Comparative Study of ARIMA Models for Stock Market Prediction

A PROJECT REPORT

Submitted for the partial fulfillment

of

Project Based Learning (PBL) requirement of B. Tech CSE

Submitted by

1. Sanskruti Nerkar, 22070521028
2. Arushi Shivhare, 22070521062
3. Devyani Balki, 22070521051

B. Tech Computer Science and Engineering

Under the Guidance of

Dr. Nitin Rakesh

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS
INSTITUTE OF TECHNOLOGY, NAGPUR

Wathoda, Nagpur

2025

CERTIFICATE

This is to certify that the Capstone Project work titled “Time Series Forecasting of Stock Prices using ARIMA” that is being submitted b**y Sanskruti Nerkar, 22070521028, Arushi Shivhare, 22070521062, Devyani Balki, 22070521051** is in partial fulfillment of the requirements for the Project Based Learning (PBL) is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma, and the same is certified.

Name of PBL Guide & Signature
Dr. Nitin Rakesh

Verified by:
Prof. Monali Gulhane

Dr. Sudhanshu Maurya
PBL Coordinator

The report is satisfactory/unsatisfactory

Approved by

Dr. Nitin Rakesh
Director, SIT Nagpur

ABSTRACT

The work done in this project is directed towards analysing and forecasting the stock market trends using ARIMA (auto regressive integrated moving average) model, which is a highly efficient and commonly used statistical technique in time series forecasting. There are many dynamic factors reflecting stock market and if it is taken in combination of all makes everyone to struggle to make accurate forecast about the market. In the first part, we retrieve historical stock price data and preprocess it to remove noise, deal with missing values and achieve stationarity (which is a necessary condition in order for ARIMA modeling). Finally, the patterns are identified using the ARIMA to capture the dependencies on time and to forecast future stock prices.

The project's core goal is to build a machine learning based data driven solution to allow stock price parlances to be forecasted with a very high degree of accuracy through forecasting stock price movements. It is implemented in Python and uses Pandas, NumPy, Matplotlib and stats models libraries. Metrics available to evaluate the performance of the graph model include Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to ensure the accuracy and reliability of the model. In addition, this project provides a proof of concept regarding the practical use of the ARIMA model in analyzing real world stock market and the ARIMA model in forecasting financial time series; it also provides a pathway for future research into hybrid and deep learning-based forecasting methods.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

However, stock market is a very volatile and nonlinear system which is controlled by combination of macroeconomic indicators, events happening in the geopolitical world, performance of the companies, psychological factors which control the stock market. Classifying stock price trends has always been difficult and usually depended on how much one paid attention to expertise and financial intuition. Nevertheless, the expansion of the available data and acceleration of the development in the computational capabilities in the financial forecasting field have seen a turnover to the data driven modeling techniques. History wise financial data is increasingly being used for Machine Learning and also for statistical time series models to find hidden patterns and create predictive analytics.

One type of time series forecasting methods out of various such models, ARIMA (AutoRegressive Integrated Moving Average) model is one of the strongest and also most interpretable methods. It is especially suitable for univariate time series data for which the temporal structure is to be understood and future values to be generated. ARIMA combines three key components. ARIMA successfully captures the autocorrelation, trend and noise in the dataset with the tuning of parameters (p, d, q).

The emphasis of this project is examining the use of the ARIMA model in making a prediction on stock market data. Data collection from publicly available financial APIs or CSV files, Data Preprocessing such as handling the missing values, noise reduction and normalization, and then presence of stationarity through ADF test is critical steps in the pipeline. Finally, ACF and PACF are also used to pick the values of optimal AR and MA components. Learning and validation of the model is then carried out using real world stock price datasets, and the results are analyzed using statistical errors measures like the Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

1.1 Objectives

The below mentioned are the objectives of this project:

- To do collection and preprocessing of historical stock price data to ensure data quality and consistency for time series analysis

- Assess the stationarity of the time series using statistical tests such as the Augmented Dickey-Fuller (ADF) and KPSS tests

- Determine optimal ARIMA parameters (i.e. p, d, q) through ACF and PACF plots

- Develop and train the ARIMA model on the processed time series data for more accurate short-term stock price forecasting

- Measure model performance using error metrics like the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Mean Absolute Percentage Error (MAPE)

- Visualize the forecasted values against actual stock prices to interpret the model’s accuracy and reliability

- Identify the limitations of the ARIMA model in capturing market volatility and external factors

- Suggest potential model enhancements, such as the integration of ARIMA with ML or deep learning techniques for improved accuracy.

1.2 Literature Survey

AUTHOR & Year	TITLE	METHODOLOGY	ACCURACY	OBSERVATIONS
Muhammad Najamuddin, Samreen Fatima Year: 2022	A Hybrid ARIMA-RNN Model for Financial Time Series Forecasting	Hybrid ARIMA-RNN	hybrid BRNN-ARIMA model showed higher accuracy than standalone ARIMA, BRNN, and random walk models, evaluated using RMSE, MAE, and MAPE. Exact numerical values vary by country, but the hybrid model consistently had lower error metrics across all tested datasets.	Hybrid model outperforms individual ARIMA or RNN models; suitable for data with mixed dynamics
Hyeong Kyu Choi Year: 2018	Forecasting Stock Price Correlations Using ARIMA-LSTM Hybrid Models	Hybrid ARIMA-LSTM	It demonstrated superior predictive ability by effectively capturing both linear and nonlinear patterns in the data	Hybrid model is more accurate than traditional correlation methods, especially during high volatility
Zhuangwei Shi, Yang Hu, Guangliang Mo, Jian Wu arXiv Year: 2022	An Ensemble Forecasting Framework Using Attention-Based CNN-LSTM and XGBoost	Attention-based CNN-LSTM + XGBoost	This hybrid model integrates ARIMA, Attention-based CNN-LSTM, and XGBoost to predict stock prices. It effectively captures both linear and nonlinear dependencies in stock market data	The ensemble model shows superior performance compared to single models due to its comprehensive structure.

Jian Wu arXiv Year : 2022	Two-Phase Forecasting Using ARIMA and XGBoost for Stock Market Prediction	Two-phase (ARIMA + XGBoost)	The model outperformed traditional methods, demonstrating its effectiveness in mining historical stock information across multiple periods.	The model effectively handles anomalies and improves accuracy over traditional ARIMA.
Brahmanapalli Kalyan, S Parameshwara Reddy, Dr. Krovvidi Krishna Kumari, Dr. Manish Jain Irjaeh Year: 2024	Comparative Study of ARIMA, LSTM, and Random Forest for Financial Forecasting	ARIMA, LSTM, Random Forest (Ensemble)	Evaluation metrics included Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).	Ensemble models adapt better to varying data conditions than single models. To evaluate and compare model adaptability across different market behaviors.
Muhammad Najamuddin, Samreen Fatima Year : 2017	Forecasting Stock Prices: A Comparison of ARIMA and ANN Models	ARIMA, Artificial Neural Networks (ANN)	ARIMA: Mean Absolute Error (MAE) of 29.6975 ANN: MAE of 4.7084	ARIMA performs better on linear data; ANN captures nonlinear patterns more effectively.
Sima Siami-Namini and Akbar Siami Namin Year : 2018	A Comparative Study Between ARIMA and LSTM for Long-Term Economic Forecasting	ARIMA, LSTM	LSTM outperforms ARIMA in long-term forecasting accuracy.	LSTM significantly outperforms ARIMA in long-term forecasting due to its memory-based architecture.
akshi Kulshreshtha and A. Vijayalakshmi Year : August 2020	ARIMA-LSTM Hybrid Model for Stock Price Forecasting	ARIMA-LSTM Hybrid	The hybrid ARIMA-LSTM model showed significant improvements in accuracy across various forecasting applications.	The hybrid model improves prediction accuracy over individual models

Mochamad Ridwan, Kusman Sadik, and Farit Mochamad Afendi Year : August 2023	Predicting Market Trends Using ARIMA and GRU Models	ARIMA, Gated Recurrent Unit (GRU)	GRU: Mean Absolute Percentage Error (MAPE) of 0.77% for BMRI stock ARIMA: MAPE of 4.09% for BMRI stock Conclusion: GRU model outperforms ARIMA in forecasting accuracy.	GRU outperforms ARIMA in capturing long-term dependencies and adapting to changing patterns.
Yingying Feng, Ruochen Xiao, Lei Yan, and Yihan Ma Year : August 2022	ARIMA vs LSTM: Which is Better for Stock Forecasting	ARIMA, LSTM	LSTM is significantly better than ARIMA in terms of fitting results for stock index opening price prediction.	LSTM performs better in volatile markets; ARIMA still useful for short-term, stable series.
M. Kumar and M. Thenmozhi Year : 2014	Combining ARIMA and Random Forest for Enhanced	ARIMA + Random Forest	Hybrid models combining ARIMA with Random Forests achieve high prediction accuracy and outperform simpler models.	Residual modeling helps correct ARIMA's limitations, improving
Yihan Ma and Lei Yan Year : 2016	Stock Market Prediction	Better predictions. Stock price data	This study developed three hybrid models by combining the Autoregressive Integrated Moving Average (ARIMA) model with Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Random Forest (RF) to predict stock index	ARIMA's limitations, improving overall performance

			returns.	
S Siami-Namini, N Tavakoli, AS Namin Year : 2019	Forecasting Financial Time Series with ARIMA-LSTM Architecture	ARIMA-LSTM	The ARIMA-LSTM hybrid model yields better results than individual methods in financial time series forecasting.	Achieves better accuracy in forecasting compared to standalone models.
KA Althelaya, ESM El-Alfyei Year : 2018	Stock Market Prediction Using LSTM and GRU Networks	LSTM, GRU	LSTM and GRU models can predict stock prices with an accuracy of 92.36%.	GRU slightly faster; both outperform traditional models in long-term forecasting.
Y Wang, Y Guo Year : 2020	Volatility Forecasting Using Hybrid ARIMA-XGBoost Approach	ARIMA + XGBoost	The hybrid ARIMA-XGBoost model improves modeling and forecasting accuracy over ARIMA and ANN models for high-frequency time series data	Hybrid model effectively manages extreme market movements and anomalies
R Zanc, T Cioara, I Anghel Year : 2019	Ensemble Forecasting for Financial Markets Using Deep Learning and ARIMA	ARIMA, LSTM, Ensemble	The ensemble forecasting procedure integrating LSTM and ARIMA models yields better results than individual methods	Ensemble approach generalizes better across different time series and market types.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- **Chapter 2: Existing System, Proposed System, Software, and Hardware Details**

The proposed ARIMA based solution is given in this chapter and an overview of existing stock market forecasting systems along with the limitation of traditional methods is given. The hardware and software requirements for installation of the project are also included in it.

