Trendcast: Demand Forecast for Fashion Retailers

Inderpartap Singh Cheema
Department of Computing Science
inderpartap_cheema@sfu.ca

Najeeb Qazi
Department of Computing Science
najeeb_qazi@sfu.ca

Pallavi Bharadwaj
Department of Computing Science
pallavi_bharadwaj@sfu.ca

Mrinal Gosain
Department of Computing Science
mrinal_gosain@sfu.ca

Summary

Forecasting retail sales includes various factors such as store location, day of the week, market trends, etc. The addition of a factor like weather can have interesting results. Studies have shown that weather affects people's behaviors and spending habits[1] [2] Through this project, we wanted to analyse the impact of weather on retail sales and devise a way to reliably forecast the sales and the quantity required by a retailer for a short forecast horizon. We integrated the data obtained from FIND AI with the weather data gathered from MeteoStat API. The data was then cleaned and transformed for further exploratory analysis, followed by modeling using various techniques. AutoML tools were used for the initial implementation which were then followed by models like FbProphet, LightGBM, XGBoost, and Stacked Ensemble. Stacking outperformed all the other implementations and was used for final predictions. The models were then deployed on a Flask application, which worked as a dashboard for a user to view forecast results for each city and department.

1 Motivation and Background

In collaboration with FIND AI, we decided to pitch on an estimated assumption that the consumer behavior may be affected by weather, especially in the fashion industry. Retailers often end up over-stocking or under-stocking their inventories due to an unpredictable and unexplained consumer behavior. Our efforts are focused in eliminating this unpredictability and help retailers better manage their inventories by making informed decisions also considering weather as an influencing factor.

2 Problem Statement

In this project, we decided to analyse if the weather has an effect on retail sales [3][4] along with various other features. If we find a relationship between weather and retail sales, then we can,

- Help retailers make better stocking and shelving decisions considering weather as an influencing factor
- Determine the period of reliable sales predictions we can make, considering weather to be highly volatile and unpredictable

3 Data Science Pipeline

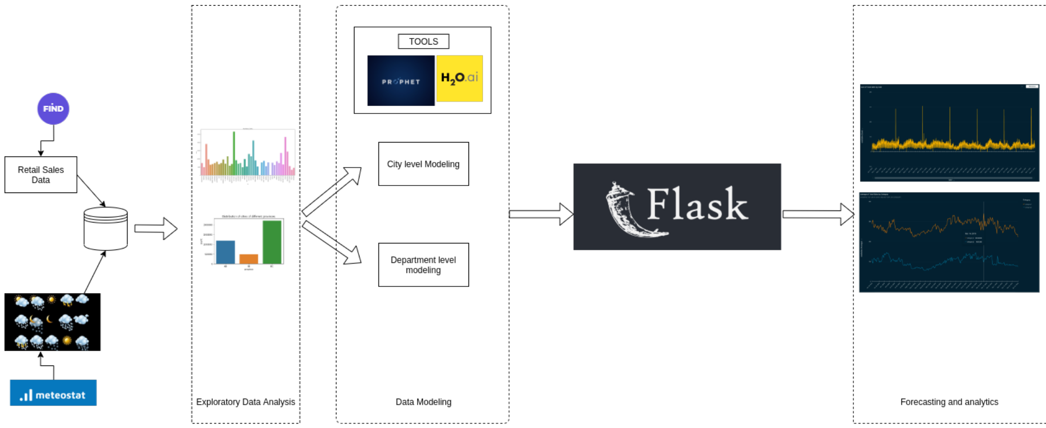


Figure 1: Workflow

3.1 Data Collection

Our pipeline incorporates datasets from two sources to solve the problem:

- Retail Data: FIND AI provided us a historic time-series dataset containing six years of anonymized retail sales transactions. The dataset was hierarchical in nature containing: Date of purchase → Province → City → Category → Department → Class → Style → Quantity of items sold and Total sales.
- Weather Data: We fetched the weather related information like min_temperature, max_temperature, precipitation, snow depth, wind speed, etc, for corresponding dates and cities in the retail data from Meteostat using their Historical Weather API. The API provides weather information based on station IDs. We initiated API calls for each city in the sales data to fetched the station IDs and the subsequent calls were to use these station IDs and fetched weather data for the cities based on their earliest and the latest dates..

