Implementation of same-store solution

Preliminary version

Task

- Design a similar store MSI which could be used to identify floor execution problems, as suggested by https://cb4.com/.

What has been implemented and experimented?

- 1. Construct entity embeddings via NN for the categorical variables (such as Stores, weekdays etc) 👍
- 2. Visualize the interpretability of entity embeddings and verified its validity with real life patterns 👍
- 3. Tested the semi-supervised approach: not working well 👎
- 4. Compared with other trendy tabular models and optimized using Bayesian optimization. 👍

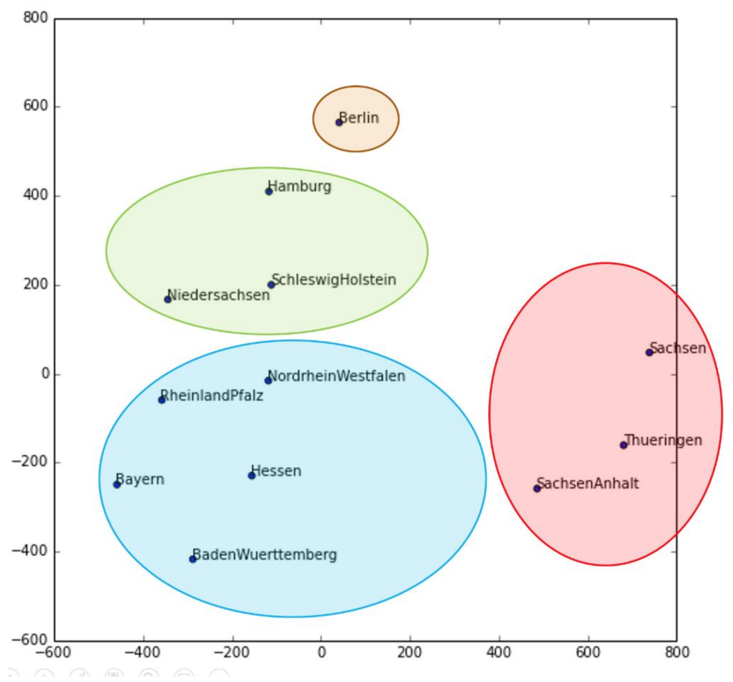
The toy dataset

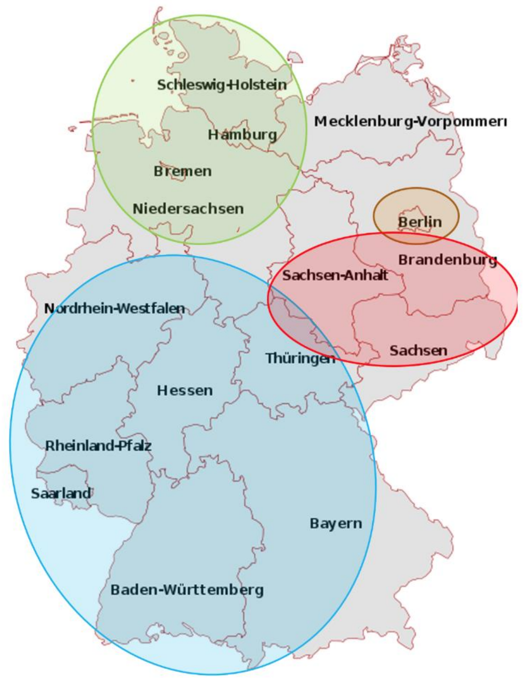
- Rossmann grocery store.
  - Historical sales data for 1,115 Rossmann stores. The task is to forecast the "Sales" column for the test set. Note that some stores in the dataset were temporarily closed for refurbishment.
  - Features include store level metadata, weather, and longitudinal metadata (e.g. holiday or not)
  - 844338 training instances
  - 41088 testing instances
  - Shortcoming: does not contain product level information.

