**Project: Predicting Stock Open Price using LSTM**

**Abstract:**

The project aims to develop a robust predictive model for forecasting stock open prices using Long Short-Term Memory (LSTM) networks. Leveraging historical time series data obtained from Yahoo Finance in CSV format, the model employs a combination of past stock open prices and volume as input features to predict the open price for the subsequent trading day. Implementation is carried out using TensorFlow and Keras, with careful consideration given to data preprocessing, model design, hyperparameter selection, and performance evaluation.

**Domain Introduction:**

In the dynamic and volatile domain of stock market forecasting, accurately predicting stock prices is essential for investors, traders, and financial analysts to make informed decisions. Such time-series data can be computationally analyzed with an aid of financial literacy to derive hidden market patterns and to predict potential future market trends.

Traditional time series analysis methods, combined with advancements in deep learning techniques, offer promising avenues for capturing the intricate patterns and trends present in financial data. LSTM networks, known for their ability to model long-range dependencies in sequential data, are particularly well-suited for forecasting stock prices.

**Dataset Description:**

The dataset comprises historical time series data (of recent 4 years) of stock open prices and volume on a daily basis, sourced from Yahoo Finance. Each data point includes the open price and volume for a specific stock on a given trading day. The dataset is pre-processed to handle missing values, normalize the features, and create input sequences suitable for training the LSTM model. Additionally, data splitting techniques are employed to separate training, validation, and test datasets to ensure robust model evaluation.

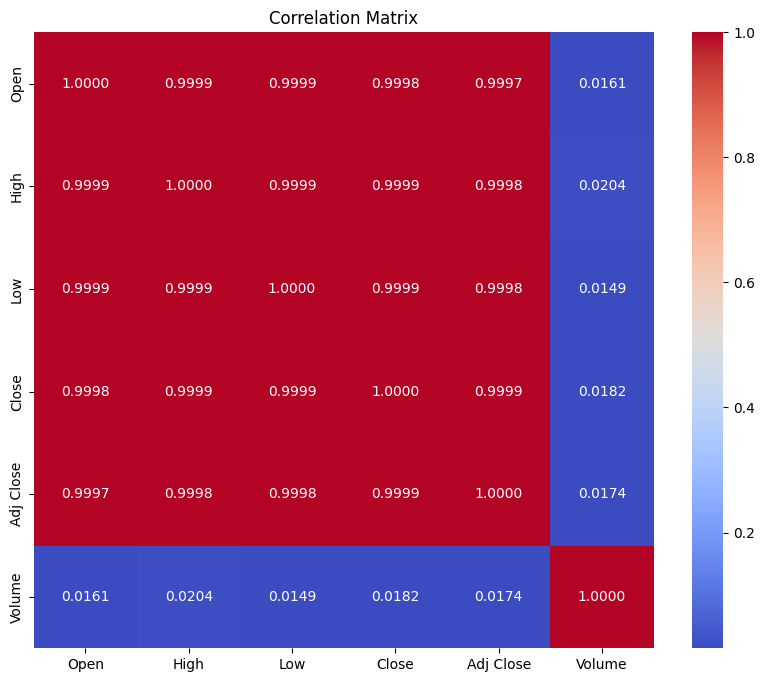
Data pre-processing:

1. Feature-extraction:

After conducting a correlation analysis on the features, we observed that the features "open", "close", "high", "low" and "adj close" exhibit a strong correlation with each other, with correlation coefficients close to 1. This implies that these features are highly dependent on each other and may provide redundant information for prediction tasks. Conversely, the feature "volume" showed no significant correlation with the other features.

Given that our objective is to predict the open price for the next day, we decided to select "open" and "volume" as the features for our LSTM model based on the correlation analysis results. While "open" provides crucial information about the starting price of the trading day, "volume" represents the trading activity and liquidity in the market, which may influence price movements.

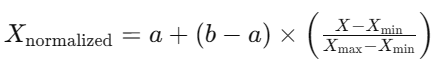
By choosing "open" and "volume" as the input features for the LSTM model, we aim to leverage their potential predictive power while avoiding redundancy in the feature space. This selection strategy ensures that the model focuses on relevant information for accurate open price prediction while disregarding highly correlated features that may introduce noise or redundancy into the model.