3.2 Extraction, Transformation and Loading [ETL]

Data integration was performed at two levels, city and date. After combining the two data sources, the following ETL steps were performed.

- Weather features with more than 80% null values were dropped since interpolating them would lead to incorrect and biased data.
- Other missing data in weather was filled with interpolation using nearest matching assuming that the weather would not vary a lot in small time-frames.
- Missing data in the retail sales records after integration were filled with zeroes assuming there were no sales that particular day.
- The departments column had 14 possible categorical values which were aggregated and transformed into 14 columns indicating the total quantity of sales for each department in each city on each date. These 14 columns were the target features in our department-wise prediction models.
- Hierarchies under the department were dropped since the sales were sparse if we drill down in the product hierarchy and time.

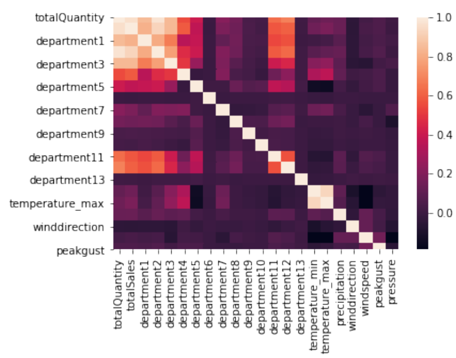
3.3 Exploratory Data Analysis [EDA]

For the initial EDA, AWS Quicksight was used to have an initial overview of the dataset. After which, we explored the distribution of the features to measure the skewness, cardinality and frequency distribution of the unique features in the feature vector.

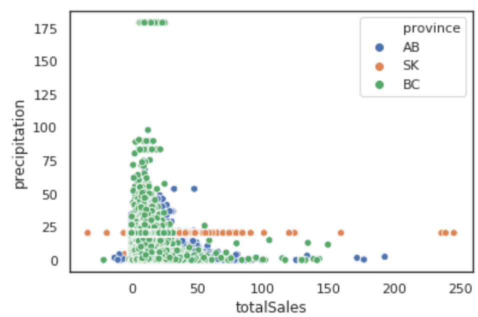
After basic EDA, we tried to find multiple feature interactions with each other for which we made a heatmap (2a) visualizing the correlation of features using Matplotlib and Seaborn.

We found some interesting results like the correlation of weather was stronger in some cities and the departments. For example, department 4 was highly correlated by min and max temperatures. Precipitation had significant correlation with department 2 as compared to other departments.

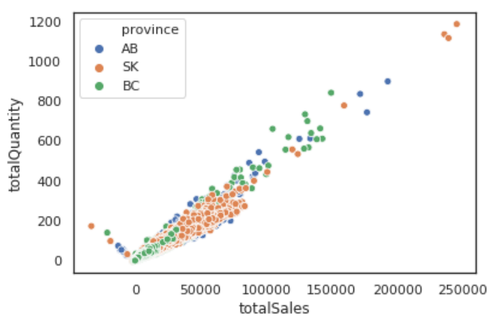
We also found that the city of Kelowna had some erroneous values for precipitation which was found by a scatterplot between precipitation and totalSales (2b). Then, due to this, we had to do data verification which led to the conclusion that there were incorrect values which we later fixed by replacing the average precipitation value of that given month.



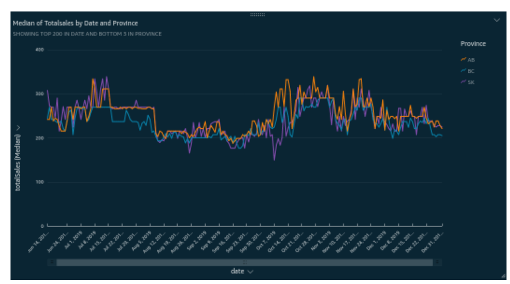
(a) Correlation heatmap



(b) Outlier detection: Kelowna



(c) Total sales v/s Total quantity



(d) Total Sales by province using AWS Quicksight

Figure 2: Some of the plots generated during EDA

3.4 Data Modeling

H20 Driverless AI, which is an autoML product by **H20.ai**, was used in the initial stages of our project to gain some insights for feature engineering and selecting the best features.

