**Data Visualization and Inference Modeling-The Case of Nifty**

CAPSTONE PROJECT REPORT

By

Team 16

E Saikiran [22R15A0514]

Naveen Rampa [21R11A0589]

J Sinduja [21R11A0574]

Lakshanya S [21R11A05H3]

A Bhavya Sri [21R11A05A5]



**Department of Computer Science and Engineering**

**Accredited by NBA**

**Geethanjali College of Engineering and Technology**

**(UGC Autonomous)**

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

Cheeryal (V), Keesara (M), Medchal.Dist.-501 301

March 2024

**ABSTRACT**

This project focuses on analysing stock market data, specifically the Nifty index, spanning a period of 20 years. The dataset includes monthly and yearly returns data, providing insights into stock price variations over time. The primary emphasis is placed on the year 2020, a period marked by significant global events, notably the COVID-19 pandemic. The analysis aims to identify patterns, trends, and anomalies within the stock market during this critical year, offering valuable insights for investors, businesses, and researchers. Understanding the fluctuations and behaviour of the stock market during such unprecedented times is crucial for making informed investment decisions and maximizing returns.**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Chapter Name** | **Page No** |
|  | **ABSTRACT** |  |
| **1** | **INTRODUCTION** | 1 |
| **2** | **PROBLEM STATEMENT** | 2 |
| **3** | **OBJECTIVES** | 3 |
| **4** | **METHODOLOGY** | 4 |
|  | **4.1 Data Source** | 4 |
|  | **4.2 Exploratory Data Analysis** | 5 |
|  | 4.2.1 Checking for Data Consistency | 6 |
|  | 4.2.2 Revised data frame | 7 |
|  | 4.2.3 Data Visualization | 8 |
| **5** | **BUILDING A PREDICTIVE MODEL** | 25 |
|  | **5.1 Linear Regression** | 25 |
|  | **5.2 Ridge Regression** | 25 |
|  | **5.3 Lasso Regression** | 25 |
|  | **5.4 ElasticNet Regression** | 26 |
|  | **5.5 Decision Tree Regression** | 26 |
|  | **5.6 Random Forest Regression** | 26 |
|  | **5.7 Support Vector Regression(SVR)** | 27 |
| **6** | **Observation** | 28 |
| **7** | **Final Implementation** | 29 |
|  | **Conclusion** | 31 |

**1. INTRODUCTION**

This project delves into the comprehensive analysis of stock market data, specifically focusing on the Nifty index, which encapsulates 23 years of financial information. The dataset encompasses monthly and yearly returns data, providing a detailed look into the performance of the stock market over an extended period. The primary objective of this analysis is to unravel insightful patterns, trends, and correlations within the stock market dynamics, with particular emphasis on the pivotal year of 2020. The year 2020 stands out due to its significance in global events, notably the COVID-19 pandemic, which had a profound impact on financial markets worldwide. By scrutinizing the data from 2020 alongside the broader historical context, this project aims to uncover valuable insights that can benefit investors, analysts, and financial decision-makers in navigating the complexities of the stock market and making informed investment choices.



**2. PROBLEM STATEMENT**

The stock market is a dynamic and complex system influenced by a multitude of factors ranging from economic indicators to global events. The challenge lies in effectively analysing and understanding this data to make informed investment decisions. This project focuses on analysing the Nifty index's stock market data, specifically targeting the year 2020, which was marked by significant global events such as the COVID-19 pandemic. The goal is to extract meaningful insights, patterns, and trends from the data to aid investors, analysts, and decision-makers in navigating the volatile stock market landscape and optimizing investment strategies. The project aims to address the following key questions:

What were the major trends and patterns in the Nifty index's monthly and yearly returns over the past 23 years?

How did the stock market perform during the critical year of 2020, especially in the context of the COVID-19 pandemic?

Are there any discernible correlations or anomalies in the stock market data that could provide valuable insights for investment decisions?

By addressing these questions, the project seeks to contribute to a better understanding of the stock market's behaviour, facilitate data-driven investment decisions, and enhance overall portfolio management strategies.

**3. OBJECTIVES**

Data Analysis: Conduct a comprehensive analysis of the Nifty index's stock market data spanning 23 years, focusing on monthly and yearly returns.

Trend Identification: Identify and analyse major trends, patterns, and fluctuations in the Nifty index's stock prices over the analysed period.

COVID-19 Impact Analysis: Specifically analyse and assess the impact of the COVID-19 pandemic on the stock market during the year 2020, examining how it influenced stock prices, volatility, and investor sentiment.

Correlation and Anomaly Detection: Explore correlations between different economic indicators, market events, and stock market performance, aiming to detect any anomalies or unexpected trends that could provide valuable insights.

Investment Strategies: Derive data-driven investment strategies and recommendations based on the analysis, focusing on optimizing portfolio management, risk mitigation, and maximizing returns for investors.

Visualization and Reporting: Create visualizations such as charts, graphs, and dashboards to present the findings and insights effectively, facilitating easy understanding and decision-making for stakeholders.

Impact Assessment: Evaluate the potential impact of the analysis and recommendations on investment decisions, risk management strategies, and overall portfolio performance for investors and financial institutions.

**4. METHODOLOGY**

**4.1 Data Source**

The dataset contains the monthly returns, Yearly returns. The data set provided to us comprised of 13 features and 23 observations

Year: This parameter represents the calendar year for which the stock market data is recorded. In the dataset you provided, it likely spans multiple years, allowing for a longitudinal analysis of stock market trends over time.

Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec: These parameters represent the monthly returns of the Nifty index or stock prices for each respective month. For example, "Jan" represents the stock market performance or returns for the month of January, "Feb" for February, and so on until "Dec" for December.

Annual: This parameter represents the annual returns or performance of the Nifty index or stock market for each year. It is often calculated as the cumulative returns or average performance across all months in a given year. The annual parameter provides a consolidated view of the stock market's overall performance on a yearly basis.

These parameters capture the monthly and yearly performance of the stock market, allowing for detailed analysis of trends, fluctuations, and overall market behaviour over time.

**4.2 Exploratory Data Analysis**

Exploratory Data Analysis (EDA) is an essential step in the machine learning process, where the main goal is to analyse and understand the characteristics of the data before applying any modeling techniques. EDA is the process of summarizing the main characteristics of the data, such as the distribution, the relationship between variables, and identifying any patterns or anomalies that may exist.

EDA is an important step because it allows us to gain a deeper understanding of the data and the underlying relationships between variables, which can help inform decisions about feature engineering, data preprocessing, and model selection. By performing EDA, we can identify any missing or erroneous data, outliers, and inconsistencies in the data, which can be addressed before training any machine learning models.

