# Sales Forecasting

*Abstract*—When it comes to the selling of products and services to consumers, retail is distinct from wholesale, which is the sale of goods and services to company or institutional clients[1]. According to Statista, the retail market accounts for 5.7 percent of the country's total gross domestic output (GDP). Founded in 1962, Walmart is a huge retail firm that is often referred to as one of the most successful in the world. As a result, beginning this week, we will be offering stock forecasts for Walmart sales on a weekly basis instead of monthly. No matter how large or small a retail establishment is, socioeconomic difficulties have a substantial influence on its sales. The result is that the number of input characteristics in the model has been enhanced by incorporating employment, the consumer price index, and fuel expenses. Various machine learning models, such as linear regression, support vector regression, decision tree regression, random forest regression, and transformer networks, will be used in this research [7]. In order to anticipate Walmart's sales, it is necessary to take into account the effect of other factors such as "Temperature," the Consumer Price Index (CPI), unemployment, and gasoline costs, among others. Another key objective of using a large number of models is to guarantee that the final model is both computationally light and competent in terms of performance, which is a tough goal to attain given the high number of models being used. The model will be deployed locally in the near future as an interactive graphical user interface online application, which will enable it to be used in a realistic setting while also enabling the response to be used to retrain the model in the future.

***Index Terms***—**Sales forecasting, Effect of socioeconomic factors on sales of a product, Linear Regression model, Support Vector Regression model, Decision Tree Regression model, Random Forest Regression model, KNN Regression model, Transformer Network model**

## I. Introduction

TODAY When it comes to business survival, the market is not the same as it was hundreds of years ago; there are many options, and as a result, there is a great deal of competition, not only for selling the most amount of goods, but also for optimising business investment, so that the greatest amount of revenue can be generated with the least amount of investment possible. Every firm employs a number of methods, one of which is the administration of advertising campaigns, and another of which is the management of inventory levels.

To operate efficient advertising campaigns, it is critical to understand your target audience as well as the sales patterns of your products. Client segmentation is often used to get a better understanding of the customer, but demand forecasting is a highly effective method of gaining an understanding of the trends in the sale of your product and the highs and lows of demand. When it comes to our use case, forecasting is being done for Walmart, and the data being analysed for the project just happens to include store-specific sales information, which is very useful. Upon further examination of the data in the sales column, we see that the performance of each shop is not the same, and we may utilise sales forecasting to possibly enhance the sales of certain locations as well. a

In addition, managing the stock of items in inventory in order to reduce waste of goods is a very important component in sales forecasting. Another use of sales forecasting is in inventory management. Those firms that do not have effective inventory management may struggle to generate a profit, and they may even fail to exist in the market altogether. If you want to ensure that you always have an adequate number of items in your inventory, you must forecast potential future demand. According to the sort of commodities sold and the type of market in which a company operates, the time period for forecasting may fluctuate for various enterprises. When compared to a supermarket shop, the timestamp for a bakery business is much smaller. Sales forecasting is one of the methods for predicting demand in the marketplace.

Aside from the fact that we can optimise the sale itself and increase the effectiveness of advertising campaigns, we can take advantage of other factors such as the CPI, unemployment, and temperature to provide better offers and bargains to clients in order to make them more connected to the firm. However, in this situation, we will limit ourselves to sales forecasting and attempt to make use of these qualities in order to estimate sales.

## II. Data Description

The Walmart dataset is divided into three separate csv files, which are referred to as "train.csv," "features.csv," and "stores.csv." The "Store," "Dept," "Date," "Weekly Sales," and "IsHoliday" fields are all included in the Train.csv file. The following sections provide an explanation of the characteristics included in the train.csv data file:

A. Store: store numbers,
B. Dept: The department numbers,
C. Date: The weekly dates,
D. Weekly_Sales: The weekly sales aggregated on corresponding dates in the region.
E. IsHoliday: The True and False values based on the fact that if it is a week of special holiday or not.

“features.csv” contains "Store", "Date", "Temperature", "Fuel_Price", "MarkDown1", "MarkDown2", "MarkDown3", "MarkDown4", "MarkDown5", "CPI", "Unemployment", "IsHoliday". The detailed description of the attributes mentioned above are provided below;

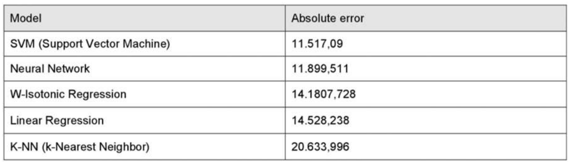
A. Store: ID of the stored
B. Date: Weekly dates
C. Temperature: Temperature aggregated on the corresponding date in the region
D. Fuel_Price: Fuel price aggregated on the corresponding date in the region
E. MarkDown1-5: Data related to promotional markdowns, these columns are not very much described they are quite anonymous
F. CPI: The Customer Price Index
G. Unemployment: The unemployment rate
H. IsHoliday: If it is a special holiday week or not

The “store.csv” file contains features like “Store”, “Type” and “Size”. The Store column contains the store numbers, type and size columns contain the type and corresponding size of the store.

## III. Related Work

ARIMA (statistical domain method) and Neural Networks are the two most often used approaches for predicting current time series (machine learning domain). Because the current values in stock price forecasting and sales forecasting are linked to prior stock prices and sales, ARIMA models are especially well suited for these sorts of situations. When ARIMA models use the relationship between the present and past data points, they may produce exceptionally acceptable and great outcomes. For example, Adebiyi [5]'s forecast of Nokia stock price in a 2010 research showed a high degree of accuracy in predicting. However, for time series analysis that includes nonlinear patterns, neural networks' ability to represent and learn complex connections in nonlinear data makes them an even better option because of their ability to represent and learn complex interactions. In many instances when the time series is supposed to be linear and ARIMA is used, the results are unsatisfactory since other socioeconomic variables such as recession and unemployment have a large influence on pricing and ARIMA is unable to adequately capture these anomalies. The same author [5] has released another research on forecasting Dell stock prices in which he uses both techniques, with the Neural network surpassing the ARIMA model by a wide margin. In our example, we investigate three models, assuming that our data may be seasonal and that other socioeconomic factors such as the Consumer Price Index (CPI), unemployment, and fuel prices may influence sales.