This was then followed by **FbProphet**, which is an open-source tool released by Facebook's Data Science Team. It was implemented for forecasting our time-series data based on an additive model where non-linear data is fit based on trends seasonality and holidays. Since it gave significantly better results, FbProphet was used to set a benchmark score for us to compare different model performances.

To try and achieve scores better than the dedicated forecasting library (FBprophet), **LightGBM** was implemented. It showed slight improvement over our benchmark, and even under-performed for predicting sales without weather. Another boosting technique was implemented by modeling **XGBoost** which showed significant increment in performance after hyper-tuning the parameters.

A **Stacked Ensemble** was then created which is a technique where data is trained on base learners and then predictions from those models are fed into a meta-model which then predicts the final values.
In our implementation, we used **random forest, extra trees, adaboost** and **gradient boosted trees regressor** as our base learners or **level-0 models**. **XGBoost** was used as the meta-model or **level-1 model** which was trained on the predictions of level-0 (base learners).
Using this stacking model, we were not only able to beat our benchmark but also the vanilla implementation of XGBoost which we modeled earlier.

3.5 Visualization

Once the model files were generated, **Flask** was used to create a front-end dashboard for the end-user. The user has the ability to choose the City and the Department for which he wants to view the predictions for. The application then identifies the date of the latest transaction from the dataset and then calls the weather API to get the weather information for the upcoming week. This information is then fed into the appropriate model and the predictions are displayed on the dashboard using **ApexCharts.js**.

4 Methodology

Our aim was to model the time-series data containing historical retail sales transactions that now include the weather as additional feature. The total sales is a continuous feature that changes over time which is in turn influenced by weather. Therefore, the model we had to build was a supervised regression algorithm. Classical algorithms like Linear regression, Logistic regression and Binary Tree regressors do not perform well on this data which requires the model to capture the trend and seasonality with respect to time.

4.1 Scoping

Weather is a volatile component and can change any moment. For the scope of this project, we assume that the weather predictions are fairly accurate for at least a week. Since the weather is our primary predictor, we scope our sales predictions for only the upcoming week and beyond that, the unpredictability in weather would lower the confidence of our sales predictions.

