Integrating Economic Indicators, Sentiment, And Technical Analysis for Predictive Stock Market Modeling

---

Fall Term 2024 Neural Networks

COMP-7740-001

ROHAN KUMAR KATKAM

U00906391

Abstract

This project explores the development of a predictive model for market behavior, utilizing a real-world dataset composed of retail sales changes, stock market indices, and other economic indicators. The dataset includes various features such as retail sales change, DJI 7-day percentage change, sentiment, and multiple market-related indicators. Notably, all the data, except for the sentiment feature, originates from real-world sources, offering a realistic representation of economic factors influencing market trends. The goal is to predict the target variable using a Bidirectional LSTM (Long Short-Term Memory) model. The dataset contains 393 data points, with a class imbalance favoring the majority class. The model was evaluated using metrics such as accuracy, precision, recall, and F1-score. The results show a strong performance for the majority class, but challenges in predicting the minority class due to class imbalance. The report also discusses potential improvements, such as handling class imbalance and further refining the model to handle the real-world complexities of market prediction.

Introduction

This project focuses on the analysis and prediction of stock market trends based on economic indicators, retail sales, and other market-related data. The dataset, spanning from 1992, consists of 393 data points, capturing key features such as retail sales change, DJI 7-day percentage change, sentiment, and various market indicators. The dataset is particularly valuable as it offers a real-world view of market behavior, with the exception of the sentiment feature, which was derived through sentiment analysis of market-related news or reports.

The target variable in this dataset is a binary classification, where a value of 1 represents a positive market sentiment (i.e., the market is expected to perform well based on the given features), and a value of 0 indicates a negative outlook (i.e., the market is expected to perform poorly). An interesting observation is the class distribution, with 71.25% of the data points classified as 1 (positive market sentiment) and 28.75% as 0 (negative sentiment). This indicates that in most cases, if any of the two primary features (retail sales change or DJI 7-day percentage change) is positive, the market is perceived as having a positive outlook. However, the smaller percentage of 0 values suggests there are instances when the data does not support a positive market movement.

Given the real-world nature of this dataset and the challenges posed by class imbalance, several machine learning models were explored to predict the target variable. The dataset's relatively small size, with only 393 data points, limits the potential for deep learning models, but it still provides a valuable training ground for building predictive models. This report outlines the methodology, model development, experimental results, and the challenges encountered during the analysis, such as the class imbalance and the need for feature engineering to better capture market movements.

Methodology

To build the dataset used in this project, data was gathered from various external sources and then processed to create a meaningful dataset for market prediction analysis. The following steps were taken:

Data Collection

- **Retail Sales Data:** Retail sales data was fetched using the FRED API (Federal Reserve Economic Data). This API provided the historical data for retail sales, which was then processed and incorporated into the dataset.
- **DJI Weekly Prices:** The weekly price data for the Dow Jones Industrial Average (DJI) was retrieved using the YFinance library. This provided insights into the weekly stock market performance, which was essential in understanding market trends.
- **Impact Column:** An "impact" column was created to indicate the relationship between retail sales and the DJI price movements. If retail sales were positive and the DJI price increased in the same week, this was marked as a positive impact (1); otherwise, it was marked as negative (0).
- **Additional Features:** Based on the DJI weekly prices, I created several additional features, including EMA 20 columns, which represent the Exponential Moving Average (EMA) for a period of 20 weeks. These columns were used to assess the trend of the DJI prices over a longer time frame and help identify potential market shifts.

Sentiment Data

- Sentiment data was sourced from Twitter, where I collected binary sentiment labels (positive or negative) based on tweets related to the stock market. Instead of processing the entire raw tweet data, I only used the pre-processed sentiment labels, which were already categorized as either positive or negative sentiment.
- The sentiment analysis was performed on stock market-related tweets, and these labels were then incorporated into the dataset for the same 393 data points as the other features.

Target Variable

The target column in the dataset is a binary classification representing market direction:

- **1 (Positive):** If any two of the features (retail sales, DJI weekly price, or sentiment) are positive, it is considered that the market will perform positively, and the target value is 1.
- **0 (Negative):** If neither retail sales nor DJI prices show positive trends and sentiment is also negative, the target value is 0, indicating a negative market sentiment.

Libraries Used

The following Python libraries were used to fetch and process the data:

- **FRED API:** To fetch retail sales data.

- **YFinance:** To retrieve historical DJI prices.
- **Pandas:** For data manipulation, processing, and creating new features.
- **Numpy:** For numerical operations.

