Walmart Sales Prediction

255 Data Mining - Team 15 - San Jose State University
Spring 2022 - Algorithm Track

Vamshidhar Reddy Parupally
Student ID: 016001427

Tirupati Venkata Sri Sai Rama Raju Penmatsa
Student ID : 016037047

***Abstract*—The aim is to model for predicting the sales for walmart using data from various outlets of the store. The data set consists of the sales of each store on different days of the year and the features of each data set is a collection of continuous and categorical data.**

- Github : https://github.com/vamshidhar199/255-FinalProject
- Dataset : https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting
- Colab for Vamshidhar's Contribution : **Link**
- Colab for Tirupathi's Contribution : **Link**
- State of art algoritm: LSTM: section **LSTM - Using Approach 1**

I. Introduction

One of the most essential components of strategic planning is predicting future sales for a company. This study examines how internal and external factors will effect Weekly Sales in the future for one of the largest firms in the United States. This project comprises a thorough data analysis, as well as time series analysis and sales forecasting using multiple models.

The data was collected between 2010 and 2012, and 45 Walmart locations across the country were examined. It's worth noting that we also have external statistics in the vicinity of each store, such as CPI, unemployment rate, and fuel prices, which should help us do a thorough research.

The focus of this paper aims to use the Walmart store data to comprehend sales trends by analyzing the data in unique and interesting ways. We devise and develop models which can aid in furthering the understanding of sales trends based on multiple factors. We incorporated regression based algorithms into our analysis in order to develop models which provided an additional layer complexity. This will be coupled with supplementary analysis, such as principal component analysis, and external research, in an attempt to extract information and extrapolate analysis. We hope that through our exploration we will see interesting patterns emerge, be able to make our own predictions, and shed some light on why these trends are occurring.

A. Work Contribution

*1) **Vamshidhar Reddy Parupally - Colab link for vamshidhar's work**:* The activities involved in the project have been divided equally among the team members and have described the activities performed by individuals in the tabular format, one for each team member. Details for Vamshidhar Reddy Parupally activities are mentioned in **table 1**

TABLE I
Vamshidhar Reddy Parupally's contribution

Module	***Contribution briefing***
II Data - A and B	Gathering data set from kaggle and merging data to single csv file to make the training process easy.
III Data Cleaning and Preprocessing - **Section A, Approach 1**	Has performed the preprocessing using **approach 1** including tasks [**Correlation, Null value analysis, Outlier analysis, average weekly sales analysis, holiday analysis**] as described in the respective section.
IV Experiment and Analysis	Analysis of the average sales and establishing the relation between sales and holiday weeks
IV Experiment and Analysis [section 1 to 4]	Developed **LSTM, Light GBM,ANN Model, Random Forest Regressor, and XG Boost** from **approach 1** data pre processing
V Comparing the models	Compared **LSTM, Light GBM, ANN,Random Forest, and XG Boost models**

*2) **Tirupati Venkata Sri Sai Rama Raju Penmatsa-Colab link for Tirupati's work**:* The activities involved in the project have been divided equally among the team members and have described the activities performed by individuals in the tabular format, one for each team member. Details for Tirupati Venkata Sri Sai Rama Raju Penmatsa activities are mentioned in **table 2**

II. Data

A. Dataset Explanation

Three different data sets from Kaggle.com about the company have been provided. These data sets contained information about the stores, departments, temperature, unemployment etc.

1) Stores: - Store: It indicates the store numbers ranging from 1–45.

- Type: Types of stores 'A', 'B' or 'C'.

- Size: Sets the size of a Store would be calculated by the no. of products available in the particular store ranging from 34,000 to 210,000

TABLE II
TIRUPATI VENKATA SRI SAI RAMA RAJU PENMATSA'S CONTRIBUTION

Module	***Contribution briefing***
II Data - A and B	Gathering data set from kaggle and merging data to single csv file to make the training process easy.
III Data Cleaning and Preprocessing - **Section b, Approach 2**	Has performed the preprocessing using **approach 2** including tasks [**Null value analysis, Anomaly Detection, Feature Engineering, One-Hot Encoding**] as described in the respective section.
IV Experiment and Analysis	Analysis of the average sales and establishing the relation between sales and holiday weeks
IV Experiment and Analysis [section 1 to 4]	Developed **ANN,Random Forest, Light GBM, Time series models like auto ARIMA and exponential smoothing, XG Boost and Decision Tree Regressor, Extra Tree Regressor** from approach 2 data pre processing and also used PCA for Dimensionality Reduction
V Comparing the models	Compared **ANN,Random Forest, Light GBM, XG Boost,Time series models like auto ARIMA and exponential smoothing, and Decision Tree Regressor, Extra Tree Regressor** , along with the above models chose the best performing models and compared them with and without PCA.

2) Sales: -Date: The date of the week where this observation was taken.

-WeeklySales: The sales recorded during that Week.

-Store: The store which observation in recorded 1–45.

-Dept: One of 1–99 that shows the department.

-IsHoliday: Boolean value representing a holiday week or not.

3) Features: Temperature: Temperature of the region during that week.

-FuelPrice: Fuel Price in that region during that week.

-MarkDown1:5 : Represents the Type of markdown and what quantity was available during that week.

-CPI: Consumer Price Index during that week.

-Unemployment: The unemployment rate during that week in the region of the store.

B. *Merging Data*

We have combine three files into one file for processing. Stores.csv and sales.csv files are combined on the basis of store attributes and its resultant file is merged with features.csv on the basis of attributes store, date and IsHoliday.

