# Walmart: Sales Forecasting

**Ji Zhao**
CSCI 5502
104319968
Ji.Zhao@colorado.edu

**Tianli Wu**
CSCI 5502
109203073
Tianli.Wu@colorado.edu

**Xuan Gao**
CSCI 5502
109565425
Xuan.Gao-1@colorado.edu

## ABSTRACT

The purpose of our project is to predict sales to support business decision making. The result could give the sales team time to change marketing strategies. We intend to explore which methods predict the most accurate sales for each store and each department. In order to better understand our dataset, we plot figures to observe the relationship between attributes. These figures help us a lot for selecting features to predict sales. We also tried to use the frequent patterns from the Apriori algorithm to discover the associations between different attributes and weekly sales. We only got a few useful pieces of information. Also, we decided to use Support Vector Machine, Neural Network and Random Forest among all regression methods we learned from the class to forecast weekly sales. Finally, we compared and evaluated the results of the selected methods.

We conclude that Random Forest has the best performance among the selected methods. Significantly, the Random Forest has an outstanding performance on unnormalized data. In comparison, it is hard to say which approach is better between the Support Vector Machine and Neural Network. Their performances both improve on using normalized input data. But, the performance is different on the input data selected. The Support Vector Machine's result is slightly better than the Neural Network's when we only trained one store. However, the Neural Network has a better result as we train all the store data.

**Keywords**: Apriori, Regression, SVM, Random Forest, Neural Network

## INTRODUCTION

Obviously, online data has increased dramatically in recent years. Extracting knowledge from a large amount of data is becoming increasingly important, especially in making business decisions. In this area, there are plenty of opportunities waiting to be explored. The common interest in data science for business gathers us together. We found that the Store Sales Forecasting task in Kaggle is attractive. Besides, many companies have applied data mining technology to help them make business decisions and easily find abundant reports. For example, Walmart used data mining technology to anticipate particular demands for products and stock them in advance ahead of hurricane Frances's handfall[9]. Therefore, we would like to apply data-driven techniques that we learned from the Data Mining Course on these data in our project. Specifically, use the apriori algorithm to find both frequent patterns and association rules. And we applied Support Vector Machine(SVM), Neural Network(NN) and Random Forest(RF) on our data to explore the accuracy of prediction of weekly sales. Using these ways, we hope we could discover useful information from our data. To sum up, that's the reason why we want to dive into this project.

The challenge of the task is that we have a lot of features. We need to spend a lot of time to prepossess. For attributes markdown 1-5, more than half values are missing. We need to handle these missing values in a proper way. In addition, it is time-consuming to select features among all features. Besides, we have learned a lot of methods from the class. We know the hash tree, partition, sampling,FP-growth and vertical data format for mining the frequent pattern. For prediction, we learn the Decision tree induction, Bayesian belief networks, Backpropagation, Support vector machines, Boosting, Bootstrap aggregating (bagging) and so on. However, We can not try all the methods. Therefore, we need to spend on research. Last but not least, the dataset just contains limited data. We only have two years of data and 45 stores. That may lead to an unexpectedly lousy result.

## RELATED WORK

Sales prediction is a extremely popular and useful since it can offer vital information to support decision making, especially in today's environment. There are so many techniques to forecast sales. Meulstee and Pechenizkiy [14] used an ensemble approach that consisted of the decision tree, SVM, logistic regression, and so on. Long short-term memory(LSTM) in Deep learning was always used to predict sales[12], and it has a significant achievement on regression[18]. In addition, this paper[2] explored the performances of deep neural networks, SVM, Random Forest and other shallow methods. Compared with Random Forest, Neural Network does not work remarkable well. Besides, Random Forest has the best performance among the SVM, Randdom Forest and ANN[2].

Since the task is Kaggle competitions, many teams have finished and obtained great results that provide us lots of great examples and methods. From [11], Kirill's notebook shows using Random Forest and Recurrent Neural Network predict the weekly sales and Random Forest has better accurate results than the Recurrent Neural Network.Besides,these papers[17][3] both apply the apriori algorithm to find frequent patterns and association rules. They improved the apriori algorithm by immediately removing the item from the candidate list when it does not meet support or confidence conditions.

Not the same as the papers mentioned above, we didn't try to improve the Apriori algorithm. We simply applied it with data transformation to mining the frequent patterns and association rules. Then We used three methods to forecast the weekly sales that are Random Forest, SVM and Neural Network. Being different from the first two methods, Neural Network, specifically we used a fully-connected three layers Neural Network, is less mentioned in papers. Apart from comparing these three methods, we focused on the various sets of conditions or parameters. For example, we prepossess our dataset with and without normalization then compare their performance. Another example is that we tried the grid search to tune the parameters of SVM and Random Forest. By comparing the result, we adjusted the conditions and parameters to get a better result.

## DATASET

### Data Introduction

We used a dataset from Kaggle. The dataset is the historical sales data for 45 Walmart stores located in different regions for two years. Each store contains several departments. Some departments' data is missing which makes our task more challenging. Besides, Walmart runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. Other than listed holidays, we also have other national holidays. But our dataset does not label these holidays. This dataset(Figure 1) has four parts listed below.

- Stores: This file contains anonymized information about the 45 stores, indicating the type and size of the store.
- Features: This file contains additional data related to the store, department, and regional activity for the given dates.
- Train: This is the historical training data, which covers from 2010-02-05 to 2012-11-01.
- Test: This file is identical to train.csv, and it covers from 2012-11-02 to 2013-07-26, except we have withheld the weekly sales.

