Time series analysis for Gas Prices

Summary

In the beginning of the project after visualizing the data set in time series format from EIA (U.S. Energy Information Administration) we found out that the data has an upward trend but no seasonality component. The data set from the website was from June of 2000 to February of 2023 monthly. The data set has 273 rows and 2 columns.

The data before processing in the Month column had string elements as “Feb, Mar” and beginning from 2023. Here we changed the string values to numeric values as “02/01/2023” and flipped the dataset so it starts from 2000.

After using multiple Regression Based models and multiple ARIMA Based models with auto correlation and seasonality in this project. The Regression and ARIMA based models where enhanced. After enhancing all the models we rounded the values using accuracy () and compared the RSME and MAPE values to find the suitable and best fitting model for the data.

In the end we found that Seasonal ARIMA(2,1,2)(1,1,2) has the lowest RMSE and MAPE models.

Introduction

Fuel Prices in California have been sky rocketing right after The Pandemic in 2020. Fuel prices have touched as high as 7$ per gallon in 2021-2022 . In this project we wanted to forecast the future fuel price per gallon for mid-grade fuel by using historical Califonria mid-grade fuel prices.We used the data provided by the Independent Statistics and Analysis U.S.Energy Information Administration. The website is proving data in civ format of multiple locations and multiple grade levels with granularity as low as daily.

We have used historical data from past 20 years to forecast the fuel prices in California for 12 months. So the main scope of this project is to analyze and forecast monthly data of 20 years and predict the fuel prices monthly. This will help us to have a simple idea regarding the fuel prices in the future if the future fuel prices do show an upward trend with no seasonality component present.

Eight Step Forecasting

Step 1: Define Goal

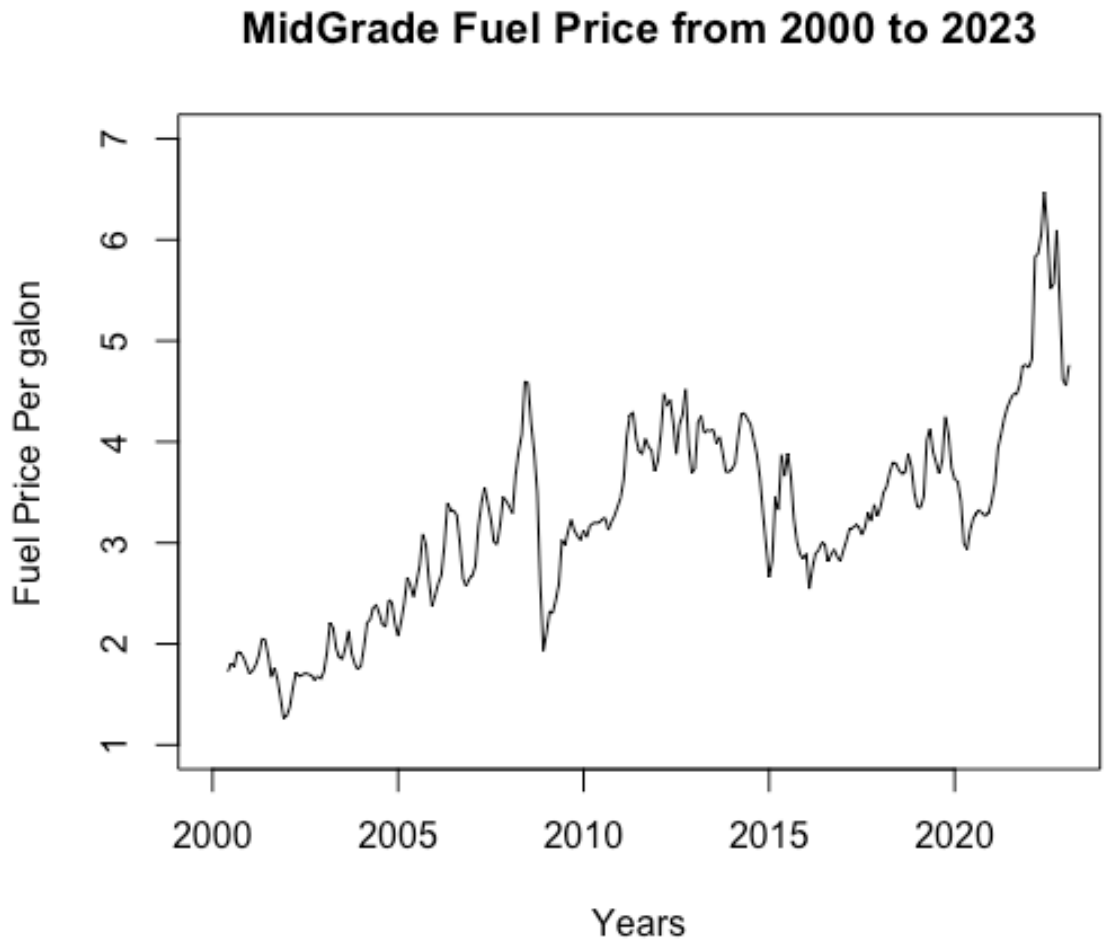
It is the aim of this time series project to predict the future price of gas in order to make accurate predictions in the future. As a result of our research, we have specifically picked the Historical California Mid-Grade Fuel Price over the past 20 years from June 2000 to February 2023. RMSE and MAPE values will be used to compare multiple models by using the data set in order to compare RMSE and MAPE values with one another. Model accuracy is generally associated with lower RMSE and MAPE values which could be attributed to low RMSE and MAPE. As a starting point, we will be implementing the models by using the R language since it is considered to be one of the best statistical tools for programmers and it supports a wide range of statistical models. Towards the end of this project, we will discuss the data that was used as a starting point for developing the models that will serve as the foundation of this project, as well as the data that was used to develop them.

