Project Report 2 – Advanced Statistics

Group Members –

1) Suraj Shivakumar
2) Pranav Dange
3) Pranav Bansod
4) Rishabh Pattewar

Contents:

Data Selection for Time Series Forecasting

The fuel Price Dataset is selected for analysing trends in different cities and for forecasting.

Problem Statement

Since there are many uncertain and unforeseen factors that affect the crude oil market, researchers are paying more attention on the crude oil price forecasting which has widely been considered as the most important and challenging issue. So, Forecasting Fuel Prices over the next month's/ weeks / days is an important factor in understanding the trend and potentially acting proactively for the same. Our paper aims to ameliorate a forecast process for fuel prices by increasing the forecast accuracy by using ARIMA and dwells deep practically into integrating Auto Regressive and Moving Averages models, using the best hyperparameter for the same.

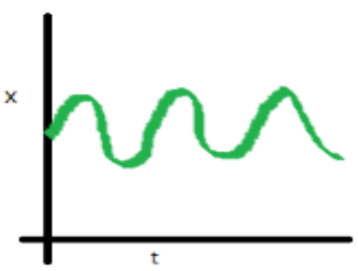
Intuition behind Time Series Forecasting

Time series forecasting is a technique for the prediction of events through a sequence of time.

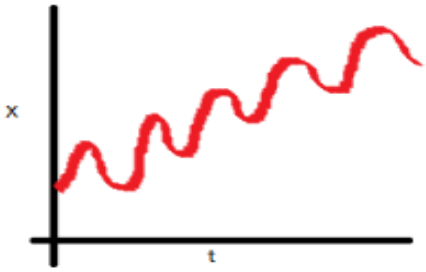
Before applying any statistical model on a time series, we want to ensure its stationary simply because of the fact that the Stationary Models are easy to predict.

What does it mean for data to be stationary?

1. The mean of the series should not be a function of time. The red graph below is not stationary because the mean increases over time.

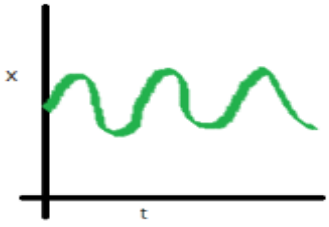


Stationary series

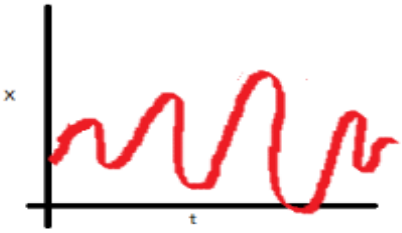


Non-Stationary series

2. The variance of the series should not be a function of time. This property is known as homoscedasticity. Notice in the red graph the varying spread of data over time.

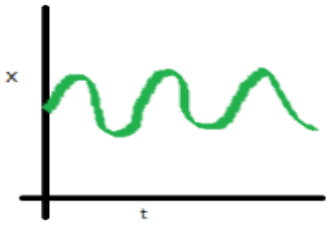


Stationary series

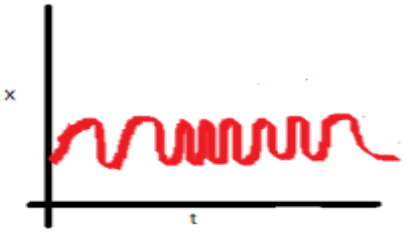


Non-Stationary series

3. Finally, the covariance of the i th term and the (i + m) th term should not be a function of time. In the following graph, you will notice the spread becomes closer as the time increases. Hence, the covariance is not constant with time for the 'red series'.



Stationary series



Non-Stationary series

Source –

Time Series Analysis

Auto Regressive Model (AR)

Autoregressive models operate under the premise that past values have an effect on current values. AR models are commonly used in analyzing nature, economics, and other time-varying processes. As long as

the assumption holds, we can build a linear regression model that attempts to predict value of a dependent variable today, given the values it had on previous days.

