

# Assignment:2

# **ML as a Service**

---

Name: Asmita Sheshrao Kamble

Student ID: 14373753

10/10/2023

Git link: [https://github.com/asmitaskamble/AT2\\_ML-as-a-Service](https://github.com/asmitaskamble/AT2_ML-as-a-Service)

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

## Table of Contents

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

## 1. Executive Summary

The goal of the project is to development of two different machine learning models and the deployment of the models to Heroku.

The first task is using a Machine Learning algorithm, predict sales revenue for a given item in a specific store at a given time using a model and second task is to forecast the total sales revenue for all stores and items for the next seven days by using a time-series analysis algorithm. After that, we can analyse which model is going to perform better based on your results.

This data pertains to an American retailer with 10 stores located in three different states: **California (CA), Texas (TX), and Wisconsin (WI)**. Each shop sells items in three categories: **hobbies, food, and household** items.

The results of the project indicate that the Random Forest model performs better than both Linear Regression and SARIMAX when it comes to forecasting sales. "Additional optimization of Random Forest is recommended in order to achieve the project's objectives."



## 2. Business Understanding

Listed below are the business use cases and motivating factors based on challenges and opportunities

### a. Business Use Cases

❖ The following are some key scenarios in which these models can be utilized:

- Predicting the lifetime value of customers
- Analyse the performance of your store
- Budgeting and financial planning
- Planned staffing and scheduling
- The marketing department
- The price optimization process
- A demand forecast
- A personalized marketing strategy
- An online recommendation system

### Motivating factors for the project

❖ The following are a few challenges encountered while working on it that helped us.

**Integrating data:** Integrating data from multiple sources (sales data, external factors, etc.) can be tricky. Making reliable predictions and forecasts with inaccurate or incomplete data is challenging.

**Data Size and complexity:** Managing and analysing data from multiple stores, items numbers, and categories across different states is tough. It's crucial to build models that can handle this complexity.

**Maintaining the model:** The accuracy of predictive and forecasting models depends on ongoing maintenance and retraining. Changes in business conditions and new data can impact model performance.

❖ Following are the opportunities out of the challenges of this project

- By utilizing accurate predictive models, retailers are able to reduce costs and increase profitability by optimizing inventory, pricing, and staffing.

- It is possible to improve customer service and personalize the experience by understanding demand patterns
- The implementation of predictive and forecasting models allows companies to make data-driven decisions, reducing the reliance on intuition and increasing the precision with which strategic decisions are made.
- Retailers can stay ahead of trends and competitors by leveraging data-driven insights.

## b. Key Objectives

### key objectives/goals of the project:

Following are the goal of the project to build model:

- Achievable outcome: Improve the accuracy of revenue predictions for specific items in specific stores.  
Goal: It goals to reduce stockouts and overstock situations by utilizing machine learning algorithms.
- Achievable outcome: Enhance customer service by aligning staff schedules.  
Goal: Optimizing staffing schedules, reducing labor costs, and ensuring high customer satisfaction will be made easier with machine learning models.
- Achievable outcome: Optimize revenue and profit by developing a data-driven pricing strategy.  
Goal: A machine learning algorithm analyzes various factors (e.g., item, category, location, date etc) to suggest optimal pricing strategies.

### the stakeholder requirements

- Provide accurate sales forecasts, demand forecasts, and pricing recommendations to optimize inventory levels, staffing levels, and pricing.
- Predictions and recommendations for staffing specific to each store to ensure the efficient operation of the store.
- A clear understanding of which products are likely to perform well and when to run marketing campaigns based on these insights.

- 
- The ability to forecast sales accurately for financial planning and budgeting.

#### Explanation on how the project has addressed the requirement

- In order to provide accurate sales and demand forecasts based on the historical sales data and external factors, we analyze historical sales data and external factors.
- Providing marketing advice based on the prediction of high-performing items and appropriate timing for their marketing campaigns.
- Preparation of reliable sales forecasts for budgeting and financial planning purposes.