Stojanovi, Nikola, Marina Soldatovi, and Milena Milievi examined and worked on the dataset we are working with in 2014 [6]. Forecasting has been performed using a number of models, including SVM, Neural Network, W-Isotonic regression, Linear regression, and the KNN model, among others. Despite the fact that the performance achieved excellent accuracy in the use case, the work was not application oriented; nonetheless, the study's performance is mentioned below;



| Model | Absolute error |
|---|---|
| SVM (Support Vector Machine) | 11.517,09 |
| Neural Network | 11.899,511 |
| W-Isotonic Regression | 14.1807,728 |
| Linear Regression | 14.528,238 |
| K-NN (k-Nearest Neighbor) | 20.633,996 |

Table 1: Performance of various models in paper[6] published in 2014

The result shown in the above table clearly illustrates that the performance of the Neural network is pretty promising; to enhance the model, we will integrate more factors such as unemployment, CPI, temperature, and so on. Assuming that this will add value to the model and lead to improved outcomes. Another distinction is that we also use the ARIMA model to generate a different version of the model. Our study will concentrate on the application viewpoint of forecasting, as opposed to the work we have seen so far, which is not application driven.

## III. Block Diagram of the Work

The Block diagram of the tentative work is shown below;

### A. *Linear Regression Model*

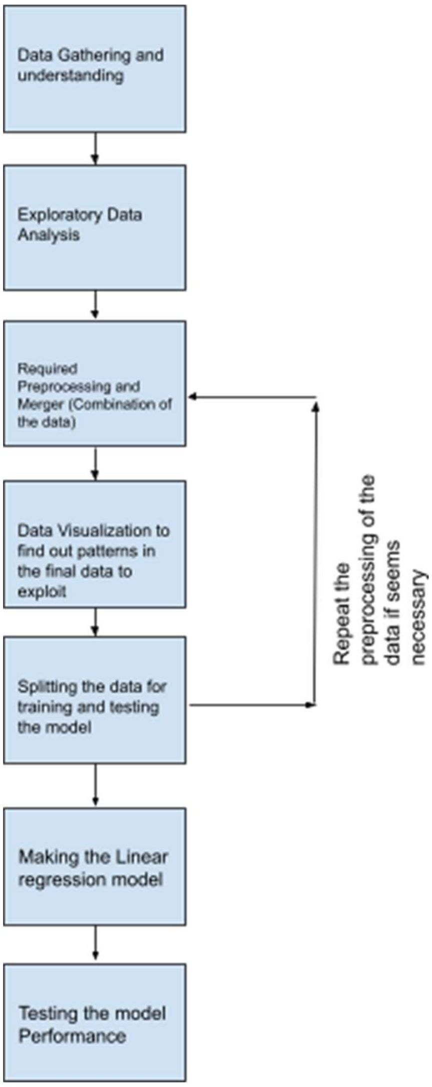

Data Gathering and understanding
Exploratory Data Analysis
Required Preprocessing and Merger (Combination of the data)
Data Visualization to find out patterns in the final data to exploit
Splitting the data for training and testing the model
Making the Linear regression model
Testing the model Performance
Repeat the preprocessing of the data if seems necessary

Figure 1: Block diagram of linear regression model

### B. *Decision Tree Regression Model*

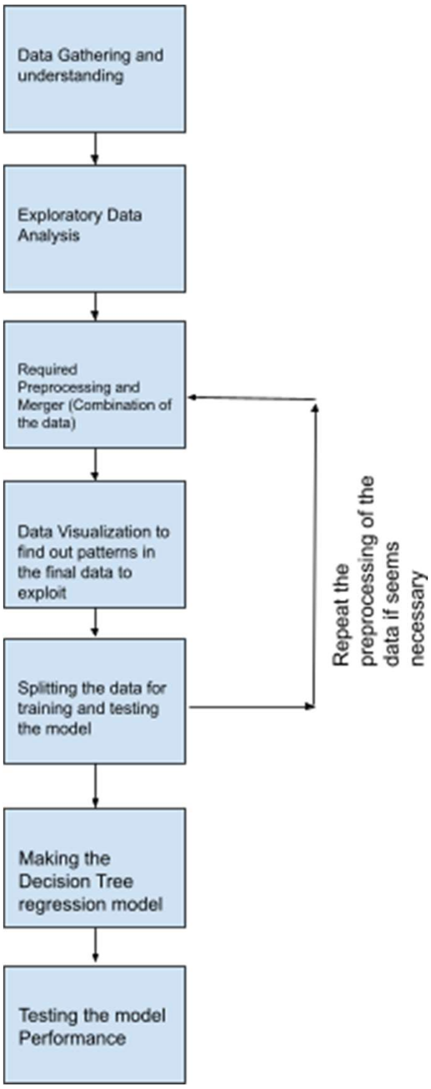

Data Gathering and understanding
Exploratory Data Analysis
Required Preprocessing and Merger (Combination of the data)
Data Visualization to find out patterns in the final data to exploit
Splitting the data for training and testing the model
Making the Decision Tree regression model
Testing the model Performance
Repeat the preprocessing of the data if seems necessary