* **Information about the Features & their data types**

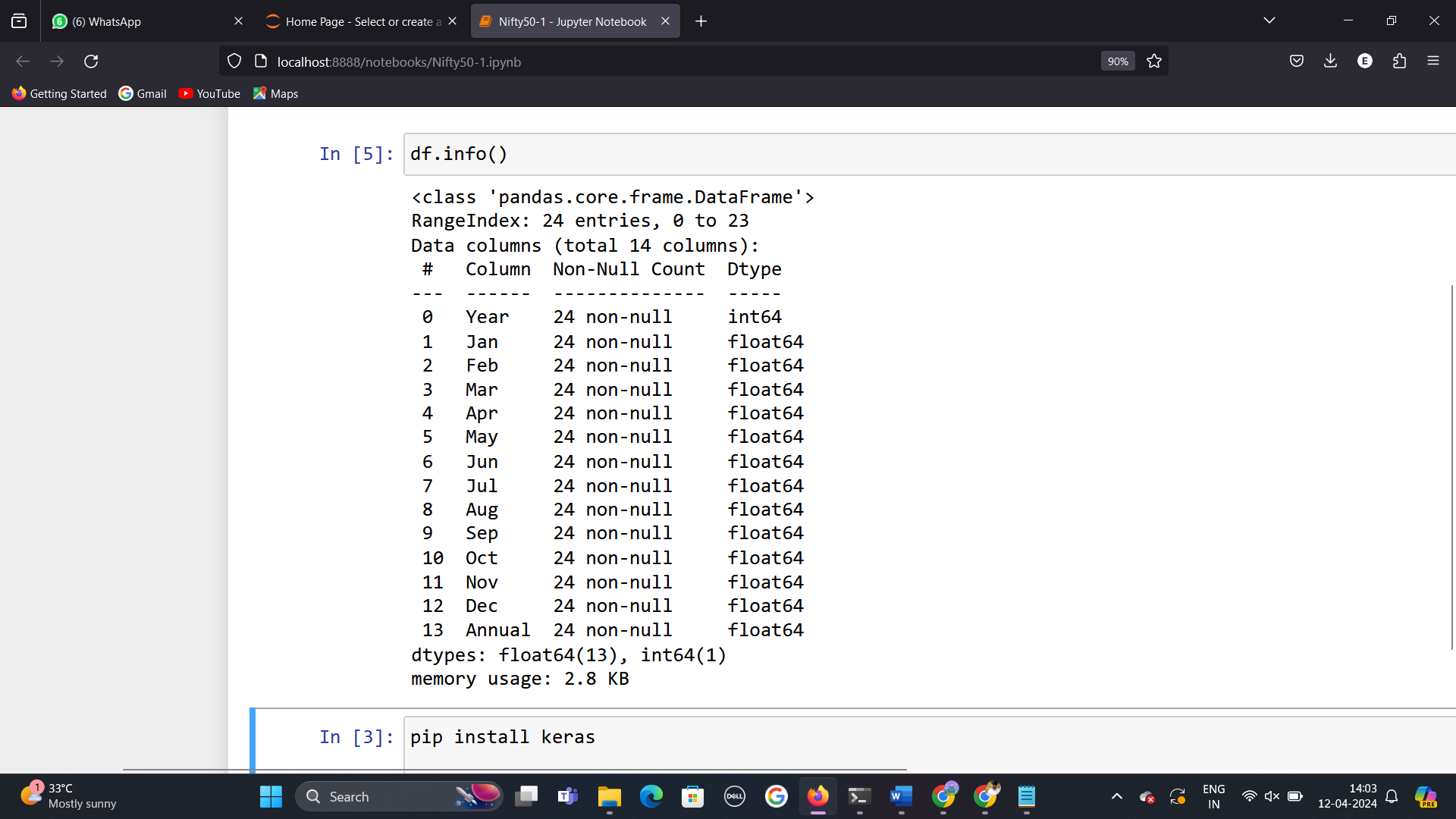


Fig 4.2.1: Dataset Information

* **Observations:**

Year (int64): This column likely represents the year for which the data is recorded. Being an integer type (int64), it indicates discrete year values.

Jan to Dec (float64): These columns appear to represent monthly data, as they are of type float64 indicating decimal values. Each column could represent some metric or value recorded monthly.

Annual (float64): This column seems to provide annual aggregated data, as it is also of type float64 like the monthly columns but presumably represents the total or average value for the entire year.

Based on these data types. This structure suggests a comprehensive analysis of trends over time, seasonal variations, and overall performance.

**4.2.1 Checking for Data Consistency**

* No duplicates found.



Fig 4.2.1.1: Number of Duplicate rows in a dataset

* Unique Values.

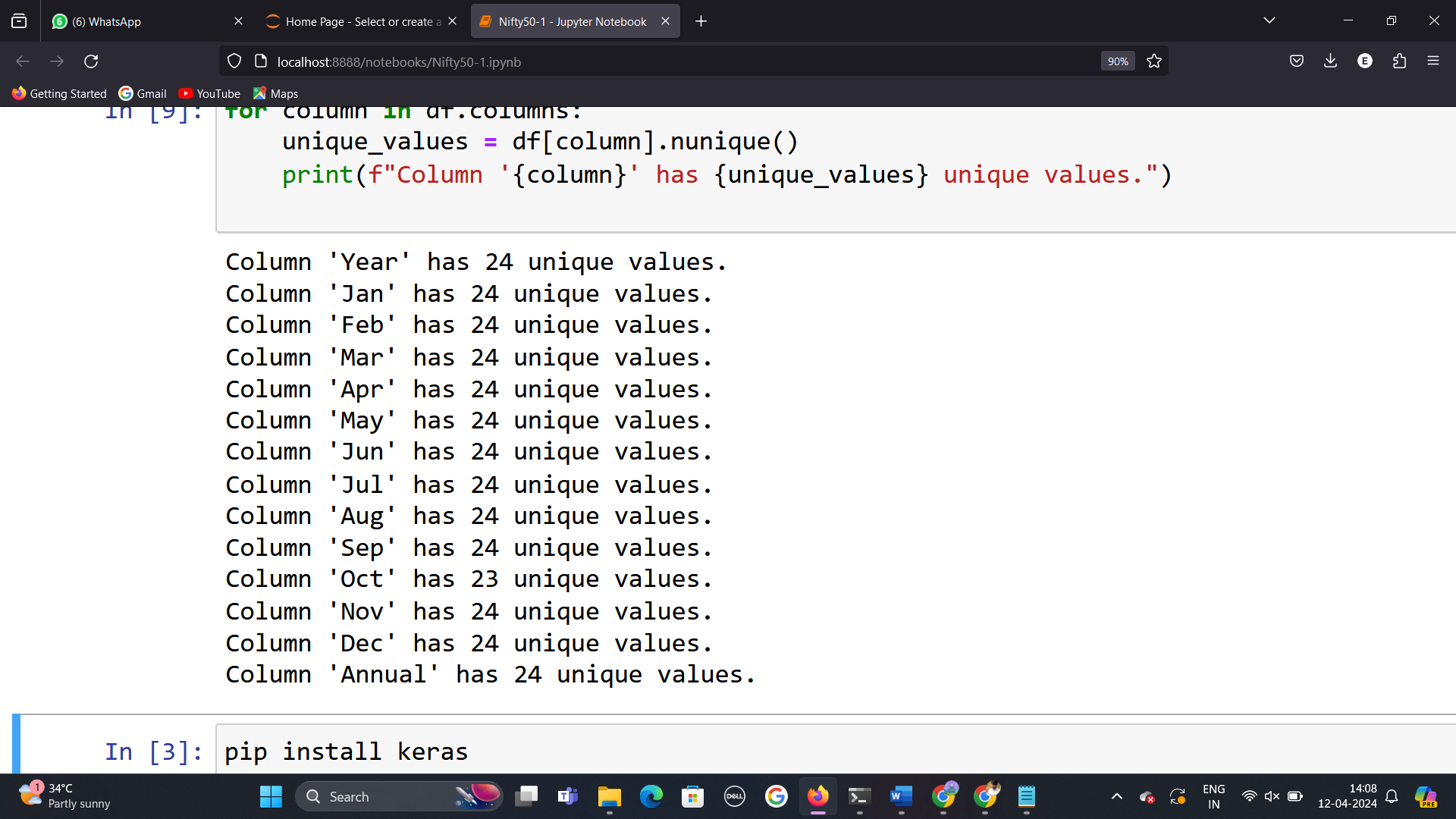


Fig 4.2.1.2: Unique value count for each column

**Observations:** It appears that most columns in the given dataset have 24 unique values, indicating that there is data available for each of the 24 years in your dataset. However, the 'Oct' column has 23 unique values, suggesting that there may be missing data or some years without data in this particular column.

Having 24 unique values for most columns indicates a consistent and complete dataset over the years, which is essential for conducting meaningful analysis and drawing reliable conclusions. Missing or incomplete data, as seen in the 'Oct' column, may require further investigation to understand the reasons behind the missing values and how they might affect your analysis or modeling efforts.

**4.2.2 Revised data frame**

After careful observation on column named ‘Oct’ The unique function contains two equal values in the year 2019 and 2020. That is why unique treated it as the missing value. But there is no missing values or outliers in the given data.