III. DATA CLEANING AND PREPROCESSING

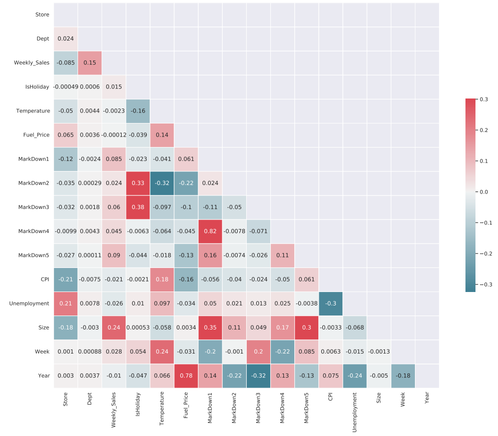
Preprocessing data is a data mining approach that entails converting raw data into a usable format. Real-world data is frequently inadequate, inconsistent, and/or lacking in specific behaviors or trends, as well as including numerous inaccuracies. Preprocessing data is a tried and true means of resolving such problems. Preprocessing raw data prepares it for subsequent processing.

A. *Approach 1*

*1) **Correlation**:* We start the data pre-processing by analysing the correlation among the features. Correlation is a bivariate analysis that measures the strength of association between two variables and the direction of the relationship. The heatmap gives us the correlation between the features and helps us with selecting relevant features.

From the correlation matrix we can find out that the Temperature, CPI, Unemployment and fuel prices have negative correlation with weekly sales. Also markdowns are not strongly correlated with weekly sales. From this observation we can conclude that markdowns can be dropped from the data set but before dropping them we have to make null value analysis. Refer Fig.1

Fig. 1. Correlation matrix.



*2) **Null value analysis**:* As we proceed with our data preprocessing we need to perform null value analysis as missing values or their replacement values can lead to huge errors in analysis output whether it is a machine learning model, KPIs or a report.

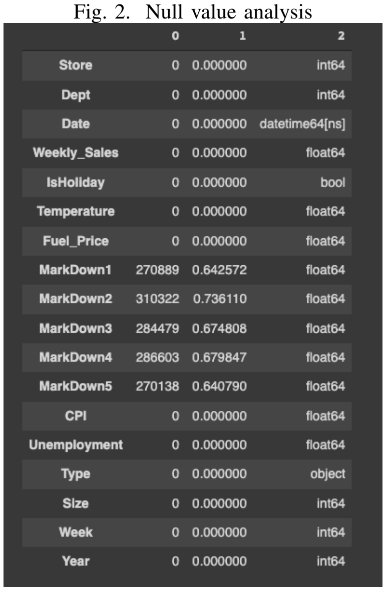
The below table outlines the details about the missing values in the data set. Refer Fig.2

It has been analysed that the markdown values have more than 60 percent of the values as null. So, they can be dropped, as replacing them with **mean or median** might impact the model efficiency and this is what has been chosen for approach 1 where we simply remove null values instead of imputing them with mean or median. Moreover they do not have a strong correlation with the weekly sales which is our target variable.

*3) **Outlier analysis**:* Outlier analysis is a type of data analysis that identifies anomalous findings in a dataset. This phase is essential if you want to draw significant conclusions from your data analysis. Fortunately, identifying outliers is a simple process. A box plot has been plotted to find out outliers if any in the data set. Refer Fig.4

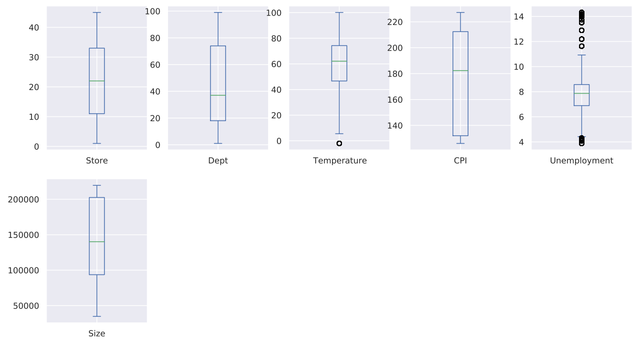
To further analyse the features temperature and unemployment, we calculate the quantiles by using the below code.

Fig. 2. Null value analysis



	0	1	2
Store	0	0.000000	int64
Dept	0	0.000000	int64
Date	0	0.000000	datetime64[ns]
Weekly_Sales	0	0.000000	float64
IsHoliday	0	0.000000	bool
Temperature	0	0.000000	float64
Fuel_Price	0	0.000000	float64
MarkDown1	270889	0.642572	float64
MarkDown2	310322	0.736110	float64
MarkDown3	284479	0.674808	float64
MarkDown4	286603	0.679847	float64
MarkDown5	270138	0.640790	float64
CPI	0	0.000000	float64
Unemployment	0	0.000000	float64
Type	0	0.000000	object
Size	0	0.000000	int64
Week	0	0.000000	int64
Year	0	0.000000	int64

Fig. 3. Box plot - outlier detection



```
uQ1 = train_detail['Unemployment'].quantile(0.25)
uQ3 = train_detail['Unemployment'].quantile(0.75)
IQR1 = uQ3 - uQ1


tQ1 = train_detail['Temperature'].quantile(0.25)
tQ3 = train_detail['Temperature'].quantile(0.75)
IQR2 = tQ3 - tQ1
```

The resultant quantiles have been used to detect how many outliers are present in the features and it has been found that the data is distributes approximately equally though box plot shows outliers. Hence the features can be left unchanged.