Figure 2: Block diagram of decision tree regression model

### C. *Random Forest Regression Model*

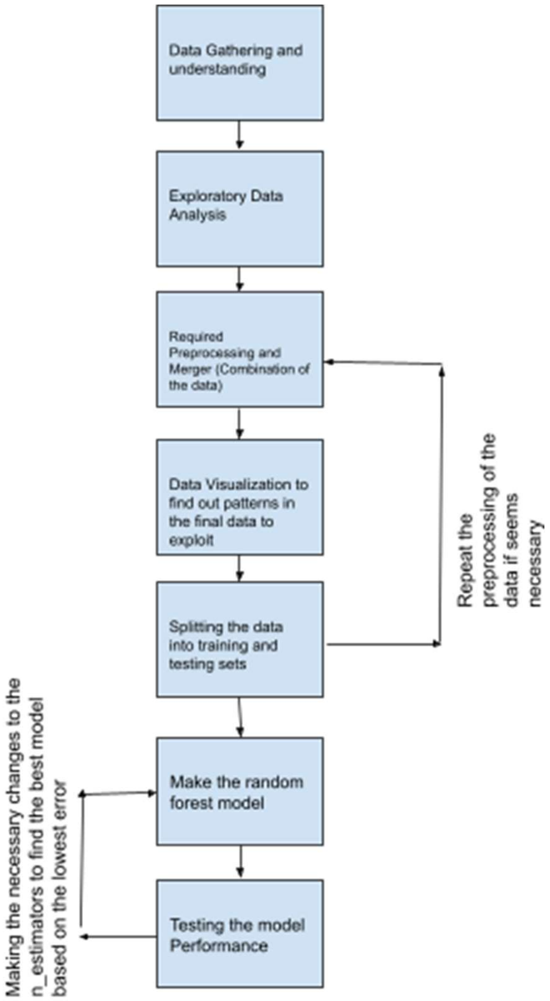
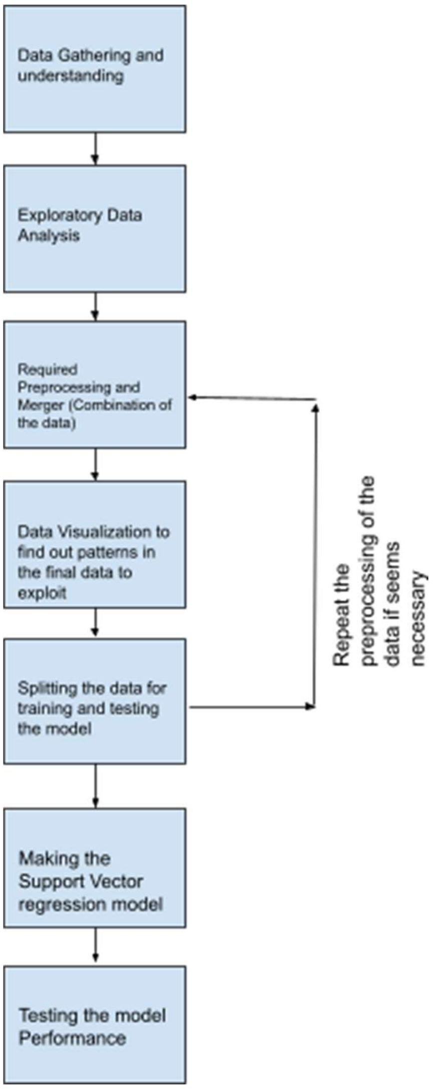

Data Gathering and understanding
Exploratory Data Analysis
Required Preprocessing and Merger (Combination of the data)
Data Visualization to find out patterns in the final data to exploit
Splitting the data into training and testing sets
Make the random forest model
Testing the model Performance
Repeat the preprocessing of the data if seems necessary
Making the necessary changes to the n_estimators to find the best model based on the lowest error

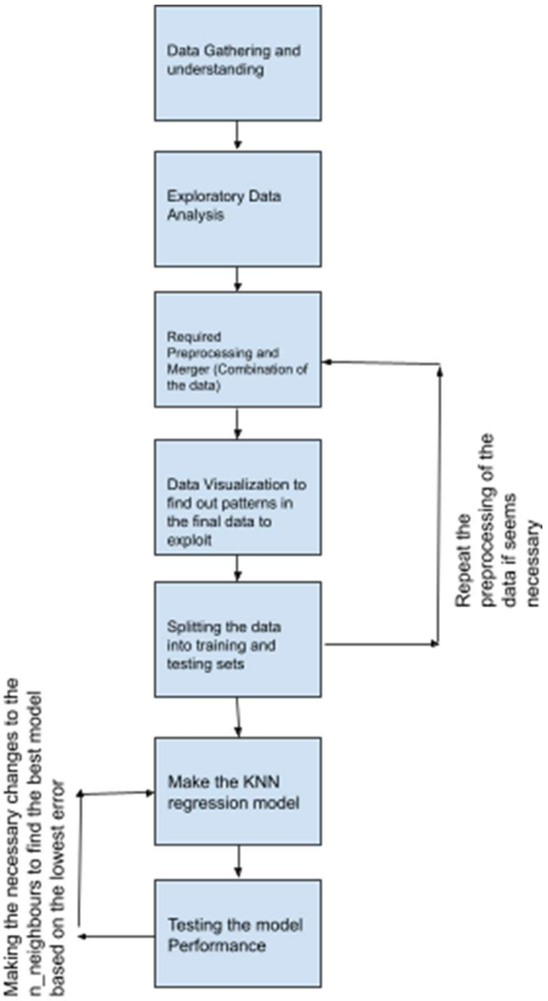
Figure 3: Block diagram of random forest regression model

### D. *Support Vector Regression Model*



Data Gathering and understanding
Exploratory Data Analysis
Required Preprocessing and Merger (Combination of the data)
Data Visualization to find out patterns in the final data to exploit
Splitting the data for training and testing the model
Making the Support Vector regression model
Testing the model Performance
Repeat the preprocessing of the data if seems necessary

Figure 4: Block diagram of support vector regression model



Data Gathering and understanding
Exploratory Data Analysis
Required Preprocessing and Merger (Combination of the data)
Data Visualization to find out patterns in the final data to exploit
Splitting the data into training and testing sets
Make the KNN regression model
Testing the model Performance
Repeat the preprocessing of the data if seems necessary
Making the necessary changes to the n_neighbours to find the best model based on the lowest error

Figure 5: Block diagram of K nearest neighbours regression model

### E. *KNN Regression Model*

### F. *Transformer Network Model*

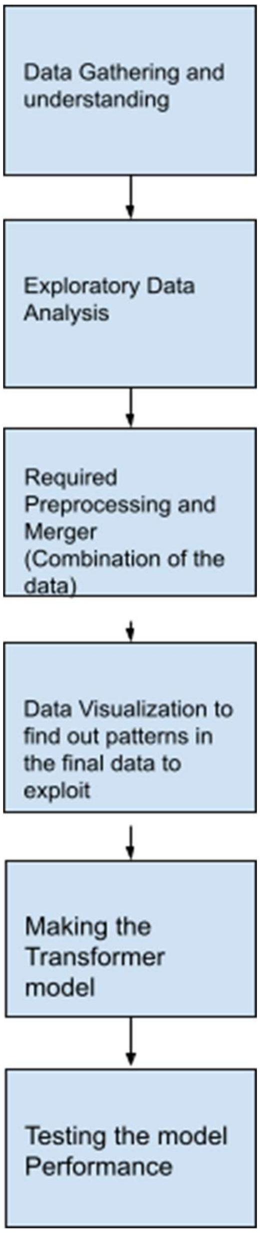

Data Gathering and understanding
Exploratory Data Analysis
Required Preprocessing and Merger (Combination of the data)
Data Visualization to find out patterns in the final data to exploit
Making the Transformer model
Testing the model Performance

Figure 6: Block diagram of transformer network model implementation

## IV. Exploratory Data Analysis

Before we can begin building the model, we must first understand all of the potential patterns and connections that exist between each and every feature. Traditional machine learning techniques and forecasting are distinct in that in forecasting the most significant column is either the time or date column, but in classical machine learning methods the time or date column is irrelevant. The date column is used to keep track of the order in which the data points were collected, so that we may compare the current sale with the sales from previous days. The analysis of weekly sales will be carried out in accordance with the date column, in addition to the above. We shall carry out the procedure. exploratory data analysis on the raw data to find out if there is any visible pattern in the columns.