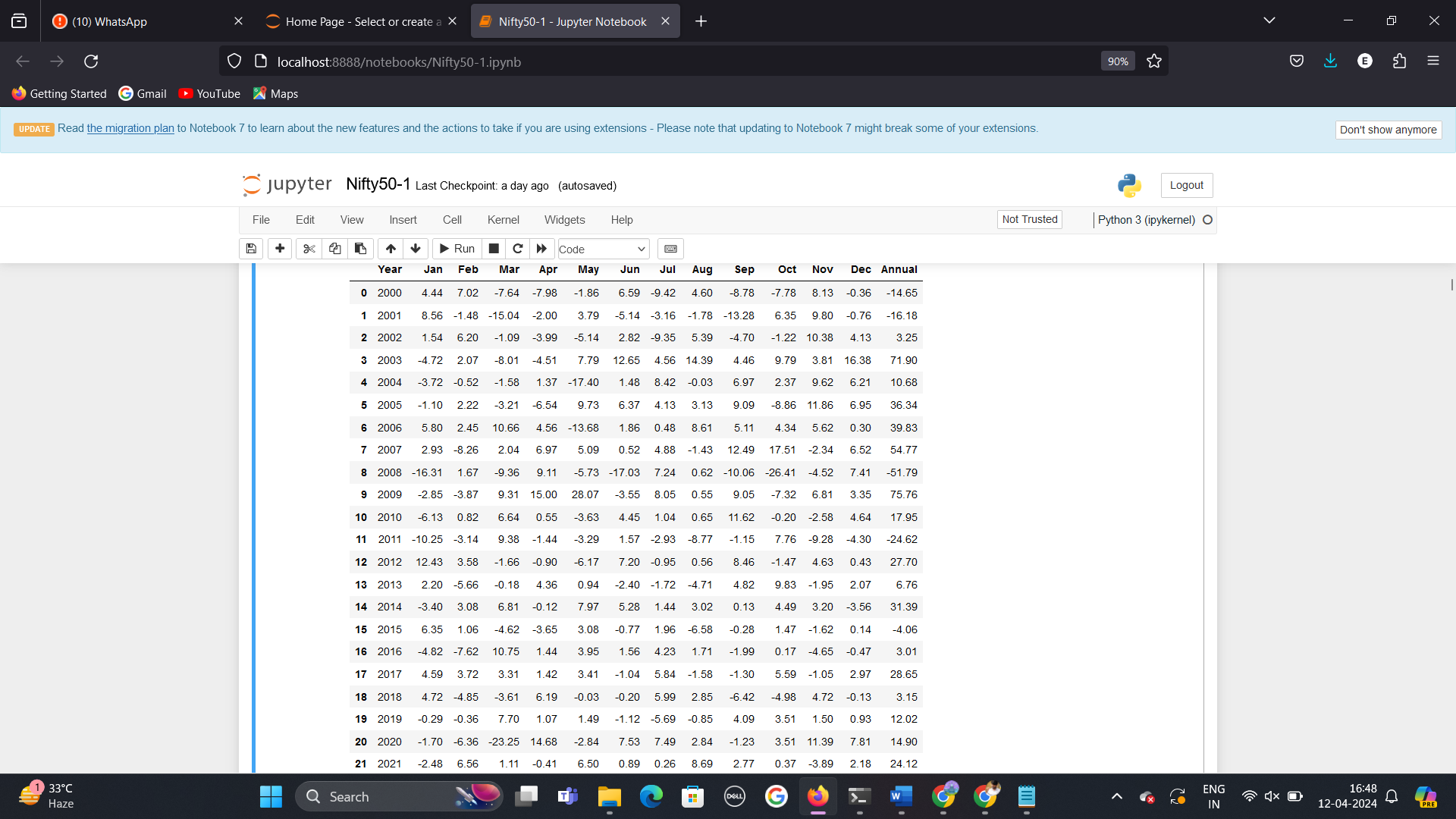


Fig 4.2.2.1: Revised DataFrame.

**4.2.3 Data Visualization**

* **Analysing the target variables**

1. **“Annual”**

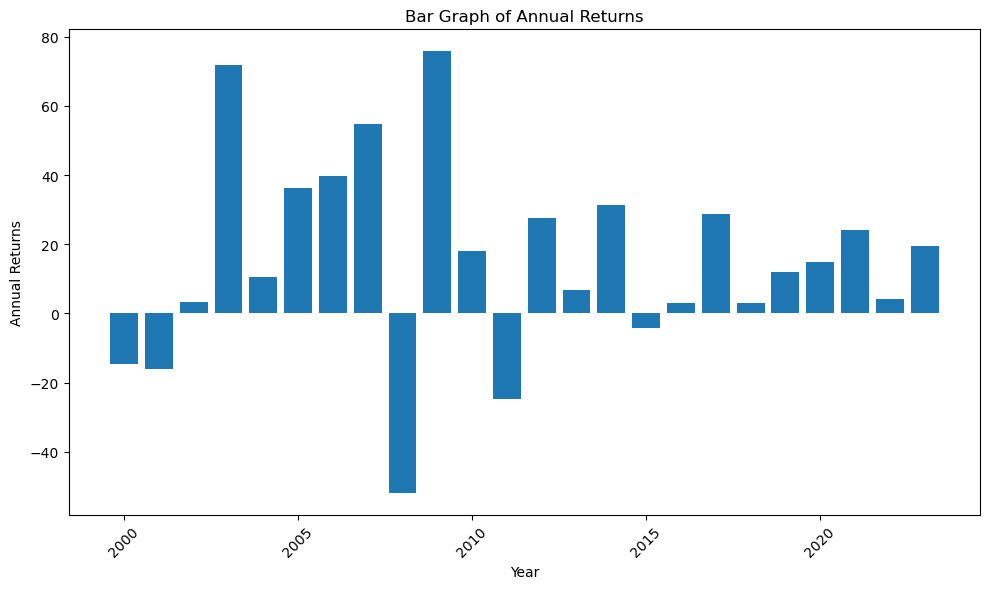
****

Fig 4.2.3.1: Histogram showcasing Annual returns for the year 2000 to 2023

**Observations:**

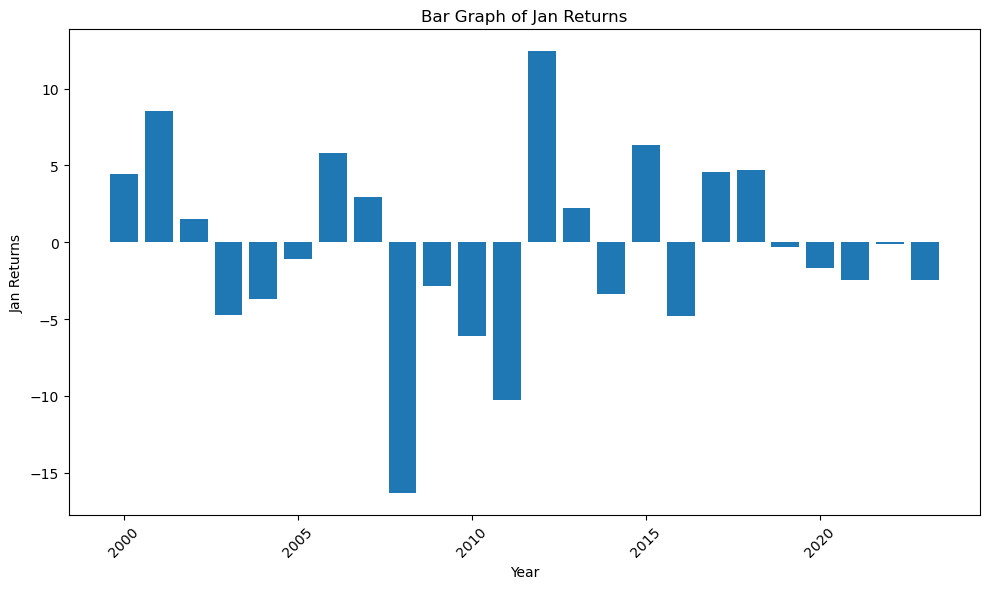
X-Axis: Represents the years, ranging from 2000 to 2023.

