ML Assignment: Demand Sensing Problem

Mark Kuiack

For Nike ML Engineer position

Challenge:

In order to optimize employee staffing the store planning team requires a reliable sales forecast up to one month in advance.

For this they require:

For any given day, up to 30 days ahead of the current day, what is the likely total product sales volume across all products and stores.

Resources:

A historical dataset with sales numbers for for each product and store from 01/01/2017 to 30/09/2019.

Methods

I investigated a number of different algorithms and feature sets to determine which ML model to deploy.

For each method I trained the model on 400 days of data then forecasted the next 30 days. I compared this prediction to the true values by calculating the Root Mean Squared Error (RMSE)

> The RMSE is a good metric because it tells what is the average absolute difference in terms of the per unit amount. This can be helpful for stakeholders to estimate the uncertainty in sales in real terms, ie. Physical units, this can particularly help with planning stock, but also staffing.

and the Mean Absolute Percent Error (MAPE).

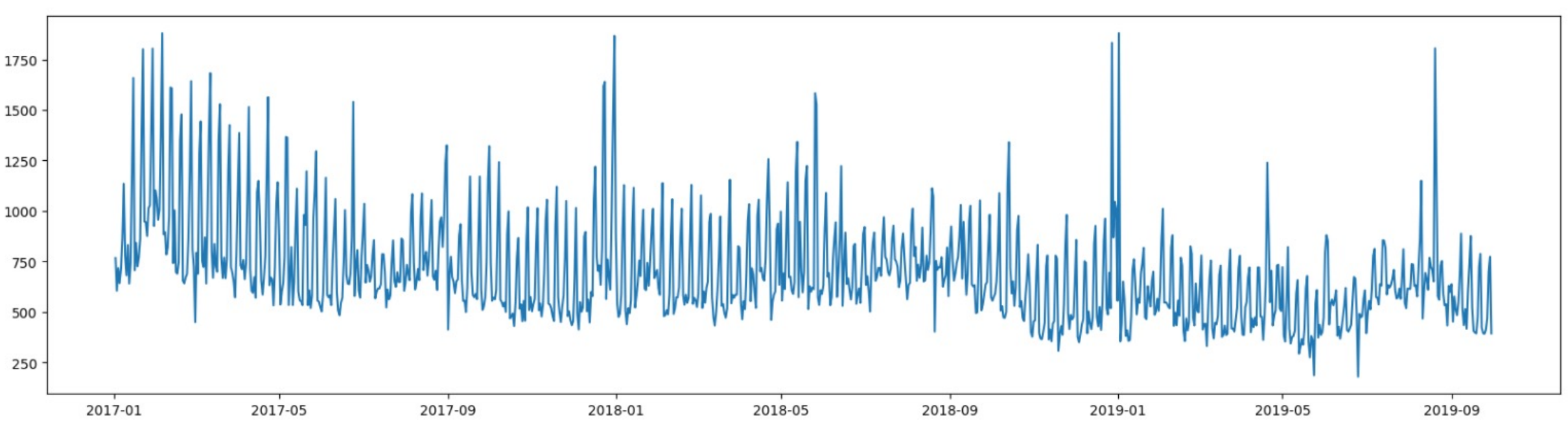
> The MAPE is also a good metric because it gives the average difference as a percent of the true value. This is generally intuitively understandable to all stakeholders, ie. product demand can be estimated within X%. This translates to margins on orders for example.

These metrics were evaluated 20 times on forecasts from 1-30 days from the end of the training data. The 30 test days always immediately followed the training data. The length of the training data was kept constant at 400 days for consistency.

Findings:

Analysis of total sales data:

1. The data primarily showed a 7 day periodicity, with sales consistently higher in the weekends.
2. There was a general trend of decreasing effect in the weekend-weekday difference. Older data was less representative, than more recent data.
3. Consistent sales increases in the weeks prior to Christmas.

