Data Visualization and Inference Modeling-The Case of Nifty

Project report submitted for
Industrial Project Based Learning in Data Science and Machine Learning

Under the Supervision of

P.Mohan
Director, Gyan Astra IT Solutions

&
Y.V.N Phani Kishore
Director, Gyan Astra IT Solutions

By

Team Members Name	(Roll Number)
P Pranavi	21R11A6742
C Manognasri	21R11A6729
A. Sai Srujana	21R11A05A8
P Moulika	21R11A05E0

Department of Computer Science & Engineering

Nov 2023 - May 2024

Geethanjali College of Engineering & Technology **Accredited by NBA (UGC Autonomous)**

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi, NAAC - A+)

Geethanjali College of Engineering & Technology

(Affiliated to J.N.T.U.H, Approved by AICTE, NEWDELHI.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Mr./Ms. C.Manognasri, A.Sai Srujana, P Moulika , P Pranavi bearing roll no 21R11A6729,21R11A05A8,21R11A05E0, 21R11A6742 has successfully completed Industrial Project Based Learning in Data Science and Machine Learning held during November 2023 to May 2024 at Geethanjali college of Engineering and Technology.

Coordinator	H.O.D - C.S.E
Mr. E. Mahender	**Dr. A. Sree Lakshmi**
Assistant Professor	**Professor & Head**
Dept of C.S.E	**Dept of C.S.E**

ACKNOWLEDGEMENT

We are greatly indebted to the authorities of Geethanjali College of Engineering and Technology, Cheeryal, R. R Dist, for proving us the necessary facilities to successfully carry out this Industrial Project Based Learning titled “Contextual Text Extraction From PDF Files Using MachineLearning Practices**”** under Value added course.

Firstly, we would like to express our sincere gratitude to our **Principal Dr. S.Udaya Kumar** for providing the necessary infrastructure to complete our Value added course.

Secondly, we thank and express our solicit gratitude to **Dr. A. Sreelakshmi**, Prof & HOD, CSEdepartment Geethanjali College of Engineering, and technology, for her invaluable help andsupport which helped us a lot in successfully completing of our course.

We express our gratitude to **P.Mohan**, Director, Gyan Astra IT Solutions, **Y.V.N Phani Kishore** Director, Gyan Astra IT Solutions, **Dr Kamakshiah Kolli**, Associate Professor, Dept of CSE, GCET, **Mr. E.Mahendra**, Dept of CSE, GCET and **N.Madhavi**, **Assistant Professor**, Dept of CSE-Data Science, GCET for their valuable suggestions and encouragement which helped us in the successful completion of our course.

Finally, we would like to express our heartfelt thanks to our parents who were very supportive both financially and mentally and for their encouragement to achieve our set goals.

Team Names and roll Numbers.

P Pranavi	21R11A6742
C Manognasri	21R11A6729
A. Sai Srujana	21R11A05A8
P Moulika	21R11A05E0

TABLE OF CONTENTS

ABSTRACT

This project undertakes a meticulous exploration of stock market data, delving into Nifty monthly and yearly returns over a comprehensive 20-year period. With a particular focus on the unprecedented events of 2020, notably the global COVID-19 pandemic, the project seeks to unravel intricate patterns and trends within the stock market. Leveraging a suite of powerful Python libraries, including Seaborn, Pandas, yfinance, and Matplotlib, the project commences with the essential task of importing and structuring the dataset. Following this initial step, the project utilizes sophisticated visualization techniques such as heatmaps to offer a comprehensive overview of monthly returns, providing insights into the inherent volatility and fluctuations in the market. Additionally, histograms are crafted to dissect the distribution of returns across different standard deviation buckets, offering a nuanced understanding of the market's risk profile over time. Through exhaustive exploratory data analysis, the project aims to extract actionable insights and meaningful observations, shedding light on the underlying dynamics of the stock market.

Expanding its scope, the project integrates the SARIMA forecasting model into a user-friendly web application using Flask. SARIMA's capability to capture seasonal patterns makes it ideal for stock market predictions. The application empowers users to interactively forecast future market values, providing a seamless interface for data exploration and decision-making. By combining analytical techniques with user-centric design, the project aims to democratize predictive insights, assisting investors in confidently navigating the complexities of the stock market.

INTRODUCTION

The NIFTY 50 serves as a crucial benchmark in the Indian stock market, reflecting the combined performance of 50 prominent companies listed on the National Stock Exchange (NSE). Managed by NSE Indices, a subsidiary of NSE Strategic Investment Corporation Limited, the index was introduced in April 1996 and has since emerged as a cornerstone of India's financial realm. Providing investors with exposure to diverse sectors of the Indian economy, it facilitates market access through a unified portfolio.

Across time, the NIFTY 50 has matured into India's paramount financial product, cultivating a resilient ecosystem encompassing exchange-traded funds, futures, and options, both domestically and globally. Despite its significance, the index witnessed a decline in market share from 65% to 29% between 2008 and 2012, partly attributable to the emergence of sector-specific indices like NIFTY Bank, NIFTY IT, NIFTY Pharma, and NIFTY Next 50.

With 13 sectors in its composition, the NIFTY 50 provides a diversified snapshot of the Indian economy. As of March 2024, the index prominently assigns substantial weightage to financial services (33.53%), information technology (13.04%), oil and gas (12.87%), consumer goods (8.15%), and automotive (7.57%). This extensive representation across various sectors underscores the index's significance as a comprehensive indicator of the Indian stock market, assisting market participants in making informed investment decisions and diversifying their portfolios.

The NIFTY index holds significant importance in the Indian stock market, serving as a key metric for investors, traders, analysts, and fund managers. Comprising the top 50 companies listed on the National Stock Exchange (NSE) based on market capitalization, the index undergoes periodic reviews and rebalancing to reflect evolving market conditions.

A primary function of the NIFTY index is as a benchmark for gauging the overall performance of the Indian stock market. Tracking its movements provides stakeholders with insights into market health and direction, allowing for comparisons of individual stock performance or portfolio returns against broader market trends.

Moreover, NIFTY serves as an investment tool, enabling investors to access the Indian stock market through index funds or exchange-traded funds (ETFs) that mirror its performance. This offers a convenient means to diversify portfolios across various companies and sectors.

Additionally, NIFTY acts as a barometer of India's economic health, reflecting the collective performance of its top 50 companies. Changes in the index thus serve as indicators of the country's economic trajectory and growth prospects over time.