* Y-Axis: Represents annual returns, with values ranging from -40 to 80.
* Blue bars represent the annual return for each year.
* The height of each bar corresponds to the value of the return.
* Some years around 2005 and 2010 show high positive returns (deep blue).
* Around 2015, there are negative returns (light blue).

Other years exhibit varying return

* **Analysing the feature variables**

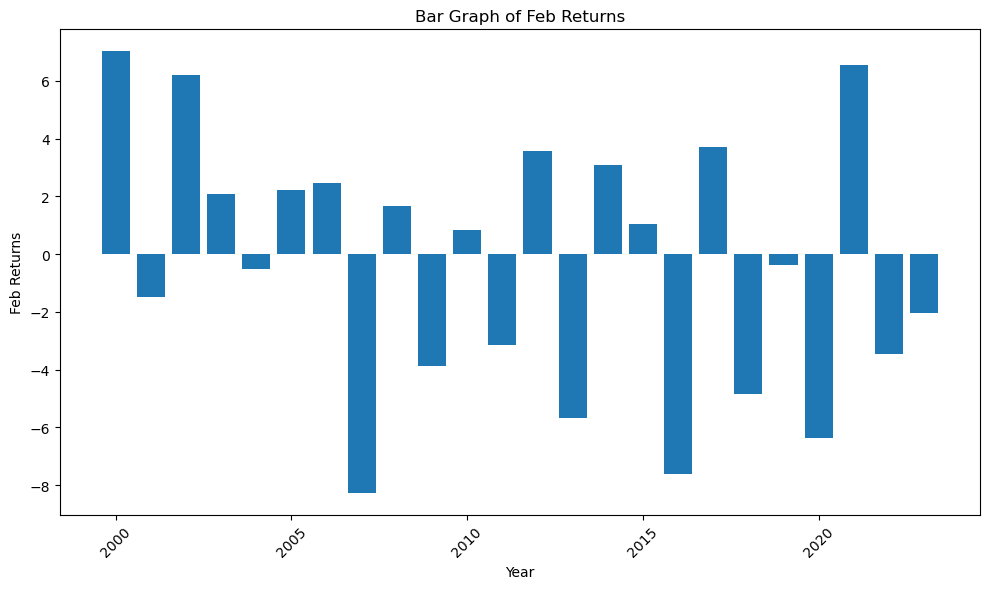
**1.’January’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents the monthly returns for January, with values ranging from -15 to 10.
* The highest return was in 2012 at around 10%.
* The lowest return was in 2008 at around -15%.

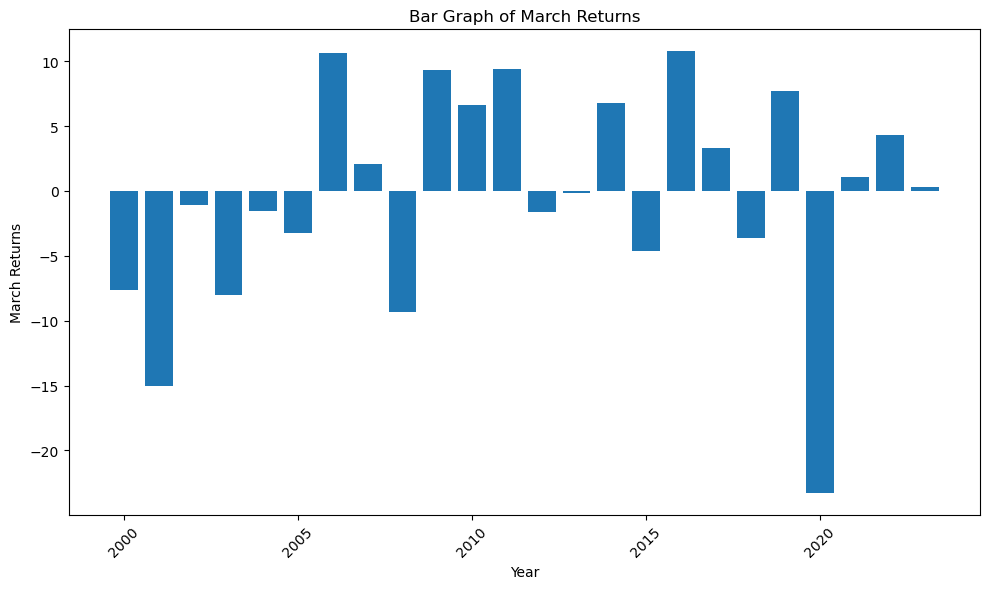
**2.’Feb’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -8 to 6.
* The highest return was in 2000 at around 6%.
* The lowest return was in 2007 at around -8%.

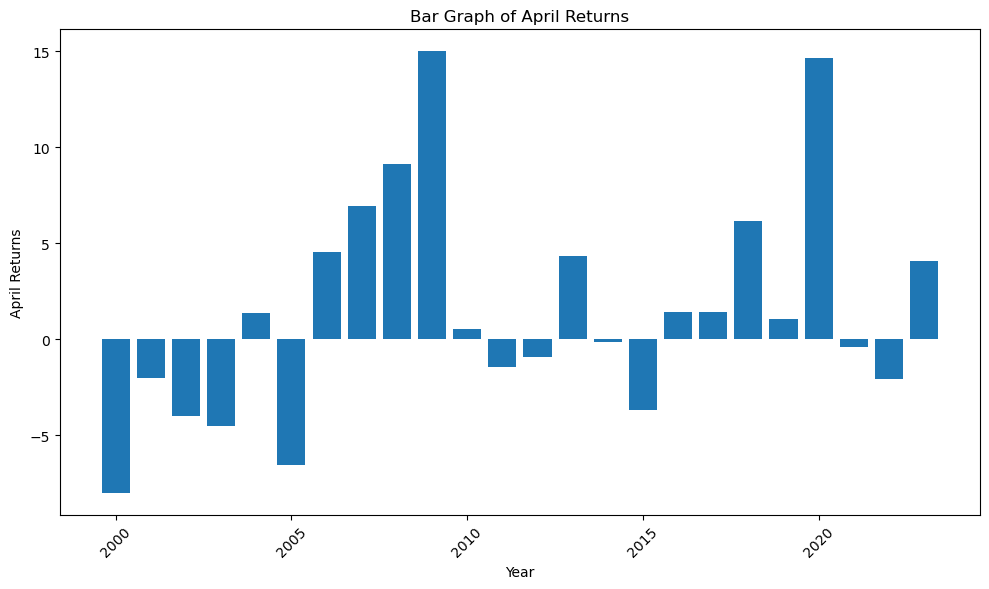
**3.’March’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -20 to 10.
* The highest return was in 2006 and 2016 at around 10%.
* The lowest return was in 2020 at around -20%.

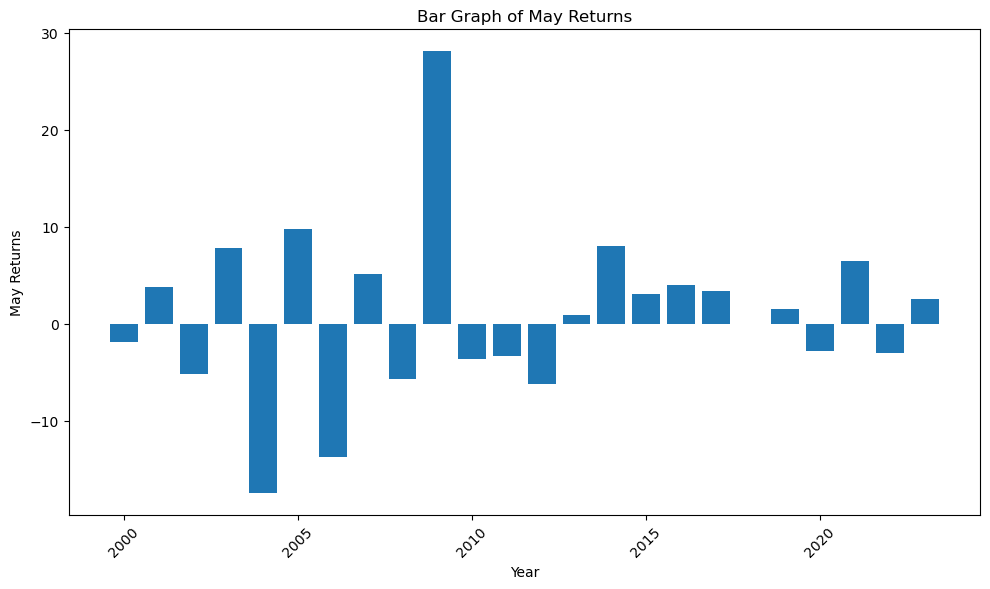
**4.’April’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -5 to 15.
* The highest return was in 2009 at around 15%.
* The lowest return was in 2000 at around -10%.

