Assignment 2

ML as a Service

Daniel Alexander
Student ID: 24591214

2023-10-05

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 **4**
4. Data Preparation **5**
5. Modeling **6**
a. Approach 1 6
b. Approach 2 6
c. Approach 3 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

XYZ, an American retail company with a presence across three states: California (CA), Texas (TX), and Wisconsin (WI) has just launched an initiative to leverage the data stored in its data warehouse to drive data-driven decision-making across various departments.

This project is part of the initiative, focusing on inventory management, cash flow management, KPI settings and resources management.

Key outcomes include:

- Operational Excellence: By making informed decisions based on data-driven insights, XYZ is able to have a realistic KPI.
- Inventory Management: Data-driven decision-making has significantly improved predictive accuracy, helping the company optimize inventory management,
- Resource Allocation: Departments can now allocate resources more effectively, ensuring optimal staffing levels, targeted marketing efforts, and budget allocation.

The success of this project is significant as it would directly impact the cash flow and profitability of the company.

2. Business Understanding

a. Business Use Cases

Historically, Inventory management has always been a challenge for our company. There are a handful of branches that are struggling with stock outs which lead to lost sales and customer dissatisfaction while overstocking ties up capital and storage space. These problems could be addressed by accurate demand prediction according to each store, date, and item.

XYZ's board members have been pushing executives to improve the company cash flow management and allocate the excess cash to be invested instead of just sitting around in branches account and hence we need a revenue forecast model so that we do not allocate excessive reserve in each stores as we can predict the expected cash coming in 7 weeks.

Another important challenge is relating to employee's welfare. As employee's bonuses are directly related with the stores' KPI and company-wide revenue target, XYZ wants to make sure that KPIs and revenue targets are fair and realistic. This challenge could well be addressed by a machine learning model that could predict the revenues of each store and item which then allow management to set KPI on a daily basis and scalable to weekly, monthly etc.

b. Key Objectives

The goal of this project is to train 1 machine learning model which can accurately predict the revenue for a specific item in a specific store on a given date, and 1 forecasting model to forecast the total revenue for the next 7 days given a specific date for all items and all stores combined.

The board members and executives are keen to see improvements in inventory management, cash flow management, and KPI expectations, meanwhile the end user which is going to be hands on handling the models produced from this project will want an easy to use interface where they can learn quickly how to use it.

The requirements from the board members and executives will be directly addressed by the success of training the machine learning model while the end users requirement will be addressed by the way we will push our model to production via API, where the end users could just input the date, item id, and store id and they will get the prediction from our trained model, and for the forecasting model, the end user would just need to input a date and it will return the forecast for the next 7 days based on the day they inputted.

3. Data Understanding

The datasets were gathered from our internal data warehouse where we store our data from all stores. The data was stored in dimension tables and a fact table, the dimensions are dates, item price, and special events related, i.e. 1 fact table and 3 separate dimensions tables.

The sales data were stored in an aggregated pivot table, with the days on columns and item identifier as the row while the values represent the number of items sold. Furthermore, the other dimensions such as the dates, item price, and indicator of special events day are stored in different tables. This kind of schema is not suitable for training a machine learning model and hence would need transforming before we could process them.

Features	Description	Significance
id	Unique ID on each item	Not significant for our purpose as it wont be used for our model.
item_id	Item ID	Significant
cat_id	Item Categories ID	Significant
store_id	Store ID	Only significant if we had enough computing resources
state_id	State	Not significant for our purpose as it wont be used for our model.
d_*	Day	Significant as it will be used to join the dates
date	Date	Significant
wm_yr_wk		Significant as it will be used for feature engineering

event_name	Event name	Not significant
event_type	Event categories	Potentially Significant

4. Data Preparation

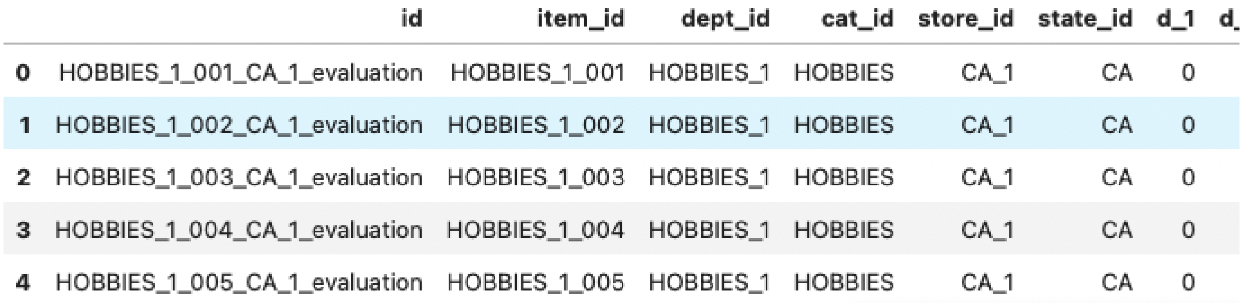
a. Data pre-processing.

