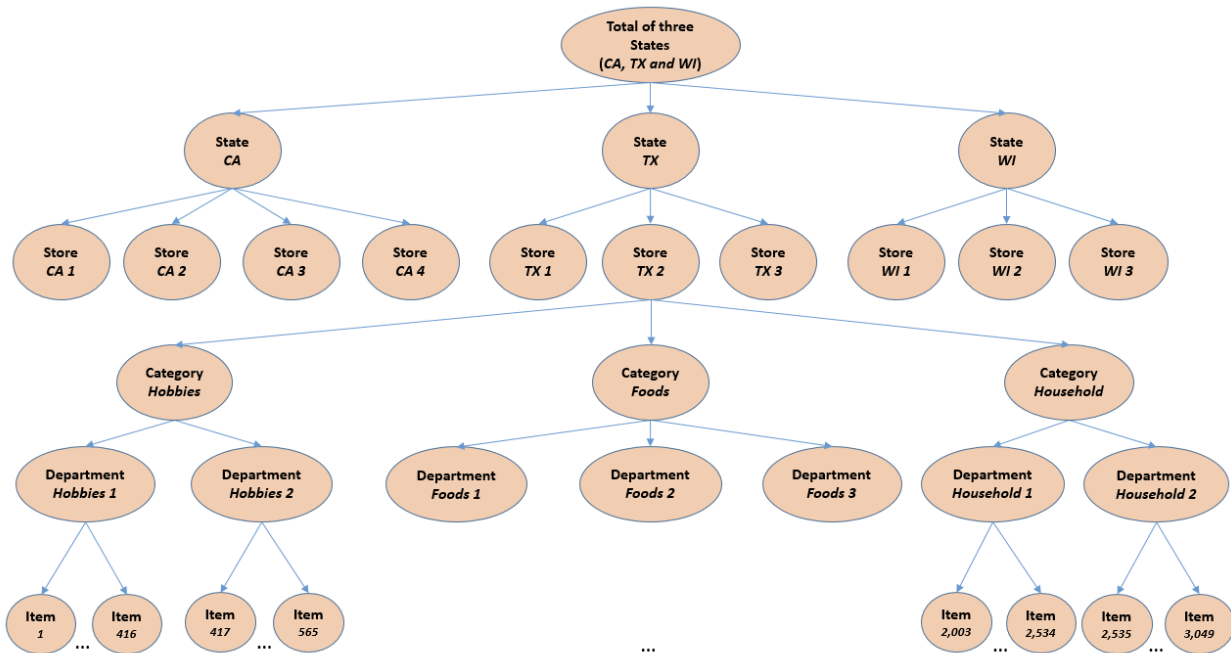


Fig. 1 Overview of data provided

b. Features and Significance

Data files that were provided:

1. calendar.csv: This file includes details on the dates that the products are sold.

- date: The date in a “y-m-d” format.
- wm_yr_wk: The id of the week the date belongs to.
- d: The day of the sale

2. calender_events.csv : This file contains the type and name of events happening for the dates.

- date: The date in a “y-m-d” format.
- event_name: If the date includes an event, the name of this event.
- event_type: If the date includes an event, the type of this event.

2. sales_train.csv: This file includes daily unit sales data for each product and store going back several years.

- item_id: The product's id.
- dept_id: The department ID to which the product is assigned.
- The product's category ID is indicated by the string cat_id.
- store_id: The identity of the retailer that sells the item.
- state_id: The store's location is indicated by state_id.
- d_1 to d_1541 : this indicates the units sold in each day

3. sales_test.csv: This file includes daily unit sales data for each product and store going back several years. This file is an extension to the train file.

4. items_weekly_sell_prices.csv: Provides details on the weekly average cost of items sold by the store in that particular week.

- store_id: The id of the store where the product is sold.
- item_id: The id of the product.
- sell_price: average weekly cost of the item in that store.

This document contains snapshots of the datasets for a data overview: [dataset understanding](#)

c. EDA

An exploratory data analysis was conducted to understand the data better. The analysis revealed that the dataset contained outliers, and other data quality issues that needed to be addressed.

