WALMART STORE FORECASTING

Atom Group: Jifu Zhao (jzhao59), Jinsheng Wang (jwang278)

Nuclear, Plasma, and Radiological Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801, USA

1. INTRODUCTION

In this project, our goal is to predict the weekly sales for each department in each store for Walmart. The original training data includes the historical data from 2010 - 02 to 2011 - 02. The test data is from 2011 - 03 to 2012 - 10, totally 20 months. In each iteration, we will be given the historical training data and the previous month's data, then we are asked to predict the sales for the next month.

The data set contains multiple features, including Store, Dept, Date, Weekly_Sales, IsHolday. After some initial analysis, we find that there are 81 unique departments and 45 unique stores. In this project, we first explore the given training data set. After some pre-processing methods, we applied three different models to predict the next month's sales. More details will be described in the following sections.

2. PRE-PROCESSING

After exploring the training dataset, the first thing we noticed is that, there are a lot of missing values for some features, and a summary of those missing values is shown in Table.

From Table **??** we can see that, there are 6 features whose missing value counted more than 15% of the total training set. So, our of first step of preprocessing is to drop those 6 features, namely, LotFrontage, Alley, FireplaceQu, PoolQC, Fence and MiscFeature.

After dropping those 6 features, for other features that still have missing values, we do the following processing: for numerical values, we replace the missing values with the median of that feature in total training set, and the same procedure was applied to test set. For those categorical data which has missing values, we add a new level of NA for each categorical feature. Then, for those categorical feature, it is not a good idea to directly transform them into numerical variables. A better idea is to transform them into vectors with dummy variables using one-hot-encoding methods.

After finishing above feature processing, our feature space expands into 287 features in total. With all of these 287 features, we can apply a lot of different models. In this project, we have tried Linear regression, Ridge regression, Lasso regression, xgboost model, random forest model and GBM model. After comparing the performance and running time of each model through cross-validation, we finally choose three models: Simple Linear regression, Lasso regression and Random Forest.

3. R LIBRARY USED

In our R code, we loaded five libraries to simplify our task. 'dummy' library was used to generate dummy variables for features with factor values. 'DAGG' was used to help with cross-validation of Linear Regression model. 'randomForest' was introduced to apply random forest technique. 'glmnet' was added to facilitated Lasso regression. Fi-

nally, 'moments' had the function that can calculate skewness of features to judge whether long tails exist and apply log transformation if necessary.

4. METHODS

4.1. Average from Nearby Weeks

4.2. Time Series Forecasting

4.3. Weighted Average of Model 1 and Model 2

5. CODE DESCRIPTION

All of our code is contained in the file named mymain.R. There are basically three parts in the R file. At the very beginning, the code will automatically check whether or not the required packages/libraries are already installed. In the second part of the code, we do data preprocessing: first read the training and test data sets, then drop the useless variables and process the features to form the new feature space. For the last part, we mainly build our three model: Linear Regression, Lasso Regression and Random Forest model. These built models will make predictions and the results are saved into local file system as required by the project description.

6. RESULTS

To evaluate our model, we choose the metric described on Kaggle:

$$WMAE = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i| \qquad (1)$$

where n is the number of test cases, $\hat{y}_i$ is the predicted sales, y_i is the actual sales and w_i is the weights. For this project, we set $w = 5$ if the week is a holiday and 1 otherwise. The final WMAE for three models are shown in Table 1.

From Table 1, one can find that, Model 1 and Model 2 performs similar to each other. Through weighted averaging, the final model, Model 3, performs better than Model 1 and Model 2.

Table 1. Summary of Models

Model	WMAE
Model 1	2000
Model 2	2000
Model 3	2000

Acknowledgement

The authors would like to thank Xichen Huang for his tutorial notebook on Piazza and David Thaler for his online code.