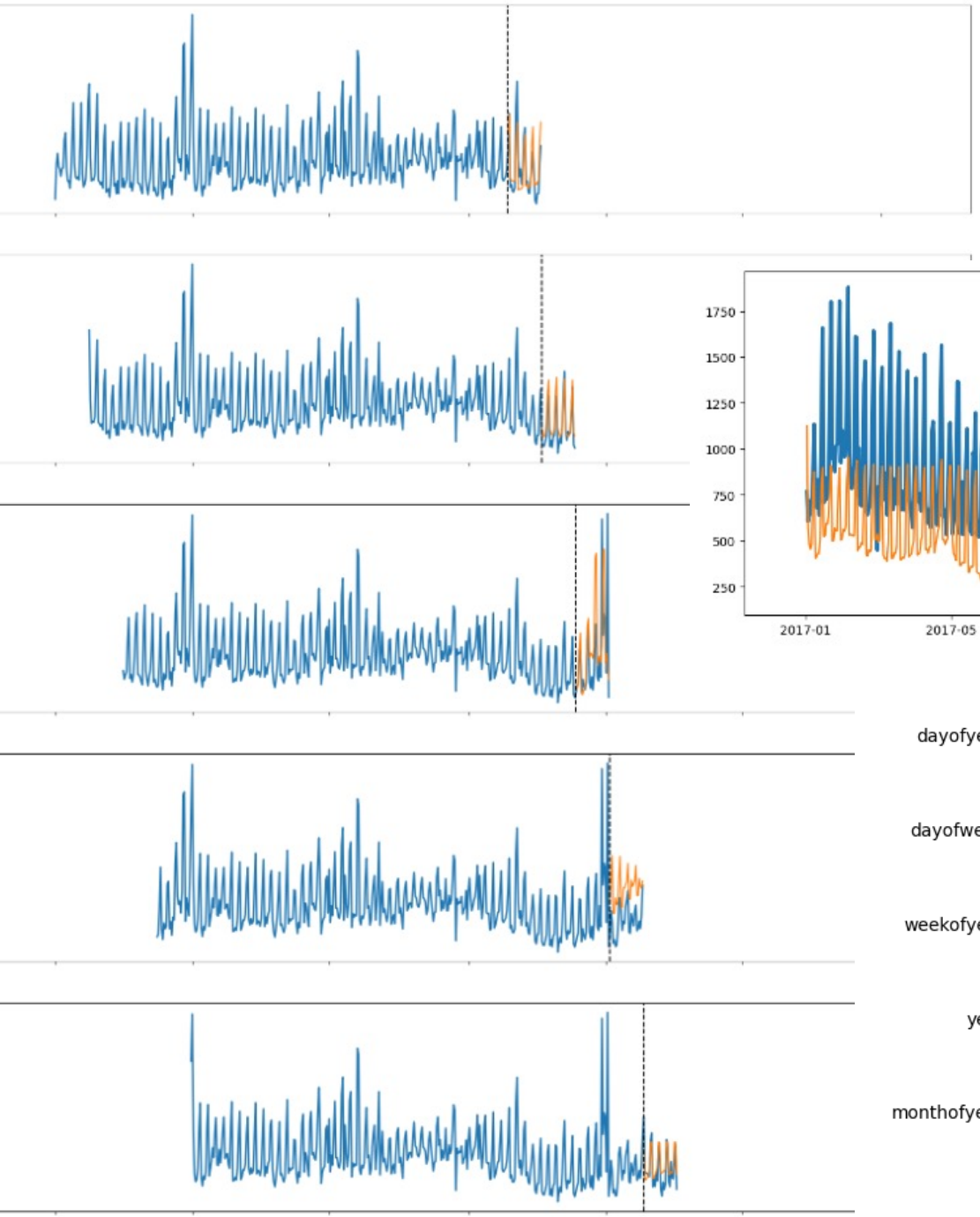


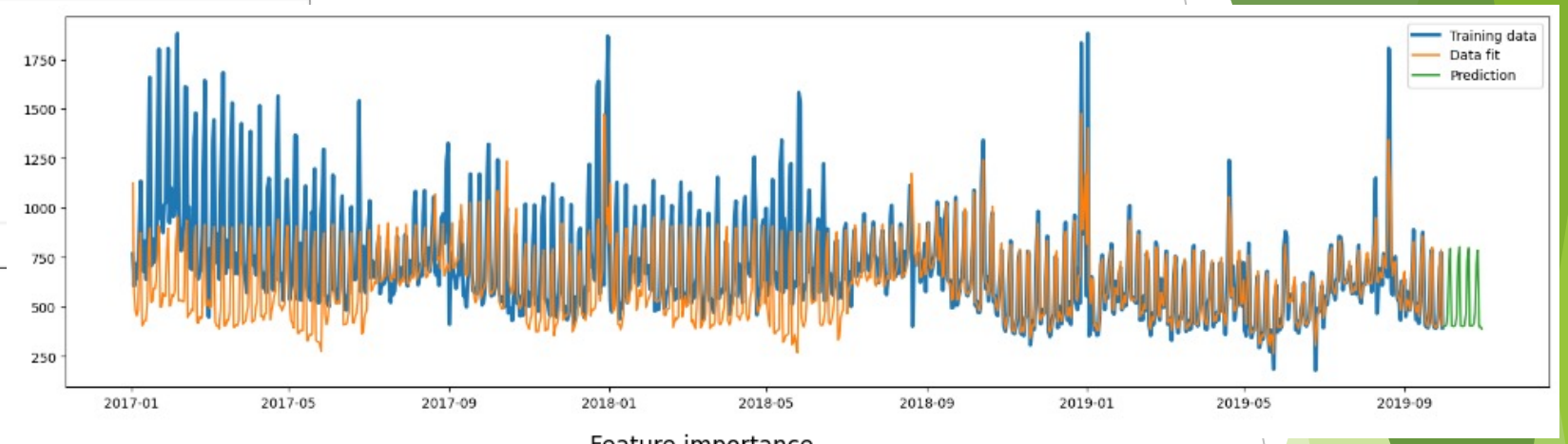
Findings:

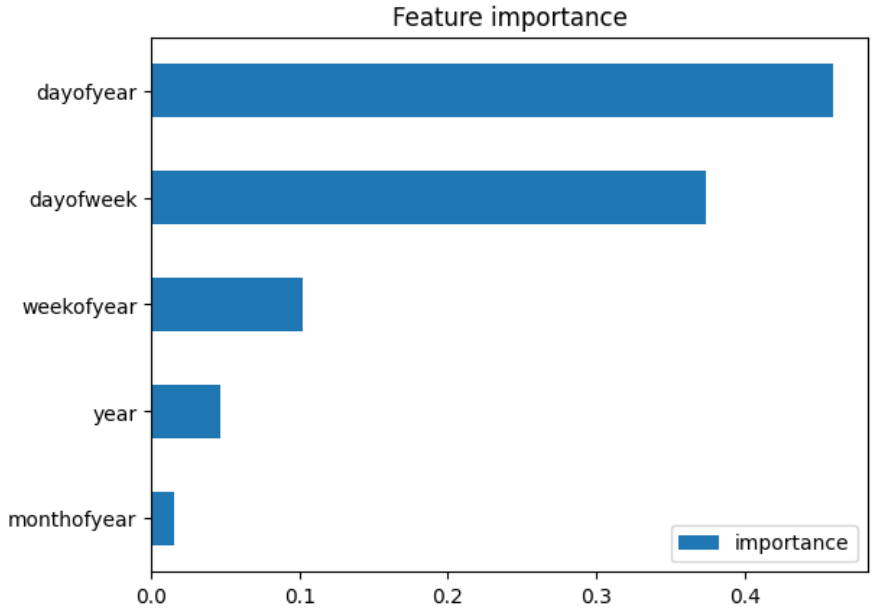
- The Random forest with the ordinally encoded dates was the model with the best statistical performance in the forecast, and was a simple implementation without the need for additional features or complications. Therefore, is the model I chose deploy
- See data_exploration.ipynb notebook for further details and figures.

Model	Feature set	Result
LinearRegression model	decomposing the date into 5 ordinally encoded features: <day of year>, <day of week> <week of year>, <month of year>, <year>	RMSE = 175.25 MAPE = 21.8 %
RandomForestRegresor	the same features above	RMSE = 159.9 MAPE = 19.3 %
RandomForestRegressor	OneHotEncoding the above features, rather than ordinal encoding,	RMSE = 165.63 MAPE = 19.6 %
RandomForestRegressor	OrdinalEncoding: <week of year>, <month of year>, <year> Binary feature: <is the date within -7 from Christmas to +3 days from new years> Binary feature: <is the date a weekend>	RMSE = 165.6 MAPE = 20.8 %
Lasso. A linear model with L1 regularization.	Repeating basis function representation of <day of year>, <day of week> <week of year>, <month of year>, <year>	RMSE = 178.12 MAPE = 22.4 %

Winning model performance!