1. **Zero sales (Predictive):** There were 16405658 instances of zero sales of a particular item in a store on a particular date. That counts as **55%** of the cleaned training set, which might hinder the model ability to accurately predict sales without being biased.

2. **Total Sales revenue in each store:** Among all the 10 stores, CA_1 has seen the highest sale ~\$2000, whereas, the highest sale recorded at CA_4 is around \$250 till 2015 year.

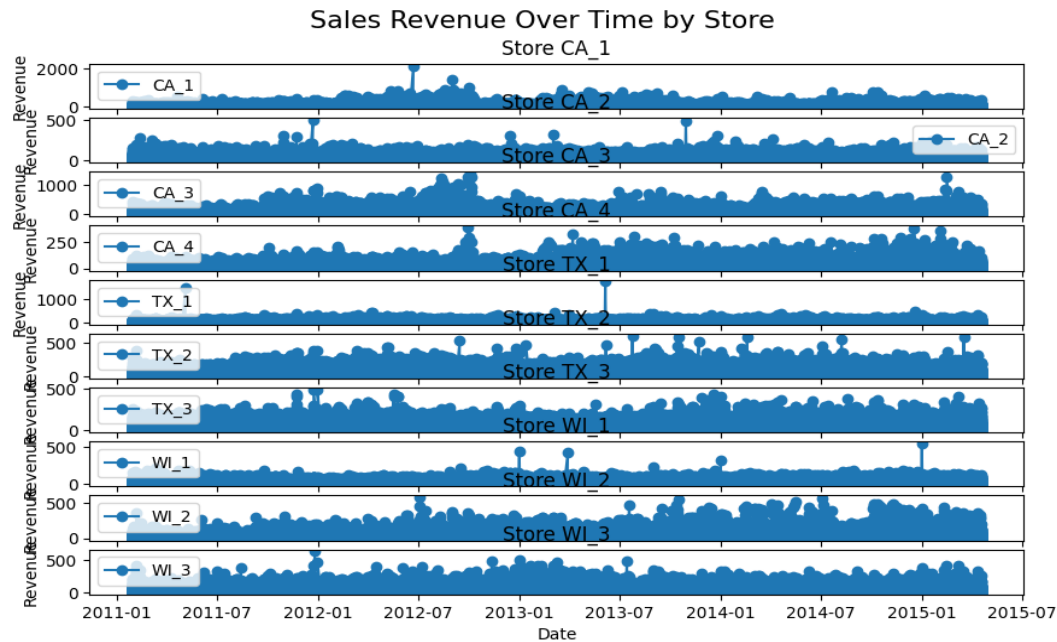


Fig 2. Sales Revenue Over time in each store.

3. **Outliers (Forecasting):** There were a few outliers present in the training data that should be taken into consideration.