■ ■ ■

### 3. Data Understanding

The project contains following datasets

✚ List of tables and its number of columns in the project:

1. **Sales\_train**: - id, item\_id, dept\_id, cat\_id, store\_id, state\_id, d\_1 to d\_1541
2. **Sales\_test**: - d\_1542 to d\_1941
3. **Items\_weekly\_sell\_prices**: store\_id (CA\_1, CA\_2),  
item\_id(FOODS, HOBBIES, HOUSEHOLD)  
wm\_yr\_wk (11101 to 11621)  
sell price (0.01 to 30.98)
4. **calendar\_events**: date, event\_name, event\_type
5. **calendar**: date, wm\_yr\_wk, d

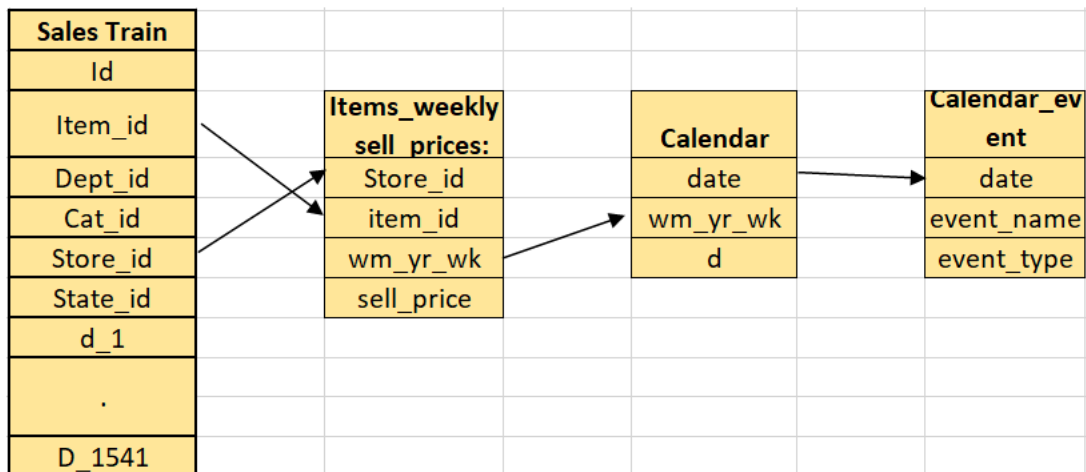


Fig: Data Merge Pattern

## Data Merging

The dataset is merged in chunk\_sizes assigned 100.

```
# chunk_size = 100
```

The sequence is following:

Chunk1: 'train\_sales' and 'item\_weekly\_sales\_prices' on 'item\_id' and 'sales\_id'

Chunk2: chunk1 and 'calendar' on 'wm\_yr\_wk'

Chunk3: chunk2 and 'calendar\_event' on 'date'

At the end we got the merged data where all the data is merged in one dataframe

Out[39]:

	id	item_id	dept_id	cat_id	store_id	state_id	wm_yr_wk	sell_price	date	d	event_name	event_type	sales	total_revenue
s_3_737_WI_3_evaluation	FOODS_3_737	FOODS_3	FOODS	WI_3	WI	11401	3.48	2014-02-01	1100	Ramadan starts	Religious	208.80	208.80	
s_3_737_WI_3_evaluation	FOODS_3_737	FOODS_3	FOODS	WI_3	WI	11401	3.48	2014-02-02	1101	SuperBowl	Sporting	208.80	417.60	
s_3_737_WI_3_evaluation	FOODS_3_737	FOODS_3	FOODS	WI_3	WI	11401	3.48	2014-02-03	1102	Ramadan starts	Religious	208.80	626.40	
s_3_737_WI_3_evaluation	FOODS_3_737	FOODS_3	FOODS	WI_3	WI	11401	3.48	2014-02-04	1103	Ramadan starts	Religious	208.80	835.20	
s_3_737_WI_3_evaluation	FOODS_3_737	FOODS_3	FOODS	WI_3	WI	11401	3.48	2014-02-05	1104	Ramadan starts	Religious	208.80	1044.00	



