**Visualizing and Forecasting of Stocks(SBIN)(NSE)**

**Introduction:**

**Investment:**

An investment is an asset or item acquired with the goal of generating income or appreciation. Appreciation refers to an increase in the value of an asset over time.

"Investing" we mean buying an asset for making a profit by selling it in the future, after it appreciates in value.

**Project Goal:** Visualizing and Forecasting of Stocks (NSE)

**Objective:** Maximize the Investment Returns

**Constraints:** Minimize the Investment Risk

# **NSEpy**

NSEpy is a library to extract historical and realtime data from NSE’s website. This Library aims to keep the API very simple.

Python is a great tool for data analysis along with the scipy stack and the main objective of NSEpy is to provide analysis ready data-series for use with scipy stack. NSEpy can seamlessly integrate with Technical Analysis library This library would serve as a basic building block for automatic/semi-automatic algorithm trading systems or backtesting systems for Indian markets.

## **NSE:**

*National Stock Exchange*

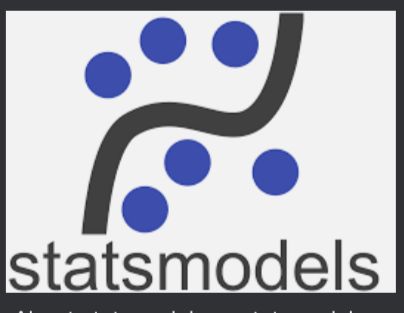
**Installing NSEpy:**

Pip install nsepy

**Data Collection:**

**Data Source:**

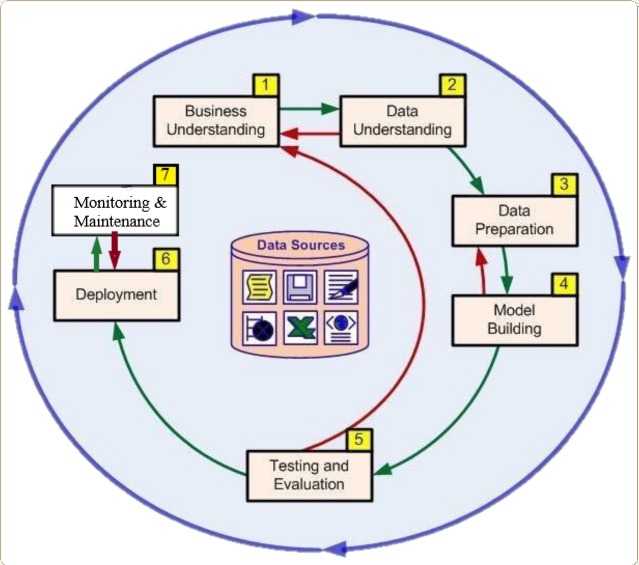
https://www.nseindia.com/get-quotes/equity?symbol=SBIN

**Technical Stacks:**

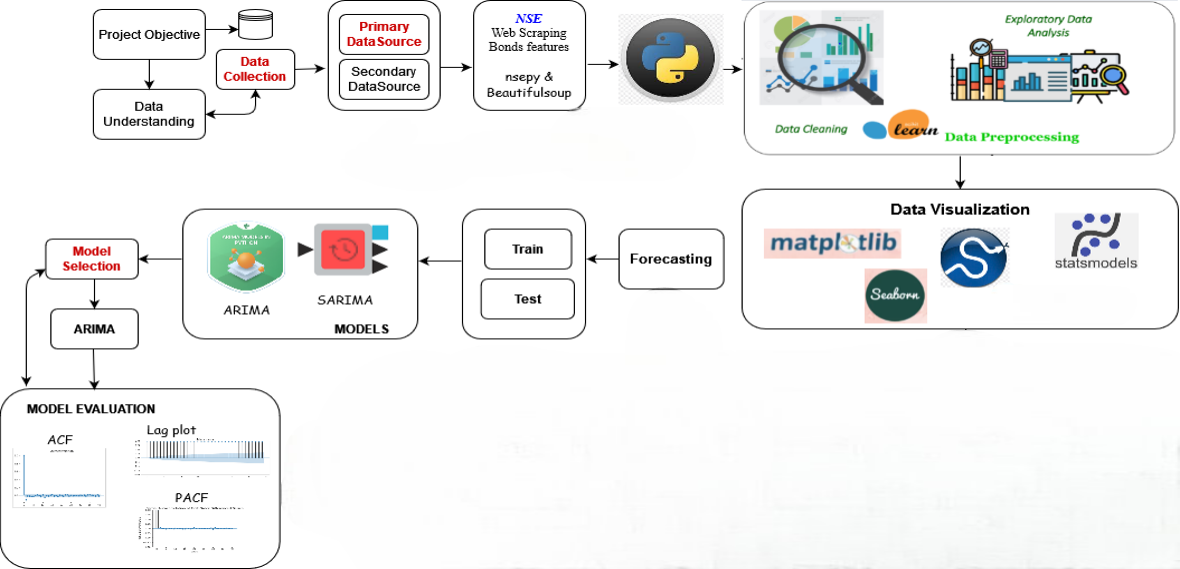
****

* Python is a general-purpose programming language. We used it for Data Cleaning, EDA, Model Building and Visualization.
* pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python
* Statsmodels is a Python package that allows users to explore data, estimate statistical models, and perform statistical tests.