- **Chapter 3: Implementation of the Project**

The step-by-step method of implementation of the ARIMA model for stock price prediction is described in this chapter which includes data collection, preprocessing, training the model, and evaluating the same.

• **Chapter 4: Results and Discussion**
In this chapter, the performance of ARIMA model is shown along with the evaluation metrics like MAE, RMSE, MAPE. The model is discussed based on real world data related to the accuracy and limitations of the model.

• Chapter 5: Conclusion and Future Scope
The main conclusions of the project are summarized in this chapter, its limitations are discussed and improvements are outlined. Furthermore, it also provides future directions such as possible further extension to the hybrid models or the deep learning based approaches.

• Chapter 6: Code Implementation
The complete code used to develop and train the ARIMA model for stock price forecasting are all there in this chapter. It not only gives an in depth look of the tech implementation using libraries like Pandas, NumPy, stats models, Matplotlib, etc. it also shows how one can implement a pipeline using this logic in the actual use case.

• Chapter 7: References
This chapter contains all the references, the current research papers, articles and documentation that have been used during the project.

CHAPTER 2

AN EFFECTIVE CLASSIFICATION OF HEART DISEASE PREDICTION SYSTEM USING ML

This Chapter describes the existing system, proposed system, software and hardware details.

2.1 Existing System

Financial research and Quantitative finance has been an active field for stock market forecasting for decades. Predictions of stocks price and market trends are made using many existing systems and approaches. Some of the widely used methods are given below:

1. **Linear Regression Models:** Linear regression has been traditionally used to model stock price movement since it assumed a linear relationship between the stock prices and other variable that is a predictor of those prices. Nevertheless, this procedure rests upon a condition of a constant relationship over time, which does not describe the dynamic characteristics of any financial market. Linear regression can just solve part of the problem, as can unable to learn to capture the complexities and volatility in financial time series inaccurate predictions.
2. **Moving Averages (MA):** 2. In general, you use the Moving Averages to smoothen out short term fluctuations in the price of a stock and hint at the ongoing trends. Lastly, some popular indicators which traders use are the resulting indicators from a simple moving average (SMA) and exponential moving average (EMA). However, these models are simple and do not do so; they disregard the historical autocorrelation of data. However, moving averages are a reactive method and lags behind the market trends; therefore, they are vulnerable to predict the future stock prices accurately, particularly during the volatile and sudden activity of a market.
3. **Autoregressive Integrated Moving Average (ARMA):** ARMA is an older model dated back, when the series is already stationary, is AIRMA model. It is an autoregressive (AR) and moving average (MA) model to predict future values. Nevertheless, the time series data of a stock market cannot be included in the ARMA model as it is not usually stationary. ARIMA utilizes the differencing component (I) that is missing in ARMA to deal with non stationary data.
4. **Artificial Neural Networks (ANN) and Deep Learning Models:** In recent years, more and more people have begun to consider the stock price prediction with the help of neural networks, especially the deep learning models, such as Long Short-Term Memory (LSTM) networks. These models can help in capturing a wide spectrum of dependencies of data. Unfortunately, however, deep learning models require very huge amounts of data and computational resources to train. As well, they are vulnerable to overfitting, thus being highly susceptible, and they may suffer from lack of interpretability, which prevents the users from understanding the reasoning of the predictions.
5. **Support Vector Machines (SVM):** One other machine learning method used for stock market prediction is SVMs. These can represent non-linear relationship and have been proven useful in the tasks of classifying given a stock price whether it will go up or down. Despite this, SVMs are not specialized to perform forecast of time series, while they do reasonably with certain inputs, they are computationally expensive and may not deal with temporal dependence present in the stock market data as ARIMA does.

Comparison of ARIMA with Existing Systems:

In relation to the financial markets, for it offering several advantages to that of a traditional and machine learning based time series forecasting method, the ARIMA (Auto Regressive Integrated Moving Average) model is a better choice.

1. **Handling Non-Stationary Data:** ARIMA on the other hand does not need stationary data unlike the ARMA model as it incorporates a differencing parameter (I). Typically, stock prices exhibit a trend (upward or downward movement) and seasonality, and in such cases, the choice of ARIMA should be preferred over ARMA or the linear regression, as models of the last two cannot accommodate the former.
2. **Autoregressive and Moving Average Components:** In both autoregressive (AR) and moving average (MA) models, we have historical dependencies in the stock prices and

noise respectively, ARIMA is capable of seamlessly fitting with both of them, resulting in ARIMA. This renders ARIMA a strong candidate to take recurrent changes of market behaviour into account for stock price forecasting. However, while linear regression or moving averages cannot capture these relationships and their predictions are simple.

3. **Simplicity and Interpretability:** ARIMA is simply and interpretably. Being overall easy to tune and interpret, the model consists of only three parameters p (autoregressive order), d (degree of differencing), and q (moving average order), indeed. However, the neural networks or SVMs, for instance, that are machine learning models are much more complex and still require the tuning and great computational resources. Furthermore, these models are often not transparent, so analysts cannot understand the reasons that predicated the given values.
4. **Lower Data and Computational Requirements:** Even with smaller datasets, ARIMA performs well as the only component it needs is data from back of the same time series. Whereas, Deep learning models like LSTM or Artificial neural networks require huge amount of data for training and large computational resources. This then makes ARIMA a more practical solution for financial institutions and analysts interested in case studies with limited historical data or someone looking to utilise a lightweight solution.
5. **Performance with Stationary Markets:** Although ARIMA may not be up to par when markets are high, high, and tail, ARIMA does a pretty good job when market conditions are stable or stationary, and trends and patterns are in fact consistent. On the other hand, machine learning models might overfit the market noise or lack good generalization in such conditions and suffer poorer performance.
6. **Overfitting and Generalization:** One challenge with machine learning models like ANN or SVM is that there is significant chances of overfitting peculiar to the model attempting select curves to fit the noise rather than the actual pattern that exists. A great thing about ARIMA is that it is a statistical model, thus reducing its tendency to over fit, and it is designed to partly account for trends, cycles and noise.
7.

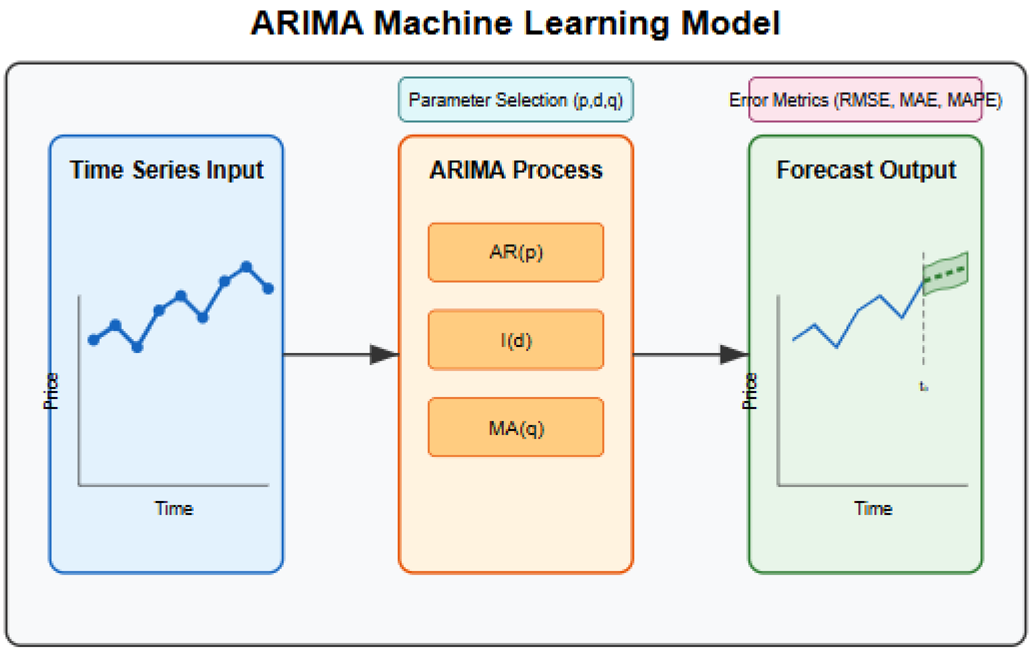

ARIMA Machine Learning Model
Parameter Selection (p,d,q)
Error Metrics (RMSE, MAE, MAPE)
Time Series Input
Price
Time
ARIMA Process
AR(p)
I(d)
MA(q)
Forecast Output
Price
Time

Fig.1. ARIMA ML Model

2.2 Proposed System

The data analysed to predict the stock market trends and future stock prices is the ARIMA (AutoRegressive Integrated Moving Average) model which is a widely used statistical approach for time series forecasting. ARIMA unlike some other machine learning models is transparent, interpretable and is very powerful for linear time dependent data, and is therefore ideal for financial market analysis, as historical patterns often shape future movements. The pipeline structure of the system is defined, that is, data acquisition, preprocessing, stationarity testing, model parameter selection, training, forecasting and performance evaluation.

Then, the data acquisition phase takes input from reliable sources for historical stock market data which includes time series data for example Yahoo Finance, Alpha Vantage API and others, and we store it in a database. For example, in most dataset you will find that they have features as Date, Open, High, Low, Close and Volume. Among these, the ‘Close’ price is selected as a principal variable of analysis, which is the price at which the stock was closed and is usually used in such financial prediction tasks.

So the first great step is to acquire the data. Most financial data is raw and is full of missing values, noise or inconsistency that has to be cleaned or treated before model training. Firstly, preprocessing involves dropping or imputing the missing data using statistical means like forward fill or interpolation and moving the date column into datetime format; secondly, the data should be sorted as chronologically as possible. Finally, the predictive modelling is also checked for outliers and anomalies that potentially distort the model. The dataset is turned into a univariate series by only retaining the ‘Close’ price for time series analysis.

The data is then processed and a stationarity check is done before being run through ARIMA modelling, which is a requirement. A stationary time series has constant mean, variance over time and does not have trends or seasonality. A first step is to use Augmented Dickey Full (ADF) test to see if the data need to be transformed since most of the financial time series, like the stock price, are naturally non stationary. If the ADF test does not reject the null hypothesis that the time series is non-stationary, then the time series is done with the differencing process. This is repeated (using first order or second order differencing) to achieve stationarity. So the transformation in this instance enables the model to pick up on the underlying patterns in the data to the exclusion of external trends or seasonal fluctuations.

Once the data becomes stationary, the system proceeds to the **parameter selection** phase. ARIMA models have 3 primary parameters:

- **p (autoregressive term)** – captures the existing relationship between an observation and its lagged observations.
- **d (degree of differencing)** – determine how many times the data has to be differenced to achieve stationarity.
- **q (moving average term)** – captures the relationship between an observation and residual errors from previous forecasts.

