

Assignment 2

ML as a Service

Sidhant Bajaj
Student ID:25246568
4th October 2024

| | |
|-----------------|---|
| Github Username | sidhantbajaj |
| Github Repos | Experiment Repo: Link to repo Package Repo: Link to repo API Repo: Link to repo Streamlit Repo: Link to repo |
| URLs | Backend: Link to service Frontend: Link to service |



Table of Contents

| | |
|--------------------------------------|-----------|
| 1. Executive Summary | 2 |
| 2. Business Understanding | 3 |
| a. Business Use Cases | 3 |
| 3. Data Understanding | 4 |
| 4. Data Preparation | 5 |
| 5. Modeling | 6 |
| 6. Evaluation | 8 |
| a. Evaluation Metrics | 8 |
| b. Results and Analysis | 8 |
| c. Business Impact and Benefits | 8 |
| d. Data Privacy and Ethical Concerns | 9 |
| 7. Deployment | 10 |
| 8. Conclusion | 11 |
| 9. References | 12 |



1. Executive Summary

The objectives of this project were to analyse and extract insights from an American retailer's data comprising 10 stores across 3 states. Furthermore, develop a product that can identify key patterns and generate actionable results.

This product was designed to provide accurate forecasting and prediction of sales revenue based on specified input parameters. This project leveraged industry-level ML, MLOps and software development techniques to develop an end-to-end web service.

The outcome of this project was a web service that can display prediction and forecasting results in the frontend that are generated using ML models in the backend.





2. Business Understanding

a. Business Use Cases

This project focuses on conducting a comprehensive analysis of an American retailer, encompassing stores across three key states: California, Texas, and Wisconsin. The sales revenue forecast and prediction product based on these analyses can have value propositions for various use cases. Some of these are mentioned below:

- **Inventory Management:** An accurate sales forecast can enable a retailer to manage inventory across all stores. For instance, managing the limited refrigerated storage optimally is crucial. If not managed well, stockouts can lead to significant losses.
- **Demand Patterns:** A retailer benefits the most from understanding the demand patterns of its consumers. Major losses are incurred due to overstocking a product that is not in demand.
- **Price Adjustments:** Based on historical patterns, the retailer can adjust the pricing of the products dynamically to enhance profitability.
- **Marketing Strategies:** The model can identify sales trends, enabling the retailer to plan marketing events accordingly. This will provide the retailer with sufficient time to make informed decisions regarding their campaigns.

b. Key Objectives

This project aims to build a web service using the CRISP-DM framework and agile methodologies to forecast and predict sales revenue. The project's key aspect is the ML model and its performance will be evaluated using RMSE (Root Mean Square Error) metric. To maximise performance the metric needs to be minimised.

The web service comprises three main components: the frontend, backend, and hosting service. The frontend handles user interaction, input parameter ingestion, and result display. The backend facilitates communication between the ML models and the frontend via API calls. Once the workflow is established, the service will be deployed using a hosting service. The web service section outlines the concepts involved in executing these steps.

Stakeholder and Requirements

| Stakeholder | Role | Requirement | Strategy |
|--------------------|---|---|---|
| Retail Management | Oversees operations and strategic decisions | Accurate sales forecasts; insights for planning | Implement ML models for sales forecasting and reporting |
| Marketing Team | Plans and executes marketing campaigns | Understanding sales trends; data for campaigns | Utilize sales insights for targeted marketing strategies |
| Inventory Managers | Manages stock levels and storage | Inventory optimization; reduce stockouts | Integrate forecasting data to optimize inventory levels |
| Customers | End-users of the retail service | Access to products; timely availability | Enhance customer experience through optimized inventory and marketing |

Table 2.1: Stakeholder Analysis

The table 2.1 displays the stakeholders, their requirements and the strategies that can be employed to maximise revenue respectively. It also displays on how customer engagement can improved.

The end goal of this project is to combine all the techniques and leverage ML as a service. This can enhance clarity, efficiency, and impact, making it a valuable approach in the retail field.



3. Data Understanding

The dataset consists of three types of subsets including sales, calendar and pricing data. Their features composition and corresponding information are mentioned below:

| Dataset Type | Dataset Name | Numeric Features | Categorical Features | Shape |
|--------------|------------------------------|-----------------------------|--|----------------------------|
| Sales | sales_train.csv | d1, d2, d3...d1541 | id, item_id, dept_id, cat_id, store_id, state_id | 30490 rows 1547 columns |
| | sales_test.csv | d1542, d1543, d1544...d1941 | N.A. | 30490 rows 400 columns |
| Calendar | calendar.csv | wm_yr_wk, d | date | 1969 rows 3 columns |
| | calendar_events.csv | N.A. | date, event_name, event_type | 167 rows 3 columns |
| Pricing | items_weekly_sell_prices.csv | wm_yr_wk, sell_price | item_id, store_id | 6841121 rows 3 columns |

Table 3.1: Dataset Composition for all datasets

To extract relevant information from the subsets mentioned in the data composition table, the sales, calendar and pricing sets were merged together to form single dataset containing all the information. This procedure will be explained in detail in the Data Preparation section.

The data composition for the combined dataset is mentioned below:

| Dataset | Numerical Features | Categorical Features | Shape |
|---------|---|---|-----------------------------|
| Train | units_sold, wm_yr_wk, sell_price, sales_revenue | id, item_id, dept_id, cat_id, store_id, state_id, d, date | 46985090 rows 12 columns |
| Test | | | 12196000 rows 12 columns |

Table 3.2: Data composition for combined sets

The combination of subsets generated the target variable “sales_revenue”, created by multiplying units sold and their sell price per day. The average sales revenue for all the items is **\$4.10** across all stores.