**Data Pre-processing**

**CRISP-ML Methodology:**

**Project Architecture/Data Pipeline:**



**Data Understanding:**

|  |  |
| --- | --- |
| Name of Feature | Description |
| Date(Index) | Date format(YYYY/MM/DD) |
| Open / Price | Opening price of the bond |
| High | Highest Price of the bond |
| Low | Low Price of the bond |
| Close | Closing price of the bond |
| Change Pct | Change in the Open Price of Present-Day Price and Close Price of Previous day Price |

**EDA: Exploratory Data Analysis:**

* Exploratory Data Analysis (EDA) is **an approach to analyze the data using visual techniques**. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.

symbol = "SBIN"

start = date(2015, 1, 1)

end = date.today()

sbin = get\_history(symbol=symbol, start=start, end=end)

sbin

* Extracting data from 2015/1/1 to present date
* Initial dataset was having 1941 rows and 15 columns.
* The main constrain is stocks available for Monday to Friday, So last day of week prices are considering to next 2 days (Saturday, Sunday)

**Following steps were taken to perform Exploratory Data Analysis using Python:**

**important packages**

from nsepy import get\_history

from datetime import date

import pandas as pd

import numpy as np

import dtale as dt

from statsmodels.tsa.seasonal import seasonal\_decompose

from dateutil.parser import parse

import matplotlib.pyplot as plt

import seaborn as sns

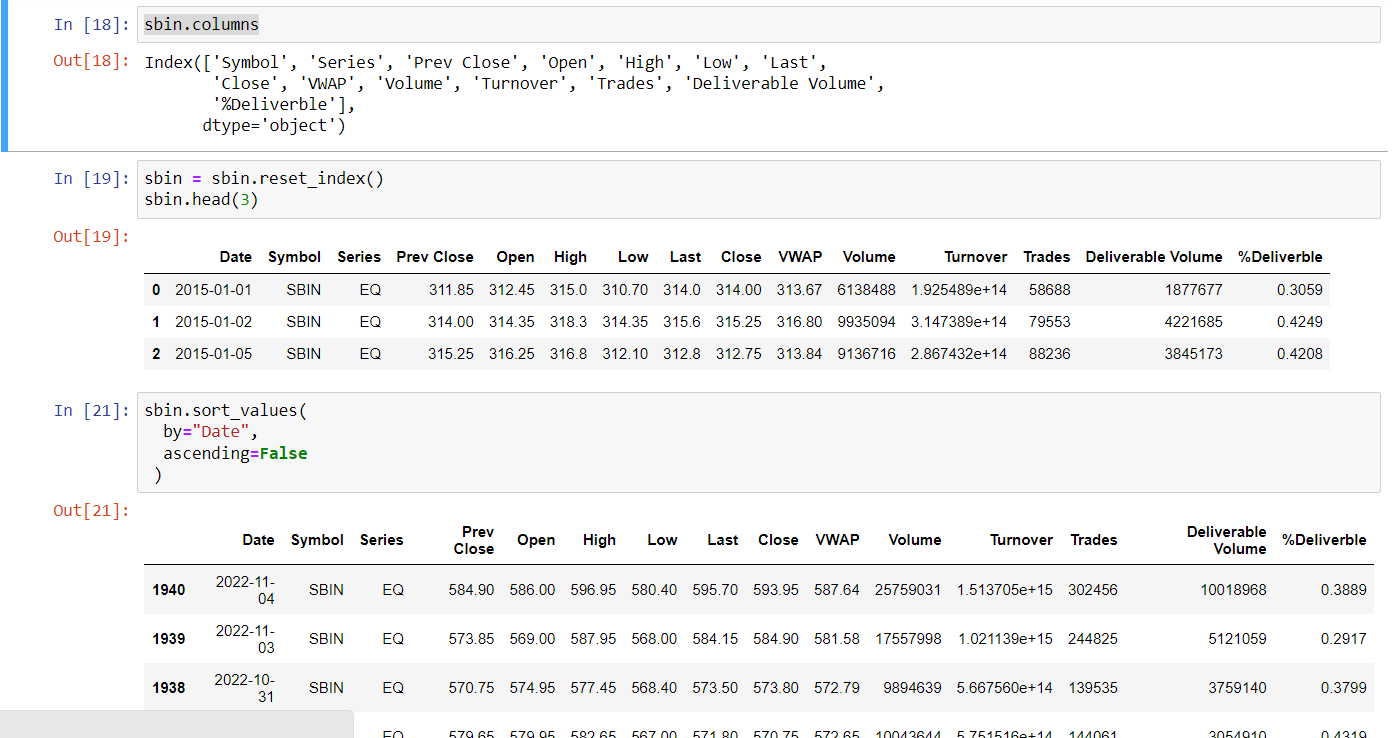
from sklearn.metrics import mean\_squared\_error

