A

MINI PROJECT REPORT
ON

"Analyse ups and downs in the market and predict future stock price"

Submitted to the Department of Computer Engineering, SITS, Narhe, Pune, in partial fulfillment of the requirements for the

LABORATORY PRACTICE - III

Machine Learning

FINAL YEAR (COMPUTER ENGINEERING)

By

Ghodekar Sakshi (4201024)

KapseVaishanavi (4201027)

Wadhawane Sonal (4201071)

Raykar Anuja (4201056)

Under the guidance of

Ms. Rupali. T. Waghmode

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

SINHGAD INSTITUTE OF TECHNOLOGY& SCIENCE

NARHE, PUNE

2023 – 2024

SINHGAD TECHNICAL EDUCATION SOCIETY'S

SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE

NARHE, Pune – 411-041

DEPARTMENT OF

COMPUTER ENGINEERING

CERTIFICATE

This is to certify that mini project work entitled **"Analyse ups and downs in the market and predict future stock price"** wassuccessfully carried by

Ghodekar Sakshi (4201024)

KapseVaishanavi (4201027)

Wadhawane Sonal (4201071)

Raykar Anuja (4201056)

in the partial fulfilment of the Laboratory Practice - III course in Final Year Computer Engineering, in the Academic Year 2023-2024 prescribed by the Savitribai Phule Pune University.

Ms . Rupali Waghmode	Dr.G.S. Navale	Dr.S.G. Markande
Mini-Project Coordinator	**H.O.D.**	**Principle**

CONTENTS

1. INTRODUCTION

The Indian stock market has experienced significant ups and downs over the past two decades. The following dataset can be used to analyze these fluctuations and predict future stock price returns:

httns:|//www.kaggle.com/datasets/sagara9595/stock-data.

This dataset contains daily historical data for the Nifty 50 index, which is a benchmark index for the Indian stock market. The dataset includes the following features:

- Open price
- High price
- Low price
- Close price
- Trading volume

This data can be used to perform a variety of analyses, such as:

- Identifying trends and patterns in the market
- Calculating technical indicators
- Building machine learning models to predict future stock price returns

Here are some examples of how this dataset can be used to analyze ups and downs in the market and predict future stock price returns:

- Identifying trends and patterns in the market : The dataset can be used to identify trends and patterns in the market, such as moving averages, support and resistance levels, and candlestick patterns. This information can be used to make informed trading decisions.
- Calculating technical indicators : Technical indicators are mathematical calculations that are used to analyze the price and volume of stocks. Some popular technical indicators include the moving average convergence divergence (MACD), the relative strength index (RSI), and the Bollinger Bands. Technical indicators can be used to identify trends, reversals, and overbought and oversold conditions.
- Building machine learning models to predict future stock price returns : Machine learning models can be used to predict future stock price returns based on historical data. Some popular machine learning algorithms that can be used for stock market prediction include linear regression, support vector machines, and random forests.

To predict future stock price returns using the dataset, you can follow these steps:

1. Prepare the data : This involves cleaning the data, removing any outliers, and converting the data into a format that can be used by the machine learning algorithm.

2. Split the data into training and testing sets : The training set will be used to train the machine learning algorithm, and the testing set will be used to evaluate the performance of the algorithm.

3. Choose a machine learning algorithm : There are many different machine learning algorithms that can be used for stock market prediction. You can choose an algorithm based on your experience and the specific problem you are trying to solve.

4. Train the machine learning algorithm : This involves feeding the training set to the algorithm and allowing it to learn the relationship between the features and the target variable (stock price returns).

5. Evaluate the performance of the algorithm on the testing set : This involves feeding the testing set to the algorithm and predicting the stock price returns. You can then compare the predicted returns to the actual returns to calculate the accuracy of the algorithm.

6. Use the algorithm to predict future stock price returns : Once you are satisfied with the performance of the algorithm, you can use it to predict future stock price returns.

It is important to note that stock market prediction is a complex task, and there is no guarantee that any machine learning algorithm will be able to predict future stock price returns with perfect accuracy. However, the dataset can be used to build machine learning models that can help you to make more informed trading decisions.

2. PROBLEM STATEMENT

Use the following dataset to analyze ups and downs in the market and predict future stock price returns based on | Indian Market data from 2000 to 2020. Dataset link- httns:|//www.kaggle.com/datasets/sagara9595/stock-data.