## Data Workflow

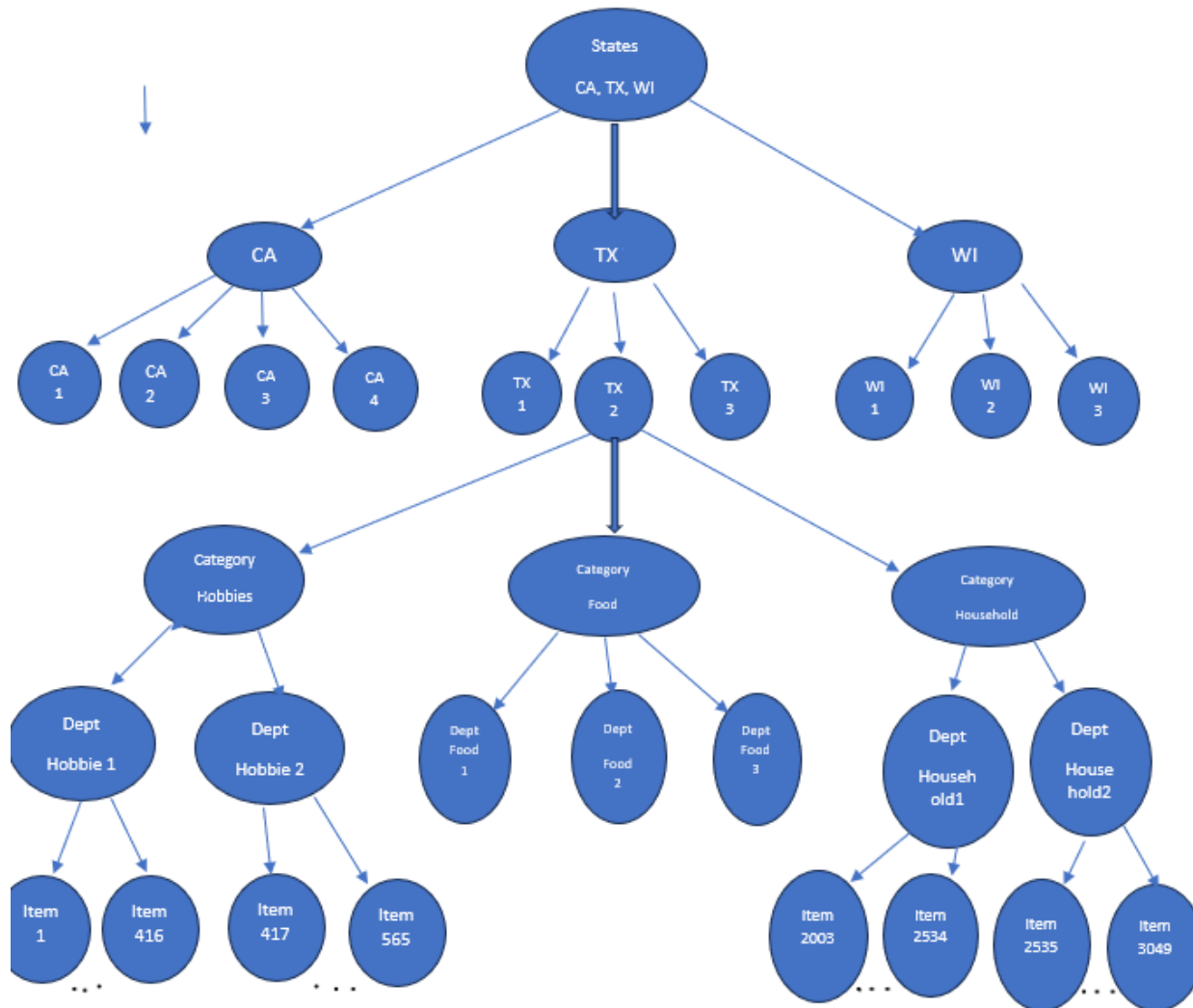
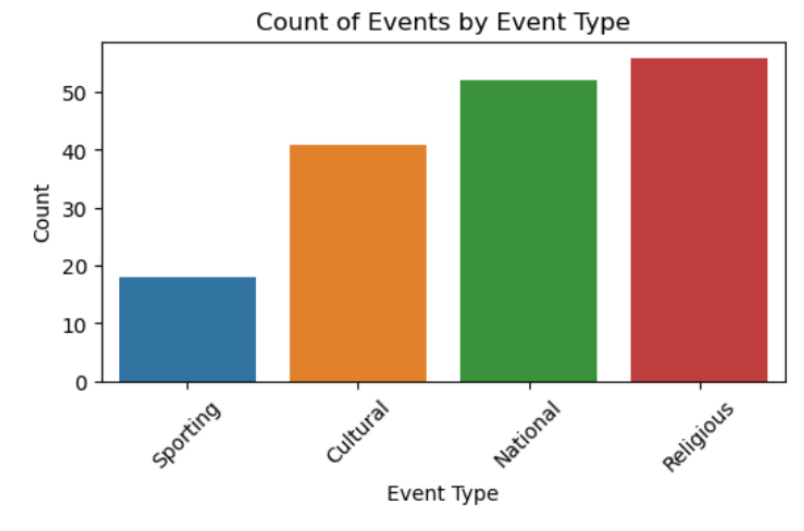