In essence, NIFTY plays a pivotal role in the Indian stock market ecosystem, providing vital insights to market participants. By monitoring index movements, investors can glean valuable information on market trends, risks, and opportunities, empowering them to make well-informed investment decisions.

2.PROBLEM STATEMENT

This project delves into analyzing stock market data, with a specific focus on the Nifty index. Spanning over 20 years, the dataset provides detailed monthly returns of Nifty, offering ample information for analysis. By scrutinizing this data, our goal is to pinpoint trends and patterns that can inform investment decisions and improve portfolio management strategies. Through thorough analysis and interpretation, we aim to offer valuable insights to empower investors in navigating the complexities of the stock market.

The year 2020 stands out for its profound significance, characterized by the unparalleled challenges brought about by the COVID-19 pandemic. Against this backdrop, our focus lies in examining how the stock market, particularly the Nifty index, reacted to the upheaval and volatility triggered by the pandemic. Through a meticulous analysis of data from this pivotal year, our objective is to unearth valuable insights into market dynamics and investor behavior amidst uncertainty. Ultimately, our aim is to empower investors with the insights and understanding necessary to navigate turbulent market conditions adeptly and make informed investment decisions that are in line with their financial objectives.

3.Objectives

- Begin by importing the dataset into Python. This initial step is crucial for laying the groundwork and acc essing the Nifty data for comprehensive examination.

- Construct a heat map to visually represent Nifty's monthly returns across the specified timeframe. This v isualization provides valuable insights into the temporal fluctuations and trends within Nifty's performan ce, aiding in understanding its behavior over different time periods.

- Categorize Nifty's monthly returns into distinct buckets based on standard deviations and generate a hist ogram to showcase the distribution of returns. This segmentation strategy offers a nuanced understandin g of return variability and highlights the prevalence of outlier events within Nifty's performance.

- Similar to the monthly analysis, create a histogram illustrating the distribution of yearly returns by categ orizing them into standard deviation-based buckets. This macroscopic view of Nifty's performance on an annual basis helps identify overarching trends and anomalies.

- Extend the project beyond mere analysis by developing a WEBUI or application using the Flask micro w eb framework. This interactive platform empowers stakeholders to engage directly with the analyzed dat a, fostering a deeper understanding of Nifty's performance and facilitating informed investment decision s.

4. METHODOLOGY

- **4.1 Data Source**

  - Brief description of the data source :

Data Overview:

The dataset offers a comprehensive portrayal of stock price data spanning from 2000 to 2023. Each row corresp onds to a specific year, with columns representing individual months from January to December. Additionally, t here is a column labeled "Annual," signifying the total stock price for the entire year.

Feature Description:

- Year: This column indicates the years covered in the dataset, ranging from 2000 to 2023.

- Jan-Dec: These columns denote the stock prices for each respective month of the year. For instance, the "Jan" column contains stock prices for January, "Feb" for February, and so forth.

- Annual: The "Annual" column presents the total stock price for the entire year, consolidating the monthly data . This column provides a holistic perspective on the yearly performance of the stock.

4.2 Exploratory Data Analysis

- Exploratory Data Analysis (EDA) plays a pivotal role as the initial phase in the machine learning workflow, aiming to thoroughly examine and understand the properties of the data before delving into modeling tasks. Through EDA, analysts aim to uncover fundamental data characteristics such as distribution, inter-variable correlations, and noticeable patterns or anomalies. This preliminary exploration is essential as it provides deep insights into the dataset, enabling informed decisions regarding feature manipulation, data preprocessing, and model selection.

- By conducting EDA, researchers can identify and address any missing or erroneous data, outliers, or inconsistencies, thereby establishing a more robust foundation for subsequent machine learning endeavors.

Information about the Features & their data types:

Column No.1: Year, Data Type: int64
Column No.2: Jan, Data Type: float64
Column No.3: Feb, Data Type: float64
Column No.4: Mar, Data Type: float64
Column No.5: Apr, Data Type: float64
Column No.6: May, Data Type: float64
Column No.7: Jun, Data Type: float64
Column No.8: Jul, Data Type: float64
Column No.9: Aug, Data Type: float64
Column No.10: Sep, Data Type: float64
Column No.11: Oct, Data Type: float64
Column No.12: Nov, Data Type: float64
Column No.13: Dec, Data Type: float64
Column No.14: Annual, Data Type: float64

The dataset consists of 14 columns. Column No.1 represents the year and is of integer data type. Columns No.2 to No.13 represent the months from January to December, each of float64 data type. Column No.14 represents the annual data and is also of float64 data type.

4.2.2 Checking for Data Consistency

Dataset statistics

Number of variables	14
Number of observations	24
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	2.8 KiB
Average record size in memory	117.5 B

Variable types

Numeric	14

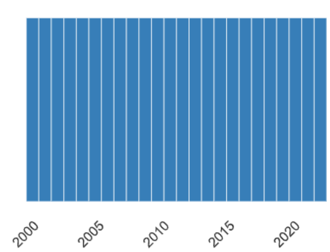
The dataset consists of 14 numeric variables and contains 24 observations. There are no missing cells or duplicate rows present in the dataset. In terms of memory usage, the dataset occupies 2.8 KiB, with an average record size of 117.5 bytes.

Year

Year
Real number (ℝ)

UNIFORM UNIQUE

Distinct	24	Minimum	2000
Distinct (%)	100.0%	Maximum	2023
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	2011.5	Memory size	324.0 B



The "Year" variable comprises real numbers (ℝ) ranging uniformly from 2000 to 2023, encompassing 24 distinct values across the entire range. There are no missing, infinite, zero, or negative values present. The mean value is 2011.5, and the memory size occupied by this variable is 324.0 bytes.

Outliers:

```
Outliers indices for column 'Jan': []
Outliers for column 'Jan': []
Outliers indices for column 'Feb': []
Outliers for column 'Feb': []
Outliers indices for column 'Mar': []
Outliers for column 'Mar': []
Outliers indices for column 'Apr': []
Outliers for column 'Apr': []
Outliers indices for column 'May': [9]
Outliers for column 'May': [28.07]
Outliers indices for column 'Jun': [8]
Outliers for column 'Jun': [-17.03]
Outliers indices for column 'Jul': []
Outliers for column 'Jul': []
Outliers indices for column 'Aug': []
Outliers for column 'Aug': []
Outliers indices for column 'Sep': []
Outliers for column 'Sep': []
Outliers indices for column 'Oct': [8]
Outliers for column 'Oct': [-26.41]
Outliers indices for column 'Nov': []
Outliers for column 'Nov': []
Outliers indices for column 'Dec': []
Outliers for column 'Dec': []
Outliers indices for column 'Annual': []
Outliers for column 'Annual': []
```

Observations:

The provided figure illustrates the indices and corresponding values of outliers detected for each column in the dataset.