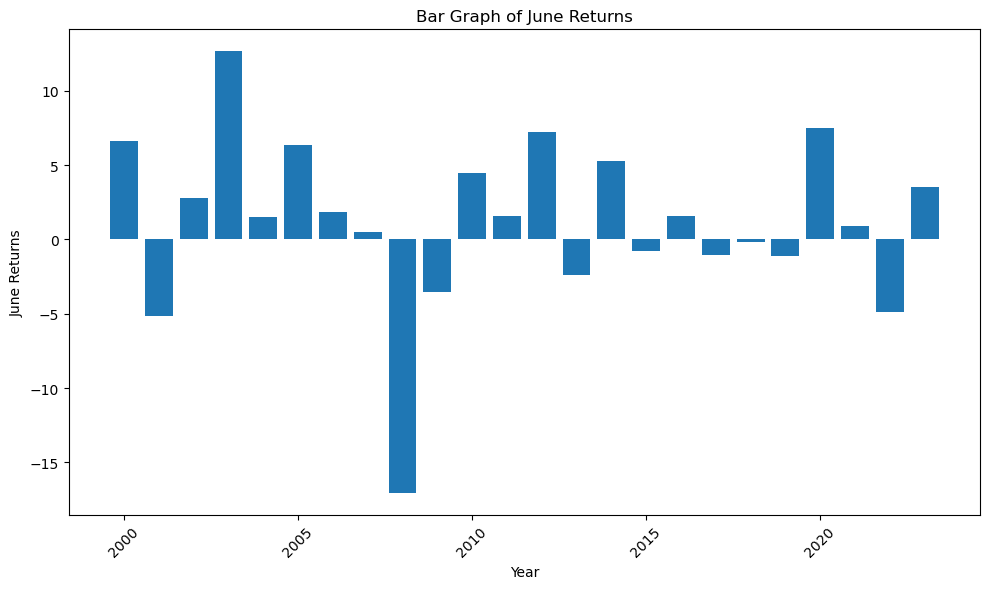
**5.’May’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -10 to 30.
* The highest return was in 2009 at around 20%.
* The lowest return was in 2004 at around -10%.

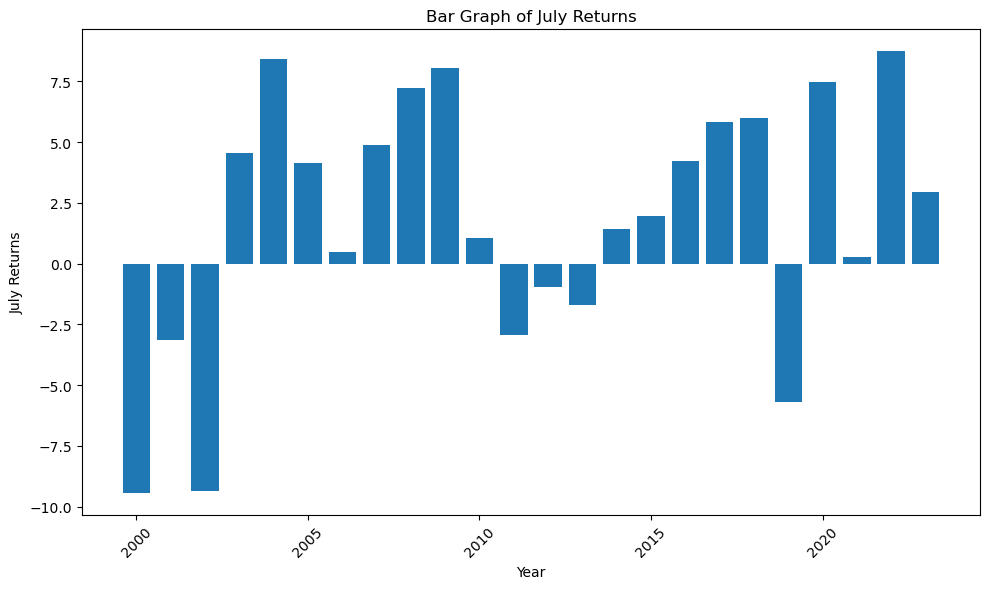
**6.’June’8**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -15 to 10.
* The highest return was in 2003 at around 10%.
* The lowest return was in 2008 at around -15%.

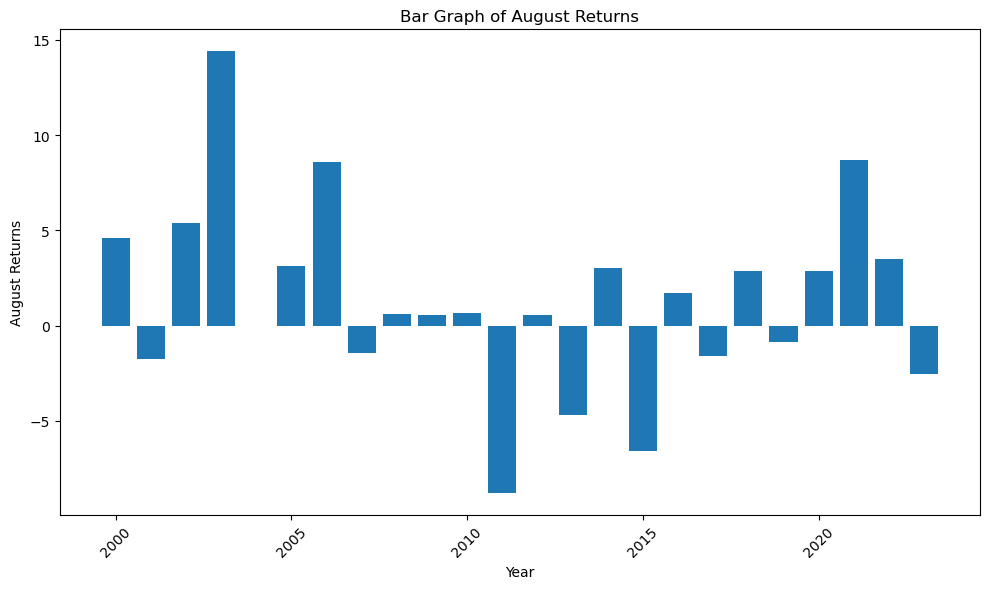
**7.’July’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -10.0 to 7.5.
* The highest return was in 2004 and 2022 at around 7.5%.
* The lowest return was in 2000 and 2001 at around -10%.

