**SentiNova**

**Overview**

SentiNova is a Long Short-Term Memory (LSTM) recurrent neural network model designed for stock price prediction, incorporating sentiment analysis to enhance accuracy. The model leverages historical stock prices and sentiment scores derived from news headlines to forecast future closing prices. It is implemented using Keras with a TensorFlow backend. The architecture allows the model to capture temporal dependencies in stock prices and incorporate the impact of public sentiment on market behavior.

**Model Architecture**

SentiNova employs a sequential LSTM network with the following layers:

**Input Layer:**

Accepts sequences of historical stock prices and sentiment scores. The input shape is decided by the two features: closing price and sentiment score.

**LSTM Layers:**

The model consists of four LSTM layers, each with 50 units. The first three LSTM layers have return\_sequences=True to feed the output sequence to the subsequent LSTM layer. The final LSTM layer has return\_sequences=False as it is followed by a dense layer. LSTM layers are chosen for their ability to effectively capture temporal dependencies and handle time-series data.

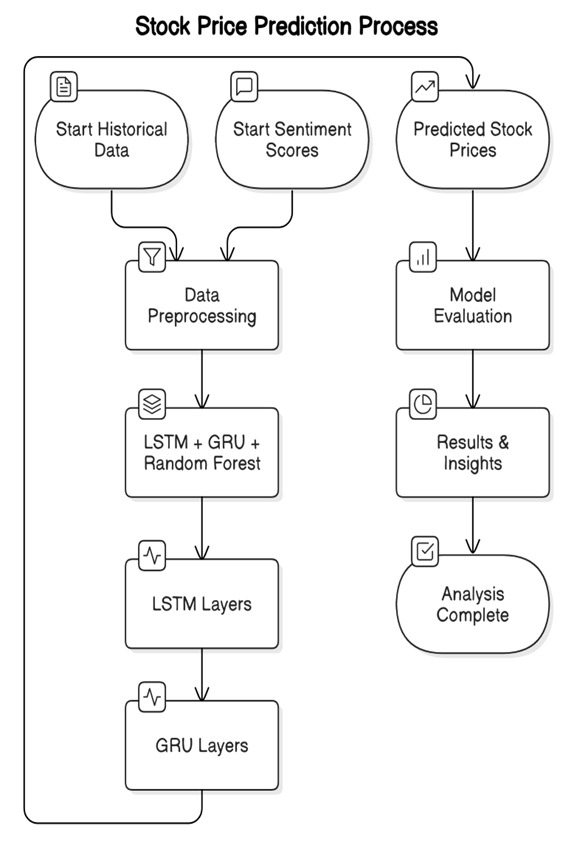
**Dropout Layers:**

Dropout layers are added after each LSTM layer with a dropout rate of 0.1 to prevent overfitting. Dropout randomly sets a fraction of input units to 0 during training, which helps to regularize the network.

**Dense Layer:**

A dense layer with one unit outputs the predicted stock price.

**Architecture Diagram**



**Features:**

Closing Price: Historical closing prices of the stock.

Sentiment Score: A sentiment score derived from news headlines using a sentiment analysis tool like VADER.

**Sentiment Analysis**

VADER (Valence Aware Dictionary and sentiment Reasoner) is employed to analyze the sentiment of news headlines. VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.

The sentiment score is a compound score ranging from -1 (most negative) to +1 (most positive).

**Data Preparation:**

Scaling: MinMaxScaler scales the input data to a range between 0 and 1. This is crucial for optimizing the training process.

Time Series Data Structure: The input data is structured into sequences using a sliding window approach to capture temporal dependencies.

**Training Process**

Optimizer: Adam (adam) - An efficient gradient descent algorithm well-suited for a variety of problems.

Loss Function: Mean Squared Error (mean\_squared\_error) - Quantifies the difference between predicted and actual values, suitable for regression tasks.

Epochs: 25 - The model is trained for 25 epochs, representing 25 complete passes through the training dataset.

**Output Data**

The model outputs a single value representing the predicted closing price for the next time step.

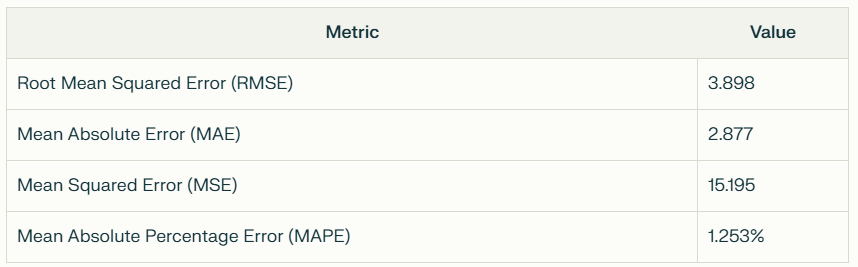
**Evaluation Metrics**

Root Mean Squared Error (RMSE): Measures the average magnitude of errors in the predicted stock prices.

Mean Absolute Error (MAE): Measures the average magnitude of the errors, less sensitive to outliers than RMSE.

Mean Squared Error (MSE): Penalizes larger errors more heavily.

Mean Absolute Percentage Error (MAPE): Provides a scale-independent measure of error.



**Future work**

Hyperparameter Tuning: Optimize hyperparameters such as LSTM units, dropout rates, and batch size.

Advanced Sentiment Analysis: Explore more sophisticated sentiment analysis techniques, such as fine-tuning a transformer model on financial news data.

Feature Engineering: Incorporate technical indicators or other relevant features.

Ensemble Methods: Combine multiple models to improve prediction accuracy.

**Limitations**

The model's performance is subject to the quality and availability of input data

**Conclusion**

SentiNova offers a framework for stock price prediction by integrating LSTM networks with sentiment analysis. The model's architecture, training process, and potential improvements are detailed in this document. The effectiveness of SentiNova depends on careful data preparation, hyperparameter tuning, and ongoing evaluation.