### Data visualization

Data visualization can help us learn the dataset and also make us choose features better. We draw many kinds of graphs, like histograms, box plots, scatter plots, heat maps, and pair plots. Pair plot describes the relationship between any two attributes. The heat map indicates the correlation between these attributes. Figure 2 is the heat map of our dataset and Figure 3 is the pair plots between all our features. These figures can give us some useful information. For example, we can see store size and weekly sales are positively correlated in the heat map, so we chose store size as an attribute to forecast the weekly sales. For the pair plot, we can find a relationship between the date and weekly sales. From the figures, we find some results and listed below.

1. when it is around the end of the year, like November and December, it has a probability that the weekly sales obviously higher than other months which is a significant repeat pattern.

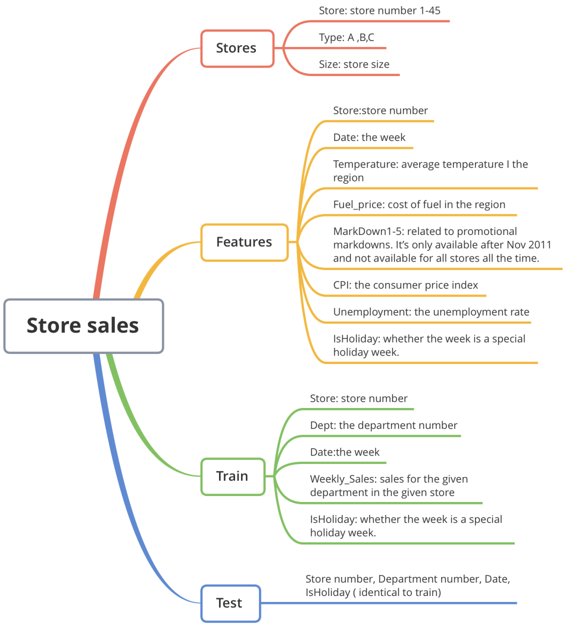
Store sales
Stores
Store: store number 1-45
Type: A ,B,C
Size: store size
Features
Store:store number
Date: the week
Temperature: average temperature I the region
Fuel_price: cost of fuel in the region
MarkDown1-5: related to promotional markdowns. It's only available after Nov 2011 and not available for all stores all the time.
CPI: the consumer price index
Unemployment: the unemployment rate
IsHoliday: whether the week is a special holiday week.
Train
Store: store number
Dept: the department number
Date:the week
Weekly_Sales: sales for the given department in the given store
IsHoliday: whether the week is a special holiday week.
Test
Store number, Department number, Date, IsHoliday ( identical to train)

**Figure 1. Dataset1:Store sales details**

2. when the unemployment rate is greater than 10, the maximum of weekly sales is lower than or equal to the median value.
3. when store size is larger, the weekly sales have a high probability of getting higher.
4. when the number of departments increases, the weekly sales also have a high probability of being larger.

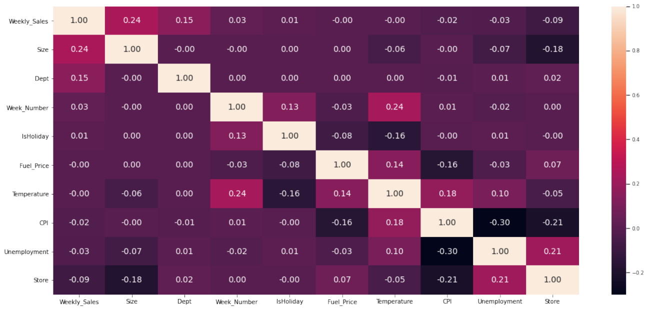

**Figure 2. Pair plot between attributes**

### Preprocessing Data

- Data integration: Data integration is used to combines data from multiple sources. Since for every source, we both have various parts of data, we used entity identification to combine different sources of data. And this step is applied to each model in our project.
- Data cleaning: Data cleaning is used to handle imperfect data, like missing data. In our project, we have lots of missing values in the columns of Markdown 1 to Markdown

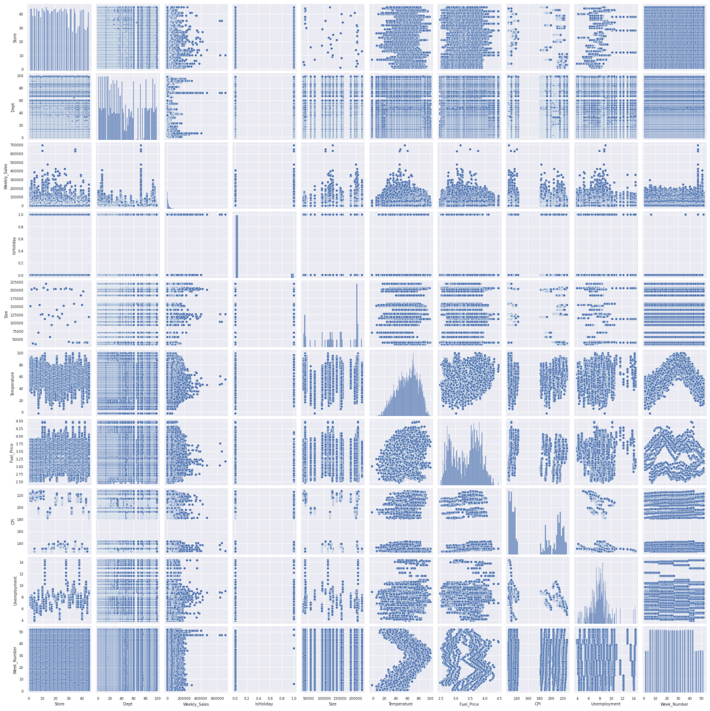

**Figure 3. Pair plot between attributes**

5. We compare two ways to deal with missing values: one is to fill 0 in these values and the other is to delete all markdown columns. Our models have different results in these ways.