The type of schema in our dataset is not optimized for training a machine learning model. A machine learning model favors a row basis schema where the features are described by the columns and each observation is represented with the rows. Therefore, we need to transform the schema to row basis schema, and join the dimension tables to the sales table.

1. Transforming aggregate data to row based

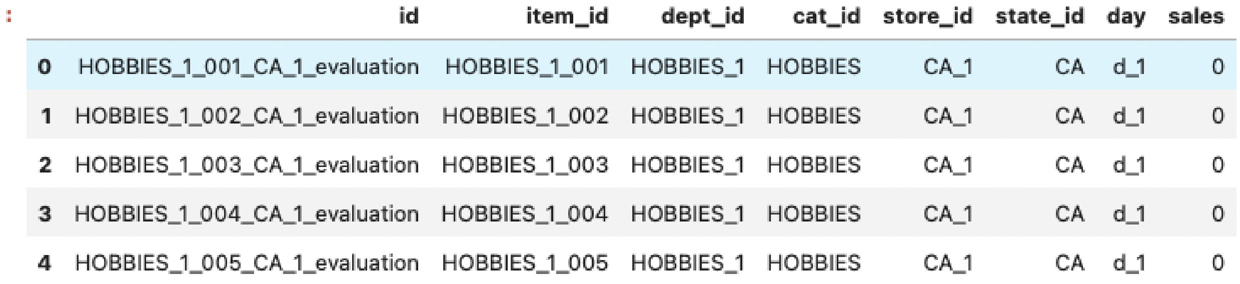
We melt the sales data with the following specification: we keep columns id, item_id, dept_id, cat_id, store_id, and state_id as it is, while melting the rest of the columns which represent the days and store the value as sales. The melted data frame now have each item per day, i.e. shape = (number of distinct item * number of days)

Before melting:



	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	

After Melting:



:

	id	item_id	dept_id	cat_id	store_id	state_id	day	sales
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0

However, after we melt the data the data size became outrageously big with more than 34 million rows meanwhile we still had not performed one hot encoding which would scale the data even bigger.

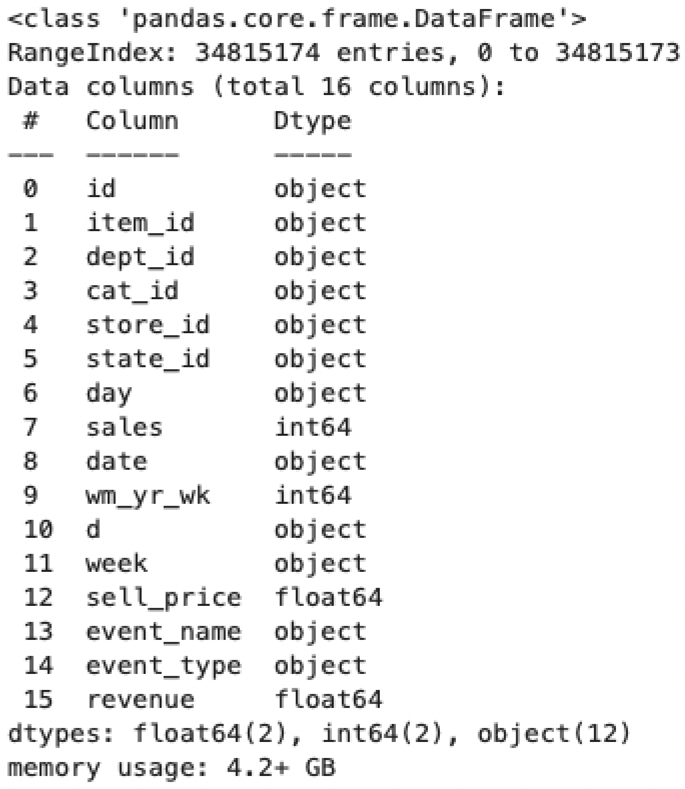
2. Joining dimension tables with the sales fact table

Firstly we joined the fact table with the calendar dataframe which stores the date and week information based on the day information. We call this interim dataframe merged_df_v1.

Second, we joined merged_df_v1 with the item price dataframe based on the item_id and store_id. We call this as merged_df_v2.