The dataset contains multiple ID columns. The ‘id’ column is the unique identifier for all the items sold in a specific store on a specific date. The rest of the ID columns are mentioned below:

| ID | Role | Unique Value Counts | Format Example |
|----------|--|---------------------|---|
| item_id | Items sold across all the stores | 3049 | HOBBIES_2_200 HOUSEHOLD_1_465 |
| dept_id | Item departments | 7 | FOODS_3 HOUSEHOLD_1 |
| cat_id | Item categories | 3 | FOODS, HOBBIES |
| store_id | Retail stores in California, Texas and Wisconsin | 10 | CA_1, CA_2, , TX_1, TX_2, WI_1, WI_2 |
| state_id | States in which stores are present | 3 | CA, TX, WI |

Table 3.3: ID columns

State ID

The retail stores are spread across 3 states including California, Texas and Wisconsin as shown in Fig 3.1 with their distribution.

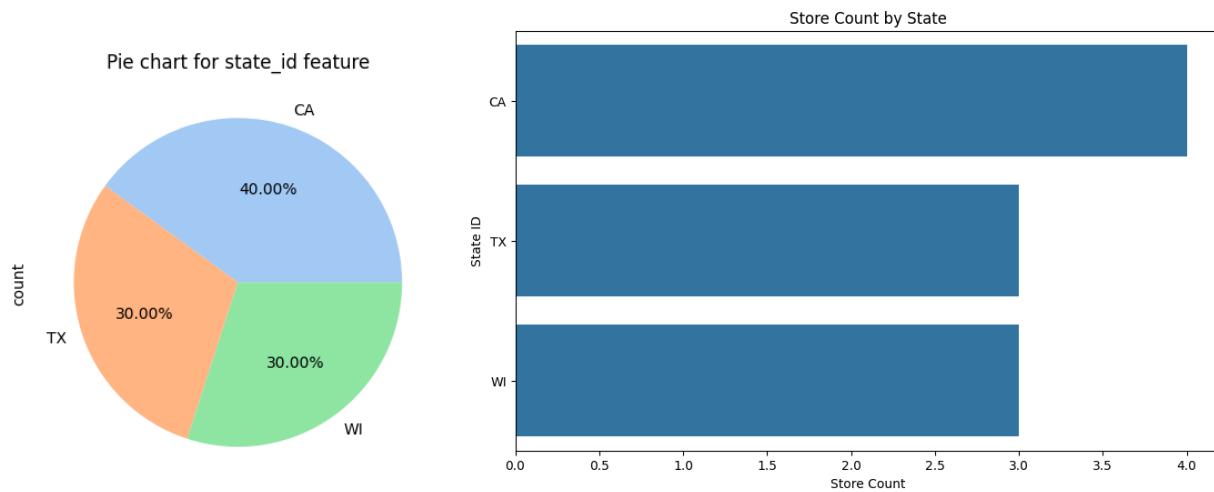


Figure 3.1: State distribution of stores

Fig 3.2 below reveals the total sales revenue generated from different states throughout the years with California in the lead, followed by Texas and Wisconsin.

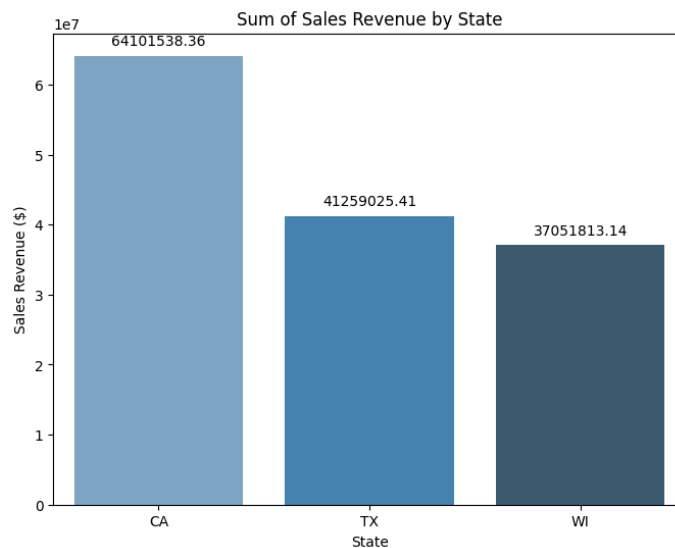


Figure 3.2: State sales revenue

Fig 3.3 depicts the record of sales revenue over the years. It is evident that 2011 was the year with highest sales revenue for all the stores and then a decline trend in a sales revenue can be observed till 2014. After 2014, the sales revenue takes a turn from the decline to an uptrend for TX and WI, except for CA.