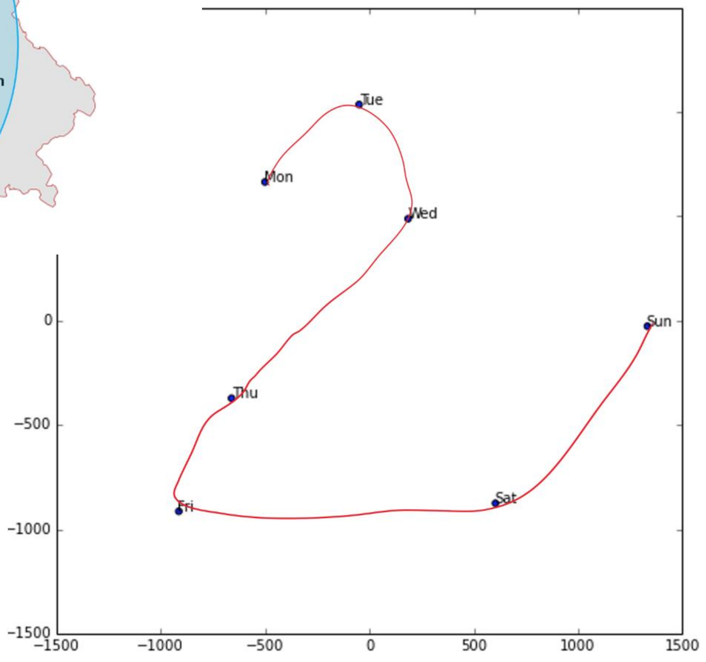
Advantage of the Entity embedding Neural Network method

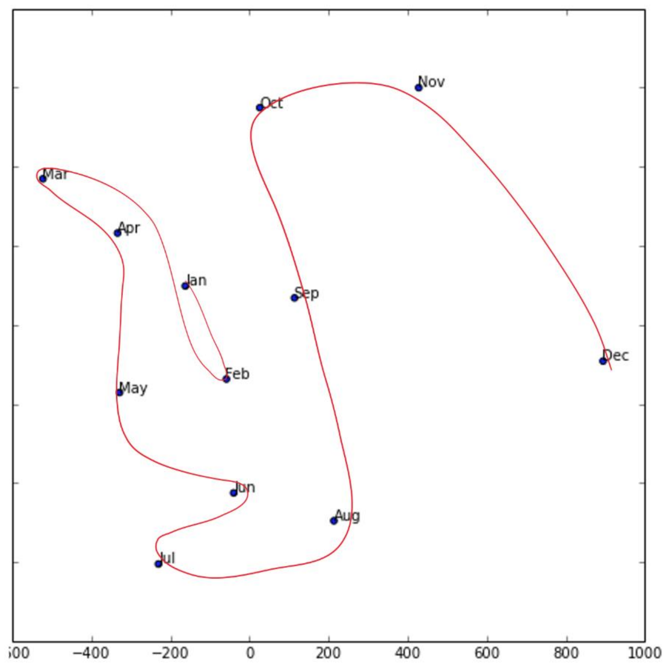
- **Transferable**
  The latent embeddings learned could be directly applied to future modeling process
- Latent embeddings carries **syntactical meanings** thus offering interpretability for the categorical variable
- Latent embeddings might be **more reliable**
  Latent embedding does not contain categorical variables which would cause troubles for clustering algorithms such as KNN
- **State of the art practices**
  Instacart
  Google
  Youtube
  Kaggle competitions
- A Good introduction to Entity embedding NN.
  which I helped proofread 🤗
- With little finetuning it achieved 188/3303 almost a silver medal for the competition. The metric RMSPE from the 1st to ours is only 0.10 to 0. 116.