Lastly, we joined merged_df_v2 with the calendar events dataframe based on the date. We call the resulting dataframe as merged_df_v3 and this will be our main source of data.

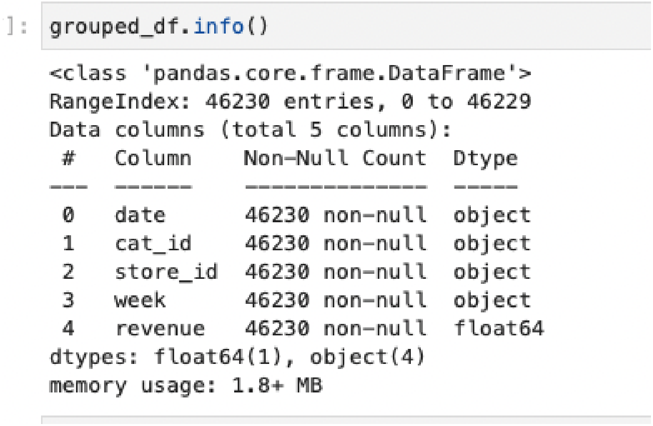
The only missing values present are from the merging of calendar events, since not everyday is a special day. However, since we do not use these features, then we did not bother cleaning them, but if we want to clean them, then filling the missing value with unspecified should do.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34815174 entries, 0 to 34815173
Data columns (total 16 columns):
 #   Column      Dtype  
---  ------      -----  
 0   id          object 
 1   item_id     object 
 2   dept_id     object 
 3   cat_id      object 
 4   store_id    object 
 5   state_id    object 
 6   day         object 
 7   sales       int64  
 8   date        object 
 9   wm_yr_wk    int64  
 10  d           object 
 11  week        object 
 12  sell_price  float64
 13  event_name  object 
 14  event_type  object 
 15  revenue     float64
dtypes: float64(2), int64(2), object(12)
memory usage: 4.2+ GB
```

3. Grouping the data

Since we are interested in forecasting the revenue across all stores and items combined, then we need to group our data based on the date. On the other hand, it would be better if we did not group the data, however, the sheer size of the data was unworkable, it took more than 48 hours for us to fit a model and the fitting failed. Therefore, we grouped the data for prediction by their date, week, cat_id, and store_id. Although not ideal, it's necessary.