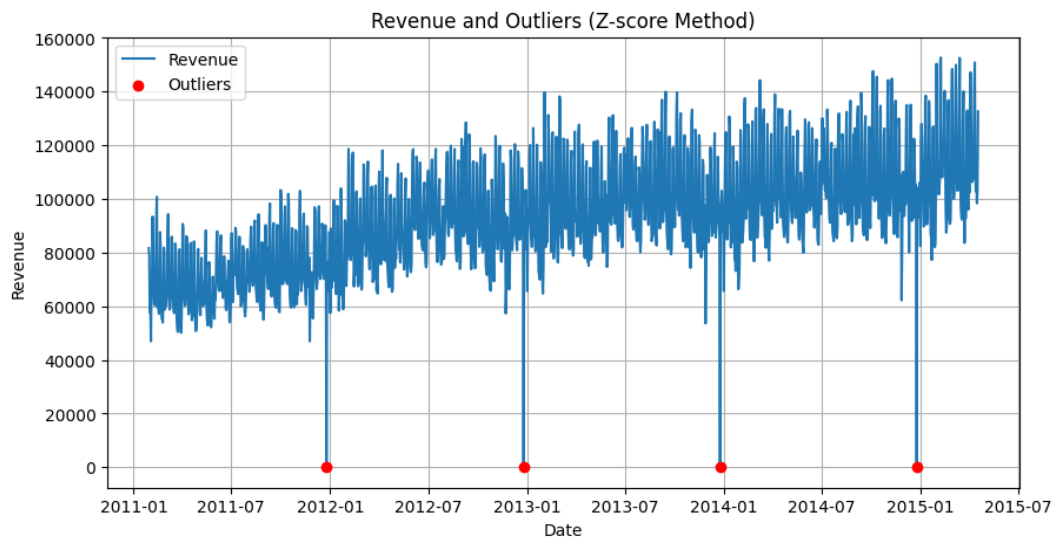


Fig 3. Revenue over time and outliers detected.

4. Data Preparation

a. Predictive

Data Cleaning & Transformation:

1. Column Removal: Unnecessary columns such as "id," "dept_id," "cat_id," and "state_id" were removed to streamline the dataset.
2. Test Set Extension: Item and store IDs were added to the test set to create a comprehensive dataset for predictions.
3. Filtering by Sales Revenue: To enhance data quality, sales revenue was filtered by a minimum non-zero percentage, reducing the number of low sales data samples.
4. Pivoting Data Frame: The data frame was transformed by pivoting the day columns into rows, with units sold each day represented in separate columns. This format was the input to the model.
5. Zero Sales Removal: Approximately 55% of zero sales revenue instances in the train set were removed to mitigate bias.
6. Data Saving: Interim cleaned data was saved into a pickle file for future use.

Data Preprocessing (Included in Model Pipeline):

1. Column Separation: Numerical and categorical columns were separated to apply specific preprocessing steps.
2. Standard Scaling: Standard scaling was applied to numeric data to normalize the features.
3. One-Hot Encoding (OHE): Categorical columns were one-hot encoded, with "handle_unknown" set as "infrequent_if_exist" to handle unknown categories during prediction.

Feature Engineering:

1. Date Merging: Date information from the calendar data was merged into the dataset.
2. Weekly Sell Price Integration: Item's weekly sell prices were merged into the dataset.
3. Daily Total Revenue Calculation: Daily total revenue was calculated by multiplying units sold by the item's weekly sell price for the corresponding week.
4. Date Features Extraction: Features like weekday, day, and year were extracted from the date, providing important temporal information influencing revenue.



b. Forecasting

Data Cleaning & Transformation:

1. Column Removal: Similar to the predictive model, unnecessary columns were removed from the dataset.
2. Test Set Extension: Item and store IDs were added to the test set to ensure a consistent dataset for forecasting.
3. Pivoting Data Frame: Similar to the predictive model, the data frame was pivoted to transform day columns into rows, allowing for revenue calculation on a daily basis.
4. Outlier Removal: Outliers in the train set were removed to normalize the mean and improve model performance.
5. Zero Sales Removal: Approximately 55% of zero sales revenue instances in the train set were removed to mitigate bias.
6. Data Saving: Interim cleaned data was saved into a pickle file for future use.

Data Preprocessing (Included in Model Pipeline):

1. Index Setting: The index of both train and test sets was set to the date, a requisite for time series analysis.
2. Box-Cox Transformation: Data normalization was performed using the Box-Cox transformation to move it closer to a normal distribution.

Feature Engineering:

1. Date Merging: Date information from the calendar data was merged into the dataset.
2. Weekly Sell Price Integration: Item's weekly sell prices were merged into the dataset.
3. Daily Total Revenue Calculation: Daily total revenue was calculated by multiplying units sold by the item's weekly sell price for the corresponding week.
4. Total Revenue Calculation Across All Stores: To forecast total sales revenue across all stores, the items and stores were grouped by date.