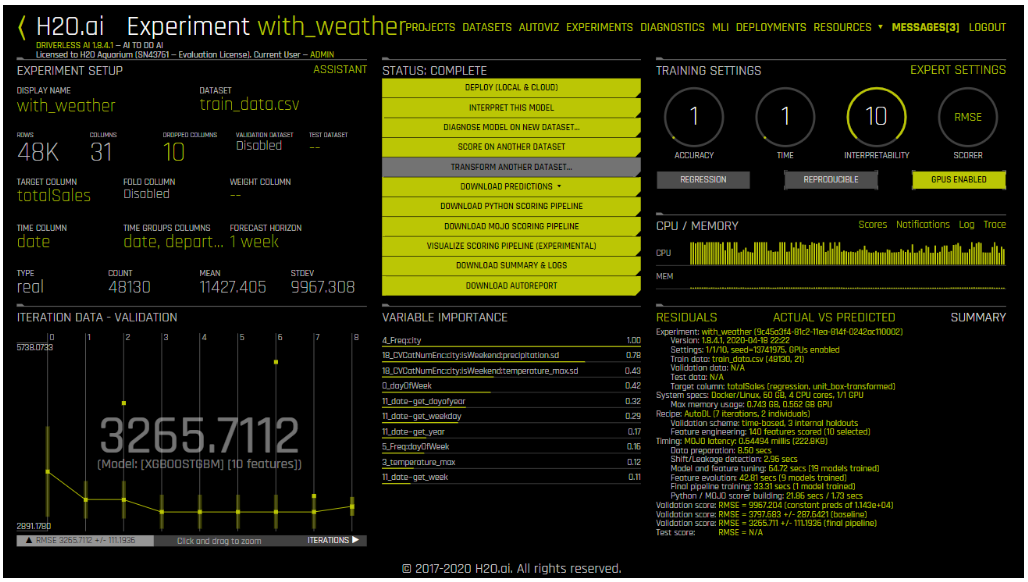
4.2 Train-Test Split

For modeling, we split the historical data from 2014 to 2019 into train and development sets. The train set consists of nearly 50000 sales transactions from Jan 2014 to Sept 2019 and the test set consists of about 3000 sales transactions in the last three months of data, Oct 2019 to Dec 2019.

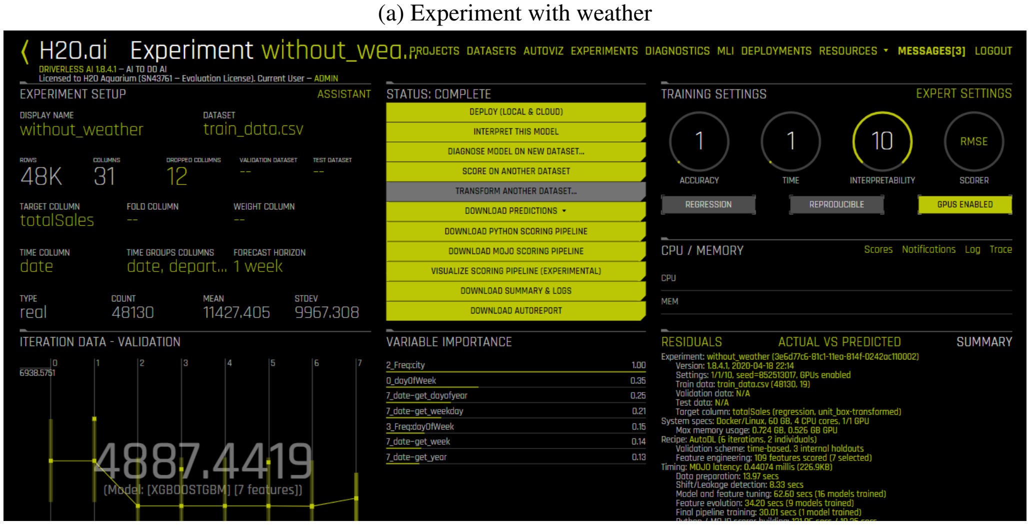
4.3 Modelling

4.3.1 H2O Driverless AI

H20 Driverless AI, a product by H2O.ai, is capable of performing feature engineering as well as finding ML models best fit for your data and are capable of deployment. This was used to compare different model performances, and to have an idea about the impact of weather on model performance. We ran two experiments (**Ablation study**), one with the weather included in the feature vectors and one without it.



(a) Experiment with weather



(b) Experiment without weather

Figure 3: H20.ai experiments run on H2O.ai Aquarium server

Inclusion of temperature and precipitation in modeling, reduced the RMSE of the model considerably. Also checking the variable importance of the XGBoost model implemented by H20.ai showed that temperature played an important contributing factor towards the sales on city-level. This confirmed our initial hypothesis, and we explored different approaches to model this.

4.3.2 FbProphet

Fbprophet is a decomposable time series model which is made of three main components: trends, seasonality and holidays.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

Fbprophet uses Fourier series to fit the seasonality component which allows it produce flexible model of periodic effects. For incorporating holidays, since they are not periodic and irregular - each effect of holiday is considered to be independent.

We therefore model for each city, with the aim of predicting sales on city level. This gave separate models for each city.

To predict quantity required for each department, we further made models for each department present in each city (with and without weather). This gave us nearly 320 models on the department level, with weather and without weather. City-departments combinations with less than 1000 training records could not be modeled as they wouldn't generate accurate results.

The following plots 4 show that for city Medicine Hat and Department 2, FbProphet picked up trends and seasonality accurately of the quantity of products sold.