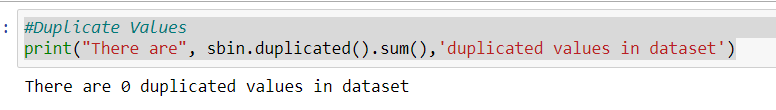
from statsmodels.tsa.arima\_model import ARIMA

from sklearn.metrics import mean\_squared\_error

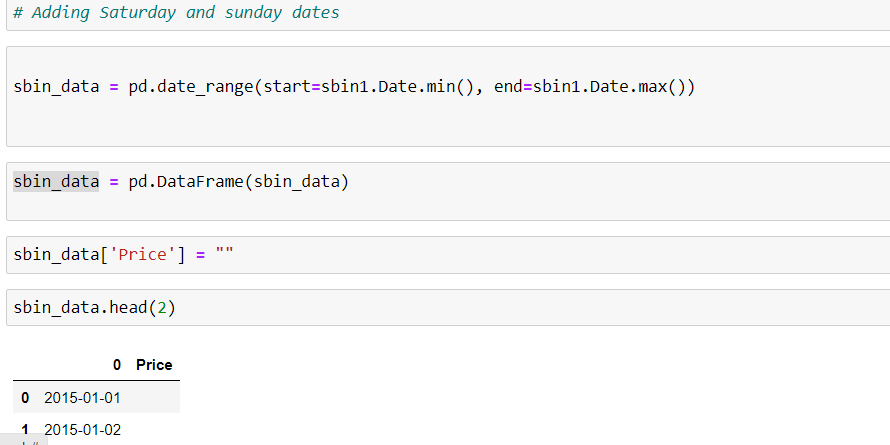
**Data Analysis:**

1. Getting column data
2. Resetting Index
3. Sorting data a/c to Date
4. Duplicates
5. Null Values

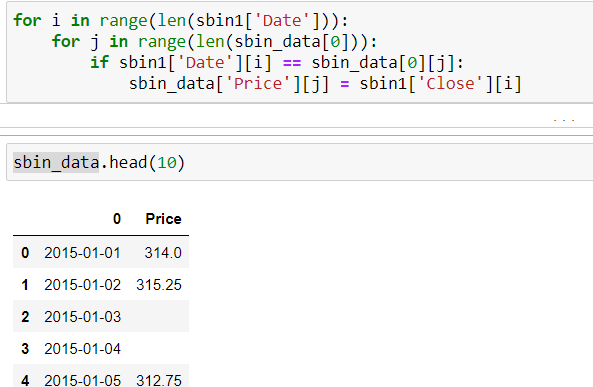


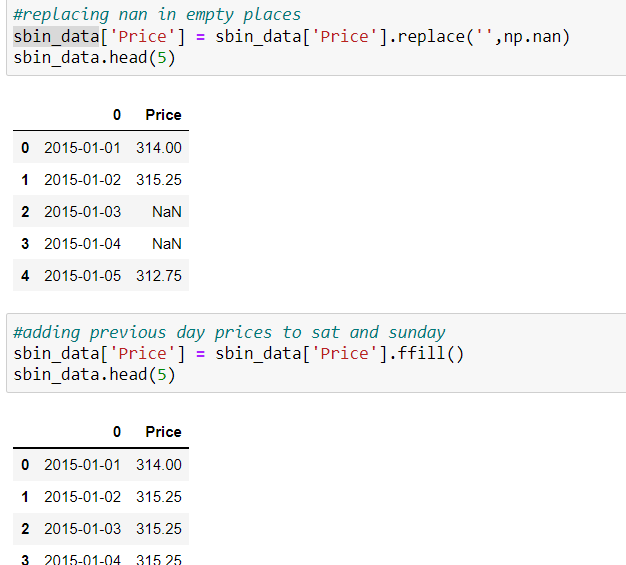


**Adding missing days:**

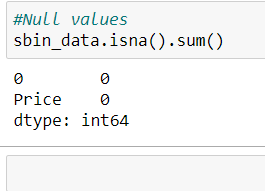
****

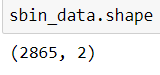
**Creating Data frame for Price and Date columns:**