For the 'Jan', 'Feb', 'Mar', 'Apr', 'Jul', 'Aug', 'Sep', 'Nov', 'Dec', and 'Annual' columns:
No outliers were detected.

For the 'May' column:
There is one outlier detected at index 9 with the value 28.07.

For the 'Jun' column:
There is one outlier detected at index 8 with the value -17.03.

For the 'Oct' column:
There is one outlier detected at index 8 with the value -26.41.

These observations indicate that the majority of the columns have no outliers based on the specified threshold. However, a few columns ('May', 'Jun', and 'Oct') contain outliers at specific indices. These outliers may represent data points that significantly deviate from the mean and could potentially be errors or indicative of unusual phenomena.

Normalization:

```
    Year    Jan   Feb    Mar    Apr       May       Jun   Jul    Aug    Sep  \
0   2000   4.44  7.02  -7.64  -7.98  0.341764  0.795822 -9.42   4.60  -8.78
1   2001   8.56 -1.48 -15.04  -2.00  0.466022  0.400606 -3.16  -1.78 -13.28
2   2002   1.54  6.20  -1.09  -3.99  0.269628  0.668801 -9.35   5.39  -4.70
3   2003  -4.72  2.07  -8.01  -4.51  0.553992  1.000000  4.56  14.39   4.46
4   2004  -3.72 -0.52  -1.58   1.37  0.000000  0.623652  8.42  -0.03   6.97
5   2005  -1.10  2.22  -3.21  -6.54  0.596657  0.788410  4.13   3.13   9.09
6   2006   5.80  2.45  10.66   4.56  0.081812  0.636456  0.48   8.61   5.11
7   2007   2.93 -8.26   2.04   6.97  0.494612  0.591307  4.88  -1.43  12.49
8   2008 -16.31  1.67  -9.36   9.11  0.256653  0.000000  7.24   0.62 -10.06
9   2009  -2.85 -3.87   9.31  15.00  1.000000  0.454178  8.05   0.55   9.05
10  2010  -6.13  0.82   6.64   0.55  0.302837  0.723720  1.04   0.65  11.62
11  2011 -10.25 -3.14   9.38  -1.44  0.310314  0.626685 -2.93  -8.77  -1.15
12  2012  12.43  3.58  -1.66  -0.90  0.246976  0.816375 -0.95   0.56   8.46
13  2013   2.20 -5.66  -0.18   4.36  0.403343  0.492925 -1.72  -4.71   4.82
14  2014  -3.40  3.08   6.81  -0.12  0.557950  0.751685  1.44   3.02   0.13
15  2015   6.35  1.06  -4.62  -3.65  0.450407  0.547844  1.96  -6.58  -0.28
16  2016  -4.82 -7.62  10.75   1.44  0.469540  0.626348  4.23   1.71  -1.99
17  2017   4.59  3.72   3.31   1.42  0.457664  0.538747  5.84  -1.58  -1.30
18  2018   4.72 -4.85  -3.61   6.19  0.382010  0.567049  5.99   2.85  -6.42
19  2019  -0.29 -0.36   7.70   1.07  0.415439  0.536051 -5.69  -0.85   4.09
20  2020  -1.70 -6.36 -23.25  14.68  0.320211  0.827493  7.49   2.84  -1.23
21  2021  -2.48  6.56   1.11  -0.41  0.525621  0.603774  0.26   8.69   2.77
22  2022  -0.09 -3.46   4.33  -2.07  0.316033  0.410377  8.73   3.50  -3.75
23  2023  -2.45 -2.03   0.32   4.06  0.439850  0.692722  2.94  -2.53   2.00
...
20  0.681239  11.39   7.81   14.90
