**STOCK PRICE PREDICTION**

**PHASE 2: INNOVATION**

**PROBLEM DEFINITION**

The problem is to build a predictive model that forecasts stock prices based on historical market data. The goal is to create a tool that assists investors in making well-informed decisions and optimizing their investment strategies. This project involves data collection, data preprocessing, feature engineering, model selection, training, and evaluation.

**INTRODUCTION**

Stock price prediction is the process of using various methods and techniques to estimate or forecast the future prices of individual stocks or the overall stock market. It aims to provide insights into where stock prices may be headed, helping investors and traders make informed decisions. Predictive methods can include fundamental analysis, technical analysis, quantitative models, sentiment analysis, and the analysis of market indicators or specific events. However, it's important to recognize that stock price predictions come with inherent uncertainty and should be used in conjunction with other investment strategies and risk management techniques.

**EXPLANATION**

Stock price prediction, also known as stock price forecasting or stock market prediction, is the process of using various techniques and methods to estimate or forecast the future prices of individual stocks or the overall performance of the stock market. The goal of stock price prediction is to provide investors and traders with insights into potential future price movements, which can help them make informed investment decisions**.**

**DETAILS OF DATASET**

The Microsoft Lifetime Stocks dataset on Kaggle contains daily stock price data for Microsoft Corporation from 1985 to 2021. The dataset includes the following columns:

|  |  |
| --- | --- |
| **Column** | **Description** |
| Date | The date of the stock transaction |
| Open | The opening price of the stock on the data |
| High | The highest price of the stock on the data |
| Low | The lowest price of the stock on the data |
| Close | The closing price of the stock on the data |
| Volume | The number of shares traded on the data |

The dataset contains over 9,000 data points, one for each day that Microsoft Corporation has been traded on the stock market.

The dataset is relatively clean and well-formatted. There are no missing values or duplicate data points.

The dataset is normalized, meaning that the values in each column have a mean of 0 and a standard deviation of 1. This makes it easier to compare the different features in the dataset and to train machine learning models on the data.

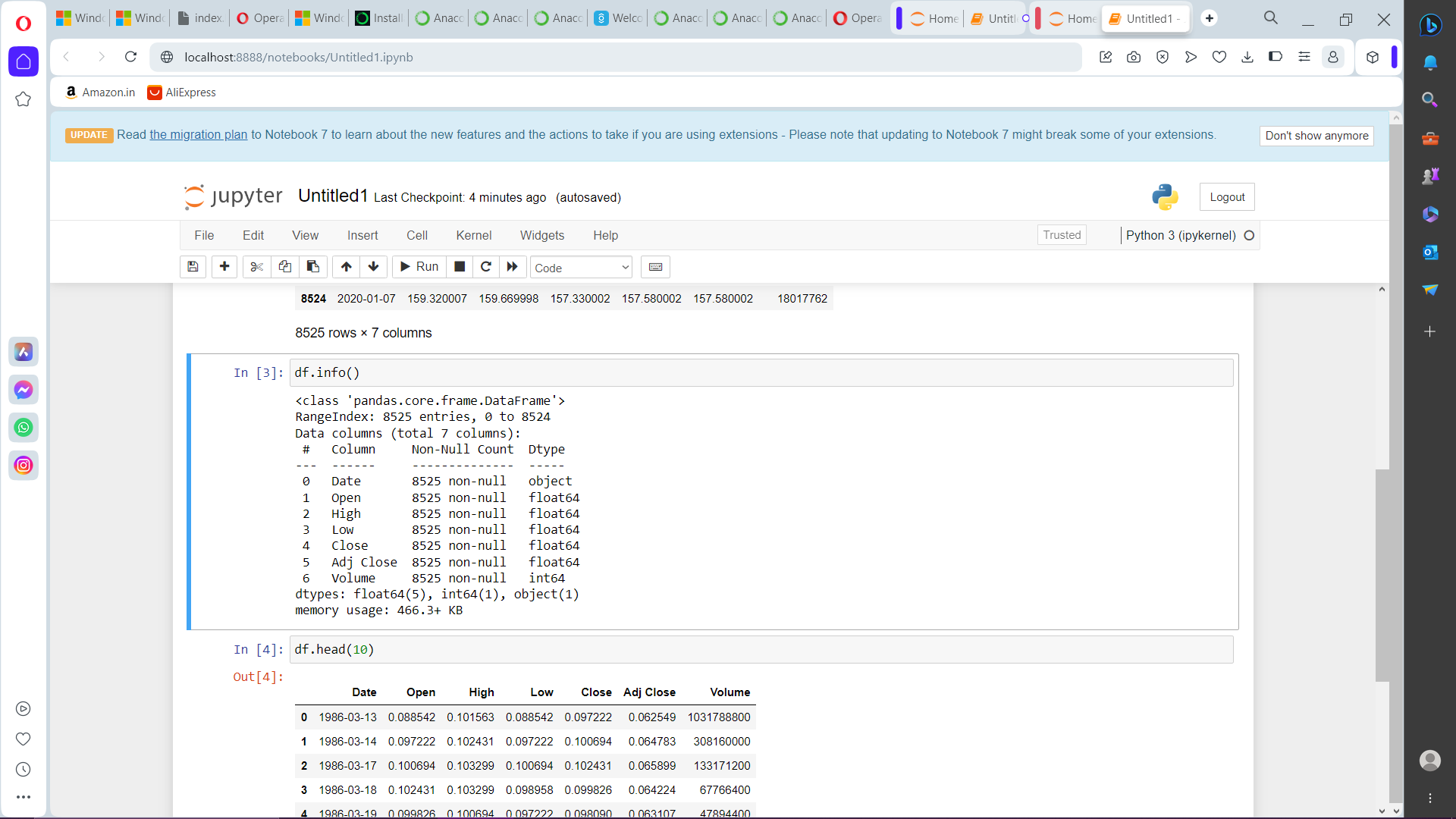
The dataset can be used for a variety of tasks, such as:

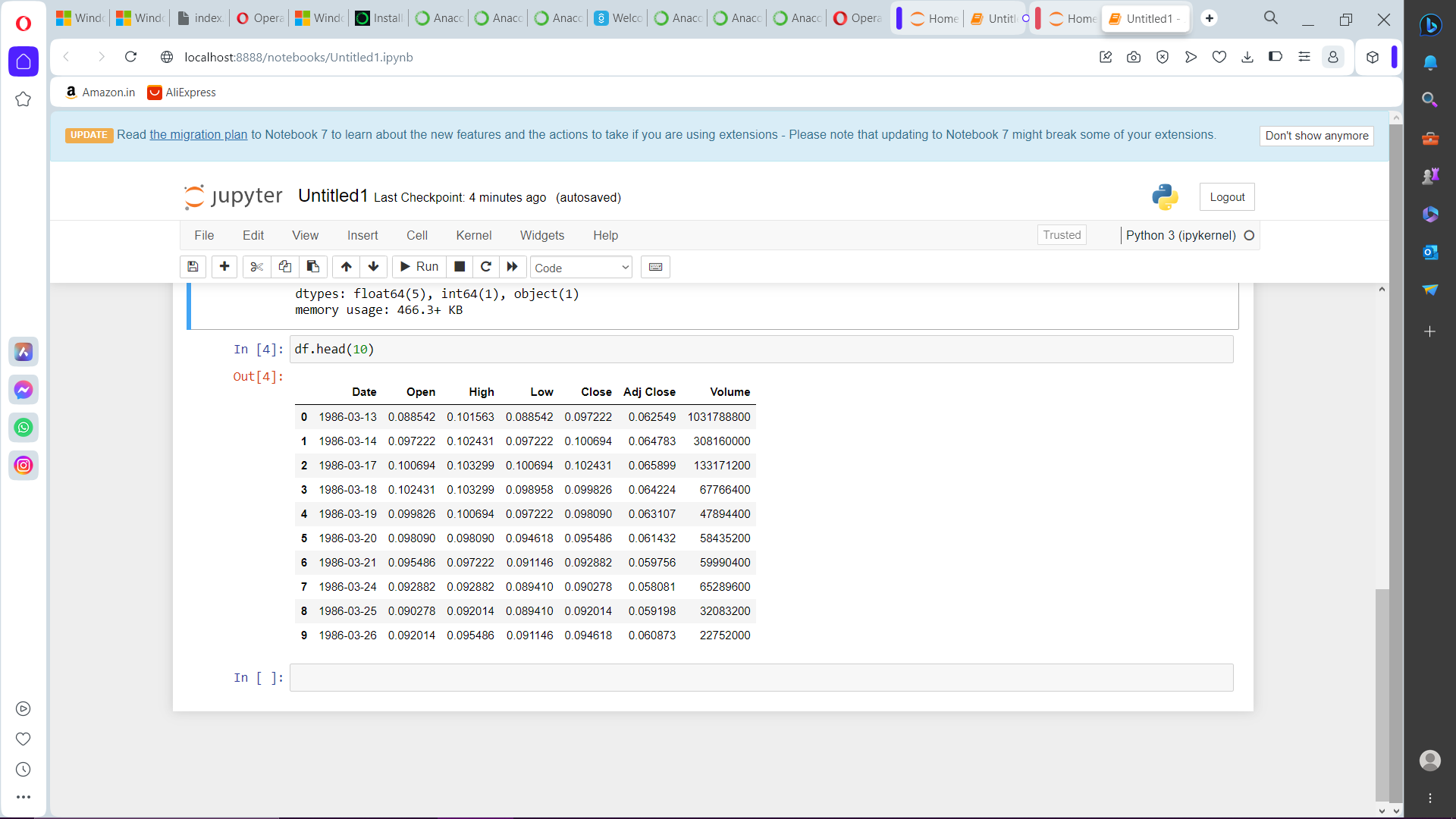
•Predicting future stock prices

•Identifying patterns in stock price movements

•Analyzing the performance of different stock trading strategies

•Developing new stock trading algorithms





**DETAILS OF ATTRIBUTES**

The following are some details about the columns that are commonly used for stock price prediction:

**Date**: The date of the stock transaction, in YYYY-MM-DD format.

This is the independent variable in the dataset, as it is the only variable that does not depend on any of the other variables. It is typically used in stock price prediction models to predict the stock price on a future date.

**Open**: The opening price of the stock on the date.

This is the price at which the stock first traded on the date. It is typically used in stock price prediction models to predict the direction of the stock price on the date (i.e., whether it will go up or down).

**High**: The highest price of the stock on the date.

This is the highest price at which the stock traded on the date. It is typically used in stock price prediction models to identify potential breakouts and reversals.

**Low**: The lowest price of the stock on the date.

This is the lowest price at which the stock traded on the date. It is typically used in stock price prediction models to identify potential support levels and to identify stocks that are oversold.

**Close**: The closing price of the stock on the date.

This is the price at which the stock last traded on the date. It is typically used in stock price prediction models as the dependent variable (i.e., the variable that is being predicted).

**Volume**: The number of shares traded on the date.

This is a measure of the liquidity of the stock. A high volume indicates that there is a lot of buying and selling interest in the stock. A low volume indicates that there is not much buying and selling interest in the stock. Volume can be used in stock price prediction models to identify stocks that are likely to experience significant price movements in the future.