Fig Data Flow chart of the project

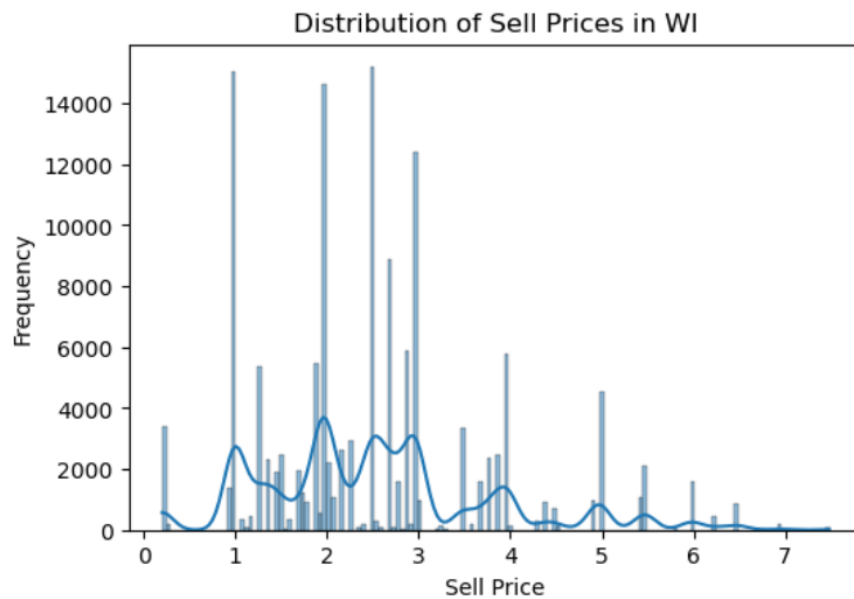
## EDA

### 1. Graphical presentation of event type vs Counts



*Fig: Count of event types*

### 2. Distribution of sell prices vs stor\_id (WI)



## 4. Data Preparation

### Data Limitation/Data Cleaning

#### 1. Missing Values:

The first thing you will need to do is to identify the columns or variables in your dataset that have missing values in them. In each column, missing values can be identified and counted by using Python's panda's library. Identify the nature of missing data helps to treat them.

# command use to check missing data

`df.isna().sum()`

```
Out[33]: id      0
item_id    0
dept_id    0
cat_id     0
store_id   0
..
sell_price 0
date       0
d          0
event_name 0
event_type 0
Length: 1553, dtype: int64


Out[31]: id      0
item_id    0
dept_id    0
cat_id     0
store_id   0
...
sell_price 0
date       0
d          0
event_name 130841
event_type 130841
Length: 1553, dtype: int64
```

*Data after performing missing data treatment*

#### 2. Outliers:

There are various ways to check outliers such as box plot, scatter, Z-score technique etc. Visualizations such as box plots and scatter plots can be beneficial for identifying data points that are located outside the bulk of the data.

# Using standard deviations, a Z-score determines the distance between a data point and the mean.



## Feature Scaling:

When the data was merged it was present in the two forms **continuous** data and **categorical** data. Scaling features are essential for normalizing variables and ensuring they have similar scales. It is possible that machine learning algorithms would be biased toward features with larger magnitudes without impacting the model's performance.

