University of Amsterdam

Project

Brane Programming Project

June 5, 2023

Students:
Adithya Vasisth
adithya.vasisth@student.vu.nl

Sudarsan Sivakumar
s.sivakumar@student.uva.nl

Konstantinos Mitropoulos
k.c.mitropoulos@student.vu.nl

Group:
Group 11

Course:
Web Services and Cloud-based Systems

1 Introduction

In this project, we have created a data processing pipeline using Brane [2]. The data processing problem we have chosen is the analysis of store sales with respect to oil prices and holidays in Ecuador [1]. We also perform the time-series analysis of the data provided and try to forecast future sales data.

2 Background on Brane

Brane is a programmable pipeline orchestration and scheduling framework. A Brane instance consists of a control node and worker plane. The worker plane consists of a set of compute backends. Brane is designed to support various compute backends, including Docker, VM, Kubernetes, and Slurm, providing flexibility in accommodating different computing environments. For our specific project, we opted to employ a single VM-based compute backend within the Brane framework.

3 Problem

The problem involves using time-series forecasting techniques to predict store sales for Corporación Favorita, a major grocery retailer based in Ecuador. The objective is to construct a model that can accurately forecast the unit sales of numerous items across various Favorita stores using the provided dataset briefed in the following section.

4 Dataset, Preprocessing, Visualization, Training, and Prediction

4.1 Dataset

The dataset provided for this competition comprises time series data obtained from Favorita stores located in Ecuador. This dataset encompasses various features, including store number, product family, promotion status, and sales figures.

The 'train.csv' file contains the training data, which encompasses details regarding store numbers, product families, promotion status, and the corresponding sales figures. This data serves as the foundation for training models and understanding the relationships between different variables.

The 'test.csv' file has similar features to the training data and the main goal is to forecast sales figures for the next 15 days after the last recorded date in the training data. This requires using trained models on the test data to produce precise predictions.

Supplementary files are available to improve the analysis process alongside the core datasets. These additional files consist of store metadata, providing more details about the stores. Additionally, daily oil prices are included, allowing potential correlations or insights to be investigated between oil prices and store sales. Lastly, information on holidays and events is provided, which may help identify external factors that could impact sales patterns during specific periods.

4.2 Preprocessing

In the data preprocessing stage, unnecessary columns are removed, selected columns are transformed to the appropriate data types, and aggregation is performed by grouping the data by store numbers and summing the sales. The processed data is merged with store location information and transaction data, including lag features. Holiday data is adjusted(one hot encoding), sales data is scaled, and lag features are created for time series analysis. The test data is modified for prediction purposes, missing values are removed, and the "date" column is set as the index for easy chronological access.

4.3 Visualization and Analysis

We tried to establish if there is any statistically significant relation between considered features and sales.

Correlation between oil prices and sales

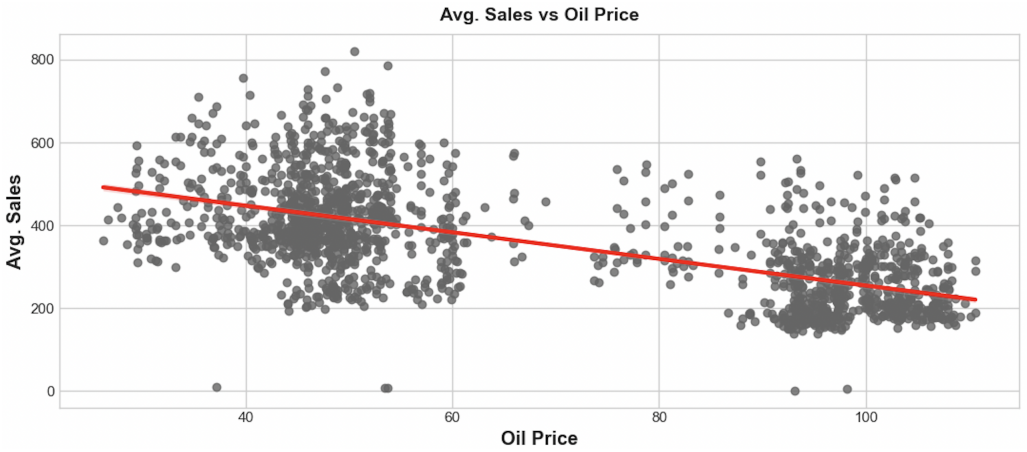

Figure 1: Correlation between oil prices and sales

The plot in Fig1 displays the average sales in relation to various oil prices. This was obtained by merging data from oil.csv and sales.csv based on dates. It appears that there is a decline in sales as the price of oil increases, but the correlation's absolute p-value (0.6) is lower than anticipated. We have chosen to exclude this feature from our training and prediction, as there was a continuous trend of a decrease in oil prices and an increase in sales.