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \beta_3 y_{t-3} + \cdots + \beta_p y_{t-p}$$

Moving Average Model (MA)

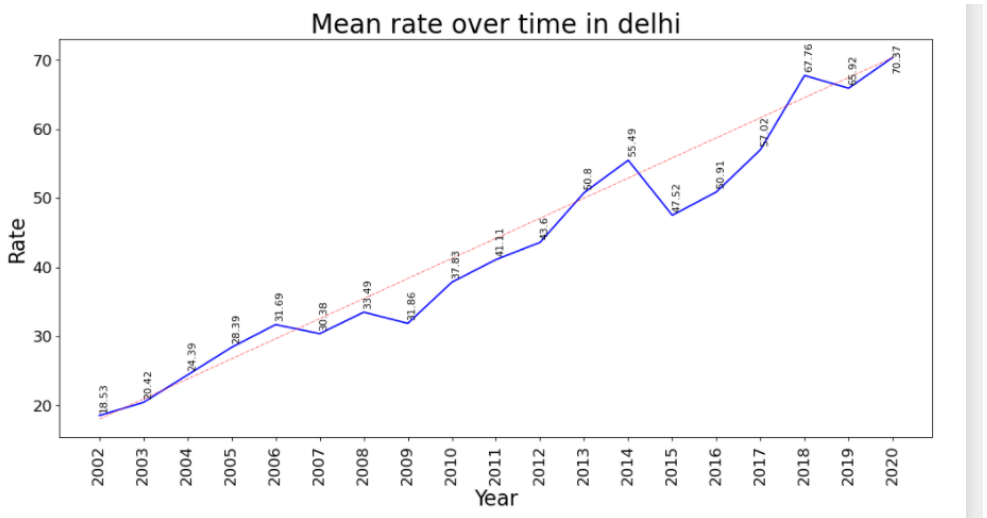
$$y_t = \varepsilon_t + \alpha_1 \varepsilon_{t-1} + \alpha_2 \varepsilon_{t-2} + \cdots + \alpha_q \varepsilon_{t-q}$$

Auto Regressive Moving Average (ARMA)

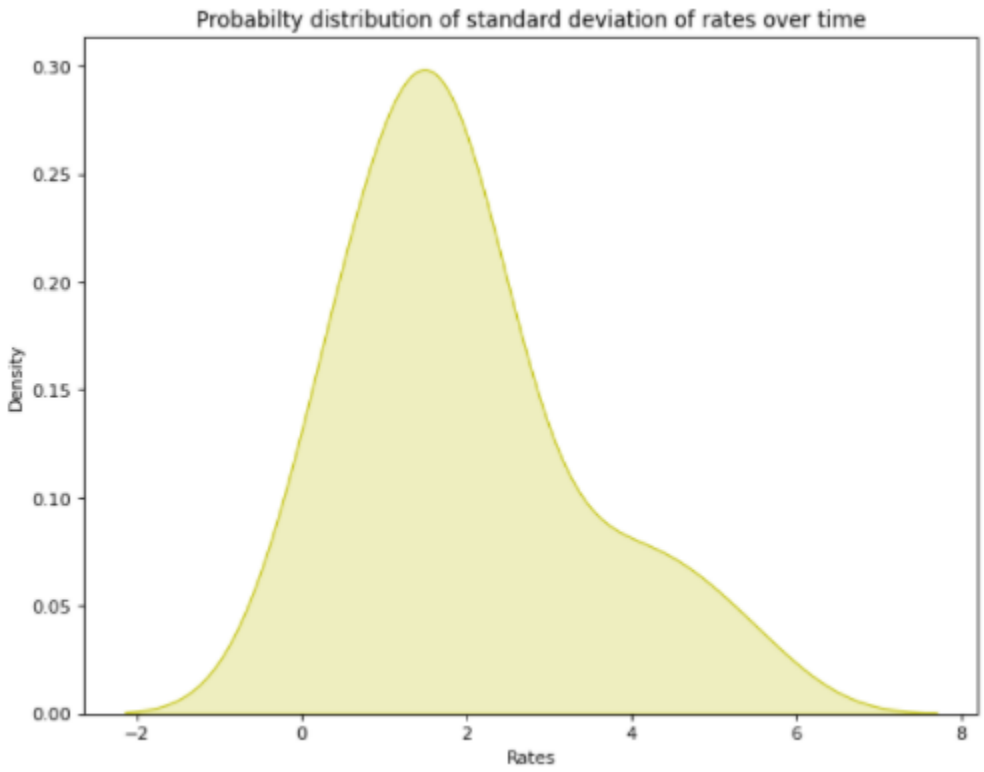
$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \beta_3 y_{t-3} + \cdots + \beta_p y_{t-p} + \varepsilon_t + \alpha_1 \varepsilon_{t-1} + \alpha_2 \varepsilon_{t-2} + \cdots + \alpha_q \varepsilon_{t-q}$$

