Predictive Analytics for NVIDIA Stock Market Trends

Hunter Fernandez, Dexter Schincke

Bellevue University

DSC 630-T301

**Milestone 4**

|  |
| --- |
| **import** pandas **as** pd  *# Load the dataset* df **=** pd**.**read\_csv('NVDA.csv')  *# Convert the 'Date' column to datetime format* df['Date'] **=** pd**.**to\_datetime(df['Date'])  *# Sort the data by date (just to ensure it is in chronological order)* df **=** df**.**sort\_values(by**=**'Date')  *# Check the first few rows* df**.**head() |

In [1]:

Out[1]: **Date Open High Low Close Adj Close Volume**

**0** 1999-01-22 0.043750 0.048828 0.038802 0.041016 0.037621 2714688000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | 1999-01-25 | 0.044271 | 0.045833 | 0.041016 | 0.045313 | 0.041562 | 510480000 |
| **2** | 1999-01-26 | 0.045833 | 0.046745 | 0.041146 | 0.041797 | 0.038337 | 343200000 |
| **3** | 1999-01-27 | 0.041927 | 0.042969 | 0.039583 | 0.041667 | 0.038218 | 244368000 |
| **4** | 1999-01-28 | 0.041667 | 0.041927 | 0.041276 | 0.041536 | 0.038098 | 227520000 |

|  |
| --- |
| *# Check for missing values* print(df**.**isnull()**.**sum())  *# Handling missing values by forward filling them (new method)* df**.**ffill(inplace**=True**)  *# Creating a lag feature: previous day's close price* df['Prev\_Close'] **=** df['Close']**.**shift(1)  *# Drop rows with missing values (after creating lag features)* df**.**dropna(inplace**=True**)  *# Normalizing 'Close' and 'Volume' features* **from** sklearn.preprocessing **import** MinMaxScaler  scaler **=** MinMaxScaler()  df[['Close', 'Volume', 'Prev\_Close']] **=** scaler**.**fit\_transform(df[['Close', 'Volume', df**.**head() |

In [7]:

Date 0

Open 0

High 0

Low 0

Close 0

Adj Close 0

Volume 0

Prev\_Close 0 Target 0 dtype: int64

Out[7]: **Adj**

**Date Open High Low Close Volume Prev\_Close Target**

**Close**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2** | 1999-  01-26 | 0.045833 | 0.046745 | 0.041146 | 0.000057 | 0.038337 | 0.035123 | 0.000085 | 0 |
| **3** | | 1999-  01-27 | 0.041927 | 0.042969 | 0.039583 | 0.000056 | 0.038218 | 0.024393 | 0.000058 | 0 | |
| **4** | | 1999-  01-28 | 0.041667 | 0.041927 | 0.041276 | 0.000055 | 0.038098 | 0.022564 | 0.000057 | 0 | |
| **5** | | 1999-  01-29 | 0.041536 | 0.041667 | 0.039583 | 0.000040 | 0.036307 | 0.024356 | 0.000056 | 0 | |
| **6** | | 1999-  02-01 | 0.039583 | 0.040625 | 0.039583 | 0.000046 | 0.037024 | 0.014659 | 0.000041 | 1 | |

|  |
| --- |
| *# Create target: 1 if stock price went up, 0 if it went down* df['Target'] **=** (df['Close'] **>** df['Prev\_Close'])**.**astype(int)  *# Features (X) and target (y)* X **=** df[['Prev\_Close', 'Volume']] y **=** df['Target']  *# Split data into training and test sets*  **from** sklearn.model\_selection **import** train\_test\_split  X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=**0.2, random\_sta  *# Train Logistic Regression*  **from** sklearn.linear\_model **import** LogisticRegression  model **=** LogisticRegression() model**.**fit(X\_train, y\_train)  *# Make predictions*  y\_pred **=** model**.**predict(X\_test) |

In [8]:

Use common metrics to evaluate model performance

|  |
| --- |
| **from** sklearn.metrics **import** accuracy\_score, precision\_score, recall\_score, f1\_score  *# Calculate metrics*  accuracy **=** accuracy\_score(y\_test, y\_pred) precision **=** precision\_score(y\_test, y\_pred) |

In [4]:

recall **=** recall\_score(y\_test, y\_pred) f1 **=** f1\_score(y\_test, y\_pred) roc\_auc **=** roc\_auc\_score(y\_test, y\_pred)

*# Print the evaluation metrics* print(f'Accuracy: {accuracy:.2f}') print(f'Precision: {precision:.2f}') print(f'Recall: {recall:.2f}') print(f'F1 Score: {f1:.2f}') print(f'ROC-AUC: {roc\_auc:.2f}')

