Project 2: Walmart Store Sales Forecasting

XXX

netID

Objective:

The training dataset provides historical data for Walmart weekly sales per department per store starting from Feb-2010 till Oct-2012 and whether the week was a holiday or not, which may impact the amount of sale .The aim of the project is to create an efficient and an accurate model that can predicts the weekly sale price based on this historical data and splitting the train/test data to 10 folds and an average weighted mean absolute error (MWAE) < 1610.

Technical Details:

Data Preparation:

1- New Features:

2 new concluded features were introduced based on the existing feature 'Date', to be used instead in the performed regression.

- **Wk:** Represents the week number of the weekly sales record.
- **Yr:** Represents the year number of the weekly sales record.

2- Weeks Alignment:

2010 weeks were noticed that they are shifted with respect to 2011/2012. Accordingly, to unify the week number naming convention, a condition was introduced to decrease 2010-week numbers by 1.

3- SVD:

Sales pattern across departments of different stores seems to be similar. Accordingly, SVD was used to decrease noise, by returning top d=7 components of the training data which provide a smoother version of the original training data.

4- Weekly Data as Categorical Feature:

Weekly historical data was transformed into 52 factor categorical features, with levels 0 or 1, to be used as input to the linear regression model.

Model Building:

1- Response Transformation

Two types of response transformations were tested, and both showed an improved performance of the used linear regression model. However, the 2nd approach provided a better average WAE.

- **Cubic Root (**Average WAE for the 10 train/test folds is **1602.301)**

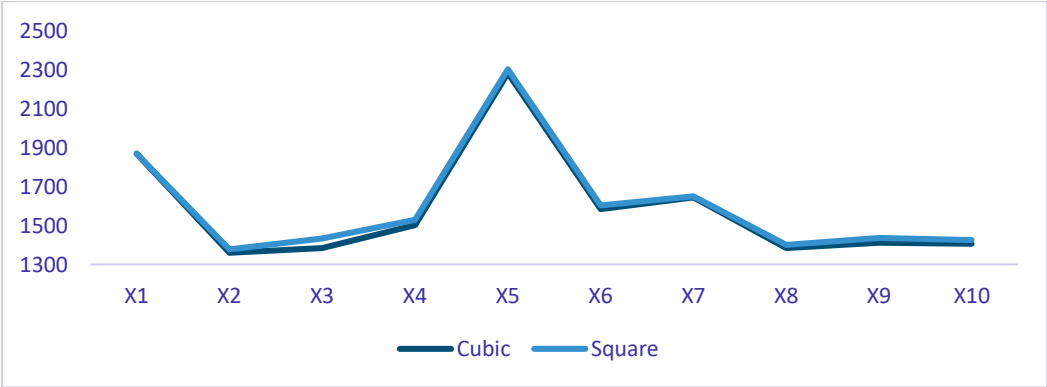
$$\sqrt[3]{y}$$

- **Square Root with Sign (**Average WAE for the 10 train/test folds is **1582.697)**

$$sign(y) * \sqrt{y}$$

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
Cubic	1868.737	1359.944	1384.55	1501.699	2281.648	1584.555	1643.559	1384.056	1411.091	1407.132
Square	1867.115	1377.596	1434.44	1529.229	2301.366	1602.985	1648.985	1401.068	1435.728	1424.497

As per below figure, both transformations almost have the same effect except in 3rd fold, square transformation is much noticeably better.



2- Processing Time Optimization

To optimize the processing time, only pairs of stores and departments which are available in both training and test data are included while performing linear model training.

3- Looping Per Department Per Store

Finally, each department/store split of training data was used to fit a linear regression model with an input of 52 Week categorical features and Year as numerical feature.

Prediction was then performed for this specific department/store and consolidated at the end of the loop for all departments/stores.

Initial Trials:

The following trials were performed with lower WAE.

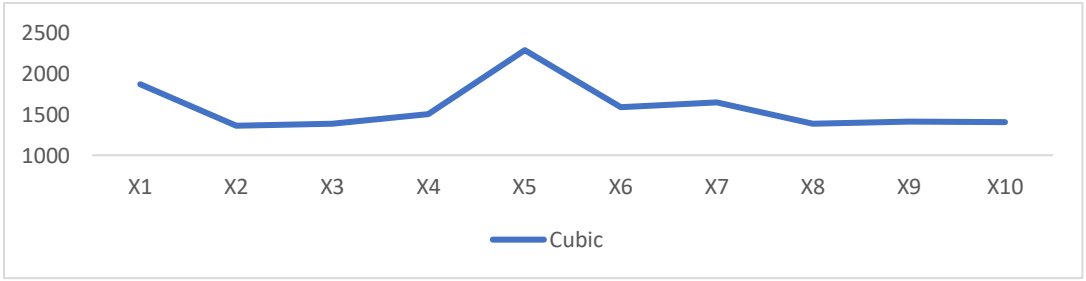
- Utilizing last year data directly to predict the current year sales value.
- Averaging the previous, current, and following week from the previous year.
- Direct linear regression without performing SVD.

Prediction Accuracy:

Based on below table the average WAE for the 10 train/test folds is **1582.697**, which is below the targeted WAE by 27.303.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
WAE	1868.737	1359.944	1384.55	1501.699	2281.648	1584.555	1643.559	1384.056	1411.091	1407.132

As per below figure, dataset 5 appears to have the highest WAE. That is because it has the largest number of holiday records, and as per the evaluation parameter is giving a 5 times higher weight to incorrect holiday week predictions.



Processing Time (10 Folds):

Using a laptop with the following configurations, training and testing the 10 folds took approximately **1.24** minutes.

Lenovo Ideapad Flex 5, 2.80GHz, 16GB Memory

```
> time.taken
Time difference of 1.237141 mins
```

Discussion:

The project showed how SVD helped in smoothing the data and improving the model performance, and that not all the data components are useful in building models and making prediction.

Also, how limiting data to only useful records and use of lm.fit instead of lm (which is the basic computing engines called by lm, where there is no formula notation) helped a lot in improving the processing time.

Finally, it showed how useful response transformation in improving the model performance.

----------End of Report----------