5. Modeling

a. Predictive

The `SGDRegressor` model was chosen to perform the given predictive task being motivated by multiple factors:

1. Scalability: SGD-based regression models work well with large datasets because of their well-known scalability and efficiency.
2. Flexibility: The model provides flexibility in modelling by allowing the use of different types of regularisation and loss functions.
3. Regularisation: To effectively handle feature selection, Elastic Net regularization—a combination of L1 and L2 regularization—was selected.
4. Early Stopping: In order to avoid overfitting and strike a balance between model complexity and performance, early stopping was enabled.
5. The selection of hyperparameters was done by fine-tuning "alpha," "max_iter," and "tol," among others, to achieve the ideal balance (more details can be found in the predictive experiment report).


Grid Search Hyperparameter Tuning:

In the beginning, a grid search was used to optimise the hyperparameters "alpha" and "max_iter." Finding the optimal trade-off between computational efficiency and model performance was the goal of the grid search. Grid search was **dropped** in favour of manual hyperparameter tuning due to its **high computational cost**.

b. Forecasting

Three forecasting models (one as a baseline) were trained and evaluated to capture the time-series nature of the data:

1. Exponential Weighted Moving Average (EWMA): This model calculates the Exponential Weighted Moving Average on the training data with an alpha value of 0.1. The choice of alpha was made based on typical values used for smoothing time series data. This simple model serves as a baseline.
2. `Auto ARIMA`: An Auto ARIMA model was chosen for its ability to automatically determine the optimal ARIMA parameters. The hyperparameters tuned include:
 - “Stepwise” set to True - fits the model faster using a subset of hyperparameter combinations (tips).

- 
- “Seasonal” set to True as it considers seasonality in the data.
 - “m” is set to 12 as the data appears to have a yearly seasonality.
 - “n_jobs” set to -1 to maximize CPU usage for faster computation.

3. SARIMAX: A SARIMAX model was chosen to capture seasonality, and it was combined with seasonal decomposition. The hyperparameters were manually selected from the best fit ARIMA model as follows:

- ARIMA order: (5, 1, 5)
- Seasonal order: (2, 0, 2, 12)

Excluded Models:

Complex machine learning models such as Random Forest and Gradient Boosting were not employed, as the primary focus was on time-series analysis using SARIMA and ARIMA. Future experiments may explore the integration of neural networks or other advanced models for additional insights.

The project's objectives, the type of data, and the requirement for scalability and interpretability all played a role in the selection of models and their hyperparameters. The models that were chosen offered a well-rounded strategy to accomplish the project's goals while guaranteeing effective computation. Subsequent studies could investigate different models and extra features to enhance performance even more.



6. Evaluation

a. Evaluation Metrics

Prediction (SGD Regressor):

Evaluation Metric: **Mean Squared Error (MSE)**

Justification: MSE is a useful metric for regression tasks such as revenue forecasting. It highlights the importance of larger errors by calculating the average squared difference between the actual and predicted values. Achieving precise revenue predictions is consistent with the objective of minimising MSE.

Forecasting models (SARIMAX, Auto ARIMA, and Exponential Weighted Moving Average):

Evaluation Metric: **Root Mean Squared Error (RMSE)**

Relevance: The RMSE metric is commonly employed in time-series forecasting. The square root of the average squared discrepancies between the expected and actual values is what it calculates. A clear understanding of forecasting accuracy is provided by RMSE, where lower values denote better good performance.

b. Results and Analysis

Predictive (SGD Regressor):

- Mean Squared Error (MSE) on Train Set: 142.76
- Mean Squared Error (MSE) on Test Set: 152.88

Analysis: The predictive model using SGDRegressor demonstrates reasonably good performance. The MSE for the test set is a little higher than the MSE for the train set, but it is still within a good range. This means that the model works well with data it hasn't seen before, but it could be better.

Data Quality as an underlying issue: We had to remove 55% of the training samples as there were zero sales. This potentially adds up to less training data and removing instances when the store was shut due to public holiday or something. Adding steps to the data preparation process to deal with such cases or data that doesn't match up could make the model even more accurate.