21  0.609745  -3.89   2.18   24.12
22  0.723588   4.14  -3.48    4.32
23  0.536658   5.52   7.94   19.42
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Observations:

The Min-Max scaling method has been applied to the 'May', 'Jun', and 'Oct' columns of the DataFrame.

Post-scaling, the values in these columns now fall within the range of 0 to 1. This normalization ensures uniform contribution from each feature in the analysis, preventing dominance by features with larger scales.

For instance, the 'May' column originally ranged from approximately -0.03 to 28.07. After scaling, these values are now mapped to a range of 0.00 to 1.00.

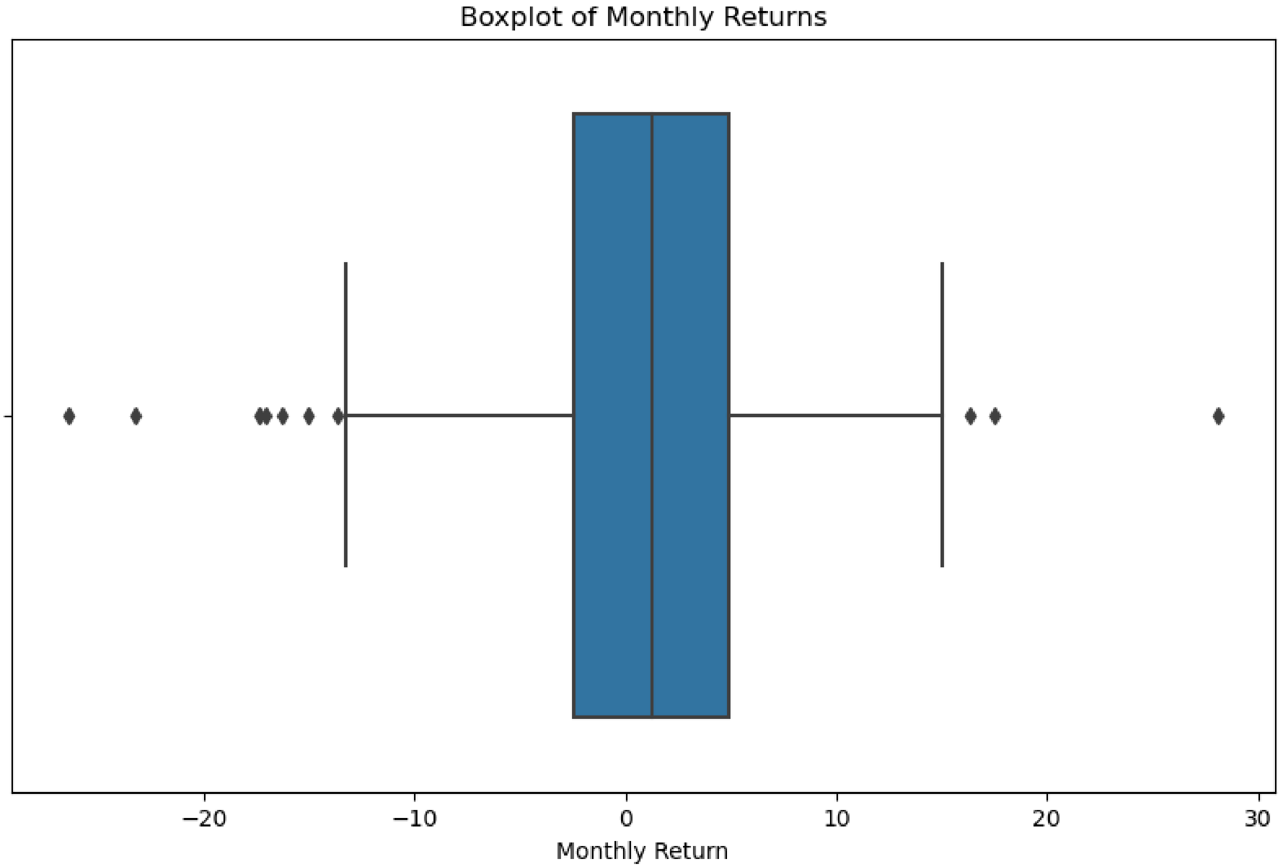
Similarly, the 'Jun' column values, originally spanning approximately -17.03 to 12.65, have been transformed to a range between 0.00 and 1.00 post-scaling.

This normalization technique optimizes the performance of machine learning algorithms, particularly those sensitive to input feature scales like gradient descent-based algorithms.

With scaled values in the DataFrame, further analysis or modeling becomes more reliable and effective.

5. Data Visualization

Boxplot visualization



1. A boxplot is a graphical representation used in exploratory data analysis to visualize the distribution of numerical data and identify skewness. Here's how to interpret a boxplot:
   - Minimum Score: Displayed at the end of the left whisker, the minimum score (excluding outliers) indicates the lowest value in the dataset.
   - Lower Quartile (Q1): Positioned at the boundary where twenty-five percent of scores fall below, Q1 delineates the first quartile of the dataset.
   - Median (Q2): Situated at the center of the box plot, the median divides the data into two equal parts, with half the scores greater than or equal to this value, and half less.
   - Upper Quartile (Q3): Representing the boundary where seventy-five percent of scores fall below, Q3 marks the third quartile of the dataset.
   - Maximum Score: Shown at the end of the right whisker, the maximum score (excluding outliers) indicates the highest value in the dataset.
   - Whiskers: Representing scores beyond the middle 50% of the data, the whiskers extend to the lower 25% and upper 25% of scores.

- Interquartile Range (IQR): Defined as the range between Q1 and Q3, the IQR illustrates the spread of the middle 50% of scores within the dataset.

2.Interpretation:

- The annual summary data significantly surpasses the monthly data, suggesting a substantial increase or accumulation over time.

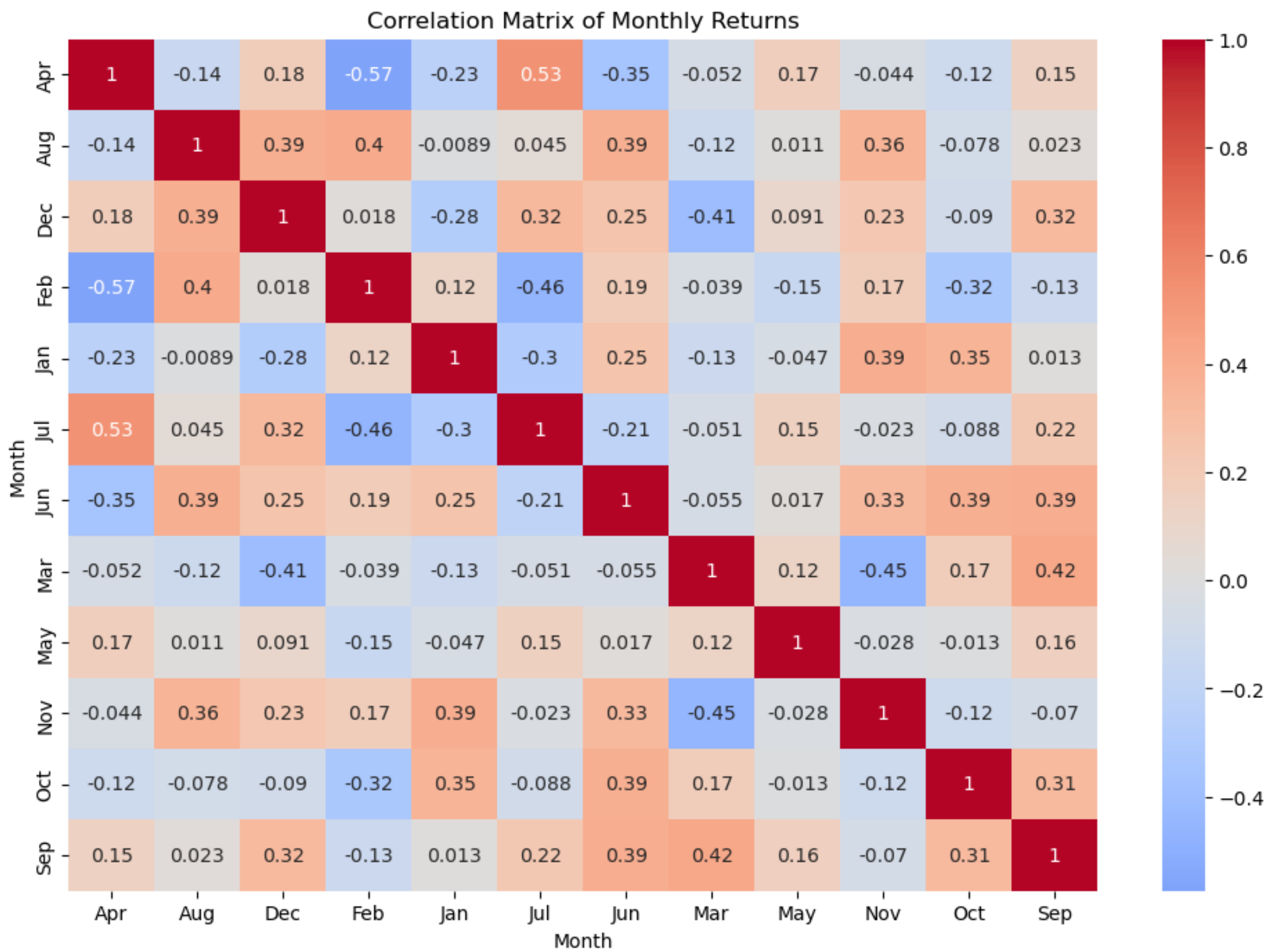