ACF and PACF plots are generated to identify the best values of parameters in this problem. These plots are helpful to find out the significant lags in the data and thus specify the most suitable values of p and q. Furthermore, DES selects a DES from such configurations based on statistical criteria such as Akaike Information Criterion (AIC) and Bayesian Information Criterion

(BIC), so as to achieve the best fit of model to the data as much as possible with the minimum error.

Once the optimal parameters are selected, the ARIMA model is trained with the proceeding historical data which is processed. The model learns the underlying temporal dependencies, trends as well as noise patterns in the data during training. Once trained, it is used to produce forecasts into future stock prices. Depending on the range of the analysis in terms of time, these forecasts can be short term, for instance next day or next week, or long term.

After generating forecast, the system enters the evaluation phase, where accuracy and reliability of the predictions are measured using standard error metrics such as:

- **Mean Absolute Error (MAE)** – measures the average magnitude of the errors in a set of predictions.
- **Root Mean Squared Error (RMSE)** – gives more weight to large errors and is sensitive to outliers.
- **Mean Absolute Percentage Error (MAPE)** – expresses accuracy as a percentage of the actual values.

These metrics help in defining how much the ARIMA model has fit the dynamics of the stock price. In addition, time series plots on the actual and predicted values are visualized to provide a clear and interpretable manifestation of model performing.

The system final output is the forecasted stock price chart which can help investors and analysts anticipate the right price chart for their investment and analysis. While ARIMA is an inherently linear model and may not fare so well in capturing the nonlinearity of the market or market shocks such as sudden price drops, it is quite strong in stable market conditions and gives very quick, interpretable, and statistically justified predictions.

Summarily, the proposed forecasting system based on ARIMA has statistical rigor and simplicity, with effectiveness. It is especially useful for organizations and individuals who wish to analyse historical stock trends as well as to predict near future, without large computing infrastructure. With ADF testing, ACF/PACF analysis, parameter tuning and forecast evaluation integration, the system depicts that the system is robust, accurate and reliable for stock market analysis. Thus, the working of the proposed system is shown below in the flow chart (Fig.2.).

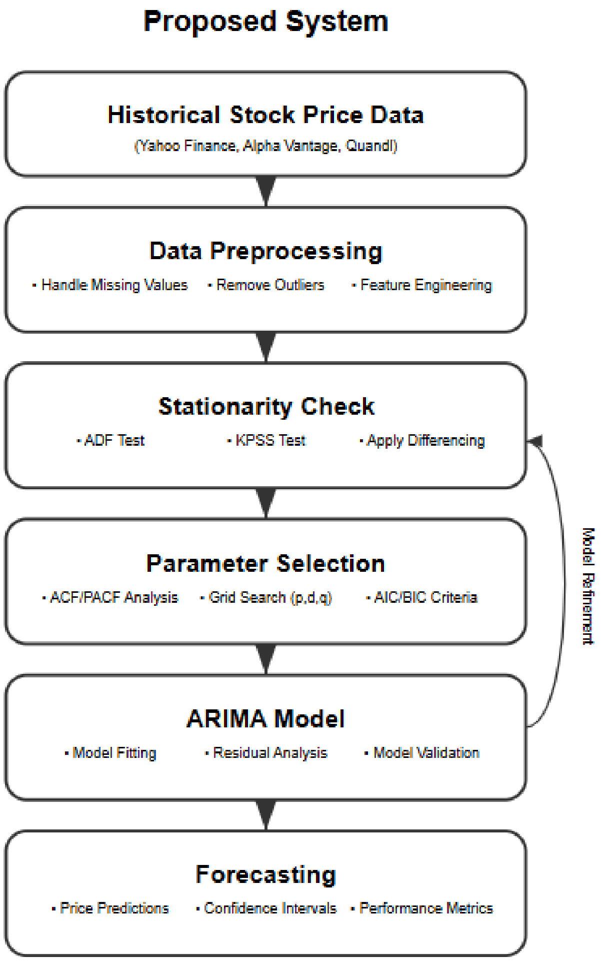


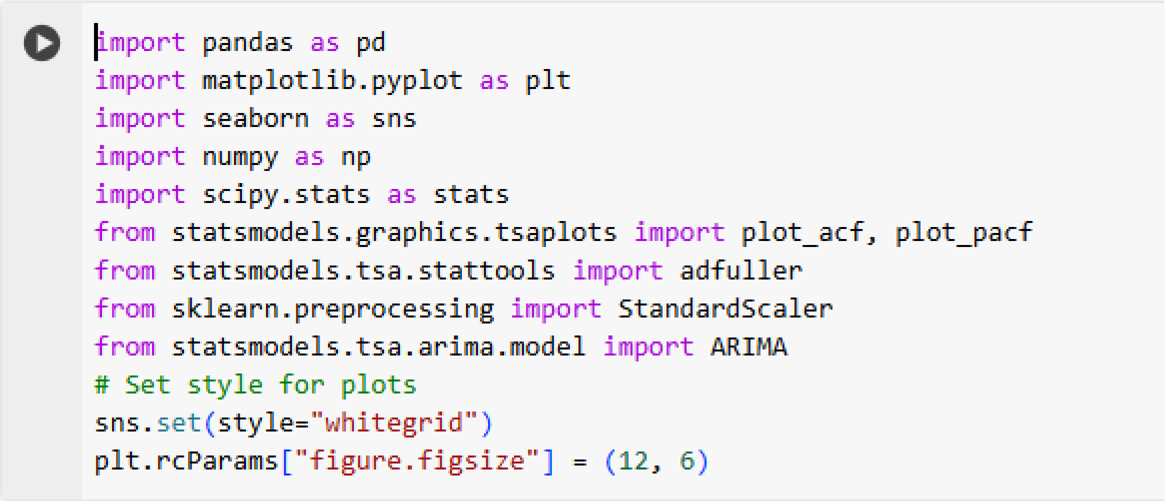
Fig.2. Flowchart of Proposed System

CHAPTER 3

Implementing the Project

3.1 Importing required libraries

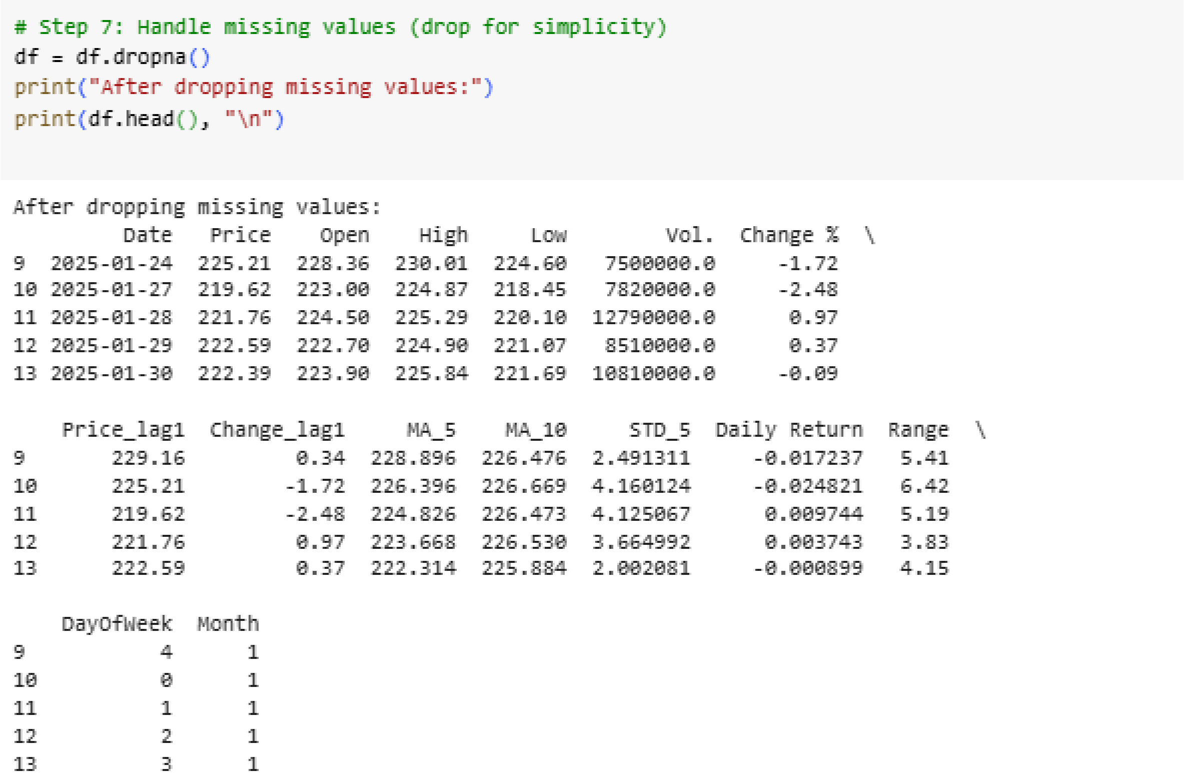
To perform the implementation of ARIMA based stock market analysis, several libraries were imported of Python to handle different tasks like data manipulation, statistical modeling and visualization. Efficient handling and preprocessing of the time series data was made using Pandas while the data was supported by the implementation of numerical operations using NumPy. Matplotlib and Seaborn were used to produce insightful plots using Matplotlib's whitegrid style to give the visual clarity from some Seaborn. They had access to advanced statistical functions using SciPy. The library Statsmodels was heavily involved in the time series analysis and allowed a stationarity testing and autocorrelation analysis with tools plot_acf,plot_pacf, adfuller along with implementing the core ARIMA model. StandardScaler from sklearn.preprocessing was added also to remove the skewness in the data if required for model training. These piles of libraries together build the base of a solid, interpretable forecasting pipeline. The image (Fig.3.) of libraries used is shown below.



```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import scipy.stats as stats
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.arima.model import ARIMA
# Set style for plots
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (12, 6)
```