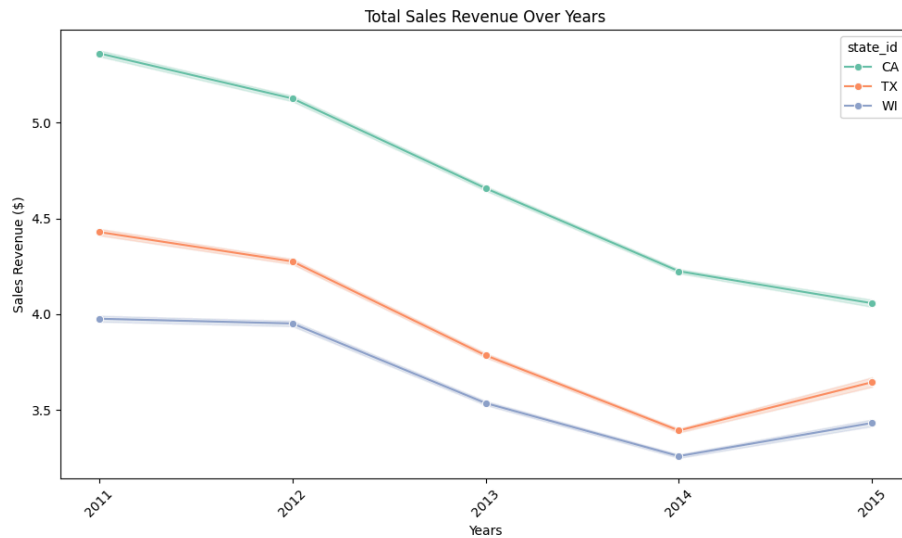


Figure 3.3: Line graph for state sales revenue

Store ID

There are 10 stores that are spread across 3 different states. The store composition is shown in Fig. 3.1 with highest number of stores in California. The distribution of the revenue generated over the years from all the stores can be seen in Fig 3.4. CA_3 store performs the best in California, TX_2 in Texas and WI_3 in Wisconsin. The performance of all the stores throughout the years can be evaluated through Fig. 3.5.

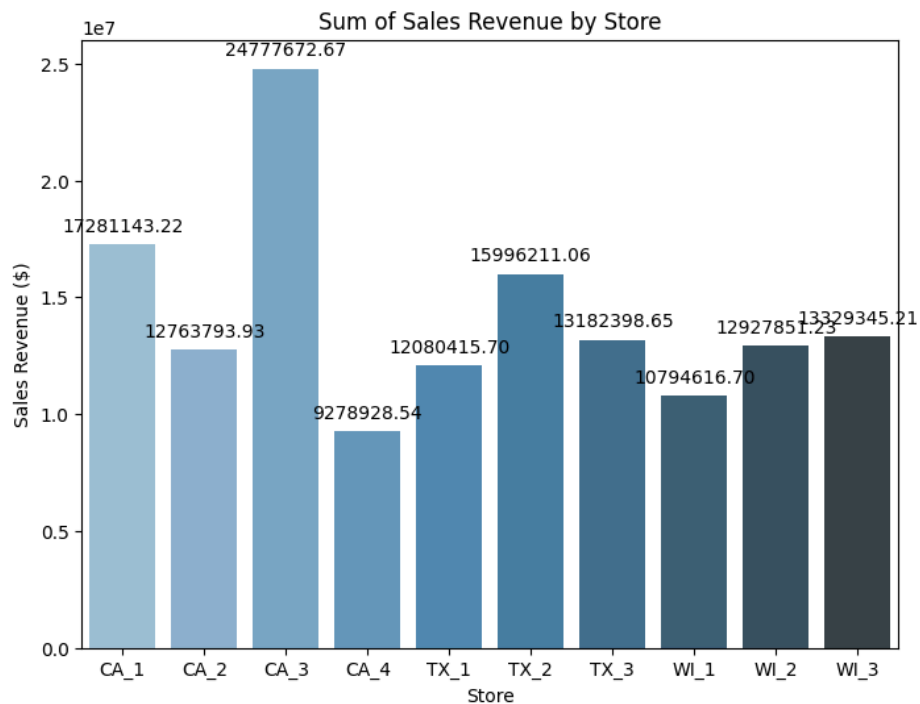


Figure 3.4: Store based sales revenue

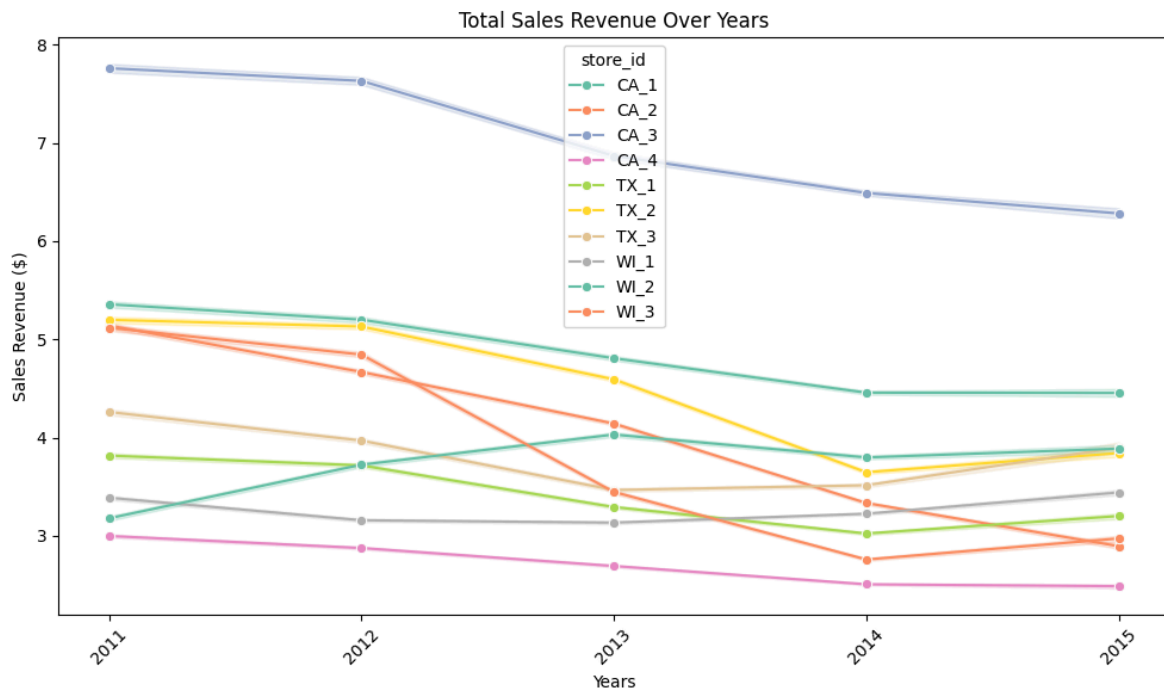


Figure 3.5: Line graph for sales across all stores

Category ID

There are 3 different categories including food, hobbies and household. The pie chart in the Fig 3.6 shows the distribution of item categories that are present in the data. The majority of the items belong to food category.

