STAT 429 Project
Sales forecast of Walmart weekly sales

Group 12

Team Members:

1- Konduru, Chiranjeevi (Net ID: konduru4)
2- Kumar, Pallaw (Net ID: pallawk2)
3- Hashmi, Muhammad Abdullah (Net ID: mhashmi2)

Abstract

Sales Forecasting is essential for retail giants like Walmart. This report discusses the analysis of multiple approaches performed to forecast the weekly sales of Walmart stores. Time series models serve to accurately predict a variable in the future, based on time series data. The initial approach was done by suggesting a SARIMA model based on the analysis of time series data. Several models were tried and tested to achieve the best predictions and the forecast accuracy was evaluated based on a few metrics that were discussed in class. The second approach was done using a classic tree-based Machine Learning algorithm called Random Forest regressor using the different features that are both categorical and continuous in nature. Comparing the forecast accuracy, we conclude that the Random Forest Regressor gave us the least absolute error, in other terms, this model gave us the best predictions and had a better forecasting ability as compared to the SARIMA models.

Introduction

Sales is the major revenue-generating contributor for retail super giants like Walmart. Having an understanding of consumer demand ahead of time will help the company generate more revenue in the future. Our project aims to provide the company with a few data-driven approaches that can be used to focus more on the departments and stores in different regions of the country. The project primarily concerns the forecast of Walmart weekly sales using historical data, provided by Walmart themselves.

The data set consists of sales data from 45 stores and includes sales information as well as other information associated with the individual stores. The data is segregated on a weekly interval. Furthermore, the data set also includes holiday periods which have been flagged accordingly. This serves as an additional data point, through which we can ascertain whether the store sales increase or decrease according to the holiday period, and to find out the effect of the said holiday period.

Goal

The goal of this project is to find out the weekly sales. This can help a company, Walmart in this case, to then keep a suitable level of inventory in stock which can make their business profitable. As the dataset includes size and time-related data as attributes, we'll examine whether or not these elements affect sales. Most significantly, how do holidays affect a week's worth of store sales.

Statistical models rely on historical data to find out the underlying patterns, and to predict a target variable in the future based on those underlying patterns. Time series models are a special type of statistical model, which use temporal data to find out the underlying correlations and relations between different variables and time. Hence, we have approached this problem statement with both statistical and tree-based ML models.

Dataset

The dataset that we are using is the Walmart Store Sales Forecasting data. This has been provided by Walmart for a Kaggle competition and is a very widely used data set for time series analysis. The original data set has a train and test set, but for this project, we will only be using the train set, and then splitting that into train and test sets, since we do not have the target variable in the original test dataset to evaluate our model on.

While a detailed data dictionary is provided in the appendix, let's have a brief overview of the data structure. The data provided was initially segregated into 3 files namely, train_df, stores_df, and features_df. Each file had important features that should be looked upon while constructing a model. A brief overview of these data is given below. The training dataset consists of 421,570 records, from the period 2010-02-05 to 2012-11-01, and includes Store, Department number, Date (week), Weekly Sales, and a Boolean variable that denotes whether a week is a holiday week or not. A snapshot of the train data set is as below:

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	2010-02-05	24924.50	False
1	1	1	2010-02-12	46039.49	True
2	1	1	2010-02-19	41595.55	False
3	1	1	2010-02-26	19403.54	False

	Store	Type	Size
0	1	A	151315
1	2	A	202307
2	3	B	37392
3	4	A	205863

Furthermore, a stores dataset (2nd image above) is also provided, which contains data about the store itself, and it is anonymized. It contains information about the type of store and also the size of the store. Finally, more data is provided with additional features that can be used to predict weekly sales. This includes information on economic factors such as the average region temperature, fuel price, CPI index (inflation), and unemployment rate. Below is a snapshot:

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	2010-02-05	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
1	1	2010-02-12	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106	True
2	1	2010-02-19	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106	False
3	1	2010-02-26	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106	False

MarkDown1-5 is anonymized data related to promotional markdowns currently running on Walmart. It is only available after Nov 2011 and is not available in all stores all the time. All the above three datasets were merged into a single data frame based on the store id column. This helped us to perform the data

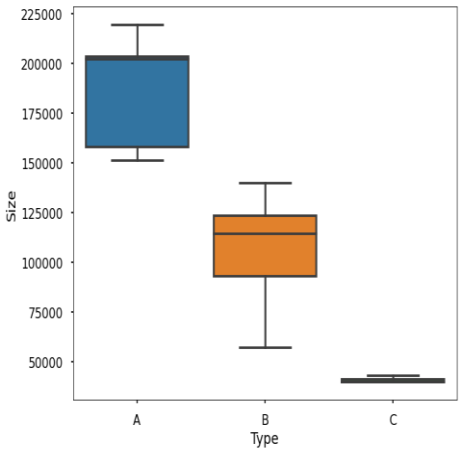
preprocessing and model training. Redundant columns were dropped from the dataset. EDA has been performed to gain insights from various features that contribute to predicting weekly sales.

Exploratory Data Analysis

The dataset consisted of 15 variables which were continuous and categorical in nature. There were negative and NaNs in the weekly sales column, which didn't make any sense as sales cannot be negative. Since the target variable is 'weekly_sales', we dropped the 1360 rows of data which was 0.3 % of the entire dataset. The holiday weeks were identified from the data description given in the Kaggle competition description. The plot below shows that the weekly holiday sales are marginally higher compared to non-holiday weeks.