Correlation between holidays and sales

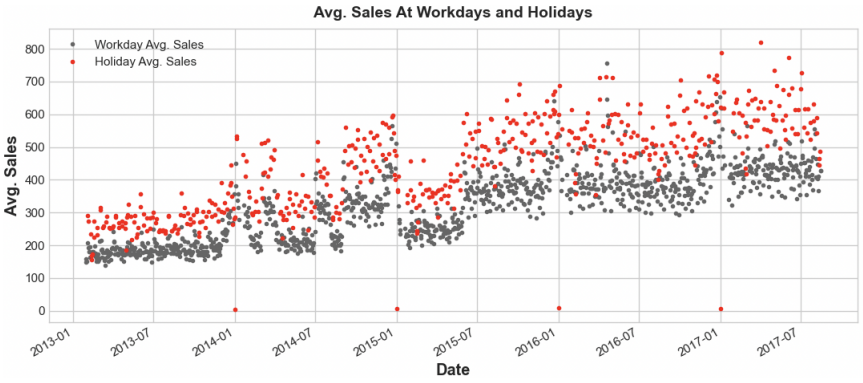

Figure 2: Visualization of sales with respect to holidays and working days

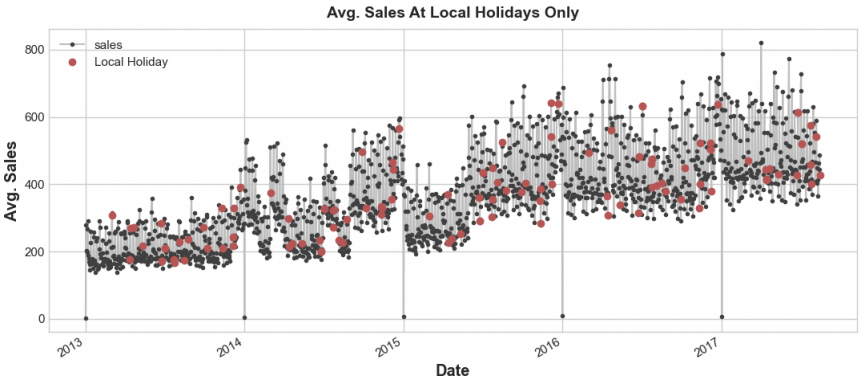

Figure 3: Local holiday vs. sales

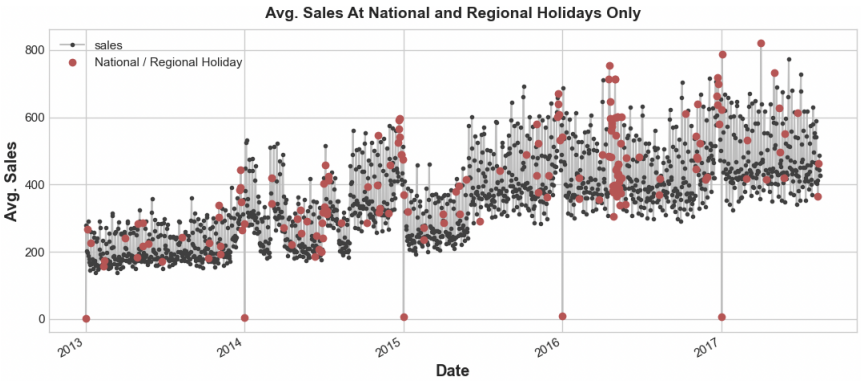

Figure 4: National holiday vs sales

It's clear from Fig2 that holidays have a significant impact on sales. We wanted to dig deeper and analyze the correlation between local holidays and national holidays, so we created Fig3 and Fig4. Although there's no correlation between different types of holidays, we noticed a pattern of increased sales during holidays every year. We then analyzed the effect of each holiday on sales and only used the data for training if the correlation value was significant.

4.4 Training and Prediction

We used the simple linear regression model to establish a relation between the input features (Date, Store number, Product family, Holiday information and etc.) to Sales. This straightforward and interpretable model should perform effectively because we have identified in the analysis previously that the relationship between the features and the target variable exhibits linearity to some degree.