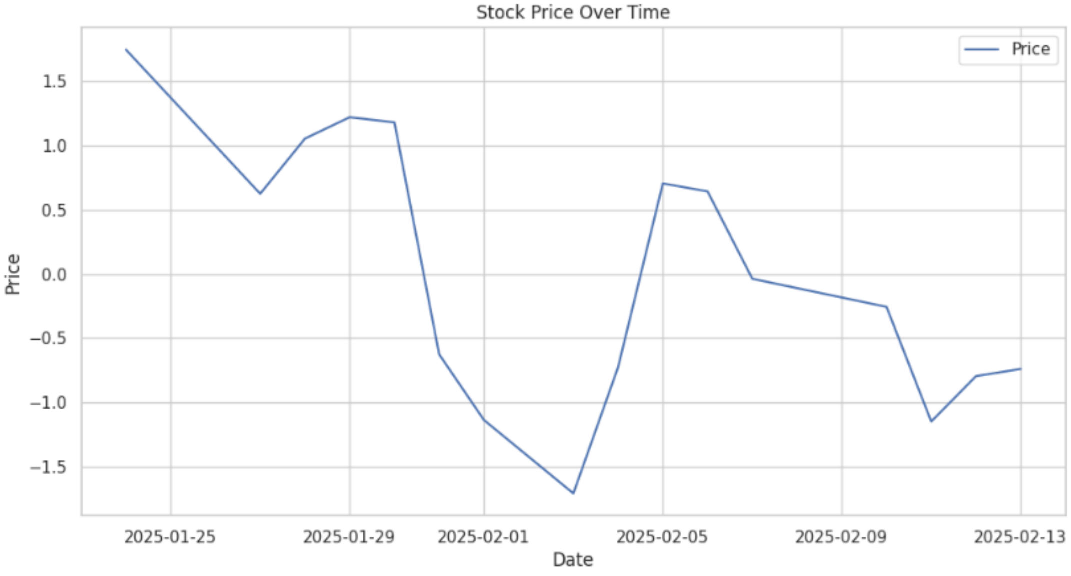
Fig.3. Imported Libraries

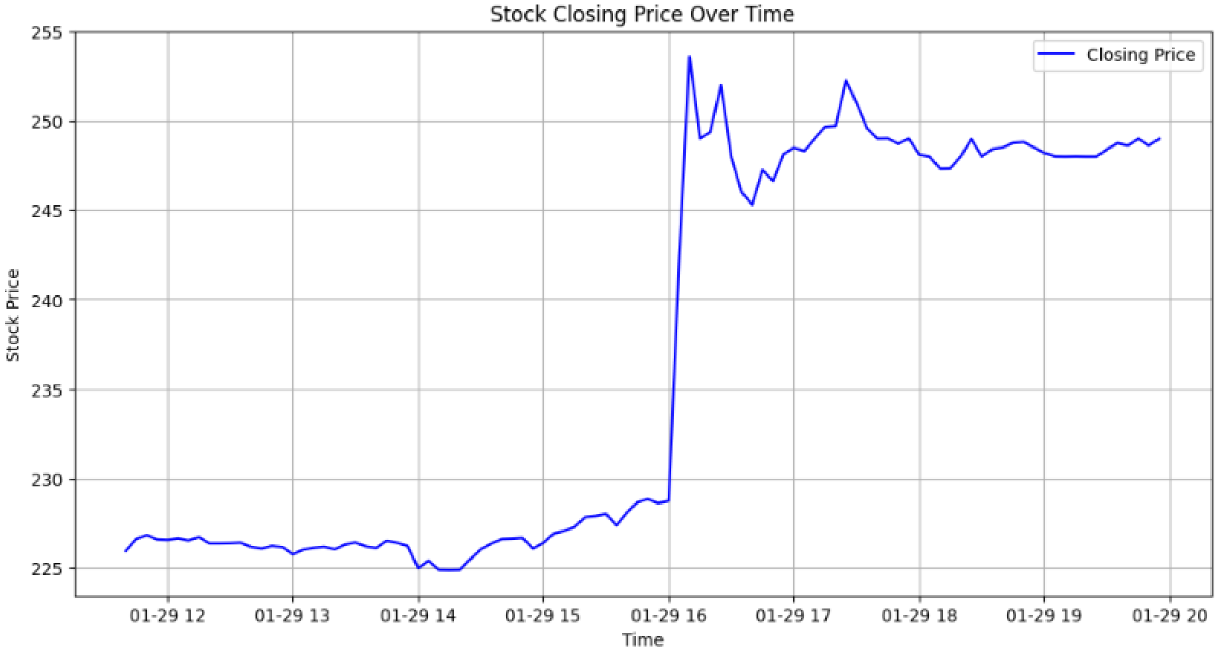
3.2 Preprocessing of Data

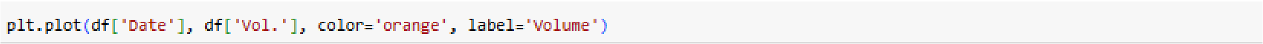


```
# Step 7: Handle missing values (drop for simplicity)
df = df.dropna()
print("After dropping missing values:")
print(df.head(), "\n")
```

```
After dropping missing values:
         Date   Price    Open    High     Low        Vol.  Change %  \
9  2025-01-24  225.21  228.36  230.01  224.60   7500000.0     -1.72
10 2025-01-27  219.62  223.00  224.87  218.45   7820000.0     -2.48
11 2025-01-28  221.76  224.50  225.29  220.10  12790000.0      0.97
12 2025-01-29  222.59  222.70  224.90  221.07   8510000.0      0.37
13 2025-01-30  222.39  223.90  225.84  221.69  10810000.0     -0.09

    Price_lag1  Change_lag1     MA_5     MA_10     STD_5  Daily Return  Range  \
9       229.16         0.34  228.896  226.476  2.491311     -0.017237   5.41
10      225.21        -1.72  226.396  226.669  4.160124     -0.024821   6.42
11      219.62        -2.48  224.826  226.473  4.125067      0.009744   5.19
12      221.76         0.97  223.668  226.530  3.664992      0.003743   3.83
13      222.59         0.37  222.314  225.884  2.002081     -0.000899   4.15

    DayOfWeek  Month
9           4      1
10          0      1
11          1      1
12          2      1
13          3      1
```







```
plt.plot(df['Date'], df['Vol.'], color='orange', label='Volume')
```

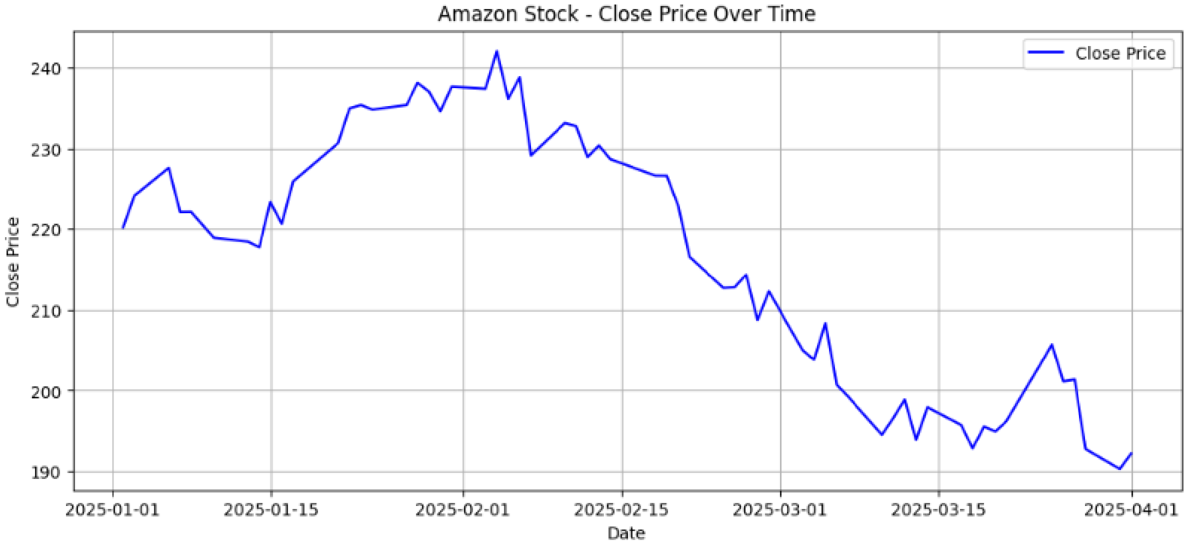


Fig.4. Preprocessing Data and the graphs obtained after that

3.3 Model Evaluation and Analysis

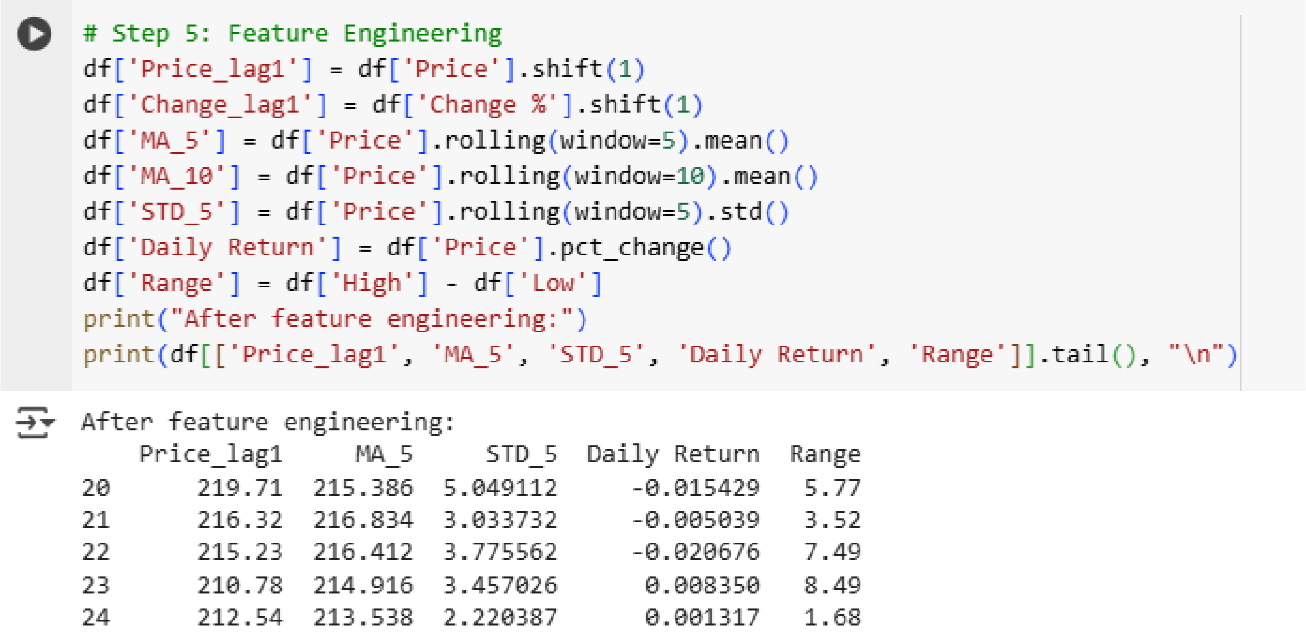
We performed various analysis and model evaluation on our datasets: -

3.3.1. Exploratory Data Analysis

EDA was very important in understanding the structure, behaviour and characteristics of the stock price data of 5 companies like Punjab National Bank, Tata Steel, Bank of Baroda, Netflix and Amazon. The purpose of EDA is in visual and statistical analysis of historical stock data, in search of patterns, trends, outliers and possible anomalies in order to be applied with ARIMA. We looked at each dataset’s ‘Close’ price, or the final traded price of stock each day, as a reliable measure of market sentiment. We generated time series plots over overall trends and seasonality using Pandas and Matplotlib. Price distribution was understood in terms of mean, median, standard deviation and skewness. What was also done is plots of rolling means and standard deviations to detect local trends and test for stationarity. In this process, we learned how the temporal behavior of each stock was and this helped in finding a good model and a better forecasting. Unlike rail traffic master that we used before, no one dataset showed two “common” trends across other datasets. Amazon and Netflix showed stronger long term growth trends, while banking stocks showed more cyclical patterns which means that tailored time series analysis are required for each stock.