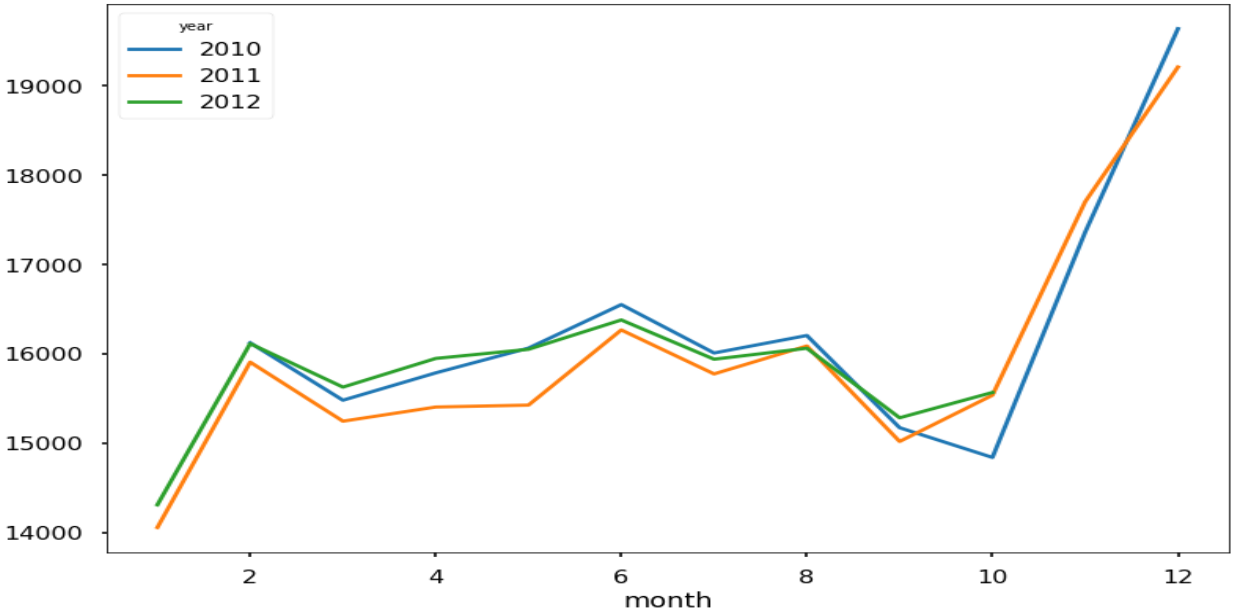
Out of total 143 weeks in the data, 133 weeks were non—holiday weeks, and only 10 weeks were holiday weeks. From the 2nd and 3rd snapshots below, the sales during black Friday are significant compared to the actual Christmas week. There are 3 Different types of Walmart stores namely A, B, and C. The distribution of weekly average sales during the holiday weeks from the different store types is given below. The red line represents the holiday avg. and the green represents the non-holiday avg. sales.

It is seen from the graph that; the highest sale average is in the Thanksgiving week between holidays. And, for all holidays Type, A stores have the highest sales. Next, we proceeded by looking at the impact of 'store size' on the sales for the specific store type. The box plot below indicates that the bigger the size of the store, the higher the sales in it. According to this graph, Walmart categorizes stores by size. B begins after the smallest size value of Type A. As soon as Type B reaches its smallest size value, Type C begins.

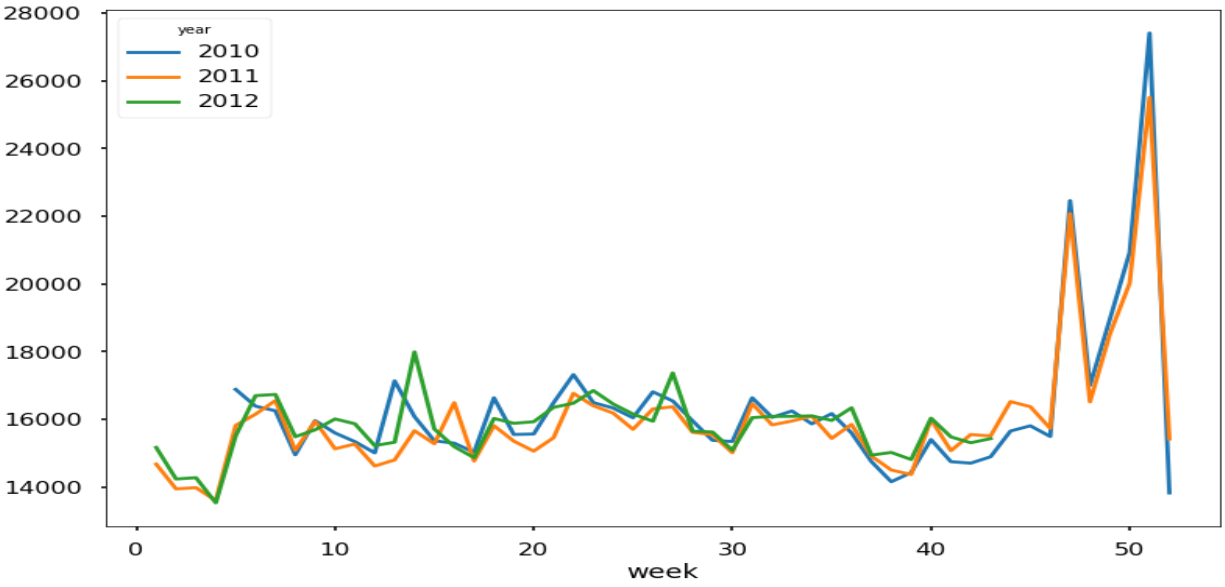


In the graph below, we can see that sales in 2011 are generally lower than in 2010. As we look at the mean sales, we see that 2010 has higher values, but 2012 has no information about November and December, which have higher values in general. Although 2012 hasn't had any sales for

the last two months, its mean is close to 2010. Most probably, it will take first place if we get 2012 results and add them.



Since we are considering the weekly, the above graph couldn't tell us anything about the holiday weeks discussed previously. Grouping the sales by holiday week, we can see that the top sales are a couple of weeks before Christmas, Thanksgiving, and Black Friday. The Weekly breakdown of sales is as follows:



As a whole, the dataset does not have too many missing or erroneous values and needs minimal preprocessing to be used in models. Since we are trying a machine learning model, the variables should be independent of each other, so it won't cause any redundancy while fitting the model. We plotted a correlation plot, to identify the variables that are linearly correlated (shown in Analysis C). As we can see,

the week and month are correlated since they both tell us the period, and also, they both are related i.e., 4 weeks = 1 month.

