Introduction

As a part of this assignment, we have been provided with data from one of the largest food chain restaurants in the US. Data included (1) transactional information about dishes sold, prices, quantities, tax, etc; (2) Ingredients usage per recipe and sub-recipes mappings (3) Store locations, measurement type, recipe names, etc. Transactional data has been collected from Early March 2015 till Mid June 2015 for 4 stores (2 in Berkeley, CA and 2 in NY).

Our main aim as a part of this assignment is to combine all data to find out the usage of **lettuce** per day at store with ID-46673, train holt-winters & ARIMA time series prediction models, and make a prediction for next two weeks lettuce requirement.

Our report is structured as per below steps:

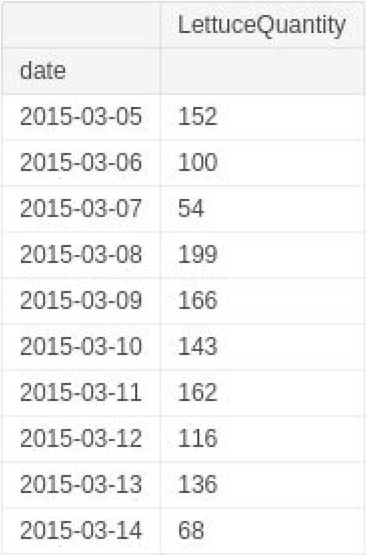
1. Data Loading, Cleaning & Preparation
2. ARIMA Application
3. Holt-Winters Application
4. Comparison between ARIMA and Holt-Winters Results
5. Conclusion

1. Data Loading, Cleaning & Preparation

Data loading step involves loading various CSV files which contains information about menu items purchase, recipe details, sale details, ingredient details, sub-recipe details, mapping from recipes to ingredients, mapping from sub-recipes to ingredients, store details, ingredients measurement unit details, etc.

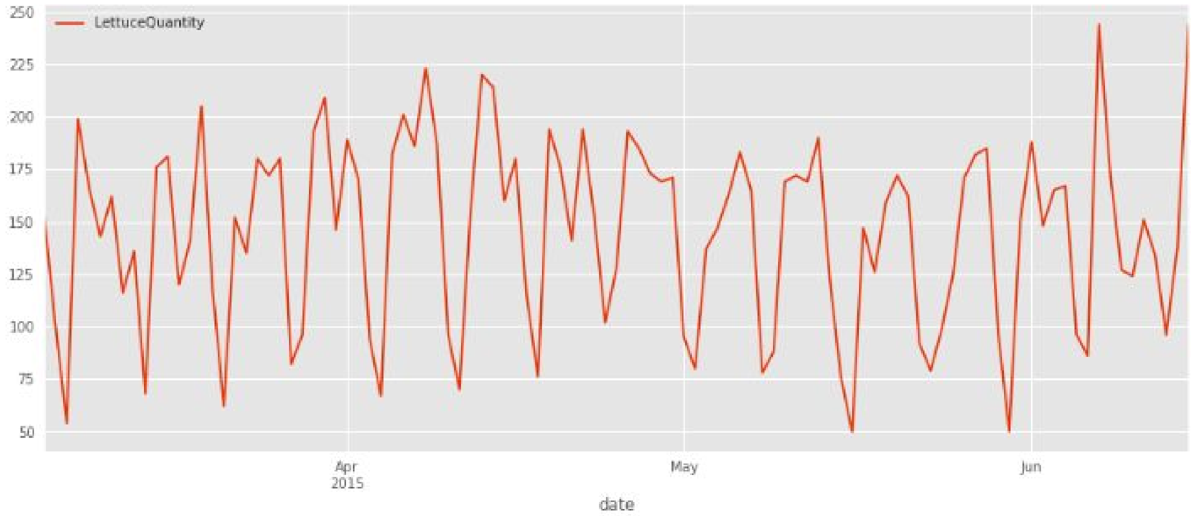
Data Cleaning step involves mainly taking care of data formats of loaded data. We have specifically converted date data from CSV into a proper format so that it can be later on used as an index of data. Other than that we have made sure that all other data like quantity, ids are in proper formats.