```
]: grouped_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46230 entries, 0 to 46229
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   date      46230 non-null  object 
 1   cat_id    46230 non-null  object 
 2   store_id  46230 non-null  object 
 3   week      46230 non-null  object 
 4   revenue   46230 non-null  float64
dtypes: float64(1), object(4)
memory usage: 1.8+ MB
```

4. Split the data to a training and validation set.

Prediction

We split the data with 90:10 ratio for training and validation respectively. The ratio 80 for training should be enough given 40k+ observations.

Forecasting

We split the data leaving out '2015-02-18' onwards for testing which is 2 months worth of data.

b. Data Engineering

1. Create a new feature called ‘week’

The raw data only contains wm_yr_wk format which is not useful for training machine learning models. Therefore we extracted the week out of wm_yr_wk and called it week, which represents the week number for the year in string format, for example ‘03’ means its week 3 out of 52 or 53 weeks in a year.

2. Create the target value ‘Revenue’

Since we want to predict the revenue, then this engineering is vital, revenue= number of items sold * sell price.

3. One Hot Encoding

Part of the pipeline will perform one hot encoding on the categorical features [cat_id, store_id, week]

We did not remove any columns nor observations for this experiment because we used a pipeline which automatically uses our features selection and ignores the rest. The features that we use in this experiment are the store_id, item_id, and week because amongst the available features, those are the features that will affect the revenue while other features are unique ID’s and date which will introduce overfitting to a prediction model.

One important step that we decided to take is using cat_id instead of item_id since one hot encoding item_id which has 1000+ unique values will make the data unworkable with the current resources that we have.

5. Modeling

a. Predictive Model

We use the random forest regressor in this experiment as it is a good algorithm to start as it can handle outliers well and generally performs quite well. We used the default hyperparameter for the initial experiment.

Feature engineering that are exclusive to the prediction model are the one hot encoding and the grouping by date, week, cat_id, and store_id. We built a pipeline which allows us to automatically perform data preprocessing preparing it to be fitted as explained in the data preparation section.

The processed data then splitted with a 90:10 ratio for training and validation.

b. Forecasting Model

For the forecasting model, we use the ARIMA time series algorithm in this experiment as ARIMA models are robust to noise and can filter out random fluctuations in the data, furthermore, ARIMA models can be relatively straightforward to implement.

The hyperparameter for this experiment is ARIMA(p,d,q) with
p=4
d=1
q=4

For the forecasting model, we summed up the revenue across all items, and stores then we grouped them by the dates since we are interested in the total revenue nationally.

After processing the data, we split them as training and validation, leaving the last 2 months for validation.

6. Evaluation

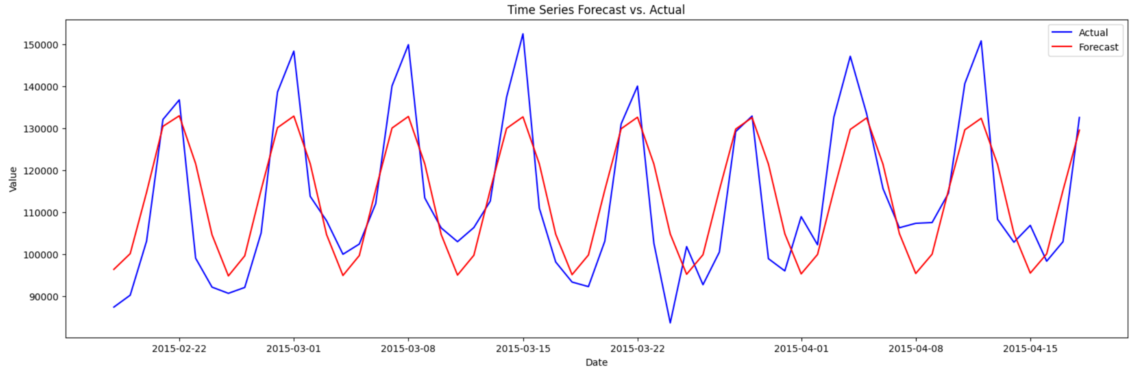
a. Evaluation Metrics

The evaluation metrics used in this project is the MAE as Easy Interpretation: MAE is easy to understand. It represents the average magnitude of errors between predicted and actual values.

The goal is to minimize the MAE both in prediction and forecasting models, the smaller the MAE means smaller the difference between the predicted revenues and the actual figures, which then indicates how accurate we are on giving guidance to the operation and finance team on the predicted revenue.

b. Results and Analysis

ARIMA MODEL



The graph for the test dataset shows a rather promising prediction as it captures the peaks and troughs on the test dataset.

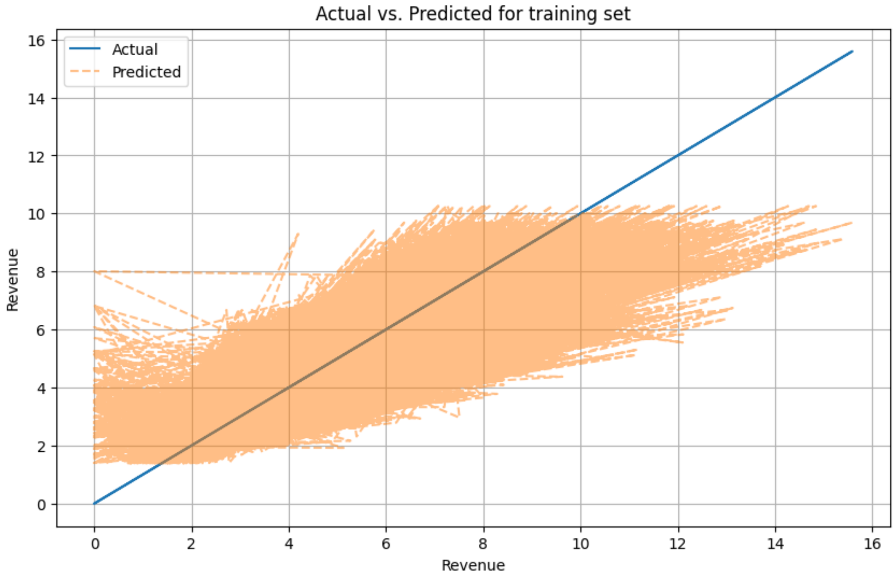
MAE for test: 8679.845370286052

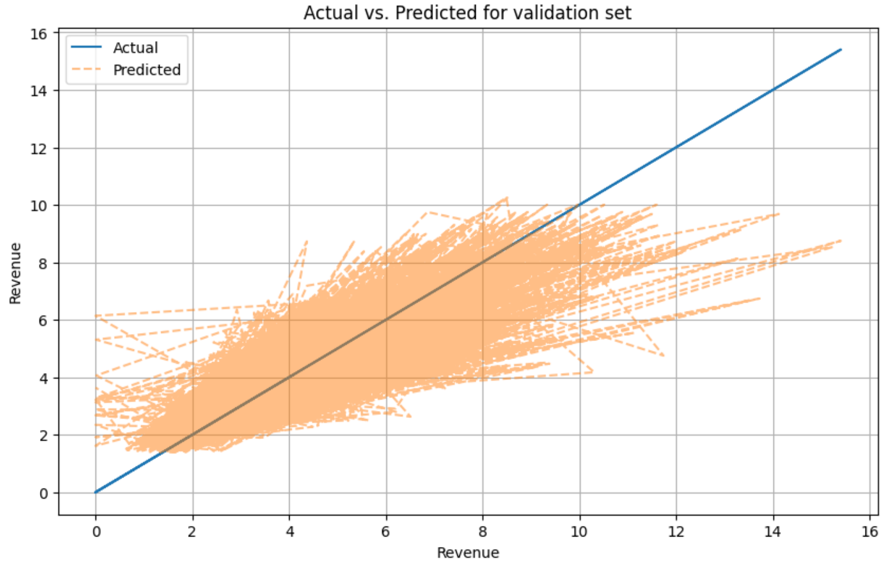
The MAE figure suggests that on average the predictions are off by 8600, from the graph we suspect that the differences are mainly caused within the peaks area. However, the model should be enough for our business objective.

Random Forest Regressor Model