### *A. Train.csv*

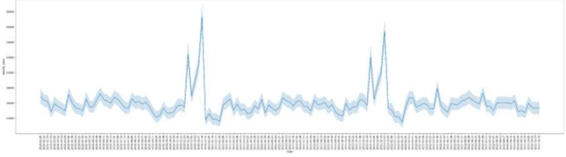


Figure 7: Variation of weekly sales with respect to date

The pattern of sales is readily obvious in the following picture; we can see that the sales have a propensity to recur throughout the course of the year, and that there are sudden rises in sales during a certain time period for each year. According to the data description provided by the source, this dramatic shift might be explained by the fact that sales surge throughout festival seasons, particularly during the events listed below;

A. Super Bowl: 12-Feb-2010, 11-Feb-2011, 10-Feb-2012, 8-Feb-2013
B. Labor Day: 10-Sep-2010, 9-Sep-2011, 7-Sep-2012, 6-Sep-2013
C. Thanksgiving: 26-Nov-2010, 25-Nov-2011, 23-Nov-2012, 29-Nov-2013
D. Christmas: 31-Dec-2010, 30-Dec-2011, 28-Dec-2012, 27-Dec-2013

The shaded zone is visible due to the various shops and their accompanying sales; the hard line depicts the mean value of sales from all of the retailers on that particular day.

### *B. Features.csv*

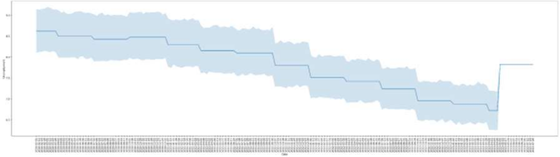


Figure 8: Variation of Unemployment with respect to the date

The unemployment rate has been dropping through time, as seen in the graph above, which displays the variance of unemployment from the beginning of time till the present. Unemployment climbed dramatically in the second half of the year 2013 and then remained stable beyond that point. Aside from the dramatic spike in 2013, there has been no other significant shift.

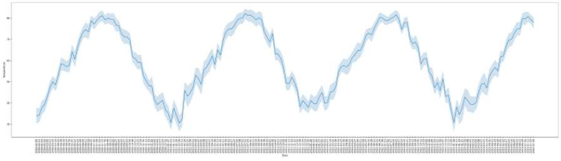


Figure 9: Variation of Temperature with respect to date

The temperature fluctuation seen in the above image is shown over a period of time, and we can clearly see that the temperature rises and drops throughout the year, with the same pattern repeating year after year. However, an intriguing thing to note is that the lowest temperature number has not been constant from year to year, but has grown overall. It is shown as a darkened zone, which indicates varied temperatures in different storage locations; nonetheless, the pattern is consistent throughout all of them.

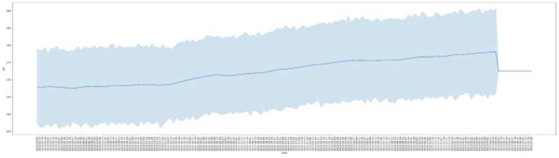


Figure 10: Variation of CPI with respect to date

The chart above depicts the variance of the Consumer Price Index over time, and we can see that, in contrast to unemployment, the CPI is growing with time. In 2013, the Consumer Price Index (CPI) is experiencing a dramatic shift, but this time the movement is not in a favourable direction, but rather in the opposite one. The probable causes of inflation in the Consumer Price Index (CPI) have been discussed in many publications during the course of the year; some of the most significant include corruption, economic development, and foreign investment. [3]

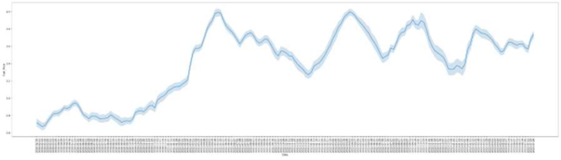


Figure 11: Variation of Fuel Prices with respect to date

The price of gasoline seems to be growing throughout the course of the year; there are ups and downs in each year, but generally the price is rising; this pattern may be supported by real-world causes that can be found in the actual world. Fuel prices influence practically every organisation in a variety of ways, both directly and indirectly[4]. Because production and transportation are heavily reliant on gasoline, the price of fuel has the ability to effect the total sale of a company.

### C. *Stores.csv*

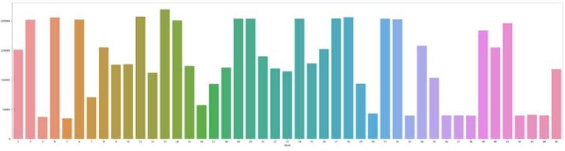


Figure 12: Variation of Fuel Prices with respect to date

Aside from the store number and size, the Stores file does not include many other relevant characteristics. The x axis of the plot in figure number 12 represents the number of stores, while the y axis represents the size of the stores.