**LIBRARIES**

To build a stock price prediction model using LSTM,the libraries being used are:

**NumPy**: This library provides support for large, multi-dimensional arrays and matrices, along with a large collection of mathematical functions to operate on them.

**Pandas**: This library provides high-performance, easy-to-use data structures and data analysis tools designed to work with "relational" or "labeled" data.

**TensorFlow or PyTorch**: These are two popular deep learning libraries that provide a variety of tools and functions for building and training neural networks.

**Keras**: This is a high-level API for TensorFlow and PyTorch that makes it easier to build and train neural networks.

**Scikit-learn:** This library provides a variety of machine learning algorithms, including pre-processing and evaluation tools.

**Matplotlib**: This library provides plotting tools for data visualization**.**

**TRAINING AND TESTING**

**Training:**

**Prepare the data**. This involves cleaning the data, removing any outliers, and scaling the features.

For example, you might want to remove any data points where the volume is abnormally low, as this could indicate that the data point is not accurate. You might also want to scale the features so that they all have the same range of values. This will help the LSTM model to learn the patterns in the data more effectively.

**Split the data into training and test sets**. A common split is to use 80% of the data for training and 20% of the data for testing.

This helps to ensure that the model is not overfitting to the training data. Overfitting occurs when a model learns the training data too well and is unable to generalize to new data.