The graph in Fig. 3.7 gives an idea about the item pricing in each store based on the categories. As expected, household items have a higher median value and food has the lowest. This could be one of the reasons why food category has more items and the sizing of the food items is small as well as compared to housing.

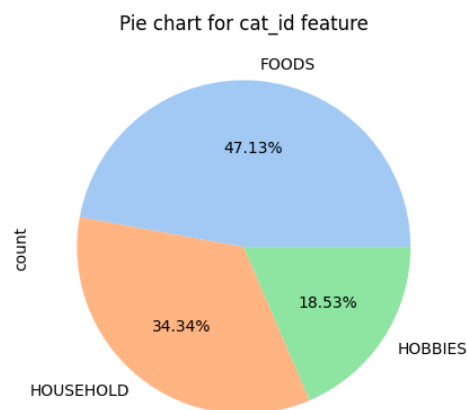


Figure 3.6 Distribution of categories



Figure 3.7: Pricing distribution of categories in all stores

Department ID

There are 7 department IDs in total which are created on the basis of the categories present in the data. The Fig. 3.8 below shows the revenue generated from all the departments with FOODS_3 in the lead.

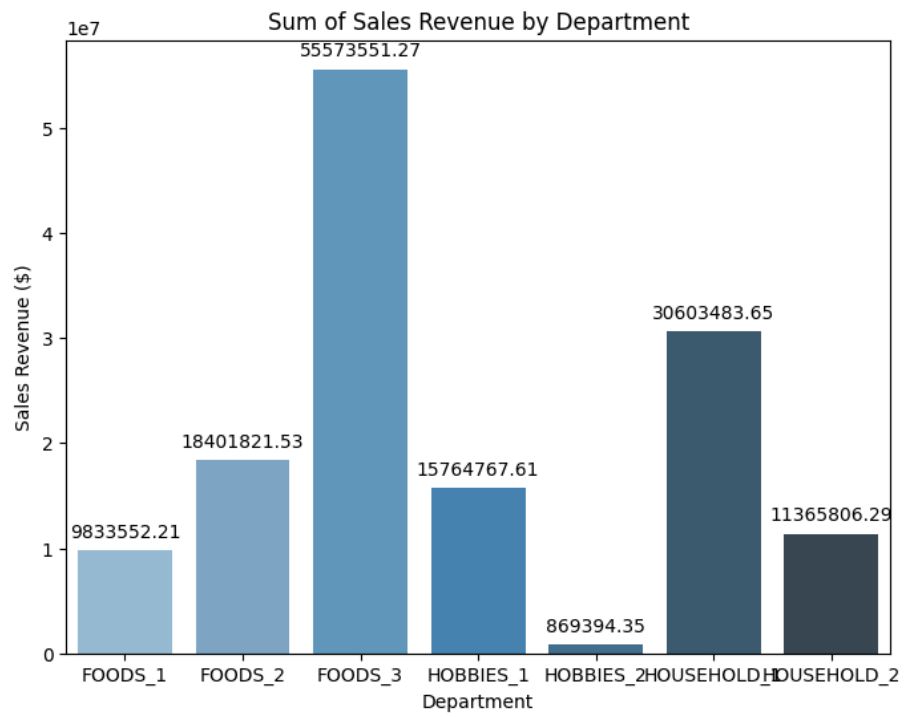


Figure 3.8: Department sales revenue

4. Data Preparation

The initial step for data preparation as mentioned in the above section was to combine the datasets together and generate the target variable 'sales revenue'. This was done using a series of steps:

- **Wide to long format conversion:** The 'd' related features are changed to rows from columns to reduce the sparsity and increase the interpretability of the dataset. This results in the removal of 1541 features and the addition of two features: 'd' and 'units_sold'.
- **Merging subsets:** Merged the sales data with calendar data on the 'd' feature to get the dates-related features. Merged this dataset again with a selling price subset to add the price for each item.
- **Sales Revenue:** Generated the target variable sales revenue by multiplying units sold with the selling price for each item on a particular day.

Missing Values


Next in the process was to search for missing values. Fig4.1(a) and Fig4.1(b) below show the count and the percentage of missing values in the train and test sets respectively. The features having missing values are the same in both sets. The units sold feature revealed that these values were missing as the items were not sold on those days. Thus, these values were removed using the data-cleaning pipeline discussed below.

| | Missing Values | % Missing Values | | Missing Values | % Missing Values |
|---------------|----------------|------------------|---------------|----------------|------------------|
| sell_price | 12264399 | 26.1 | sell_price | 35014 | 0.3 |
| sales_revenue | 12264399 | 26.1 | sales_revenue | 35014 | 0.3 |

Data Cleaning

For this part, a data cleaning pipeline was created using a cleaning transformer that caters to specific requirements. Since similar cleaning was to be applied to the train and test set, this pipeline helped reduce code and effort. The steps involved are:

1. Creating a transformer class.
2. Input Parameters: Defined the input parameters such as the datetime_col variable and a list of columns that need to be removed.



3. Dropped missing values.

4. Extract year, month and day from datetime column and create separate columns.

5. Dropped unwanted columns such as 'id' (unique identifier), 'date' (since date has been expanded), 'wm_yr_wk'(already have time-related feature), 'd' and 'units_sold' (sales revenue is created using units sold).