### D. *Combined Data*

Upon combining the train and feature data frames, as well as aggregating the data throughout time periods, we can get the average Weekly sales and other characteristics. The first five rows of the final data frame are included inside the following table.

| Date | Weekly_Sales | IsHoliday | Temperature | Fuel_Price | MarkDown1 | MarkDown2 | MarkDown3 | MarkDown4 | MarkDown5 | CPI | Unemployment | Size |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2010-02-05** | 16836.121997 | FALSE | 33.277942 | 2.717869 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | 167.398405 | 8.576731 | 137430.535364 |
| **2010-02-12** | 16352.056032 | TRUE | 33.361810 | 2.696102 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | 167.384138 | 8.567309 | 137622.780785 |
| **2010-02-19** | 16216.658979 | FALSE | 37.038310 | 2.673666 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | 167.338966 | 8.576351 | 137278.637219 |
| **2010-02-26** | 14899.549688 | FALSE | 38.629563 | 2.685642 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | 167.691019 | 8.561375 | 137346.344629 |
| **2010-03-05** | 15921.015727 | FALSE | 42.373998 | 2.731816 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | -9999.0 | 167.727351 | 8.572689 | 137576.841033 |

Table 2: Aggregated (over Date) merged(features + train) dataset

Aside from this aggregate version, we have generated a data frame that is comparable to this one for each and every Walmart shop. Now we'll attempt to develop some broad generalisations in order to finalise the qualities that will be employed in the modelling process in the future. Our initial motivation will be to add as many features as we possibly can; however, the ultimate choice will be based on whether or not the feature is linked to the Weekly Sales feature in some way or another.

We will now do exploratory analysis on the combined dataset, which is necessary in order to determine the link between the other columns and the Weekly Sales column.

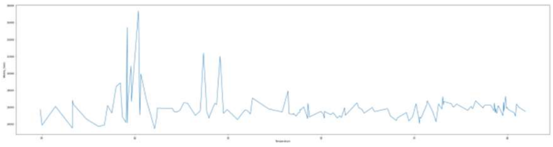


Figure 12: Variation of Weekly_Sales with respect to temperature

When the temperature is lower, the value of sales is greater, but the difference is not statistically significant.

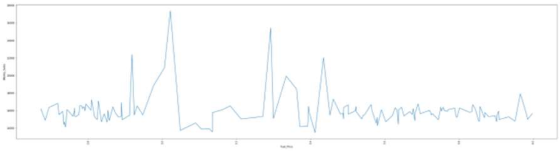


Fig 13: Variation of Weekly_Sales with respect to Fuel_price

Similarly to the graph of weekly sales with regard to temperature, the graph of weekly sales with respect to fuel price shows that the quantity of sales is larger when the fuel price is lower. However, once again, the pattern is not statistically significant.

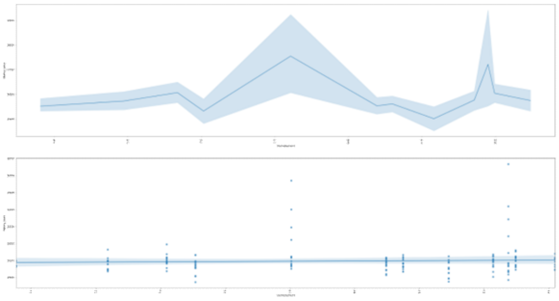


Fig 14: Variation of Weekly_Sales with respect to Unemployment rate
(lien plot and regression plot)

The graph of the Unemployment rate vs Weekly Sales does not reveal any statistically significant trends in the volatility of sales. Despite the fact that sales seem to be greater when the unemployment rate is higher, this is not the case. The regression plot validates the hypothesis that there is no statistically significant association between the unemployment rate and weekly sales (or weekly sales). The regression line has a horizontal slope to it.

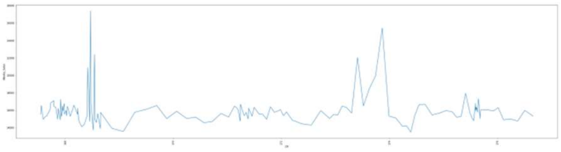
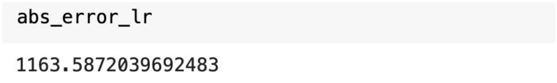


Fig 15: Variation of Weekly_Sales with respect to CPI rate

In Figure 7, we can see that the Sale on holidays days is greater than on non-holiday days, which is supported by the barplot of IsHoliday column with regard to the Weekly Sales mean column. This trend can also be seen in the overall plot of Weekly Sales with respect to time in Figure 7.

## V. Applying Linear Regression

By the time we start building linear regression model for the forecasting of the sales, will be having Weekly_Sales, IsHoliday, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3, MarkDown4, MarkDown5, CPI, Unemployment, Size, diff_1. We will be using the weekly sales column as our target variable and all the other columns as the input. As mentioned earlier, our main goal is to do a comparative study between different machine learning techniques to find out the regression between the input variables and weekly sales[15]. We will be using the default variables for the linear digressions model we make. The error achieved in the linear regression model is mentioned as below;



```
abs_error_lr
```

```
1163.5872039692483
```

The error we have received in the linear regression model is not a benchmark and because it is the very first model we are making, So we cannot be sure about the error to be minimal yet. The prediction made by the linear regression model is shown in the following graph where the blue line shows the actual values and the red line shows the predicted values.

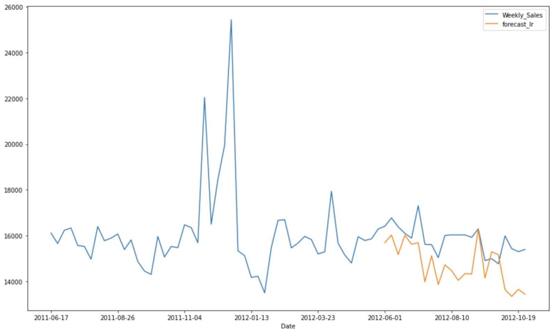
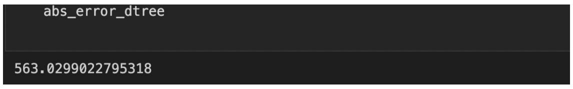


Figure 16: Prediction from linear regression model

## VI. Applying Decision Tree Regression

The decision trees are used to fit a sine curve to a set of noisy observations that have been added. As a consequence, it learns to approximate the sine curve using local linear regressions that are learned locally [8][9]. Just like the linear regression model will be using the default values for the decision tree regression model as well. The absolute error achieved using decision tree regression is shown below;



```
abs_error_dtree

563.0299022795318
```

The absolute error achieved by the decision tree regression is almost half of the same achieved by linear regression, so we can be sure that the decision tree regression model is better than the linear regression model. The prediction made by the decision tree regression model is shown in the following graph where the blue line shows the actual values and the red line shows the predicted values.