5 Brane Pipeline Implementation

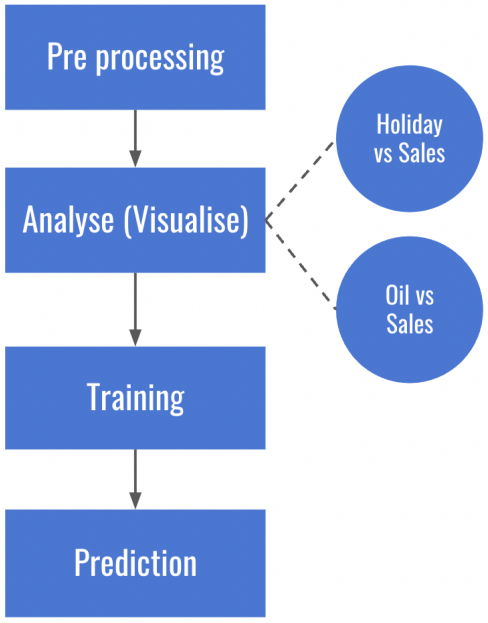

Figure 5: Brane pipeline implementation flow

We implemented our solution to be in sequential order. We first store all the necessary *CSVs* in a folder called *data*, which we use to build the dataset. We passed the entire folder instead of the files, as we had about 7 input files, and passing the folder was easier for the code processing.

In the *pipeline.bs* file, we then run the three packages in the following order:

1. Preprocessing
2. Visualize
3. Compute
   (a) Training
   (b) Prediction

In the preprocessing package, we provide the path where the datasets were built and are available in the Brane platform. We use this *dataset-path* to read the datasets and perform some preprocessing and cleaning for the further packages. Once the package processing is at its final stages, we also plan to use the same *dataset-path* to store the pickles for the next two packages. We initially planned to run the visualization package and the compute package parallelly as they did not have dependencies on each other. But as we were not able to get our pipeline running, we did not parallelize them.

The visualization package takes as input the dataset path, where it should read the data that was processed as pickles, and the path where it should output the images. It also takes an argument *holiday, oil, or both* to generate the images for our research question. The compute package also takes the same inputs as the visualization package - regarding the dataset path. Once the package is executed, we will find the *submission.csv* file that holds the prediction of future sales and the solution to the Kaggle challenge.

The *container.yaml* for each package is specified for each package. It has a few common dependencies, such as *python3 and python3-yaml*. We then install the necessary libraries using *pip3* for each package.

6 Conclusion

We tried our best to carry forward the great momentum we had from the previous three assignments, but we were unable to completely implement what we set out to do. We were unable to set up the pipeline, and we started facing errors in the beginning stages of the preprocessing package, which is the first in the pipeline that should have been executed. We spent about three-five days completing our code for preprocessing, analysis, visualization, training, and prediction. After which, we set out to deploy our code on Brane. We kept facing multiple issues. We prioritized our focus initially on running example projects and setting up our Brane infrastructure on two VMs. But we were facing errors, and we also got some clarity from the TA on how the scheduling takes place and the challenges of scheduling especially concerning sensitive data and policies.

We then went ahead to set up our infrastructure on a single VM, i.e., by having the control node and worker node in the same VM. We then encountered a major roadblock while setting up the data processing pipeline for our project. Despite our best efforts, we faced numerous errors that proved difficult to troubleshoot. One persistent issue was the termination of our preprocessing function with an error message stating, *"Internal package call was forcefully stopped with signal 9"*. The code itself didn't contain any syntax errors, but it abruptly terminated without providing any additional information. To investigate further, we resorted to commenting out each line of code, only to find that it consistently failed at line 27 in preprocess.py. Unfortunately, we couldn't identify any apparent reasons for the failure, leaving us stuck at this point. Consequently, the progress on other files became stalled, as we couldn't proceed with the pipeline until the preprocessing package issue was resolved. We suspect that the handling of a large number of input files amplified the challenges we faced. If we had been dealing with just one dataset and passing it as an intermediate file along the pipeline, the situation might have been less complex.

In the future, we would aim to fix existing issues and learn about how multiple datasets are handled in the pipeline and how time-series analysis problems are dealt with in the Brane platform. We would also want to implement our original idea of performing the compute and visualization parallelly after the preprocessing package has finished its execution. We also would like to implement multiple training and prediction models and not just one linear regression model. We also had the idea to make the preprocessing generic enough to perform time-series analysis on a dataset provided in any format, by having the user enter their code for preprocessing as input and we could use this code further in the pipeline. This could be an interesting project, if not explored already.

7 Work distribution

We believe we distributed the work evenly between the three team members in coding and writing this report. We had a few discussion sessions and had no friction when working together. The work split is mentioned below:

- **Konstantinos** - Implementing the analysis done by Adithya in Brane.
- **Sudarsan** - Report writing and helping Konstantinos with Brane implementation.
- **Adithya** - Data analysis, helping in debugging the issues in Brane, and writing some parts of the report.

References

[1] Kaggle. *Store Sales - Time Series Forecasting*. URL: https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data.

[2] Onno Valkering, Reginald Cushing, and Adam Belloum. *Brane*. Version 0.4.1. Aug. 2021. DOI: 10.5281/zenodo.5205666. URL: https://doi.org/10.5281/zenodo.5205666.