EXPERIMENT REPORT

Student Name	Daniel Alexander
Project Name	Machine Learning as a Service
Date	04-10-2023
Deliverables	alexander_daniel-24591214-ARIMA_forecasting.ipynb ARIMA model GitHub Repo for the Model: https://github.com/daniel-alexandr/Assignment2_ML_as_a_service GitHub Repo for the API Deployment: https://github.com/daniel-alexandr/assignment_2_api

---

1. EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	This project is designated to help the finance team to forecast the revenue for financial planning and budgeting, including cash flow management and resource allocation. Inaccurate model will result in misguidance on the revenue forecast which results in unrealistic company wide revenue target and incorrect cash flow management and resource allocation which could cause other complications.
1.b. Hypothesis	It is possible to build a machine learning model which accurately forecasts total sales revenue across all stores and items for the next 7 days given a date input. It's worthwhile because an accurate model would help to solve the business problem that we want to address.

1.c. Experiment Objective	1. If we can successfully train a machine learning model which is accurate enough for our needs, then we will push the model to production via API hosted by Heroku, so the end user can input the dates and get the prediction based on our trained model. 2. If the experiment does not produce an accurate model, then we should pursue more experiments with different approaches before serving the model to be used by the end user.

---

2. EXPERIMENT DETAILS

Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.

2.a. Data Preparation

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 type of schema 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 dataframe now have each item per day, i.e. shape = (number of distinct item * number of days)

Before melting:

After Melting:

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.

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

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

2.b. Feature Engineering	We had to create additional features as following: 1. Summing The Revenue Since we are interested in the total revenue across all stores and items, then we have grouped our dataframe based on the date. Bear in mind that we are using the same master processed data as the one from the prediction model, however for time series forecasting purposes in this experiment we only need the date and the total revenue.
2.c. Modelling	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 future experiments we might want to try more complex algorithms like prophet as they can handle holidays and special events potentially improving the model performance.

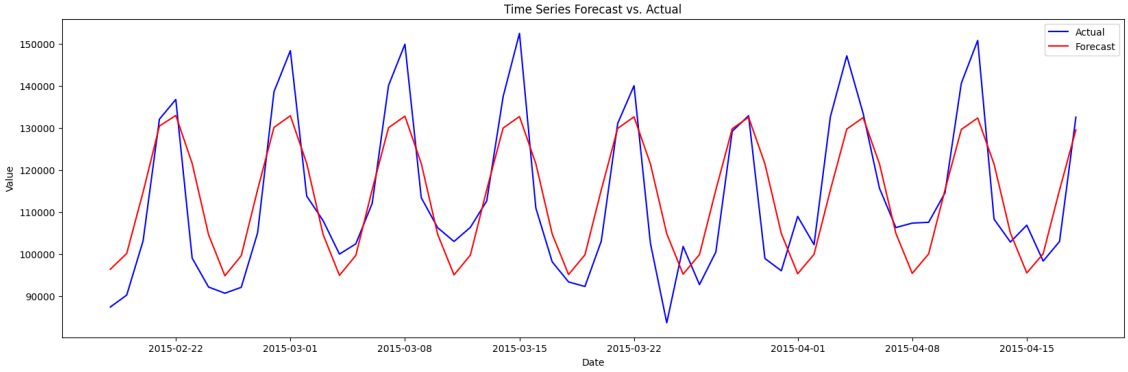
3. EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance

MAE for test: 8679.845370286052

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



3.b. Business Impact

This model should help the finance team manage the cash flow and hence making more money available for the company to invest.

3.c. Encountered Issues

1. The schemas of the raw data which are not suitable for machine learning purposes, which require changing its schemas by melting them.

---

4. FUTURE EXPERIMENT

Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.

4.a. Key Learning

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, we should push this to production.

4.b. Suggestions / Recommendations	1. Push the model for production to be deployed via API, as the model should perform well enough. 2. Experiment with other algorithms such as prophet, as more complex algorithms tend to fit better. However, due to time constraints it's better if we don't spend more time experimenting and spend more time on deploying the model.