In Autocorrelation we are comparing the time and the delayed version of the same. After applying the same in the model here, we can that there is autocorrelation for first 50 days and then the relation is lost and slowly depreciating -

So, to prove we don't have a stationary wave we plotted the mean price of Diesel for every consecutive year and it shows that - Mean is not constant over time from 2002 to 2020 which in turn proves that the data is not stationary.



The trend also suggests a decent increase in fuel price over the year. Similarly, variance is not constant as well typically over time.



Application of Auto Regressive Model and Moving Averages Model for forecasting fuel Price

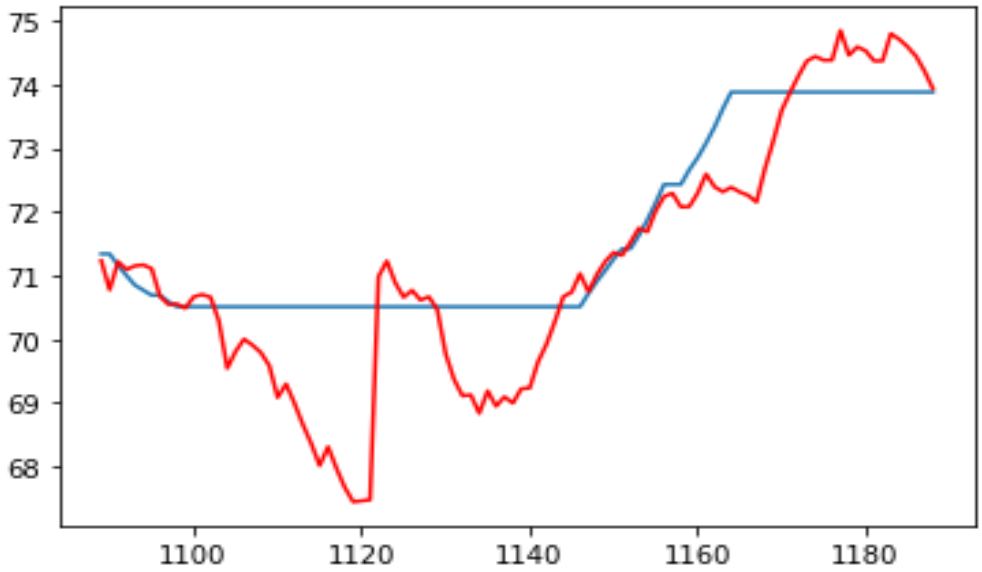
As it can be found from the graph that is highly correlated for last 3-4 days and not so over a long time as in 100 days or so