Interpretability of the categorical variables

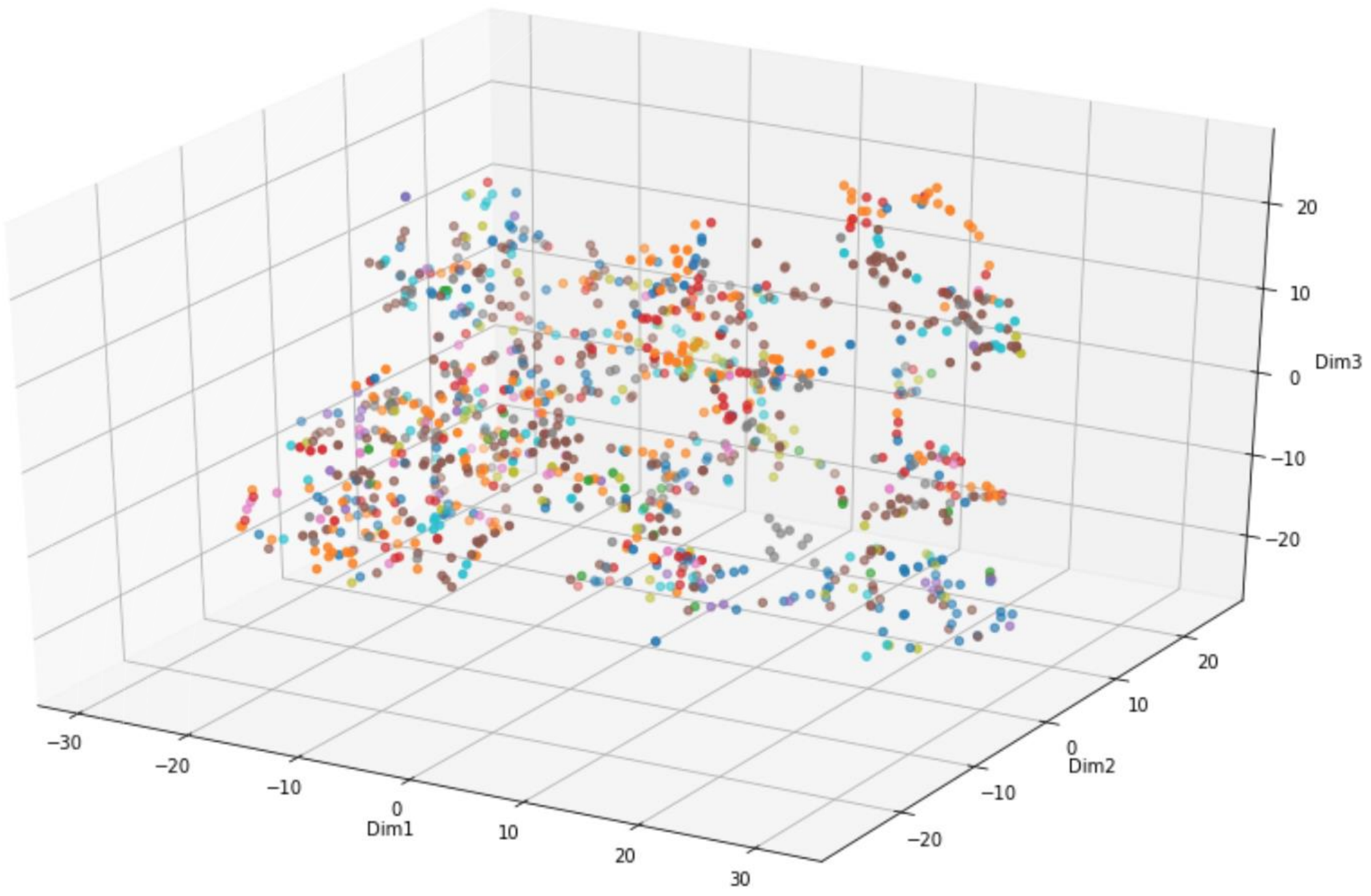




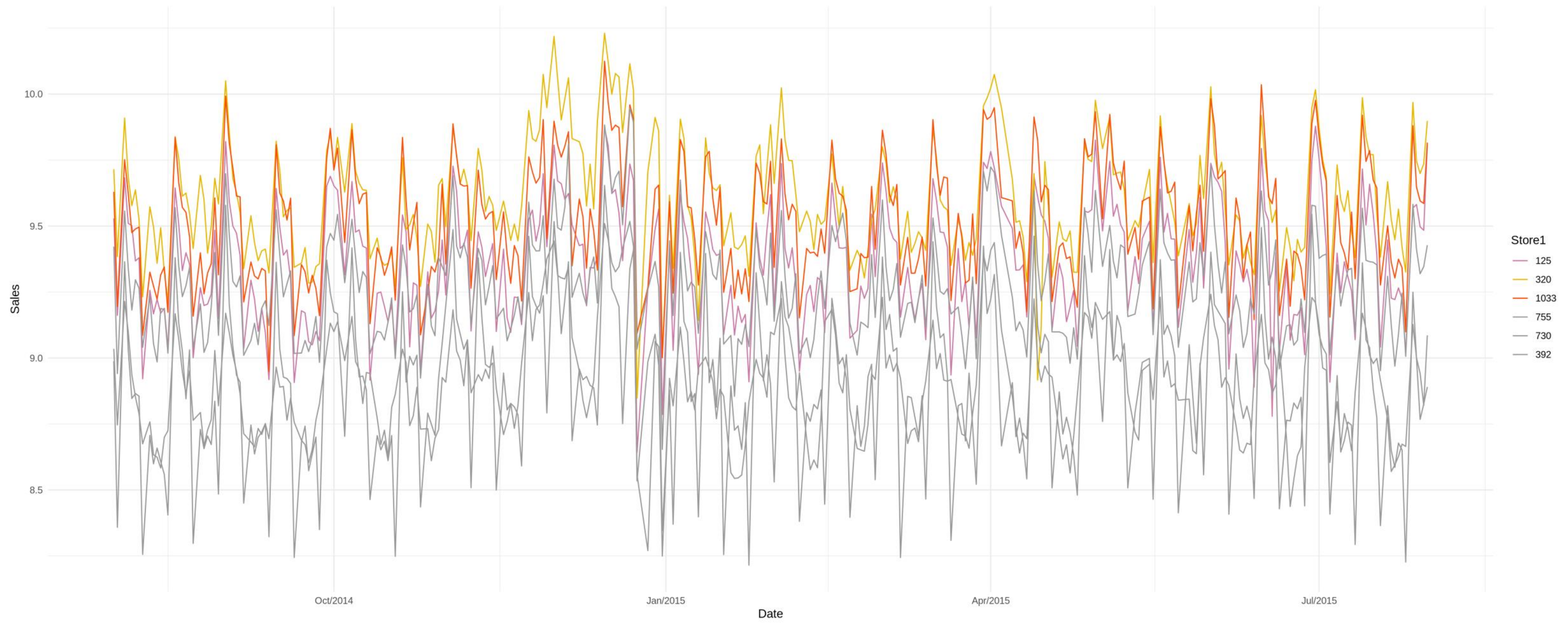




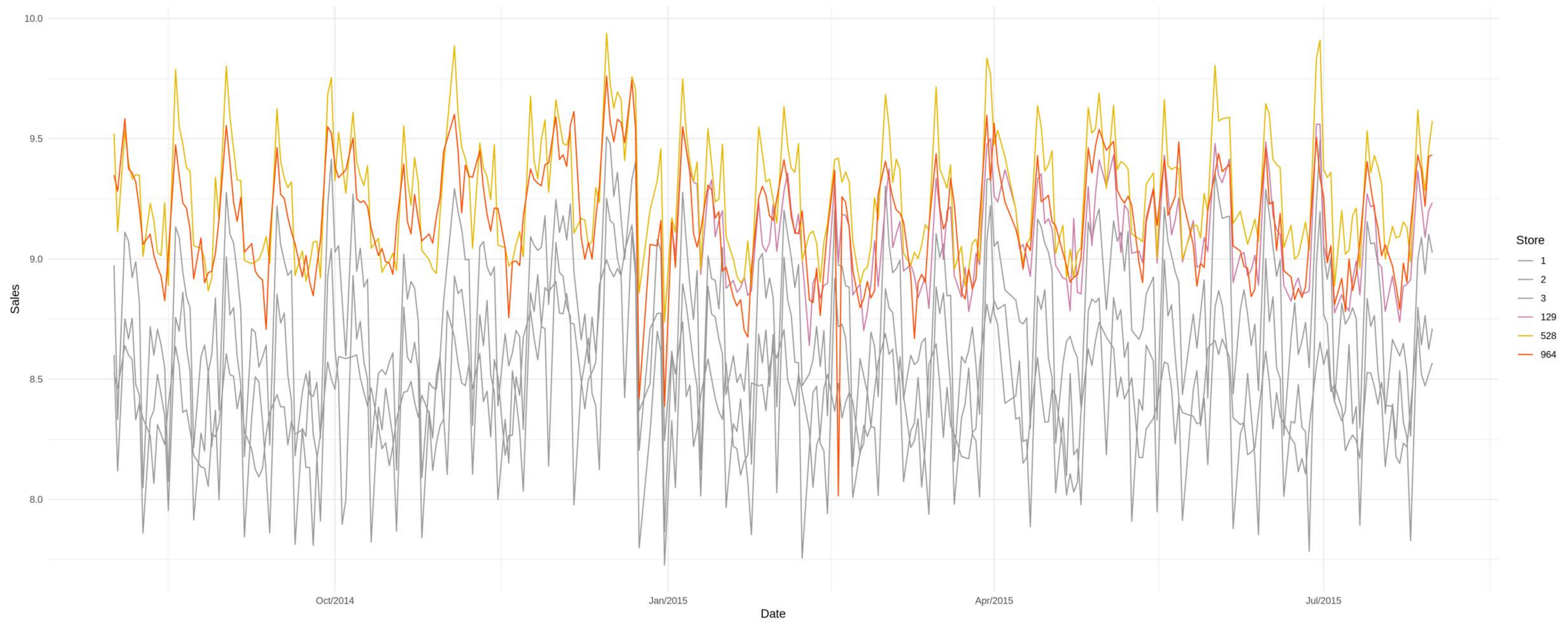
These were created in here: https://github.com/fastai/fastbook/blob/master/09_tabular.ipynb
I have recreated these plot but the figures look better with the Germany map or the line connection on it.