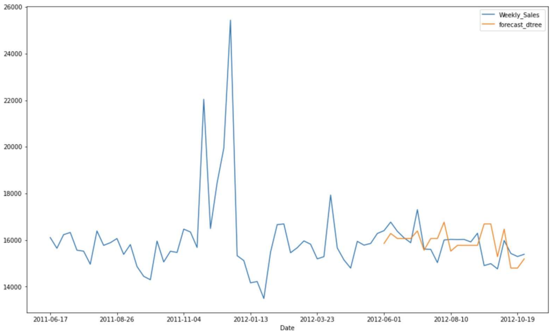
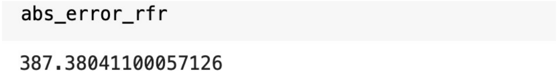


Figure 17: Prediction from decision tree regression model

From the graph as well we can see that the production done by decision tree regression is very close to the actual weekly sale values.

## VII. Applying Random Forest Regression

A supervised learning technique, Random Forest Regression employs the ensemble learning approach to predict the outcome of a regression. The ensemble learning method is a methodology that combines predictions from numerous machine learning algorithms to get a forecast that is more accurate than a single algorithm. In the first attempt we make the random forest regression model with the default parameters, the error achieved by default parameter is is mentioned below;



```
abs_error_rfr

387.38041100057126
```

To achieve the best value of mean absolute error we need to find out the best value for the number of estimators in this case decision trees. To find out the optimal number of estimators we will use brute force and check for the absolute error for various values of the number of estimators. The variation of absolute error with respect to the number of estimators as shown in the following graph.

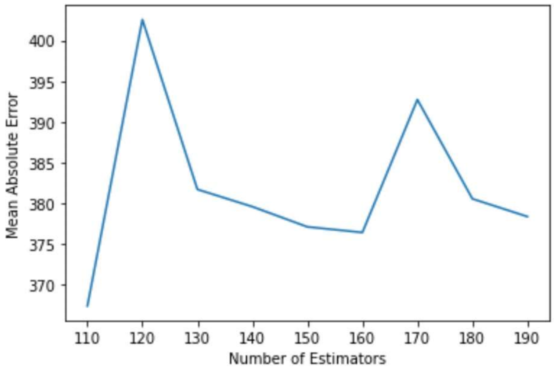


Figure 18:Variation of mean absolute error with respect to number of estimators in random forest regression

In the graph above the X axis represents the number of estimators and the Y axis represents the mean absolute error. We can see from the plot above that the best value of Mane absolute error is achieved by the number of neighbours equal to 110, now we will use numbers of neighbours as 110 for our final model and calculate the error.

The error achieved by their best value of number of neighbours is 387.51, from the mean absolute error we got from the random forest regression, we can assume that the best model so far is random forest regression.

The prediction made by the random forest regression model is shown in the following graph where the blue line shows the actual values and the red line shows the predicted values.

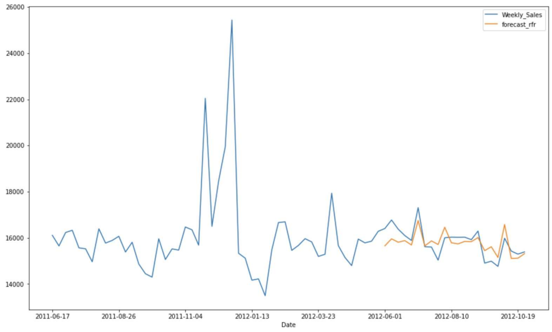


Figure 19: Prediction from random forest regression model

From the graph as well we can see that the production done by random forest regression is closest to the actual weekly sale values.

## VIII. Applying Support Vector Regression

So far we have witnessed that the best regression model for our use case is random forest regression, now it is time to move on to our next regression model which is Support Vector Regression. Just like the previous cases will be making the support vector regression with default parameters, and then we will analyse its performance and compare it with the previous models we have prepared.

After training the support vector regression model with our data set the mean absolute error achieved on the test data is mentioned below;

```
abs_error_svr
```

```
520.0152451944873
```

Now let's look at the prediction made by the support vector regression model, although the mean absolute error achieved by support vector regression is not better than the Random Forest regression model but still it is not useless and it is better than the linear regression model.

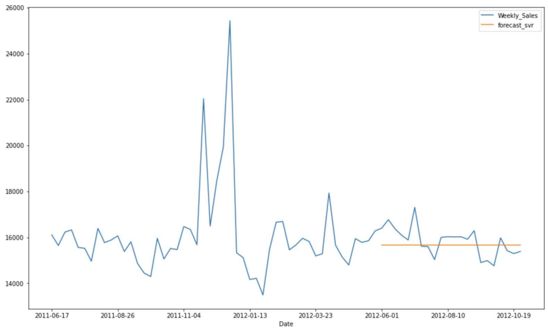


Figure 20: Prediction from support vector regression model

As we have seen the error of the support vector regression model is quite low but on perceiving the prediction made by the support vector regression model we can tell that it is a bad model as the prediction made by this model is a straight line. Considering all the results we have acquired so far, we can say that the best model for weekly sales forecasting is random forest regression[10].

## IX. Applying KNN Regression

K nearest Neighbour regression is another regression model we are interested in to forecast the weekly sales. Just like the random forest regression model, in the case of the K nearest neighbours regression model we have the hyper parameter “n_neighbours”. To achieve the best result using a K nearest Neighbour regression we have to be certain about the number of numbers we are using for the model.

The very first implementation of KNN regression would be using the default parameters, the error achieved by the default K nearest neighbours regression is mentioned below.

```
abs_error_knr
```

```
677.5019043127078
```

After making the default models and getting its Absolute loss now it is time to get the best value of the number of neighbours which can lead us to the lowest error possible. Just like the random forest regression we will try to find the best value of the number of neighbours using a brute force approach. The change in Mean absolute error with respect to the number of neighbours is shown in the following figure.

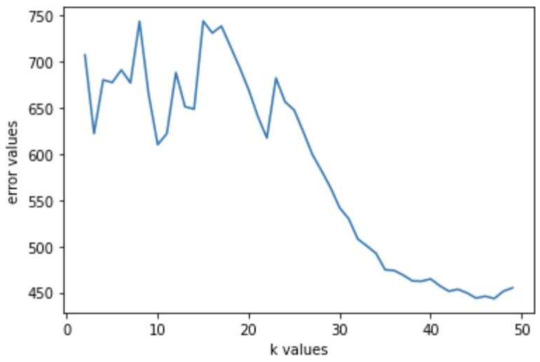


Figure 21: Variation of mean absolute error with respect to number of neighbours in KNN regression

From the above figure it is clear that the best value of absolute error is possible by using a k value equal to 48. Now we will use 48 as the number of neighbours for the K nearest neighbours regression model and re-train it[12].The error achieved by using the best value of the number of neighbours is 443.41.