After the cleaning process, the train set had 34720691 rows and 9 columns. Similarly, test set had 12160986 rows and 9 columns. This data was further transformed for model usage. The transformation step was divided into two parts as two models were to be created; forecast and predictive.

Data Transformation (Forecast Model)

The forecast model requires the data to have two columns; ds (containing dates) and y (containing target variable). In the case of the forecast model, the target variable is the sales revenue of all items across all the stores on a single day. The process mentioned below is applied to both the test and train sets.

The column 'date' is created by combining the day, year and month columns and converting them to datetime format. The data is grouped on dates and the sum of sales revenue is aggregated for each date. Finally, the date and sales revenue columns are renamed to 'ds' and 'y' for model fitting.

The holiday data is also used by the model as a parameter in which the date column is renamed to 'ds' and the event name is renamed to 'holiday'.


Date Transformation (Prediction Model)

A couple of techniques were applied to the dataset for the predictive model as there were 6 categorical columns and 3 unscaled numerical values.

The 'item_id' variable was transformed to numerical using **Target Mean Encoding**. This concept handles the categorical variables by replacing each category with its corresponding mean value of the target variable.

Variables including 'dept_id', 'store_id', 'cat_id' and 'state_id' were converted to numerical using **Ordinal Encoding**.

Finally, all the variables were scaled using **Standard Scaler**.



Dataset sampling

Ran a few tests on un-tuned models such as Linear regression, Elasticnet, Decision trees, AdaBoost, XGBoost and LightGBM models. Using samples such as 20%, 40% and 60% of the dataset, verified the size which made the models perform the best. Finally, 60% of the training set gave the best results which were chosen.

Dataset Splitting

The dataset was split into training and validation sets using a 7:3 ratio.

■ ■ ■

5. Modeling

a. Predictive Models

Three different predictive models were tested for this project. Multiple experiments were conducted on these models to find the best model for this task. The rationale and hyperparameters tuned in each model are mentioned below:

| Model | Description | Rationale | Hyperparameter(s) Tuned |
|--------------------------|--|---|---|
| Linear Regression | Linear regression aims to find the values that reduce the sum of squared errors between the predicted values and the actual values in the training data. | It is simple to implement, requires minimal tuning and helps in understanding whether there is a linear relation between the dependent and independent variables. | fit_intercept (True or False) which allows the model to use an intercept |
| Decision Tree | Decision support hierarchical model that uses a tree-like structure to make predictions about the target variable based on the input features. | It is simple and interpretable. Captures complex relationships between features and the target variable. It also generates a feature importance graph. | max_depth deepens tree structure (reduces overfitting). Min samples split prevents the model from creating unwanted nodes. min_samples_leaf sets the number of leaf samples (reduces overfitting) |
| XGBoost | It employs an ensemble learning technique based on decision trees, utilizing gradient boosting to optimize performance. | The algorithm minimizes prediction errors by sequentially training weak learners, thus improving accuracy and robustness. | learning_rate controls the step size at each iteration while moving toward a minimum. max_depth of the trees. Increasing depth increases model complexity. gamma sets the minimum loss reduction to make leaf partition alpha and lambda alpha are L1 and L2 regularization. min_child_weight is the min-sum of instance weight in a child. sub_sample is a fraction of samples to use for each tree |

Table 5.1: Predictive model Information

b. Forecast Model

The forecast model tested in this project is facebook's **Prophet**. The application of this model in this project is to forecast sales revenue for 7 days in the future. The model information is shared in the table below.

| Model Description | Rationale | Hyperparameter Tuned |
|--|---|--|
| Prophet is a forecasting tool developed by Facebook that is designed to handle time series data. It is particularly useful for data with clear seasonal patterns and historical trends, such as daily, weekly, or yearly fluctuations. | <p>It can capture seasonal trends such as yearly or weekly patterns, which is useful for business metrics like sales, web traffic, or demand forecasting.</p> <p>It is robust to outliers and can adjust them without significant impact.</p> <p>Users can easily specify custom holidays, events, and include additional regressors to improve the forecasting model.</p> <p>Its component based structure allows users to generate graphs making it more interpretable.</p> | Holidays hyperparameter allows you to incorporate known holidays or special events into your forecasting model. |

Table 5.2: Forecast model information

6. Evaluation

a. Evaluation Metrics

The evaluation metric used to assess both models' performance is **RMSE** (Root mean square error). The detailed information is contained in the table below.

| Aspect | Description |
|--------------------------------------|--|
| RMSE definition | Measures the average magnitude of errors between the predicted and actual values. |
| Rationale for sales revenue use case | Helps to identify how close the predicted revenue is to the actual revenue. This is critical for making informed business decisions. |
| Advantages for prediction models | <ul style="list-style-type: none">- Easy to understand how well sales predictions align with actual revenue.- Highlights larger errors that could impact sales forecasts and strategies.- Allows comparison between different models based on their predictive accuracy. |
| Advantages for forecast model | <ul style="list-style-type: none">- Essential for evaluating how well models predict future sales revenue based on historical data.- Provides a consistent measure to assess different forecasting methods.- Helps in recognizing deviations from expected sales trends over time. |

Table 6.1: RMSE analysis

b. Results and Analysis

Prediction Model Results

| Model (Best Model) | Set Type | RMSE | Findings |
|--------------------|------------|-------|---|
| Linear Regression | Training | 8.81 | <ul style="list-style-type: none">- The final hyper-parameters used were 'fit_intercept': True.- The validation set shows that model performs well for the task.- However, the test score revealed that the model is overfitting and it might not be appropriate for this task. |
| | Validation | 8.75 | |
| | Testing | 9.75 | |
| Decision Tree | Training | 4.44 | <ul style="list-style-type: none">- The training result provides a high quality result.- The validation and testing score verify that the model performs poorly on unseen data with very high overfitting.- Multiple hyperparamters were employed to increase the model's performance, however, it was futile. |
| | Validation | 9.04 | |
| | Testing | 10.58 | |
| XGBoost | Training | 7.97 | <ul style="list-style-type: none">- The training and validation sets revealed that the model performs well on unseen data with very almost no overfitting.- The testing score was the closest to validation and training as compared to the rest of the model.- The hyperparameter that helped achieve this score was max_depth=7.- This model was finally selected for usage due to its high performance. |
| | Validation | 7.99 | |
| | Testing | 8.59 | |

Table 6.2: Predictive model results

The decision tree assisted in understanding the underlying identifiers that are being captured by the model. A feature importance graph was generated to get a better understanding shown in Fig. 6.1.