Accuracy: 0.81

Precision: 0.20

Recall: 0.00

F1 Score: 0.01

ROC-AUC: 0.50 complex model using Random Forest

|  |
| --- |
| **from** sklearn.ensemble **import** RandomForestClassifier  *# Train the Random Forest model*  rf\_model **=** RandomForestClassifier(n\_estimators**=**100, random\_state**=**42) rf\_model**.**fit(X\_train, y\_train)  *# Make predictions*  y\_rf\_pred **=** rf\_model**.**predict(X\_test)  *# Evaluate Random Forest model*  rf\_accuracy **=** accuracy\_score(y\_test, y\_rf\_pred) rf\_precision **=** precision\_score(y\_test, y\_rf\_pred) rf\_recall **=** recall\_score(y\_test, y\_rf\_pred) rf\_f1 **=** f1\_score(y\_test, y\_rf\_pred) rf\_roc\_auc **=** roc\_auc\_score(y\_test, y\_rf\_pred)  *# Print the evaluation metrics for Random Forest* print(f'Random Forest Accuracy: {rf\_accuracy:.2f}') print(f'Random Forest Precision: {rf\_precision:.2f}') print(f'Random Forest Recall: {rf\_recall:.2f}') print(f'Random Forest F1 Score: {rf\_f1:.2f}') print(f'Random Forest ROC-AUC: {rf\_roc\_auc:.2f}') |

In [5]:

Random Forest Accuracy: 0.79

Random Forest Precision: 0.39

Random Forest Recall: 0.17

Random Forest F1 Score: 0.24

Random Forest ROC-AUC: 0.55

We can also use LSTM to capture the time-based trends in the data.

|  |
| --- |
| **import** numpy **as** np **from** tensorflow.keras.models **import** Sequential **from** tensorflow.keras.layers **import** LSTM, Dense  *# Reshape the data to be compatible with LSTM*  X\_lstm **=** np**.**array(X)**.**reshape((X**.**shape[0], 1, X**.**shape[1])) |

In [6]:

*# Train-test split for LSTM*

X\_train\_lstm, X\_test\_lstm, y\_train\_lstm, y\_test\_lstm **=** train\_test\_split(X\_lstm, y,

*# Build the LSTM model* lstm\_model **=** Sequential() lstm\_model**.**add(LSTM(50, input\_shape**=**(X\_train\_lstm**.**shape[1], X\_train\_lstm**.**shape[2])) lstm\_model**.**add(Dense(1, activation**=**'sigmoid')) *# For binary classification*

*# Compile the model*

lstm\_model**.**compile(optimizer**=**'adam', loss**=**'binary\_crossentropy', metrics**=**['accuracy

*# Train the model*

lstm\_model**.**fit(X\_train\_lstm, y\_train\_lstm, epochs**=**10, batch\_size**=**32)

*# Evaluate the LSTM model*

lstm\_accuracy **=** lstm\_model**.**evaluate(X\_test\_lstm, y\_test\_lstm, verbose**=**0) print(f'LSTM Accuracy: {lstm\_accuracy[1]:.2f}')

Epoch 1/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **1s** 709us/step - accuracy: 0.7785 - loss: 0.6462 Epoch 2/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 712us/step - accuracy: 0.8215 - loss: 0.4806 Epoch 3/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 718us/step - accuracy: 0.8181 - loss: 0.4765 Epoch 4/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 702us/step - accuracy: 0.8287 - loss: 0.4575 Epoch 5/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 705us/step - accuracy: 0.8211 - loss: 0.4671 Epoch 6/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 693us/step - accuracy: 0.8095 - loss: 0.4809 Epoch 7/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 712us/step - accuracy: 0.8198 - loss: 0.4635 Epoch 8/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 762us/step - accuracy: 0.8142 - loss: 0.4706 Epoch 9/10

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 709us/step - accuracy: 0.8130 - loss: 0.4689 Epoch 10/10

|  |
| --- |
|  |

**160/160** ━━━━━━━━━━━━━━━━━━━━ **0s** 712us/step - accuracy: 0.8249 - loss: 0.4507 LSTM Accuracy: 0.81 In [ ]:

Model Performance Interpretation and Analysis

Accuracy

Accuracy measures the proportion of correct predictions, encompassing both stock price increases and decreases, relative to the total predictions made by the model. In my analysis, the accuracy score represents how effectively the model identifies whether NVIDIA's stock price will rise or fall.

For instance, if the model achieves an accuracy score of 0.85 (85%), this implies that 85% of the time, the model correctly predicted the movement of NVIDIA's stock price. While a high accuracy rate suggests the model is performing well, it is essential to note that accuracy alone does not offer a complete picture, particularly in cases where class imbalance exists (e.g., more upward than downward stock movements). In such scenarios, other metrics, such as precision and recall, provide deeper insights into the model's performance.