- The lack of boxes for individual months implies that their data points may exhibit greater concentration or lesser variability compared to the annual data.The x-axis represents **months of the year** from **January** to **December**, with additional labels for **“Year”** and **“Annual”**.

3. The y-axis presents numerical values ranging from 0 to 2000, with markings at intervals of 500.

4. Notable observations:

- A distinct marker is visible at the "Year" label on the x-axis, extending up to 2000 on the y-axis.
- Black markers are present at each month, although their significance remains unclear without additional context.

Histogram visualization



Histogram of Monthly Returns:

The distribution of monthly returns resembles a bell-shaped curve, suggesting an approximation of a norma l distribution, with a clustering of values around the mean. There are tails on both ends of the distribution, i ndicating months with notably higher and lower returns, though less frequently observed.

Boxplot of Monthly Returns:

The boxplot depicts a median close to zero, with returns evenly dispersed above and below the median. Out liers are evident at both extremes, suggesting months with exceptionally high or low returns, possibly corre sponding to periods of market volatility or exuberance.

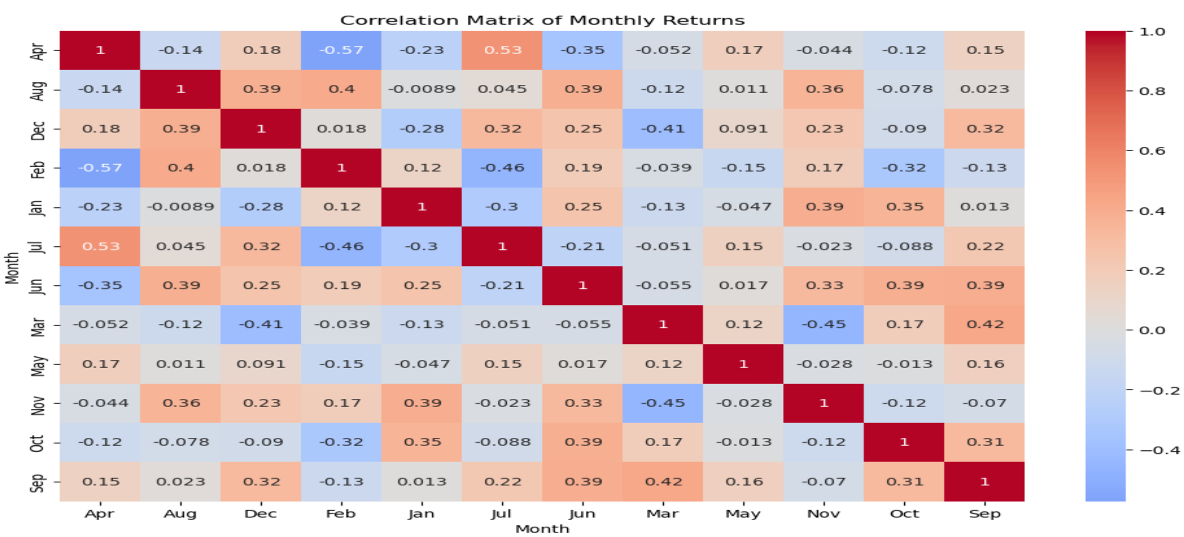
Bar Chart of Average Monthly Returns:

The bar chart reveals potential seasonality in returns, with certain months, like December, exhibiting higher average returns, possibly influenced by year-end effects such as the "Santa Claus rally." Conversely, some months, like January, February, and March, show lower average returns, prompting further exploration into phenomena like the "January effect."

Time Series of Monthly Returns:

The time series chart showcases volatility in returns, with no discernible long-term upward or downward tre nd, aligning with the random walk hypothesis of stock prices. Noteworthy fluctuations could be associated with specific market events, financial crises, or economic news releases, warranting closer scrutiny to comp rehend the underlying factors influencing such significant movements.

5.1 Correlation matrix



From the heatmap provided, here are a few observations we can infer:

High Volatility: Certain years exhibit high volatility, with monthly returns varying significantly, evident fro m the presence of both warm and cool colors in the same row. For instance, 2009 displays months with bot h high positive and negative returns, indicating a turbulent market environment.

Periods of Growth and Decline: Clear periods of growth and decline are observable. For instance, 2003 pre dominantly features warm colors, signifying a generally positive year for returns. Conversely, 2008 depicts a significant period of cooler colors, particularly towards the year's end, coinciding with the global financial crisis.