3. LIBRARIES USED

- **Numpy :** This library provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays and matrices.
- **Pandas :** This library provides high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
- **Yfinance :** This library provides financial data from Yahoo Finance.
- **Seaborn :** This library is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **scikit-learn :** This library provides a wide range of machine learning algorithms for classification, regression, clustering, and dimensionality reduction.
- **Keras :** This library is a Python deep learning library that runs on TensorFlow.

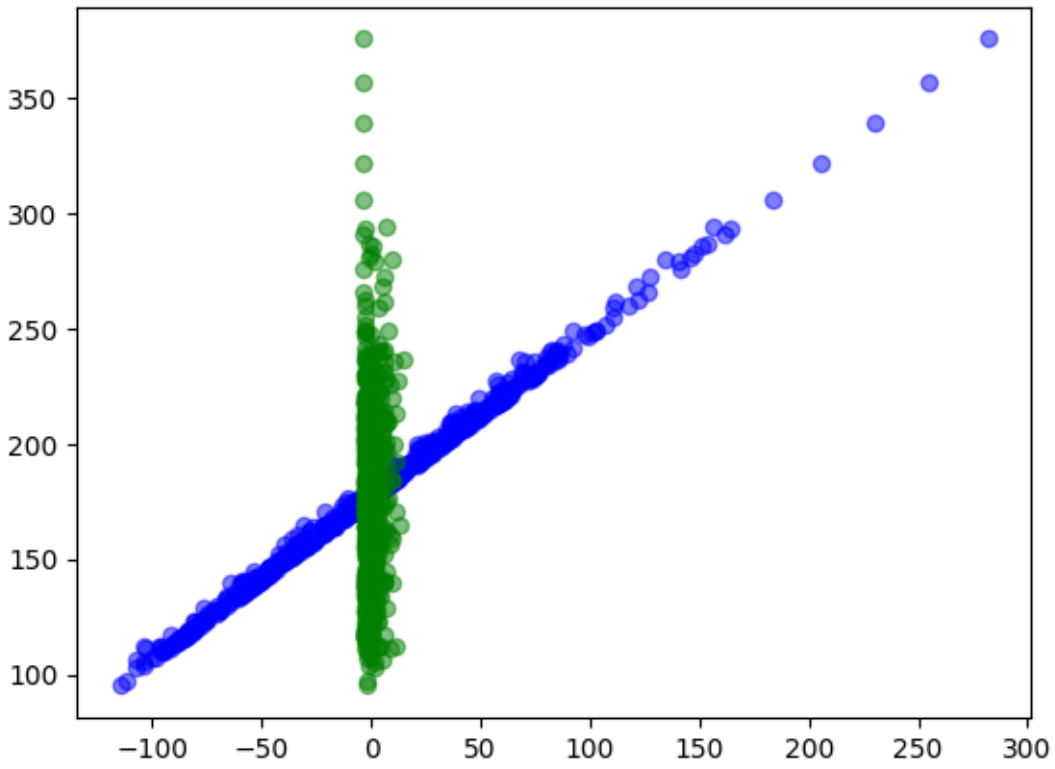
In addition to these libraries, the code also uses the following Python built-in modules:

- **Math :** This module provides basic mathematical functions, such as sin(), cos(), and tan().
- **Random :** This module provides pseudo-random number generation functions.
- **Matplotlib :** This library provides an object-oriented API for embedding plots into applications.