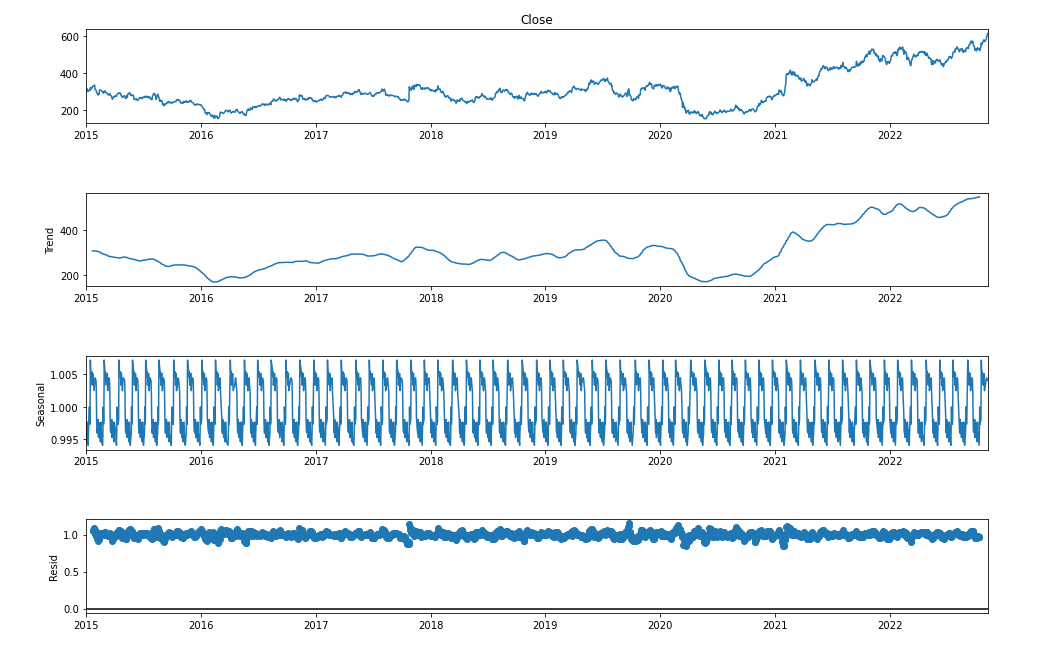
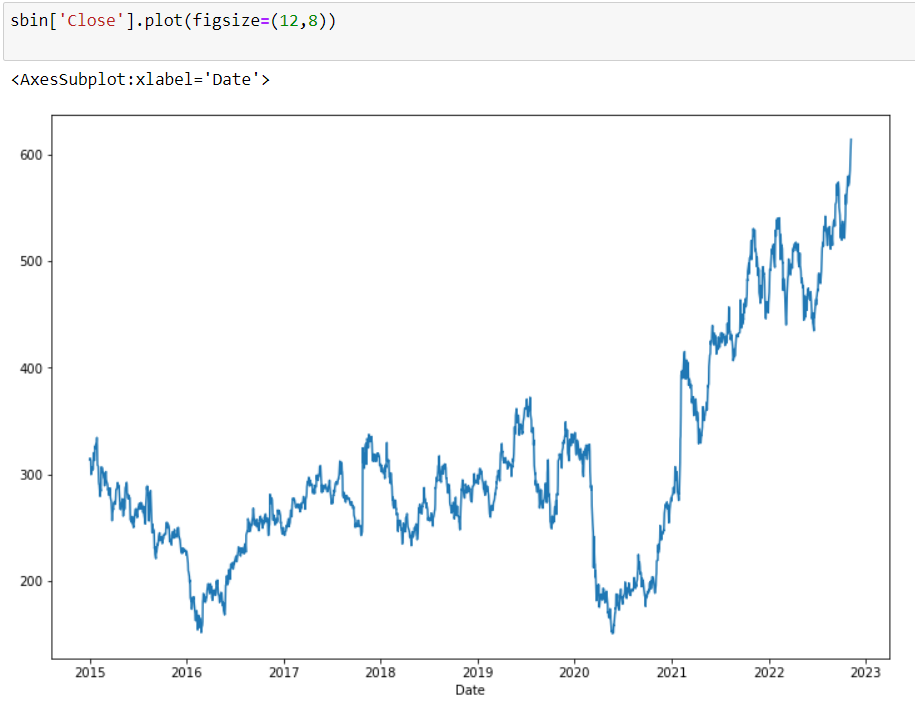


**Replacing Null Values in Empty Rows:**

**Count of Null Values:**

****

**Counting Rows and Columns:**

**Visualization:**

Plotting Price values a/c to year wise,

* Trend, as its name suggests, is the overall direction of the data.
* Seasonality is a periodic component
* Residual is what’s left over when the trend and seasonality have been removed. Residuals are random fluctuations. You can think of them as a noise component.
* By observing Plot Data is not stationary it is seasonal. We need to use the Seasonal ARIMA (SARIMA) model for Time Series Forecasting.

We need to use the Seasonal ARIMA (SARIMA) model for Time Series Forecasting on this data.

Stationary time series is one whose properties do not depend on the time

**Properties:**

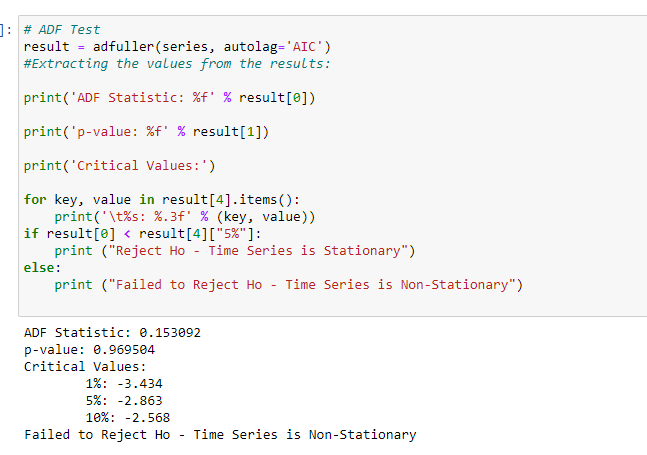
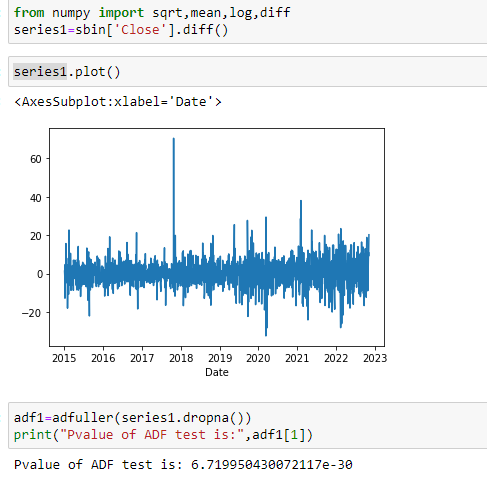
* Mean -- constant mean
* Variance -- variance should be constant with time
* Auto correlation -- correlation b/w to points depends on distance b/w 2points (lags b/w 2 points)

**Checking for stationary with Dickey-fuller Test**

* low Pvalue(lower than 0.05) implies series is stationary
* High PValue(greater than 0.05)implies not stationary