Step 2: Get data

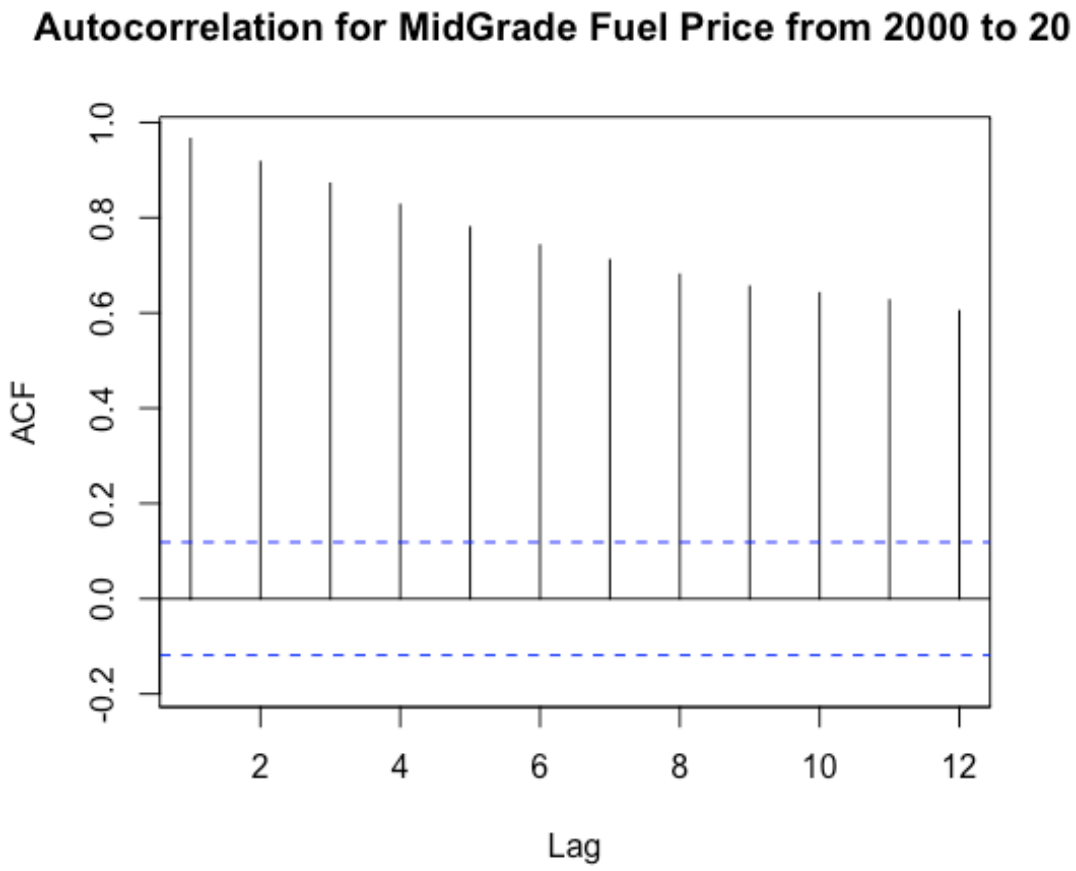
This report will focus on the time series dataset provided by the EIA (U.S. Energy Information Administration). The time period for the dataset ranges from June of 2000 to February of 2023. The data set contains 2 columns (Month: the month, year, and price per gallon) and 273 rows (one row represents each month). The data available on the website contains String data as "January 2001" as this does not work with our model we have to pre process the data to convert