4) ***Holiday analysis:*** The data contains details for every friday and we essentially need week number and year so we calculate the week number and add week number and year to the data frame which will be used in further analysis.

```
train_detail.loc[(train_detail.Year==2010) &
(train_detail.Week==13),'IsHoliday'] = True

train_detail.loc[(train_detail.Year==2011) &
(train_detail.Week==16),'IsHoliday'] = True

train_detail.loc[(train_detail.Year==2012) &
(train_detail.Week==14),'IsHoliday'] = True
```

Plotted a graph with average sales per week for each of the year present in the data set and made IsHoliday field true for some of the holidays that have not been marked as true. Visualizations for the same could be found in the code.

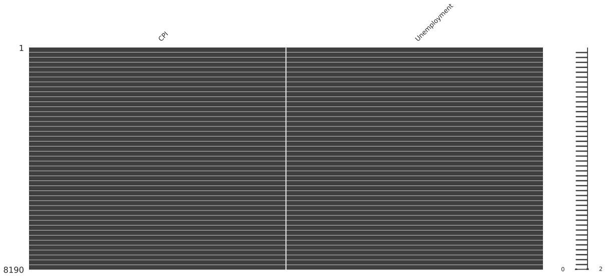
5) ***Label encoding:*** Label encoding is the process of translating labels into numeric format so that they may be read by machines. Machine learning algorithms can then help consider how those labels should be used. In supervised learning, it is a crucial pre-processing step for the structured dataset.

Encoded isHoliday and Type features, where isHoliday was converted from true, false to 0 and 1 respectively.

B. Approach 2

1) ***Null value analysis:*** Before beginning with other analyses, we need to check the null values which can help in getting a better understanding of the data we are working with. This will also will help us in having a better starting point for working on the data. So we can see that the fields that have null values are ”CPI”, ”Unemployment” and the ”MarkDown” fields. Going deeper into the missing values of ”CPI” and ”Unemployment”, we can find a pattern that there is a missing field of ”Unemployment” whenever null value occurs in ”CPI”. This kind of error is actually pretty common data entry error.So the loss of data would be pretty low when we remove null rows where ”CPI” and ”Unemployment”, which is just 7.14 percent of the data. ”MarkDown” fields have a large number of NA values which actually makes sense as there is a certain time period in the dataset which the store has not logged any data and same has been mentioned.So the null values of MarkDown fields have been made to 0 to prevent the huge 60 percent data loss from the other fields when using algorithms.

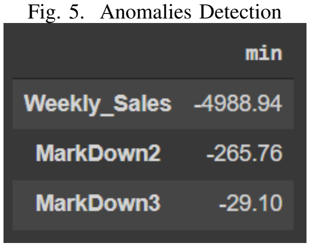
Fig. 4. msno - cpi and Unemployment null value occurrence



2) ***Anomaly Detection:*** Anomalies are referred to as data points that usually do not fit the expected pattern.

When going through the data we have found minimum values of ”Weekly Sales”, ”MarkDown2” and ”MarkDown3” to have negative values which are very odd, as Weekly Sales have to have at least a positive value and MarkDown values need to be either a positive value if markdown exists or zero if markdown does not exist. Thus such anomalies have been cleared to create a more representative dataset. Apart from this, outliers also have been handled using the quantile function.

Fig. 5. Anomalies Detection



	min
Weekly_Sales	-4988.94
MarkDown2	-265.76
MarkDown3	-29.10

*3) **Feature Engineering**:* Feature Engineering is the process of manipulating and extracting useful information from the raw dataset, It can greatly improve the performance of the model when done properly.

1) Date: Month,Week and Year have been extracted from ”Date” column to gain some extra insights into the data and also to use week for further models.

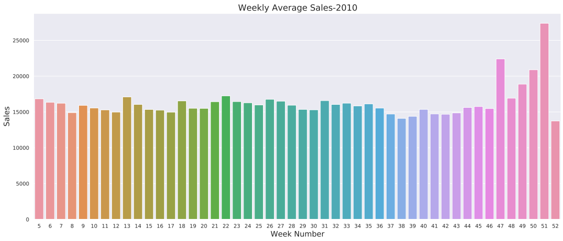
2) Statistical Data: Getting statistical values of Weekly Sales like ’max’,’mean’, ’median’, ’std’ and ’min’, which can be further used in the models and also to get some useful visualisations of the previous patterns.

*4) **One-Hot Encoding**:* One-Hot Encoding is a preprocessing method, to convert categorical columns in dataset into a numerical encoding that can be used to train the models. Advantage of One-Hot compared to Label Encoder is that, no particular category gets an unfair advantage due to numerical hierarchy. Although there is a ”Dimensionality-Curse” that is associated with One-Hot where in we get huge number of sparse columns get generated, this can be later addressed with some other methods but for now we get a good set of information that can be worked with.

IV. Experiment and analysis

In developing our experiments and analysis there were many models and visualizations that offered key and interesting insight about the data. These figures visualize and represent a broader range of modeling and help interpret our sales data. However, for now we will specifically highlight some examples which excited us. Refer Fig 6

Fig. 6. Avg Weekly Sales 2010.



The analysis of the visualizations revealed the increase in the sales during weeks with more pre-holiday days which means that the more number of days before a holiday, the more are the sales. Apart from this it has been realised that the certain weeks had higher sales even when the is holiday field is false, deeper analysis has revealed that those weeks had holidays and hence the necessary changes have been made to the data to accommodate it.

Fig. 7. Avg Weekly Sales 2011.