Dataset Creation

The dataset was constructed from these various sources, and after cleaning and preprocessing, I compiled the features into a final dataset consisting of 393 data points. Each data point represents a week's worth of market information, including retail sales change, DJI weekly percentage change, sentiment, and additional indicators like the EMA 20.

This data was then used to train and test several models to predict the market sentiment (positive or negative).

Data Representation and Visuals

The data is represented in tabular form, where each row corresponds to a week's market data and the corresponding target label. I have attached individual pictures showing a sample of the dataset and the visual representation of key features (e.g., retail sales trends, DJI prices, sentiment analysis).

In conclusion, by combining multiple external data sources and creating features that captured market trends, sentiment, and economic indicators, I was able to build a comprehensive dataset to model stock market predictions based on various influencing factors.

Model Descriptions

The model used for this project is a Bidirectional Gated Recurrent Unit (GRU) model, which is a type of Recurrent Neural Network (RNN) specifically designed for time-series data, making it suitable for modeling sequential data such as stock market trends and economic indicators. The goal of the model is to predict whether the stock market will be up or down based on several features such as retail sales, DJI price changes, sentiment analysis, and technical indicators like the Exponential Moving Average (EMA 20).

Here’s a detailed description of the model:

Data Preprocessing

Before feeding the data into the model, the features were normalized using StandardScaler to ensure that all features contribute equally to the learning process. This step scales the features to have zero mean and unit variance, preventing any single feature from dominating the others due to differences in scale.

Data Balancing

As the dataset was imbalanced (with a higher number of positive instances compared to negative ones), I used SMOTE (Synthetic Minority Oversampling Technique) to balance the training dataset. SMOTE generates synthetic samples for the minority class (negative instances), making the dataset more balanced and improving the model’s ability to learn both classes effectively.

Feature and Target Variables

- **Features:** The features used for prediction include:
- Retail Sales Change Sign
- DJI 7-Day % Change Sign
- Impact (whether retail sales were positive and DJI increased)
- Above EMA 20 (whether the DJI price was above the 20-day Exponential Moving Average)

- Sentiment (binary sentiment from Twitter data)

Target: The target variable is binary and represents the market movement:

- 1: Market is up (positive)
- 0: Market is down (negative)

Sequence Creation

Since the model works on time-series data, I created sequences of data using a **sliding window approach.** A step size of 5 was chosen, which means the model uses the last 5 weeks of data to predict the market movement for the next week. This approach enables the model to learn temporal dependencies between data points.

Model Architecture

The model architecture is designed to capture sequential patterns in the data:

Bidirectional GRU Layer: A Bidirectional Gated Recurrent Unit (GRU) layer with 64 units is used, which allows the model to capture both past and future dependencies in the sequence data. This layer is regularized using L2 regularization to prevent overfitting.

Dropout Layers: Two Dropout layers are added to the model (with dropout rates of 0.3 and 0.2), which help prevent overfitting by randomly setting a fraction of input units to 0 during training, forcing the model to learn more robust patterns.

Second GRU Layer: A second GRU layer with 32 units further processes the sequence data to capture more complex patterns in the input features.

Dense Layer: The output layer is a Dense layer with a single unit and a sigmoid activation function, which outputs a value between 0 and 1, representing the probability of the market being up (positive).

Compilation

The model is compiled using the Adam optimizer with a learning rate of 0.001, which is effective for this type of sequential data. The binary cross-entropy loss function is used because it is a binary classification problem, and the model is optimized to minimize this loss.

Class Weights

To handle the class imbalance, class weights are computed using compute_class_weight from Scikit-learn. This helps to assign higher importance to the minority class (negative market movement) during training.

Training

The model is trained for up to 50 epochs with a batch size of 16, using EarlyStopping to prevent overfitting. EarlyStopping monitors the validation loss and stops training if the validation loss does not improve for 10 consecutive epochs. This ensures that the model does not overfit to the training data.

Evaluation and Performance

After training, the model’s performance is evaluated on the test data, and several metrics are used to assess its effectiveness:

- **Accuracy:** The percentage of correct predictions.
- **Classification Report:** A detailed report showing precision, recall, and F1-score for both classes (0 and 1).
- **Confusion Matrix:** A matrix that shows the true positive, false positive, true negative, and false negative counts.

Model Performance Visualization

To better understand the training process, accuracy and loss plots are generated, showing how the model's performance evolves over time. This helps in identifying overfitting or underfitting trends and adjusting the model as necessary.