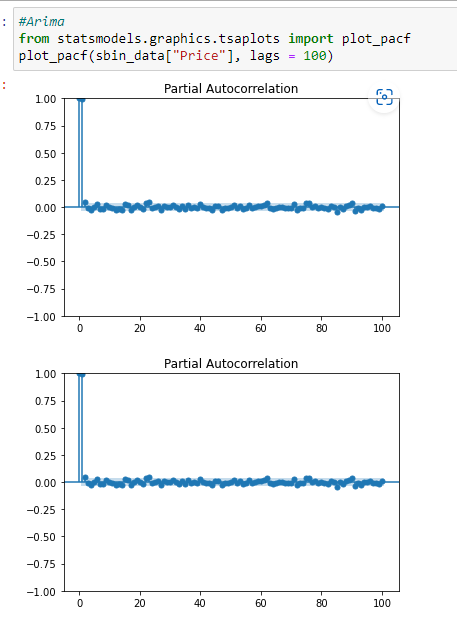
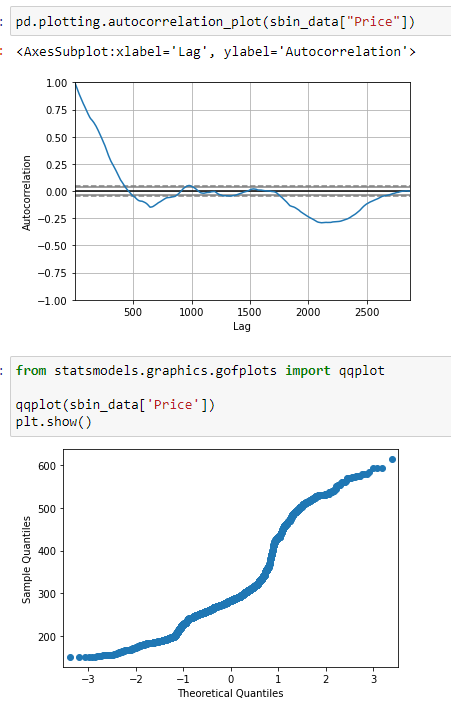
**Import Adfuller Library**

* + - from statsmodels.tsa.statstools import adfuller
    - from statsmodels.tsa.stattools import adfuller#library for finding d



From Augmented Dickey-Fuller unit root test, P value is 0.96<0.05 implies stationary due to trend. For making stationary differentiating ADF we can remove trend.

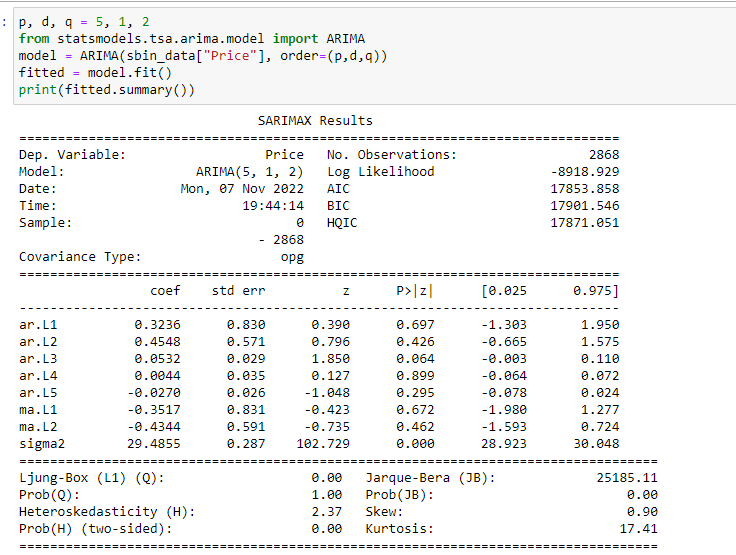
Autocorrelation is the correlation between two observations at different points in a time

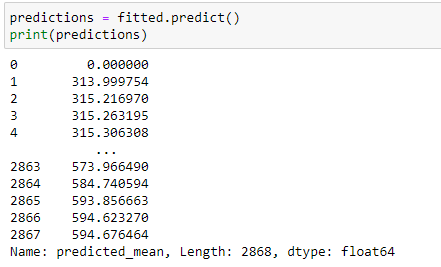
series.

* In the above autocorrelation plot, the curve is moving down after the 5th line of the first boundary. That is how to decide the p-value. Hence the value of p is 5.
* In the above partial autocorrelation plot, we can see that only two points are far away from all the points. That is how to decide the q value. Hence the value of q is 2.

**ARIMA(Autoregressive Integrated Moving Average)**

* Time Series Forecasting means analyzing and modeling time-series data to make future decisions.
* Arima is one of the statistical method for forecasting Time series data
* ARIMA models have three parameters like ARIMA(p, d, q).
* p is the number of lagged values that need to be added or subtracted from the values (label column). It captures the autoregressive part of ARIMA.
* d represents the number of times the data needs to differentiate to produce a stationary signal. If it’s stationary data, the value of d should be 0, and if it’s seasonal data, the value of d should be 1. d captures the integrated part of ARIMA.
* q is the number of lagged values for the error term added or subtracted from the values (label column). It captures the moving average part of ARIMA.



