# Sales Prediction by using LSTM

Shidi Yang

# # 1. Exploratory Data Analysis

- Features:
  - WEEK
  - YEAR
  - INVDT
  - MCAT
  - SUBCAT
  - MRP_VALUE
  - NETSALE_VALUE
  - TAX_VALUE
  - PRODUCT
  - SHOP

- Label:
  - SALES_QTY

Based on requirements, our goals use data from Jan 2019 to predict the sales_qty per product per shop for Sep 2019.

In this case, we can take SALES_QTY as a label, and our features are from the rest.

First, I load the dataset into data frame so that i can have a close look. Because it contains a time feature, so I use INVDT as the index, and sort the dataset by this index.

This first thing I may curious about the dataset is, how does that dataset arranged? how does the data been recorded? what's the "primary key" for this dataset? so I group the dataset by using . . . Is it recorded per week per shop per product

# 2. Data cleaning

# 2.1 Find Outliers

- MRP_VALUE, NETSALE_VALUE, TAX_VALUE can't be negative
  - Could it be 0?
- SALE_QTY > 500 are outliers

Per day per product per shop has more than 500 sale_qty are outliers

# 2.2 Taking Care of Missing Data

- df.isnull()
- df.isna()

# 3. Feature engineering

- Splitting the dataset into Training and Test set
- Selecting features
  - Removed WEEK, YEAR features
- Feature Scaling
  - Normalisation
  - $$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$
  - Comparing with Standardisation, Normalisation is recommended when using RNN, especially we are using sigmoid function as an activation function in output layer

- Splitting Features and Labels
- Creating a data structure with 7 time-steps and 1 output
    - Predict the SALE_QTY at t(7) by using the features (MCAT, SUBCAT, MRP_VALUE, TAX VALUE, PRODUCT, SHOP) from the current day + SALE_QTY from t(0)-t(6)

| | | |
|---|---|---|
| t(8) Features | t(1) SALE_QTY | t(8) SALE_QTY |
| t(8) Features | t(2) SALE_QTY | t(8) SALE_QTY |
| t(8) Features | t(3) SALE_QTY | t(8) SALE_QTY |
| t(8) Features | t(4) SALE_QTY | t(8) SALE_QTY |
| t(8) Features | t(5) SALE_QTY | t(8) SALE_QTY |
| t(8) Features | t(6) SALE_QTY | t(8) SALE_QTY |
| t(8) Features | t(7) SALE_QTY | t(8) SALE_QTY |
| | | |

# 4. Forecast modeling

- Stacked LSTM with some dropout regularization to prevent overfitting
- Compiled the network by using Adam
  - Adam is always an safe choice that can update the relevant weights

# 5. Evaluating the Model

- Root Mean Squared Error (RMSE)
- $$RMSE = \sqrt{\overline{(f - o)^2}}$$

# Results

| | Results |
|---|---|
| RMSE on Normalized Data | 0.00121652174852868 |
| RMSE on SALE_QTY per INVDT per SHOP per PRODUCT | 45.3717820284871 |
| RMSE on SALE_QTY per PRODUCT | 334.672403266212 |
| RMSE on SALE_QTY per SUBCAT | 2061.53524983232 |
| RMSE on SALE_QTY per MCAT | 3319.30269544219 |

# Improvements

- One-hot Categorical Data
  - Product, Shop
    - Use One-hot encoder to encode PRODUCT and SHOP columns
    - Then the features will be increased to 521
- Try different parameter
  - time-step: 30, 60
  - neurons
  - optimizer: RMSprop
    - recommended by Keras