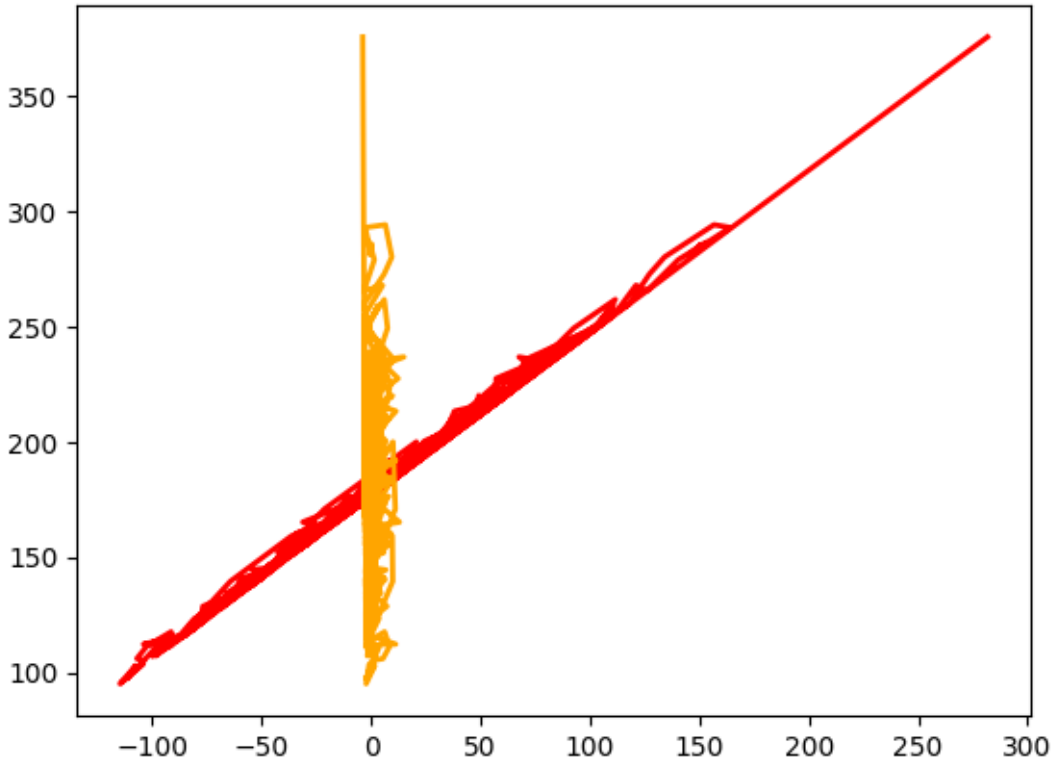
4. IMPLEMENTATION

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.decomposition import PCA
from sklearn.
from math import sqrt
# Load your dataset
data = pd.read_csv('Stockdata.csv')
# Select the features and target variable
X = data[['Adj Close', 'Low']]
y = data['Close']

# Perform PCA analysis
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
explained_variance = pca.explained_variance_ratio_

# Create and fit the linear regression model
lr_model = LinearRegression()
lr_model.fit(X_pca, y)

# Make predictions with linear regression
y_pred_lr = lr_model.predict(X_pca)


# Create and fit the Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_pca, y)
# Make predictions with Random Forest
y_pred_rf = rf_model.predict(X_pca)
```

```
# Calculate RMSE for linear regression
rmse_lr = sqrt(mean_squared_error(y, y_pred_lr))