The data preparation step involves making data ready for usage with the machine learning model. As a part of this step, we first merged menu_items which has transactional information about recipes sale with menu item which has information about recipes ids. This helped use get data about all recipes sold between that period from all stores per day into one data frame. We then went on to create a generic method that takes as input ingredient name and returns dictionary which is mapping from recipe id to quantity of usage of that ingredient in that recipe. We used this method to get usage of lettuce quantity per recipe and incorporated it into the merged transaction data frame created earlier. We then filtered data to get only rows with hotel id 46673, grouped by on a daily basis and summed up lettuce usage quantity to get daily lettuce usage to create the final data frame.



	LettuceQuantity
date	
2015-03-05	152
2015-03-06	100
2015-03-07	54
2015-03-08	199
2015-03-09	166
2015-03-10	143
2015-03-11	162
2015-03-12	116
2015-03-13	136
2015-03-14	68

We also have plotted lettuce usage over the period to get an idea about a trend, seasonality, etc.



We can clearly see from the above graph that lettuce usage is showing no trend but there seems to be a seasonality of around 7-days(weekly).

2. ARIMA Application

Autoregressive Integrated Moving Average model is a class of statistical models for forecasting time-series data. ARIMA as its name suggests consists of 3 components (AR - A model that uses the dependent relationship between observation and some of its lag observations, I - Use of differencing of raw observations to make time-series stationary(i.e remove trends), MA - A model that uses dependency between model and residual error when moving error model is applied to past few observations.)

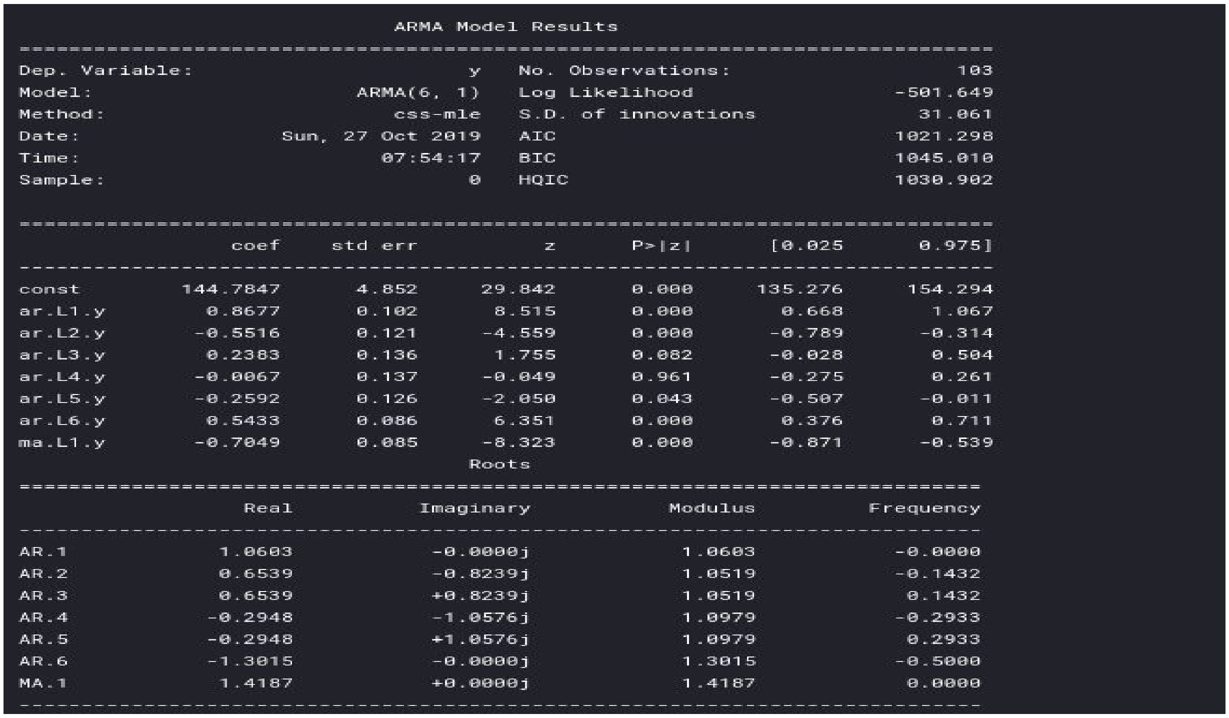
For our purpose, we have used the statsmodels library which provides the implementation of ARIMA. ARIMA model provided by the library accepts 3 parameters that we need to optimize to find the best results. We have mentioned that 3 parameters below:

p - number of lagged observations to consider
d - number of time raw observations are differenced.
q - size of moving average window.

We have done grid search through a list of values for these parameters to find the best fit as per below:
p - [0, 1, 2, 3, 4, 5, 6]
d - [0, 1, 2]
q - [0, 1, 2, 3]

We divided data into train and test sets. We then did a grid search through all combinations of values of these 3 parameters to find the best one. We did record the performance of the model on test data based on mean standard deviation(MSD) and selected parameter combination which gives the least value for MSD.

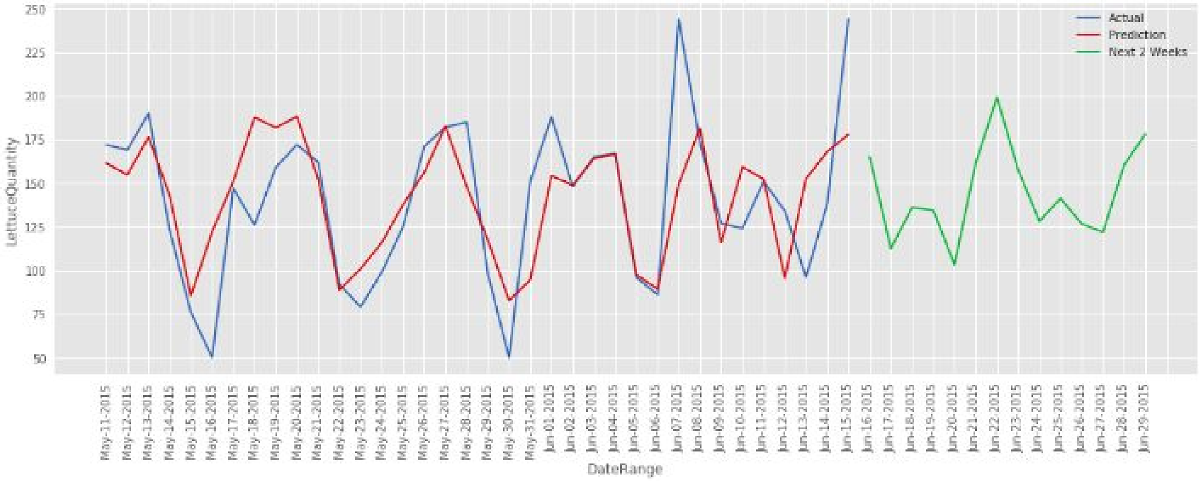
Best Parameter Settings: p - 6, d - 0, q - 1
Least Mean Standard Deviation: 1097.33