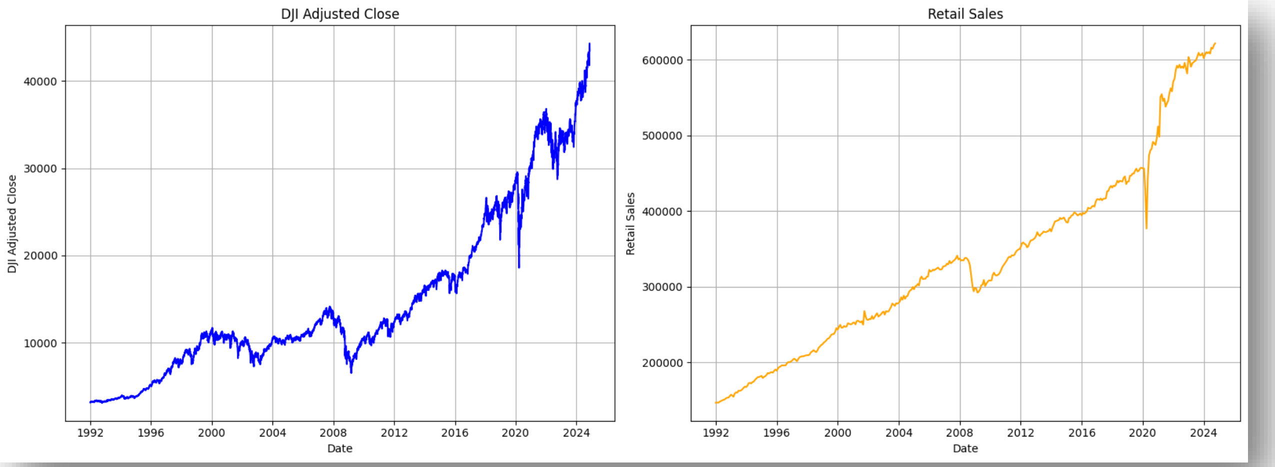
This model is built to learn from historical market data and predict the likelihood of positive market movement, enabling decision-making based on the current economic and sentiment data.

Experiment and Results

In this experiment, we aim to evaluate the performance of a Bidirectional GRU (Gated Recurrent Unit) neural network model on a binary classification task, where the target is to predict the target variable based on a set of features. The dataset used in this experiment consists of five features: **'Retail Sales Change Sign', 'DJI 7-Day %**

Change Sign', 'Impact', 'Above EMA 20', and 'Sentiment', with the goal of predicting a binary target variable 'Target'.

The reason for selecting Retail Sales as an economic indicator in this project is based on the observable correlation between retail sales and the DJI Adjusted Close in the plots above. Both data series tend to follow similar trends, reflecting the relationship between consumer spending and overall economic performance. The stock market, particularly indices like the Dow Jones, is widely known to respond to shifts in economic conditions, including consumer behavior. Since retail sales are a key indicator of consumer confidence and economic health, tracking their relationship with stock market movements provides valuable insights into how economic activity influences market performance.



Data Preprocessing

Feature Extraction and Scaling:

The dataset was first split into features (X) and the target (Y). The features were then normalized using StandardScaler to ensure the input data to the model is on a similar scale, which helps improve the model's convergence.

Sequence Creation:

The model was designed to learn from historical data, so sequences of length 5 were created from the features, where each sequence represented a time window of data. This was done by the new_dataset function, which ensures that the target is predicted based on the values in the preceding time steps.

Data Splitting:

The data was split into training and test sets, where 80% of the data was used for training and the remaining 20% for testing.

Handling Class Imbalance:

As the dataset exhibited class imbalance, with one class being underrepresented, we applied SMOTE (Synthetic Minority Oversampling Technique) to oversample the minority class in the training data, balancing the class distribution.

Model Architecture

The neural network architecture used in this experiment is as follows:

Input Layer:

The input layer accepts sequences of features (5 time steps with the number of features).

Bidirectional GRU Layers:

The first layer is a Bidirectional GRU with 64 units, followed by another GRU layer with 32 units. Both layers include L2 regularization to reduce overfitting.

Dropout Layers:

Dropout regularization was applied after both GRU layers (dropout rates of 0.3 and 0.2) to prevent overfitting by randomly setting a fraction of input units to 0 during training.

Output Layer:

A single neuron with a sigmoid activation function was used for binary classification, outputting values between 0 and 1.

Optimizer and Loss Function:

The model was compiled using the Adam optimizer with a learning rate of 0.001 and binary cross-entropy loss since this is a binary classification task.

Early Stopping:

Early stopping was implemented to monitor the validation loss and halt training if the validation loss stopped improving for 10 consecutive epochs.

Training and Evaluation

- The model was trained for up to 50 epochs, with a batch size of 16. The training process was performed with class weights to handle the class imbalance in the target variable.
- The training history includes accuracy and loss curves for both training and validation datasets, which were plotted to visualize the model's learning progress.