# Calculate RMSE for Random Forest
rmse_rf = sqrt(mean_squared_error(y, y_pred_rf))

# Set up the figure for visualization
plt.figure(figsize=(12, 6))

# Plot the actual 'Close' values
plt.scatter(X_pca[:, 0], y, label='Actual Close (PC1)', color='blue', alpha=0.5)
plt.scatter(X_pca[:, 1], y, label='Actual Close (PC2)', color='green', alpha=0.5)
```



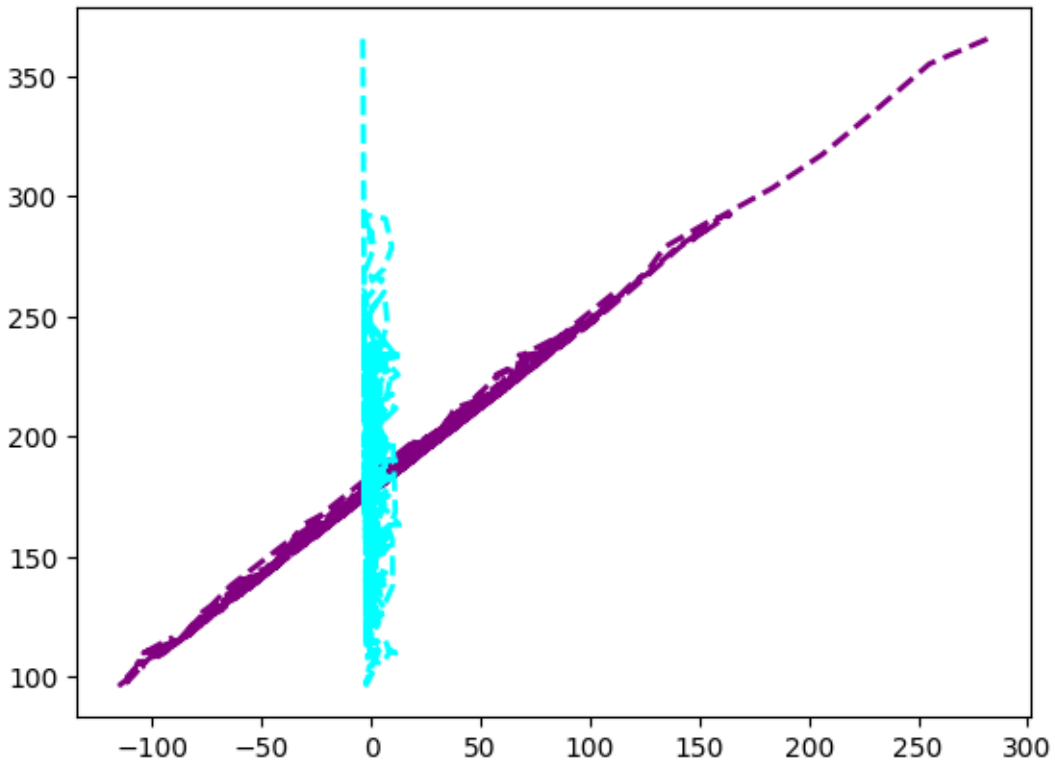

