Technical Report: Project 2

Fan Yang (fanyang3) and Xiaozhu Ma (xiaozhu3)

Apr 10, 2021

Pre-processing Data

For both training and testing data, we first create features that will feed into the model.

One-hot encoder

- For each week in a year, we use one-hot encoder to make it to week_1, week_2,... week_52.
- For holidays, as there are four holidays across the year, we use one-hot encoder to make them as holiday_1, holiday_2, … holiday_4.

Some holiday may always happen in the same week in 2010 (i.e., Christmas). In this case, we drop that week's indicator variable.

Trend

For each week in sequence starting from 2010, we make a numeric variable `week_count`. For example, if a week has `week_count = n`, that means it is the `n`th week since 01/01/2010.

Previous Sales

We think the previous sales may help to predict future sales. As we only need to predict sales for the next 8 weeks in each fold, we create a variable that represents the previous sales of 9 weeks ago.

It turned out that this new variable adds more time and complexity in the training and prediction process, but does not improve the prediction power at all. So eventually we decide to not use it.

Training Method

We have tried:

1) Build a single model to predict all departments of all stores, with variable `store` and `department` being one-hot encoded.
2) Build different models for different departments, with variable `store` being one-hot encoded.
3) Build different models for different departments and different stores.

The method 3 performed the best.

Lasso vs Linear

We also tried both linear models and Lasso models. Linear models have better training performance, but their coefficient values tend to be off the scale in Python's scikit-learn (i.e., 10+16), and perform much worse than Lasso does.

We suspect that Lasso does a better job here because we have too many features and so of them may be irrelevant to the models.

Performance

Our performances for the 10 folds are

```
[2053.167060130686, 1476.5989977618594, 1459.056801047943,
1598.325720981124, 2340.961897031291, 1679.7237542894243,
1725.9148932138892, 1431.0930879612342, 1446.63182487708,
1445.6296619697382]
```

We use a MacBook Pro (16-inch, 2019) with 2.6 GHz 6-Core Intel Core i7 and 16 GiB memory, and training and prediction for the 10 folds takes 40 minutes.