- Data reduction: Because our datasets are massive, using data reduction can decrease the mining time, especially for those attributes that just have a bit of influence on the predictions. We applied data reduction into our models: We dropped different columns for Random Forest, Neural Network and SVM.

- Data transformation: Since most of the columns in the dataset have such different ranges, which may affect the accuracy of prediction, we used data normalization for Neural Network, Random Forest and SVM methods. Plus, we compared the result with the normalization and without the normalization. To explore the influence of date, we constructed a new attribute week number by the attribute date. Besides, we used data discretization for the Apriori algorithm: we converted continuous range into intervals, like equal width and equal frequency.

After data preprocessing, we have about 420,000 rows of data. Because we want to predict future sales, we split the train-test data according to the date—thirty months' data for training and last three months' data for the test.

## METHODOLOGY

Our project tried the Apriori algorithm to mine the frequent pattern and association between features and weekly sales. We wanted to find the relationship between each feature and weekly sales. Then we attempted to use three kinds of models are Neural Network, Support vector machine and random forest to predict the weekly sales.

### Apriori

We used the Apriori method from the Apyori package. The Apriori is usually to discover frequent patterns in categorical data. The majority of the type of data is numerical. Since the Apriori method only takes String or Boolean type for input, we need to convert numerical data into discrete string data. Before we apply the Apriori, we need to prepossess our data to use the Apriori method. To prepossess data, we use two ways to classify numerical data to the string. One is the equal width. The other is equal frequency.

For equal width, we simply spilt data into equal range and replace it with the mean of the range. The result is for support rate at 0.1 and confidence rate at 0.3. We found some impressive results. The small store size has a positive relationship with low weekly sales. The low CPI has a positive relationship with high weekly sales and the high CPI has a positive relationship with low weekly sales.

For equal frequency, we used distribution to split the numerical data. Same like equal width, we use a specific number to represent each value. We also find some impressive results.The smaller the store size, the fewer weekly sales. No holiday weeks leads to less weekly sales. And extremely high or extremely low temperature leads to less weekly sales.

The last thing we need to decide is whether to use the Apriori algorithm or not. At first, we were going to calculate the association between each feature and weekly sales. And use ratios to predict the weekly sales. However, we do not get much information from Apriori. And most of the lift ratio is close to 1, which means these two attributes are independent. From above, the conclusion is Apriori algorithm is not a useful tool for feature selection.

### Models

Since there are so many models that can make the prediction, it took us a lot of time to select the appropriate models. After researching different models and comparing those models, we decided to use Random Forest, SVM and Neural Network to train our dataset. We explored various parameters or conditions for each model mentioned above and tried to figure out the best result. We measured these methods' performance by accuracy, which is computed by MSE. We explored the different effects between normalized data and un-normalized data, affecting the accuracy in three ways. Also, we explored the difference between one store data and all store data affecting the accuracy of selected methods.

### 1. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression, and other tasks. Random forests consist of tree predictors and each tree relies on the values of a random vector sampled independently.[4]

So why we use random forests to complete the regression prediction? According to our research, it can deal with binary

| Attributes | Temperature | CPI | Fuel-Price | IsHoliday | Type |
|---|---|---|---|---|---|
| MSE | $5.34 \times 10^8$ | $5.30 \times 10^8$ | $5.20 \times 10^8$ | $5.15 \times 10^8$ | $4.97 \times 10^8$ |
| Attributes | Week Number | Unemployment | Size | Store | Department |
| MSE | $5.13 \times 10^8$ | $4.93 \times 10^8$ | $4.68 \times 10^8$ | $4.68 \times 10^8$ | $2.39 \times 10^8$ |

**Table 1. MSE of Random Forest for using each feature to predict weekly sales**

| Experiment setting | MSE |
|---|---|
| With all attributes | $1.13 \times 10^8$ |
| Without Temperature,CPIFuel Price | $9.79 \times 10^7$ |
| Without Type | $9.78 \times 10^7$ |
| Without Type, Store | $8.59 \times 10^7$ |
| Without Type, Store,Week-number | $1.23 \times 10^8$ |
| With type, Store using one-hot encoding | $2.79 \times 10^8$ |
| Week-number using one-hot encoding and drop type and store | $2.69 \times 10^8$ |
| Without Type,Temperature,CPI,Fuel-Price,IsHoliday and Unemployment | $3.98 \times 10^7$ |
| Without Type ,Temperature,CPI,Fuel Price ,IsHoliday ,Unemployment and test-train split( test time started on 2012-08-03) | $1.43 \times 10^7$ |

**Table 2. MSE of Random Forest for Different Features**

features, categorical features, and numerical features. For outliers and noise, random forest has a robust performance.

For error estimations, it provides useful internal estimations and strength, correlation and variable importance.[4] After analyzing our dataset, we find that our dataset has some characteristics:

- There are many kinds of attributes in our dataset: categorical, numeric, binary and ordinal attributes.
- Some numeric attributes have an extensive range of values or outliers
- Some attributes have the correlation relationship.

Based on the above analysis, we chose Random Forest as one method to make the weekly sales prediction.

Since the missing proportion of Markdown 1 to Markdown 5 is about 60%, we dropped all Markdown columns. Before selecting the appropriate features, we tried to use normalization in Random Forest. However, we got a worse performance than without normalization. To determine the features that can give a better-predicted performance, we first selected the attributes one by one to predict weekly sales. And the result is shown in Table 1. Second, we used different combinations of attributes and attributes with lower MSE in the last step would have a higher possibility of being chosen. We also tried to encode some attributes, like using the one-hot vector to indicate every store and week number. Besides these steps, we attempted to use data split by date, like training data using the first 30 months data while remaining data are used to test The detailed result of these experiments is shown in Table 2.