Now as we have the best K Nearest Neighbour regression model we will do that prediction. The prediction graph using the best KNN regression model is shown in the following figure.

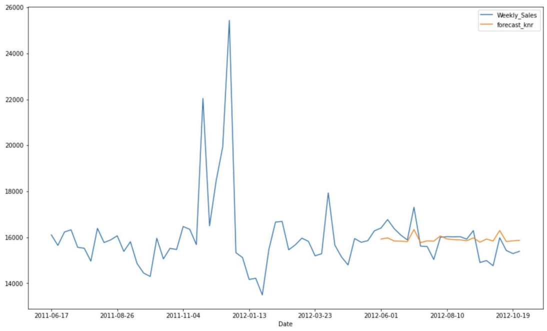


Figure 22: Predictions from K nearest neighbours regression model

The performance of the current model is quite promising but it is still not better than the random forest regression model.

## X. Applying Transformer Network

We are already through all the different machine learning regression models, now we will proceed with the neural network approach to implement the transformer network for regression.

```
______________________________________________________________________________
 Layer (type)                Output Shape         Param #     Connected to
==============================================================================
==================
 input_16 (InputLayer)       [(None, 1, 12)]      0           []

  layer_normalization_101   (Layer      (None,   1,   12)                   24
['input_16[0][0]']
 Normalization)

  multi_head_attention_48   (Multi      (None,   1,   12)                52236
['layer_normalization_101[0][0]',
                                                              HeadAttention)
'layer_normalization_101[0][0]']

  dropout_75  (Dropout)                        (None,  1,  12)               0
['multi_head_attention_48[0][0]']

  tf.__operators__.add_96   (TFOpL      (None,   1,   12)                    0
['dropout_75[0][0]',
                                                                     ambda)
'input_16[0][0]']

  layer_normalization_102   (Layer      (None,   1,   12)                   24
['tf.__operators..... add_96[0][0]']
 Normalization)

  dense_128  (Dense)                           (None,  1,  64)             832
['layer_normalization_102[0][0]']

  dense_129  (Dense)                         (None,  1,  64)              4160
['dense_128[0][0]']

  dense_130  (Dense)                           (None,  1,  12)             780
['dense_129[0][0]']

  tf.__operators__.add_97   (TFOpL      (None,   1,   12)                    0
['dense_130[0][0]',
                                                                     ambda)
'tf.__operators__.add_96[0][0]']

  layer_normalization_103   (Layer      (None,   1,   12)                   24
['tf.__operators.....add_97[0][0]']
 Normalization)

  multi_head_attention_49   (Multi      (None,   1,   12)                52236
['layer_normalization_103[0][0]',
                                                              HeadAttention)
'layer_normalization_103[0][0]']

  dropout_76  (Dropout)                        (None,  1,  12)               0
['multi_head_attention_49[0][0]']

  tf.__operators__.add_98   (TFOpL      (None,   1,   12)                    0
['dropout_76[0][0]',
                                                                     ambda)
'tf.__operators__.add_97[0][0]']

  layer_normalization_104   (Layer      (None,   1,   12)                   24
['tf.__operators.....add_98[0][0]']
 Normalization)

  dense_131  (Dense)                           (None,  1,  64)             832
['layer_normalization_104[0][0]']

  dense_132  (Dense)                         (None,  1,  64)              4160
['dense_131[0][0]']

  dense_133 (Dense)                            (None,  1,  12)             780
['dense_132[0][0]']

  tf.__operators__.add_99   (TFOpL      (None,   1,   12)                    0
['dense_133[0][0]',
                                                                     ambda)
'tf.__operators__.add_98[0][0]']

  layer_normalization_105   (Layer      (None,   1,   12)                   24
['tf.__operators......add_99[0][0]']
 Normalization)

  multi_head_attention_50   (Multi      (None,   1,   12)                52236
['layer_normalization_105[0][0]',
                                                              HeadAttention)
'layer_normalization_105[0][0]']

  dropout_77  (Dropout)                        (None,  1,  12)               0
['multi_head_attention_50[0][0]']

  tf.__operators__.add_100   (TFOp      (None,   1,   12)                    0
['dropout_77[0][0]',
                                                                    Lambda)
'tf.__operators__.add_99[0][0]']

  layer_normalization_106   (Layer      (None,   1,   12)                   24
['tf.__operators.....add_100[0][0]'
 Normalization)                                                  ]

  dense_134  (Dense)                           (None,  1,  64)             832
['layer_normalization_106[0][0]']

  dense_135  (Dense)                         (None,  1,  64)              4160
['dense_134[0][0]']

  dense_136 (Dense)                            (None,  1,  12)             780
['dense_135[0][0]']

  tf.__operators__.add_101   (TFOp      (None,   1,   12)                    0
['dense_136[0][0]',
                                                                    Lambda)
'tf.__operators__.add_100[0][0]'
                                                                 ]

  layer_normalization_107   (Layer      (None,   1,   12)                   24
['tf.__operators.....add_101[0][0]'
 Normalization)                                                  ]

  multi_head_attention_51   (Multi      (None,   1,   12)                52236
['layer_normalization_107[0][0]',
                                                              HeadAttention)
'layer_normalization_107[0][0]']

  dropout_78  (Dropout)                        (None,  1,  12)               0
['multi_head_attention_51[0][0]']

  tf.__operators__.add_102   (TFOp      (None,   1,   12)                    0
['dropout_78[0][0]',
                                                                    Lambda)
'tf.__operators..... add_101[0][0]'
                                                                 ]

  layer_normalization_108   (Layer      (None,   1,   12)                   24
['tf.__operators.....add_102[0][0]'
 Normalization)                                                  ]

  dense_137  (Dense)                           (None,  1,  64)             832
['layer_normalization_108[0][0]']

  dense_138  (Dense)                         (None,  1,  64)              4160
['dense_137[0][0]']

  dense_139  (Dense)                           (None,  1,  12)             780
['dense_138[0][0]']

  tf.__operators__.add_103   (TFOp      (None,   1,   12)                    0
['dense_139[0][0]',
                                                                    Lambda)
'tf.__operators..... add_102[0][0]'
                                                                 ]

  dense_140  (Dense)                          (None,  1,  128)            1664
['tf.__operators.....add_103[0][0]'
                                                                 ]

  dense_141  (Dense)                          (None,  1,  1)               129
['dense_140[0][0]']

==============================================================================
==================
Total params: 234,017
Trainable params: 234,017
Non-trainable params: 0
______________________________________________________________________________
__________________
```

From the above summary of the model we can see that the total number of trainable parameters in the model are 234017. The mean absolute error achieved from the validation of the transformer network is around 1635 +- 100. Although the expectations were pretty high for this particular network, it does not outperform the random forest regression or decision tree regression[13][14]. We did some rough executions with increasing the model complexity but on increasing the model complexity further the model was overfitting.