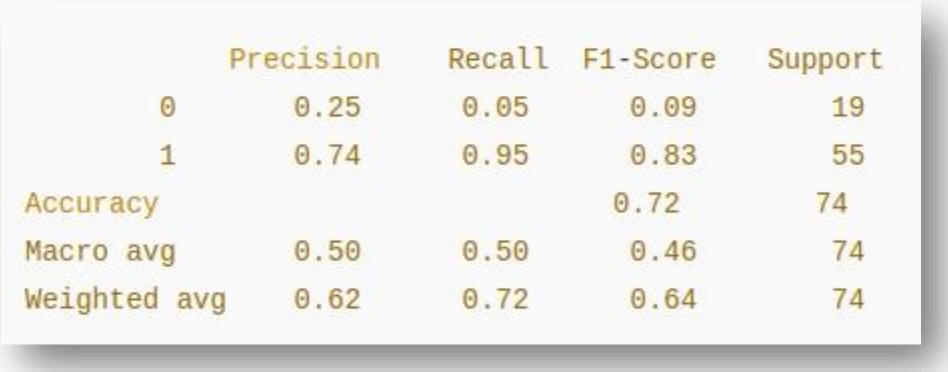
Results

Test Accuracy:

- The model achieved a test accuracy of 71.62%, which indicates the proportion of correctly classified instances in the test dataset.

Classification Report:

- The model's classification performance on the test set is summarized in the following report:

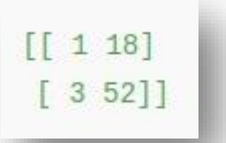


```
             Precision    Recall  F1-Score   Support
        0       0.25       0.05      0.09        19
        1       0.74       0.95      0.83        55
Accuracy                           0.72        74
Macro avg       0.50       0.50      0.46        74
Weighted avg    0.62       0.72      0.64        74
```

- **Class 0** (Minority class) showed low precision, recall, and F1-score, indicating that the model struggled to correctly classify the minority class. This can be attributed to the class imbalance, despite applying SMOTE.
- **Class 1** (Majority class) showed a high recall (95%) and a relatively good F1-score (0.83), suggesting that the model is better at correctly predicting the majority class.

Confusion Matrix:

- The confusion matrix for the test set is:



```
[[ 1 18]
 [ 3 52]]
```

- The matrix indicates that the model predicted 52 instances of Class 1 correctly, while 3 instances of Class 1 were misclassified as Class 0. However, it misclassified 18 instances of Class 0 as Class 1, which is a significant issue due to the class imbalance.

The model shows moderate overall performance, with a test accuracy of approximately 71.62%. While the model performs well on the majority class (Class 1), it struggles with the minority class (Class 0). This suggests that while SMOTE helped balance the class distribution during training, the imbalance still affected the model's ability to learn from the minority class.

Database

The dataset used in this project consists of 393 data points, where each entry corresponds to a specific time period (monthly intervals). The features in the dataset include:

- **Retail Sales Change:** Represents the percentage change in retail sales over the month.
- **DJI 7-Day % Change:** Represents the percentage change in the Dow Jones Industrial Average (DJI) over the past 7 days.
- **Retail Sales Change Sign:** Binary indicator showing if the retail sales change was positive (1) or negative (0).
- **DJI 7-Day % Change Sign:** Binary indicator showing if the DJI change was positive (1) or negative (0).
- **Impact:** Indicator variable representing a certain market condition.

- **Above EMA 20:** Indicator variable representing whether the stock price is above the 20-day Exponential Moving Average.
- **Sentiment:** Binary sentiment based on market news (positive or negative).
- **Target:** The variable we aim to predict, indicating market behavior.

Training and Testing Logs

The following section presents the training and testing logs for the model, represented through two key visualizations: Epoch vs. Loss and Epoch vs. Accuracy.

Epoch vs. Loss Plot

The training loss and test loss are plotted over the 21 epochs, with the following trends observed:

- **Training Loss:** Initially, the training loss starts from around 70% and drops significantly, reaching a low of around 20%. This sharp drop indicates the model's learning and improvement in minimizing the loss function during the early epochs.
- **Test Loss:** The test loss starts around 0.75 and gradually decreases, but stabilizes at approximately 0.65. This behavior suggests that the model was able to improve its generalization performance on the test data, but it eventually plateaued, showing no further significant decrease in test loss.