it into 01/01/2001. Furthermore, the data set on the website starts from 2023 and ends with 2000 and this is inverted to fit our model.

Step 3: Explore and Visualize Series



From the above graph “MidGrade Fuel price from 2000 to 2023” we observe that the time series data has an upward Trent but no seasonality. This means that the variable we are measuring which is the gas prices were it is increasing and decreasing over time without any pattern observable such monthly.



From the above chart “Autocorrelation for MidGrade Fuel Price from 2000 to 2023” we observe that the data is highly correlated, as the ACF from lag1 to lag 12 are substantially higher than Zero which means it is higher than the horizontal threshold. A +ve ACF in lag 1 which is higher than Zero or horizontal threshold indicates an upward trend and a +ve ACF in lag 12 which is higher than Zero or horizontal threshold indicates that it doesn't have a seasonal component.

As the data indicates an upward trend and no seasonal component so the data is not just flat or constant over time and a pattern is formed.

Step 4: Data Preprocessing

The data that we got from U.S. Energy Information Administration website has only 2 columns (Month and Price.per.galon). The dataset contains 273 rows ranging from monthly June of 2000 to February of 2023.

```
> dim(mydata_midgrade)
[1] 273   2
```

The data before processing in the Month column had string elements as “Feb, Mar” and beginning from 2023. Here we changed the string values to numeric values as “02/01/2023” and flipped the dataset so it starts from 2000.

By using the below R code we where able to convert the dataset for the time series models .

```
##### DATA SETS #####
mydata_midgrade <- read.csv("California_Midgrade.csv")

#reordering the rows from bottom to up
mydata_midgrade <-mydata_midgrade %>% arrange(desc(row_number()))

#Chamging the string date to numeric date

# For mydata_midgrade
mydata_midgrade$Month <- as.Date(paste0("1 ", mydata_midgrade$Month), format = "%d %b %Y")
mydata_midgrade$Month <- format(mydata_midgrade$Month, "%m/%d/%Y")
mydata_midgrade
```

```
> head(mydata_midgrade)
     Month Price.per.galon
1 Feb 2023           4.761
2 Jan 2023           4.555
3 Dec 2022           4.619
4 Nov 2022           5.387
5 Oct 2022           6.096
6 Sep 2022           5.567
```

```
> head(mydata_midgrade)
       Month Price.per.galon
1 06/01/2000           1.726
2 07/01/2000           1.810
3 08/01/2000           1.774
4 09/01/2000           1.919
5 10/01/2000           1.911
6 11/01/2000           1.866
```

Step 5: Partition Series

The data partition can be divided into 2 parts Training data and Validation data. Training data is used for the model and Validation data is used compare it to the model. Here we have 213 records of training data and 60 records of validation data.

```
> nTrain
[1] 213
> nValid
[1] 60
```