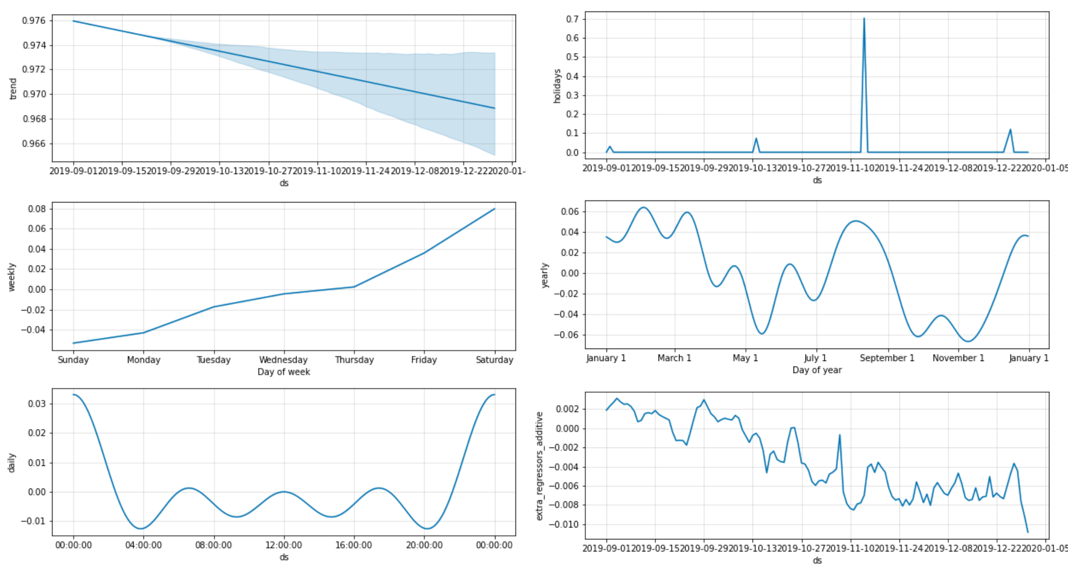


Figure 4: Seasonality and trends for Medicine Hat and Department 2 for the quantity of products sold

4.3.3 LightGBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency
- Lower memory usage
- Support for parallel and GPU learning
- Capable of handling large-scale data

Modeling with LightGBM requires an ample amount of feature engineering. 50 additional features were created apart from the original features present in the dataset.

- Features capturing holidays, weekends and other important events like Black Friday and Cyber Monday were also generated

- To apprehend the cyclic nature of the feature like dayOfMonth, month, weekOfYear, we took sine and cosine of them
- Frequency and Label encoding for the province and city columns was done as they were categorical features

4.3.4 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost required a lot of manual hypertuning. Due to large training time, manual hyperparameter tuning was chosen over GridSearch or RandomSearch hyperparameters tuning. We tuned some of the parameters like colsample_bytree, n_estimators, max_depth and some of the regularization parameters like reg_alpha and reg_lambda to make sure that out model isn't overfitting.

XGBoost showed some prominent results as the error decreased significantly from our previous two models.

4.3.5 Stacked Ensemble

Stacked generalization consists of stacking the output of individual estimators and use a regressor to compute the final prediction. Stacking allows using the strength of each individual estimator by using their output as an input of a final estimator.

