Predict Future Sales: "How to win a data science competition"

Anurag Bejju , Savitaa Venkateswaran

Objective

- To build a **predictive** model that estimates the future sales of *1-C company* for the *November* month for every shop and item pair, given 3 years' historical sales between years, *2013* to 2015.

Scope

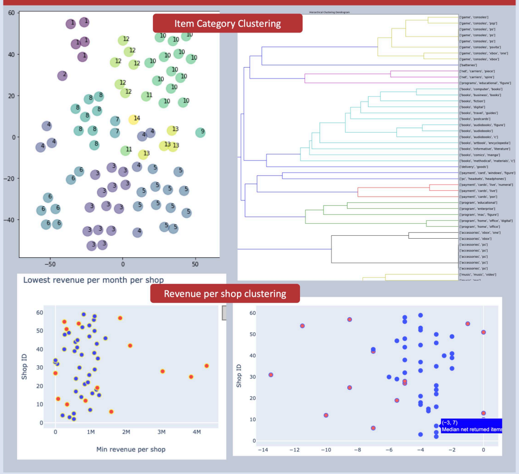
- Perform Exploratory data analysis and data cleaning to gain meaning full insights and design features for our model.
- Explore different naïve and state-of-the-art data mining algorithms to build this predictive model.
- Improve model performance and tune hyper parameters.

State-of-the-Art Models

- **Prophet** [1]: This model developed by *Facebook*, forecasts time series data based on an additive model where non-linear trends are fit with yearly, monthly, and daily seasonality features.
- **Sales Demand Forecast in E-commerce** [2] **:** This Long Short-Term Memory network (LSTM) model developed by researchers at *Walmart*, exploits the non-linear demand relationships available on an E-commerce platform. It uses features like sales demand and sales sparsity to group products.
- **Fast ES-RNN** [3] **:** This model uses *UBER's* ES-RNN (Exponential Smoothing-Recurrent NN) algorithm, a hybrid between classical state space forecasting models and modern RNNs.

Exploratory Data Analysis

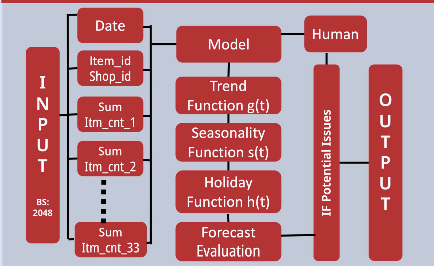
Cluster Analysis



Features

- Shop_id
- item_id
- date_block_num
- item_cnt_month
- Item_cnt_mth_lag_1
- item_category_id
- date_avgitmcnt_lg_1
- delta_revenue_lag_1
- item_shop_last_sale
- item_last_sale
- date_avgitmcnt_lg_2

M1: Prophet

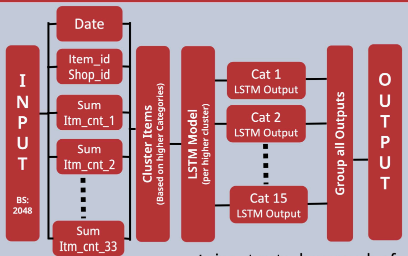


Prophet Equation

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

here g(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes and h(t) represents the effects of holidays. The error term is any idiosyncratic changes which are not accommodated by the model; later we will make the parametric assumption that it is normally distributed.

M2: LSTM

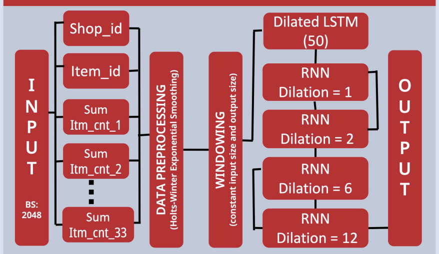


LSTM Equations

- $i = \sigma(x_t U^i + s_{t-1} W^i)$
- $f = \sigma(x_t U^f + s_{t-1} W^f)$
- $o = \sigma(x_t U^o + s_{t-1} W^o)$
- $g = \tanh(x_t U^g + s_{t-1} W^g)$
- $c_t = c_{t-1} \circ f + g \circ i$
- $s_t = \tanh(c_t) \circ o$
- $y = softmax(V s_t)$

i : input gate, how much of the new information will be let through the memory cell. f : forget gate, responsible for information should be thrown away from memory cell. o : output gate, how much of the information will be passed to expose to the next time

M3: Fast ES-RNN



Holts-Winter exponential smoothing:

$$\widehat{y_{t+h}} = l_t b_t^h s_{t-m+h_m}$$

where l is state variable for level, b is a state variable for trend and s is a multiplicative seasonality coefficient. Since the Holt-Winters and the RNN methods were merged to create the final model. $\widehat{y_{t+1...t+h}} = RNN(X_t) * l_t * s_{t+1...t+h}$ Where X_t is a vector of normalized, de-seasonalized features.

Model Evaluation

- Explored 4 models - *XGBoost*, *LSTM*, *Prophet* and *Fast ES-RNN* models in total. XGBoost algorithm with a comprehensive feature-set consisting of 39 attributes gave us the best accuracy. However, Fast ES-RNN algorithm and LSTM showed promising results (close to XGBoost) for this challenge. This seems reasonable considering that deep learning models are data hungry and are more prone to overfitting.

Model	RMSE- VE	RMSE- TE	Ranking
XGBoost (Naïve approach)	0.88	**0.899**	**1**
Fast ES-RNN	0.943	0.971	**2**
Sales Demand Forecast in E-commerce	0.98	1.123	**3**
Prophet	1.37	1.47	**4**

Future Work

- Enhance our feature set for deep learning approaches considered so far by adding item-price based features.
- Perform enhanced hyper-parameter tuning for all four approaches.
- Experiment with our stacked LSTM approach. Train LSTM for clusters based on "sum of item_cnt per month per item per shop" value.

References

1. Prophet: Forecasting at Scale - Sean J. Taylor & Benjamin Letham - Facebook, Menlo Park, California, United States
2. LSTM: Sales Demand Forecast in E-commerce using a LSTM Neural Network MethodologyKasun Bandara, Peibei Shi, Christoph Bergmeir, Hansika Hewamalage, Quoc Tran and Brian Seaman -2019
3. Fast ES-RNN: A GPU Implementation of the ES-RNN Algorithm authored by Andrew Redd, Kaung Khin, Aldo Marini from Carnegie Mellon University. This paper is based on the original work of Slawek Smyl from Uber on hybrid Exponential Smoothing-Recurrent Neural Networks (ES-RNN), which was the winning model of M4 forecasting competition held in the year 2018.