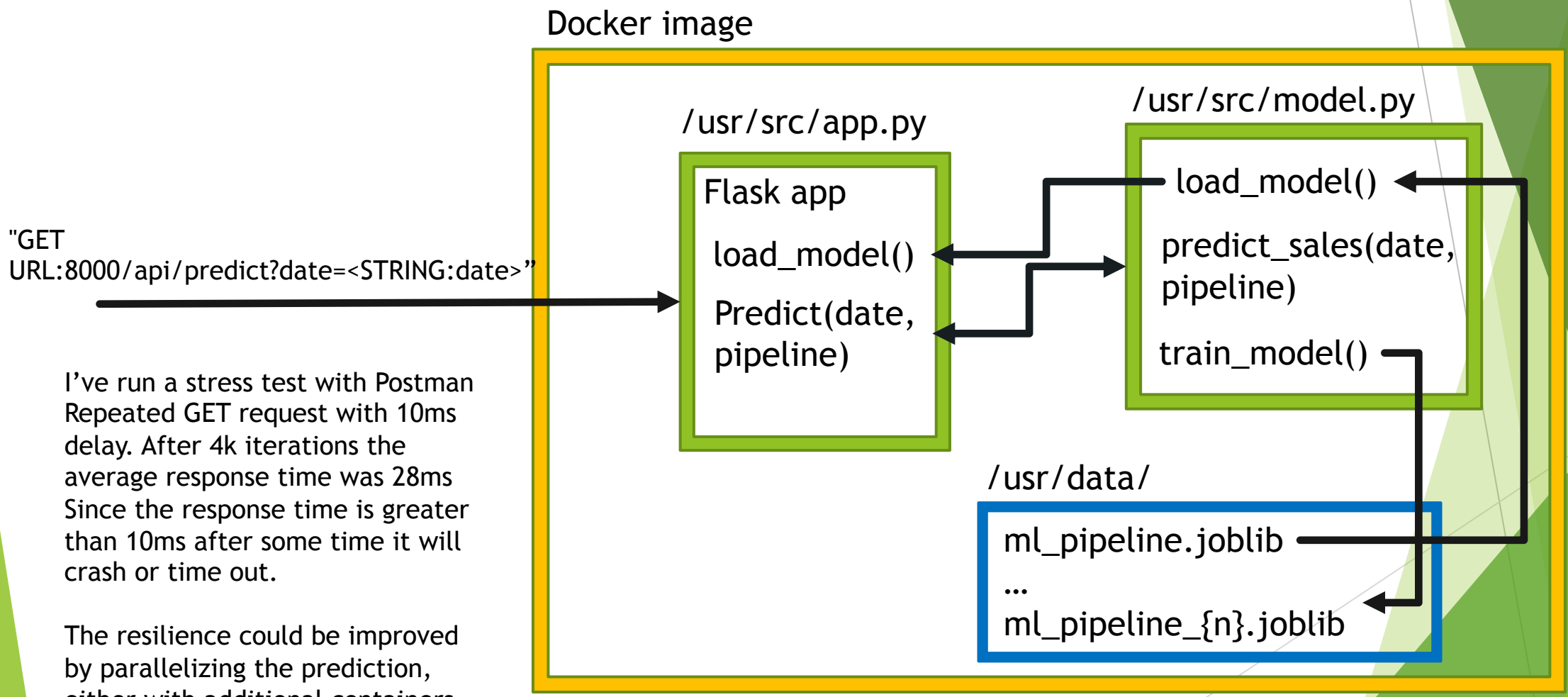






Technical

- Schematic of the forecasting app



I've run a stress test with Postman Repeated GET request with 10ms delay. After 4k iterations the average response time was 28ms Since the response time is greater than 10ms after some time it will crash or time out.

The resilience could be improved by parallelizing the prediction, either with additional containers, and a load balancer, like AWS ECS.