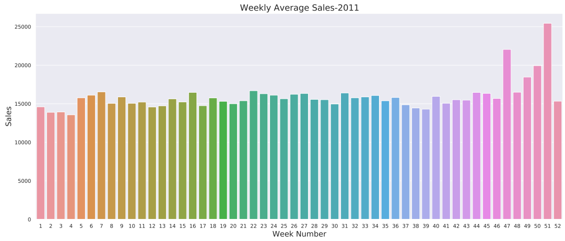
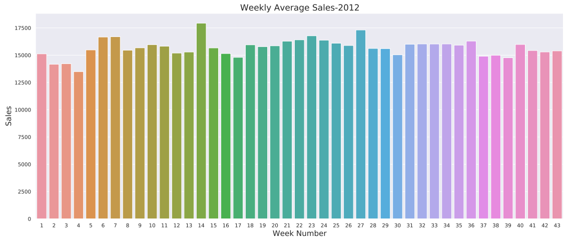


Fig. 8. Avg Weekly Sales 2012.



*1) **ANN Model**:* The severe simplicity of human neural networks is represented by ANN models. Artificial neurons, which are computational units that are comparable to the neurons of the biological nervous system, make up an ANN. In general, the ANN model is made up of three layers: input, hidden, and output.

Using Approach 1: This model has been used to develop a solution for the walmart sales prediction, and several configurations have been tried to come up with an accurate model. These configurations range from adding drop out, using different activation functions, and using kernel regularizers. Training the model with these configurations has yielded reduced loss for each epoch and reduced error, but when the square root of mean squared error has been calculated it turned out to be a higher number which is 25641.

Using Approach 2: This model has been used to develop a solution for the walmart sales prediction, and several configurations have been tried to come up with an accurate model.But when the square root of mean squared error has been calculated it turned out to be a higher number which is 7862.

*2) **Random Forest Regressor**:* While adjusting the hyperparameters of a single decision tree may result in modest gains, combining the results of numerous decision trees trained with slightly varying settings is a considerably more effective technique. A random forest model is what this is termed.

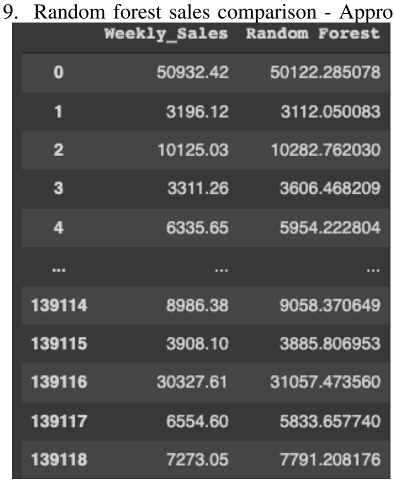
The fundamental concept here is that each decision tree in the forest will make different sorts of errors, and many of them will cancel out when they are averaged. The ”wisdom of the crowd” is another name for this concept.

Using Approach 1: Hypertuning the parameters can be easy with grid search cv, but owing to huge data it takes a lot of time to actually. So instead tested the model with different parameters and the best results have been observed for the following parameters, which yielded a weighted mean average error as 1684.87.

```
RandomForestRegressor(max_depth=27,max_features=6,
min_samples_split=4,n_estimators=58)
```

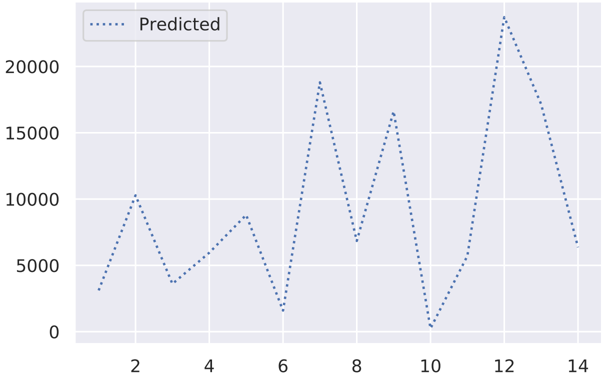
The actual vs the prediction values are plotted below and they tend to be very close in most of the instances.

Fig. 9. Random forest sales comparison - Approach 1



	Weekly_Sales	Random Forest
0	50932.42	50122.285078
1	3196.12	3112.050083
2	10125.03	10282.762030
3	3311.26	3606.468209
4	6335.65	5954.222804
...	...	...
139114	8986.38	9058.370649
139115	3908.10	3885.806953
139116	30327.61	31057.473560
139117	6554.60	5833.657740
139118	7273.05	7791.208176

Fig. 10. Random forest predicted sales - Approach 1



Using Approach 2: Initial results when used random forests RMSE values were pretty high almost close to 3681, but the values have been greatly reduced after the PCA with 5 components. Once PCA has been done, model has been trained again which has led to the values getting drastically reduced with final rmse value after PCA to 675.40

3) ***Light GBM*** *:* Light GBM is a gradient boosting framework based on the decision tree technique that may be used for ranking, classification, and a variety of other machine learning applications.

It splits the tree leaf wise with the best fit because it is based on decision tree algorithms, whereas other boosting methods split the tree depth wise or level wise rather than leaf wise.

Fig. 11. Random forest actual sales - Approach 1

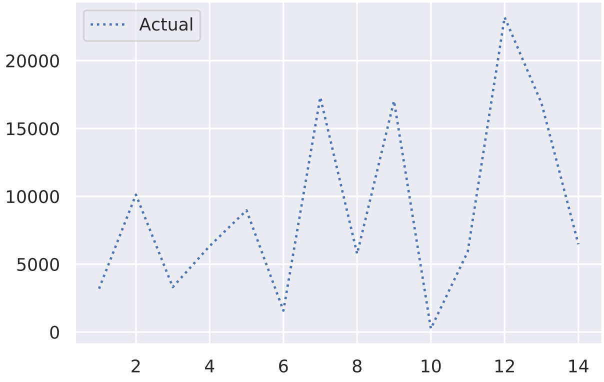
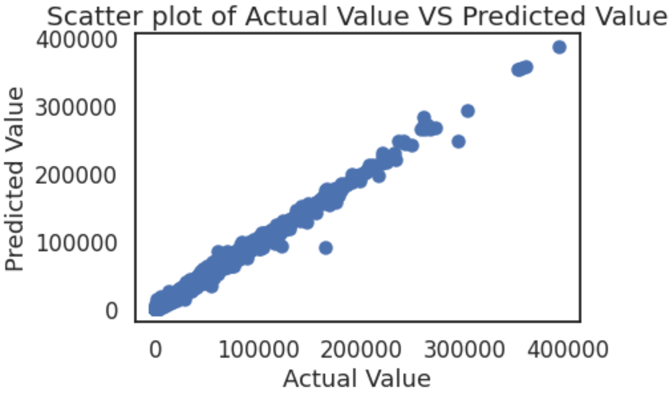


