Viraj Bhalala
CS 598: Practical Statistical Learning

Project 2: Walmart Stores Forecasting

Introduction

In this project, our goal is to predict the sales for Walmart using the data from Kaggle.com. This dataset includes historical weekly sales data of different departments in different stores owned by Walmart. The training data set includes the historical data from Feb 2010 to Feb 2011, and test data set includes data from March 2011 to Oct 2012. This data set contains various information including store number, department number, weekly sales, as is the week contains holidays. After doing an exploratory data analysis, there are total unique 45 stores and 99 departments.

10-Fold Cross Validation Splits

We used 10 Fold cross validation method to evaluate our 3 best models' error. We have used professor's provided code to generate 10 validation splits using our test data. Each of this fold contains 2 months of test data for all stores and departments. Our goal in the following modelling and evaluation parts is to predict sales and obtain its error for each of 2 months in all 10-fold. When training model in each fold, we use all previous data as training data. This includes original train data and all previous folds' data.

Preprocessing

Before training model and predicting for each fold, we preprocessed data using professor's provided code. In general, we first check if start and end date of train and test set is valid. Secondly, we split data by department. Thirdly, we reshape dataframe in such a way that each column contains weekly sales data for the given department. Lastly, we apply SVD to smooth out the seasonal pattern.

Modelling

We have used forecast and forecastHybrid packages to test all of our models. In general, we tested combination of tslm and stlf, tslm, snaive, and auto arima in all 10 folds. Table 1 shows accuracy and run time of each of these different model experiments. We did not do much custom hyper tuning in each of these models.

For the first model, we used tslm for folds 1 to 6 and used stlf for remainder of folds. The reason of switching to stlf model after 6^{th} fold is because stlf can take benefit of training data that has 2 full cycles (2 years) data for remaining of the folds. This is necessary for stlf to carry out seasonal, trend, and error decomposition (STL) from the training data. For tslm model in models 1 and 2, we used season and trend in its formula. Tlsm is a linear model for time series and a wrapper of lm(). It generates season and trend features on the fly and uses them as variables in the model. Our third model is seasonal naïve model. This model is useful as it uses

seasonal pattern that is observed in our training data. The last model uses auto.arima. This automatic model selection technique returns and uses best ARIMA model according information criteria (AICC, AIC, and BIC). It experiment different model combination using different values of p,q, and d within the constraint provided. For this model, we had used its default parameter settings.

Out of these four modelling experiments, we have selected top three best performing models to submit for this project. See the highlighted rows in Table 1. Table 2 shows detailed accuracy of each of 10 folds for these 3 best performing models.

Results

	Model	Accuracy (WMAE)	Runtime for 10 folds
1	tslm and stlf	1571.524	9.19 mins
2	tslm	1611.366	5.73 mins
3	Snaive	1821.882	6.14 mins
4	Auto.arima	2259.804	~2.5 hours

Table 1: Summary of modelling experiments. Runtime is measured using 2015 macbook pro i7 16GB

Fold	Model1(tslm and stlf)	Model 2(tslm)	Model 3(snaive)
1	1945.51659	1945.51659	2196.3555
2	1365.79677	1365.79677	1728.07748
3	1384.04484	1384.04484	1743.2541
4	1535.27843	1535.27843	1651.79973
5	2313.043	2313.043	2380.50244
6	1632.61495	1632.61495	1611.89494
7	1602.38958	1686.12464	2003.02879
8	1360.58791	1399.84215	1667.45603
9	1287.5051	1417.94597	1640.48952
10	1288.45818	1433.45694	1595.96201

Table 2: Accuracy Breakdown for each fold