## 2. Support Vector Machine

SVM is a method for the classification of both linear and nonlinear data. It uses a nonlinear mapping to transform the original training data into a higher dimension, then it searches the optimal hyperplane to separate the classes. [8] SVM operates on the kernel matrix. It used a kernel matrix to describe the similarity between the samples. Figure 4 below shows the data being linearly separated. [8]

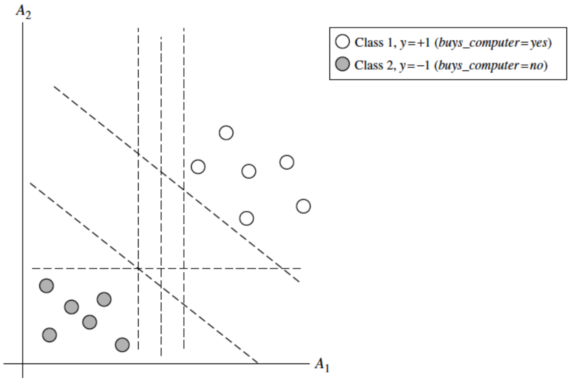
$A_2$
○ Class 1, $y=+1$ (*buys_computer=yes*)
● Class 2, $y=-1$ (*buys_computer=no*)
$A_1$

**Figure 4. 2-D training data are linearly separable [8]**

In the paper [15], we found that SVM performs well when compared to other algorithms. SVM works on a balance between model accuracy in the training process and model ability inforecasting outputs, which optimizes the SVM in predicting projects [7]. Due to the proficient ability in non-linear data [6], SVM has been widely used for regression problems [5].

The data we used is non-linear. We want to use regression to predict future weekly sales while SVM can approximate any function or decision boundary very well with enough training data; therefore, We decided to use it as one of the algorithms to analyze the data.

For the SVM model, firstly, we selected the most useful features.

Then we compared the performance of normalized and unnormalized data. Since the quality of SVM models depends on a proper setting of SVM hyper-parameters, the main issue for practitioners trying to apply SVM regression is setting these parameter values (to ensure good generalization performance) for a given data set[6]. In our project, we used the grid search to tune the parameters of SVM.

### 3. Neural Network

The reason why we chose Neural Network is we have a massive amount of data. We have 45 stores of data for around two years. There are more than 4 million lines of data. We could use a Neural Network to predict weekly sales by using these enough amounts of data.

In addition, Neural Network is popular nowadays and has a significant achievement in both classification and regression. We can find a lot of works of exploration Neural Network on series data. A similar task which is to predict house price used Long Short-Term Memory[18]. Kevin uses the Recurrent Neural Network to forecast time series data[13]. Usually, Recurrence Neural Network and Long Short-Term Memory only take previous weekly sales as input and predict future weekly sales. Kirill notebook shows the performance of weekly sales prediction by Recurrent Neural Network[11]. However, the result of Recurrent Neural Network is much worse than Random Forest[11]. We think that, except for previous weekly sales, we also need to consider other features, such as store size, fuel price, temperature and so on. From above, we decided that we applied a Fully Connected Neural Network for our task.

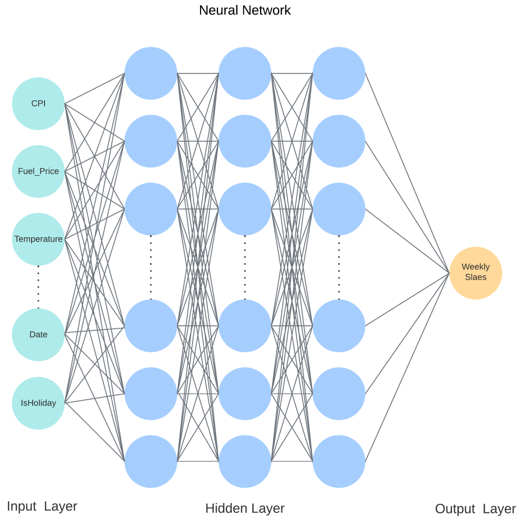

**Figure 5. Structure of Neural Network**

Since the value of attributes has significantly different ranges. For example, the unemployment rate and fuel Price are low, but store size and weekly sales are enormous. Without normalization, some attributes which have a larger scale would add more weights compared with others. Therefore, normalization is an essential step in the prepossessing of Neural Network.

We built a three fully connected layer Neural Network. Look at figure 5, each layer has the same number of hidden units, 256. We set the batch size to 10. And set epoch to 10 as well. All the hyper-parameters setting are based on [2]. We took normalized data as our inputs. And the output layer is the prediction.

We split data by date '2012-08-03'. All the data before that date sets to train data, and all the data after that data sets to test data. We tried to train not only all features but also some specific features. We did not see a large difference between them.

## EVALUATION

The project is a regression problem. Thus, we used the mean squared error to evaluate the result.

Firstly, we separated a hold-out set from the train data as a test set. In the experiment, we divided data by time. The training data is the data before '2012-08-03', the rest is the test data. After splitting the train and test data, the number of train and test data objects is 383040 and 38530, respectively.

Secondly, our dataset is from the Kaggle competition, so the test data is provided. We uploaded the predicted data to the Kaggle to tune the parameters and evaluate the result.

### 1 Metrics

Mean Squared Error (MSE) shows how close a regression line is to the points you tested. [1] It calculates the distance between the ground truth value and the predicted value and squares the result.

MSE equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

$Y_i$ is the ground truth value. $\hat{Y}_i$ is the corresponding predicted value.

### 2 Evaluation Setup

We predicted the data of store1 and compared the predicted result with the actual value. Based on the value, we tuned the parameters. Repeated the process until we got the smallest mse value.