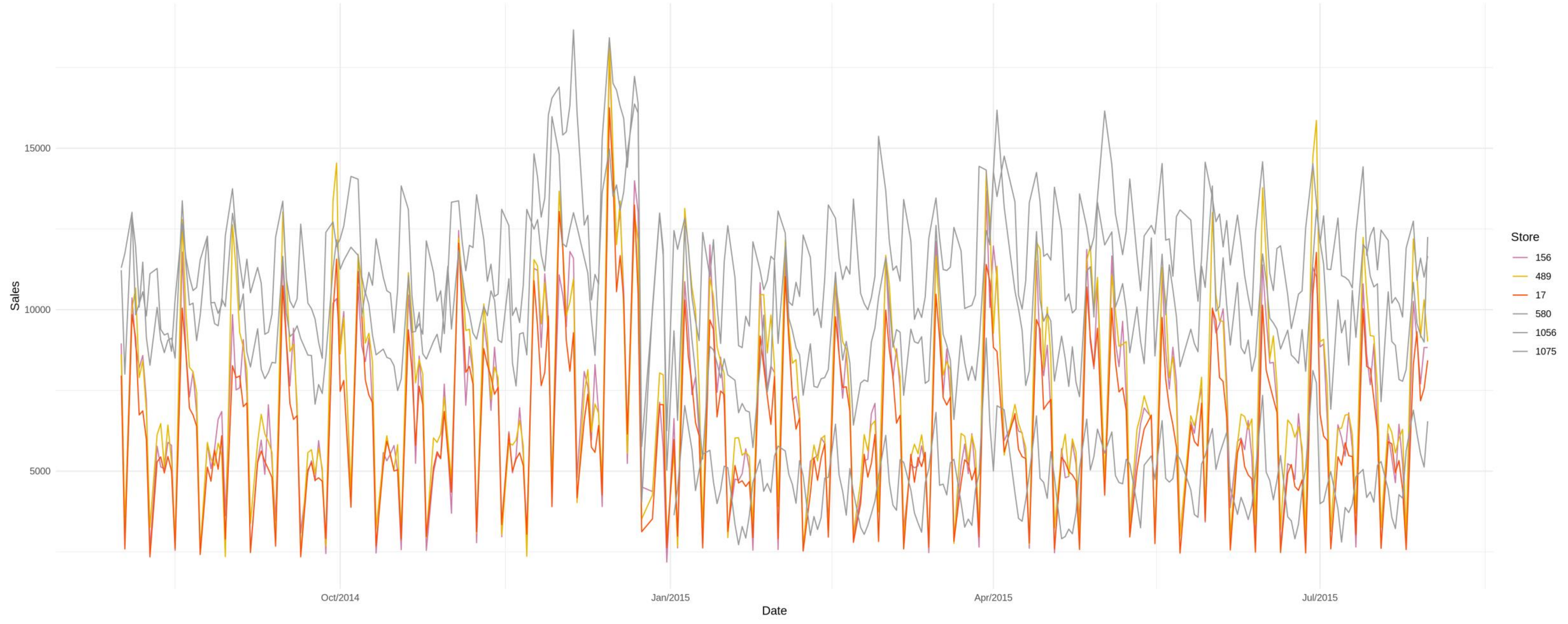
3D t-SNE visualization of 1115 stores colored by the state it locates. We could see stores are NOT distinctively different from each other if they are from different states.