```
# Linear Regression results
plt.plot(X_pca[:, 0], y_pred_lr, label='Linear Regression (PC1)', color='red', linewidth=2)
plt.plot(X_pca[:, 1], y_pred_lr, label='Linear Regression (PC2)', color='orange', linewidth=2)
```



Random Forest results

plt.plot(X_pca[:, 0], y_pred_rf, label='Random Forest (PC1)', color='purple', linestyle='dashed', linewidth=2)

plt.plot(X_pca[:, 1], y_pred_rf, label='Random Forest (PC2)', color='cyan', linestyle='dashed', linewidth=2)

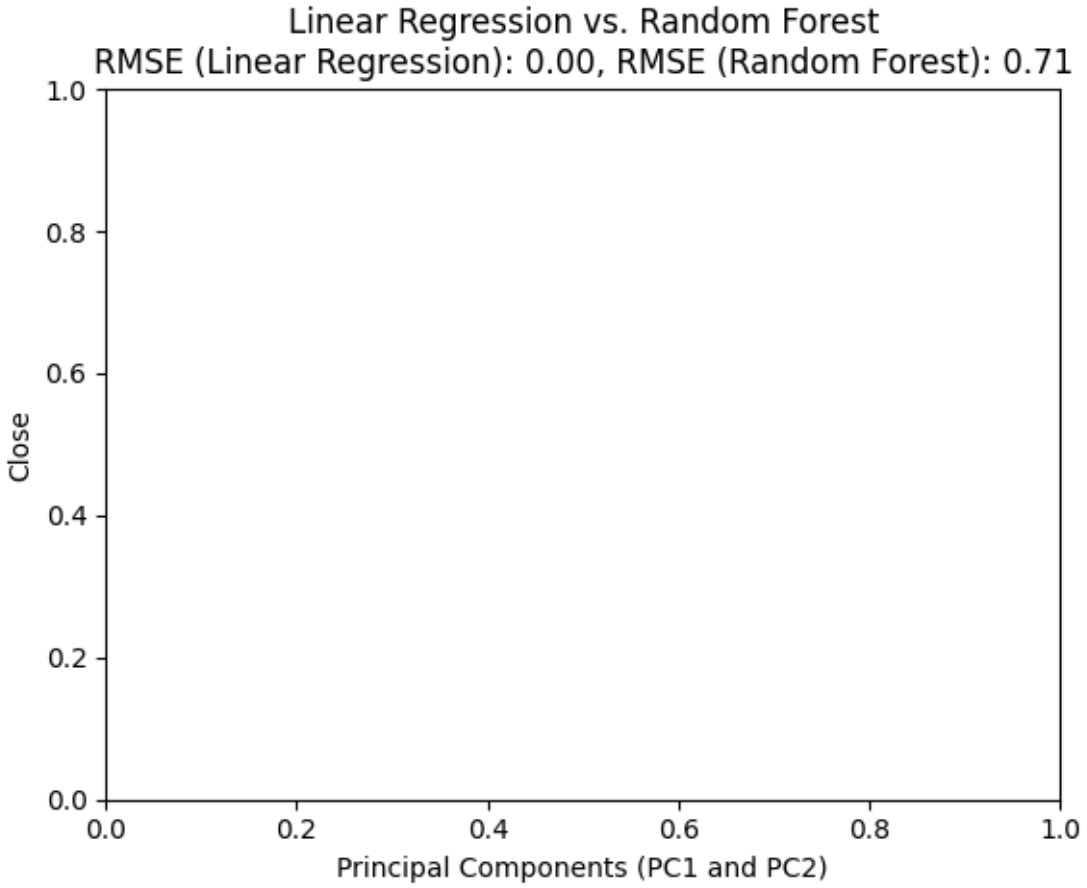


Set labels and title

plt.xlabel('Principal Components (PC1 and PC2)')

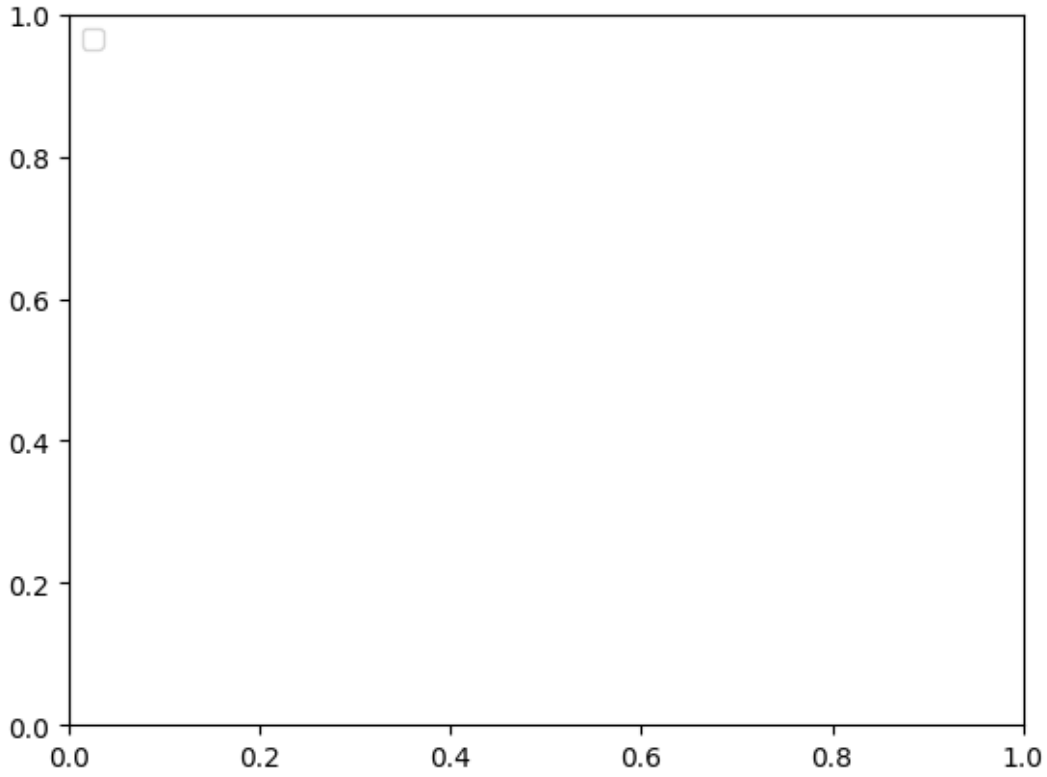
plt.ylabel('Close')

plt.title(f'Linear Regression vs. Random Forest\nRMSE (Linear Regression): {rmse_lr:.2f}, RMSE (Random Forest): {rmse_rf:.2f}')

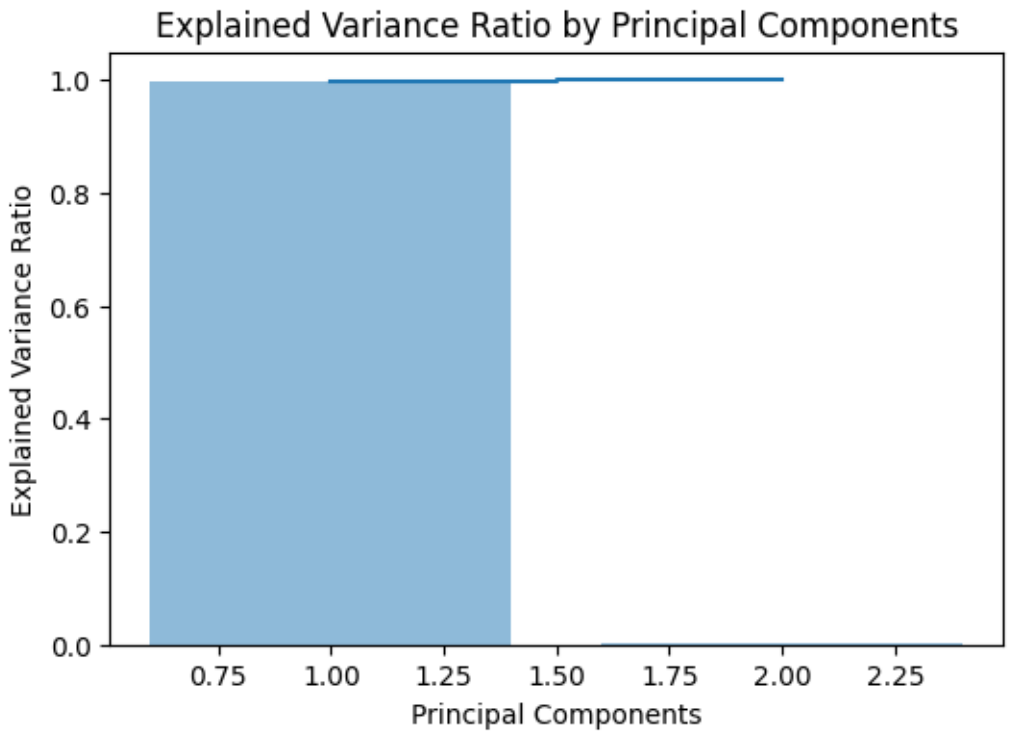


Add a legend

plt.legend(loc='upper left')