The Training data is from 2000 June to 2017 September.

```
> train.ts
      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2000                                1.726 1.810 1.774 1.919 1.911 1.866 1.786
2001 1.705 1.737 1.794 1.890 2.050 2.037 1.884 1.672 1.766 1.650 1.461 1.264
2002 1.288 1.377 1.567 1.717 1.680 1.693 1.710 1.699 1.689 1.641 1.682 1.651
2003 1.725 1.914 2.211 2.157 1.952 1.868 1.851 1.973 2.128 1.903 1.805 1.746
2004 1.785 1.975 2.206 2.245 2.359 2.382 2.294 2.191 2.174 2.433 2.409 2.203
2005 2.077 2.223 2.404 2.654 2.579 2.468 2.614 2.777 3.086 2.980 2.626 2.376
2006 2.481 2.598 2.681 2.982 3.396 3.320 3.319 3.271 2.996 2.650 2.569 2.645
2007 2.674 2.770 3.165 3.400 3.547 3.394 3.237 3.013 2.986 3.174 3.454 3.413
2008 3.359 3.294 3.670 3.906 4.073 4.593 4.577 4.191 3.906 3.507 2.575 1.933
2009 2.108 2.321 2.305 2.435 2.589 3.027 2.980 3.117 3.228 3.124 3.069 3.027
2010 3.126 3.056 3.163 3.198 3.197 3.197 3.233 3.246 3.127 3.208 3.269 3.358
2011 3.449 3.634 4.060 4.266 4.290 4.026 3.904 3.882 4.030 3.952 3.912 3.712
2012 3.810 4.086 4.474 4.353 4.413 4.192 3.881 4.169 4.268 4.519 3.954 3.688
2013 3.739 4.189 4.254 4.090 4.112 4.109 4.118 3.981 4.048 3.887 3.702 3.704
2014 3.728 3.789 4.046 4.272 4.283 4.226 4.172 4.027 3.882 3.648 3.299 2.980
2015 2.661 2.820 3.451 3.326 3.870 3.661 3.884 3.666 3.243 3.010 2.888 2.844
2016 2.892 2.546 2.748 2.892 2.927 3.002 2.985 2.816 2.880 2.935 2.865 2.818
2017 2.924 3.023 3.138 3.146 3.180 3.158 3.080 3.153 3.299
```

And the Validation data is from 2017 October to 2022 September.

```
> valid.ts
      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2017                                                      3.215 3.377 3.267
2018 3.351 3.497 3.559 3.696 3.797 3.784 3.721 3.679 3.704 3.877 3.754 3.489
2019 3.352 3.355 3.458 4.014 4.132 3.903 3.796 3.686 3.821 4.248 4.077 3.743
2020 3.633 3.602 3.414 2.996 2.937 3.123 3.238 3.295 3.323 3.289 3.269 3.297
2021 3.411 3.589 3.937 4.074 4.228 4.338 4.415 4.476 4.469 4.560 4.748 4.759
2022 4.736 4.820 5.829 5.867 6.052 6.470 6.087 5.513 5.567
```