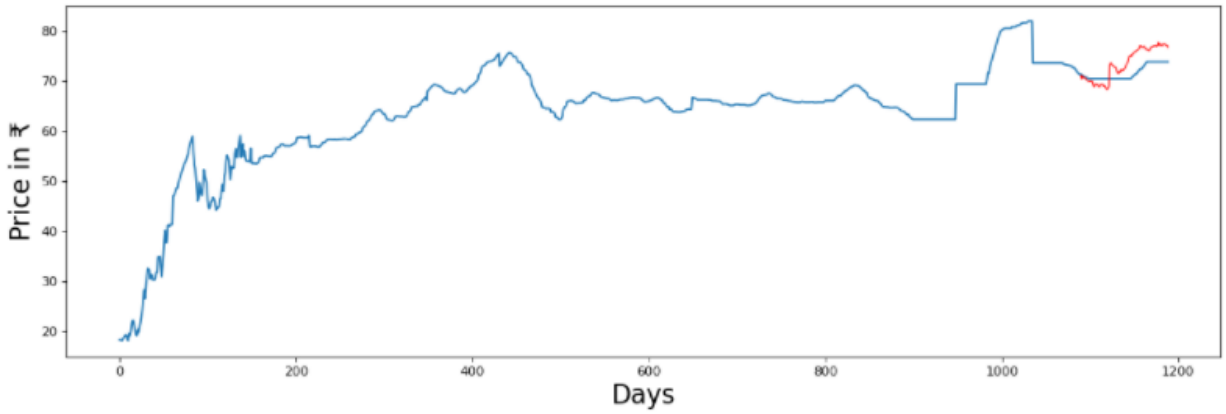
Here, alternately we can say that the data is partially related for t-1 day

- t being current day and
- t-1 day before present day

Then, by using the hyper parameter tuning we got 184 as an optimal number of lags to be used while predicting for next 100 days with an error of just 1 Rs at max.

Application of **Auto Regressive Model -**





This is an application of Auto Regressive model and its forecast.

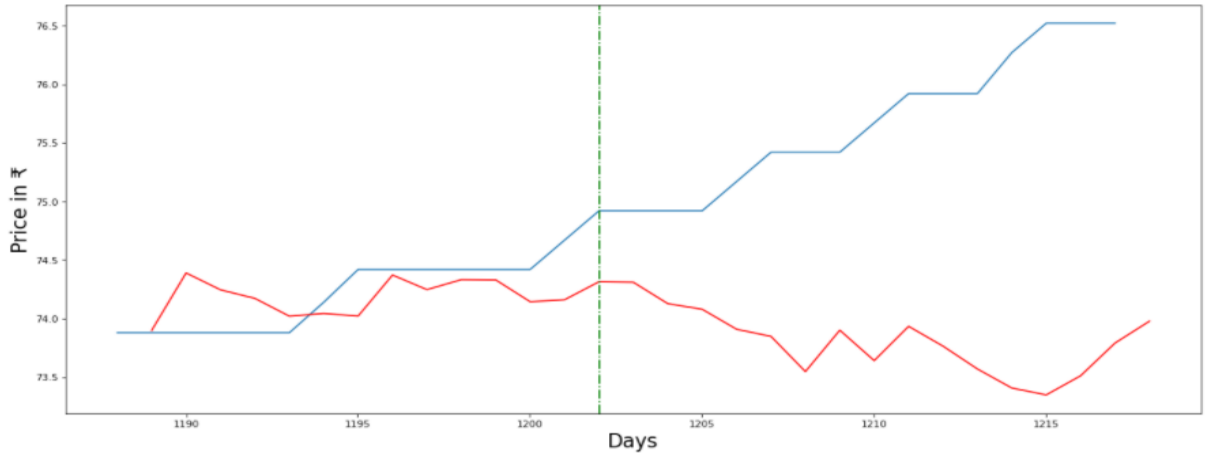
Now, we have next 30 days Validation Data completely taken for month January.

```
In [457]:
1 test_df = pd.DataFrame(data = {'Rate':rates_jan_2021[:30]},index=list(range(1188,1188+30)))
2 test_df.head()
```