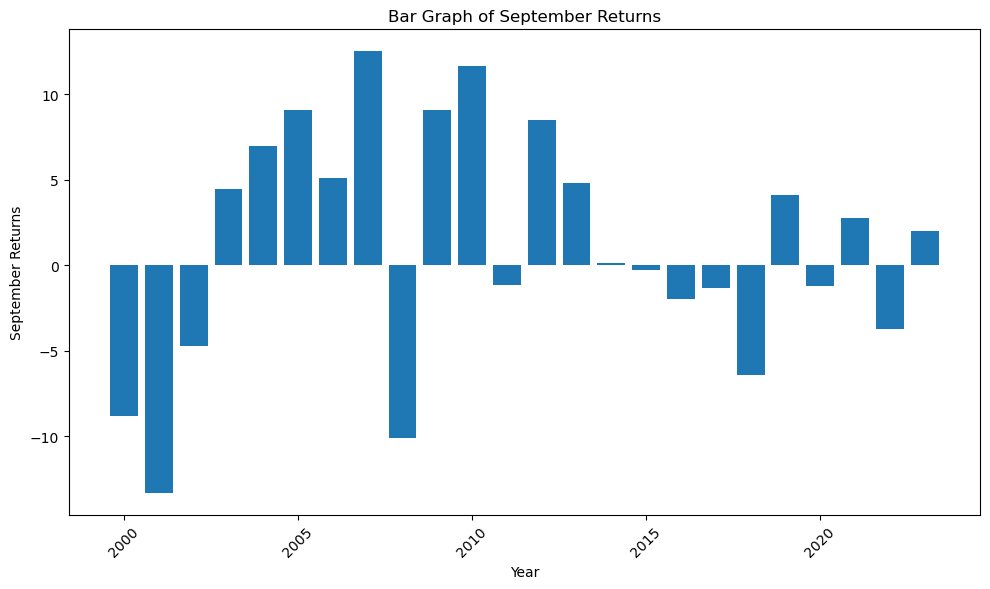
**8.’August’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -5 to 15.
* The highest return was in 2003 at around 15%.
* The lowest return was in 2011 at around -5%.

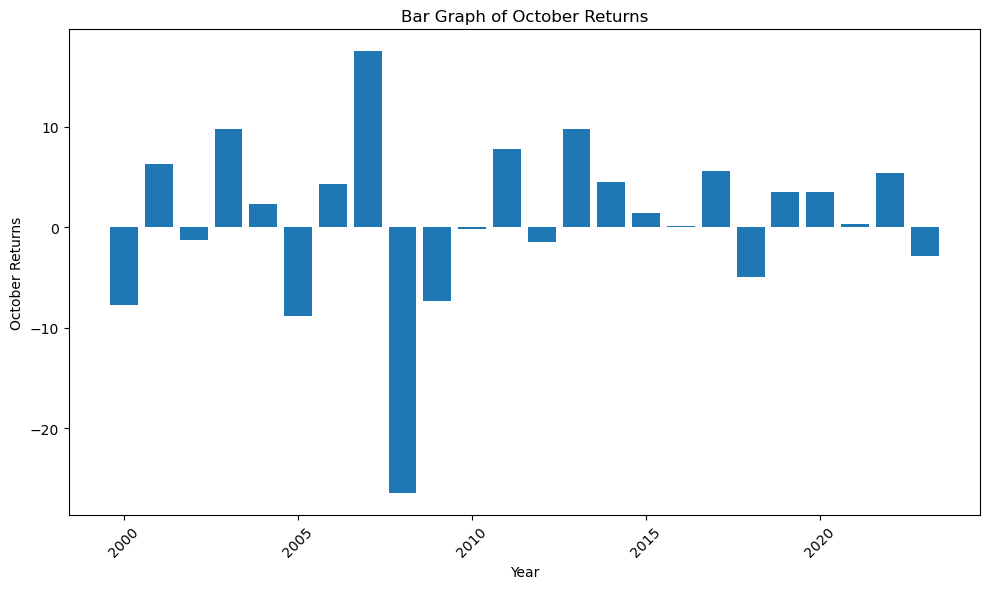
**9.’September’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -10 to 10.
* The highest return was in 2007 at around 10%.
* The lowest return was in 2001 at around -10%.

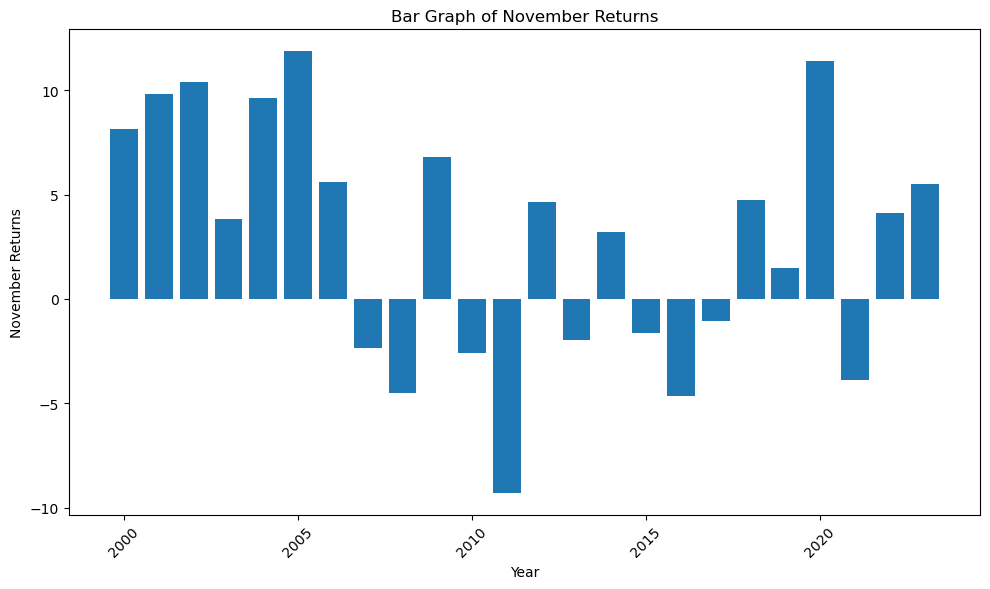
**10.’October’**

****

**Observation:**

* The bar graph of October in the year 2019 and 2020 indicates equal monthly returns.
* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -20 to 10.
* The highest return was in 2007 at around 10%.
* The lowest return was in 2008 at around -20%.

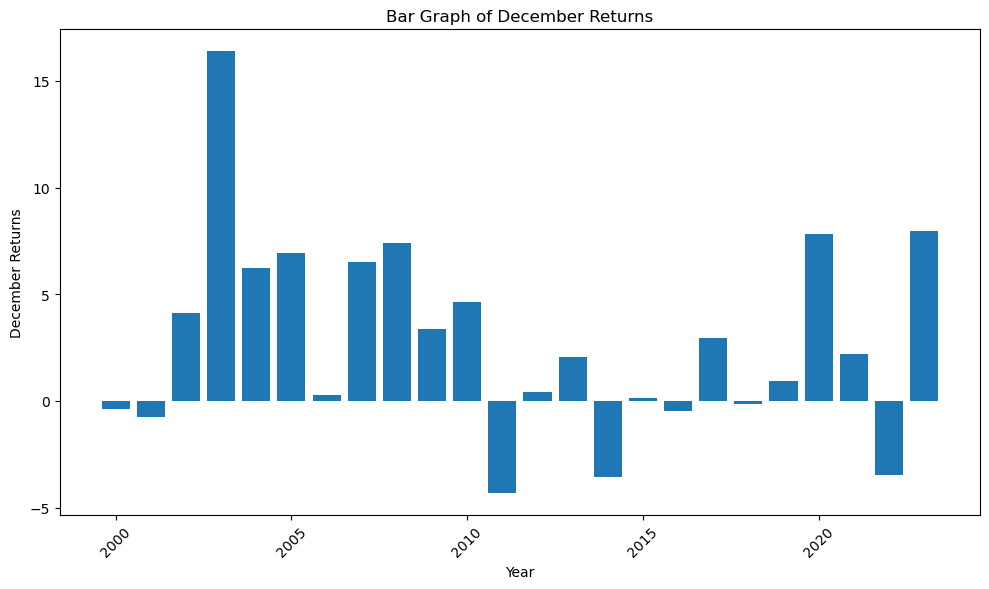
**11.’November’**

****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -10 to 10.
* The highest return was in 2005 at around 10%.
* The lowest return was in 2011 at around -10%.

