Grocery Sales Prediction

Group 1 - Shiqi Duan, Shuyao Hao, Jordan Leung, Jingkai Li, Peter Li

December 6th, 2017

Agenda

- Introduction
- Data Explanation
- Model Building
- Results

Introduction

Kaggle - popular data science competition website

Grocery Sales Forecasting Competition

- Currently, grocery stores do not have a reliable method of predicting how much of each item to stock
  - Too much - food is wasted, lose money
  - Too little - customers upset, lose potential profit

Data Explanation

Train: 8/16/2014 - 8/15/2016

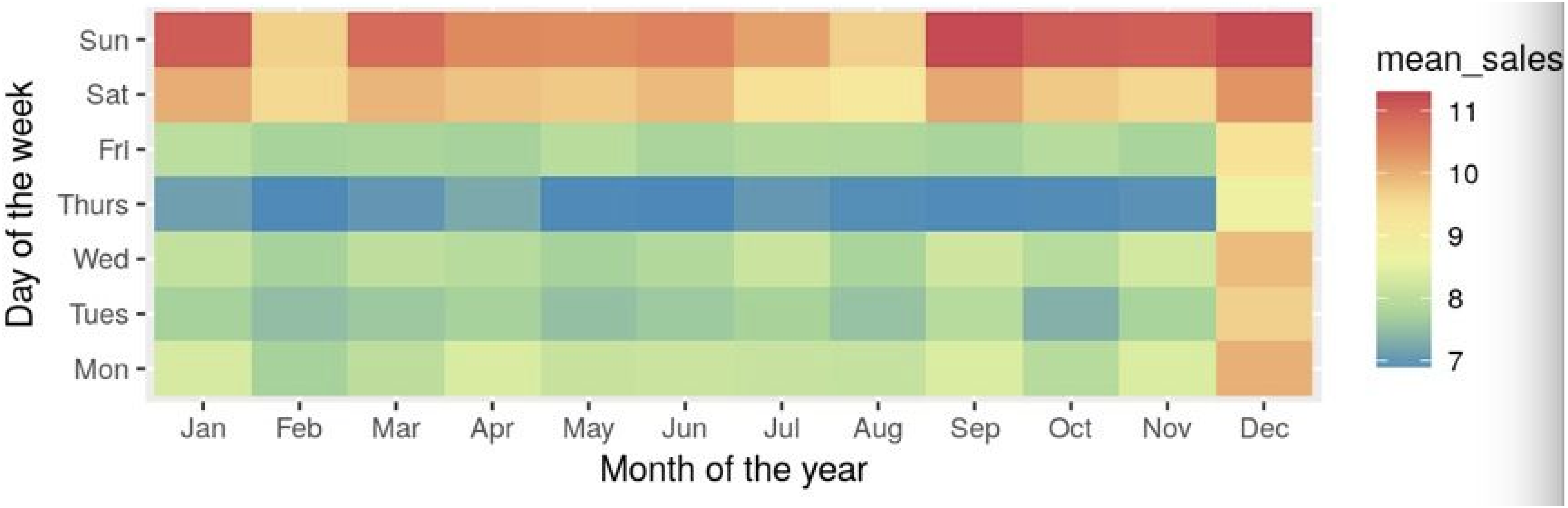
Subtrain: 8/16/2014 - 4/15/2016

Validation: 4/16/2016-8/15/2016

Test: 8/16/2016-8/15/2017

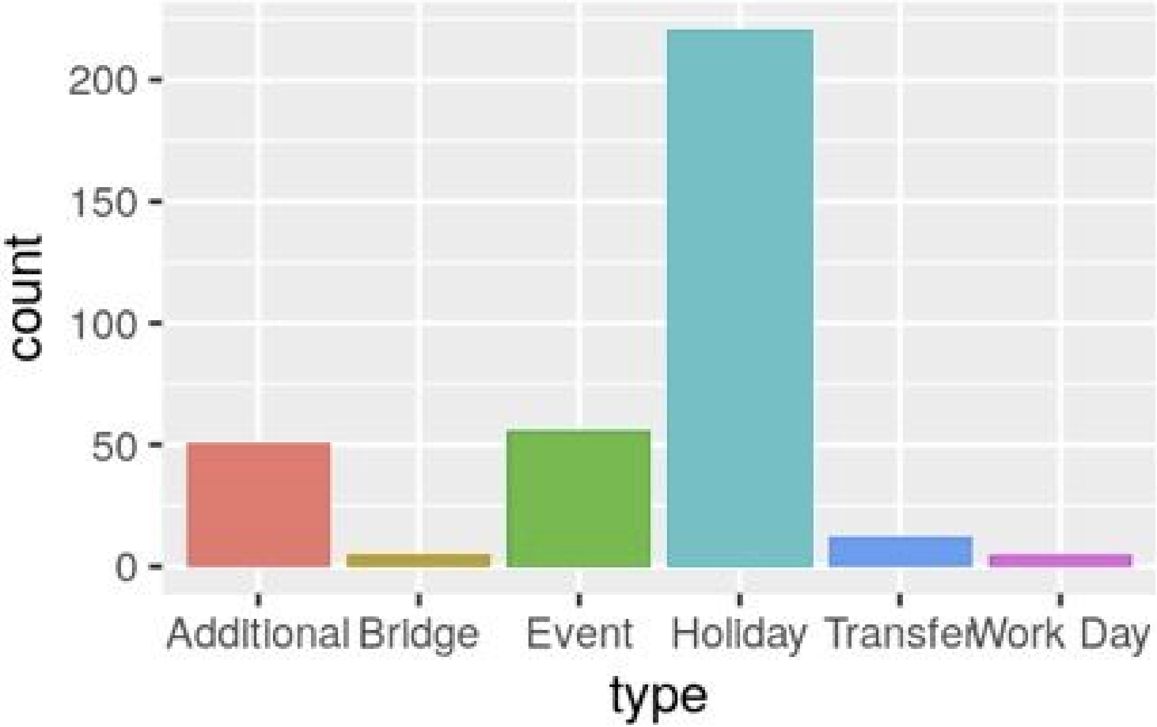
Total have: 10 stores & 200 products

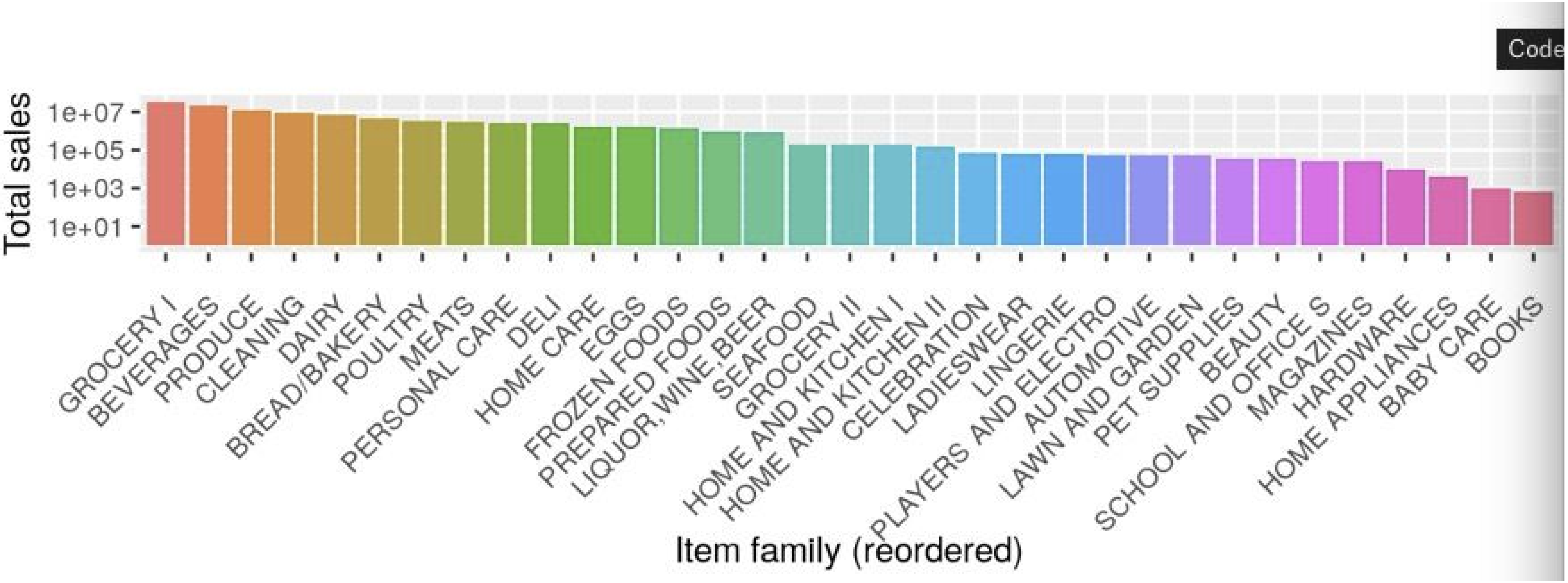
Features Used: Date, Store ID, Oil price, Special Days, Promotion, Item Family, Perishable