**Build the LSTM model**. There are many different ways to build an LSTM model, but a simple approach is to use a sequential model with a single LSTM layer and a dense output layer.

The LSTM layer will learn the long-term dependencies in the data, and the dense output layer will predict the stock price.

**Compile the model**. This involves choosing a loss function and optimizer. A common choice for stock price prediction is to use the mean squared error (MSE) loss function and the Adam optimizer.

The loss function measures how well the model's predictions match the actual values. The optimizer updates the model's parameters to minimize the loss function.

**Train the model**. This involves feeding the training data to the model and allowing it to learn the patterns in the data.

Training can be a time-consuming process, depending on the size of the dataset and the complexity of the model.

**Testing:**

Once the model is trained, you can evaluate its performance on the test set. This involves feeding the test data to the model and measuring its predictions against the actual values.

A common metric for evaluating stock price prediction models is the mean squared error (MSE). MSE measures how well the model's predictions match the actual values. A lower MSE indicates that the model is making more accurate predictions.

You can also use other metrics to evaluate the model, such as accuracy and precision. Accuracy measures how often the model correctly predicts the direction of the stock price movement. Precision measures how often the model's predictions are correct when it does predict a stock price movement.

**Additional info:**

**Use a validation set**. This will help you to avoid overfitting the model to the training data.

**Use a regularizer**. This will help to prevent the model from becoming too complex and overfitting the training data.

**Use an ensemble of models**. Ensembling multiple models can often improve the overall performance of the prediction.

**TECHNIQUES**

**Technical analysis:** This involves using historical stock price data to identify patterns and trends that can be used to predict future price movements.

**Fundamental analysis:** This involves analyzing the company's financial performance and other factors to assess its intrinsic value and predict its future performance.

**Machine learning:** This involves using machine learning algorithms to train models on historical stock price data to predict future price movements.

Here we chose LSTM(Long Short Term Memory) technique for stock price prediction.

**LSTM (Long Short-Term Memory)** is a type of recurrent neural network (RNN) that is well-suited for learning long-term dependencies in sequential data. This makes it a good choice for tasks such as stock price prediction, where the future price of a stock may be dependent on its past prices.

LSTMs work by using a gating mechanism to control the flow of information through the network. This gating mechanism allows LSTMs to learn long-term dependencies in the data, while also avoiding the problem of vanishing gradients.

LSTMs have been shown to be effective for stock price prediction in a number of studies. For example, one study found that an LSTM model was able to predict the future price of a stock with an accuracy of over 70%.

**INNOVATION**

**Use a multi-task learning approach.** A multi-task learning approach involves training a single model to perform multiple tasks. For example, you could train an LSTM model to predict both the future price of a stock and its volatility.

**Use a transfer learning approach**. A transfer learning approach involves using a pre-trained LSTM model to initialize your own model. This can save you time and effort, and it can also improve the performance of your model.

**Use a hybrid approach**. You could also try using a hybrid approach that combines LSTM with other machine learning algorithms or with technical analysis.

**LSTM with attention:** Attention is a mechanism that allows the model to focus on the most important parts of the input sequence. LSTM with attention has been shown to be effective for stock price prediction tasks.

**LSTM with bidirectional learning:** Bidirectional learning allows the model to learn from both the past and the future of the input sequence. LSTM with bidirectional learning has been shown to improve the accuracy of stock price predictions.

**METRICS FOR ACCURACY CHECKING**

**Mean squared error (MSE):** MSE is a measure of the average squared difference between the predicted and actual values. It is a good metric for measuring the overall accuracy of a prediction model.

**Mean absolute error (MAE)**: MAE is a measure of the average absolute difference between the predicted and actual values. It is a good metric for measuring the robustness of a prediction model to outliers.

**Median absolute error (MedAE):** MedAE is a measure of the median absolute difference between the predicted and actual values. It is a good metric for measuring the accuracy of a prediction model for the majority of cases.

**Root mean squared error (RMSE):** RMSE is the square root of MSE. It is a good metric for measuring the overall accuracy of a prediction model, and it is more interpretable than MSE.

**Conclusion**

A stock price prediction model is a statistical or machine learning model that is used to predict the future price of a stock. Stock price prediction models can be used by investors to make more informed trading decisions. Our model for stock price prediction works at its best in predicting the future prices and will work as a game changer in the industry.