Fig. 12. Random forest Actual Sales vs Predicted Sales -Approach 2



As a result, when growing on the same leaf in Light GBM, the leaf-wise approach reduces more loss than the level-wise algorithm, resulting in substantially higher accuracy than any of the existing boosting strategies.

Using Approach 1: Feature selection can play key role in improving the accuracy of the model and hence in Light GBM the same has been tested with and without feature selection. The mean square error turned out to be 3685.59 before removing some features and 2491.73 after retaining important features and removing the rest. Feature importance is as shown below.

4) *LSTM - Using Approach 1:* **Several variants of the Long Short-Term Memory (LSTM) architecture for recurrent neural networks have been proposed since its inception in 1995. In recent years, these networks have become the state-of-the-art models for a variety of machine learning problems[from paper : https://arxiv.org/abs/1503.04069]**

The memory space is so small in RNN, the RNN problem is concatenated with it, resulting in a vanishing gradient problem. And then there's **LSTM**. The acronym LSTM stands for Long Short Term Memory networks, and it is a variant of RNN. This upgrade makes it easier for networks to remember previous data, which is critical in our problem-solving scenario.

Fig. 13. Feature Importance Light GBM - Approach 1

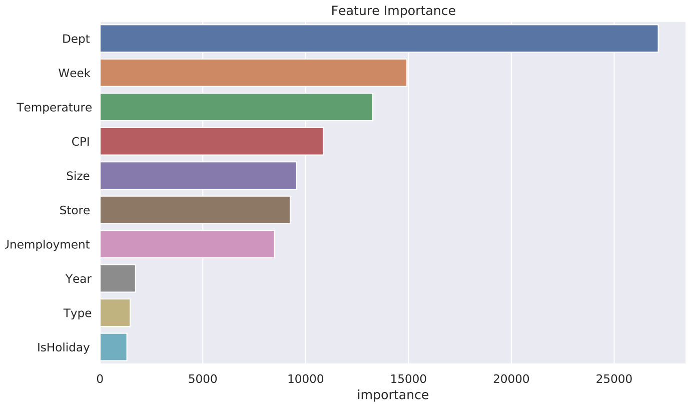
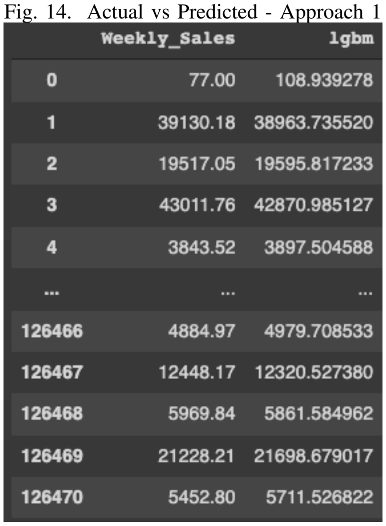
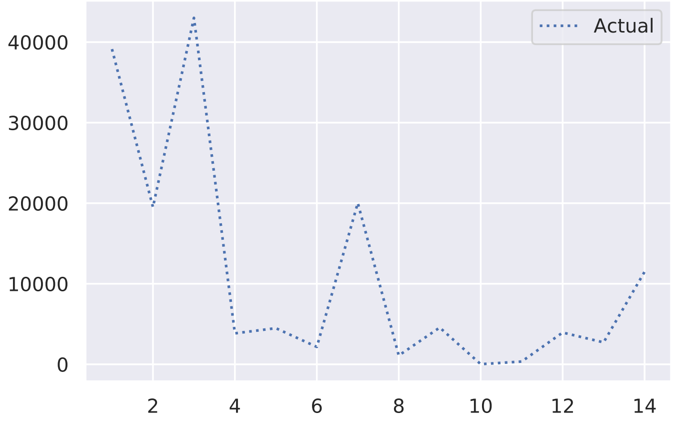


Fig. 14. Actual vs Predicted - Approach 1



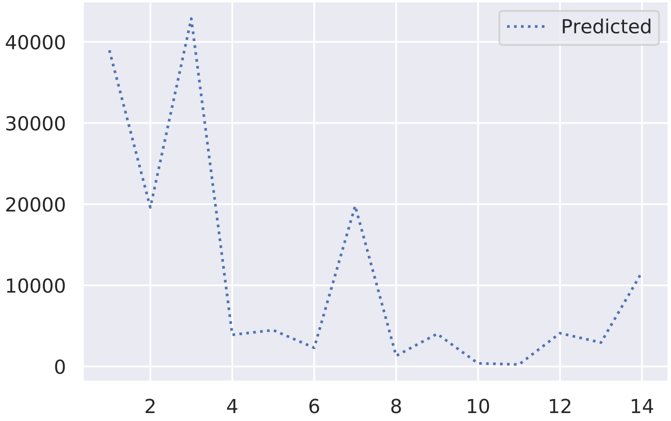
	Weekly_Sales	lgbm
0	77.00	108.939278
1	39130.18	38963.735520
2	19517.05	19595.817233
3	43011.76	42870.985127
4	3843.52	3897.504588
...	...	...
126466	4884.97	4979.708533
126467	12448.17	12320.527380
126468	5969.84	5861.584962
126469	21228.21	21698.679017
126470	5452.80	5711.526822

