Time Series Prediction Using LSTM and Daily News

Arka Roy

01-08-2024

Introduction

In this project, I aimed to predict the daily price of the Dow Jones Industrial Average (DJIA) index by incorporating sentiment analysis from news headlines. I collected data on the top 25 news stories (Kaggle) and the daily price of DJIA from 08-08-2008 to 01-07-2016 (Yahoo finance). The goal was to see if the sentiment derived from news could improve the accuracy of stock price predictions.

Motive

The motive behind this project is to enhance the predictive power of stock price models by integrating sentiment analysis. Financial markets are significantly influenced by news and investor sentiment. Traditional models relying solely on historical price data often fail to capture the impact of market sentiment. By incorporating sentiment scores derived from news headlines, this project aims to develop a more robust model capable of predicting stock prices with greater accuracy.

Data Collection and Preprocessing

The data collection process involved two main sources:

- **News Headlines**: Gathered from Kaggle, the dataset contains daily news headlines of the top 25 news stories from 08-08-2008 to 01-07-2016.
- **Stock Prices**: Historical daily price data of the DJIA index was sourced from Yahoo Finance for the same period.

Data Cleaning

The initial step involved thorough data cleaning. For the news headlines:

- All headlines for a particular day were combined into a single text field.
- Non-alphanumeric characters were removed, leaving only letters, numbers, spaces, and periods.
- Extra spaces and punctuation were cleaned up to ensure uniformity.

For the DJIA data:

- The data was formatted consistently to match the dates of the news headlines.

Sentiment Score Calculation

Next, I calculated sentiment scores for each day using different models:

1. **VADER**: Provides a compound score indicating the overall sentiment on a scale from -1 (negative) to 1 (positive). VADER is particularly effective for social media texts.
2. **TextBlob**: Provides a sentiment polarity score ranging from -1 (negative) to 1 (positive). TextBlob uses a simple rule-based approach.
3. **BERT**: This powerful model for natural language understanding provides a sentiment score for the text. BERT captures context and nuances in text better than traditional models.
4. **BERT Fine-Tuned on Emotion Dataset**: This model provides scores for different emotions such as sadness, joy, love, anger, fear, and surprise. It offers a detailed emotional breakdown of the sentiment.

The sentiment scores from each of these models were stored in separate columns in the dataset. This allowed for a comprehensive analysis of how different sentiment models impact stock price prediction.

Feature Engineering

To prepare the data for the LSTM model, the following features were engineered:

- **Lag Features**: Past 7 days of stock prices and volumes traded were used as input features.
- **Sentiment Features**: Daily sentiment scores from the different models were included.
- **Combined Features**: Both the lagged stock prices and sentiment scores were used to create a combined feature set for the model.

Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) that is capable of learning long-term dependencies. Unlike traditional RNNs, LSTMs can effectively handle the vanishing gradient problem, making them suitable for time series prediction tasks. The key components of LSTMs include:

- **Cell State**: Acts as a conveyor belt, carrying relevant information across the sequence.
- **Forget Gate**: Decides what information to discard from the cell state.
- **Input Gate**: Decides which new information to store in the cell state.
- **Output Gate**: Decides what information to output based on the cell state and the current input.

Model Training and Testing

The dataset was split into a training set (70%) and a testing set (30%). The LSTM model was then trained using the following steps:

- **Normalization**: All features were normalized to ensure uniform scaling.
- **Model Architecture**: The LSTM model was designed with multiple layers to capture temporal dependencies.
- **Training**: The model was trained using the training dataset with early stopping and dropout to prevent overfitting.
- **Evaluation**: The model was evaluated on the testing set using the R^2 (coefficient of determination) metric.

Results

The performance of the models was evaluated using the R^2 metric. Here are the results:

- **VADER**: $R^2 = 0.680813$
- **TextBlob**: $R^2 = 0.867332$
- **BERT**: $R^2 = 0.885703$
- **BERT Fine-Tuned on Emotion Dataset**: $R^2 = 0.854689$
- **Without News**: $R^2 = 0.835450$

Analysis of Results

The results indicate that incorporating sentiment analysis from news headlines significantly improved the accuracy of stock price predictions. Among the models, BERT provided the highest R^2 score, indicating the best performance. The BERT model's ability to understand context and capture nuanced sentiment likely contributed to its superior performance.

Conclusion

Incorporating sentiment analysis from news headlines significantly improved the accuracy of stock price predictions. Among the models, BERT provided the highest R^2 score, indicating the best performance. Additionally, in the plot of test and predicted prices, we can see that sudden spikes or drops in the data cannot be predicted by the model trained only on the past 7 days of prices. In contrast, the model trained along with BERT scores can capture these sudden ups or downs. This shows that advanced sentiment analysis techniques, especially those using BERT, can effectively capture the market sentiment reflected in news headlines, leading to better prediction models for stock prices.

Future Work

Future work can explore the following directions:

- **Additional Sentiment Models**: Exploring other sentiment models and ensembling them could further improve prediction accuracy.
- **Longer Time Horizons**: Experimenting with different time windows for lag features could provide insights into the optimal lookback period.
- **Real-Time Data**: Incorporating real-time news data and testing the model's performance in a live trading environment.
- **Feature Selection**: Using advanced feature selection techniques to identify the most significant features for stock price prediction.
- **Github Repository**: Link