MAE for train: 0.7319614594115664 - MAE for test:0.7667440922577484

The model seems to be performing quite well and not overfitted, MAE scores are relatively low and consistent through training and testing. However, the model performs rather poorly for low revenue and high revenues, possibly because those revenues rarely occur in the data.





The model will be a relatively good guidance for the business objectives that we want to reach. However, since it does not predict high revenue well, then the business should check on combinations of stores and categories which are likely to accomplish such revenue and manually adjust the prediction by their average.

c. Business Impact and Benefits

By using the prediction models XYZ should be able to predict the revenue to be generated for a specific item in a specific store on a given date. However, due to resources limitation our prediction model was built based on category id instead of the individual item, and hence is better suited to be used to predict revenues based on the category instead of item, albeit it will be deployed in a way that it will map the item id to category.

This added value will contribute to optimize inventory management and help the business to improve their KPI settings to be realistic but not underwhelming.

Meanwhile the forecasting model will help the finance team to forecast the money coming inwards on a weekly basis. This would allow them to manage the cash reserved for day to day basis to the optimal level so that any excess cash could be allocated for other investment opportunities.

d. Data Privacy and Ethical Concerns

There are no ethical concerns relating to this project as all data are internal and contain no sensitive or private data which could cause privacy or ethical issues.

7. Deployment

The trained models are then deployed in such a way so that the end users can access it and utilize them easily. Before deploying them online, we performed local testing ensuring API endpoints and the functionalities work as we expected. Finally, we deployed our model through API using the fastAPI package which was containerized using docker and then launched online hosted by Heroku the app name is guarded-beach-15214. The model should be available to be accessed through https://guarded-beach-15214-da8d14c531d7.herokuapp.com/docs

List of endpoints:

1. '/health' : Checking whether connection is good to go

2. '/sales/stores/item/' : API endpoint to use the predictive model. This endpoint will output the revenue prediction for an item in a store on a date

Expected inputs are:

item_id: Only enter one from the item_id (must be registered item in the training dataset)

store_id: Only enter one from store_id (must be registered store in the training dataset)

Date: Has to be in the following format 'yyyy-mm-dd' and a valid factual date.

3. '/sales/national/': API endpoint to use the forecasting model. This endpoint will output the forecast for the next 7 days from an inputted starting date.

Date: Has to be in a following format 'yyyy-mm-dd' and a valid factual date and strictly need to be a date > 2011-01-28 as the algorithm only accounts for 7 days forecast and the model was made based on earliest date = 2011-01-28

8. Conclusion

To conclude, this project has met its objectives. The models trained perform relatively well, albeit few hiccups due to resources availability. Upon adopting these models on day to day operations, the business will see improvements on their inventory management as they will have relatively accurate guidance on where the revenue will sit on for a given date, store, and items. These perks are also going to be impacting the KPI settings for each store advocating more realistic KPI, achievable but not too low.

On the other endpoint, the finance team will get their hands on guidance on the forecasted cash inflow on a weekly basis. This will help them achieve what the executives and board members have been focusing on.

In the future, the model must be monitored and regularly updated when needed. As market behaviors change, new product lines are coming, and inflation affects the revenue, the model would need to be updated.

It would be great if we can add more computing resources so that we can train algorithms with a smaller granularity which potentially delivers a better performing model.

9. References

ChatGPT. (2023, October 06)