### 1. Standardization:

To handle continuous data, we use standardization or normalization however, I used standardization to do feature scaling of continuous data because It assists in the analysis and modeling of data when the data are standardized to have a mean of 0 and a standard deviation of 1, thus simplifying the comparison of variables with different units and magnitudes. Normalization, It may distort important patterns and make them difficult to interpret, which, in certain cases, may compromise the performance of models.

```
# Standerdization  
  
ss=StandardScaler()
```

### 2. Lebel encoding

The label encoding process converts categorical data into numerical format, which is essential to machine learning algorithms. Each category is assigned a unique integer, while ordinal information is preserved when necessary. As a result of this transformation, algorithms are able to process and make sense of categorical data, making them suitable for training models and improving their predictive capability.

```
# Lebel Encoding  
  
le = LabelEncoder()
```



## 5. Modeling

### a. Model\_1 Linear Regression & Random Forest

#### **Algorithm: Linear Regression**

**Hyperparameter:** Predefine value used in the algorithm

**Data Cleaning:** To ensure data consistency, missing values are treated.

**Feature Scaling:** The data was available in two forms, continuous data and categorical data. Standardization transforms Continuous data into the same structure, and categorical data is handled by label encoding.

**Feature Engineering:** In order to select relevant features, the constant features technique was tested, but it resulted in the loss of some of the most significant columns, such as store\_id, item\_id and date. Therefore, this method is neglected. Furthermore, the above three columns will be provided as inputs, which explains why it was not dropped.

#### **Training Process:**

**Data Splitting:** To assess model performance, the dataset was divided into two subsets. The **training (80)** set was used to train the model, while the **testing(20)** set was used to evaluate its performance. In this manner, the performance of the model can be generalized to unseen data.

**Model Fitting:** I trained the linear regression algorithm using the training set. By analyzing historical data, the model was able to capture the relationships between input features and sales revenue.

```
# model = LinearRegression()
```

```
# model.fit(X_train, y_train)
```

**Evaluation:** On the testing set, performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) were calculated to measure the model's

accuracy. It is through these metrics that we can determine whether the model's predictions are in line with the actual sales figures.

```
# y_pred = model.predict(X_test)
```

Note: The Linear Regression algorithm was not sensitive to class distribution, so no special handling was required for the imbalanced data.

### Algorithm: Random Forest

Hyperparameter: (n\_estimators=100, random\_state=21)

Data cleaning and feature scaling is applied same as linear regression

#### Training Process

**Data Splitting:** The data is splitted same as Linear regression model

**Model fitting:** The training data were used to train a Random Forest model with specified

## b. Model\_2 ARIMA and Seasonal ARIMA

**Algorithm:** SARIMAX (Seasonal auto regression integrated moving average with exogenous factor)

#### **Hyperparameter:**

An automatic ARIMA model (pm. auto\_arima) is used to determine 'order', with the seasonal component disabled ('seasonal=False'). The 'order' variable is extracted from the auto ARIMA model and used to initialize the SARIMAX model.

#### **Preprocessing:**

'Merged\_data' is assumed to be a DataFrame format with an index of 'date' and a column of 'sales'. The data is divided between a training set (80%) and a testing set (20%).

```
# train_size = int(len(merged_data) * 0.8)
```

```
# train, test = merged_data[:train_size], merged_data[train_size:]
```



### Model Training:

A SARIMA model is fitted to the training data using the auto ARIMA model (p, d, q). SARIMAX models are initialized according to the determined order. It is fitted using the SARIMAX model to the training data (train['sales']). On the test set, the performance of the model is evaluated.

```
# p, d, q = auto_arima_model.order
```

**Imbalance data:** As time series forecasting does not typically involve class distribution, imbalanced data does not apply.

### Forecasting:

We define a forecast horizon of 7 days. The SARIMAX model is used to generate forecasts over a specified period of time. As a result, forecasted values and confidence intervals are obtained.

```
# forecast_horizon = 7
```



## 6. Evaluation

### a. Evaluation Metrics

#### Model 1

##### 1. Linear Regression

###### **Evaluation:**

I evaluated the performance of our model using Mean Square Error and R-squared.

```
# y_pred = rf_regressor.predict(X_test)
```

**Note:** There is less sensitivity to class imbalances in Random Forest due to the fact that it is an ensemble method

**Conclusion on models:** Based on our results, it can be concluded that the Random Forest model performed significantly better than Linear Regression in terms of prediction accuracy.