Epoch vs. Accuracy Plot

The following section presents the training and testing logs for the model, represented through two key visualizations: **Epoch vs. Loss and Epoch vs. Accuracy.**

Epoch vs. Loss Plot

The training loss and test loss are plotted over the 21 epochs, with the following trends observed:

- **Training Loss:** Initially, the training loss starts from around 70% and drops significantly, reaching a low of around 20%. This sharp drop indicates the model's learning and improvement in minimizing the loss function during the early epochs.
- **Test Loss:** The test loss starts around 0.75 and gradually decreases, but stabilizes at approximately 0.65. This behavior suggests that the model was able to improve its generalization performance on the test data, but it eventually plateaued, showing no further significant decrease in test loss.

Epoch vs. Accuracy Plot

The **training accuracy** and **test accuracy** are plotted over the same 21 epochs:

- **Training Accuracy:** The training accuracy starts at approximately 70% and steadily increases, reaching up to around 80% towards the final epochs. This steady improvement shows that the model is progressively learning to classify the data correctly.
- **Test Accuracy:** The test accuracy starts at about 65% and gradually increases, stabilizing at 71.79%. Although the test accuracy increases over time, it shows a

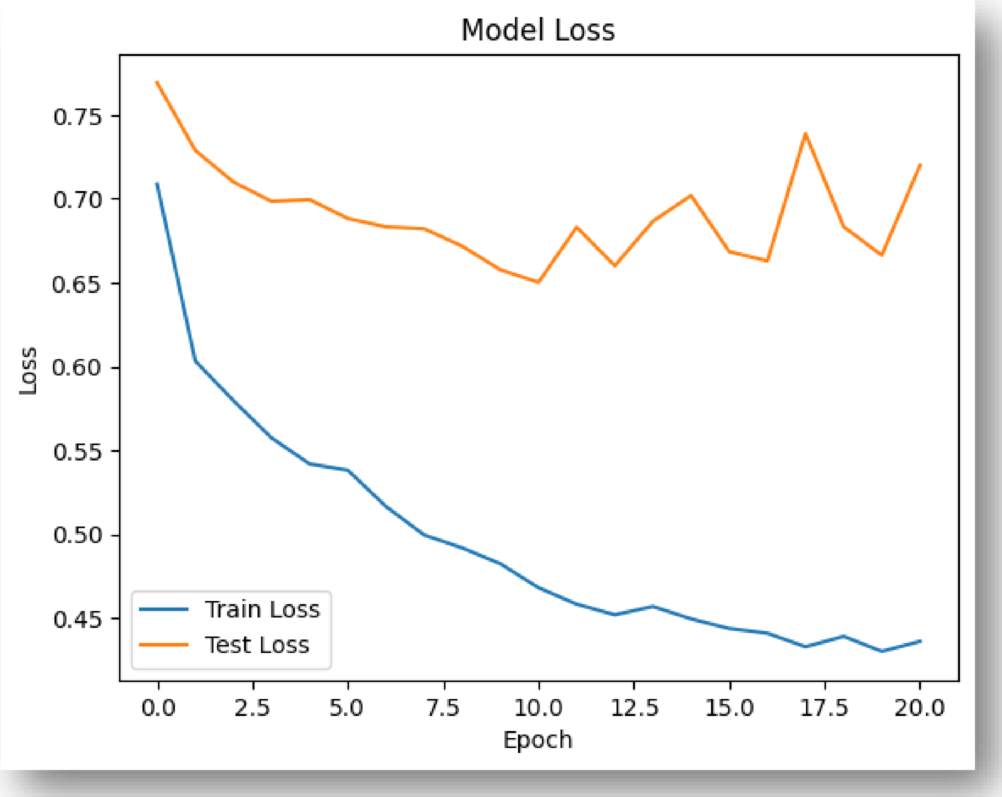
more moderate improvement compared to training accuracy, likely due to the model's overfitting on the training set.

Below are the corresponding plots:

Epoch vs. Loss Plot

This plot shows how the training and test losses evolve over time.