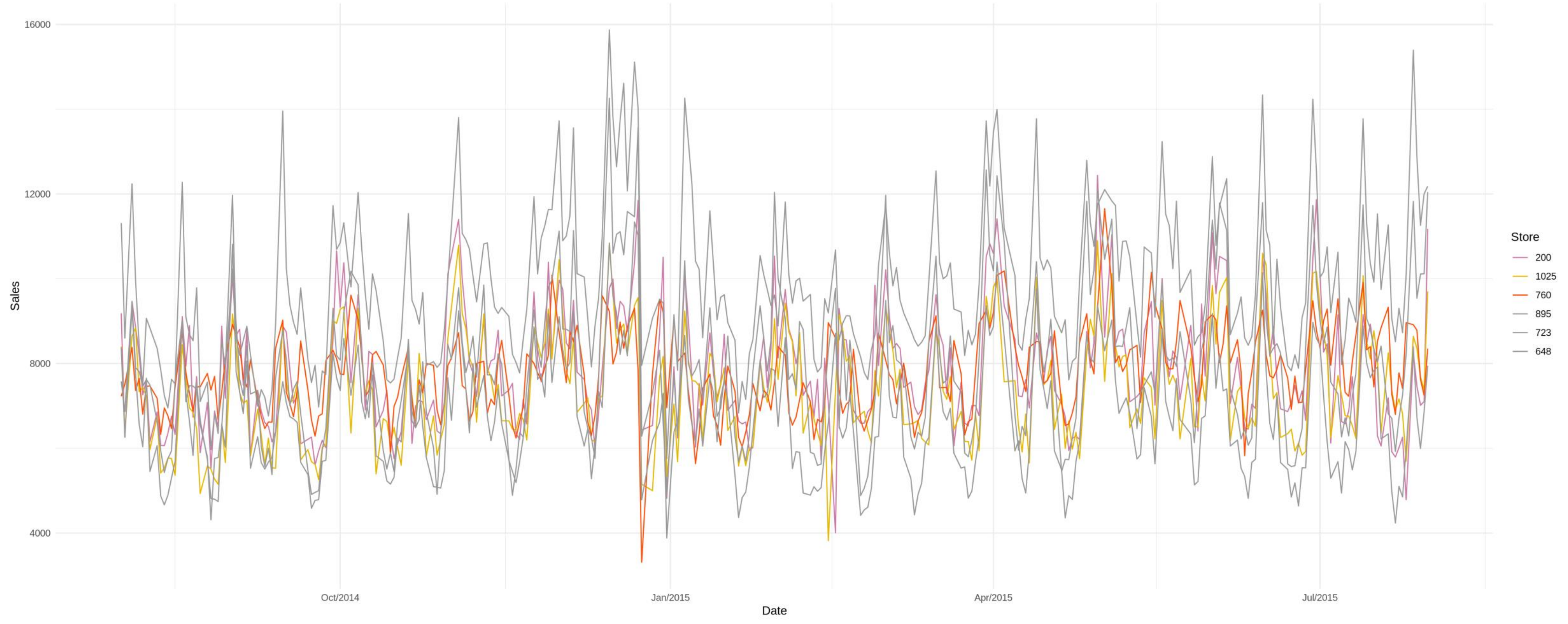
The colored lines are similar stores in sales daily; greyed ones were stores randomly drawn
The entity embedding offers us pretty good distinctions between similar and dissimilar stores



The colored lines are similar stores in sales daily; greyed ones were stores randomly drawn
The entity embedding offers us pretty good distinctions between similar and dissimilar stores

