EXPERIMENT REPORT

Student Name	Daniel Alexander
Project Name	Machine Learning as a Service
Date	04-10-2023
Deliverables	alexander_daniel-24591214-rf_prediction.ipynb Random Forest Regressor 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 operation team within the business to plan ahead for the optimal level of inventory required for any given day, and to provide each stores' performance target. Inaccurate model will result in misguidance on the target which results in unrealistic target and could potentially affect the welfare of the stores' employee as their KPI depends on it. Furthermore, inaccurate models will result in inaccurate inventory management which introduces inefficiency and financial loss due to overstocking or understocking.
1.b. Hypothesis	It is possible to build a machine learning model which accurately predicts the revenue based on the item, store, and date. 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, item id, and store id 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 data frame now have each item per day, i.e. shape = (number of distinct item * number of days)

2. 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.

3. 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.

	4. Split the data to a training and validation set. We split the data with 90:10 ratio for training and validation respectively. The ratio 80 for training should be enough given 40k+ observations.
2.b. Feature Engineering	We had to create additional features as following: 1. 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. 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.

2.c. Modelling	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. For future experiments we might want to try an adaptive booster algorithm or XGBoost as they might give a bump to the model performance compared to random forest.

---

3. EXPERIMENT RESULTS

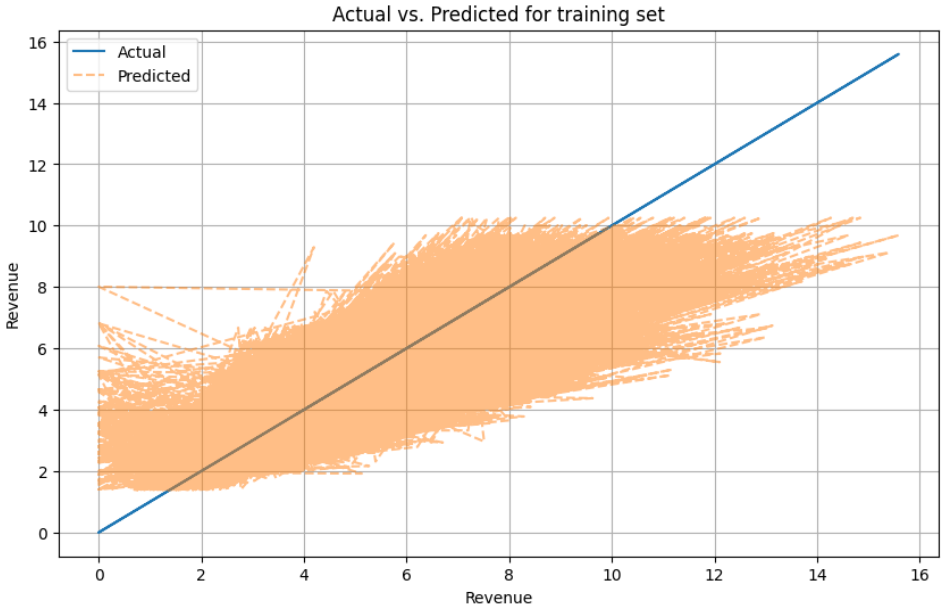
Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

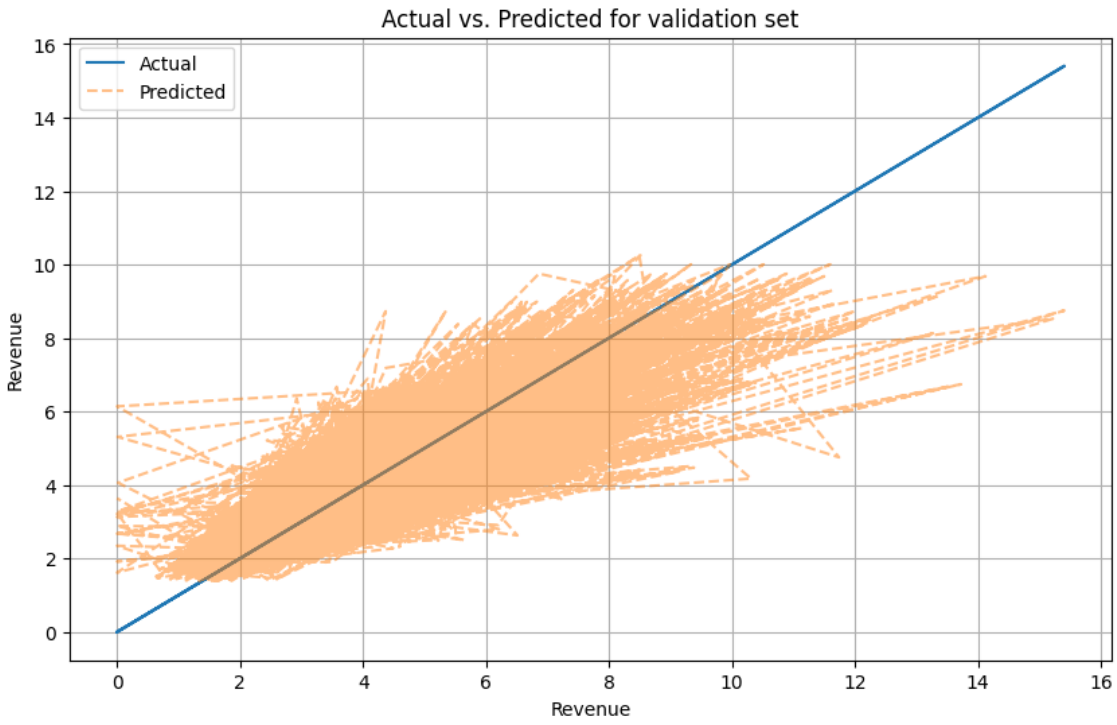
3.a. Technical Performance

Score of the relevant performance metric(s). Provide analysis on the main underperforming cases/observations and potential root causes.

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.

3.b. Business Impact

This model should be our benchmark, and by the very least, it is suitable to be used for production. The model will be a relatively good guidance for the business objectives that we want to reach. Since it does not predict high revenue well, then the business should check on stores and categories which are likely to accomplish such revenue and manually adjust the prediction by their average.

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.
2. It is not clear why the week starts from January 29, which introduces complication and requires manipulation on the API production so the logic matches with the fitted model.
3. The size of the data is big and takes plenty of resources, which could grow exponentially when fitting a more complex model. Solutions would be adding more computing resources or choosing a less memory intensive algorithm to train or train less data by doing sampling.
4. The API that we want to build which aims to use item_id will cause the data to grow exponentially such that our existing resources could not handle the data. Therefore, we use category_id instead for training our data and group the raw data based on date, week, category_id, and store_id

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	If we can relax the requirement of predicting each item specifically to categories instead, then this experiment is a success. The model that we trained was supposed to perform relatively good and sufficient for the business objective given. We still have room for improvement by experimenting with more complex algorithms such as XG boost algorithm if the project deadline allows.
4.b. Suggestions / Recommendations	1. Deploy model for production through API, as the model performs relatively well when the level of prediction granularity is reduced such that we can utilize the category instead of each item specifically. This is the most cost benefit efficient choice 2. Experiment with another algorithm, given the same data preparation treatment. Other more sophisticated algorithms like XGBoost arguably should perform better, however, project deadline is the issue here. 3. Buy more powerful resources or pay for external service like databricks so that we can train a model with smaller granularity basis, arguably will perform better.