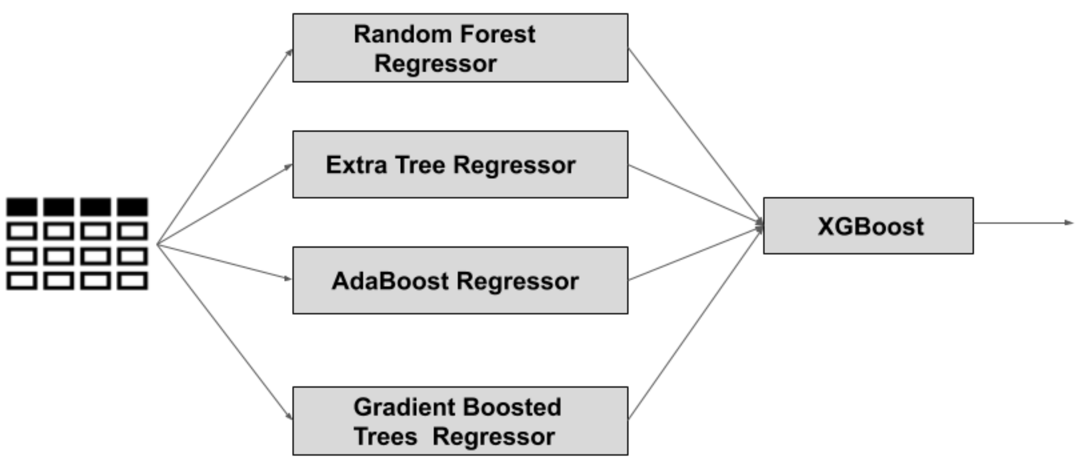


Figure 5: Architecture of Stacking model

We used four models in the level-0 layer which were:

- Random Forest regressor
- Extra Trees regressor
- Adaboost regressor and
- Gradient boosted trees

For the level-1 layer, we used XGBoost which worked as the meta-model and runs on the predictions from the level-0 layer regressors. We used the same hyperparameters as that of the vanilla XGBoost as it gave good results.

5 Evaluation

In the plot 6 below, we can see that the model has successfully taken into account major events like Black Friday Sales, Christmas Holidays and average temperature. We see major rise in the month of November (shown as dip in log values). The actual values closely match with the sales predicted by our model.

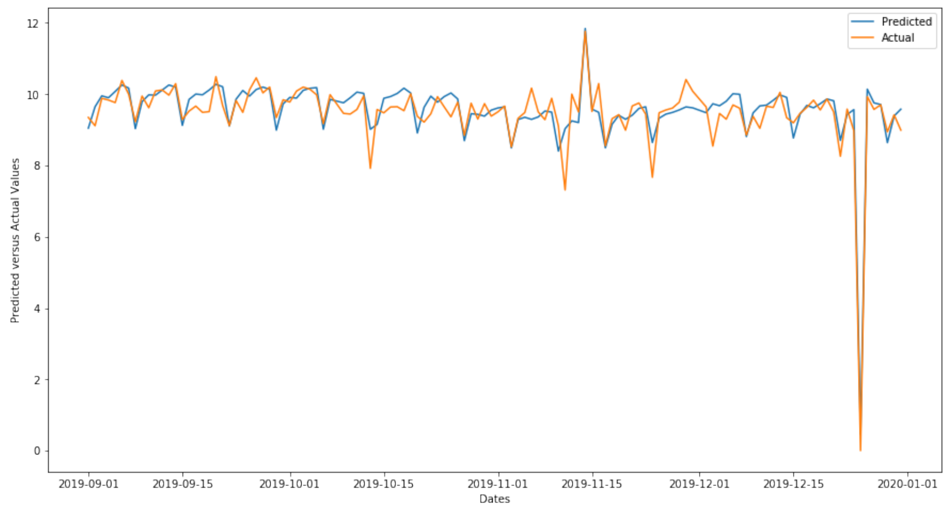


Figure 6: Predicted versus Actual Total Sales for Kelowna City

We conducted this experiment with and without considering weather features to find if weather actually has an impact on sales. Turns out, however minute, weather does have an impact on the retail sales.

We also made models for department level to forecast the quantity required in each department for the upcoming week. These city-department combinations produced nearly 320 models. We did not model for departments that had less than 1000 data points.See 7