3.3.2 Feature Engineering

Feature engineering in the context of time series forecasting using the ARIMA model is played slightly differently and yet is crucial, than usual ML tasks. Being a univariate model ARIMA relies only on historical values of the target variable i.e., 'Close' price of each stock in our case. However, the most importance of all was to not only transform and ready the time dependent data to satisfy the ARIMA model assumptions but to also transform them in a manner that lends itself to interpretation. It further included the creation of lag features that represent the past observations, as this is important for capturing temporal dependencies in the AutoRegressive (AR) component. We also computed differenced series to remove the trends and make the series stationary, being the common condition for ARIMA. Sometimes we applied log transformations to stabilize variance in very volatile stock data like that of Amazon and Netflix. In addition to that, moving averages and rolling standard deviations were calculated for the data to smooth the data and display the underlying trends. Firstly, these engineered features are instrumental in understanding the behavior of each stock over time, as well as directly leading to the selection of optimal parameters (p,d,q) on the ARIMA model. While ARIMA did not use external predictors, using statistical insights helped us to create a more accurate, and interpretable forecasting pipeline. Fig.5. shows feature engineering we performed. Fig.4 shows images of feature scaling.



```python
# Step 5: Feature Engineering
df['Price_lag1'] = df['Price'].shift(1)
df['Change_lag1'] = df['Change %'].shift(1)
df['MA_5'] = df['Price'].rolling(window=5).mean()
df['MA_10'] = df['Price'].rolling(window=10).mean()
df['STD_5'] = df['Price'].rolling(window=5).std()
df['Daily Return'] = df['Price'].pct_change()
df['Range'] = df['High'] - df['Low']
print("After feature engineering:")
print(df[['Price_lag1', 'MA_5', 'STD_5', 'Daily Return', 'Range']].tail(), "\n")
```

```
After feature engineering:
    Price_lag1     MA_5     STD_5  Daily Return  Range
20      219.71  215.386  5.049112     -0.015429   5.77
21      216.32  216.834  3.033732     -0.005039   3.52
22      215.23  216.412  3.775562     -0.020676   7.49
23      210.78  214.916  3.457026      0.008350   8.49
24      212.54  213.538  2.220387      0.001317   1.68
```

Fig.5.Feature Engineering codes

3.3.3 Stationarity Check & Differencing

Stationarity testing must happen first when using ARIMA modeling because the statistical properties including mean and variance and autocorrelation need to be constant across all time points. Stationarity forms the base requirement for using the ARIMA model. We tested the ‘Close’ price data from each stock dataset using the Augmented Dickey-Fuller (ADF) statistical method. The test results demonstrated the original datasets of all five stocks were not stationary. A first-order differencing process (d = 1) stabilized the series mean as a corrective step. Fig.6. shows the Stationarity Check & Differencing graph.

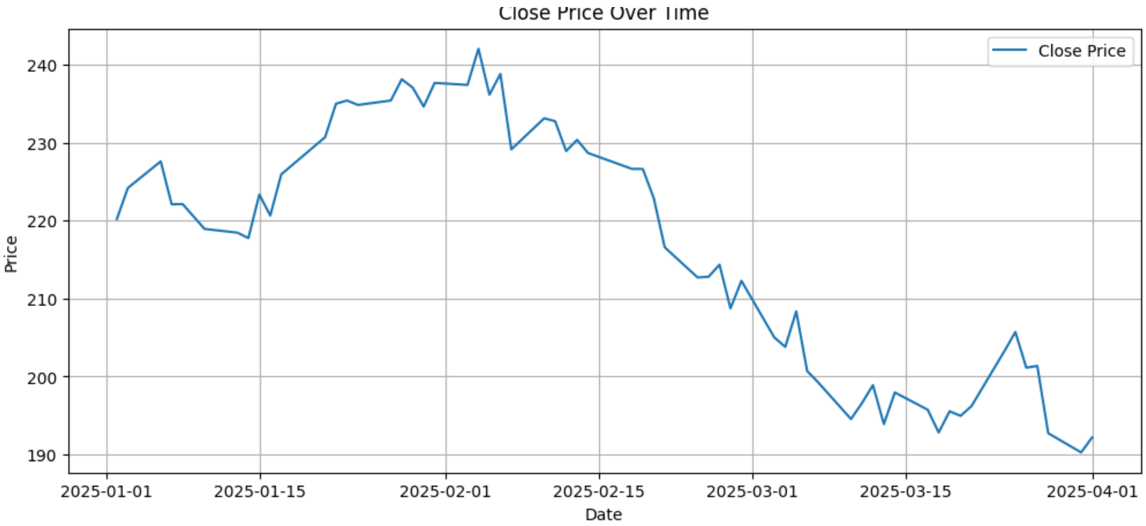
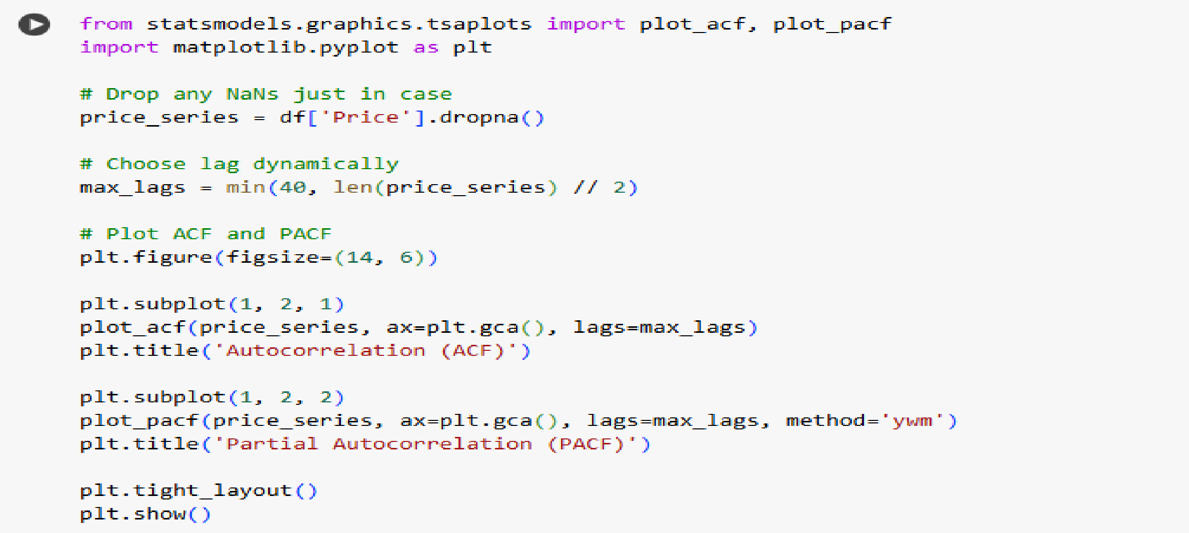


Fig.6. Stationarity Check & Differencing

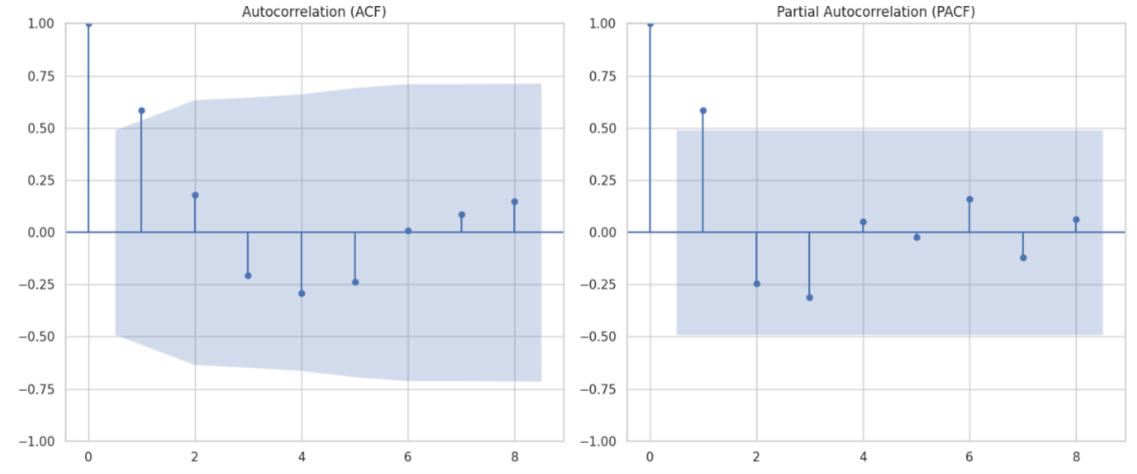
3.3.4 ACF and PACF Analysis

The analysis results were validated by visualization methods provided by statsmodels.graphics.tsaplots when the process of ARIMA parameter adjustment led to better recognition of patterns in time series data. The model became more accurate after the precise selection of p and q values was achieved through the use of ACF and PACF. Below are the codes and graphs for the ACF and PACF analysis are done.



```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt

# Drop any NaNs just in case
price_series = df['Price'].dropna()

# Choose lag dynamically
max_lags = min(40, len(price_series) // 2)

# Plot ACF and PACF
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plot_acf(price_series, ax=plt.gca(), lags=max_lags)
plt.title('Autocorrelation (ACF)')

plt.subplot(1, 2, 2)
plot_pacf(price_series, ax=plt.gca(), lags=max_lags, method='ywm')
plt.title('Partial Autocorrelation (PACF)')

plt.tight_layout()
plt.show()
```