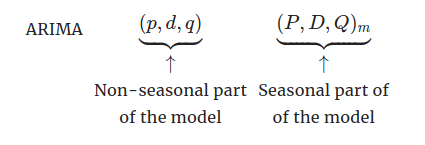


he predicted values are wrong because the data is seasonal. ARIMA model will never perform well on seasonal time series data. So, here’s how to build a SARIMA model:

**SARIMA model**

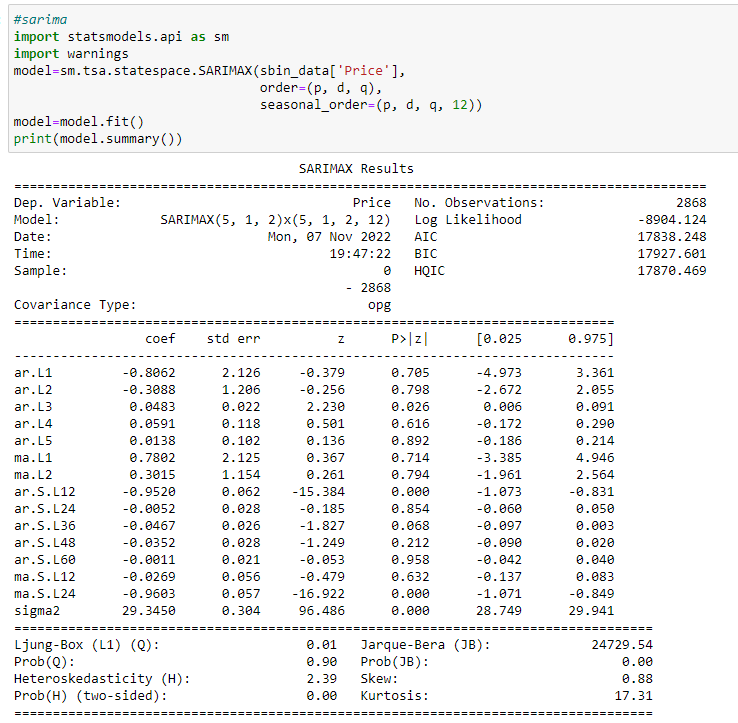
SARIMA stands for Seasonal-ARIMA and it includes seasonality contribution to the forecast. The importance of seasonality is quite evident and ARIMA fails to encapsulate that information implicitly.

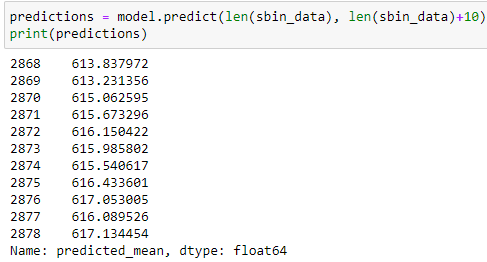
The Autoregressive (AR), Integrated (I), and Moving Average (MA) parts of the model remain as that of ARIMA. The addition of Seasonality adds robustness to the SARIMA model. It’s represented as:



where m is the number of observations per year. We use the uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

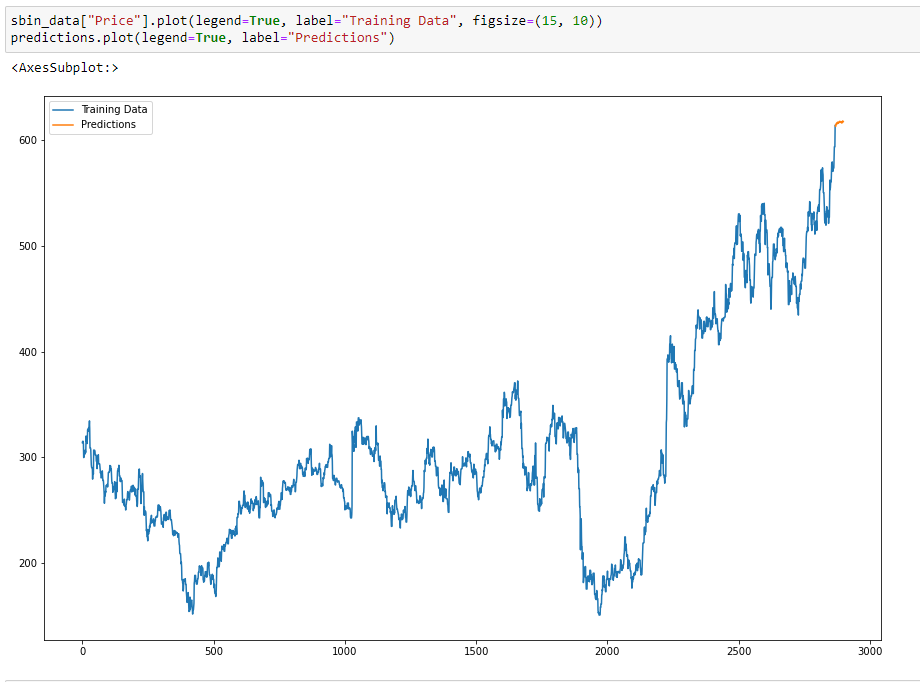
Similar to ARIMA, the P,D,Q values for seasonal parts of the model can be deduced from the ACF and PACF plots of the data.





Predicted values for next 10days with SARIMA model.