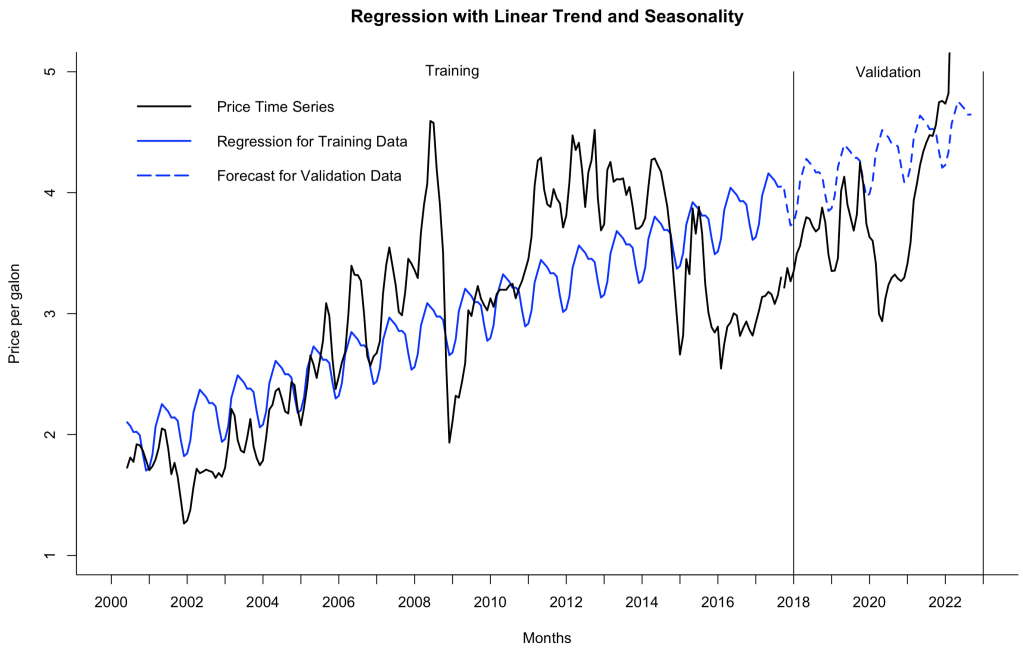
Step 6 & 7: Apply Forecasting & Comparing Performance

Regression Based Models:-

This model is considered as the most basic Time series model. We will be using linear trend and seasonal model by adding trend and season to training data. Below is the summary for the linear trend and seasonal model.

```
> summary(train.lin.season)

Call:
tslm(formula = train.ts ~ trend + season)

Residuals:
     Min       1Q   Median       3Q      Max
-1.11365 -0.41637 -0.03292  0.45692  1.55147

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.6445509  0.1592832  10.325   <2e-16 ***
trend        0.0099398  0.0006799  14.619   <2e-16 ***
season2      0.0971778  0.2018470   0.481   0.6307
season3      0.3216497  0.2018505   1.594   0.1127
season4      0.4053570  0.2018562   2.008   0.0460 *
season5      0.4876524  0.2018642   2.416   0.0166 *
season6      0.4471457  0.1990239   2.247   0.0258 *
season7      0.4068725  0.1990227   2.044   0.0423 *
season8      0.3460994  0.1990239   1.739   0.0836 .
season9      0.3382151  0.1990274   1.699   0.0908 .
season10     0.3004666  0.2018562   1.489   0.1382
season11     0.1301150  0.2018505   0.645   0.5199
season12    -0.0125896  0.2018470  -0.062   0.9503
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5885 on 195 degrees of freedom
Multiple R-squared:  0.5434,    Adjusted R-squared:  0.5153
F-statistic: 19.34 on 12 and 195 DF,  p-value: < 2.2e-16
```

For the above summary of Linear trend and regression model we observe that the model is statistically significant as the **p-value is very low at 2.2e-16** which is very low when compared to 5% alpha assuming the null hypothesis is True it states that there is no linear relation between predictor and the fuel prices. The R-squared tells how it measures how well the model fits the data which is at 94.71% which tells that the variation can be explained using predictors. Overall, this states that the data fits the model very well.



From the above graph we see how the Regression for training data (Solid blue), Forecast for Validation Data (Dotted Blue) and Original Time series data of Fuel prices in the data set (Solid Black). As explained in the beginning the data doesn't show any seasonal component which is presented in the above graph and has an upward trend.

Below table shows the Root Mean Square Error and Mean Absolute Percentage Error for Validation Period forecast.