**12.’December’**

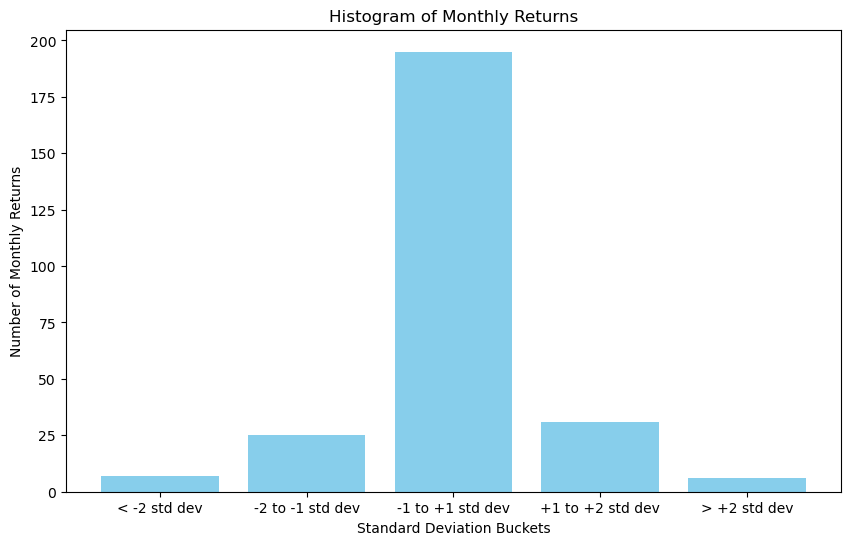
****

**Observation:**

* X-Axis: Represents the years, ranging from 2000 to 2023.
* Y-Axis: Represents annual returns, with values ranging from -5 to 15.
* The highest return was in 2003 at around 15%.
* The lowest return was in 2011 at around -5%.
* **Analysing Data with Heatmap.**

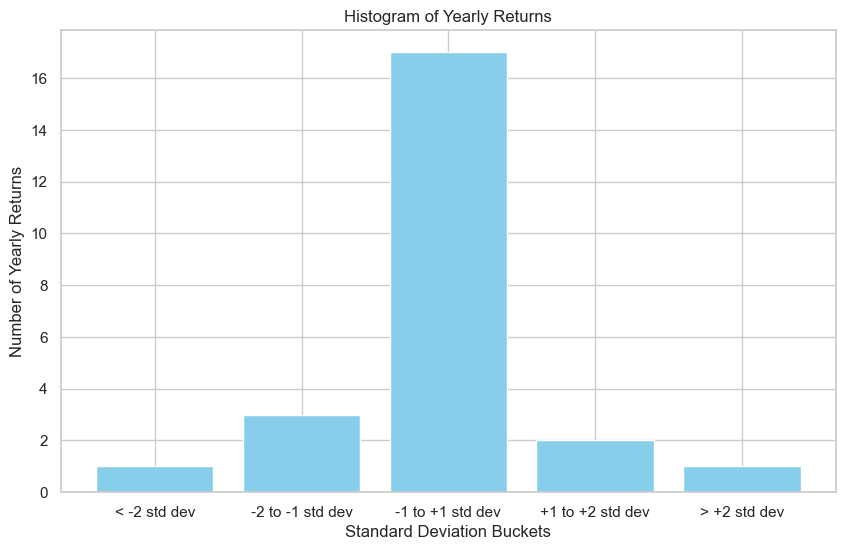
****

* **Inference**
* Heat map representing the monthly returns of the Nifty Fifty index over several years.
* X-Axis: Represents each month from January to December. Y-Axis: Represents different years.
* Shades of red indicate positive returns. Shades of blue indicate negative returns. The intensity of the colour reflects the magnitude of the return.
* In March 2020, there was a significant negative return (deep blue). In May 2009, there was a high positive return (deep red). Other months and years also exhibit varying returns.
* **Analysing histogram of monthly returns demonstrating the distribution of monthly returns in five or more buckets ( -1 to +1 std deviation, -1 to -2 std deviation, < -2 std deviation, +1 to +2 std deviation, > 2 std deviation)**

****

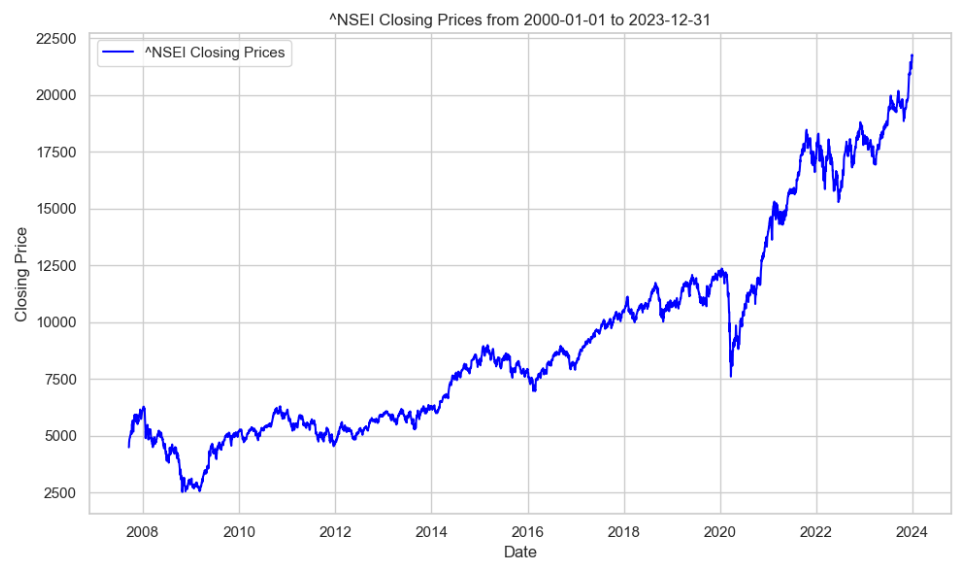
**Observation:**

* The histogram of monthly returns provides insights into the distribution of returns based on standard deviation.
* Standard Deviation Buckets:
* Most yearly returns fall within the range of -1 to +1 standard deviation from the mean.
* Fewer returns fall in the more extreme buckets (< -2 std dev or > +2 std dev).
* This histogram helps investors understand the volatility of an investment. When returns cluster around the mean, it suggests stability, while wider dispersion indicates greater risk.
* **Analysing a histogram of yearly returns demonstrating the distribution of yearly returns in five or more buckets ( -1 to +1 std deviation, -1 to -2 std deviation, < -2 std deviation, +1 to +2 std deviation, > 2 std deviation).**

****

**Observations:**

* The histogram of yearly returns provides insights into the distribution of returns based on standard deviation. Here are the key takeaways:
* Standard Deviation Buckets:
* Most yearly returns fall within the range of -1 to +1 standard deviation from the mean.
* Fewer returns fall in the more extreme buckets (< -2 std dev or > +2 std dev).
* This histogram helps investors understand the volatility of an investment. When returns cluster around the mean, it suggests stability, while wider dispersion indicates greater risk.
* **Analysis done by the Historical Data yfinance (Fetching Historical data from Yahoo Finance)**



**Inference:**

* The line graph titled “NSEI Closing Prices from 2000-01-01 to 2023-12-31” displays the closing prices of NSEI stock over the specified period.
* The NSEI closing price has increased significantly over the past 23 years. In 2000, the closing price was around 5,000. In 2023, the closing price was around 22,500. This represents an increase of over 400%.
* The graph appears to show a generally upward trend, with some periods of volatility. There have been a few years in which the closing price has declined from the previous year. For example, the closing price declined in 2008, 2011, and 2018.
* Overall, the graph suggests that the NSEI has grown significantly over the long term. However, there has also been some volatility in the market.

**5.BUILDING A PREDICTIVE MODEL**

During the development of a predictive model, various algorithms were trained and evaluated to ascertain their precision and accuracy. Given the constraints of a small dataset, comprising 23 rows and 13 columns, the selection of algorithms was tailored to suit the characteristics of the dataset. The given dataset was split into 80:20 ratio for the purpose of training and testing. This approach aimed to optimize model performance within the context of the available data, ensuring a robust and relevant predictive framework.­

**5.1 Linear Regression**

Linear regression models the relationship between a dependent variable (here, annual returns) and one or more independent variables (monthly returns) by fitting a linear equation to the observed data. In simpler terms, it helps understand how changes in monthly returns impact the annual returns. The model is trained using historical data, evaluated for accuracy, and then used for predicting future annual returns based on new monthly returns. It's a fundamental tool in data analysis and prediction tasks due to its simplicity and interpretability.