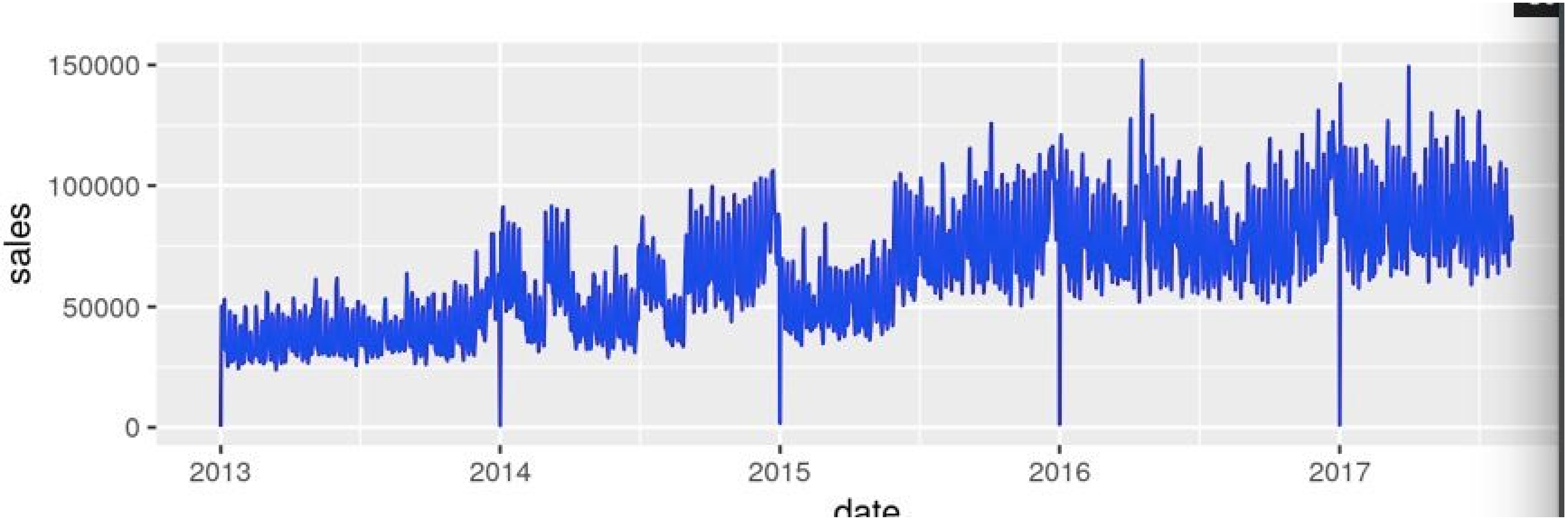
Features : Day, Month

Reason: high sales in the weekends and Dec, while low sales on Thursday


count
200
150
100
50
0
Additional
Bridge
Event
Holiday
Transfer
Work Day
type



Code
Total sales
1e+07
1e+05
1e+03
1e+01
GROCERY I
BEVERAGES
PRODUCE
CLEANING
DAIRY
BREAD/BAKERY
POULTRY
MEATS
PERSONAL CARE
DELI
HOME CARE
EGGS
FROZEN FOODS
PREPARED FOODS
LIQUOR,WINE,BEER
SEAFOOD
GROCERY II
HOME AND KITCHEN I
HOME AND KITCHEN II
CELEBRATION
LADIESWEAR
LINGERIE
PLAYERS AND ELECTRO
AUTOMOTIVE
LAWN AND GARDEN
PET SUPPLIES
BEAUTY
SCHOOL AND OFFICE S
MAGAZINES
HARDWARE
HOME APPLIANCES
BABY CARE
BOOKS
Item family (reordered)



Seasonal trends in sales -> Time Series Model

Evaluation Metric

$$NWRMSLE = \sqrt{\frac{\sum_{i=1}^{n} w_i \left(\ln(\hat{y}_i + 1) - \ln(y_i + 1)\right)^2}{\sum_{i=1}^{n} w_i}}$$

w_i = weight

non-perishable

$w_i = 1$

perishable

$w_i = 1.25$

Method: Stacking Method

Stacking - Ensemble method in machine learning

First layer - train predictive models on original training data

Second layer - train predictive models, using the first layer’s predictions as features

Goal - Use the strengths of each model and minimize the weakness of each model

Stacking - Layer 1

In the first layer, we trained Time Series Models and Machine Learning Models on training set, and predict on validation and test set.

1. ETS (Exponential Smoothing State Space)
2. ARIMA (Autoregressive Integrated Moving Average)
3. Prophet
4. XGBoost
5. Random Forest

Stacking - Layer 1 (Time Series)

ETS

Best Model: ETS with lambda=1.1

Score: 0.65

ARIMA

Best Model: ARIMA (2,2,1)

Score: 0.66

Prophet

Best Model: trained automatically in model.