Forecasting models:

Here are the RMSE scores for each model on both the training and test sets:

1. Exponential Weighted Moving Average (EWMA):
 - Train Data RMSE: 14,174.62
 - Test Data RMSE: 18,781.73
2. Auto ARIMA:
 - Train Data RMSE: 56,688.42
 - Test Data RMSE: 17,238.10
3. SARIMA with Seasonal Decomposition (SARIMAX):
 - Train Data RMSE: 27,509.99
 - Test Data RMSE: 21,173.81

Analysis:

- ARIMA model has significantly higher training RMSE compared to test set. This could be one of the reasons: Data Characteristics - The training data may have some unique characteristics or outliers that make it challenging for the ARIMA model to fit accurately. These characteristics might not be as prevalent in the test data.

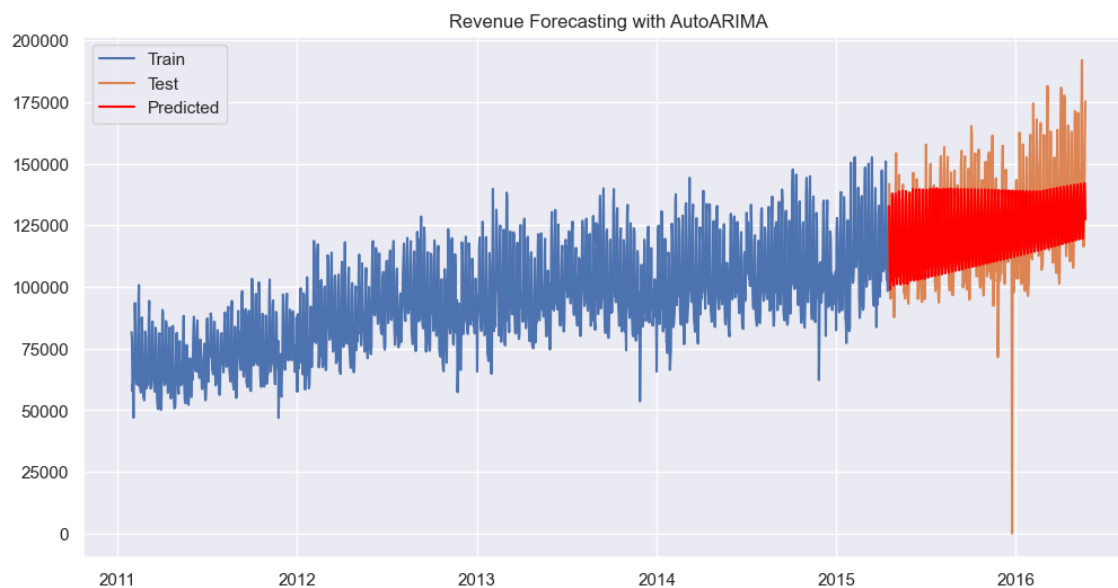


Fig 4. Test set predictions with ARIMA model

- SARIMAX also provides good performance on the test data, although it has a higher RMSE compared to ARIMA. The RMSE on the training data is closer to the test data, indicating better generalization.