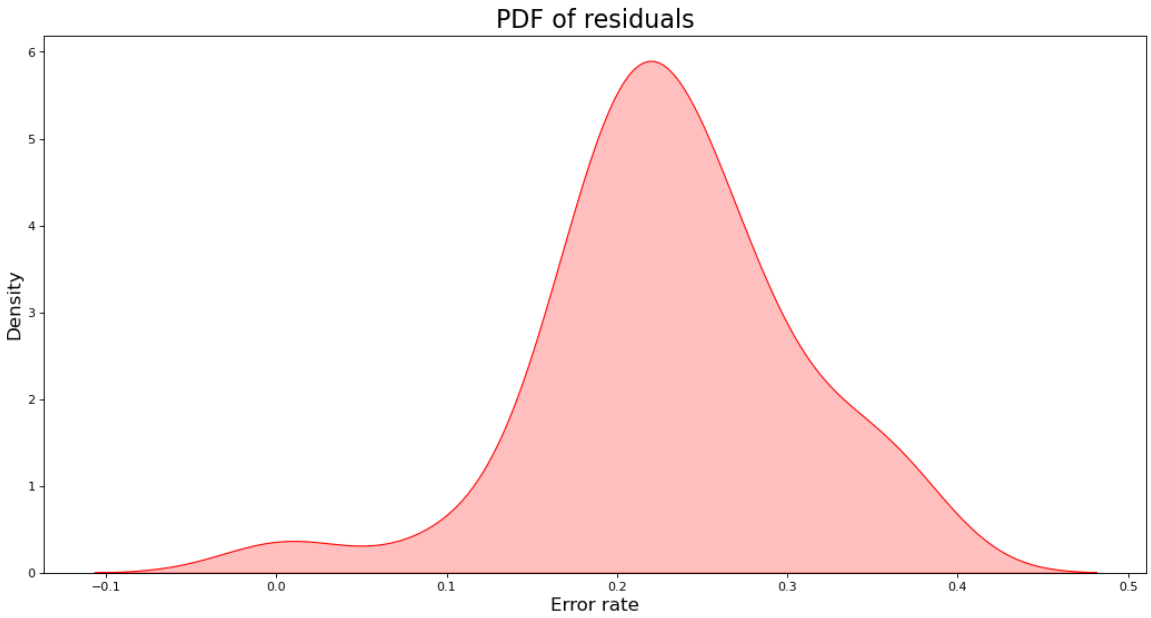
Out[457]:

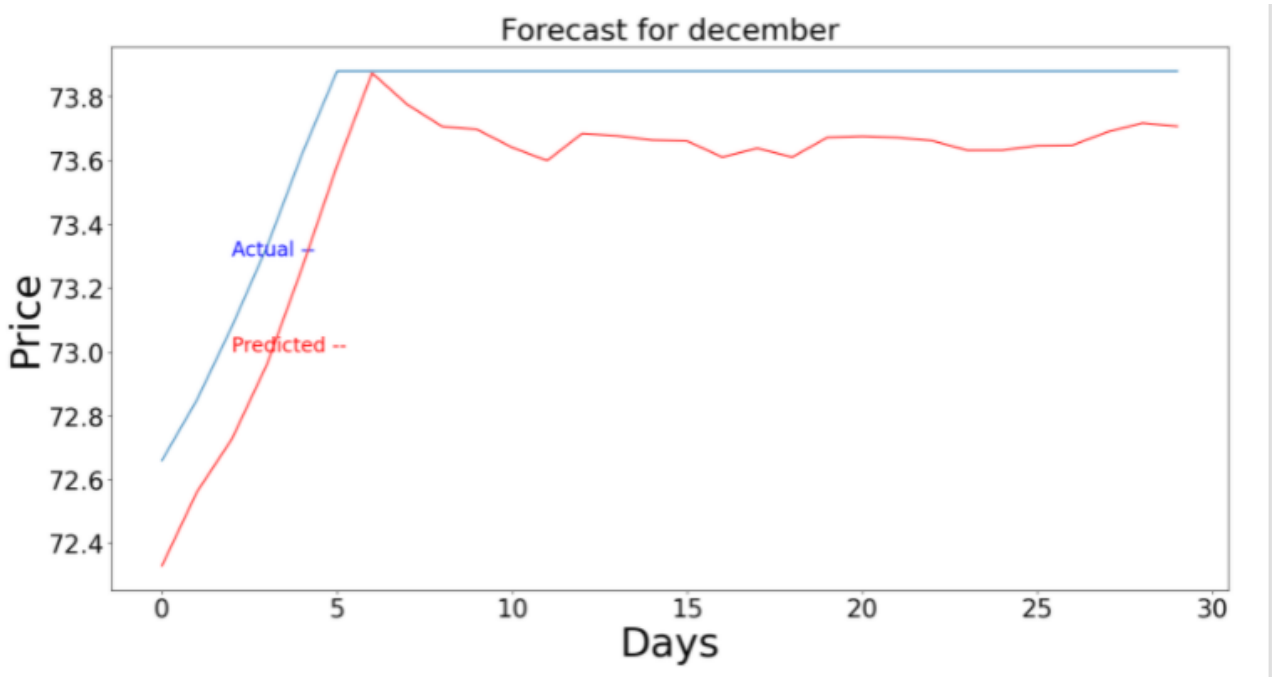
	Rate
1188	73.88
1189	73.88
1190	73.88
1191	73.88
1192	73.88