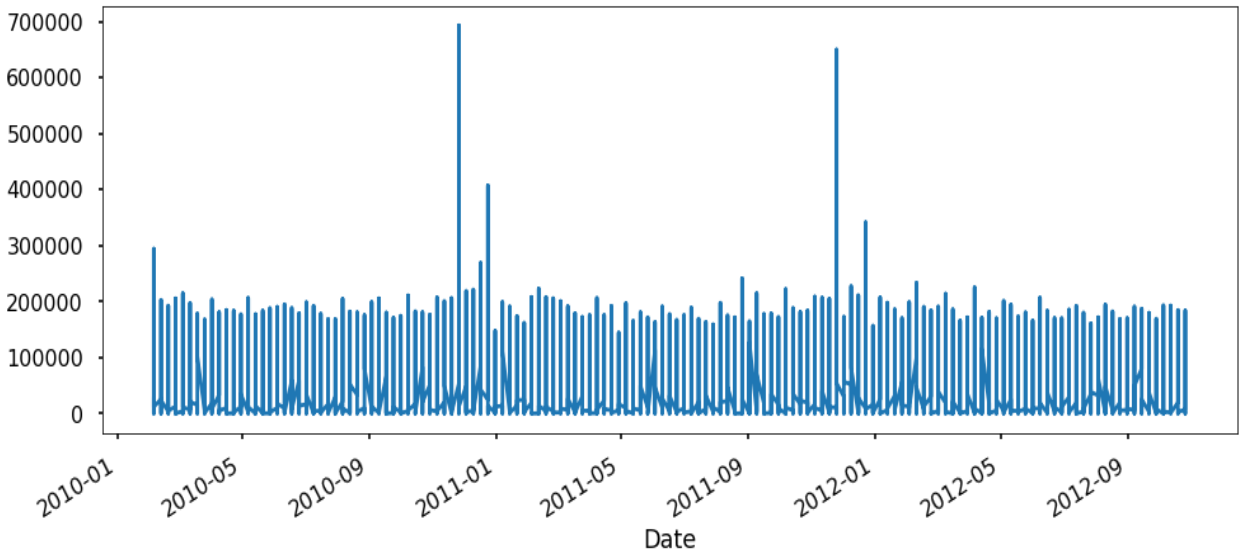
Temperature, unemployment, and CPI have no significant effect on the target variable 'weekly_sales' and were dropped. Also, Markdowns 4 and 5 highly correlated with Markdown 1. We can see those weekly sales are correlated with the store type and size of the store. We also tried applying log transformation to the data, however, there wasn't a significant improvement in the forecast of weekly sales.

Statistical Methods:

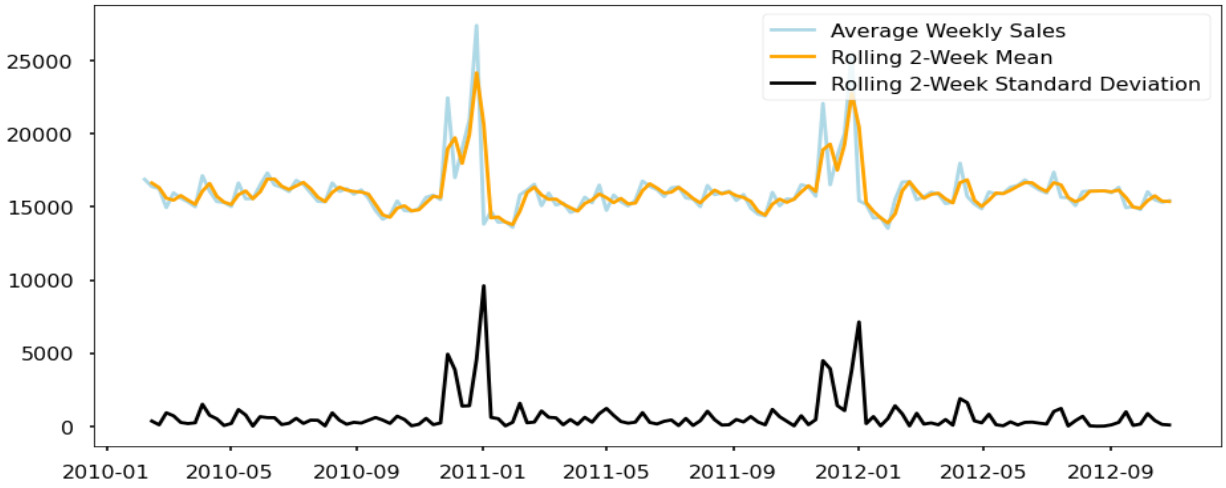
Analysis B - SARIMA model

The data consisted of weekly sales at the department level of multiple Walmart stores in several regions. Since the data covered 2 years of weekly sales, we have tried plotting the sales graphs on a weekly, monthly basis for each year from 2010 to check for trend and seasonality in it. We came up with two approaches for suggesting a SARIMA Model for the given time series.

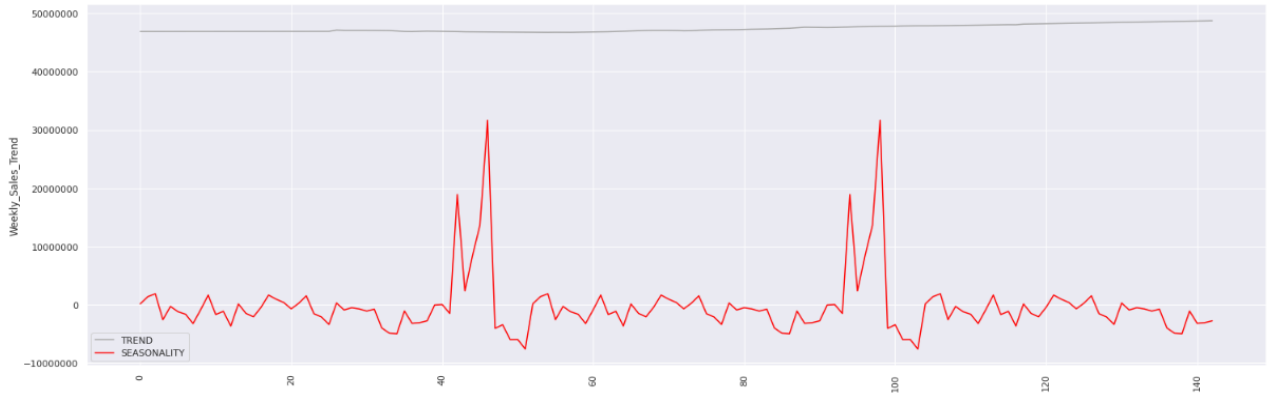
The below plot shows the sales of Walmart stores that are grouped for each week in the year. We can observe that the two spikes here represent the sales during the holiday season of the year. This indicates that there is seasonality in the data.



When plotting the time series data, the fluctuations we see above prevented us from clearly gaining insights about the peaks and troughs in the plot. So, to get value from the data, we used the rolling average concept to make the time series plot. This was done by calculating the rolling average by summing up the previous 'sales values' and dividing them by 'sales' itself. It helped in getting a clearer picture of the trend in the data. The below plot is constructed by using a rolling 2-week rolling mean and standard deviation of the train data, we can see that the data is not stationary as the mean is not constant and the variance is also not stationary.