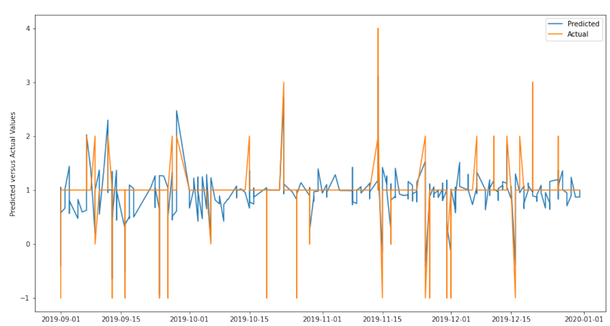


Figure 7: Predicted versus Actual Quantity for Medicine Hat, Department 1

Models	With Weather Data			Without Weather Data		
Evaluation	**RMSE**	**MAE**	**R2**	**RMSE**	**MAE**	**R2**
H2O.ai	3.62	NA	NA	3.51	NA	NA
FbProphet	0.72	0.44	0.73	0.72	0.43	0.73
LightGBM	0.75	0.33	0.85	0.71	0.29	0.85
XGBoost	**0.07**	0.15	**0.98**	0.21	0.21	0.96
Stacked Ensemble	0.20	**0.14**	**0.98**	**0.17**	**0.14**	**0.98**

Table 1: City level model evalutation metrics

Models	With Weather Data			Without Weather Data		
Evaluation	**RMSE**	**MAE**	**R2**	**RMSE**	**MAE**	**R2**
H2O.ai	NA	NA	NA	NA	NA	NA
FbProphet	0.22	0.20	-0.02	0.22	0.20	-0.02
LightGBM	0.08	0.13	0.55	0.08	0.13	0.55
XGBoost	**0.07**	0.12	**0.57**	**0.07**	0.12	**0.56**
Stacked Ensemble	0.08	**0.11**	0.55	0.08	**0.12**	0.54

Table 2: Department level model evaluation metrics

6 Data Product

After choosing our best-fit model, **Stacked Ensemble**, we saved the files generated for each city and the city-department combination. **Flask** app was used to create a dashboard for the end-user. User has the flexibility to either view the historic data as a time-series, or select the City and the Department for which they wants to view the predictions for. Based on the last recorded entry in the dataset, the application then gets the dates of the upcoming week and then calls the API to get the weather information for that week. This information is then fed into the appropriate model and the predictions are displayed on the dashboard using **ApexCharts.js**.

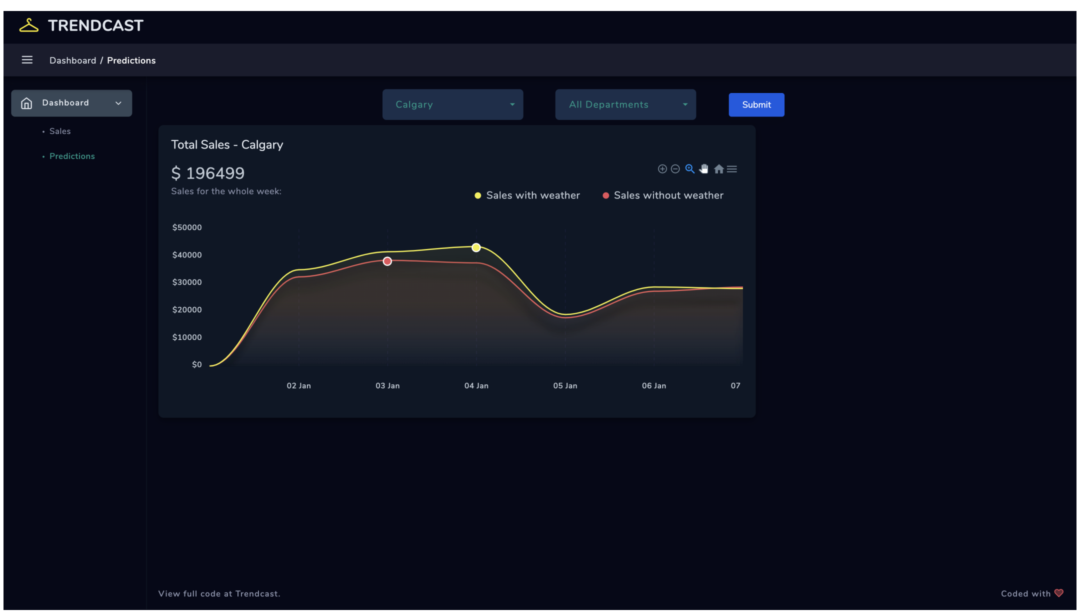


Figure 8: City level forecasting

City level view (Fig: 8) gives the sales forecast of the city for the upcoming week while the other (Fig:9) gives the forecast of quantity for the upcoming week for the city and the department selected.