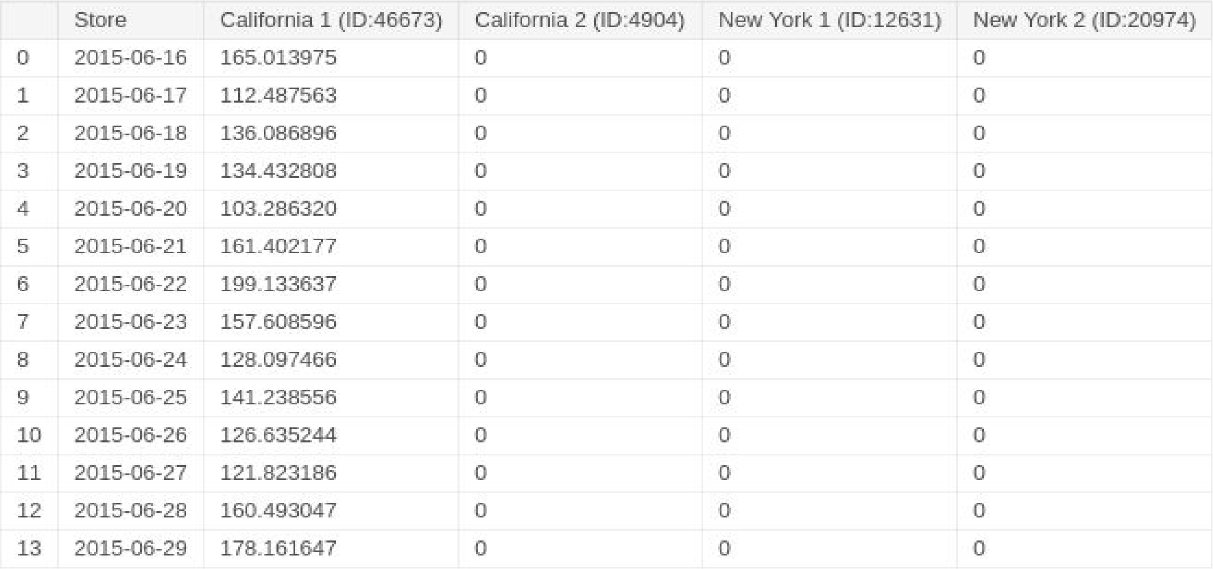


```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  103
Model:                     ARMA(6, 1)   Log Likelihood                -501.649
Method:                       css-mle   S.D. of innovations             31.061
Date:                Sun, 27 Oct 2019   AIC                           1021.298
Time:                        07:54:17   BIC                           1045.010
Sample:                             0   HQIC                          1030.902

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        144.7847      4.852     29.842      0.000     135.276     154.294
ar.L1.y        0.8677      0.102      8.515      0.000       0.668       1.067
ar.L2.y       -0.5516      0.121     -4.559      0.000      -0.789      -0.314
ar.L3.y        0.2383      0.136      1.755      0.082      -0.028       0.504
ar.L4.y       -0.0067      0.137     -0.049      0.961      -0.275       0.261
ar.L5.y       -0.2592      0.126     -2.050      0.043      -0.507      -0.011
ar.L6.y        0.5433      0.086      6.351      0.000       0.376       0.711
ma.L1.y       -0.7049      0.085     -8.323      0.000      -0.871      -0.539
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0603           -0.0000j            1.0603           -0.0000
AR.2            0.6539           -0.8239j            1.0519           -0.1432
AR.3            0.6539           +0.8239j            1.0519            0.1432
AR.4           -0.2948           -1.0576j            1.0979           -0.2933
AR.5           -0.2948           +1.0576j            1.0979            0.2933
AR.6           -1.3015           -0.0000j            1.3015           -0.5000
MA.1            1.4187           +0.0000j            1.4187            0.0000
-----------------------------------------------------------------------------
```

We then used the model with these best parameters to predict lettuce usage for the next 2 weeks from 15th June 2015 to 29th June 2015. We also have plotted results for it.





	Store	California 1 (ID:46673)	California 2 (ID:4904)	New York 1 (ID:12631)	New York 2 (ID:20974)
0	2015-06-16	165.013975	0	0	0
1	2015-06-17	112.487563	0	0	0
2	2015-06-18	136.086896	0	0	0
3	2015-06-19	134.432808	0	0	0
4	2015-06-20	103.286320	0	0	0
5	2015-06-21	161.402177	0	0	0
6	2015-06-22	199.133637	0	0	0
7	2015-06-23	157.608596	0	0	0
8	2015-06-24	128.097466	0	0	0
9	2015-06-25	141.238556	0	0	0
10	2015-06-26	126.635244	0	0	0
11	2015-06-27	121.823186	0	0	0
12	2015-06-28	160.493047	0	0	0
13	2015-06-29	178.161647	0	0	0

3. Holt-Winters Model

Holt-winters model also commonly known as a triple exponential smoothing model is a time-series forecasting model with support for trend and seasonality. It provides additive and multiplicative support for trends and seasonality.