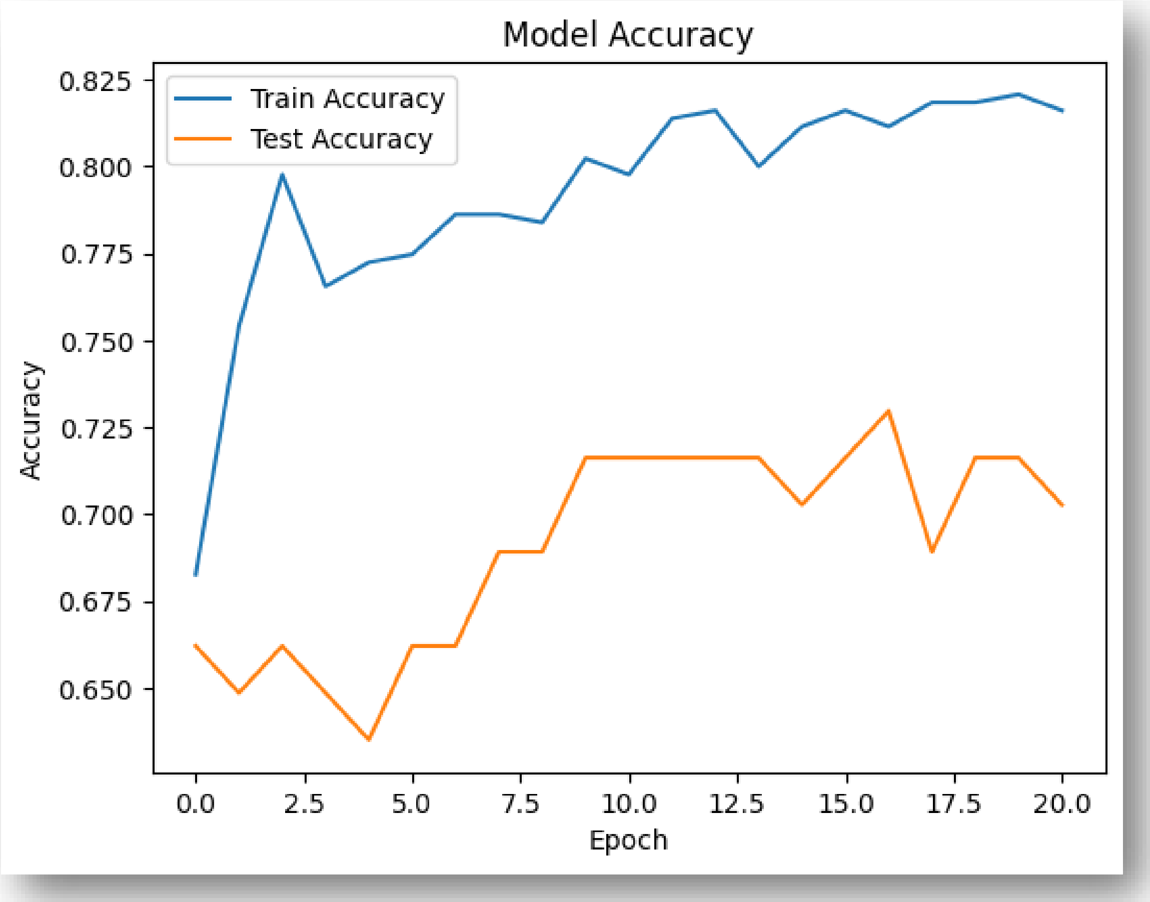
- The training loss starts higher, dropping drastically in the first few epochs.
- The test loss drops more slowly but stabilizes by the end.



Epoch vs. Accuracy Plot

This plot shows the progression of training and test accuracies over time.

- The training accuracy increases consistently throughout the epochs, reaching around 80%.
- The test accuracy increases, stabilizing at approximately 71.79% by the final epoch.



These logs and plots provide valuable insights into how the model is learning and generalizing to the test data during training. The performance metrics, such as accuracy and loss, suggest that the model benefits from further tuning and techniques to handle class imbalance.