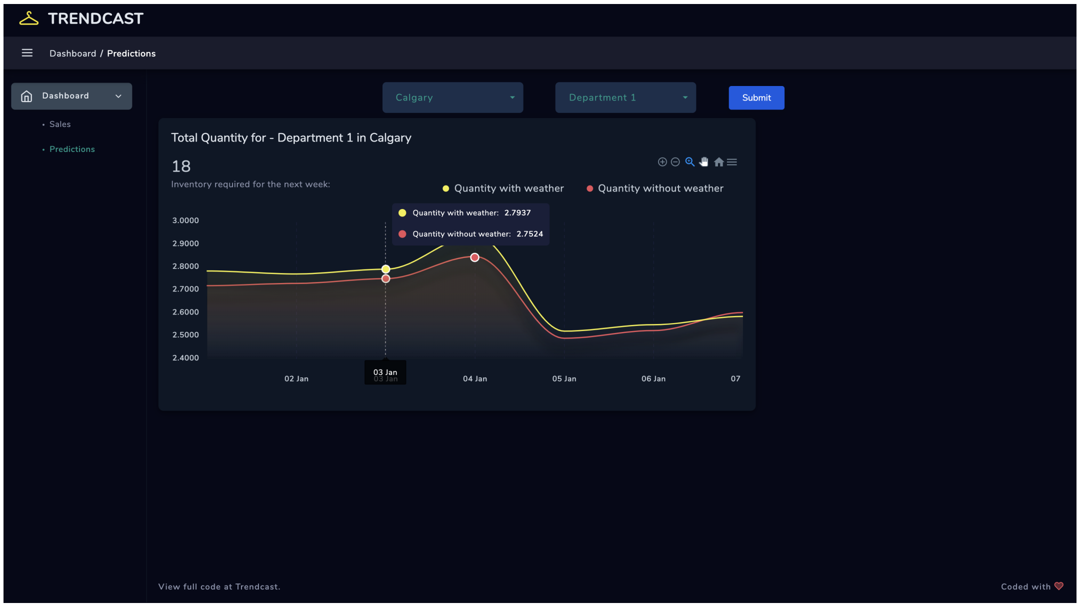


Figure 9: Department level forecasting

7 Lessons Learnt

1. Configuring Auto-ML tools such as H20.ai can be beneficial for understanding the underlying assumptions in data during the analysis phase.
2. Understanding various time-series forecasting techniques. Time-series data needs to be modeled differently from the conventional regression techniques
3. When there are multiple distributions, it is difficult to generalize a model. It can be a better approach to model separately for each independent distribution.

8 Conclusion

This project showed that weather does have a modest impact on retail sales for some cities and classes of departments. Future analysis could work on modeling on the granular level of data such as style and class, which would give even more accurate forecast regarding the items needed to stock for the coming week. In this project we focused on **ablation study**, to analyse the impact of weather. A statistical approach to understand causality of weather such as Granger-Causality test could also be applied. The forecast horizon could also be extended for reliable predictions in future studies.

References

[1] F. Badorf and K. Hoberg, "The impact of daily weather on retail sales: An empirical study in brick-and-mortar stores," *Journal of Retailing and Consumer Services*, vol. 52, p. 101921, 2020.

[2] K. B. Murray, F. D. Muro], A. Finn, and P. P. Leszczyc], "The effect of weather on consumer spending," *Journal of Retailing and Consumer Services*, vol. 17, no. 6, pp. 512 – 520, 2010.

[3] M. Starr, "The effects of weather on retail sales," *SSRN Electronic Journal*, 01 2000.

[4] M. T. Islam Molla, "Impact of weather on u.s. apparel retail and wholesale sales," *University of Missouri–Columbia*, p. 111, 05 2016.