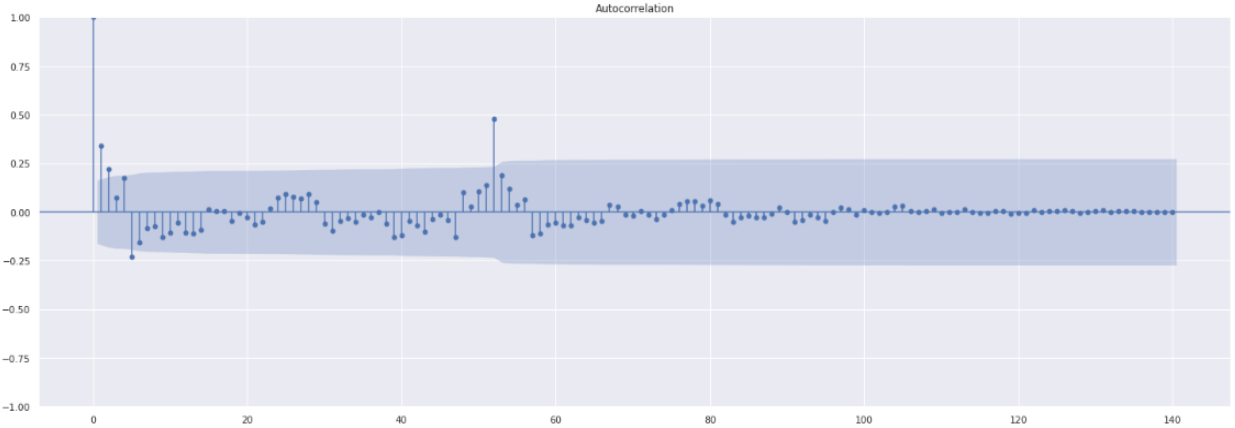
To further analyze the data before plotting the ACF and PACF plots, we decomposed the time series data to obtain the trend and seasonality separately in the image below.

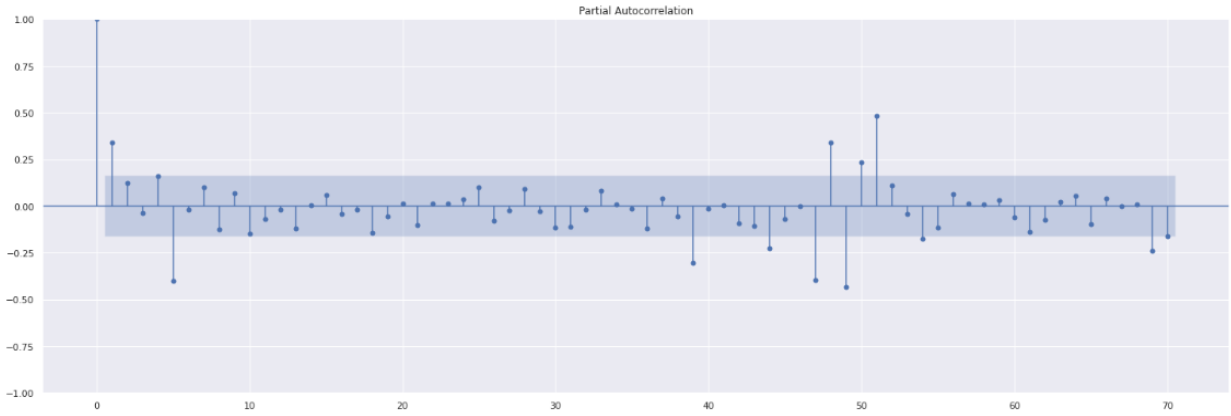


As we can see clearly, the trend (light gray) is linearly increasing over time and there is significant seasonality at a specific time of each consecutive year.

Approach 1: ACF and PACF

The first approach we tried was to suggest a SARIMA model by plotting the ACF and PACF plots. The below plot shows the ACF and PACF of the original train data.

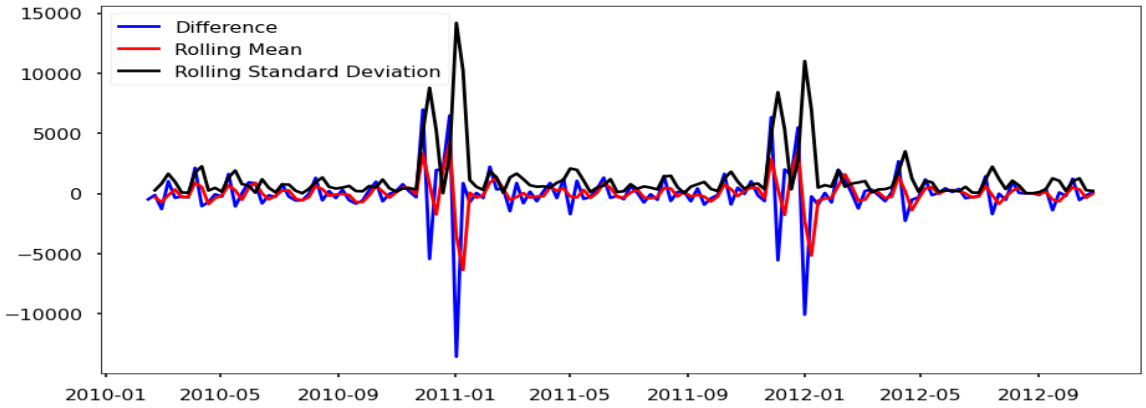




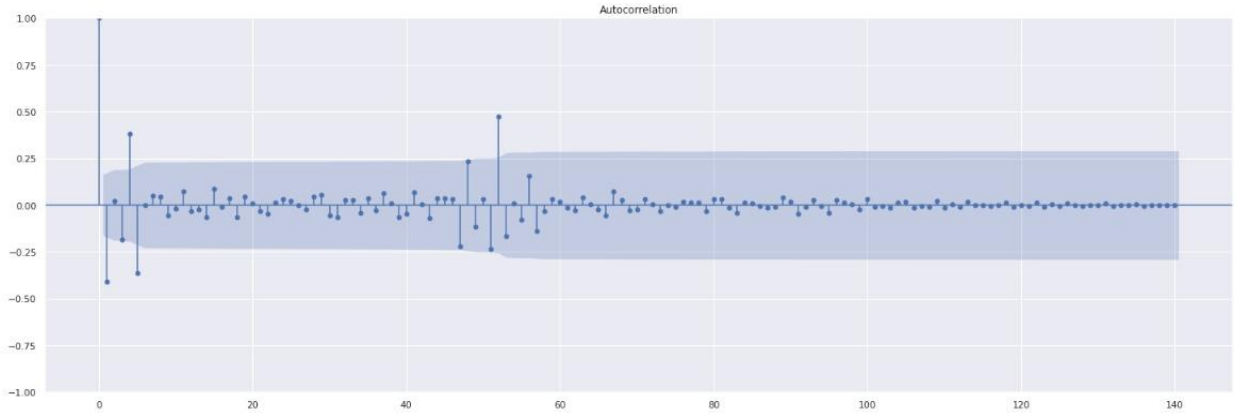
Observations:

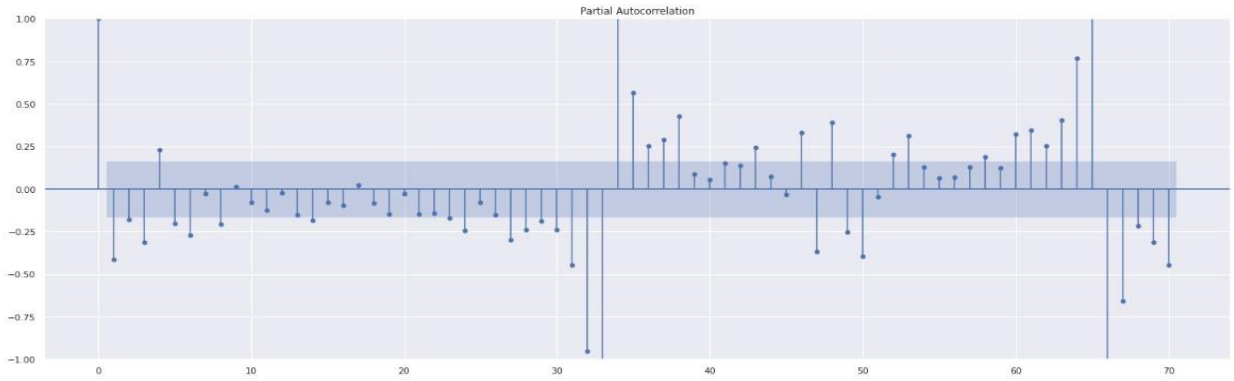
- **Non-Seasonal**: ACF is tailing off after lag 2 and PACF is tailing off after lag 1.
- **Seasonal**: At lag k = 52 we can see a spike in both ACF and PACF plots.

This indicates that there is seasonality and trend in the dataset. Performing the first order of differencing made the data stationary. From the plot below we can observe that the rolling mean and standard deviation are constant (near 0).



Proceeding with ACF and PACF plots, we could observe that noise is better removed which helped in suggesting a couple of SARIMA Models for forecasting.





Observations:

- **Non-Seasonal**: ACF is cut off after a few lags and PACF is tailing off.
- **Seasonal**: At lag k = 52 we can see a spike in ACF, but the PACF plot is tailing off.

From these observations, the following SARIMA Models could be suggested:

1) SARIMA (0,1,1) x $(1,1,0)_{52}$

This model gave us reasonable AIC and BIC values. We got to this model as we observed from the ACF plot the series is cutting off after lag 1 and PACF is tailing off. The Summary statistics of the model can be found below.

SARIMAX Results

Dep. Variable:	Weekly_Sales	No. Observations:	99
Model:	SARIMAX(0, 0, 1)x(1, 0, [], 52)	Log Likelihood	-845.073
Date:	Mon, 05 Dec 2022	AIC	1696.145
Time:	18:57:53	BIC	1703.931
Sample:	02-14-2010 - 01-01-2012	HQIC	1699.295
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.9932	0.008	-122.513	0.000	-1.009	-0.977
ar.S.L52	0.9458	0.009	109.581	0.000	0.929	0.963
sigma2	4.58e+05	6.67e+04	6.871	0.000	3.27e+05	5.89e+05

Note: Since the Data passed was already differenced, we gave the order as 0 while fitting the model.

2) SARIMA (0,1,5) x $(1,1,0)_{52}$

This model looks slightly better than the previous model in terms of the AIC. However, when we look into the p-values in the summary statistics below, it suggests that the coefficient for MA(2) is not statistically significant. Hence this model can be rejected.