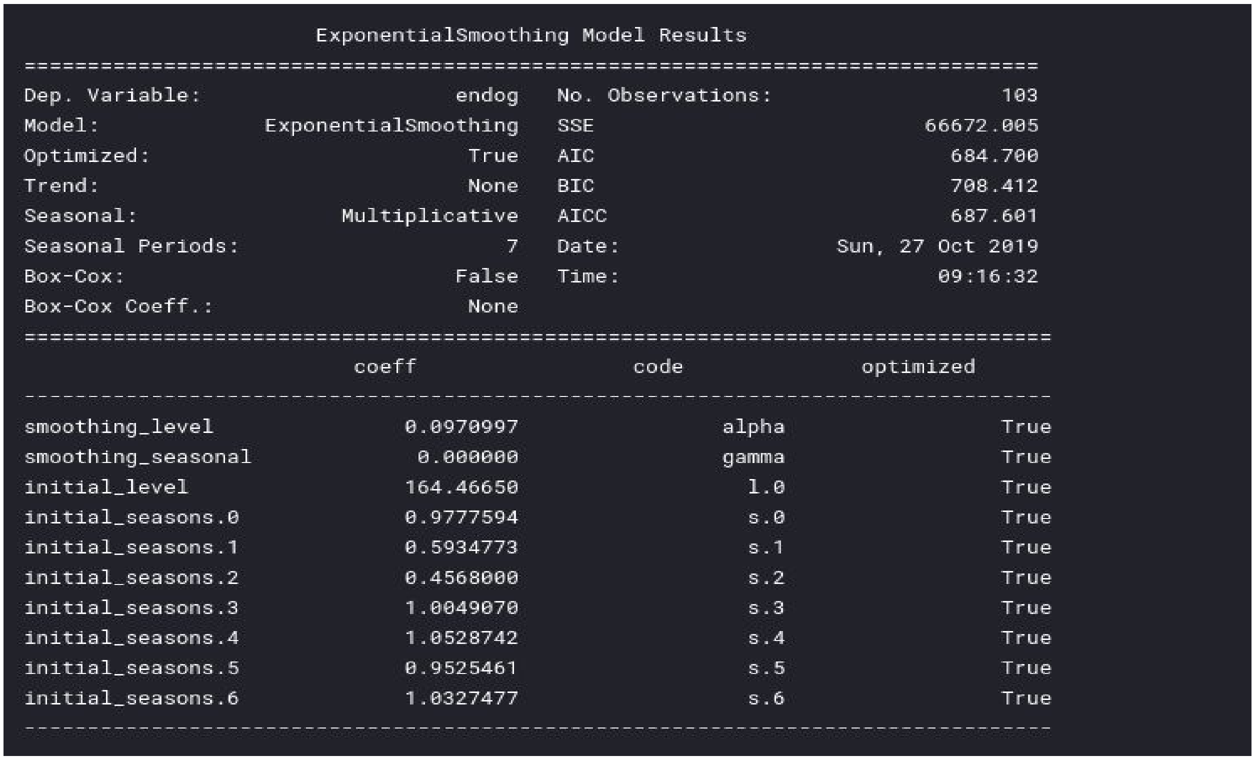
For our purpose, we have used the ExponentialSmoothing model provided by the statsmodels library. It accepts a list of parameters that decides the type of trend/seasonality, level of trend seasonality, etc. Below is a list of parameters that it takes as input.

trend - Additive or multiplicative.
damped - True or False
seasonal - Additive or multiplicative.
seasonal_periods - Time steps in the seasonal period.
use_boxcox - Whether or not to perform the power transform of series.
remove_bias - Whether to consider bias or not,