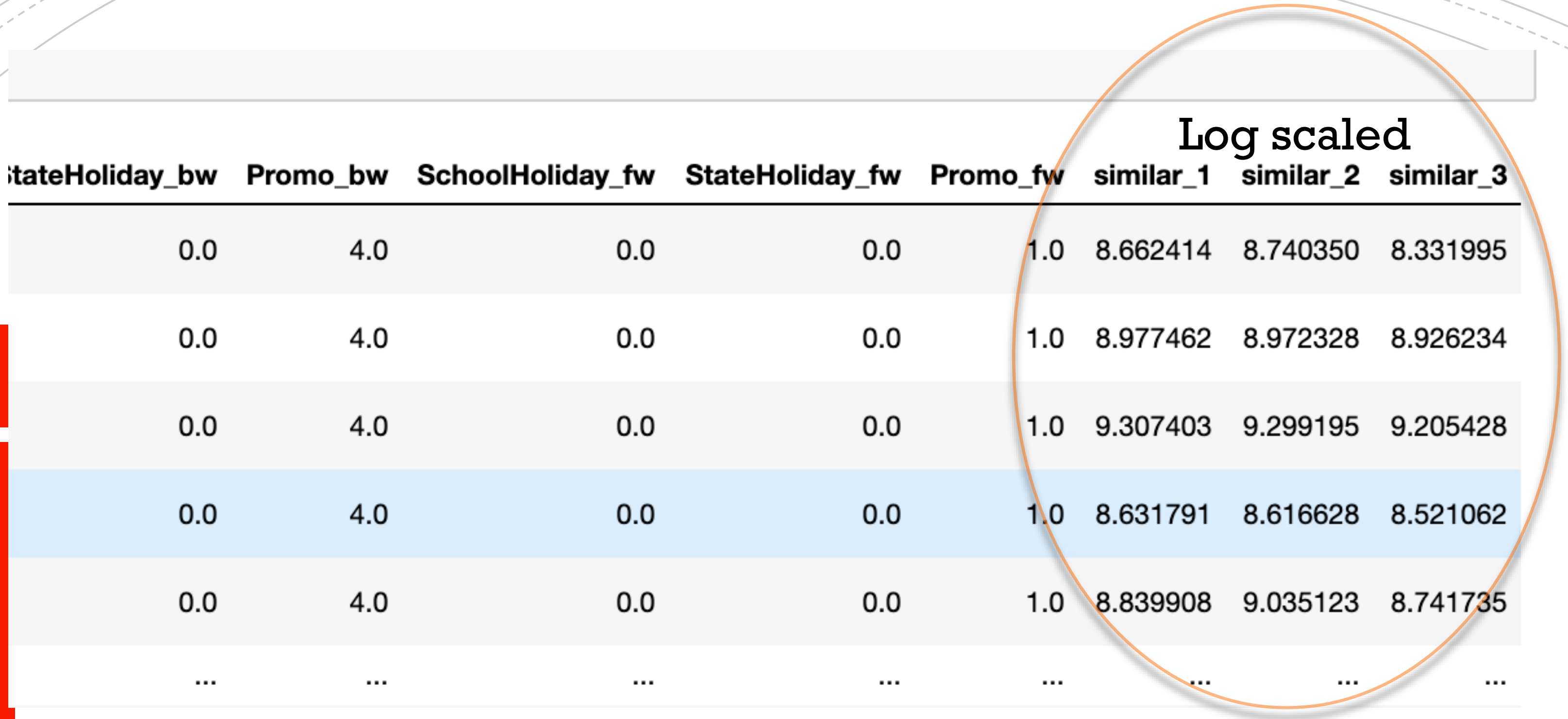


The colored lines are similar stores in sales daily; greyed ones were stores randomly drawn
In raw sales scale, not the log scale as before



The colored lines are similar stores in sales daily; greyed ones were stores randomly drawn
In raw sales scale, not the log scale as before

Can use sales of similar stores to discover abnormal floor execution problems



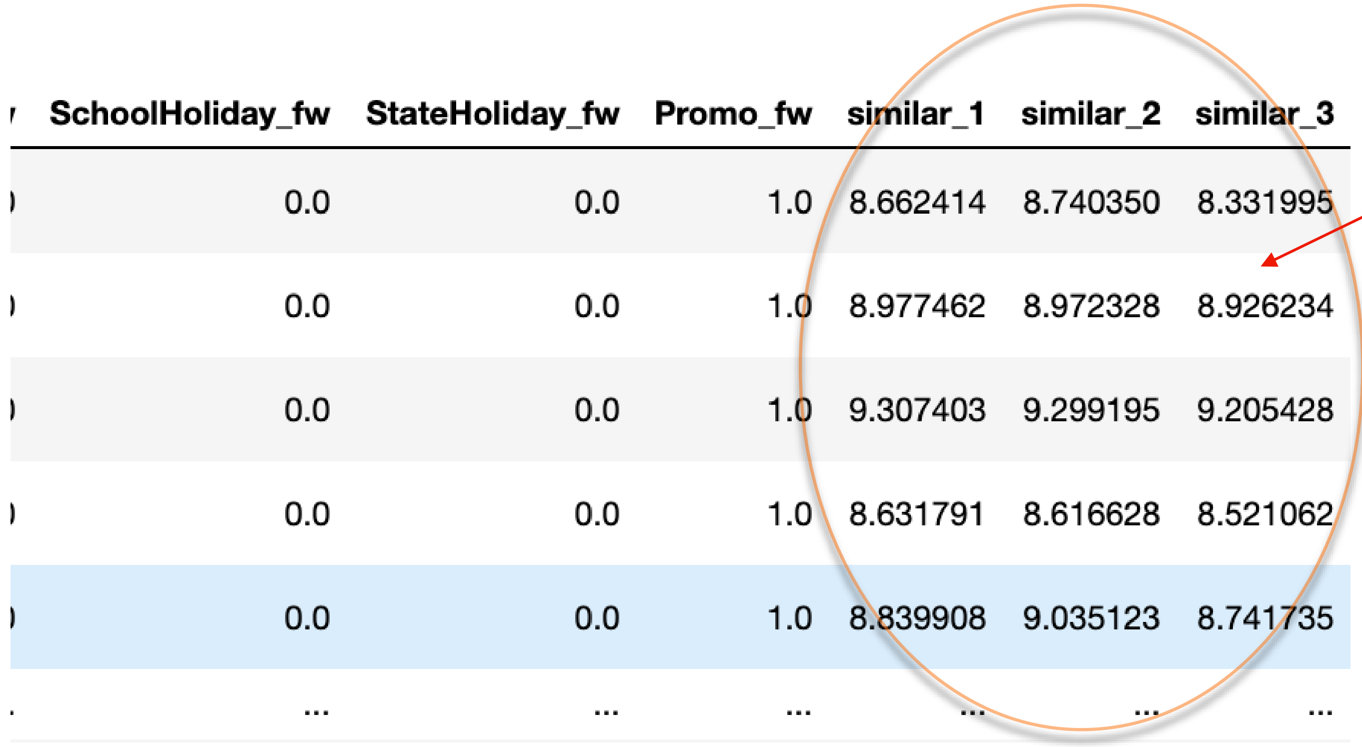
Log scaled

tateHoliday_bw	Promo_bw	SchoolHoliday_fw	StateHoliday_fw	Promo_fw	similar_1	similar_2	similar_3
0.0	4.0	0.0	0.0	1.0	8.662414	8.740350	8.331995
0.0	4.0	0.0	0.0	1.0	8.977462	8.972328	8.926234
0.0	4.0	0.0	0.0	1.0	9.307403	9.299195	9.205428
0.0	4.0	0.0	0.0	1.0	8.631791	8.616628	8.521062
0.0	4.0	0.0	0.0	1.0	8.839908	9.035123	8.741735
...	...	...	...	...	...	...	...