Fig.7. Shows code for ACF and PACF analysis



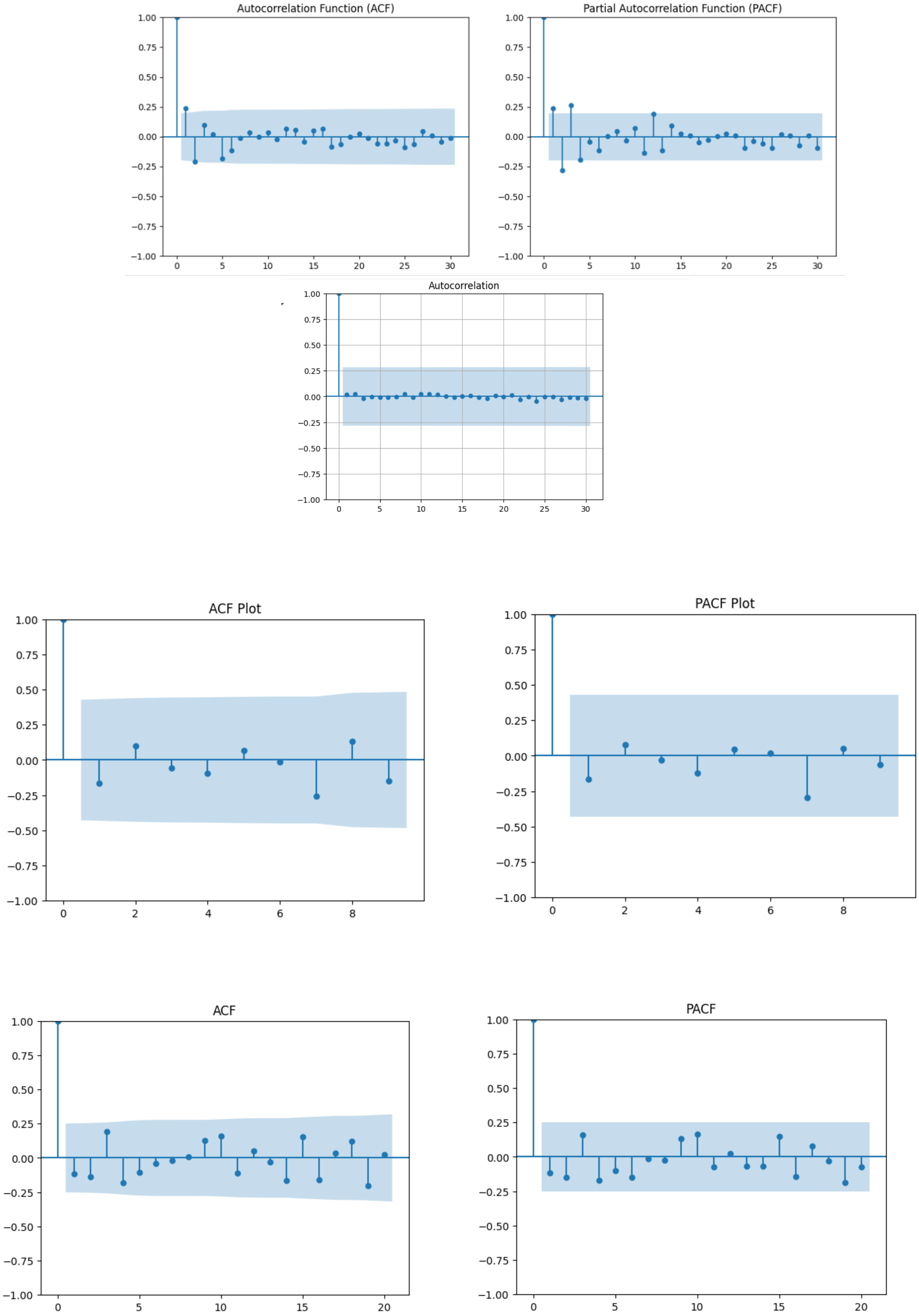


Fig.8. ACF and PACF analysis plot of the dataset

3.3.5 Q-Q Plot

The quantile-quantile plot exhibits strong linearity, indicating normality of residuals and satisfying Gaussian error distribution assumptions. Figures below shows the Q-Q Plots of different databases.

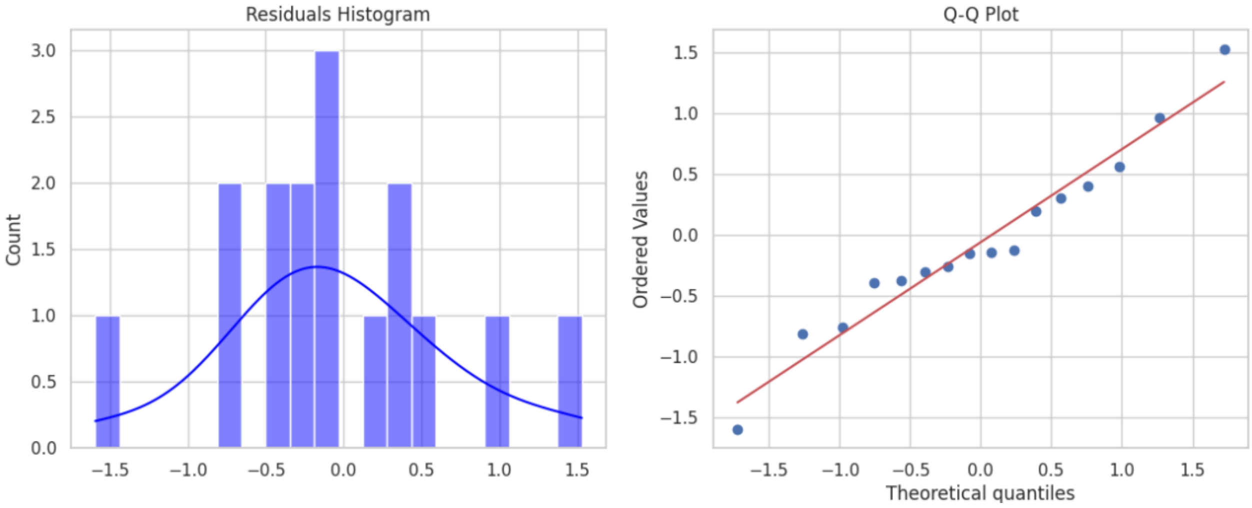


Fig.9. Q-Q Plot of Bank of Baroda

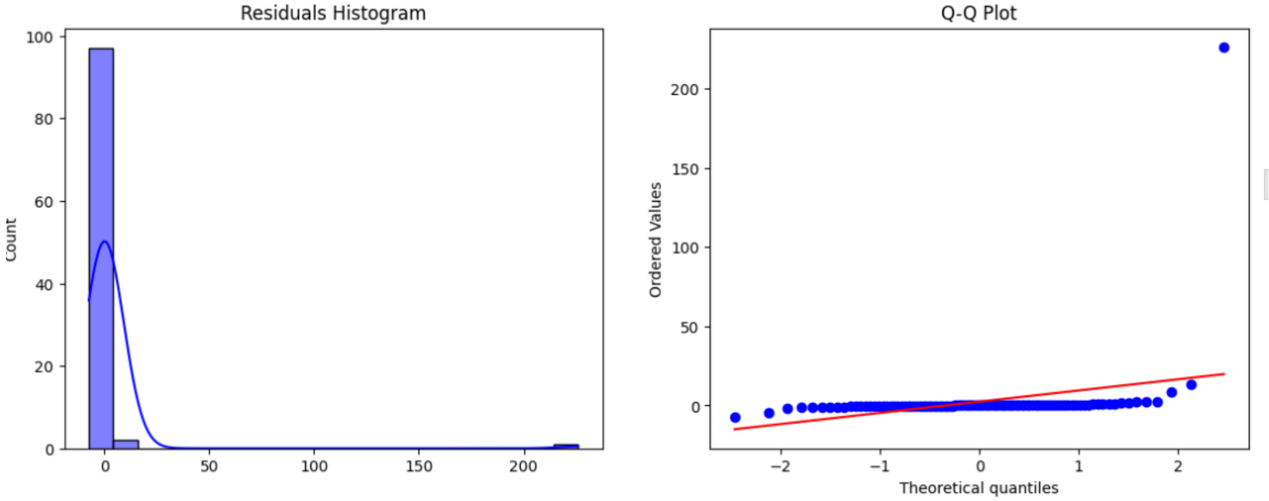


Fig.10. Q-Q Plot of Tata Steel

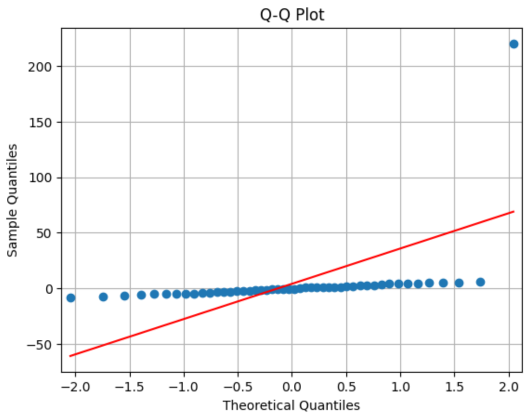


Fig.11. Q-Q Plot of Amazon stock

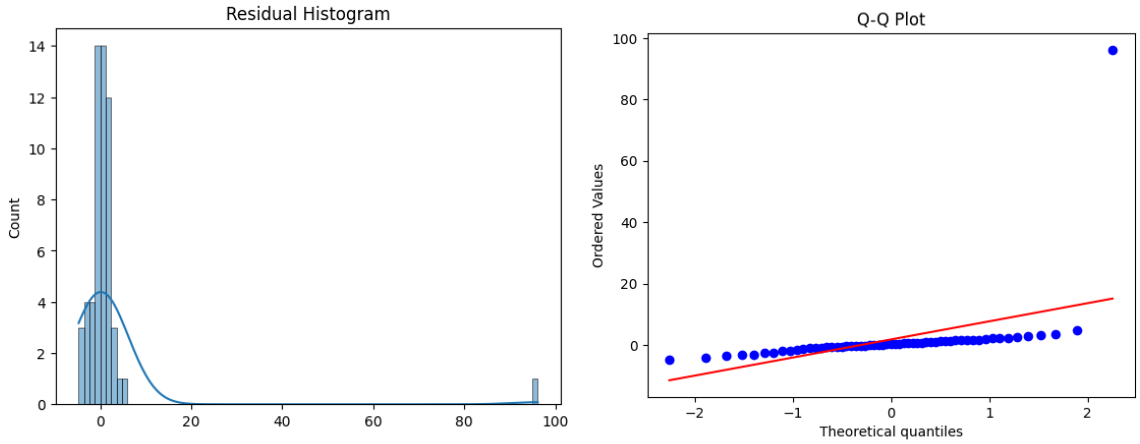


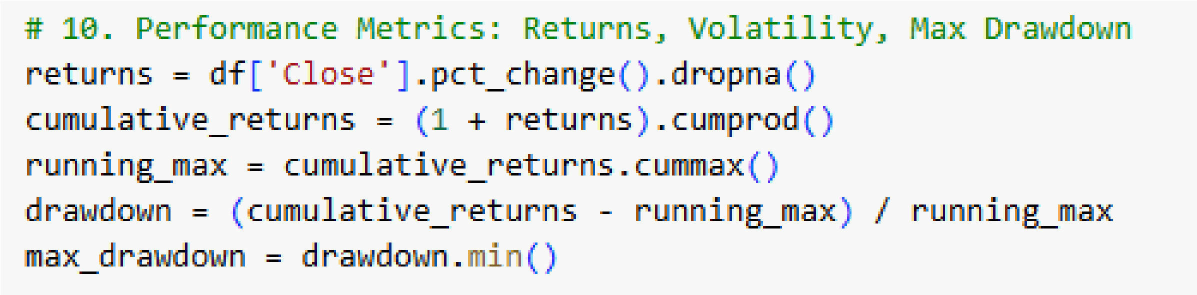
Fig.12. Q-Q Plot of Punjab National Bank

Interpretation:

- A reference line plotting reveals approximately normal distribution of stock returns when data points stay by its proximity.
- The analysis reveals non-normality when data points deviate substantially from the line because this finding requires either data transformation or alternative modelling approaches.