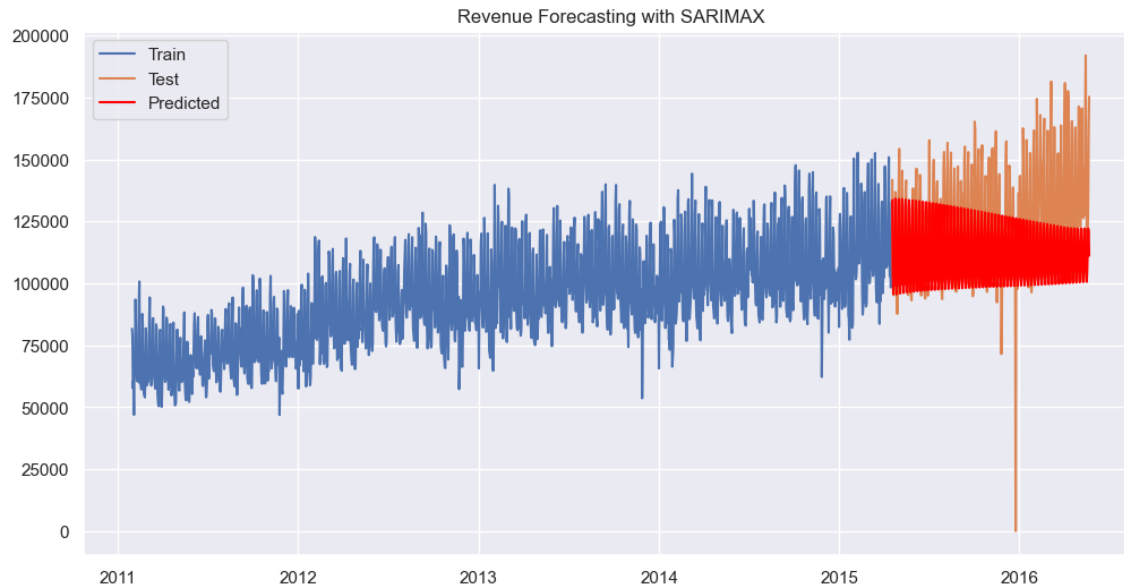


Fig 5. Test set predictions with SARIMAX model.

Overall, **ARIMA** currently stands out as the best-performing model, but it may require optimization to prevent overfitting. SARIMAX also performs well and could be fine-tuned for further improvement. Future experiments could explore ensembling techniques or incorporating additional exogenous features to enhance forecasting accuracy.

c. Business Impact and Benefits

The conducted experiments provide evidence supporting the performance of the predictive model in optimising inventory management, sales forecasting, customer satisfaction, operational effectiveness, and overall profitability. Nevertheless, it is imperative to engage in the process of fine-tuning the model, effectively addressing any outliers, and optimising the quality of the data. Overall, the results indicate that the model shows promise but may benefit from further refinement.

While the ARIMA and SARIMAX models show improvements over the baseline EWMA model, there is still room for enhancement in forecasting accuracy. Continual monitoring, model refinement, and potentially exploring advanced forecasting methods are essential for achieving more precise revenue forecasts, minimizing the impact of incorrect results on business operations, and maximizing revenue potential.



d. Data Privacy and Ethical Concerns

The project entails using pricing and sales information for a range of products and retailers. This data includes information about sales revenue, items, stores, and dates, but it may not contain sensitive personal data. The items, stores, departments, etc are encoded with IDs, respecting the privacy and confidentiality of business data. Other considerations:

- **Fairness and Bias:** It's critical to keep an eye out for any biases in the data or model predictions that might lead to unjust results. For instance, eliminating data with zero sales could unintentionally result in bias, so this problem needs to be carefully addressed.
- **Data Security:** To guard against illegal access, data breaches, and other security threats, data security is essential. To protect data, access controls, secure storage, and appropriate encryption should be put in place.
- **Transparency:** It's crucial to be transparent when developing and implementing models. It is important for users and stakeholders to comprehend the models' operation, the data that is being used, and the possible ramifications of the predictions made by the models.

To uphold data privacy, model interpretability techniques have been explored to ensure that model predictions can be explained and validated. This helps in identifying potential biases and ensuring fairness. In addition, data encryption and strict access controls are used to prevent unauthorised access to sensitive information. Data retention policies follow regulatory guidelines, ensuring that data is only kept for as long as is necessary. Furthermore, the project is subjected to ethics reviews, which evaluate potential biases and fairness concerns and address them thoroughly.