RMSE	MAPE
0.753	16.353

Now we will be applying Forecasting on the Entire dataset below is the summary of the models to know how well the entire data fits the Model.

```
> summary(lin.season)

Call:
tslm(formula = price_per_galon.ts ~ trend + season)

Residuals:
     Min       1Q   Median       3Q      Max
-1.46388 -0.42999 -0.07798  0.49358  1.84620

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.660097   0.148122  11.208   <2e-16 ***
trend        0.009437   0.000485  19.456   <2e-16 ***
season2      0.095215   0.186167   0.511   0.6095
season3      0.308032   0.188280   1.636   0.1030
season4      0.391413   0.188276   2.079   0.0386 *
season5      0.475931   0.188273   2.528   0.0121 *
season6      0.462928   0.186197   2.486   0.0135 *
season7      0.414056   0.186189   2.224   0.0270 *
season8      0.338402   0.186182   1.818   0.0703 .
season9      0.340791   0.186177   1.830   0.0683 .
season10     0.326050   0.186172   1.751   0.0811 .
season11     0.168787   0.186169   0.907   0.3654
season12    -0.001302   0.186167  -0.007   0.9944
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6313 on 260 degrees of freedom
Multiple R-squared:  0.602,     Adjusted R-squared:  0.5836
F-statistic: 32.77 on 12 and 260 DF,  p-value: < 2.2e-16
```

For the above summary of Linear trend and regression model we observe that the model is statistically significant as the **p-value is very low at 2.2e-16** which is very low when compared to 5% alpha assuming the null hypothesis is True it states that there is no linear relation between predictor and the fuel prices. The R-squared tells how it measures how well the model fits the data which is at 60.20% which tells that the variation can be explained using predictors. Overall, this states that the data fits the model very well.

```
> lin.season.pred
         Point Forecast     Lo 0     Hi 0
Mar 2023       4.553835 4.553835 4.553835
Apr 2023       4.646653 4.646653 4.646653
May 2023       4.740608 4.740608 4.740608
Jun 2023       4.737042 4.737042 4.737042
Jul 2023       4.697607 4.697607 4.697607
Aug 2023       4.631389 4.631389 4.631389
Sep 2023       4.643215 4.643215 4.643215
Oct 2023       4.637911 4.637911 4.637911
Nov 2023       4.490085 4.490085 4.490085
Dec 2023       4.329433 4.329433 4.329433
Jan 2024       4.340172 4.340172 4.340172
Feb 2024       4.444824 4.444824 4.444824
```

Above we have the predicted values when the entire dataset is applied to the model.

Now we will be applying to ARIMA of order (1,0,0) which is AR(1)

```
> summary(res.ar1)
Series: train.lin.season$residuals
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1     mean
      0.9536  -0.0913
s.e.  0.0200   0.2376

sigma^2 = 0.0299:  log likelihood = 69.68
AIC=-133.37   AICc=-133.25   BIC=-123.36

Training set error measures:
                     ME      RMSE       MAE      MPE    MAPE      MASE      ACF1
Training set 0.002164164 0.1720905 0.1266469 22.05362 81.0292 0.3023896 0.2570892
>
```

Here we observe that MAPE is very high which is not suitable. So after applying to two-level model (linear trend and seasonal model + AR(1) model for residuals) we get the bellow metrics.

RMSE	MAPE
0.662	11.445

After applying two-level model (linear trend and seasonal model + AR(1) model for residuals), linear trend and seasonality model only and seasonal naive forecast. We get the bellow metrics.

Model	RMSE	MAPE
2-level + AR(1) +Residuals	0.19	4.296
Linear + Seasonality	0.616	16.11
Seasonal naive forecast	0.656	15.414

Fitting AR(2) Model:-

Now will be fitting AR(2) model we will use Arima() function to fit AR(2) model. The ARIMA model of order = c(2,0,0) gives an AR(2) model. Bellow we will use summary() to show AR(2) model and its parameters.

```
> summary(train.ar2)
Series: train.ts
ARIMA(2,0,0) with non-zero mean

Coefficients:
         ar1      ar2    mean
      1.3319  -0.3712  2.9046
s.e.  0.0641   0.0644  0.3102

sigma^2 = 0.03562:  log likelihood = 51.6
AIC=-95.2   AICc=-95   BIC=-81.85

Training set error measures:
                     ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set 0.005820479 0.1873768 0.1405155 -0.2483896 4.925463 0.3216772 0.01906252
```