##### 2. Random Forest

###### **Evaluation:**

Mean Square Error (MSE) and Root Mean Square Error (RMSE) are computed to evaluate the accuracy of the model's predictions.

```
# mse = mean_squared_error(test['sales'][-7:], forecast_values)
```

Code demonstrating the use of SARIMAX for sales time series forecasting, including data preparation, SARIMAX modeling with auto ARIMA hyperparameters, and evaluation of forecast accuracy using MSE and RMSE. The univariate method is ideal for making sales predictions based on a single variable.



## Model 2

ARIMA and SAMIXA

Evaluation:

Forecasting is evaluated using the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics. The accuracy of these metrics can be measured by comparing the predictions made by the SARIMAX model with data from the actual test.

```
# mse = mean_squared_error(test['sales'][-7:], forecast_values)
```

```
# rmse = np.sqrt(mse)
```

### b. Results and Analysis

#### **Linear Regression**

Mean Absolute Error (MAE): 6589.5113134022795

Root Mean Squared Error (RMSE): 9137.762321411927

R-squared ( $R^2$ ) Score: 0.011070571115474603

#### **Random Forest**

Mean Squared Error: 4199.08673635689

R-squared: 0.9999502674839733

#### **SARIMAX**

Mean Squared Error (MSE): 463632191.69441235


Root Mean Squared Error (RMSE): 21532.120000000286

#### **Analysis:**

Random Forest displays the lowest RMSE in the three models, indicating that it is the most accurate. Although Linear Regression had the highest RMSE, SARIMAX performed better than Linear Regression but not as well as Random Forest.

#### **Implications:**

Based on the results, Random Forest appears to be the most promising model for sales forecasting. According to the RMSE, it performs better than both Linear Regression and SARIMAX.



It is possible that the Random Forest hyperparameters can be further optimized and fine-tuned to achieve even better performance. Furthermore, it may be possible to achieve superior accuracy by exploring other ensemble methods and more advanced time series models.

## c. Business Impact and Benefits

### Impact and Benefits of the Model

- It is possible to maximize revenue and profit by using data-driven pricing strategies. In order for retailers to determine the optimal price for different items and categories, it is necessary to understand the relationship between pricing and sales.
- In addition to saving labor costs, efficient staffing can enhance customer service and result in a better customer experience.
- The ability of retailers to accurately predict store and item sales can reduce the likelihood of overstocking and stockouts. As a result, carrying costs are reduced and customer satisfaction is improved.

### Improvement in the model

As a result of the improvements, there can be a substantial impact on the business value. A business can achieve millions of dollars in savings each year if it combines a reduction in overstock costs with an increase in revenue and shrinkage in labor costs together with the reduction in inventory costs.

## d. Data Privacy and Ethical Concerns

A merged dataset contains information about the sale of items with regard to the different stores and locations that the items were sold on a daily basis.

### data privacy and mitigate ethical concerns

As the misuse of sensitive data could cause legal and reputational issues as well as violations of customer privacy laws, the data is used within projects to avoid any potential legal and reputational issues related to the misuse of sensitive data. In order to collect data, the access point has to be selected. In addition to mitigating risks, it demonstrates a commitment to ethical and responsible data practices. This is crucial for maintaining trust between customers and stakeholders, demonstrating a commitment to security.

## e. Challenges and issues

### 1. Large data issue while merging:

```
Unable to allocate 26.0 GiB for an array with shape (1541, 2268620) and data type int64
```

I have tried using dask to reduce the memory size, but it does not seem to make any difference in reducing the memory size. After using dask, the previous size of sales\_train data was 370.2 MB, but after using dask, the memory size was 359.9 MB.

### 2. Installing Libraries:


```
error: metadata-generation-failed
```

It was an issue with the downloading fbprophet.

### 3. Data Scalability:

There will often be a need for scalable infrastructure and algorithms when dealing with large datasets, especially with multiple stores, items, and time series data.