In addition, we uploaded the result to the Kaggle website to evaluate the result.

#### *2.1 Feature Selection*

There are 16 features in the dataset, but not all of them positively affect the result. Thus, firstly, we used forward-stepwise to select the most effective features. It starts with an empty model, then adds the feature that has the smallest MSE result. Rinses the feature and repeats the process until it obtains the best result. The final subset includes the most effective features.

The feature selection process was applied to all the algorithms - Random Forest, SVM, and Neural Network - and obtained

| Algorithms | The Smallest MSE | | | |
|---|---|---|---|---|
| | Unnormalized Data | | Normalized Data | |
| | Store1 Data | All Data | Store1 Data | All Data |
| Random Forest | $1.41 \times 10^7$ | $1.41 \times 10^7$ | $1.01 \times 10^8$ | $3.75 \times 10^8$ |
| SVM | $9.44 \times 10^8$ | $5.19 \times 10^8$ | $2.51 \times 10^8$ | $3.66 \times 10^8$ |
| Neural Network | $2.14 \times 10^9$ | $1.73 \times 10^9$ | $6.31 \times 10^8$ | $2.87 \times 10^8$ |

**Table 3. The smallest MSE for All algorithms**

| Drop Attributes | Test Loss |
|---|---|
| Markdown 1-5, Type, Temperature, CPI, Fuel Price | $1.26 \times 10^{-2}$ |
| Markdown 1-5 | $1.24 \times 10^{-2}$ |
| Type | $9.66 \times 10^{-3}$ |
| None | $8.74 \times 10^{-3}$ |

**Table 4. The Test Loss for All algorithms**

the different remaining features. However, there is a few difference between all features selected and some specific features chosen for Neural Network.

### *2.2 Comparison between normalized and unnormalized data*

For the algorithms that we use, we also applied the normalized and unnormalized data to them and compared the accuracy. Random Forest had a better result on unnormalized data, SVM and Neural Network had better performance on normalized data.

## 3 Baseline Methods

After combining data, we found some features are missing 60% of the data and it is hard to fill in the blank. Thus, we deleted these kinds of features. The remaining data is complete. From the remaining data, we selected the data of store1 to test and evaluate the algorithms. We chose this as our baseline setting and for each method, we changed the setting according to the performance.

## 4 Results

The smallest MSE of the three algorithms is listed in table 3. For all data, random forest achieved the best result, followed by SVM. The worst one is the Neural Network. From figure 6 to figure 14 are the results of all the methods. The figure's blue line represents the ground-truth value and the orange line represents the predicted value.

We found that the random forest took the shortest running time and SVM took the longest time from the experiments. The features used by random forest are the least, thus, the amount of data used by random forest is also the least. The store number, store type, and the number of departments were processed using a one-hot encoding method before normalization. The number of columns increased a lot after one-hot encoding; therefore, the amount of data increased more than the data used in the random forest. The prediction costs for non-linear SVM is proportional to the number of support vectors, and the costs grow linearly with the increasing size of the training set.[10] Thus, the longest time was used to train the SVM model.

### *4.1 Random Forest*

We applied the normalized data and unnormalized data of store1 to test which did better with the random forest algorithm. The unnormalized data obtained a smaller MSE than the normalized one. Thus, the following training process, e.g., feature selection and parameter tuning, used unnormalized data.

Using only store 1 data, we made the feature selection. We started with only one feature and chose the feature which gained the best prediction result. Then we added one more feature to the previous result to obtain the smallest MSE. If none of the rest features decreased the MSE value, we would start from the beginning to re-select the second-best feature as the base feature, then repeat the previous step to add and rinse the features until we acquired the best combination of features. We found the most efficient features for random forest were weeks' numbers, size of the store, store numbers, and the number of departments.

After selecting the features, we tuned the parameters with grid search. There are six features tuned: the number of trees (n_estimators), the number of features to consider when looking for the best split (max_features), the maximum depth of the tree (max_depth), the minimum number of samples required to split an internal node (min_samples_split), the minimum number of samples required to be at a leaf node (min_samples_leaf), and whether bootstrap samples are used when building trees (bootstrap). After executing grid search, we found that when n_estimators=100, max_features=sqrt, Max_depth=80, min_samples_split=2, min_samples_leaf=1, and bootstrap=true reached the best result.

Figure 6 presents the result of all data after feature selection. Figure 7 shows the first 1000 data of figure 6 since it is hard to read the trend of all data. It is clear to see that the predicted value and the true value in figure 7 are tightly stitched. From table 3, we can also observe that, with unnormalized data, the random forest achieved the smallest MSE value among all algorithms.

### *4.2 SVM*

Firstly, The store1 data was used to select the features. Using the forward-stepwise method introduced in the Feature Selec-