RMSE	MAPE
0.187	4.925

Fitting AR(2,2) Model:-

Now will be fitting AR(2,2) model we will use Arima() function to fit AR(2,2) model. The ARIMA model of order = c(2,0,2) gives an AR(2,2) model. Bellow we will use summary() to show AR(2,2) model and its parameters.

```
> summary(train.arma2)
Series: train.ts
ARIMA(2,0,2) with non-zero mean

Coefficients:
         ar1     ar2     ma1     ma2    mean
      0.1869  0.7323  1.1127  0.3095  2.8873
s.e.     NaN     NaN     NaN  0.0517  0.3560

sigma^2 = 0.03604:  log likelihood = 51.43
AIC=-90.85   AICc=-90.43   BIC=-70.83

Training set error measures:
                      ME     RMSE       MAE        MPE     MAPE      MASE      ACF1
Training set 0.006807026 0.187537 0.1396137 -0.2004312 4.877817 0.3196128 0.0439257
```

RMSE	MAPE
0.187	4.877

Fitting AR(2,1,2) Model:-

Now will be fitting AR(2,1,2) model we will use Arima() function to fit AR(2,1,2) model. Bellow we will use summary() to show AR(2,1,2) model and its parameters.

```
> summary(train.arima)
Series: train.ts
ARIMA(2,1,2)

Coefficients:
          ar1      ar2     ma1     ma2
      -0.3080  -0.3182  0.6814  0.5486
s.e.   0.2085   0.1948  0.1806  0.1926

sigma^2 = 0.0357:  log likelihood = 53.07
AIC=-96.13   AICc=-95.84   BIC=-79.47

Training set error measures:
                      ME      RMSE       MAE       MPE     MAPE      MASE       ACF1
Training set 0.005688876 0.1866695 0.1372668 0.1124264 4.746124 0.3142401 0.01197349
```

RMSE	MAPE
0.186	4.746

Fitting ARIMA(2,1,2)(1,1,2) Model:-

Now will be fitting ARIMA(2,1,2)(1,1,2)For trend and seasonality. Bellow we will use summary() to show ARIMA(2,1,2)(1,1,2) model and its parameters.

```
> summary(train.arima.seas)
Series: train.ts
ARIMA(2,1,2)(1,1,2)[12]

Coefficients:
          ar1      ar2     ma1     ma2    sar1     sma1    sma2
      -0.5259  -0.5467  0.7569  0.7174  0.8090  -1.9711  0.9967
s.e.   0.2255   0.1448  0.1968  0.1331  0.0854   0.3933  0.3976

sigma^2 = 0.02776:  log likelihood = 53.05
AIC=-90.1   AICc=-89.32   BIC=-63.91

Training set error measures:
                      ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set -0.007473163 0.1583942 0.1111218 -0.3186003 3.756514 0.2543872 0.01376314
```

RMSE	MAPE
0.158	3.756

Fitting AUTO ARIMA Model:-

Now will be fitting AUTO ARIMAFor trend and seasonality. Bellow we will use summary() to show AUTO ARIMA model and its parameters.

```
> summary(train.auto.arima)
Series: train.ts
ARIMA(1,1,3)

Coefficients:
         ar1      ma1      ma2      ma3
      0.6643  -0.3366  -0.1884  -0.2700
s.e.  0.1119   0.1206   0.0704   0.0766

sigma^2 = 0.03408:  log likelihood = 57.8
AIC=-105.6   AICc=-105.3   BIC=-88.94

Training set error measures:
                     ME      RMSE       MAE       MPE     MAPE      MASE          ACF1
Training set 0.01141907 0.1823724 0.1341507 0.2197834 4.628205 0.3071065 -0.0001487627
```

RMSE	MAPE
0.182	4.628

Now we will be using accuracy function to identify common accuracy measures for validation period forecast.