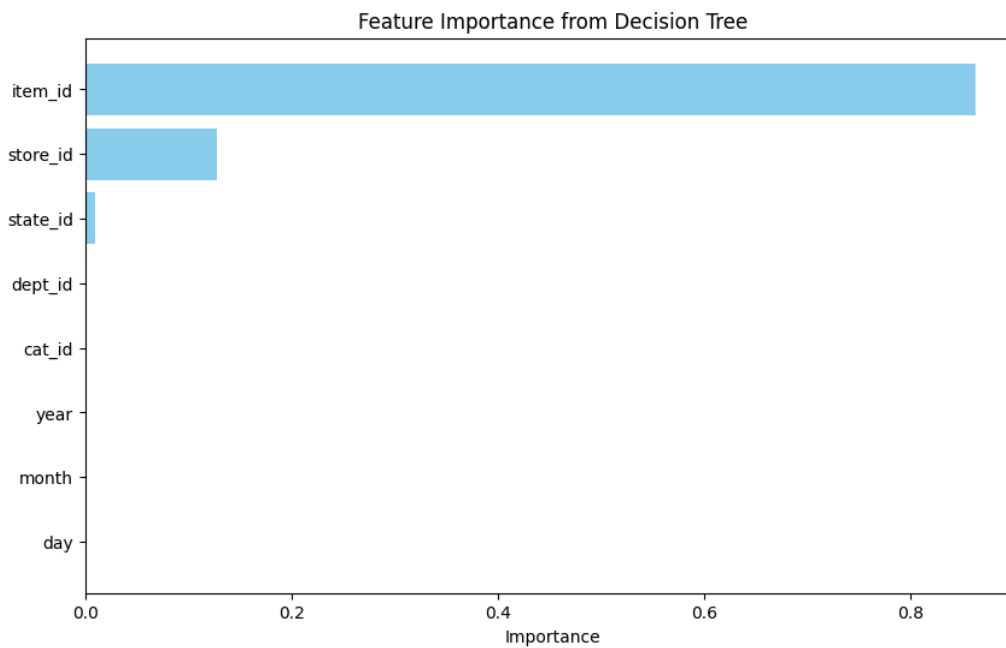


Figure 6.1: Feature importance graph

Forecast Model Results

| Models | Set Type | RMSE | Findings |
|-----------|------------|----------|--|
| Prophet 1 | Training | 8658.15 | <ul style="list-style-type: none"> - This model was run on default parameters which gives satisfactory results. - Slight overfitting can be identified. |
| | Validation | 10330.04 | |
| Prophet 2 | Training | 6795.62 | <ul style="list-style-type: none"> - This model is training using Holiday parmeter. - The training and validation are better than the default model. - This model was selected. |
| | Validation | 9096.95 | |
| | Testing | 14415.56 | |
| Prophet 3 | Training | 6795.62 | <ul style="list-style-type: none"> - This model was trained using cross validation technique. - The results were similar to pervious model. |
| | Validation | 9096.95 | |

Table 6.3: Forecast model results

The forecast model generated component graph which help identify the trend line the model is basing its results on. Fig. 6.2(a) and Fig. 6.2(b) display the forecast results for validation and test set. The shaded region shows the forecasted data points in both the cases.