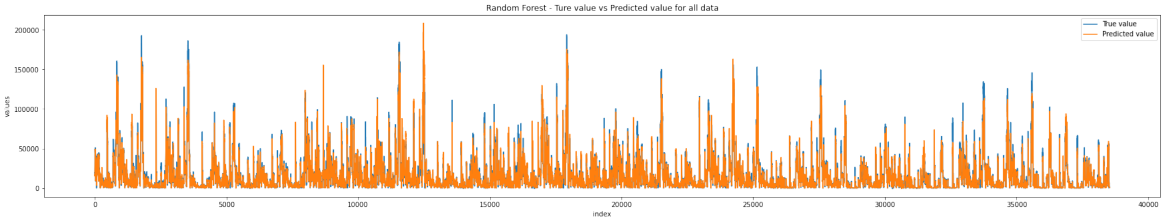

**Figure 6. Random Forest result for all the unnormalized data after feature selection**

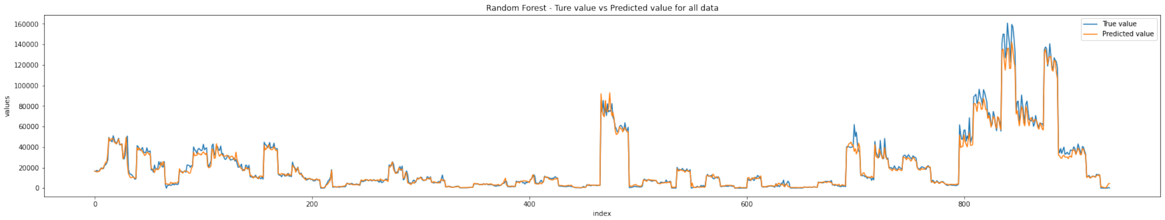

**Figure 7. Random Forest Resullt - The first 1000 data of Figure 6**

tion part, we found the data with features of except Markdown 1 to Markdown 5 obtained the smallest MSE. Figure 8 is the result before feature selection, figure 9 is the result after feature selection. In figure 8, we can obtain that the predicted value has an upward trend when the true value increases, but the fitness is very poor with true value. Comparing figure 8 and figure 9, we can obviously see that after feature selection, the trend of predicted value and the ground truth value is more consistent.

Secondly, we tested the result with the normalized data and unnormalized data. Figure 10 shows the result of unnormalized data with the same features that were selected in the previous step. Comparing figure 9 and figure 10, it is clearly shown that the normalization is very effective to SVM.

Thirdly, we used the selected features to tune the parameters with Grid Search. The best parameters were a combination of Radial Basis Function (RBF) kernel, auto gamma value, and regularization parameter set to 1.

Finally, we applied the features and parameters for all data. Compared with the result from different features and parameters, the MSE is the smallest with the selected features and tuned parameters. Figure 11 shows the result from all training data. Due to the large amount of data, it is hard to see the trend. Thus, in figure 12, the first 1000 data of the whole data was used as an example to display the trend. Comparing with figure 9, we found the trend of figure 9 is better.

It is not difficult to understand the result. Figure 9 only used the data of store1 as the train and test data, while figure 12 used the data of all stores for training and testing. The data in figure 12 have the characteristics of different stores, but the data in figure 9 only has the characteristic of store1. For algorithms, the prediction result is always better with the data which has obvious characteristics. Thus, the data of store 1 obtained a better result.

### *4.3 Neural Network*

In the neural network algorithm, we tested the normalized data and unnormalized data. From table 3, we can obtain that the MSE of normalized data of store1 is half of the unnormalized data. Thus, the following data processing part used normalized data only.

There are many models in the Neural Network. After reading the related papers[18][2], we decided to use long short-term memory (LSTM) and fully connected neural network (NN). The test results showed that NN has a smaller MSE than LSTM. Thus, we used the Neural Network to select features and tuned the parameters.

For the feature selection part, we used the data of store one as an example to select. Most authors used the previous few weeks' weekly sales data to predict the next few weeks' data with very limited features in other papers. For example, some people just used the number of departments and store numbers as their features. However, we used a different method to predict and the result showed that with all features, our Neural Network model obtained the best result.

Since we already obtained some effective features from the random forest and SVM algorithms, firstly, we tried the features. The loss result showed that using SVM features can get a smaller loss. The features selected by random forest are less than SVM, thus, we thought Neural Network might perform well with more features. Then we dropped fewer features in the process and found that with all features, the result was the best. As shown in table 4, it is easy to see that with the number of features increase, the test loss decreases. The data with all features achieves the least test loss.

Figure 13 and figure 14 exhibit the predicted result with all features. Figure 13 displays the result of the first 1000 data with using only store 1 data to train, figure 14 shows the result of the first 1000 data with using all data to train. In the first 800 data, the result after training all data performs better. However,