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                      Weekly_Sales   No. Observations:                   99
Model:             SARIMAX(0, 0, 5)x(1, 0, [], 52)   Log Likelihood                -836.521
Date:                             Tue, 06 Dec 2022   AIC                           1687.042
Time:                                     00:08:50   BIC                           1705.208
Sample:                                 02-14-2010   HQIC                          1694.392
                                      - 01-01-2012
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.6449      0.086     -7.468      0.000      -0.814      -0.476
ma.L2         -0.0688      0.132     -0.521      0.603      -0.328       0.190
ma.L3         -0.1841      0.095     -1.937      0.053      -0.371       0.002
ma.L4          0.2147      0.081      2.643      0.008       0.056       0.374
ma.L5         -0.3152      0.088     -3.573      0.000      -0.488      -0.142
ar.S.L52       0.9272      0.012     76.820      0.000       0.904       0.951
sigma2      4.524e+05   6.24e+04      7.255      0.000     3.3e+05    5.75e+05
==============================================================================
```

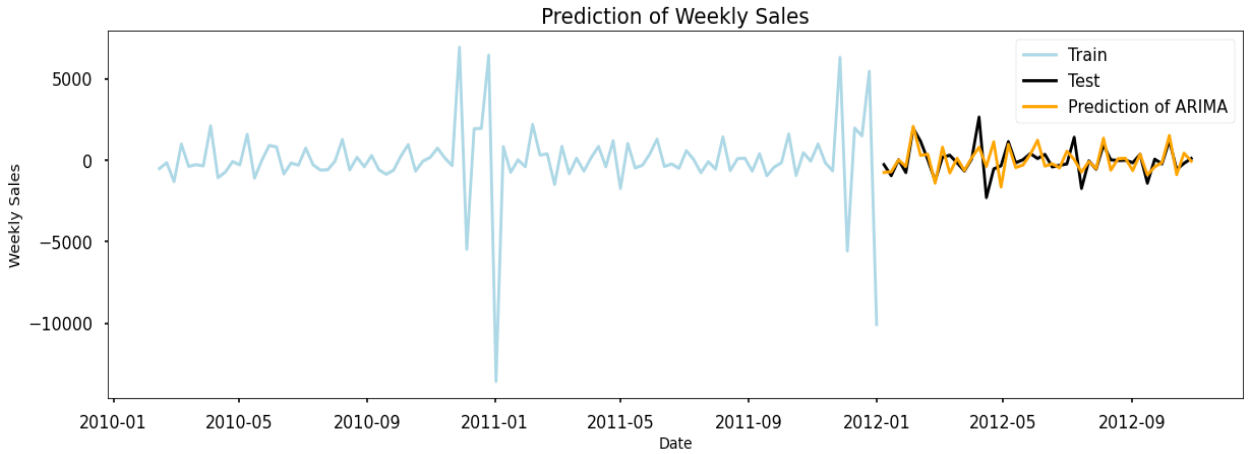
Approach 2:

The Models were constructed by identifying the optimal values for parameters: p, d, q, P, D, Q that gives the least AIC and BIC values. First, we use a loop and try out various models using a range of values for the parameters. This is similar to the grid search approach used on machine learning algorithms for performing hyperparameter tuning. The ranges of these parameters were given by looking at the possible lags from the ACF and PACF plots.

The result below shows the top 5 SARIMA models out of 81 combinations that were tried and were filtered using least AIC and BIC as criteria.

	AR(p)	Diff(d)	MA(q)	SAR(P)	SDiff(D)	SMA(Q)	Seas(S)	AIC	BIC
12	0	1	1	1	1	0	52	1696.145174	1703.930534
26	0	1	2	2	1	2	52	1697.124786	1715.290625
13	0	1	1	1	1	1	52	1698.145166	1708.525646
15	0	1	1	2	1	0	52	1698.145702	1708.526182
41	1	1	1	1	1	2	52	1699.774471	1715.345190

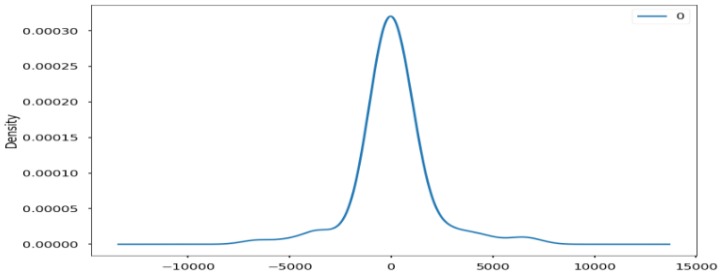
The above table shows that model -1 from the first approach is the top best model followed by the next set of models. Hence, we proceeded with **SARIMA (0,1,1) x $(1,1,0)_{52}$** model as our final model for forecasting. The Below plot shows the sales forecast for the next 43 weeks by this model.



Model Evaluation:

The Model performed considerably well in forecasting the next observations. While there is decent predictive ability in our model, there is still some divergence. Since we know that the holiday week plays a major role in the spike in sales, more weightage is given to the holiday week in the data. So, we chose Weighted Mean Absolute Error (WMAE) as our metric. The table below shows the WMAE values of various models we tried for forecasting the weekly sales.

Model	WMAE
SARIMA $(0,1,1)x(1,1,0)_{52}$	499.8
SARIMA $(0,1,5)x(1,1,0)_{52}$	517.9
SARIMA $(0,1,1)x(1,1,1)_{52}$	580.3
SARIMA $(0,1,1)x(2,1,0)_{52}$	717.3



The residual analysis has been performed on the final model. It was done to capture whether the final model has captured adequate information from the data. A residual density plot above has been plotted and it is around 0, which suggests the model has decently captured the data.

Analysis C- Advanced Model (Random Forest)

Random forest regressor is the advanced model used in this project. A random forest is an ensemble method that learns from many weak learners to make powerful predictions. Random Forest is a technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as the bagging technique. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

A machine learning model such as a random forest regressor avoids overfitting effectively with high performance making it a good choice as a predictive model. The collection of weak learners is what helps the model be robust to unseen data. The model has 200 estimators (number of trees) with a maximum depth (height of the trees) of 35 nodes and a minimum sample split (number of observations) of 10.

The features listed after data pre-processing are: ['Store', 'Dept', 'Date', 'Weekly_Sales', 'IsHoliday', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemployment', 'Type', 'Size', 'Super_Bowl', 'Labor_Day', 'Thanksgiving', 'Christmas', 'week', 'month', 'year'].

Experiments:

In this project, several experiments are executed to find the best-performing random forest model through feature selection. Several models are selected based on the combination of the features correlation matrix and feature importance generated through Random Forest. Table 1. Shows the results obtained through the experiments.

1) **Model 1** – This model is built using all the features on the preprocessed data set.
2) **Model 2** – Model built using selected features: Fig. C.1 shows feature importance on the complete data set. Based on the feature importance several columns are dropped and a model is trained

using the remaining features. The best result is obtained using the data set selected in this experiment.

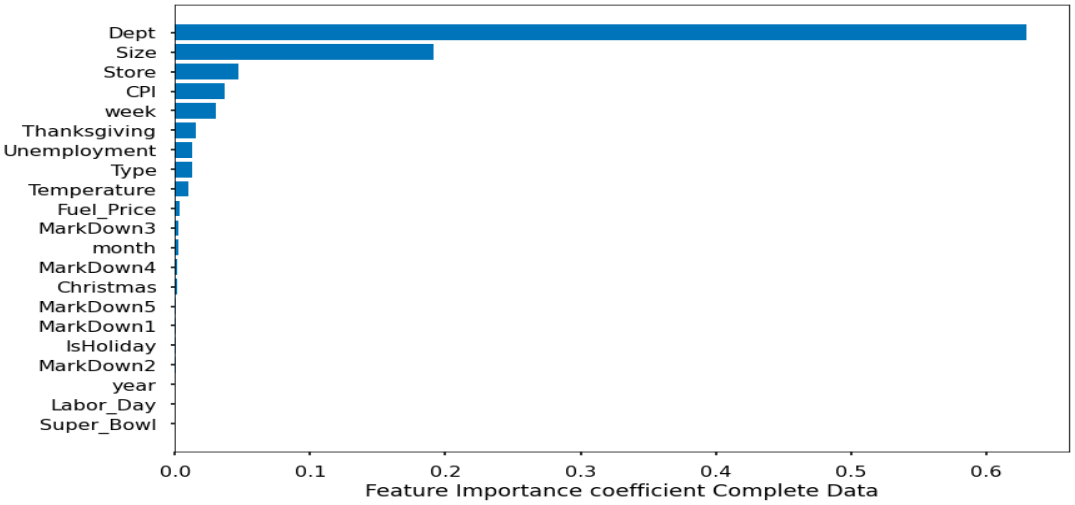


Fig. C.1 Features with importance coefficient generated by random forest.

3) **Model 3** - after dropping features based on the Pearson correlation coefficient matrix: Features with high correlation coefficients above 0.7 cause multicollinearity problems in the model. Thus, highly correlated features need to be removed. From the correlation matrix shown in fig C.2, it is shown that the temperature, unemployment, and CPI features have no significant effect on weekly sales (target). Thus, these features are dropped. The features ` Markdown 4` and `Fuel Price` are dropped because the former feature is highly correlated with a coefficient of 0.79 with `Markdown1` and the 'Fuel_price' feature is correlated with `year` with a coefficient of 0.78. So, we will also drop the Fuel Price.

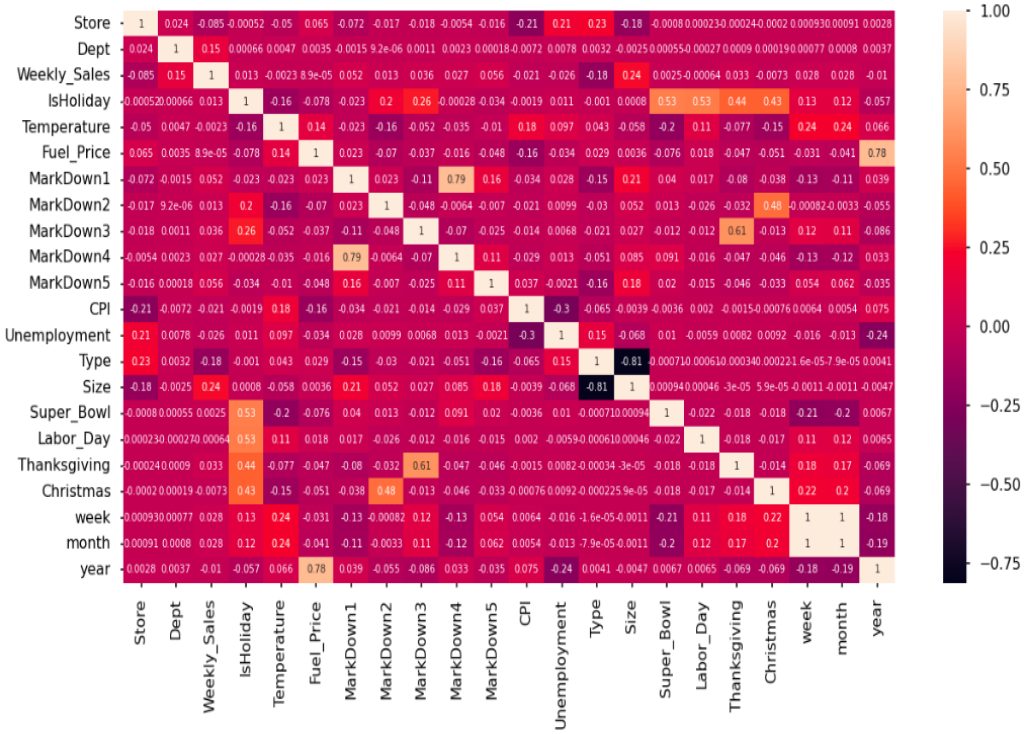


Fig. C.2 Pearson Correlation Coefficient Matrix obtained using python corr() function.