Fig. 15. LGBM Actual - Approach 1



RNN requires our data frame to be in time series format and a pivot table is formed for the same, where we have 150 features which is combination of store number and department number. Starting with simple input layers, where we have 128 neurons with `return_sequences` set as True which basically means pass all the weights and state to the next hidden layer and proceed with hidden layer with 64 neurons and finally output layer with 150 as we have 150 features. Then the predictions are made for each of the time step in the batch data. Then the model proceeds with the next batch. The performance of the model stood at a mean squared error of 4992, which is more than what we have for Random forest and hence can conclude that random forest performs better in this scenario.

Fig. 16. LGBM Predicted - Approach 1



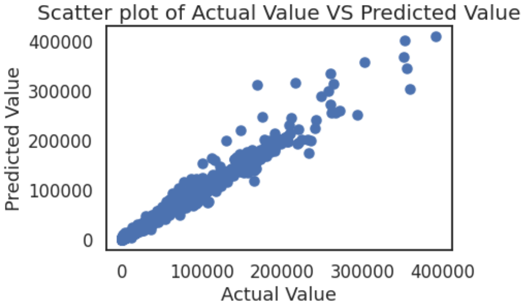


Fig. 18. LGBM Scatter - Approach 1

5) XG Boost: **Using Approach 1:** Extreme Gradient Boosting (XGBoost) is an open-source toolkit that implements the gradient boosting technique in an efficient and effective manner. The model has showed good predictions with an error of 6038.37

```
XGBRegressor(max_depth=12, n_estimators=20,
n_jobs=-1, random_state=48)
```

Using Approach 2: This model performed worst for the approach 2 apart from the ANN, with RMSE value of 5300.48 (before PCA) and RMSE value of 3666.31

*6) **Regression Trees**:* Couple of Regression trees have been tested namely extra-tree and decision tree regressors and found out that the extra tree performed the best among these two

Fig. 19. LSTM actual vs predictions - Approach 1

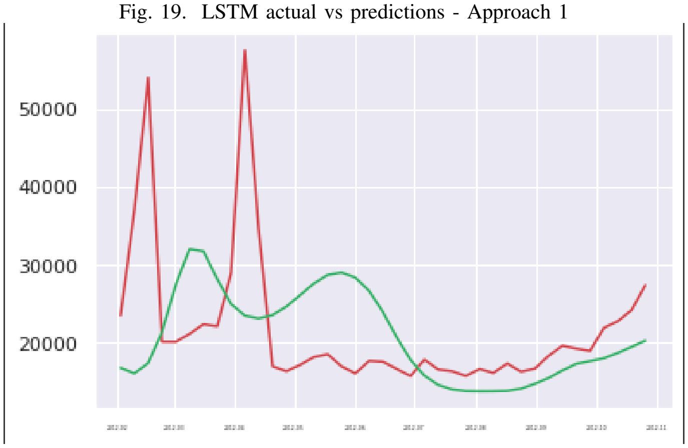


Fig. 20. XG Boost - sales prediction - Approach 1

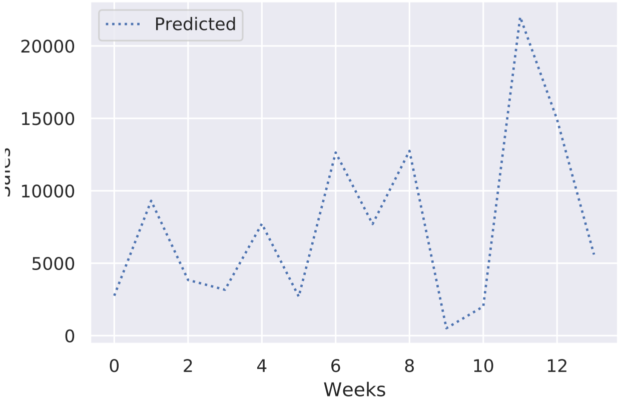
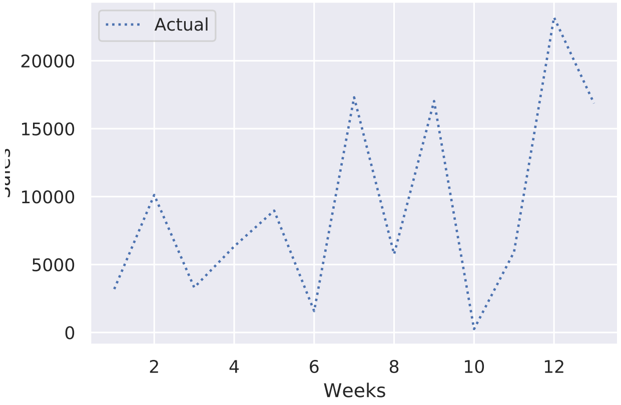


Fig. 21. XG Boost - actual sales - Approach 1



with RMSE of 4082.83 compared to decision tree regressor's RMSE value of 5485.82

But after the application of PCA it outperformed even the Random Forest Regressor, with RMSE value of just 485.54 compared to Random Forest Regressor's RMSE value of 675.48