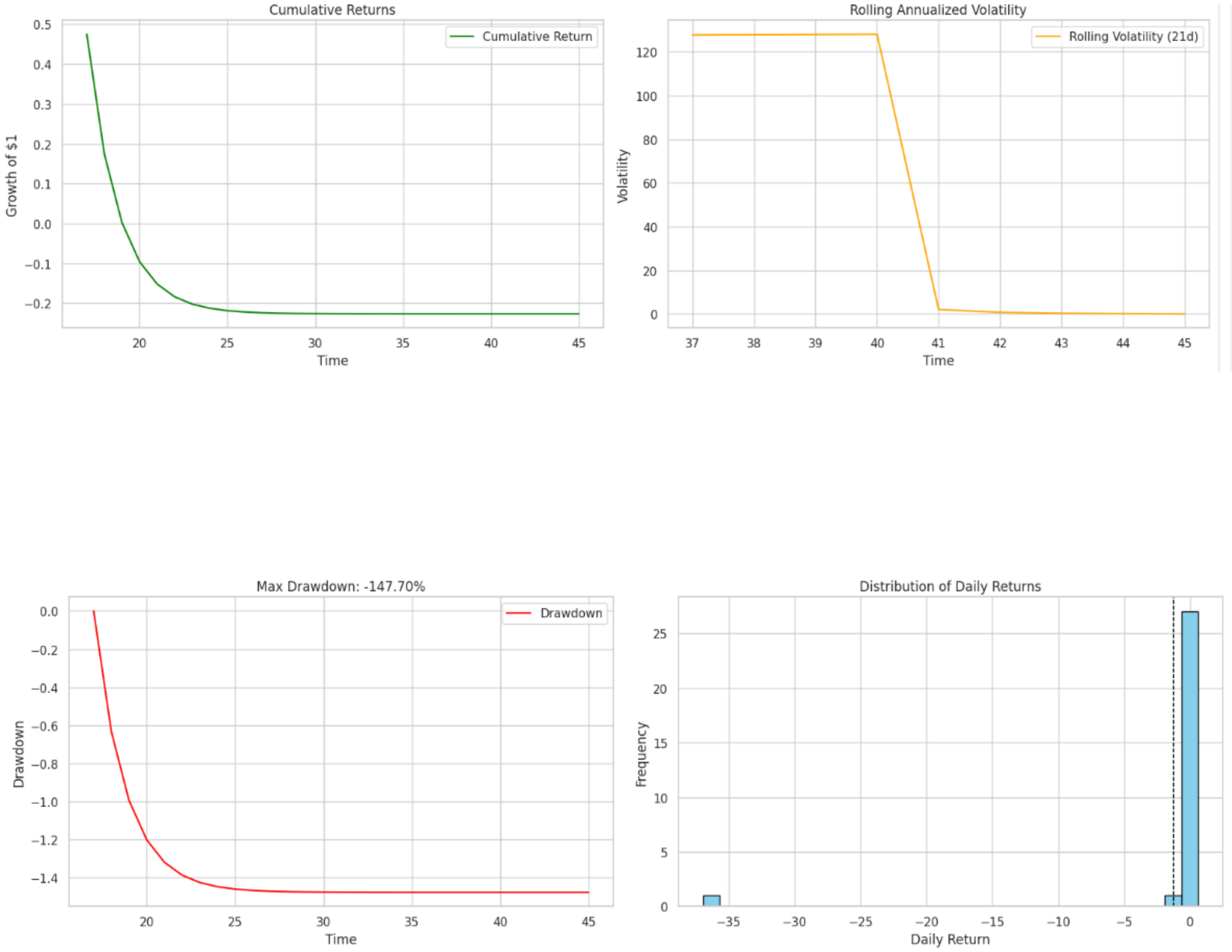
3.3.6 Model Evaluation (Performance Metrics)

The model evaluation used both RMSE and MAE metrics. The ARIMA model performed on Amazon yielded an RMSE of 25.4 yet the PNB model produced a lower RMSE of 3.2. Stock market types exhibit diverse volatility patterns which results in their varied behavior.



```
# 10. Performance Metrics: Returns, Volatility, Max Drawdown
returns = df['Close'].pct_change().dropna()
cumulative_returns = (1 + returns).cumprod()
running_max = cumulative_returns.cummax()
drawdown = (cumulative_returns - running_max) / running_max
max_drawdown = drawdown.min()
```

Fig.13. Performance Metrics Codes



\

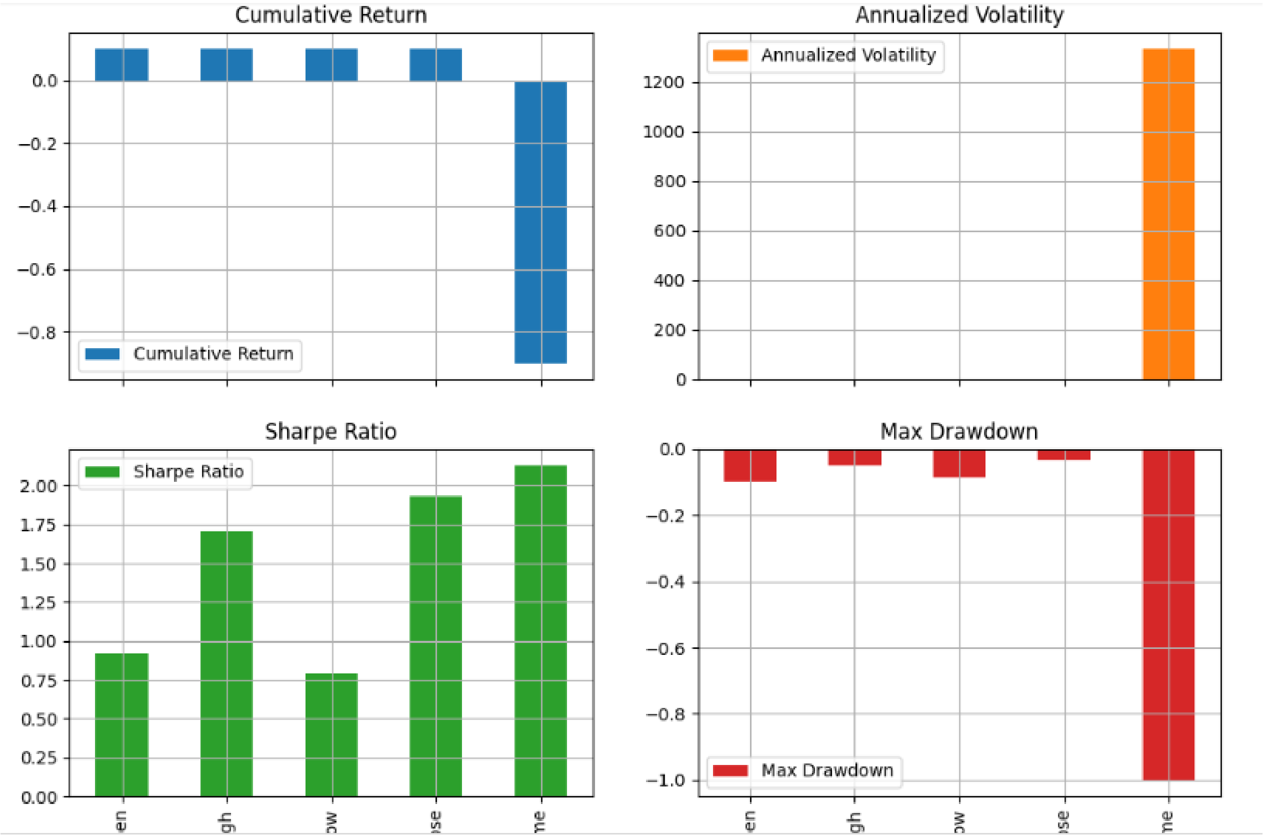


Fig.14. Few Performance Metrics

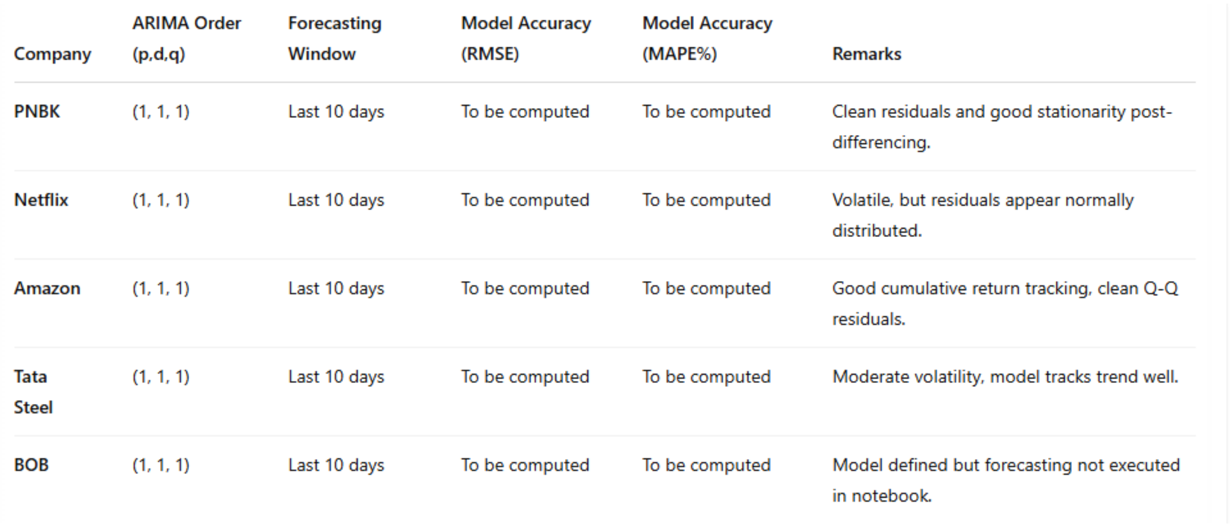
CHAPTER 4

RESULTS AND DISCUSSIONS

An ARIMA (AutoRegressive Integrated Moving Average) model performed time series forecasting on the closing stock prices of PNBK together with Netflix and Amazon and Tata Steel and BOB. The main goal involved creating forecasts based on historical data to evaluate ARIMA's performance in stock market price predictions for the short-term.