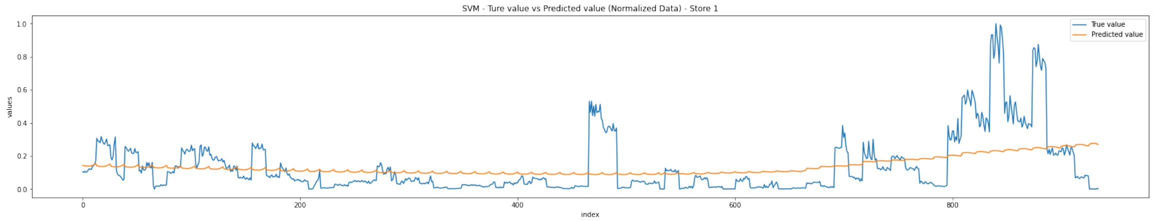

**Figure 8. SVM result for the normalized data of Store 1 with random features**

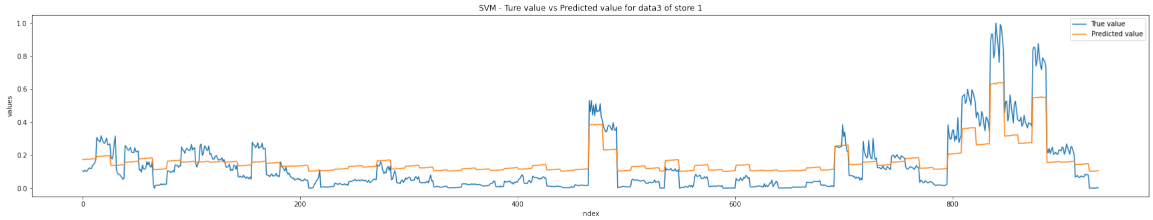

**Figure 9. SVM result for the normalized data of Store 1 after feature selection**

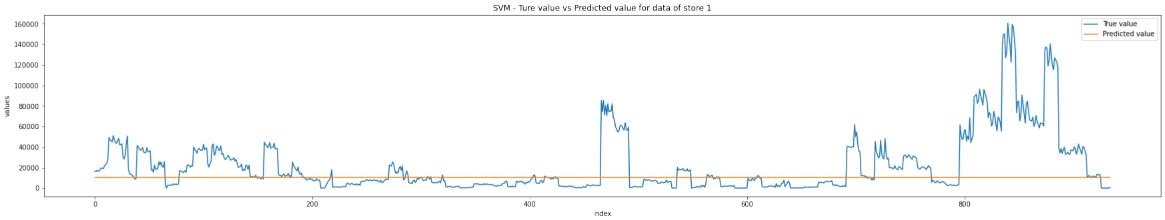

**Figure 10. SVM result for the unnormalized data with using the same feature of Figure 11**

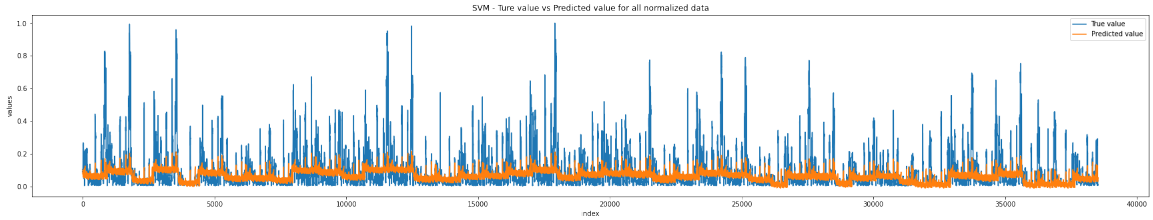

**Figure 11. SVM result for all the normalized data after feature selection**

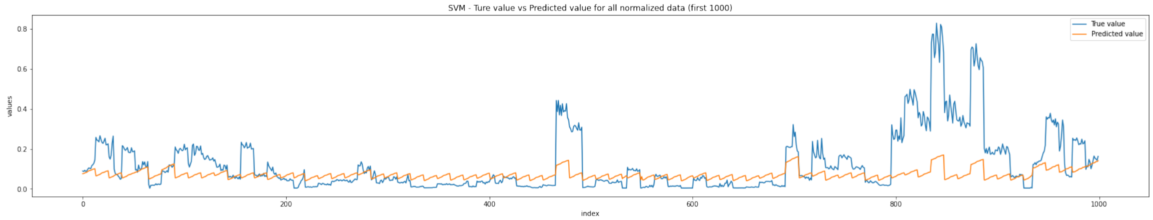

**Figure 12. SVM Result - The first 1000 data of Figure 11**

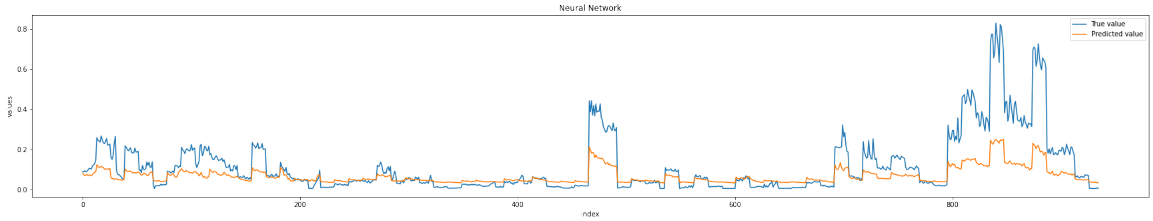

**Figure 13. Neural Network Result - The first 1000 data after training all data**

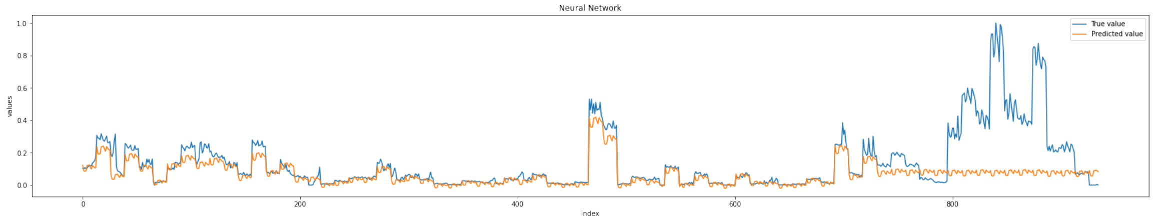

**Figure 14. Neural Network Result after training only store1 data**

when the weekly sales increase, figure 13 fits better than figure 14. That is, using only store1 data to train and test achieved a better result than using all data. The conclusion is the same as we obtained in the precious SVM section.

## DISCUSSION

### 1 Lessons Learned

There are so many lessons we learned during the project.

Firstly, we realized that data visualization plays a significant role in not only understanding the dataset itself but also features selection. Various kinds of graphs provide diverse views to find out the characteristics of data. For example, heat maps give us information about correlations between attributes that help us select the features. At the same time, box plots tell us the distribution of the data, particularly about the outliers.

Secondly, numerical data split into discrete data should be based on data distribution. In order to apply the Apriori method, we need to covert the numerical data into strings of discrete data. We applied both equal width and equal frequency on numerical data to discrete our data. The result of equal frequency is similar to the facts that we concluded from the data visualization. At the same time, the result of equal width is farther away from the facts of data visualization. Therefore, we need to verify the distribution before splitting data.

Thirdly, normalization is an excellent tool to get rid of the unbalanced data set, but it does not work well on Random Forest. Compared with SVM and Neural Network, the performance of Random Forest is the worst on normalized input data.