Fig. 22. XG Boost - actual sales - Approach 2

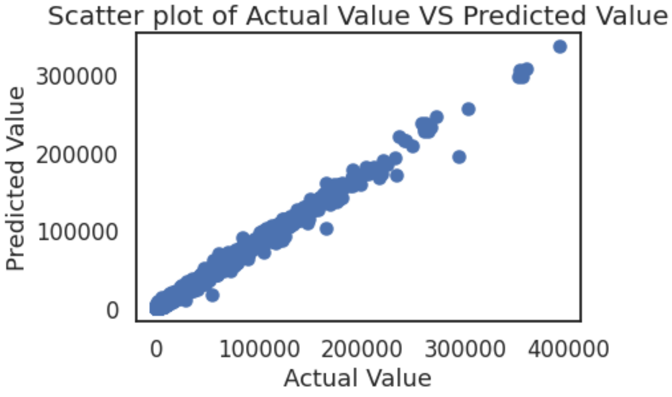


Fig. 23. Decision Tree Scatter Plot

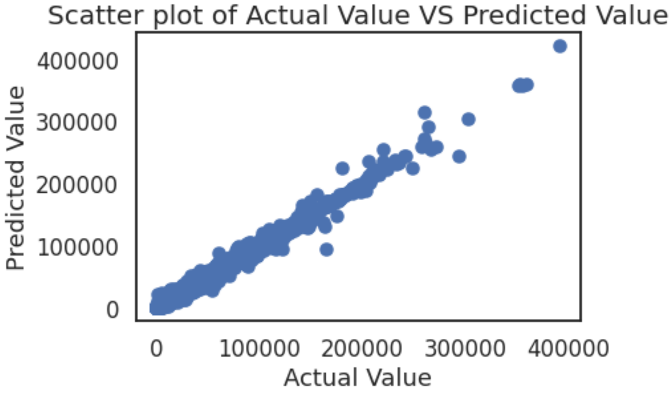
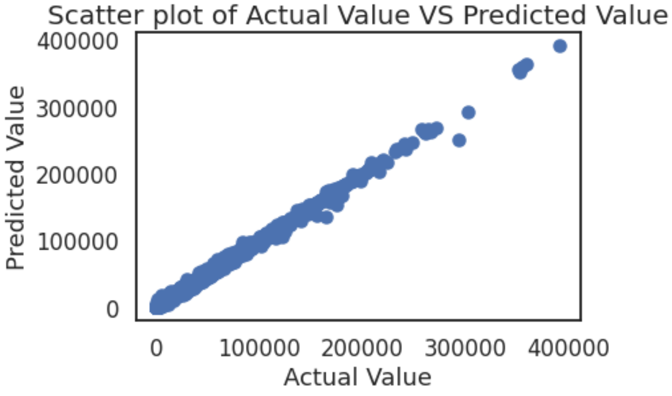


Fig. 24. Extra tree Scatter plot

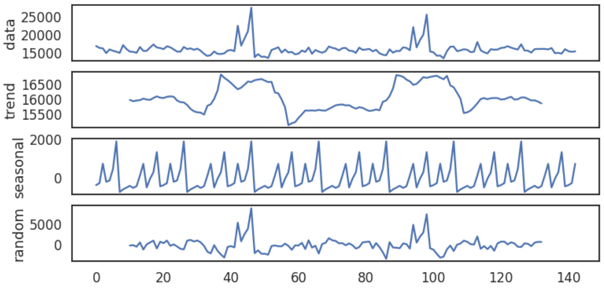


*7) **Approach 2-Time Series Models:*** **Auto ARIMA model** ARIMA model is one of the go to models for forecasting the sales and using time series data, in ARIMA model we need to give out 'p','d' and 'q' values and statistical techniques are used to generate those values. Whereas Auto ARIMA model itself generates the optimal 'p','d' and 'q' values for the given data set.

To analyse the patterns we have decomposed the data to analyze the patterns of WeeklySales and found out that the trends are pretty similar which seems promising to apply Time

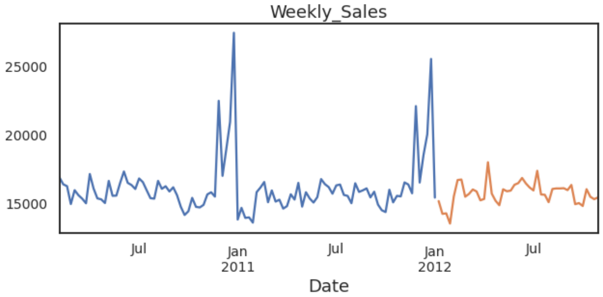
Series models.

Fig. 25. Trends and Decomposition



Since we need a series of data and it needs to be forecasted we haven't taken a random split of train test.

Fig. 26. Train Test Split



Also we analyzed different time series trends like rolling means and rolling standard deviation, to check which pattern had a smoother pattern and also without abnormal spikes, after plotting the data, it was quite evident that weekly difference was the best.Using this on the Approach 2 gave a decent results with RMSE value of 1681.42

Fig. 27. Deviations of different data

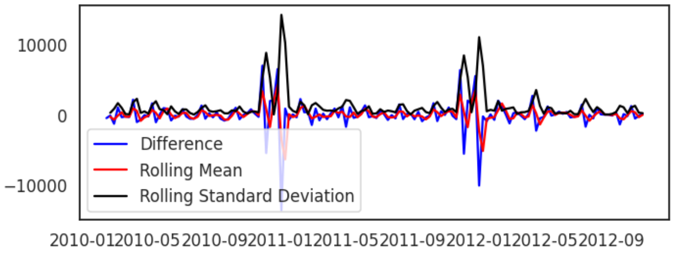
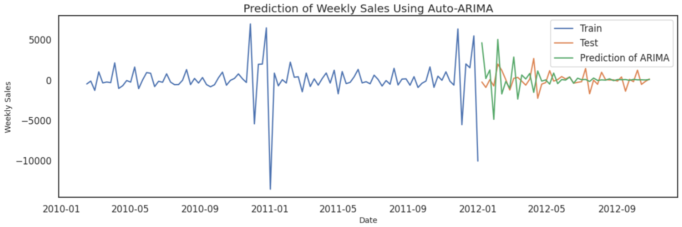