Score: 0.91

Stacking - Layer 1 (Machine Learning)

XGBoost

Best Model: xbg (η=0.7, γ = 1, max_depth = 8, min_child_weight = 10,

subsample = 0.7, colsample_bytree = 0.7)

Score: 1.18

Random Forest

Best Model: RandomForest (n_trees=800, mtry=11)

Score: 0.71

Stacking - Layer 1 Results + Layer 2 Baseline

Mean average prediction from each of our five models yields a score of 0.6562.

Baseline score for our second layer predictors.

Stacking - Layer 2

In the second layer, we used the validation prediction from the first layer, as the train feature, and trained several Machine Learning Models on it, and achieved the final prediction on test set from the test prediction of the first layer.

1. Linear Regression
2. Random Forest
3. Gradient Boosting
4. XGBoost

Stacking - Layer 2 (Machine Learning)

Linear Regression

Best Model: too complex to list out…

Score: 0.86

Random Forest

Best Model: RandomForest (n_trees=600, mtry=1)

Score: 0.59

Gradient Boosting

Best Model: GBM(“gaussian”, n_trees=500, interaction_depth=2, shrinkage=0.15)

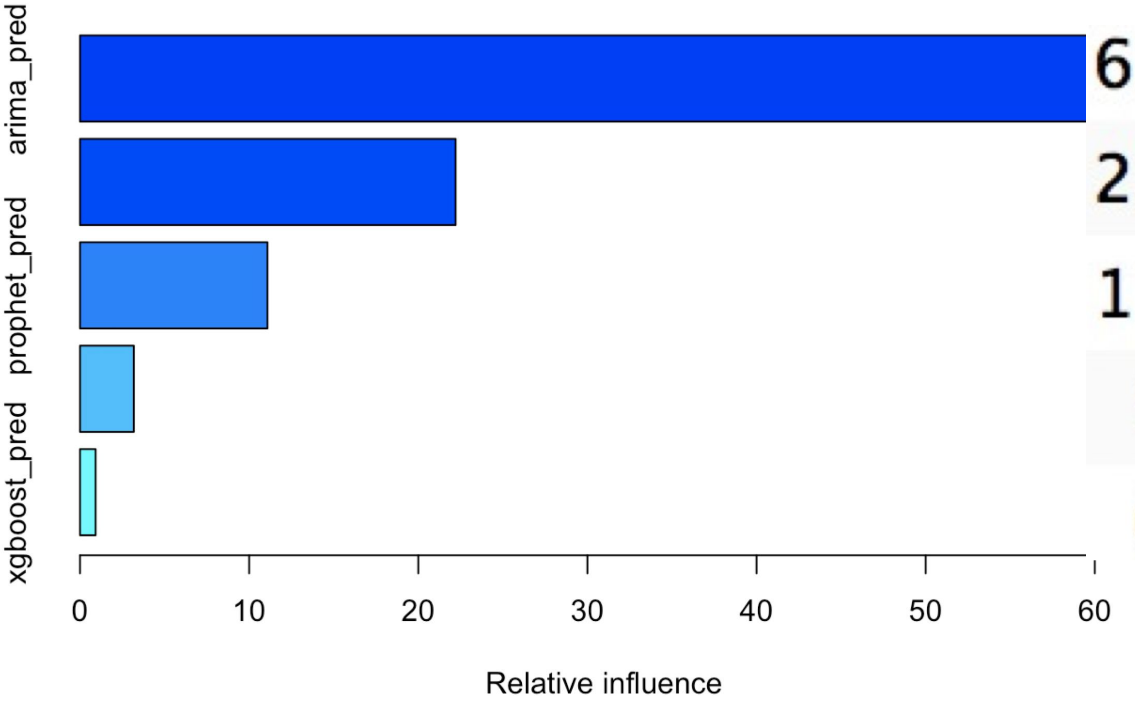
Score: 0.62

XGBoost

Best Model: xgb (η=0.01, γ = 1, max_depth = 8, min_child_weight = 5,
subsample = 0.6, colsample_bytree = 0.6)

Score: 0.59

arima_pred	62.599486
rf_pred	22.210812
prophet_pred	11.084890
ets_pred	3.182055
xgboost_pred	0.922757



Final Prediction + Results

Mean of our 2 best models: Random Forest and XGBoost

Score = 0.58

Best score on kaggle is 0.51, so our model is quite good.

BUT, still needs improvements!

Improvements

- Use more data to train the model with parallel method.
- Improve XgBoost for 1st stack(run more loops).
- Train ARIMA model with longer time slots.
- Adjust the fraction of models for final prediction, instead of using Mean simply.

Final Thoughts

- Kaggle is a great platform to work on real life data science problems
- Learn how other data scientists solve data problems/implement algorithms
- Helpful community with active discussion
- Excellent way to keep practicing and improving data science skills

Thank you!

https://github.com/TZstatsADS/fall2017-project5-group1