4) **Model 4** – after dropping feature based on the importance scores from the above model3: A model is trained on a new set of features. The new set is compiled by dropping the six least important features (Labor_Day, Super_Bowl, MarkDown2, MarkDown5, IsHoliday, Christmas, month) based on the feature importance coefficient predicted by random forest as shown in Fig C.3. and developed the model. From the figure, it is shown feature `Dept` has the highest feature importance while ` MarkDown2` and ` IsHoliday` have the least significance on the feature importance score scale.

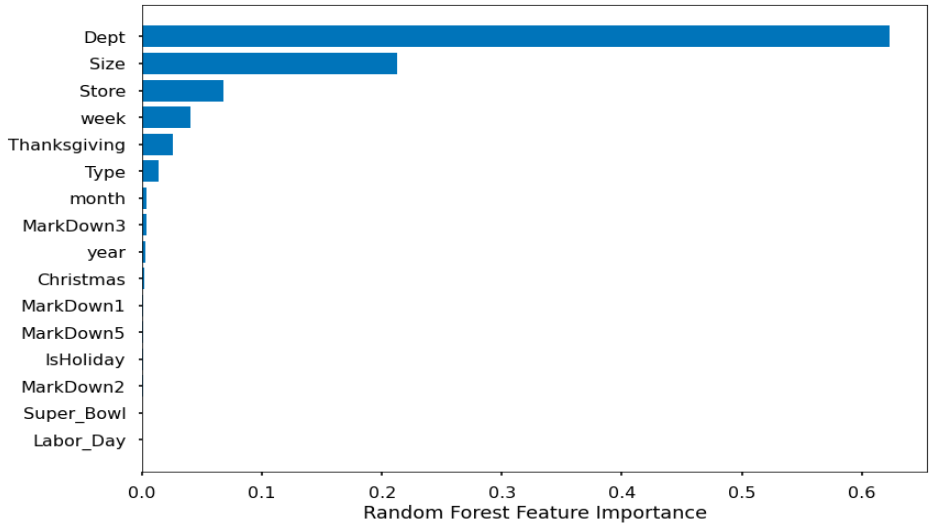


Fig. C.3 Features, and their importance predicted by Random Forest

Models	Description	WMAE	R^2 Score
Model1	Complete data	1931.35	0.6414
Model2	Complete data with feature selection (best Model)	1383.74	0.7343
Model3	excluding collinear features	1669.98	0.9753
Model4	excluding least important features on Model3	1662.68	0.9750

Table C.1. Model Performance Metrics (WMAE & R^2). Model 2 is the based model with the least WMAE.

Model Evaluation:

The performance metric used is Weighted Mean Absolute Error (WMAE) given by the equation.

$$\text{WMAE} = \frac{1}{\sum w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i|$$

In the equation `n` is the number of observations, Y_i is the actual sales, and $\overline{Y_i}$ is the predicted sales. W_i are weights. If the week is a non-holiday week, the weight is 1 otherwise with this metric, the error at holiday weeks has 5 times the weight more than normal weeks. So, it is more important to predict sales during holiday weeks accurately.

Model Interpretation

The Seasonal ARIMA model (**SARIMA (0,1,1) x $(1,1,0)_{52}$**) was the best model that gave us accurate forecasting from the set of models we tried. This Model captured the additive changes over the span of 140 weeks accurately, which helped in identifying the seasonality and trend. The First order differencing gave us a stationary dataset to proceed with the ACF and PACF plots. The non-seasonal components were easily interpretable from the plots. The seasonal components had only one significant lag (s=52) in the ACF plot, and the series immediately cut off at the next seasonal lag, indicating a possible MA(Q=1) model.

The seasonal part was symmetric in nature and was gradually tapering off. If we could have access to a larger time frame of sales data, the plot would have captured the probable seasonal lags and a better model could have been obtained. We also tried to change the scale of the data before fitting it into the SARIMA model by transforming it into a 'log' scale. However, there wasn't much improvement in AIC, BIC, and WMAE values for this model. So, we proceeded to construct the model with the original scale.

Discussion

On the whole, a SARIMA model can be used in time series analysis to capture the seasonality in the input data, and accurately capture the correlation of the lags with the current timestep. A random forest regressor takes a different approach and considers the various features in the dataset to predict a target variable, in this case, the weekly sales. While the random forest has a better predictive ability on the majority of the data, the SARIMA gave us statistically significant results. There were other approaches pertaining to deep learning algorithms and neural networks that could have given us better accuracy. The Random Forest approach itself took us 4-5 hrs. to train the model apart from hyper-parameter tuning time. But this helped us in identifying the top contributing features for building a decision tree in the random forest, which sort of helped in interpreting the underlying nuance of this approach.

Neural Networks would require heavy computation to train the model, and at this time we didn't have enough resources to accommodate it for the project. We intended to try constructing an LSTM network, which is based on RNN architecture. This is the latest approach used in real-time to work on time-series data in the industry alongside GARCH Models and Spectral Analysis.

Results & Conclusion

Comparing the two models, the random forest and SARIMA, we can see that the Seasonal ARIMA model provides better forecast accuracy and predictive ability. The reason that the Random Forest model didn't perform well in this situation, is because, to reduce the absolute error, we performed hyper-parameter tuning which ended up building us the deeper trees for each random decision tree in it. Generally, as the depth of the tree increases, the model tends to overfit the data, which means the model performs better on the training dataset, but when faced with new data, in this case, the test data, it performs poorly. The best random forest regressor model that we tried has a WMAE of 1383.7, which means that it has 3 times the absolute error than our SARIMA model. As the SARIMA model is statistically significant, and other model diagnostic plots of SARIMA are also significant, practically this led to a better forecast of weekly sales.

References

Beheshti, N. (2022, March 2). *Random Forest Regression*. Retrieved from Medium: https://towardsdatascience.com/random-forest-regression-5f605132d19d

Sklearn. (n.d.). *sklearn.ensemble.RandomForestRegressor*. Retrieved from scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Stoffer, D. S., & Shumway, R. H. (2011). *Time Series Analysis and Its Applications.* Davis: Springer.

Walmart. (2014, February 20). *Walmart Recruiting - Store Sales Forecasting*. Retrieved from Kaggle: https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting

Appendix:

- The variables in the Model are explained below.

Features:	Description:
Store	The store number
Date	Specifying the Week (Friday of every Week)
Size	Size of the Store
Type	Store Type
Dept	Department of the Store
Weekly Sales	Sales for the given store
Holiday Flag	Whether it is Holiday week -True, Non-holiday week - False
Temperature	The temperature on the day of the sale
Fuel Price	Cost of fuel in the region
CPI	The prevailing consumer price index
Unemployment	Prevailing unemployment rate
MarkDown 1-5	Anonymized data related to promotional markdowns that Walmart is running. Markdown data is only available after Nov 2011 and is not available for all stores all the time. Any missing value is marked with an NA.

- The Code snippets of the SARIMA and Random Forest Models are given below for reference.

SARIMA Analysis – Approach 1 code:

```
# Creating the model
sarima = SARIMAX(train_data, order=(0,1,1), seasonal_order=(1,1,0,52))

# Fitting th model
sarima_fit = sarima.fit()

# Summary Statistics of the model
sarima_summary = sarima_fit.summary()
print(sarima_summary)
```

SARIMA Analysis – Approach 2 code:

```
# Opening hyperparamter recording list
order_aic_bic = []
model_number = 1

# Executing exhaustive search
for p in range(3):
        for q in range(3):
            for P in range(3):
                    for Q in range(3):
                        total_models = 3*3*3*3
                        try:
                            model = SARIMAX(train_data_diff, order=(p,0,q), seasonal_order=(P,0,Q,52))
                            results = model.fit(disp=False)
                            order_aic_bic.append((p, 1, q, P, 1, Q, 52, results.aic, results.bic))
                            print(f'{model_number} / {total_models}')
                            model_number = model_number + 1
                        except:
                            print(f'{model_number} / {total_models}')
                            model_number = model_number + 1
                            continue

# Creating a DF with the results
order_df = pd.DataFrame(order_aic_bic, columns=['AR(p)', 'Diff(d)', 'MA(q)', 'SAR(P)', 'SDiff(D)', 'SMA(Q)', 'Seas(S)', 'AIC', 'BIC'])
sorted_order_df = order_df.sort_values('AIC')
print(sorted_order_df.head())
```

Random Forest Model code:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.pipeline import make_pipeline, Pipeline
rf = RandomForestRegressor(n_estimators=50, random_state=42, n_jobs=-1, max_depth=30,
                           min_samples_split = 10)

from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()



#making pipe tp use scaler and regressor together
pipe = make_pipeline(scaler,rf)

pipe.fit(X_train, y_train)

# predictions on train set
y_pred = pipe.predict(X_train)

# predictions on test set
y_pred_test = pipe.predict(X_test)
```