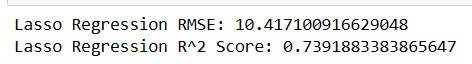
**5.2 Ridge Regression**

Ridge Regression is a type of linear regression that adds a penalty to the traditional least squares method. It's useful for handling multicollinearity in datasets where predictors are highly correlated. In the provided dataset, Ridge Regression could predict annual returns based on monthly returns. It's implemented easily using Python libraries like scikit-learn, and its regularization parameter helps find a balance between model complexity and performance.



**5.3 Lasso Regression**

Lasso Regression is a variant of linear regression that adds a penalty term to the model's objective function, encouraging simpler models by shrinking some coefficients to zero. This feature selection property makes it useful for datasets with many features, like the monthly returns data provided. It can help identify the most relevant monthly returns for predicting annual returns while disregarding less important ones.



**5.4 ElasticNet Regression**

ElasticNet regression balances Ridge and Lasso methods, aiding feature selection and multicollinearity handling. Trained on historical data, it predicts future outcomes based on new data, serving as a fundamental tool in data analysis and prediction tasks due to its adaptability and interpretability.

**5.5 Decision Tree Regression**

Decision tree regression is a non-linear regression technique that partitions data into subsets based on the value of independent variables. It recursively splits data to minimize variance in each subset, creating a tree-like structure. Each leaf node represents a predicted value. It's intuitive, easy to interpret, and handles non-linear relationships well. However, it's prone to overfitting complex datasets and lacks robustness compared to ensemble methods. Decision tree regression finds applications in finance, healthcare, and marketing, aiding in decision-making and predictive modeling tasks.



**5.6 Random Forest Regression**

Random Forest Regression is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and generalization. Each tree is trained on a random subset of data and features, reducing overfitting and increasing robustness. It handles non-linear relationships well, providing better performance than individual decision trees. Random Forest Regression is widely used in various fields such as finance, healthcare, and ecology for its ability to handle large datasets and complex relationships, making it a powerful tool for predictive modeling and decision support systems.



**5.7 Support Vector Regression (SVR)**

Support Vector Regression (SVR) is a supervised learning algorithm used for regression tasks. It identifies a hyperplane that best fits the data while maximizing the margin around it. SVR is effective for handling non-linear relationships through kernel functions. It's particularly useful for small datasets where it minimizes overfitting and generalizes well. However, SVR's performance heavily depends on proper parameter tuning. Despite this, it's widely employed in various fields, including finance, healthcare, and engineering, for its ability to model complex relationships and make accurate predictions, contributing significantly to decision-making and predictive analytics tasks.



**6. OBSERVATIONS:**

**1. RMSE Comparison:** Lasso Regression (RMSE: 10.42) and ElasticNet Regression (RMSE: 10.41) have the lowest RMSE values, indicating they provide the most accurate predictions among the models evaluated. Linear Regression (RMSE: 14.68) and Ridge Regression (RMSE: 14.37) also perform relatively well, with slightly higher RMSE values. Decision Tree Regression (RMSE: 40.51), Random Forest Regression (RMSE: 38.97), and SVR (RMSE: 41.70) have significantly higher RMSE values, suggesting poorer performance.

**2. R^2 Score Comparison:** Lasso Regression (R^2: 0.739) and ElasticNet Regression (R^2: 0.740) have the highest R^2 scores, indicating that they explain a substantial portion of the variance in the target variable. Ridge Regression (R^2: 0.504) also performs well in terms of explaining variance. On the other hand, Decision Tree Regression (R^2: -2.944), Random Forest Regression (R^2: -2.650), and SVR (R^2: -3.179) have negative R^2 scores, indicating poor model fit.

**3. Performance vs. Complexity**: Lasso Regression and ElasticNet Regression, despite being relatively simple models, outperform other models in terms of both RMSE and R^2 score. This suggests that they effectively capture the relationships in the data while avoiding overfitting. Decision Tree Regression, Random Forest Regression, and SVR, which are more complex models, perform poorly compared to simpler linear models, indicating potential overfitting or misfitting.

**Conclusion:**

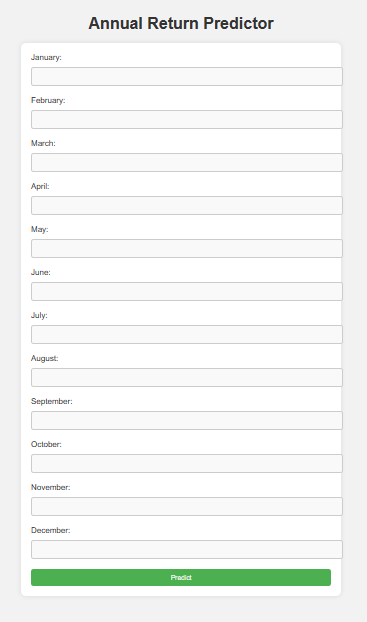
**1. Best Performing Models:** Among the evaluated models, Lasso Regression and ElasticNet Regression demonstrate the best performance in terms of both RMSE and R^2 score. These models effectively capture the relationships in the data and provide accurate predictions of annual returns based on monthly returns.

**2. Preferable Model Choice:** Given their superior performance and simplicity, Lasso Regression or ElasticNet Regression would be preferable choices for predicting annual returns in this dataset. These models offer a good balance between accuracy and interpretability.

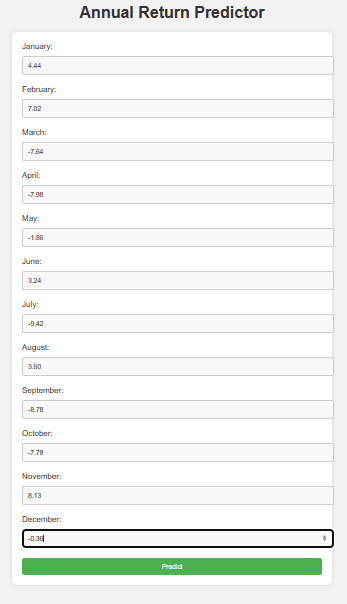
**3. Caution with Complex Models:** Decision Tree Regression, Random Forest Regression, and SVR perform poorly compared to simpler linear models, indicating that they may not be suitable for this dataset. Care should be taken when using complex models, as they may lead to overfitting and unreliable predictions, especially with small datasets like the one provided.

**7. Final Implementation:**

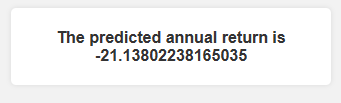
For the final implementation, a web application is developed using Flask showcasing the implementation of inference modeling in Case of Nifty. The monthly returns from January to December are the inputs for the web application, the result predicted by the model is shown as the output. The output is represented in the terms of annual return of the year whose monthly returns were given as the input. We used ElasticNet regression model for the annual return predictions.

****

* **Taking monthly return as input for the annual return predictor.**

****

* **Clicking on the predict button we get the following output:**

****

**Conclusion:**

The analysis of the stock market data spanning 20 years, with a focus on the year 2020, revealed several key insights. The dataset provided a comprehensive view of monthly and annual returns for the Nifty index. During the COVID-19 pandemic in 2020, significant fluctuations were observed, with some months experiencing sharp declines while others showed recovery and growth. The data analysis included exploratory data analysis (EDA) to understand the distribution of returns, and identify any interesting patterns.

Overall, the analysis provides valuable insights for investors, researchers, and businesses looking to understand the dynamics of the stock market, particularly during challenging periods like the COVID-19 pandemic. Further research and modeling could be explored to forecast future market trends and optimize investment strategies based on historical data patterns.