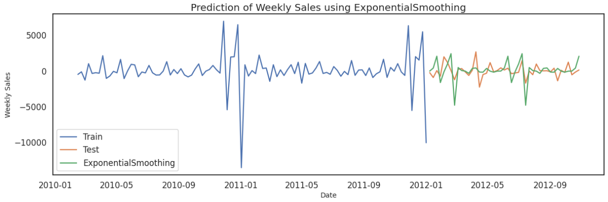
Fig. 28. Forecasting using Auto-ARIMA



Exponential Smoothing Exponential Smoothing is one of popular methods to ssmooth the time series data by applying weighted average in an exponentially decreasing manner with respect to time to give the recent data higher weightage than the older data. This actually makes sense in the real-world as the trends keep on changing and most probably the future data is most correlated with the recent trends than the old trends.

When this model has been applied we got much more better results than the Auto-ARIMA, with RMSE value of 1271.59

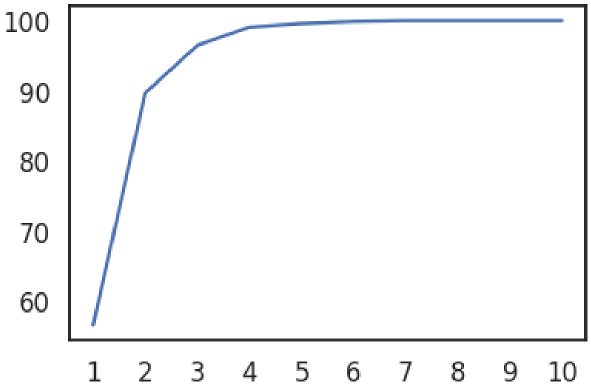
Fig. 29. Exponential Smoothing



*8) **PCA:*** Principal Component Analysis is a dimensionality reduction technique to reduce the number of features but preserving the information as much as possible. This technique has been crucial for Approach 2 as the results before and after PCA have been discussed.

In order to retain maximum information, explained variance ration plot has been plotted and seen that n=5 preserves the most information.

Fig. 30. Selecting Components for PCA



V. Comparison

The comparison of the models clearly reveals that the simpler models like random forest regressor out performs other model like artificial neural network, Light GBM, XG Boost and LSTM.

Also dimensionality reduction techniques like PCA has greatly helped the models in the forecasting with RMSE value improvement of atleast 150 percent.

Considering the fact that not much of data available, Time Series models like auto-ARIMA and Exponential smoothing have performed very well compared to the Decision tree models and Regression models before PCA.

Fig. 31. Comparision Approach - 1

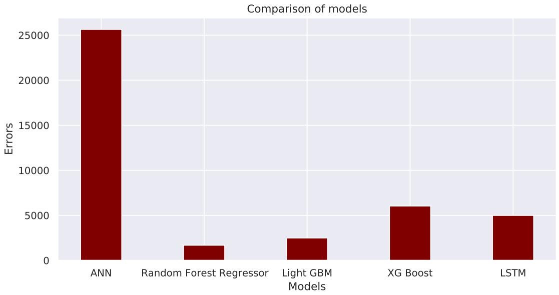


Fig. 32. Comparision Approach - 2

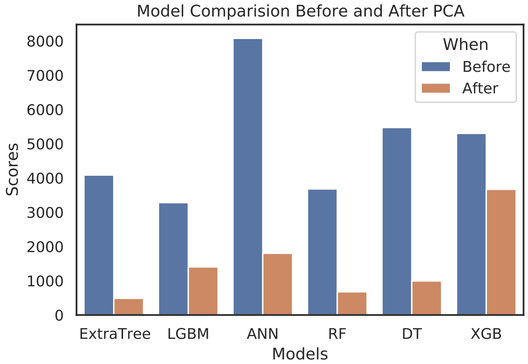


TABLE III
COMPARING MODELS APPROACH 1

Model	***Root Mean Square Error***
ANN	25641
Random Forest Regressor	1684.87
Light GBM	2491.73
LSTM	4992
XG Boost	6038.37

TABLE IV
COMPARING MODELS APPROACH 2

Model	***RMSE(Before PCA)***	***RMSE(After PCA)***
ANN	7862	1856
Random Forest Regressor	3680.52	675.48
Decision Tree Regressor	5485.82	992.27
Extra Tree Regressor	4082.83	485.54
Light GBM	3253.49	1401.12
XG Boost	5300.48	3666.31

TABLE V
COMPARING MODELS

Model	***Root Mean Square Error***
Auto-ARIMA	1684.87
Exponential Smoothing	1271.59

REFERENCES

[1] https://www.kaggle.com/code/bhatnagardaksh/walmart-sales-predictionRandom-Forest
[2] https://medium.datadriveninvestor.com/walmart-sales-data-analysis-sales-prediction-using-multiple-linear-regression-in-r-programming-adb14afd56fb
[3] https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting/code
[4] https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting
[5] https://www.kaggle.com/code/datamany/random-forest-rnn-walmart-sales-forecast
[6] https://machinelearningmastery.com/xgboost-for-regression/
[7] https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/
[8] https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem
[9] https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/
[10] https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/
[11] https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314
[12] https://towardsdatascience.com/time-series-data-analysis-resample-1ff2224edec9
[13] https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775/
[14] https://towardsdatascience.com/efficient-time-series-using-pythons-pmdarima-library-f6825407b7f0/
[15] https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/
[16] https://arxiv.org/abs/1503.04069