Each row is a store (could be a store and a product), and we could match the sales of similar stores and compare them to discover if there is abnormal floor execution problems (e.g. 40% lower than the average of other 3 similar stores).

This is the solution Cb4 implements I think.

Semi-supervised prediction



SchoolHoliday_fw	StateHoliday_fw	Promo_fw	similar_1	similar_2	similar_3
0.0	0.0	1.0	8.662414	8.740350	8.331995
0.0	0.0	1.0	8.977462	8.972328	8.926234
0.0	0.0	1.0	9.307403	9.299195	9.205428
0.0	0.0	1.0	8.631791	8.616628	8.521062
0.0	0.0	1.0	8.839908	9.035123	8.741735
...	...	...	...	...	...

On the test set, use a predictor A to predict sales for all entries; then match the similar store predictions and append them at the end like this plot shows; train a new predictor B to make predictions

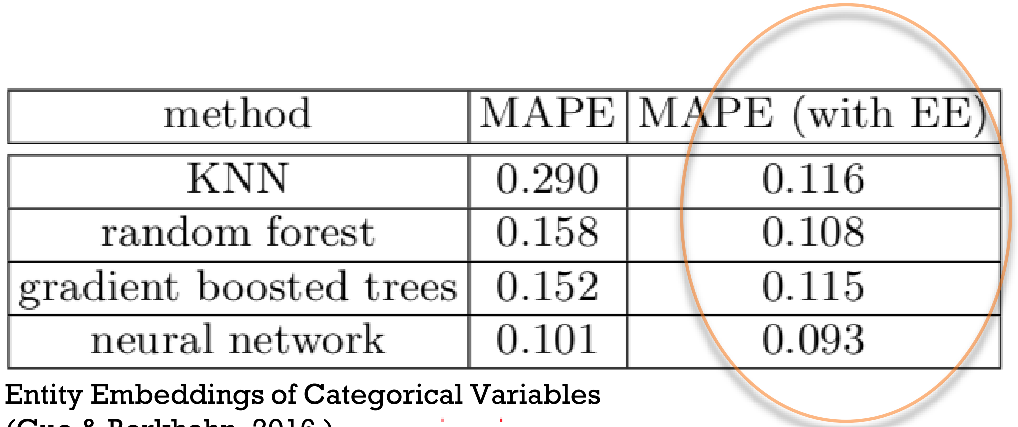
- However, this method backfires and performed a little worse than traditional supervised methods ☹

Optimized trendy ML methods and compared their performances

- All of them are Bayesian optimized. Bayesian optimization proved to be achieving better hyperparamter tuning than grid search or random search.
- These codes could be directly applied for model selection in the future.

	Private leaderboard	**Public leaderboard**
No.1 solution (1/3303)	0.10021	0.08932
Entity embedding NN	0.11886	0.10833
XGBoost	0.20285	0.20914
LightGBM	0.12365	0.11304
CatBoost	0.14772	0.12952
Semi-supervised approach with NN	0.13938	0.12882
Average ensemblign of NN and LightGBM	0.11599	0.10379

Apply entity embeddings learned from NN to boosted tree models



method	MAPE	MAPE (with EE)
KNN	0.290	0.116
random forest	0.158	0.108
gradient boosted trees	0.152	0.115
neural network	0.101	0.093

Entity Embeddings of Categorical Variables
(Guo & Berkhahn, 2016)

The entity embedding paper claimed that when they imported the trained entity embedding (EE) from NN, all of the methods are improved.
However, in my preliminary experiment to port the store EE into a Lightgbm, it backfired on the performance.
Need further investigation.
Will dig into their source code for this implementation

Next steps

- Apply H-E-B dataset on it!! 🙆
- Training separate embeddings from the sales of each product type, or even key product. This way we could have a more accurate monitoring of sales situation product wise.
- For better prediction, try other optimizers (now used Adam) for the NN; Try some meta-ensembling other than average-ensembling.
- Integrate embeddings to current MSI systems.
- Build a pipeline to automate the similar stores MSI; build the dashboard to visualize the anomaly detection via similar store MSI.