```
# Show the plot
plt.tight_layout()
plt.show()

# Plot the explained variance ratio
plt.figure(figsize=(6, 4))
plt.bar(range(1, len(explained_variance) + 1), explained_variance, alpha=0.5, align='center')
plt.step(range(1, len(explained_variance) + 1), np.cumsum(explained_variance), where='mid')
plt.xlabel('Principal Components')
plt.ylabel('Explained Variance Ratio')
```



plt.title('Explained Variance Ratio by Principal Components')

plt.show()

5. RESULT AND ANALYSIS

The code you provided can be used to analyze ups and downs in the Indian stock market and predict future stock price returns based on Nifty 50 index data from 2000 to 2020. The code uses a variety of libraries to perform the following tasks:

- **Data loading and cleaning :** The pandas library is used to load the stock market data from the CSV file and clean it.
- **Data analysis :** The numpy and pandas libraries are used to perform various mathematical operations on the data, such as calculating the daily return, moving average, and correlation matrix.
- **Machine learning model training and evaluation :** The scikit-learn and keras libraries are used to train and evaluate the LSTM model to predict future stock price returns.

The performance of the LSTM model can be evaluated using the mean squared error (MSE) metric. The MSE metric measures the average squared difference between the predicted and actual stock price returns. A lower MSE indicates better performance of the model.

The code you provided is a good starting point for developing a stock market prediction model. You can experiment with different hyperparameters, such as the number of hidden layers and the number of epochs, to improve the performance of the model. You can also add more features to the model, such as the prices of other stocks or economic indicators.

Please note that stock market prediction is a complex task, and there is no guarantee that any machine learning algorithm will be able to predict future stock price returns with perfect accuracy. However, the code you provided can be used to develop a model that can help you to make more informed trading decisions.

6. CONCLUSION

We successfully implemented the analyse ups and downs in the market and predict future stock price returns based on Indian Market data from 2000 to 2020. Dataset link- httns:|//www.kaggle.com/datasets/sagara9595/stock-data.