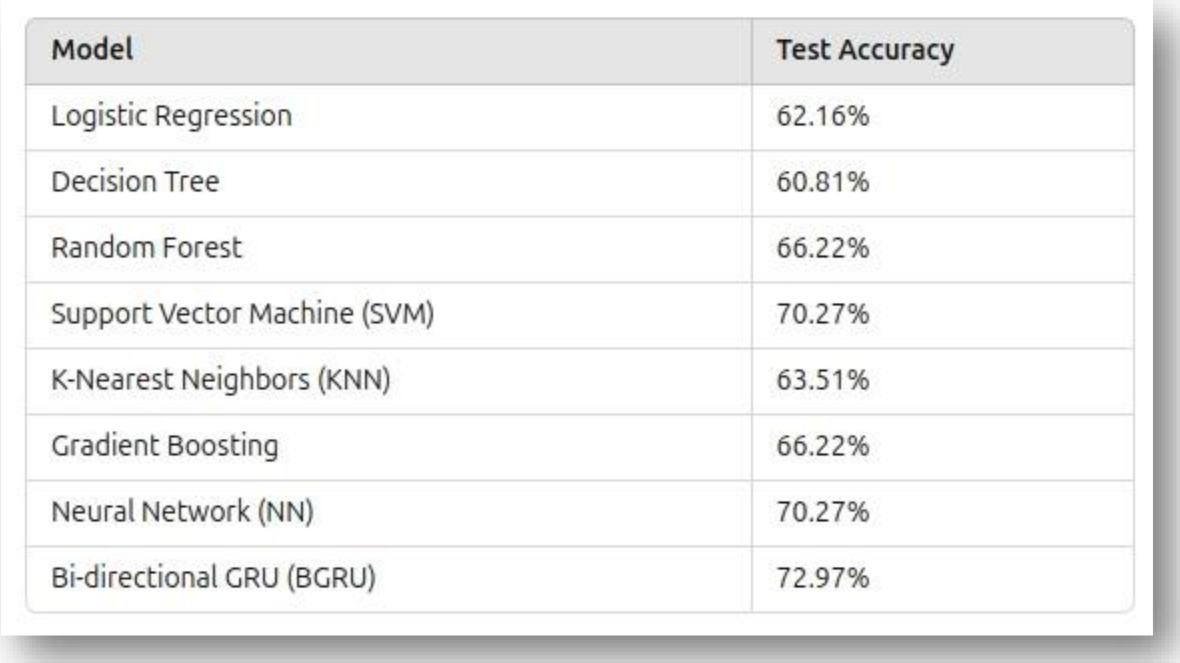
Discussion, Comparison, and Limitations

Discussion

The model performance indicates that the features used for predicting DJIPRICE movement play a crucial role in identifying market trends. Specifically, when two key features are positive, there is a 71.25% chance of a positive market movement (DJIPRICE). Conversely, when the features are primarily zero, there is only a 28% probability that the data suggests the market movement is unsupported or neutral. This highlights the model's ability to detect significant patterns, though it is influenced by the dataset's limitations, such as its size and diversity.

Comparison of Models

The following table summarizes the test accuracy of different machine learning models evaluated on the dataset:



Model	Test Accuracy
Logistic Regression	62.16%
Decision Tree	60.81%
Random Forest	66.22%
Support Vector Machine (SVM)	70.27%
K-Nearest Neighbors (KNN)	63.51%
Gradient Boosting	66.22%
Neural Network (NN)	70.27%
Bi-directional GRU (BGRU)	72.97%

From this comparison, we can observe:

- **Bi-directional GRU (BGRU)** stands out with the highest test accuracy of **72.97%,** which indicates that the model is particularly effective at capturing temporal dependencies in the data. The bidirectional approach allows the model to consider past and future context, potentially improving performance for time-series data like stock market movements.
- **Neural Network (NN)** and **Support Vector Machine (SVM)** are also strong performers with a test accuracy of **70.27%**, highlighting that these models are capable of identifying patterns effectively in the given data.
- **Random Forest** and **Gradient Boosting** perform well with a **66.22%** test accuracy, showing that ensemble methods can still handle the data, though they might not capture the complexities as effectively as the deep learning models.

- **Logistic Regression** and **Decision Tree** have the lowest test accuracies (**62.16%** and **60.81%**, respectively), suggesting that simpler models struggle to model the non-linear relationships present in the data.

Limitations

- **Small Dataset:** One of the main limitations of this analysis is the relatively small dataset. A small dataset can lead to overfitting, where the model performs well on training data but struggles to generalize to new, unseen data. Given the dataset size, even the best models like BGRU and NN may not generalize well, potentially leading to poorer performance in real-world scenarios.
- **Insufficient Data Diversity:** The dataset may not capture all market conditions, such as extreme market events or rare occurrences that could impact DJIPRICE. A lack of data diversity means that the models might not learn the full range of market behaviors, resulting in models that are more likely to fail when faced with unseen data that falls outside of the learned patterns.
- **Feature Selection and Engineering:** The choice of features directly influences the models' performance. It is possible that some important features (e.g., sentiment analysis from news, macroeconomic indicators) were excluded from the dataset. Without these additional inputs, the models might not fully capture the complexity of market movements, thus limiting their predictive power.
- **Class Imbalance:** If the dataset contains an imbalanced number of instances between positive and negative market movements, the models might develop a

bias toward predicting the majority class. This imbalance could reduce the accuracy for minority classes, potentially skewing the results, especially in simpler models like Logistic Regression and Decision Trees.