Extreme Values: Extreme positive or negative values occur in specific months, such as the significant negat ive value in May and June 2004 or the large negative value in March 2020, likely associated with the onset of the COVID-19 pandemic.

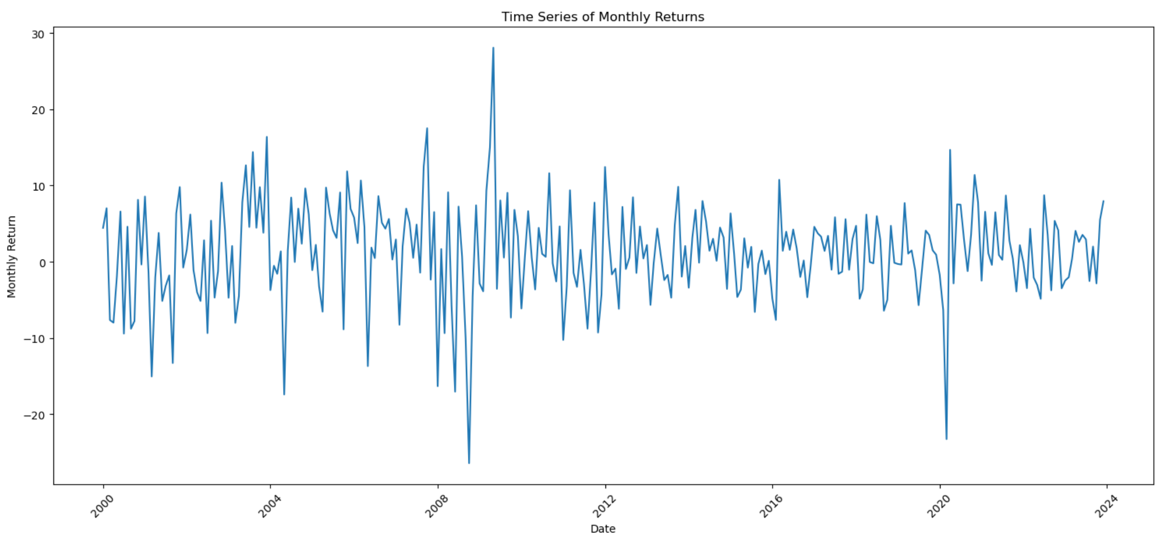
Overall Trend: Over the two decades, the general trend showcases more periods of positive returns than neg ative ones, evidenced by the prevalence of warm-toned squares over cool-toned ones.

Recent Performance: Recent years, like 2020 and 2021, display a mix of positive and negative returns with some extreme values, indicating ongoing market instability and uncertainty post the initial shock of the CO VID-19 pandemic.

Annual Perspective: The annual columns on the far right provide a succinct visual summary of yearly perfo rmance, indicating that despite monthly fluctuations, annual returns may still conclude positively in several years.

Historical Context: The observed patterns can be correlated with specific historical events such as market cr ashes, economic booms, or global crises, offering insights into how these events influence market behavior over time when cross-referenced with historical data.

5.2 Time Series Monthly Returns



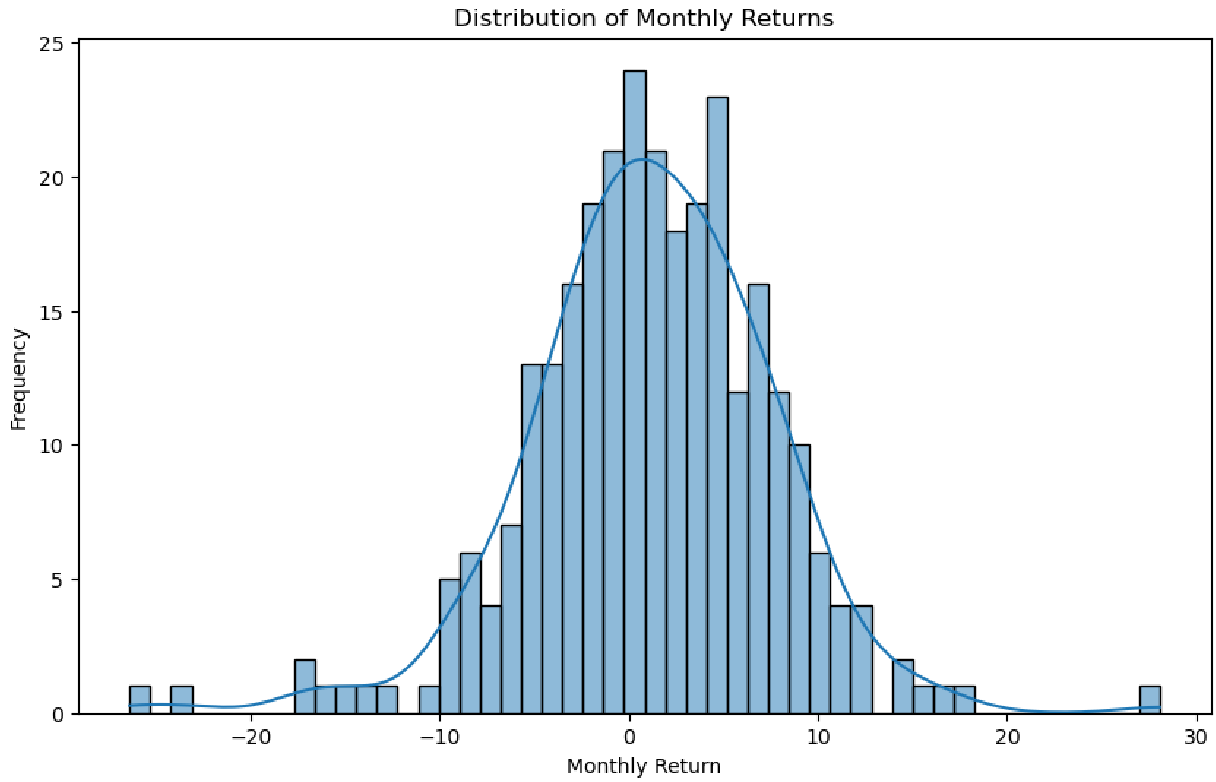
Trend Over Time: The overall direction of the line plot reveals whether there is a consistent trend in monthl y returns across the specified period. An upward slope suggests increasing returns over time, while a down ward slope indicates decreasing returns.

Seasonality: Look for recurring patterns or cycles in the plot, which may indicate seasonality in monthly ret urns. These fluctuations could be related to specific time periods, such as certain months of the year or quar ters.

Volatility: Periods of high or low volatility in monthly returns can be identified based on the fluctuations in the line plot. Sharp peaks and valleys indicate periods of high volatility, whereas smoother segments sugge st more stable returns.

Outliers or Anomalies: Significant spikes or dips in the plot may indicate outliers or anomalies in monthly returns, which could be caused by extraordinary events or unusual market conditions.

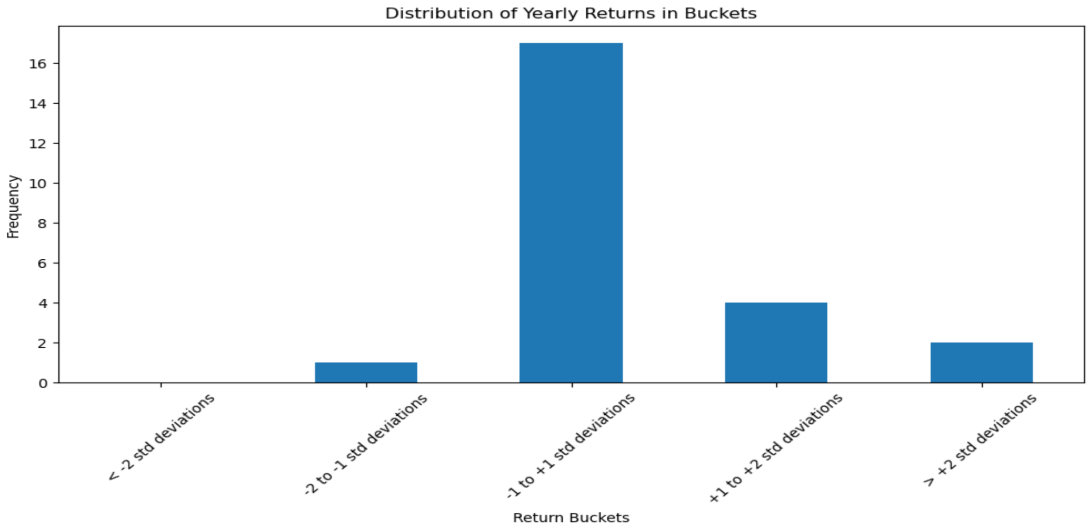
5.3 Distribution of Monthly Returns



Negative Returns: In the histogram, negative returns are depicted on the left side of the plot, with bars extending below the zero line. The tallest bar in the negative returns range denotes the most frequent range of negative monthly returns observed in the dataset.

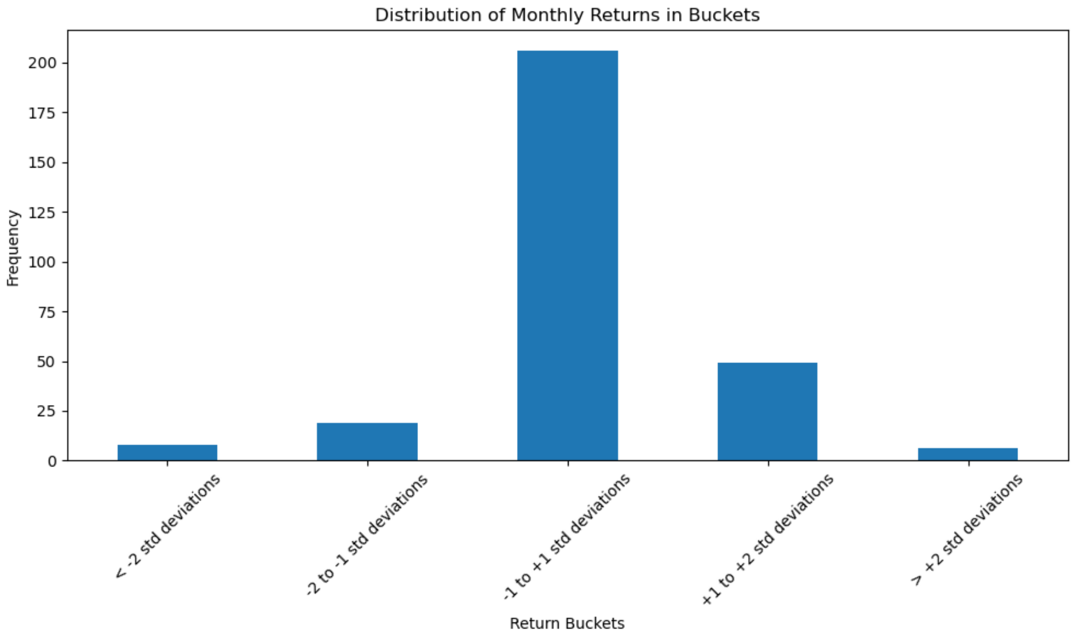