We have done grid search through a list of values for these parameters to find the best fit as per below:
trend - ['add', 'mul', None]
damped - [True, False]
seasonality - ['add', 'mul', None]
seasonal_periods - [7, 14, 30]
use_boxcox - [True, False]
remove_bias - [True, False]

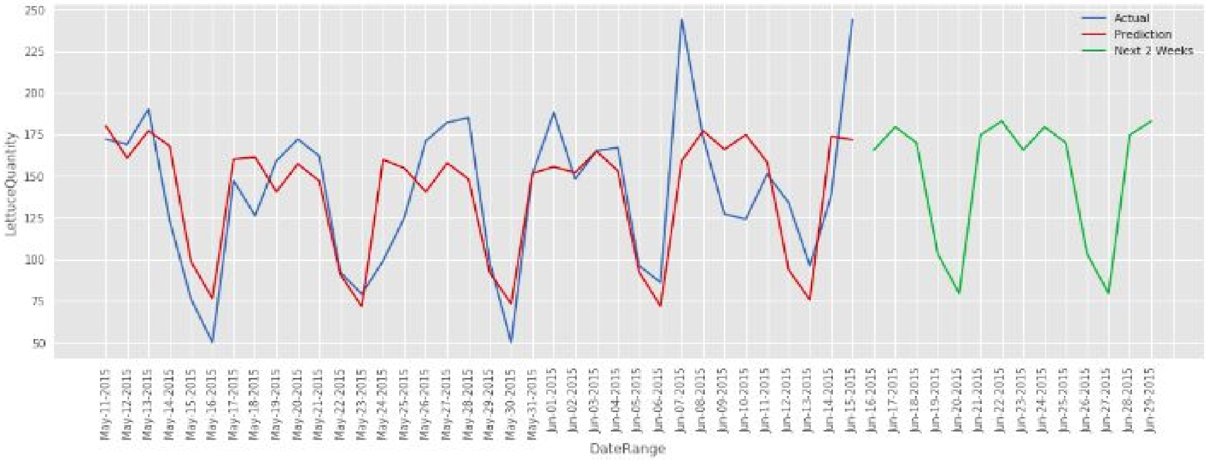
We divided data into train and test sets. We then did a grid search through all combinations of values of these 3 parameters to find the best one. We did record the performance of the model on test data based on mean standard deviation(MSD) and selected parameter combination which gives the least value for MSD.