**plotting the predicted values:**



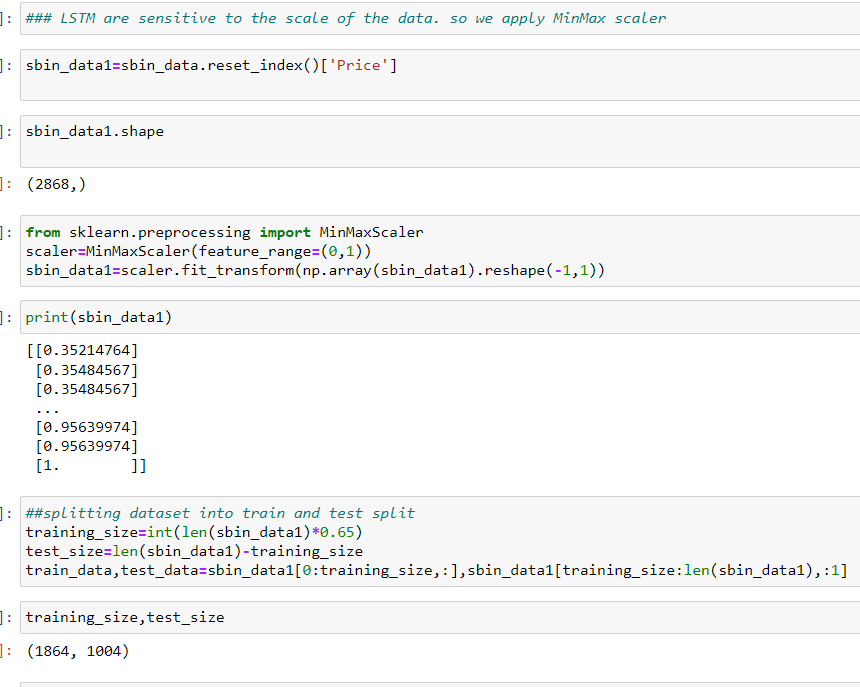
**Conclusion:** ARIMA is an algorithm used for forecasting Time Series Data. If the data is stationary, we need to use ARIMA, if the data is seasonal, we need to use Seasonal ARIMA (SARIMA).

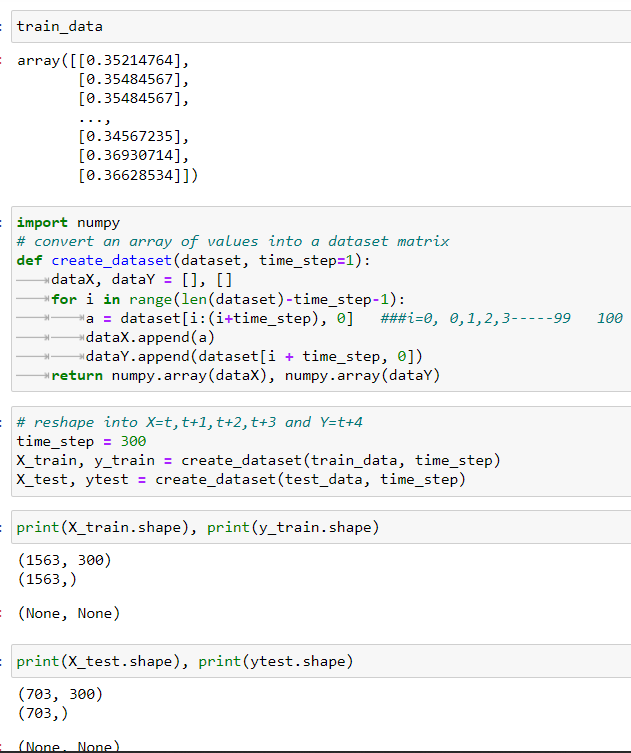
By using ARIMA( SARIMA) model forecasted next 10days values

The ARIMA algorithm will be a great asset for brokers and investors for investing money in the stock market since it is trained on a vast collection of historical data and has been chosen after being tested on a trial data.