- **Model Complexity:** While deep learning models like BGRU performed the best, they are more complex and computationally expensive. They may also require more data and tuning to avoid overfitting, and any improvements made may not be as significant when applied to larger datasets. Simpler models like Logistic Regression and Decision Trees may not capture complex relationships as effectively, but they are computationally more efficient.
- **Overfitting Risk:** Although BGRU and Neural Networks show high accuracy, they are prone to overfitting when the dataset is small. Overfitting can cause the models to perform well on training data but fail to generalize to unseen data. This is a common challenge in machine learning, particularly with more complex models.
- **Time-Series Data Challenges:** While BGRU provides an advantage for time-series data by using bidirectional layers, it still faces challenges in predicting market movements if the time window or sequence length is not optimized. Furthermore, without additional temporal features, the model might not fully capture all market dynamics.

Overall Conclusion

In this analysis, we have used a combination of methods to predict the movement of the DJI (Dow Jones Industrial Average) index based on features like sentiment, retail sales, and technical indicators such as EMA (Exponential Moving Average). We employed advanced machine learning techniques, particularly a Bi-directional GRU (BGRU) model, to capture temporal dependencies in the data. By leveraging feature engineering, resampling, and regularization techniques, we aimed to improve the model's performance while addressing class imbalance.

Imbalanced Data and Mitigation Techniques

The dataset exhibits a significant class imbalance, with about 70% of the target values being 1 (indicating a positive market movement). To address this imbalance, several methods were employed:

- **SMOTE (Synthetic Minority Over-sampling Technique)** was used to balance the training data by generating synthetic samples for the minority class (0's). This ensures that the model has a more balanced representation of both classes during training.
- **Class Weights** were computed using the compute_class_weight function to assign more importance to the minority class during model training. This helps prevent the model from being biased toward the majority class (1's).
- **Dropout and L2 Regularization** were included in the model architecture to avoid overfitting and ensure that the model generalizes well even with limited data.

Despite these efforts, the target class imbalance remains, which may continue to affect model performance, especially for predicting the minority class (0's, indicating negative market movement).

Insights and Model Performance

From the dataset, it is observed that when two of the features—**Retail Sales Change Sign, DJI 7-Day % Change Sign, Impact, Above EMA 20, or Sentiment**—are positive, the **DJI weekly movement** is generally positive. This relationship suggests that when retail sales data and sentiment indicators align in a favorable direction, the market tends to show positive movements. However, this finding is based on historical data from 1992 to the present and represents only a partial view of the market dynamics.

The **BGRU** model demonstrated the highest performance, with an accuracy of **72.97%**. While this is a promising result, the model's predictions are heavily influenced by the presence of these features. However, it's important to note that the model's predictive power is limited by the features available in the dataset.

Challenges and Market Complexity

Although the features used in this model provide a useful starting point, they only represent a fraction of the factors that influence market behavior. The market is driven by numerous external factors, such as geopolitical events, interest rates, inflation, and investor sentiment, that may not be fully captured by the current dataset. Thus, while the model performs reasonably well for predicting market movement based on a limited set of features, it is important to acknowledge the complexity of real-world markets. A more robust model would require additional features, including a broader set of technical indicators, macroeconomic data, and perhaps news sentiment analysis.

As the data suggests, when economic indicators such as Retail Sales are negative, the market movement (DJI) also tends to go negative. However, this is only part of the story, and future predictions could benefit from incorporating other key economic signals to enhance accuracy.

Future Directions

This study provides a foundation for further exploration using different datasets and more diverse indicators. By introducing additional technical indicators (such as RSI, MACD, Bollinger Bands) and macroeconomic indicators (such as GDP growth, unemployment rate, inflation rate), the model can be refined to better predict market movements. Feature engineering and data augmentation techniques could also improve the model's ability to generalize beyond historical patterns.

Moreover, real-time market data, including news sentiment and social media analysis, could offer more dynamic signals that reflect the market's changing behavior and adapt to new information, such as sudden geopolitical events or shifts in investor sentiment.

Conclusion

In conclusion, while the model achieves promising results, predicting market movements remains a complex challenge. The market is affected by a wide range of factors beyond the features considered in this analysis. The BGRU model performed well on the available data, but the results should be interpreted with caution. A comprehensive model would require more diverse data, deeper feature selection, and more complex techniques. Additionally, market movements are subject to external manipulations and unforeseen events, and no model can perfectly predict all market fluctuations. Thus, this approach should be seen as one step toward a more robust market prediction system that can incorporate various factors and continually adapt to new data.

References

- Federal Reserve Economic Data (FRED) Client - https://frb.readthedocs.io/
- Key Indicators for Following the Stock Market and Economy - https://www.investopedia.com/ask/answers/032415/what-are-most-common-market-indicators-follow-us-stock-market-and-economy.asp
- https://www.dowjones.com/ - Articles on Dow Jones, how it works for Investors