1. Normalization of the input data:

In our project, we employed a normalization technique to scale the "Open" and "Volume" columns of the stock price dataset. This normalization method, commonly known as **Min-Max scaling or feature scaling**, is a widely used preprocessing technique in machine learning and data analysis.

The normalization process involves transforming the data such that it falls within a specific range, typically between 0 and 1. This is achieved by applying the following formula:



where:

* *X* represents the original data values.
* 𝑋min​ and 𝑋max​ are the minimum and maximum values of the data, respectively.
* 𝑎and 𝑏denote the desired range for the normalized data, typically 0 and 1, respectively.

By applying this normalization technique, we ensure that the features "Open" and "Volume" are transformed to a common scale, facilitating more effective training of machine learning models. This normalization process helps prevent features with larger scales from dominating the learning process and ensures that each feature contributes proportionally to the model's predictions.

1. Handling the missing values and creating input sequences suitable for training.

Dropped the missing values and created input sequences suitable for training.

**Implementation Methodology with Brief Justification:**

The implementation methodology encompasses several key steps:

1. Data Pre-processing: Raw data undergoes rigorous pre-processing, including handling missing values, normalization, and sequence generation. Special attention is given to maintaining the temporal ordering of the data to preserve its sequential nature.
2. Model Design: An LSTM-based neural network architecture is designed using TensorFlow and Keras. The model architecture consists of LSTM layers followed by dense layers with LeakyReLU activation functions. This design choice aims to capture complex temporal dependencies and patterns present in stock market data.
3. Hyperparameter Selection: The Adam optimizer is chosen for its adaptive learning rate capabilities and robustness to hyperparameter choices. A careful selection of hyperparameters, including learning rate and activation functions, is made to optimize model performance.
4. Model Compilation and Training: The LSTM model is compiled with the chosen optimizer and loss function (Mean Squared Error), and then trained using historical stock data. Model training is conducted over multiple epochs, with Dropout Layer and model checkpointing techniques employed to prevent overfitting and save the best-performing model.
5. Performance Evaluation: The trained model's performance is evaluated using standard evaluation metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). Additionally, visualizations such as loss curves and prediction plots are generated to gain insights into the model's behavior and predictive capabilities.

**Results:**

A graph showing a line of growth

Description automatically generated with medium confidence

Test Data:

Mean Squared Error (MSE): 0.0003969882

Mean Absolute Error (MAE): 0.015484679

R-squared (R2): 0.8356645725295121

Validation Data:

Mean Squared Error (MSE): 9.0114445e-05

Mean Absolute Error (MAE): 0.007428201

R-squared (R2): 0.9651531296039377

The LSTM model trained using the Adam optimizer demonstrates promising results in predicting stock open prices. The model achieves low MSE and MAE values on both the test and validation datasets, indicating its ability to accurately forecast stock prices. Furthermore, visualizations of the model's predictions against true values reveal patterns and trends captured by the model, providing valuable insights into the stock market dynamics.

The project demonstrates the feasibility and effectiveness of using LSTM networks for stock open price prediction. Through meticulous data preprocessing, model design, and hyperparameter tuning, the LSTM model achieves promising results in forecasting stock prices.

**Future Scope:**

The project offers numerous avenues for future enhancement and exploration:

1. Feature Engineering: Incorporating additional features such as technical indicators, market sentiment data, and economic indicators to improve prediction accuracy and robustness.
2. Model Architecture Optimization: Experimenting with different LSTM architectures, including variations such as stacked LSTMs, bidirectional LSTMs, and attention mechanisms, to capture more complex temporal relationships.
3. Hyperparameter Tuning: Conducting extensive hyperparameter tuning experiments to optimize model performance further, including learning rate schedules, regularization techniques, and optimizer variants.
4. Ensemble Learning: Exploring ensemble learning techniques such as model averaging and boosting to combine multiple LSTM models for enhanced prediction accuracy and model robustness.
5. Deployment and Monitoring: Deploying the trained model in real-world trading environments and monitoring its performance over time to assess its effectiveness in practical applications.