Positive Returns: Positive returns are represented on the right side of the plot, with bars extending above the zero line. The tallest bar in the positive returns range indicates the most frequent range of positive monthly returns observed in the dataset.
Magnitude of Returns: The height of the bars signifies the frequency of returns within each range. Taller bars indicate higher frequencies of returns falling within the corresponding range.
Skewness: If one side of the histogram (positive or negative returns) exhibits significantly taller bars compared to the other side, it suggests skewness in the distribution of monthly returns towards that side.

5.4 Distribution of Yearly Returns in Buckets



• X-Axis: The x-axis delineates the return buckets, with each bucket labeled to denote various ranges or categories of yearly returns, such as "Low Returns," "Moderate Returns," "High Returns," etc.
• Y-Axis: The y-axis portrays the frequency or count of yearly returns, representing how many data points are categorized within each return bucket.
• Bar Heights: The height of each bar correlates with the frequency of yearly returns within the corresponding bucket. Taller bars signify a higher frequency of returns falling into that bucket, whereas shorter bars indicate a lower frequency.

5.5 Distribution of Monthly Returns in Buckets



X-Axis: Depicts the return buckets or categories, with each bucket labeled to denote various ranges or classifications of monthly returns.

Y-Axis: Exhibits the frequency or count of monthly returns, indicating the number of data points categorize d within each return bucket.

This bar plot provides a visual summary of the distribution of monthly returns across different buckets or ca tegories, enabling viewers to swiftly comprehend the frequency distribution and discern any patterns or tren ds in the data.

6.ALGORITHMS

1. SARIMAX MODEL

Among the array of available approaches, the SARIMAX (Seasonal Autoregressive Integrated Moving Ave rage + exogenous variables) model emerges as a potent tool for modeling and forecasting both trends and se asonal variations in temporal data. It distinguishes itself by integrating exogenous variables into the analysi s, thereby enhancing prediction accuracy.

The SARIMAX model marks a significant advancement in time-series analysis, facilitating the inclusion of covariates. By incorporating external variables, it enriches the analytical framework, offering deeper insigh ts into future trends and forecasts. However, mastery of model parameters and a comprehensive understand ing of results are paramount to derive pertinent and dependable predictions from this methodology.

7.Implementation

Step 1: run the python code app.py