The following graph shows the variation of training and validation laws with respect to epochs.

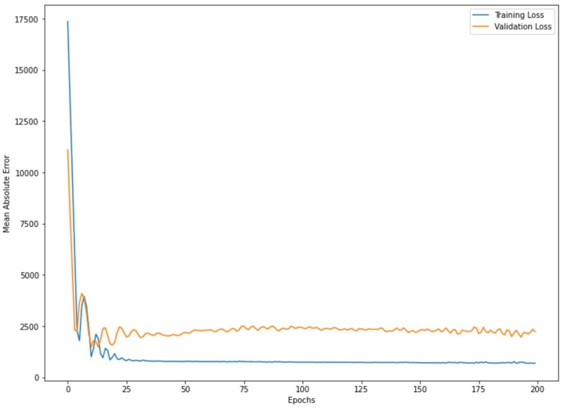


Figure 23: Change in training loss and validation loss with respect to number of Epochs

As mentioned in the legend of the graph, the blue line represents feeling glass and the red line represents the validation loss[11]. We can see that the loss saturates after some point of time and no more improvement is seen.

Now let's have a look at the prediction graph of the model where actual values and the predicted values are plotted on the same canvas, the blue line represents the actual values and the orange line represents the predicted values.

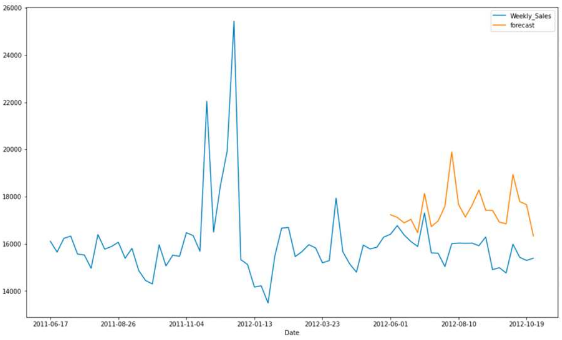


Figure 24: Prediction from transformer network model

In the above graph the X axis represents the weekly dates and the Y axis represents the weekly sale. It is evident from the graph that the patterns of the predicted and the actual sale are very close but still the error received from the model is not up to the mark.

## XI. Model Comparison

Before I start comparing the models we prepared in this project I need to mention that I have done some previous work on the same topic. Before we started with all the regression models mentioned in this report we did some work on SARIMA and LSTM models. The results achieved from our previous work can be seen in the following figure.

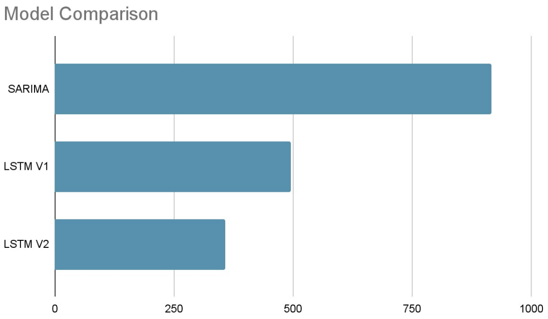


Figure 25:Figure showing their comparison of mean absolute error achieved in our previous attempts on the same problem.

From all the models we made for the forecast the best performance was received from random forest regression model and the worst performance was received by support vector regression model. In terms of error values the highest error is from the transformer network but, we are acknowledging the support vector regression model as the worst model because the production of the weekly sale from the support vector regression is a straight line so we can assume that it is not learning the patterns at all.

The following graph shows the mean absolute error received from each model, it also represents the best and the worst Mein absolute error received so far.

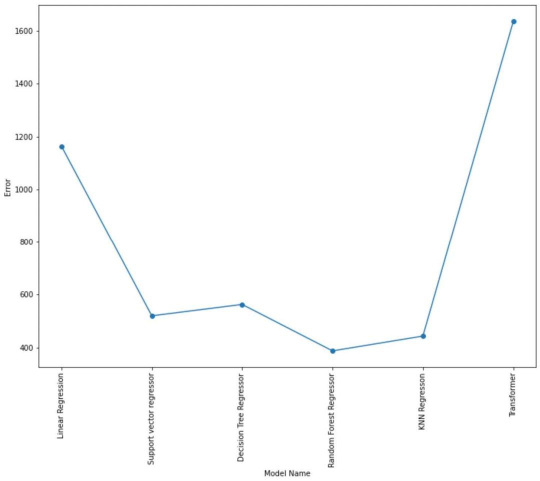


Figure 26: Comparison of mean absolute error achieved in each of the regression models we have made so far.

## XII. Future Scope

From the entire analysis we have seen that the best performance was received by random forest regression, but we are very certain that as the data points for training increase over the time the transformer network will prove to be a better model. We already know that the random forest and decision three models are very likely to be over-fitted and the support vector regression model is very bad for large data sets on top of that the K nearest neighbours regression is also very bad as the data size increases. So our expectations for the future work are that as the data size increases the best model will be the transformer network. In our knowledge we have tried to perform the best possible data preprocessing and exploratory data analysis but there is a very high chance that our work is not the best there is always a spot for upgrade. If someone can improve the explorer tree data analysis then there is a very high chance that the model performance will also increase in future.

## XIII. Acknowledgment

We are very thankful to our institution for giving us such excellent tools and possibilities, which have enabled us to complete a mission of this magnitude. Because of the amazing resources and possibilities provided by our institution, we were able to successfully complete our project. We owe our instructors a debt of gratitude for their assistance and guidance over this long period of time. Working on this project has taught us a lot as a group. In addition, we are thankful to our parents for their encouragement and supp