Sales Forecaster

Purpose

To optimize the inventory in order to maximize the company's sales but minimize the space, budget required for storage.

Inputs

Version 1.0

- historical sales
- historical weather data (need scraper)
- historical traffic data (need scraper)
- number of bulk buying (if possible)

Version 1.1

- google adwords / search data
- social listening
- stock prices
- GDP
- other financial market indicators

Version 1.2

- whatever you can think of

Outputs

Version 1.0

- sales forecast for each item for the next month (in each branch) 100 items total

Version 1.1

- sales forecast for each item for the next month (in each branch) 1000 items total

Version 1.2

- sales forecast for all items for each branch

Accessory work

- model to parse and clean the data
- Automatic retraining
- Staff training
- Scrapers to obtain data for
    - weather
    - traffic
- visualizing tools
- performance evaluation and endpoint activation
- Logic for reordering eg calculate size of item
- Auto ordering api, verify with human

Solution proposal

Following tasks are required

1. Obtain input data
    1. obtain **encoded** data from the database
    2. scrape data from weather websites for temp, wind speed, rainfall per day for each store location
    3. scrape traffic data from google traffic/ maybe need to pay for it
2. clean the data
    1. data need to be in time series linear divided into years
    2. one hot encode all categorial data (if any)
3. Model creation and training
    1. Main Models consideration
        1. LSTM (deep) network
        2. standard deep nn (with a fixed amount of periods as input data

      2. Models for comparison (non neural net models)
            1. linear regression
            2. SGD regressor
            3. Lasso
            4. elasticnet
            5. RidgeRegressor
            6. SVR(linear kernel)
            7. SVR(rbf)
            8. EnsembleRegressor

4. evaluate hyperparameters and do a first run of hyperparameter optimization
5. create a logic algorithm to convert the output into order suggestion (lambda & apigateway triggers)
6. monitor performance
7. evaluate models every fixed period eg 2 weeks
      1. optimize hyperparameter
      2. retrain model on recent data
      3. explore alternative models eg. RL
      4. revise input parameters
8. Train staff to operate the training and maintenance system

Evaluation metrics

- Relative absolute error and mean absolute error
  - these are suitable for the purpose because we expect a high number of outliers due to the uncontrollability of the human factors

Further improvements

- Clustering to improve generalization and reduce workload
- More visualization (of data/resources/ and the model)

Other Indicators

- Minimize Overstock (stock which is kept for over a month)
- Minimize out of stock item

Benchmark

- Linear regression model (current model)

Model selection for reference (sklearn)