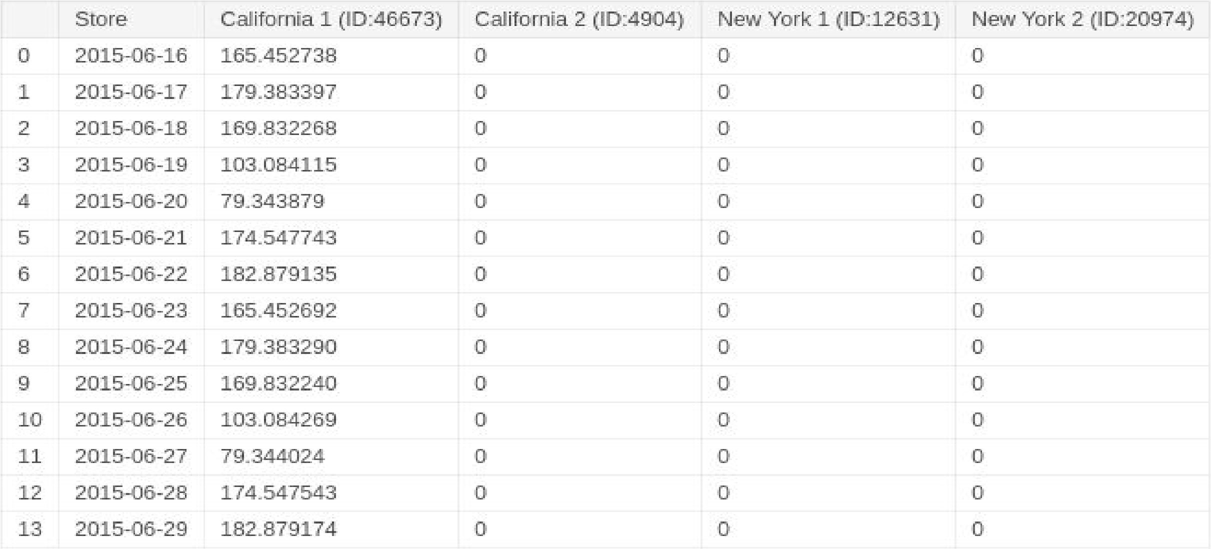
Best Parameter Settings: trend - None, damped - False, seasonality - 'mul', seasonal_periods - 7, use_boxcox - False, remove_bias - False
Least Mean Standard Deviation: 977.67



```
                        ExponentialSmoothing Model Results
================================================================================
Dep. Variable:                    endog   No. Observations:                  103
Model:             ExponentialSmoothing   SSE                          66672.005
Optimized:                         True   AIC                            684.700
Trend:                             None   BIC                            708.412
Seasonal:                Multiplicative   AICC                           687.601
Seasonal Periods:                     7   Date:                 Sun, 27 Oct 2019
Box-Cox:                          False   Time:                         09:16:32
Box-Cox Coeff.:                    None
=================================================================================
                          coeff                 code              optimized
---------------------------------------------------------------------------------
smoothing_level               0.0970997                alpha                True
smoothing_seasonal             0.000000                gamma                True
initial_level                 164.46650                  l.0                True
initial_seasons.0             0.9777594                  s.0                True
initial_seasons.1             0.5934773                  s.1                True
initial_seasons.2             0.4568000                  s.2                True
initial_seasons.3             1.0049070                  s.3                True
initial_seasons.4             1.0528742                  s.4                True
initial_seasons.5             0.9525461                  s.5                True
initial_seasons.6             1.0327477                  s.6                True
---------------------------------------------------------------------------------
```

We then used the model with these best parameters to predict lettuce usage for the next 2 weeks from 15th June 2015 to 29th June 2015. We also have plotted results for it.





	Store	California 1 (ID:46673)	California 2 (ID:4904)	New York 1 (ID:12631)	New York 2 (ID:20974)
0	2015-06-16	165.452738	0	0	0
1	2015-06-17	179.383397	0	0	0
2	2015-06-18	169.832268	0	0	0
3	2015-06-19	103.084115	0	0	0
4	2015-06-20	79.343879	0	0	0
5	2015-06-21	174.547743	0	0	0
6	2015-06-22	182.879135	0	0	0
7	2015-06-23	165.452692	0	0	0
8	2015-06-24	179.383290	0	0	0
9	2015-06-25	169.832240	0	0	0
10	2015-06-26	103.084269	0	0	0
11	2015-06-27	79.344024	0	0	0
12	2015-06-28	174.547543	0	0	0
13	2015-06-29	182.879174	0	0	0

4. Comparison between ARIMA and Holt-Winters Results

After applying various combinations of ARIMA and Holt-Winters, we can now do a comparison of the performance of both models on our dataset. Our main graph which showed usage of lettuce for a given time period also depicted that it has seasonality of 7 days. It did not have any trend though.

After trying various combinations of ARIMA model best model that we got had an MSD of **1097.33** which was best we can do with it. Various combinations of the holt-winters model gave the best MSD of **977.67** which was quite better than the ARIMA model. Holt-winters model also was able to capture that data does not have trend returning ‘None’ for trend parameter during grid search.

We also plotted graphs of the performance of both models on test data and prediction for the next 2 weeks as well. From both plots, we can see that the holt-winters model seems to be capturing seasonality quite well compared to ARIMA.

5. Conclusion

After performing a detailed analysis of lettuce quantity usage and trying various combinations of ARIMA and Holt-winters model, we have come to the conclusion that the holt-winters model is able to capture seasonality in our data better than ARIMA.