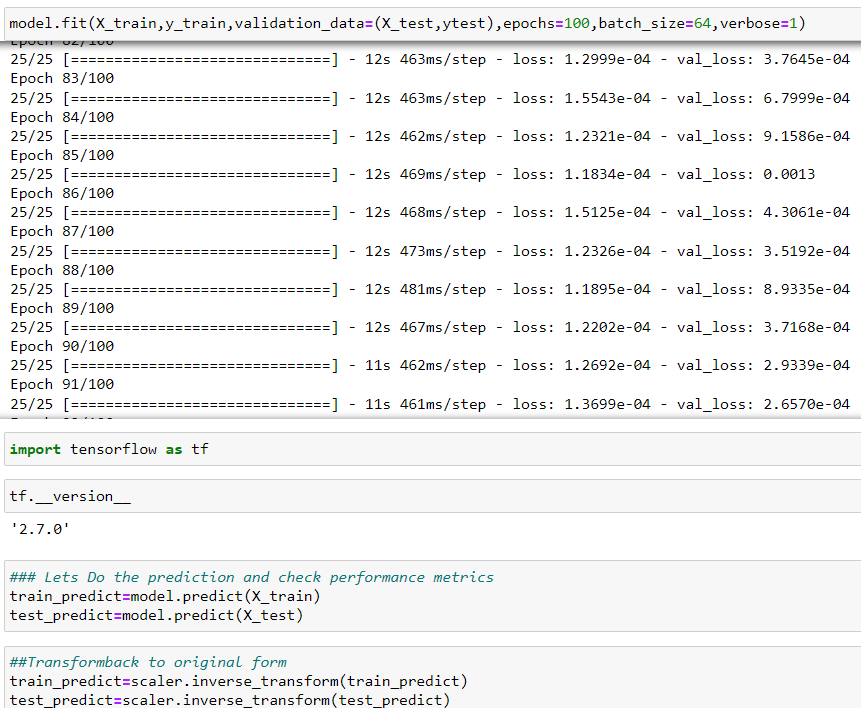
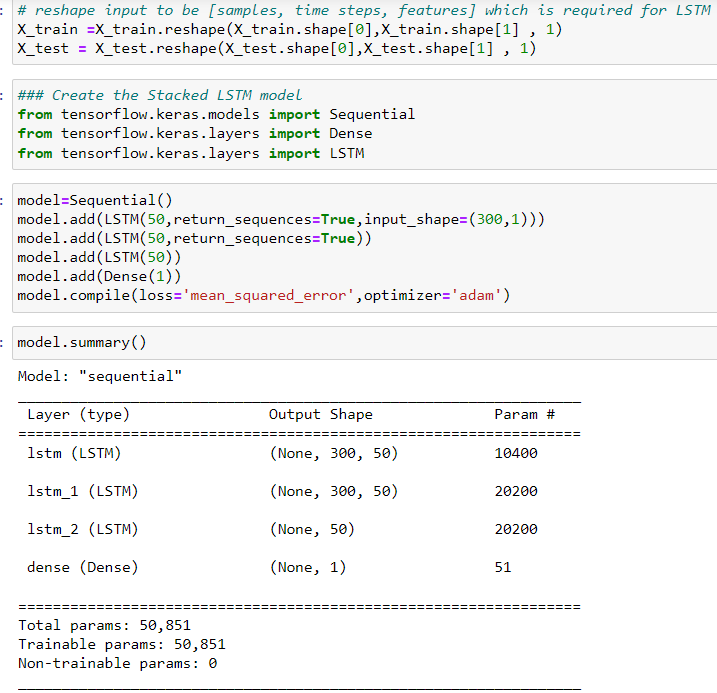
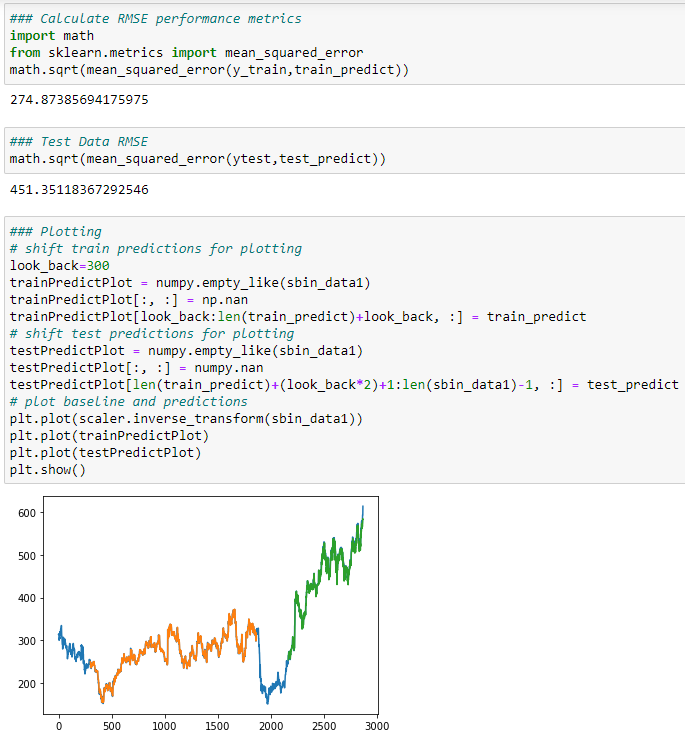
The project demonstrates the machine learning model to predict the stock price with more accuracy as compared to other machine learning models.

**LSTM model**

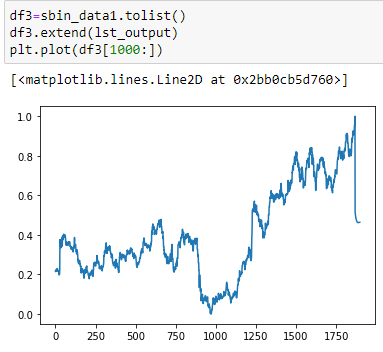
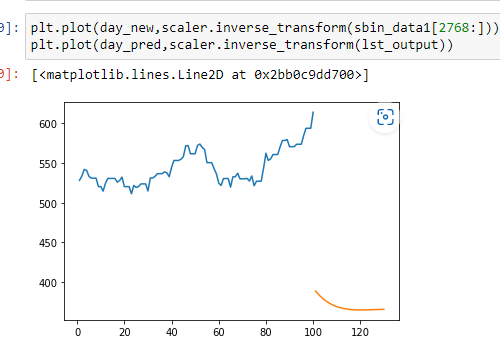
LSTM stands for Long-Short Term Memory. LSTM is a type of recurrent neural network but is better than traditional recurrent neural networks in terms of memory. Having a good hold over memorizing certain patterns LSTMs perform fairly better.



Splitting data set into training and testing, and reshaping the input to be samples, time steps, features required for LSTM.



Predicting[close] Price values for next 30 days

Visualizing Forecasted data:

**Conclusion:**

Forecasted Future 30 days Close(price) by using LSTM model.

Predicting stock market returns is a challenging task due to consistently changing stock values which are dependent on multiple parameters which form complex patterns.

**Conclusion**

By comparing LSTM and ARIMA(SARIMA) models, ARIMA model predicted good accurate price values, So for SBIN(State Bank of INDIA)(NSE) stocks concluded ARIMA is finalized for SBIN Stocks.

Thank You