Lastly, literature research is very necessary. We need to get to know related work which offers us some guidance and suggestions. For instance, we adopt our models based on literature research. When we get a strange or confusing result, reading others' papers always brings us explanations or meaningful thoughts.

### 2 Better Performance

First of all, applying normalization on input data has a significant improvement in Neural network and SVM performance. The differences among different features are a lot. Some features are in the range of no more than 100, but some are in range of more than 100000. The features with large numbers have a more significant impact on the result. Thus, from figure 10, We can hardly see the changes in the forecast results, and the MSE also shows that the prediction is terrible for normalized data.

Moreover, a split train-test set on time has better achievement compared with cross-validation. Our data is a time series. Split by time can contain more information. As we split data randomly, the performances of all selected methods are all worse than the performances of splitting by date.

Ultimately, feature selection is remarkable significance. All three methods all applied feature selection. Based on comparison with figure 8 and figure 9, the performance of SVM after feature selection has a remarkable improvement.

subsection3 Future Work In the results part of the evaluation section, we mentioned that the prediction cost for SVM is much greater than other algorithms. After reading the papers [10] [16] , we found that two more methods to decrease the time costs of SVM. Paper [10] optimized over the space of tree-structured features, paper [16] accelerated the training of kernel machines by mapping the input data to randomized low-dimensional feature space and applying the fast linear algorithms to train. We will try the two methods in our future work to see if it can speed up the SVM prediction.

In addition, we did some researches on neural networks and found that when combining different models together, e.g., LSTM and artificial neural network (ANN), will obtain a better outcome. Thus, we will try combining different models together to train the data.

Since Kaggle provided other sales data related to Walmart, we will use our method and apply it to the other datasets to see if we can obtain the same conclusion as we obtained in this paper.

## CONCLUSION

### 1 Summay

In the whole experiment, we read the data first and found that data was incomplete, so we did the data preprocessing, including combining data from different tables, filling the blank values, deleting features with too many vacancies, transforming data by one-hot encoding method, and changing string type to integer type.

Next, we decided to use the random forest, SVM, and Neural Network in the project.

We also did experiments on deciding if normalized or unnormalized data should be used. At the same time, we did feature selection and parameters were tuning. From the experiment, we obtained different most useful features and the best parameters for different algorithms.

Finally, we used the data and parameters to predict the result with different algorithms. The result showed that the random forest is the best algorithm in this dataset.

### 2 Key Tasks

- Data preprocessing
- Algorithms selection
- Normalization
- Feature selection
- Tuning parameters

### 3 Findings

- Understanding the dataset is very important. We can learn more about the dataset through data visualization.
- Feature selection is critical. Using different features has a huge impact on the predicting result.
- Splitting numerical data into discrete data should be based on data distribution.
- Normalization is an excellent tool to get rid of an unbalanced data set, but it does not work well on all algorithms.
- Literature research, like reading papers, is very imperative.
- For this dataset, splitting the train and test data on time is better than cross-validation because the data is related to time series.

## REFERENCES

[1] Mean squared error. `https://www.statisticshowto.com/mean-squared-error/`.

[2] Lucas F.M. da Silva A.L.D. Loureiro, V.L. Miguéis. Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, 114:81–93, 2018.

[3] Fifit Alfiah, Bagus Wahyu Pandhito, Ani Trio Sunarni, Deni Muharam, and Pradiko Roliwinsyah Matusin. Data mining systems to determine sales trends and quantity forecast using association rule and crisp-dm method. *International Journal of Engineering and Techniques*, 4(1):186–192, 2018.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Christopher JC Burges, Bernhard Scholkopf, and Alexander J Smola. *Advances in kernel methods: support vector learning*. MIT press Cambridge, MA, USA:, 1999.

[6] Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126, 2004.

[7] Márcio das Chagas Moura, Enrico Zio, Isis Didier Lins, and Enrique Droguett. Failure and reliability prediction by support vector machines regression of time series data. *Reliability Engineering & System Safety*, 96(11):1527–1534, 2011.

[8] M. Pei J. Han, J. Kamber. Data mining concepts and techniques. 2012.

[9] Constance L Hays. What wal-mart knows about customers' habits. *The New York Times*, 14, 2004.

[10] Cijo Jose, Prasoon Goyal, Parv Aggrwal, and Manik Varma. Local deep kernel learning for efficient non-linear svm prediction. In *International conference on machine learning*, pages 486–494, 2013.

[11] Kirill. Random forest, rnn, walmart sales forecase. `https://www.kaggle.com/datamany/random-forest-rnn-walmart-sales-forecast`.

[12] Xin Liu and Ryutaro Ichise. Food sales prediction with meteorological data—a case study of a japanese chain supermarket. In *International Conference on Data Mining and Big Data*, pages 93–104. Springer, 2017.

[13] Kevin Maclver. Retail sales analytics through time series forecast using rnn. `https://medium.com/@kmacver/retail-sales-analytics-through-time-series-forecast-using-rnn-1`

[14] Patrick Meulstee and Mykola Pechenizkiy. Food sales prediction:" if only it knew what we know". In *2008 IEEE International Conference on Data Mining Workshops*, pages 134–143. IEEE, 2008.

[15] Ping-Feng Pai. System reliability forecasting by support vector machines with genetic algorithms. *Mathematical and Computer Modelling*, 43(3-4):262–274, 2006.

[16] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

[17] Nirav Shah, Mayank Solanki, Aditya Tambe, and Dnyaneshwar Dhangar. Sales prediction using effective mining techniques. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 6(3):2287–2289, 2015.

[18] Jiaxin Xu Xiaochen Chen, Lai Wei. House price preidction using lstm. `https://arxiv.org/pdf/1709.08432.pdf`.

## APPENDIX

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. All the work has been done by individual group members.