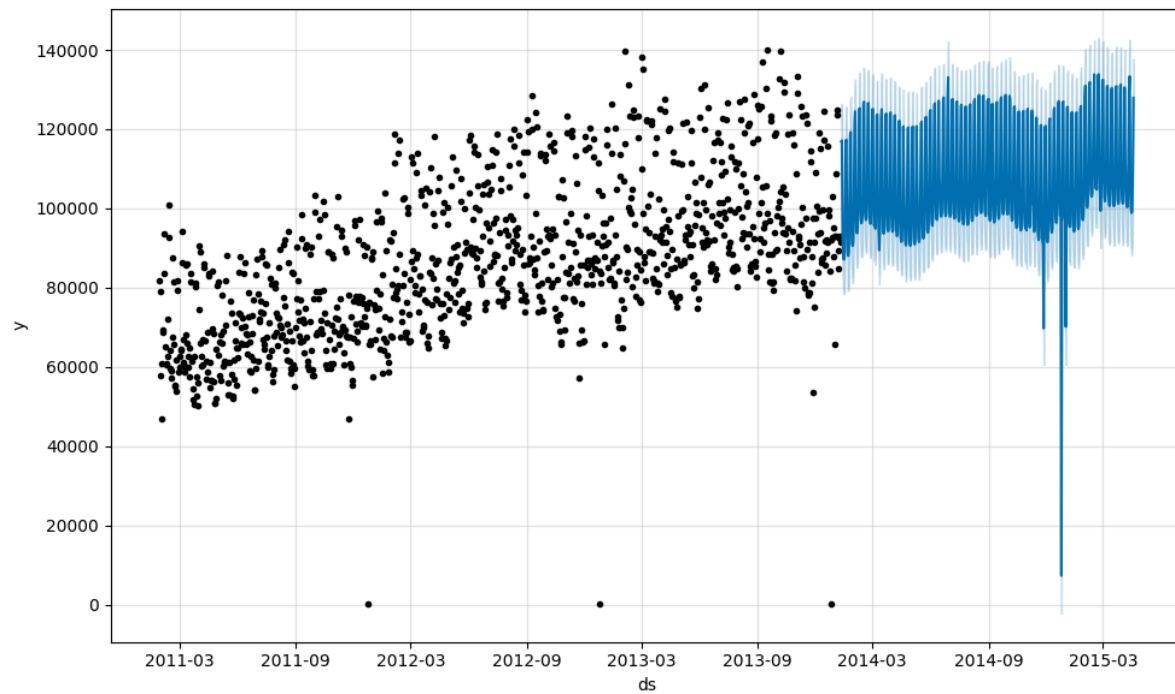


Figure 6.2(a): Forecast graph for validation set

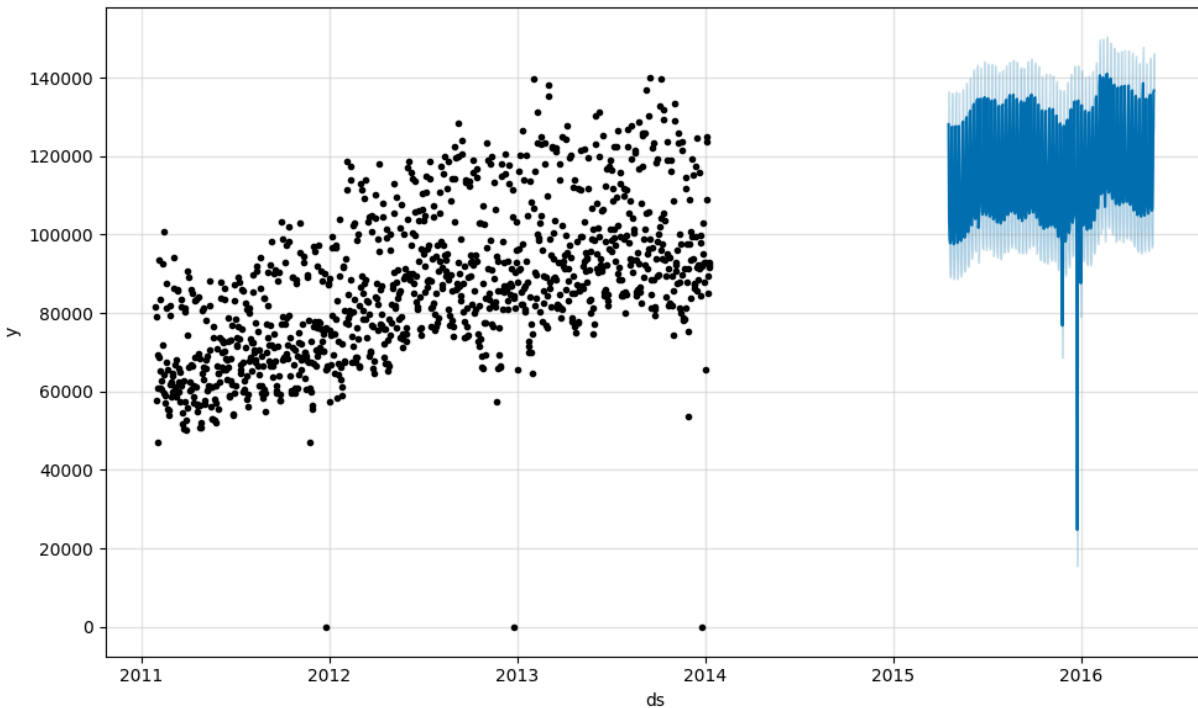


Figure 6.2(a): Forecast graph for test set

c. Business Impact and Benefits

XGBoost Model for Sales Revenue Prediction


The XGBoost model enables the organisation to generate efficient and precise sales revenue prediction, which will result in informed-decision making. This will lead to improved profitability and resource allocation.

This model is capable of consistently providing high accuracy scores, essential for making the right decision that can cause significant impact. The strategy formulation can be highly benefited from such a model.

The model's lightweight and low complexity attributes are capable of handling large datasets that can facilitate scalability in the future.

Prophet Model for Sales Revenue Forecast

The Prophet model enables the organisation to peek into the future due to the reliable sales forecast. This will assist in capturing market trends and seasonal patterns that can help grow the sales revenue.



Retail stores have very high variation in sales during specific timings in a year. These trends if not captured appropriately can lead to significant losses. This model allows us to share such points in time to the model so that strategies can be planned out.

d. Risks and Incorrect Predictions

Financial Losses: If the model predicts higher sales revenue than actual, the resources allocated for that time period will be wasted due to wrong assumptions. Conversely, if lower sales are predicted than actual, then it can cause inadequate resource allocation leading to losses.

Inventory Management Issues: Overstocking based on incorrect predictions can increase wastage and storage costs. Conversely, understocking can lead to reduced customer satisfaction due to the unavailability of the products.

Poor Strategy Planning: Organisations can enter new markets or discontinue a product line which might be based on faulty revenue forecasts, resulting in poor strategic planning and misallocation of resources.

e. Data Privacy and Ethical Concerns

There are several important privacy and ethical concerns to consider regarding this model. This includes how the sensitive data of the customers is handled, work around potential data bias and deploying the model ethically in the real world.

Data Privacy Implications


Customer Loyalty Plans: Many organisations create customer loyalty plans which customers can be a member of. This membership requires them to add their personal information. This personal information can be attacked, therefore data storage firewalls need to be placed to keep the data safe.