Forecasting on the next 30 Days data –



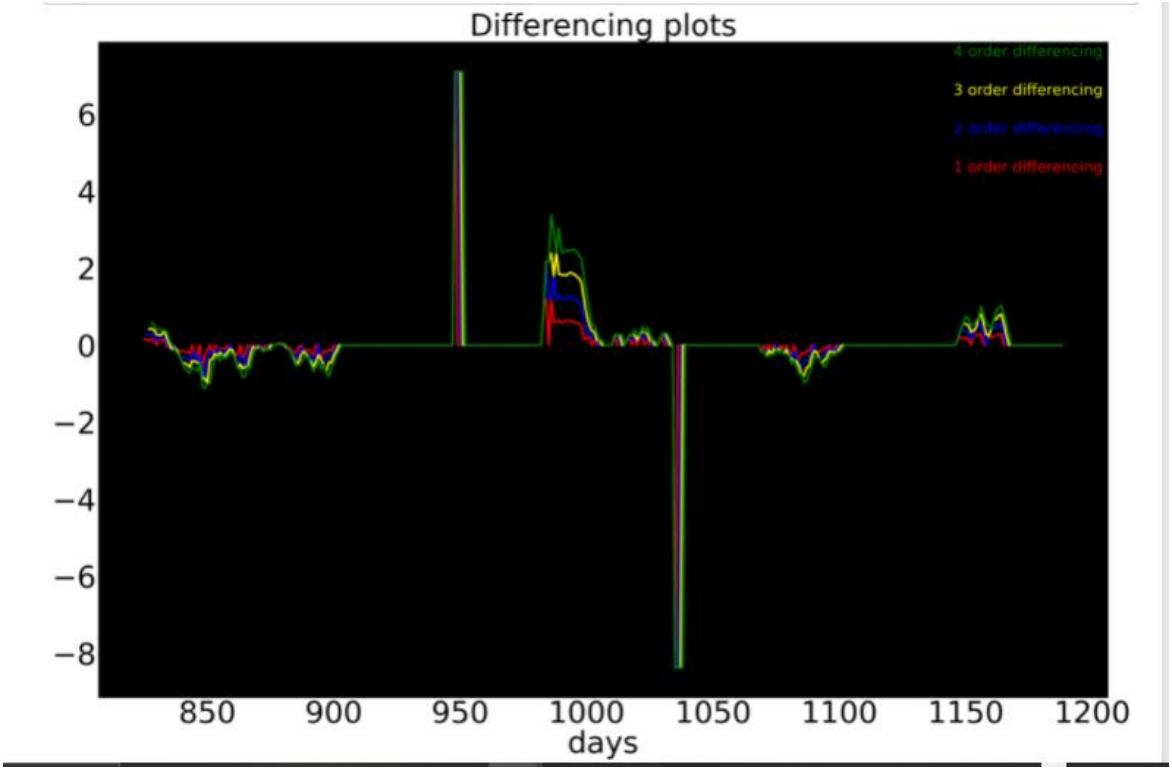
Moving Averages Model -





Differencing Time Series

Differencing is a method of transforming a time series dataset. It can be used to remove the series dependence on time, so-called temporal dependence. ... Differencing can help stabilize the mean of the time series by removing changes in the level of a time series, and so eliminating (or reducing) trend and seasonality.