7. Deployment

Deployment process for this project included the following steps:

1. Building a docker container: In order to deploy the model and api to Heroku web application, we chose docker to push the built image to the web release.
2. Hosting the Models: A cloud-based platform is required to host the forecasting and prediction models that have been trained. Heroku was utilised in this instance for deployment.
3. Development of APIs: In order to offer endpoints for communicating with the models, APIs were developed. Users can submit data to these APIs and get predictions back in return.
4. API Documentation: Detailed information about each endpoint, anticipated input parameters, output format, and sample usage was produced in the comprehensive documentation for the APIs. The following are the accessible API endpoints:
 - / - An overview of the project's goals, a list of all project endpoints, the model's expected input and output formats.
 - /health/ - Status code 200 with a welcome message.
 - /sales/national/ - Returns next 7 days sales revenue forecast.
 - /sales/stores/items/ - Returns predicted sales revenue for an input item, sell price, store, and date.
5. Endpoint Testing: To make sure API endpoints operate as intended, thorough testing of each one was performed. This testing phase includes response validation, error handling, and input validation.

Considerations:

1. Added "handle_unknown" as "infrequent_if_exist" (parameter in One hot encoding) to `handle unknown categories` during prediction. By default it's set to "error", meaning it will rise error when the model is given any unseen value during transform.
2. I used the `dask dataframe` to make batches of data, it was working well in a different environment. But as soon as I tried to use it in poetry shell, there was some issue with the installation (No module named...). Could not resolve the problem due to time constraint. It can definitely save considerable memory while handling large datasets.

3. I performed batch processing to reduce the training time and memory usage while loading the model. With a batch size of 512, there were 3 batches. Here are some results of the ARIMA forecasting model with batch processing:

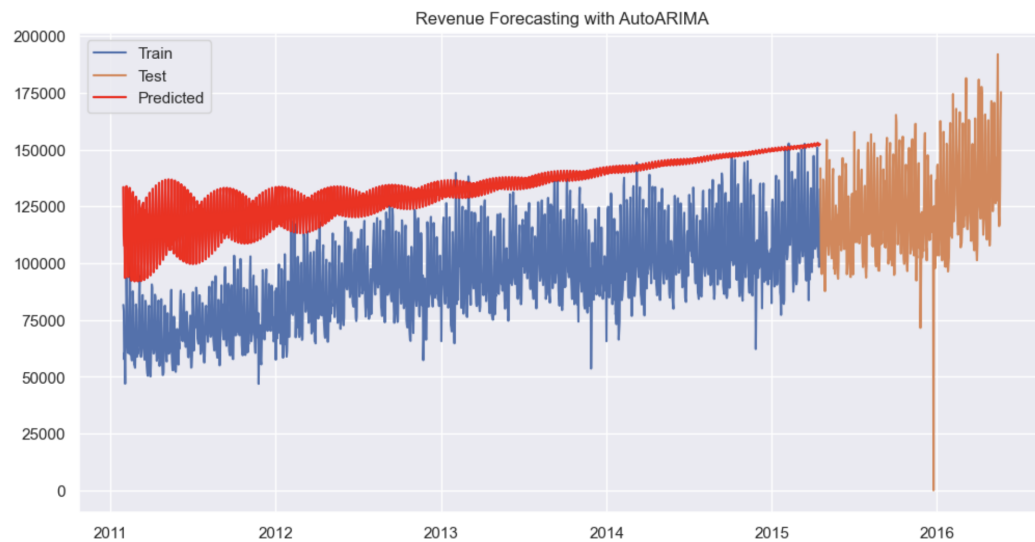


Fig 6. ARIMA predictions with batch processing on train set

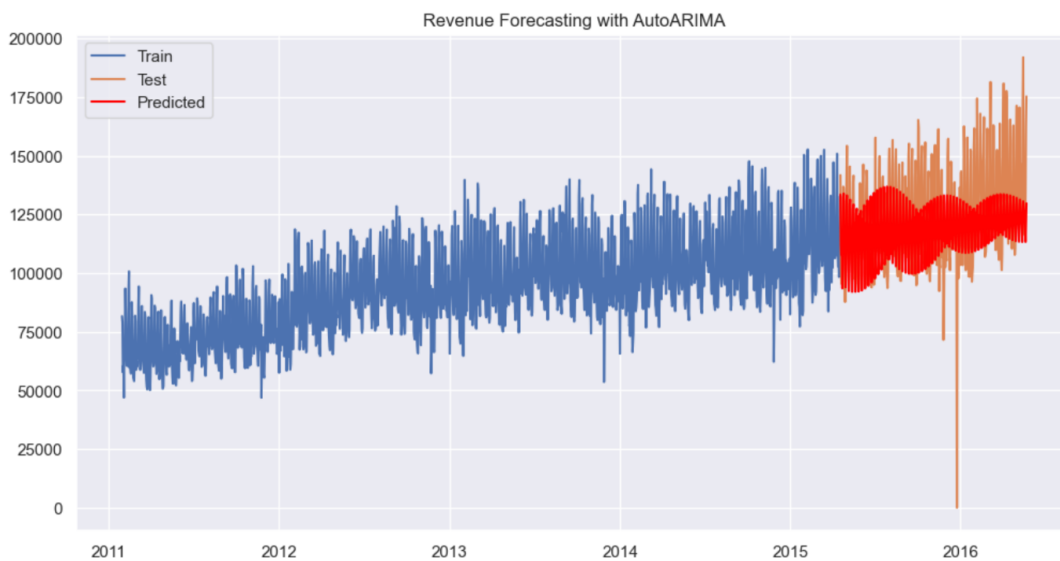


Fig 7. ARIMA predictions with batch processing on test set.

It has a nice curve dancing shape. I could not figure out the reason behind these results.

Challenges:

1. **Heroku deployment issue:** I was not and am still not able to deploy the model using the Heroku git commands, nor with GitHub. It says: "Your app does not include a heroku.yml build manifest". (Everything being in the right directory) Resolved: Deployed the app using Container Registry (Built docker first and then pushed the image to Heroku.) Ref: [stackoverflow](#), maybe there was an issue with the git LFS I setup.
2. **Heroku memory exceeded - Resolved:** Imported the forecasting model (size ~300mbs) inside the forecast function (only be loaded when forecast api is called) and implemented cache miss. So if the model was already present in the cache, it would not be loaded again.



8. Conclusion

The project effectively produced sales revenue prediction and forecasting models, meeting the requirements of retail industry stakeholders. The predictive model performed admirably, allowing for accurate sales revenue predictions for particular products and stores. On the other hand, the forecasting models offered insightful predictions about future trends in sales revenue for every store and item. ARIMA model was the best performer and demonstrated its potential for precise revenue forecasting.

These models can be used by stakeholders, such as business analysts and decision-makers, to help them make data-driven choices about sales tactics and inventory management. But the project also showed how important it is to deal with issues related to data quality, like handling outliers and zero sales, in order to improve the accuracy of the model. In the future, work should concentrate on continuous improvement through the addition of new features, the implementation of strong data quality controls, and the optimisation of deployment procedures. Furthermore, creating a feedback loop with users to get practical insights will be essential for improving and adjusting the models, which will ultimately guarantee their continued applicability and efficacy in the ever-changing retail environment.



9. References

- [1] McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).
- [2] 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, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- [3] Waskom, M., Botvinnik, Olga, O’Kane, Drew, Hobson, Paul, Lukauskas, Saulius, Gemperline, David C, ... Qalieh, Adel. (2017). mwaskom/seaborn: v0.8.1 (September 2017). Zenodo. <https://doi.org/10.5281/zenodo.883859>
- [4] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.
- [5] Pedregosa, F., Varoquaux, Gaël, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- [6] Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.
- [7] Joblib Development Team (2023) Joblib/joblib: Computing with python functions., GitHub. Available at: <https://github.com/joblib/joblib>.
- [8] Tiangolo/FASTAPI: FASTAPI framework, high performance, easy to learn, fast to code, ready for production, GitHub. Available at: <https://github.com/tiangolo/fastapi>
- [9] Encode/uvicorn: An ASGI web server, for Python. 🦄, GitHub. Available at: <https://github.com/encode/uvicorn>
- [10] Asteriou, D., & Hall, S. G. (2011). ARIMA models and the Box–Jenkins methodology. *Applied Econometrics*, 2(2), 265–286.