Data Security: Data breach and unauthorised access might happen so the data must be encrypted with secure storage methods.

Ethical Concerns

Bias in data: The dataset may reflect biases such as wrong item pricing which could skew the predictions and cause unfavourable predictions.

Impact on Customers: This can cause customer to bear losses and lead to the loss of customer trust.



Steps to ensure Data Privacy and Mitigate Ethical Concerns

Anonymisation and Data Minimisation: Remove or anonymize identifiable information before sharing the dataset or deploying the model to limit the risk of data misuse.

Fair and Ethical Model Deployment: Ensure that the model is deployed in such a way that it compliments human decision-making rather than completely ignoring it.



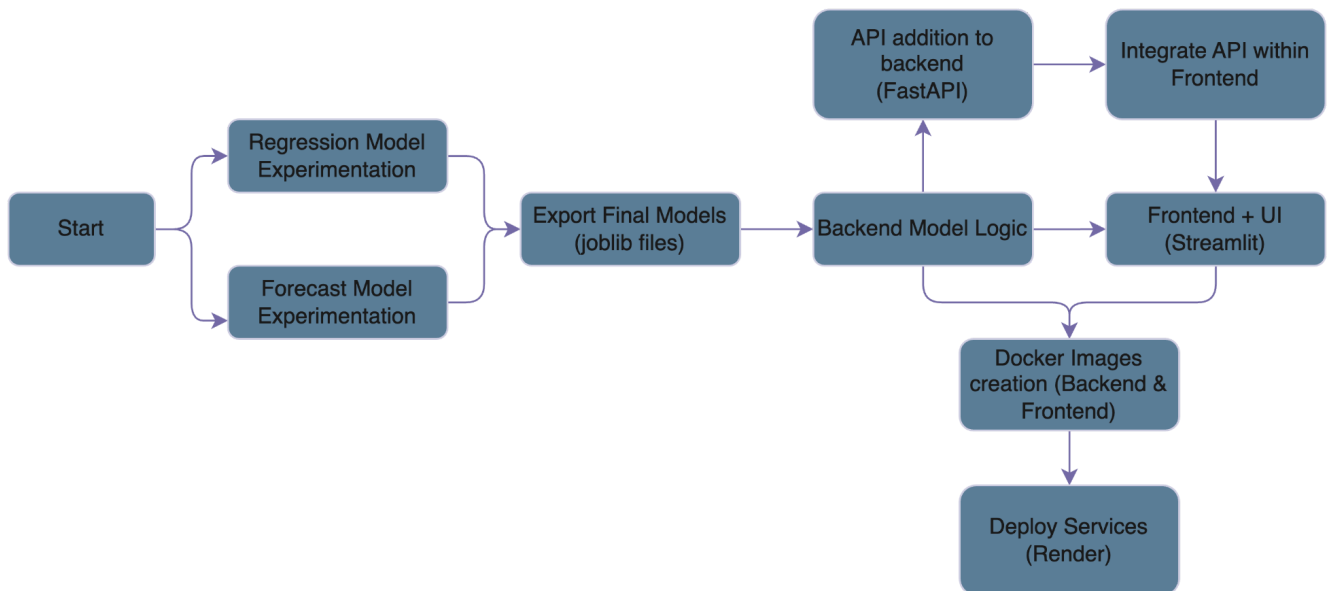
7. Deployment

Deployment Process, Integration Steps and Considerations

The deployment stages are shown below in the flow chart. The initial stage is to experiment the models and export the final models into joblib files. The web-service consists of two major components; frontend and backend. The backend component comprises the model logic and APIs for interaction with frontend. The APIs were created using FastAPI framework. The frontend which is the face of the service interacts with the user, ingests data, relays the input to the models using the backend API and finally displays the results received from the backend service. Both these service are made available by using the combination of docker (for image creation) and Render (hosting web service).


Backend url - <https://sales-revenue-backend-latest.onrender.com>

Web service with frontend url - <https://sales-revenue-app.onrender.com>



APIs created for the model are:

- `√` (GET): Displaying a brief description of the project objectives, list of endpoints, expected input parameters and output format of the model, link to the Github repo related to this project

- 
- `/health/` (GET): Returning status code 200 with a string with a welcome message of your choice`
 - `/sales/national/` (GET): Returning next 7 days forecasted sales (in dollars) for the following expected input parameters:`
 - ``date``: date from which the model will predict the forecasted sales for the following 7 days (excluding the input date). The expected date format is YYYY-MM-DD. - `/sales/stores/items/` (GET): Returning predicted sales (in dollars) for the following expected input parameters:`
 - ``date``: date from which the model will predict the sales on. The expected date format is YYYY-MM-DD.
 - ``store_id``: identifier of the store from which the model will predict the sales on.
 - ``item_id``: identifier of the item from which the model will predict the sales on.

These APIs can be accessed by using the backend url link or can be tested using the web service link.





8. Conclusion

The development of an end-to-end web service incorporating both regression and forecasting models suggest that the project has met its goal. By deploying these models, I have not only fulfilled stakeholder expectations but have also established a robust framework that can adapt to evolving business needs. Continuous testing and validation of model precision are essential to ensure sustained performance and to maintain the trust of stakeholders in the accuracy of predictions. The utilisation of Machine Learning algorithms such as XGBoost and Prophet can enhance organisation's operations and capture market trends proactively.

The project led me to understand the basics of regression and forecasting, which can be applied to various domains. With that understanding, an additional layer of model deployment assisted me in understanding the technologies employed in the industry. Overall the project has provided a realistic hands-on experience that elevates our skills as data scientists.