Models	RMSE	MAPE
AR(2)	1.332	21.839
MA(2)	1.345	23.545
ARIMA(2,2)	1.324	21.463
ARIMA(2,1,2)	1.064	15.206
ARIMA(2,1,2)(1,1,2)	0.823	12.918
AUTO ARIMA	1.169	18.052

Fitting Seasonal ARIMA and AUTO ARIMA Models for Entire data set:-

First we will be using ARIMA (2,1,2)(1,1,2) with seasonal component for entire dataset. Now we will be using summary() to show auto ARIMA model and its parameters for entire data set.

```
> summary(arima.seas)
Series: price_per_galon_arima.ts
ARIMA(2,1,2)(1,1,2)[12]

Coefficients:
         ar1      ar2     ma1     ma2     sar1     sma1     sma2
      -0.4979  -0.6419  0.7336  0.6684  -0.6130  -0.3649  -0.6351
s.e.   0.1817   0.1631  0.1688  0.1789   0.4936   0.4832   0.4809

sigma^2 = 0.03629:  log likelihood = 46.81
AIC=-77.62   AICc=-77.05   BIC=-49.14

Training set error measures:
                      ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set -0.001255247 0.1833837 0.1286932 -0.1701348 3.934814 0.2567746 0.03090554
```

RMSE	MAPE
0.183	3.934

Now we will fit ARIMA model fro entire data set

```
> summary(auto.arima)
Series: price_per_galon_arima.ts
ARIMA(1,1,2)(0,0,1)[12]

Coefficients:
         ar1      ma1      ma2    sma1
      0.8444  -0.5501  -0.3614  0.1272
s.e.  0.0774   0.0836   0.0527  0.0671

sigma^2 = 0.04213:  log likelihood = 46.55
AIC=-83.1   AICc=-82.87   BIC=-65.07

Training set error measures:
                    ME     RMSE       MAE       MPE    MAPE      MASE       ACF1
Training set 0.0179591 0.203372 0.1448002 0.3471113 4.51058 0.2889119 0.01248307
```

RMSE	MAPE
0.203	4.510

Now we will compare above models accuracies

Models	RMSE	MAPE
Seasonal ARIMA(2,1,2)(1,1,2)	0.183	3.934
Auto ARIMA	0.203	4.511
Seasonal naive forecast	0.656	15.414
Naive forecast	0.221	4.997

Step 8: Implement Forecast

Models	RMSE	MAPE
Seasonal ARIMA(2,1,2)(1,1,2)	0.183	3.934
ARIMA(2,1,2)(1,1,2)	0.823	12.918
2-level + AR(1) +Residuals	0.19	4.296

From the above table we see that Seasonal ARIMA(2,1,2)(1,1,2) model fits the data best with lowest **RMSE of 0.183** and Lowest **MAPE of 3.934**.

	Point Forecast	Lo 0	Hi 0
Mar 2023	4.912358	4.912358	4.912358
Apr 2023	4.953767	4.953767	4.953767
May 2023	5.134002	5.134002	5.134002
Jun 2023	5.144187	5.144187	5.144187
Jul 2023	5.049048	5.049048	5.049048
Aug 2023	5.004775	5.004775	5.004775
Sep 2023	5.031084	5.031084	5.031084
Oct 2023	5.006190	5.006190	5.006190
Nov 2023	4.846937	4.846937	4.846937
Dec 2023	4.687972	4.687972	4.687972
Jan 2024	4.686869	4.686869	4.686869
Feb 2024	4.791222	4.791222	4.791222

The above Image shows the best predicted values of Mid-Grade fuel prices from March 2023 to Feb 2024 by using Seasonal ARIMA(2,1,2)(1,1,2) model.

Conclusion :-

At the end of this Time series Project to Predict the Fuel prices we found out that Seasonal ARIMA model that combines both autoregressive and moving average components with seasonal differencing [Seasonal ARIMA(2,1,2)(1,1,2)] is the best model for the dataset. After analyzing the historical data ranging for 20 years we where able to predict the Mid-Grade fuel prices.