```
127.0.0.1 - - [11/Apr/2024 14:03:13] "POST /predict HTTP/1.1" 200 -
PS D:\everythingfromc\si\nifty> python -u "d:\everythingfromc\si\nifty\app.py"
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 891-660-042
```

Create Flask App:

- Import the Flask class and instantiate it to create a Flask app instance.
- Define routes using the @app.route('/myurl') decorator to specify URL endpoints for different functionalities.

Create Index.html File:

- Create an HTML file named index.html to serve as the user interface for inputting data.
- Design the HTML page to contain input fields for 2 attributes required for the Annual return prediction.

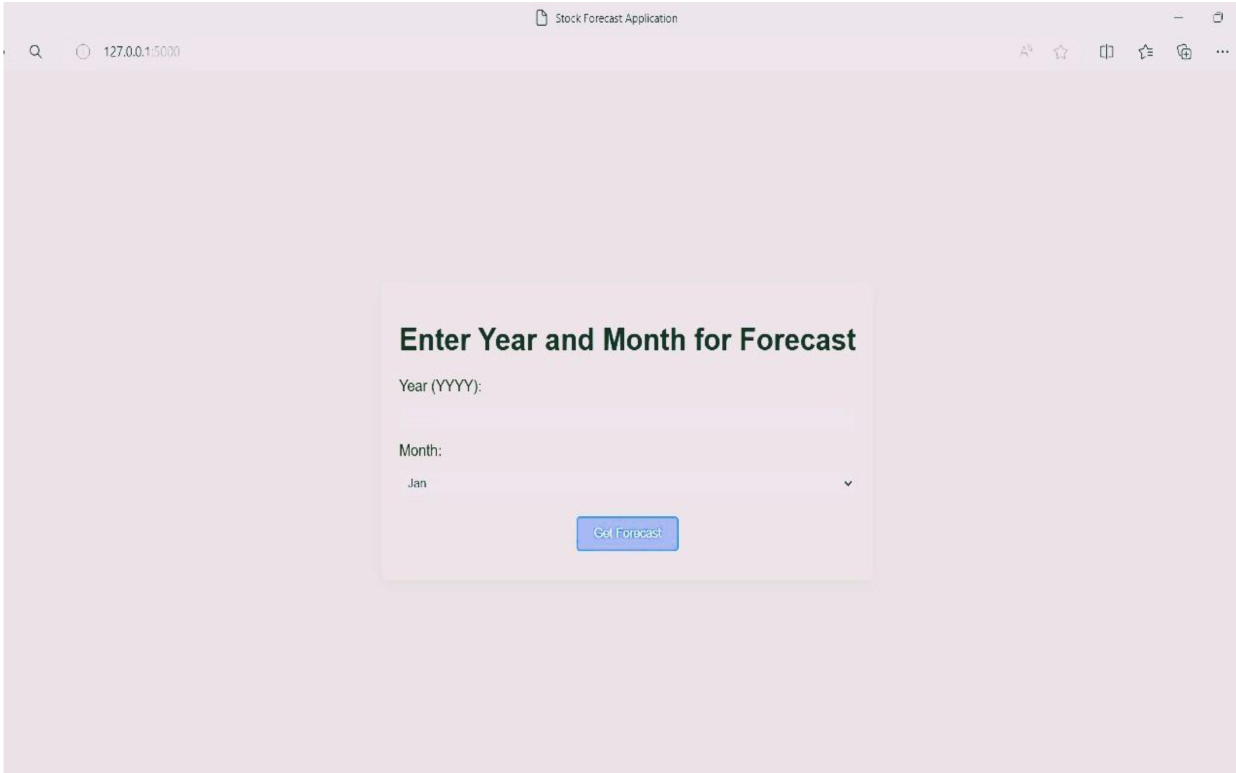
Implement Flask Routes:

- Define Flask routes to handle requests from the client-side.
- For example, create a route to render the index.html template and another route to handle form submissions.

Run Flask App:

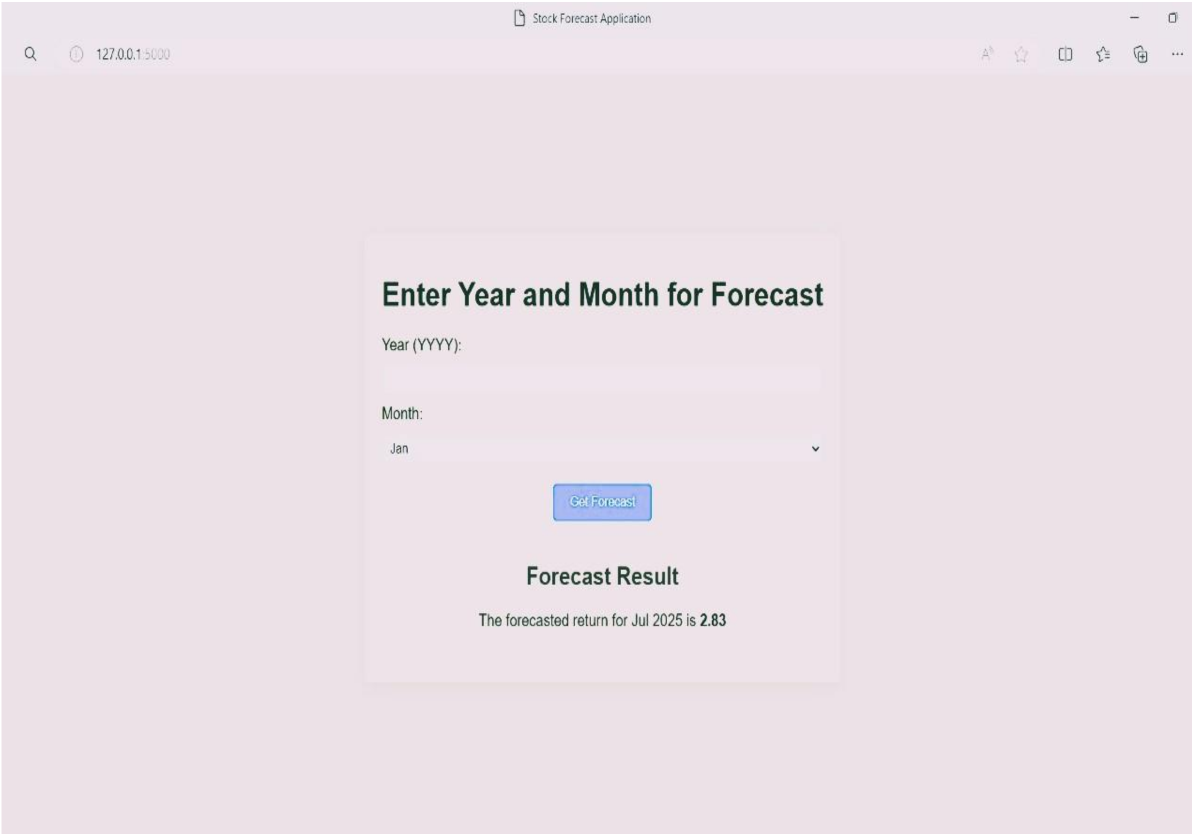
- Run the Flask app using the flask run command or by executing the Python script containing the Flask app.
- Access the Flask app through a web browser to interact with the user interface and make predictions.

Step 2: open the link in browser:



- Once the web application is open, you'll see input fields corresponding to attribute required for predictin g Annual stock price.
- Enter the values for attribute in the respective input tabs.
- Select the month whatever You want to predict.

Step 3: click on "Get Forecast" button



- After entering the necessary input values, click on the "Get Forecast" button. This action triggers the Fla sk app to process the input data using the trained model.
- The predicted value will be displayed on the web page.

9.Final Observation

- We receiving the predicted Annual stock price.