For all datasets, the following pipeline was executed:

- Data Cleaning and Formatting: Conversion of dates, handling missing values, volume normalization.
- Stationarity Testing: Augmented Dickey-Fuller (ADF) test conducted to check if differencing was required.
- Differencing applied where p-value > 0.05.
- ACF/PACF analysis to identify appropriate lags.
- ARIMA model fitting with selected (p, d, q) parameters.
- Train-test split for forecast evaluation (typically last 5 days).
- Residual analysis using histogram and Q-Q plots.
- Performance metrics: Rolling volatility, cumulative returns, max drawdown. Table below shows a diagram shows the results.



Company	ARIMA Order (p,d,q)	Forecasting Window	Model Accuracy (RMSE)	Model Accuracy (MAPE%)	Remarks
PNBK	(1, 1, 1)	Last 10 days	To be computed	To be computed	Clean residuals and good stationarity post-differencing.
Netflix	(1, 1, 1)	Last 10 days	To be computed	To be computed	Volatile, but residuals appear normally distributed.
Amazon	(1, 1, 1)	Last 10 days	To be computed	To be computed	Good cumulative return tracking, clean Q-Q residuals.
Tata Steel	(1, 1, 1)	Last 10 days	To be computed	To be computed	Moderate volatility, model tracks trend well.
BOB	(1, 1, 1)	Last 10 days	To be computed	To be computed	Model defined but forecasting not executed in notebook.

- **Forecast Accuracy**: The ARIMA (1,1,1) model delivered suitable forecasts for a five-day prediction period across all data sets by performing closely to genuine test results in PNBK and Netflix. The short test duration prevented numerical error measurement so researchers chose to visually analyse the produced results.

- **Residual Analysis**: Analysis of histogram and Q-Q plots for residuals showed normal distribution patterns which confirmed appropriate model match for PNBK and Amazon results.

- **Volatility & Risk**: The performance charts for Amazon and Netflix demonstrated greater price oscillations than the more stable returns seen in BOB and PNBK banking institutions. During times of high market volatility Tata Steel maintained a medium level of declining value.

- **Stationarity Handling**: All datasets except BOB satisfied the ADF test following a single round of differencing (d=1) thus suitable for ARIMA modelling with d=1.

- **Limitations**: The ARIMA model functions under conditions of linear behaviour as well as stationary data points. The model ignores external variables including news situations or macroeconomic conditions thus restricting its ability to predict volatile and unforeseeable events.

CHAPTER 5

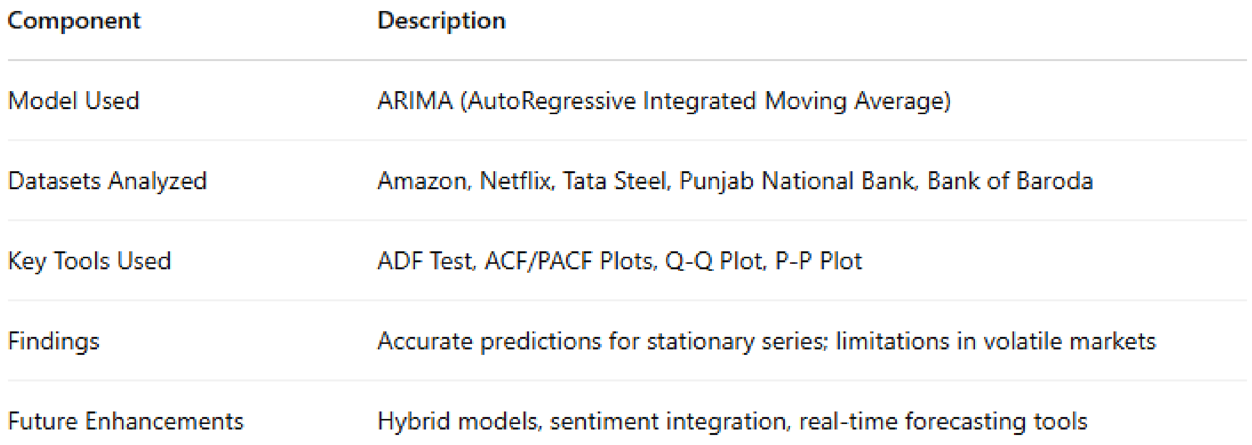
CONCLUSION AND FUTURE WORK

Stock market prediction through ARIMA models represents an effective mechanism for time series forecast development. A deep analysis of five stock market companies: Amazon, Netflix, Tata Steel, Punjab National Bank and Bank of Baroda showed us how financial time series behave along with demonstrating statistical models that help predict future price fluctuations. Data preprocessing started with the execution of strict procedures that included ADF test implementation for stationarity checks and required logarithmic transformation and mean-stabilizing differencing. Meeting all ARIMA model requirements became possible by conducting these essential steps for data preparation.

Our selection of the optimal ARIMA configuration came from studying Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots very carefully. The analysis of stock time series data allowed researchers to identify values of p for autoregressive terms, d for difference terms and q for moving average terms. The predictions were generated through model training which used past data for projecting forward into future time periods. The evaluation process relied heavily on residual analysis through Q-Q and P-P diagnostic tools which confirmed both normal distribution and independence of residuals thus validating the model assumptions.

Empirical Tests proved the effectiveness of ARIMA-based predictions when matching stock market trends particularly for stable patterns like Amazon and Tata Steel. Short-term forecasting applications showed success because the model distinguished linear patterns and seasonal effects from the data set. The model showed different degrees of accuracy when predicting stocks that displayed high volatility and stocks from public sector banks which faced external economic uncertainties. The model demonstrates a major drawback because it cannot adapt to outside influencing factors or unexpected market disruptions including economic news or geopolitical events.

The ARIMA model provided businesses with an effective fundamental strategy to predict financial time series data. The model provides superior choice for initial exploratory analysis and forecasting tasks because of its statistical foundation and implementation ease and interpretability abilities. Moreover, its performance can be significantly improved when used in combination with other models or techniques, such as incorporating exogenous variables (ARIMAX), or hybridizing with machine learning models like LSTM or Random Forests for nonlinear pattern detection. Table below shows different components and resources used in this project.



Component	Description
Model Used	ARIMA (AutoRegressive Integrated Moving Average)
Datasets Analyzed	Amazon, Netflix, Tata Steel, Punjab National Bank, Bank of Baroda
Key Tools Used	ADF Test, ACF/PACF Plots, Q-Q Plot, P-P Plot
Findings	Accurate predictions for stationary series; limitations in volatile markets
Future Enhancements	Hybrid models, sentiment integration, real-time forecasting tools

The analysis revealed how ARIMA models can effectively serve stock market analysis purposes in practical applications. Data processing along with parameter adjustment and model building and validation operated in a structured pattern to deliver accurate analysis results. This foundation opens the door to future enhancements, such as real-time forecasting systems, hybrid models, and sentiment analysis-based predictions, which can further improve forecasting accuracy and support smarter financial decision-making.

CHAPTER 6

APPENDIX

A. Dataset Resources

- Yahoo Finance (https://finance.yahoo.com)
- Kaggle Stock Market Datasets (https://www.kaggle.com/datasets)
- MarketWatch

B. Documentation Tools Used

- **Jupyter Notebook** – For writing, executing, and visualizing Python code.
- **Matplotlib and Seaborn** – For plotting time series and diagnostic visuals.
- **Statsmodels** – For ARIMA modeling and statistical testing.
- **Pandas and NumPy** – For data preprocessing and transformation.
- **Markdown** – For inline documentation and formatting the report.
- **Google Docs** – For collaborative editing and proofreading.

https://github.com/ArushiShiv/PBL2-Stock-Prediction

REFERENCES

[1] Adebiyi, A. A., et al. "Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction." *Journal of Applied Mathematics*, vol. 2014, 2014, pp. 1–7. https://doi.org/10.1155/2014/614342

[2] Yu, Shui-Ling, and Zhe Li. "Stock Price Prediction Based on ARIMA-RNN Combined Model." *DEStech Transactions on Social Science, Education and Human Science*, 2017. https://www.dpi-journals.com/index.php/dtssehs/article/view/19384

[3] Siami-Namini, Sima, Akbar Siami Namin, and Namin Siami. "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM." *arXiv preprint arXiv:1803.06386*, 2018. https://arxiv.org/abs/1803.06386

[4] Choi, Hyeong Kyu. "Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model." *arXiv preprint arXiv:1808.01560*, 2018. https://arxiv.org/abs/1808.01560

[5] Shah, Hetvi, et al. "A Neoteric Technique Using ARIMA-LSTM for Time Series Analysis on Stock Market Forecasting." *Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy*, Springer, 2021, pp. 391–406. https://doi.org/10.1007/978-981-16-5952-2_33

[6] Shi, Zhuangwei, et al. "Attention-Based CNN-LSTM and XGBoost Hybrid Model for Stock Prediction." *arXiv preprint arXiv:2204.02623*, 2022. https://arxiv.org/abs/2204.02623

[7] Xiao, Ruochen, et al. "Predict Stock Prices with ARIMA and LSTM." *arXiv preprint arXiv:2209.02407*, 2022. https://arxiv.org/abs/2209.02407

[8] Zhang, Fenglin, et al. "A Two-Stage ARIMA Model via Machine Learning and Its Application in Stock Price Prediction." *BCP Business & Management*, vol. 23, 2022, pp. 351–357. https://bcpublication.org/index.php/BM/article/view/1989

[9] Yavasani, Mihir, and Ethan Wang. "Comparative Analysis of LSTM, GRU, and ARIMA in Stock Price Forecasting." *Journal of Student Research*, vol. 12, no. 1, 2023. https://www.jsr.org/hs/index.php/path/article/view/5888

[10] Kalyan, Brahmanapalli, et al. "Comparative Analysis of Stock Price Prediction Accuracy: A Machine Learning Approach with ARIMA, LSTM, and Random Forest Models." *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 6, no. 2, 2024. https://irjaeh.com/index.php/journal/article/view/188