Now, after applying 1, 2, 3, 4 order differencing we can say that the 1 order differencing will do the job.

Hyper Parameter Tuning for the Final Model-

Hyper-parameter tuning

In [465]:
```
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
from math import sqrt
# load dataset
# split into train and test sets
X = diesel_delhi.rate
train, test = X[0:len(X)-30], X[len(X)-30:]
rmse = []
avg = []
indices = []
# walk-forward validation
for j in tqdm([1,2,3,4,5]):
    for i in [1,2,3,4,5]:
        predictions = list()
        history = [x for x in train]
        for t in test:
            model = ARIMA(history, order=(i,1,j))
            model_fit = model.fit()
            output = model_fit.forecast()
            yhat = output
            predictions.append(yhat)
            obs = t
            history.append(obs)
        #print('predicted=%f, expected=%f' % (yhat, obs))
        # evaluate forecasts
        rmse.append(sqrt(mean_squared_error(test, predictions)))
        indices.append((i,j))


#print('Test RMSE: %.3f' % rmse)
# plot forecasts against actual outcomes
pyplot.show()
```

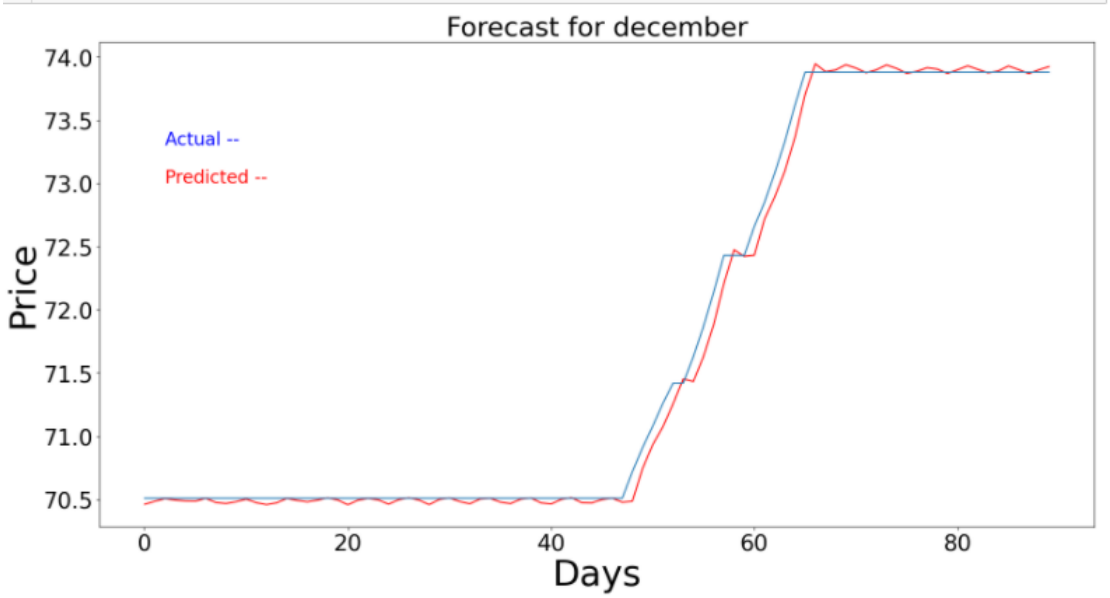
```
100%|██████████| 5/5 [10:36<00:00, 127.38s/it]
```

In [466]:
```
indices[np.argmin(rmse)]
```

Out[466]: (5, 4)

Application of final ARIMA Model on the data

After application of Test RMSE: 0.088. Final Application of **ARIMA Model** -



Conclusions and Interpretations –

Firstly, we analysed the dataset, and reinforced the fact that the data is not stationary using all the properties. When done we applied both the models –

Auto Regressive Model (AR)

Moving Averages Model - and forecasted the data for the completely unknown datapoints for next one month. We in the same applied hyperparameter tuning for forecasting.

After that we applied differencing, used auto corelation and partial autocorrelation and finally applied ARIMA, performed hyperparameter tuning and at last forecasted the price for the next 90 days for Delhi City.