### 4. Datatype conversion



Due to large data constantly getting an error of memory storage, it is resolved by merging in chunks.

Error: Unable to allocate 25.7 GiB for an array with shape (1541, 2235021) and data type int64

#### 5. Complexity of the model:

Developing and implementing complex machine learning or time-series analysis models is a complex process.

#### 6. Deployment:

It was technically challenging to deploy machine learning models as APIs or integrate them into existing systems due to the expected lack of support files and the missing pre-installation files.

#### 7. Cookie clutter installation failure:

python.exe returned an exit code of 1 fail to install cookie clutter.

### f. Limitations and Opportunities

- Among the mered\_data we have available with us, only the WI column is present. I tried to merge all the unique identifiers from each store, but I wasn't able to do so. As a result, the prediction is also performed based on a single store id, namely the WI identifier.
- There is an opportunity provided by this model in that we can add more stored identifiers for future experiments so that we can do prediction on them to make more valuable information for the business in the future.



## 7. Deployment

Following steps are need to be followed for deployment on heroku

### Step 1: Create the project

- To create a new project, I'll open a terminal and run `npm init -y`. In order to install the Express module for the dummy server, run the `npm install express` command.
- Having installed this library, we can create a new file, named `app.js`, for our project.

### Step 2: The version control system

- Create a new repository for your application on GitHub.
- The following commands must be executed after you click the Create repository button on Github in order to upload your local code to a repository:

```
# Git init
# Git add README.md
# Git commit -m "first commit"
# Remote add origin http.....
# Git push -u origin master
```

### Step 3: Link the repository to heroku

- To begin, create a new application on Heroku .
- On the top navigation, you will find Overview, Resources, Deployment, Metrics, and so on. Make sure that Deploy is selected. Once the GitHub icon has been selected, click on it.
- Search for the desired application, which in our case is `demo-deploy-app-09`. Click on the Connect button.

### Step 4: Configure heroku to run the application

- The only thing we need to do to update our application is to commit a new version to GitHub. The code will be automatically pulled to Heroku if we have enabled the Automatic Deploys option. Otherwise, we will need to click on Deploy Branch again.



## 8. Conclusion

As a result of the project, predictive and forecasting models for retail sales were developed successfully. As a result of the study, key findings include improved inventory management, forecasting of demand, and price optimization, resulting in cost savings and revenue growth. The project met their requirements to meet stakeholders' expectations for data-driven decision-making. It is planned that future work will include continuous monitoring and updating of the model, integrating real-time data sources, and expanding predictive capabilities to address the changing needs of businesses.



## 9. References

1. *How to merge two csv files by specific column using Pandas in Python?* (2021, January 13). GeeksforGeeks. <https://www.geeksforgeeks.org/how-to-merge-two-csv-files-by-specific-column-using-pandas-in-python/>
2. *How do I combine large csv files in python?* (n.d.). Stack Overflow. Retrieved October 5, 2023, from <https://stackoverflow.com/questions/56494140/how-do-i-combine-large-csv-files-in-python?rq=3>
3. *15+ Use Cases of Data Science in Retail.* (n.d.). Wwww.knowledgehut.com. <https://www.knowledgehut.com/blog/data-science/use-case-of-data-science-in-retail>
4. Mcleod, S. (2019, May 17). *Z-Score: Definition, Calculation and Interpretation.* Simplypsychology.org; Simply Psychology. <https://www.simplypsychology.org/z-score.html>
5. scikit-learn. (2019). *scikit-learn: machine learning in Python.* Scikit-Learn.org. <https://scikit-learn.org/stable/>
6. George, D. (2020, December 13). *A Brief Introduction to ARIMA and SARIMA Modeling in Python.* The Startup. <https://medium.com/swlh/a-brief-introduction-to-arima-and-sarima-modeling-in-python-87a58d375def#:~:text=SARIMAX%20is%20similar%20and%20stands>
7. *Heroku Deploy – How to Push a Web App or Site to Production.* (2020, August 5). FreeCodeCamp.org. <https://www.freecodecamp.org/news/how-to-deploy-an-application-to-heroku/>

—

...