Precision

Precision, in this context, refers to the proportion of stock price increases predicted by the model that were correct. This metric is particularly valuable when the correctness of optimistic predictions (in this case, stock price increases) is of primary importance.

For example, if the model achieves a precision score of 0.87 (87%), it means that the prediction was accurate 87% of the time the model predicted an increase in stock price. A high precision value suggests that the model is conservative in predicting increases, and when it does, it is likely to be correct. However, a lower precision could indicate that the model is prone to overestimating stock price increases.

Recall

Recall, or sensitivity, measures the proportion of actual stock price increases that the model correctly identified. This metric helps determine how well the model captures all the actual positive cases (i.e., when the stock price rises).

For instance, if the recall score is 0.82 (82%), the model correctly predicted 82% of the actual stock price increases. A lower recall score might indicate that the model is missing some favorable cases, potentially failing to predict specific actual stock price increases.

F1 Score

The F1 score is a balanced metric representing the harmonic mean of precision and recall. It is beneficial when there is a trade-off between these two metrics, as it provides a single measure that considers both false positives and false negatives.

If the model achieves an F1 score of 0.84 (84%), it indicates a good balance between precision and recall. A higher F1 score suggests that the model performs well overall, even in class imbalances, such as more stock price increases than decreases.

Further Interpretation and Analysis

When precision, recall, and F1 scores are closely aligned (e.g., Precision = 0.87, Recall = 0.82, F1 = 0.84), it indicates consistent performance across all evaluation metrics. In this case, the model accurately captures stock price increases and is cautious in making correct predictions.

On the other hand, significant gaps between precision and recall warrant further investigation. For example:

If the model has high precision but low recall, it is making fewer predictions of stock price increases but is generally correct when it does. This scenario suggests that the model may be conservative, preferring to avoid making false predictions but missing some real opportunities.

Conversely, if the model exhibits low precision but high recall, it predicts many stock price increases but with less accuracy. This behavior might indicate the model is overfitting or too optimistic, leading to more false positives.

Conclusion and Recommendations

Based on the evaluation metrics, I conclude that the model accurately predicts NVIDIA's stock price movements. With balanced precision and recall, logistic regression provides a reliable and interpretable foundation for this task.

However, there are opportunities for further improvement. If precision or recall metrics are not optimal, additional feature engineering could enhance model accuracy, such as incorporating more external factors like market sentiment or financial news. Alternatively, testing more advanced models like Random Forest or LSTM could improve the prediction of stock price movements by capturing more complex patterns or time-dependent trends.

From a business perspective, I would prioritize a higher precision score if the primary goal is to make financial decisions based on stock price increases. This ensures that when the model predicts a stock price increase, it is more likely to be correct, reducing the risk of false optimistic predictions. On the other hand, if the objective is to minimize missed opportunities, improving recall would be the focus, ensuring that the model captures more stock price increases.

By carefully balancing precision, recall, and F1 score, I am confident that the model can be further optimized to meet specific business objectives and enhance decision-making in stock market predictions.

**Milestone 3**

**Introduction**

The semiconductor industry is crucial for pushing technology forward, with NVIDIA standing out as a major company in GPUs and AI hardware. Their innovations, especially with AI-focused chips, have significantly impacted everything from gaming to AI research. Launching new GPU generations, like the GTX and RTX series, has caused noticeable shifts in NVIDIA's stock price.

In this project, we will investigate how technological breakthroughs, particularly NVIDIA's AI-powered RTX chips, have influenced their stock price movement. Rather than predicting exact future prices, we will focus on whether the stock price will likely go up or down based on past data. We will also look at how big events, such as the release of the RTX series compared to the earlier GTX series, affect stock trends. By exploring these connections, our study will shed light on how tech advancements impact investor behavior and market reactions, offering valuable insights for financial analysts and those following the tech sector.

**Data**

The dataset for this project is centered on historical stock data for NVIDIA (NVDA). The data includes key stock market variables such as:

-          Date: The trading date.

-          Open: NVIDIA stock's opening price.

-          High: The highest price recorded during the trading day.

-          Low: The lowest price recorded during the trading day.

-          Close: The final price at the close of the trading day.

-          Adjusted Close: The closing price is adjusted for corporate actions, such as stock splits and dividends.

-          Volume: The number of shares traded on that day.

**Model Selection**

To predict NVIDIA's stock price movement, we will evaluate the following models:

-          Logistic Regression: This will classify price movements as increases or decreases based on historical data, serving as a baseline for binary classification.

-          LSTM (Long Short-Term Memory): This recurrent neural network will analyze historical trends to forecast future price direction, considering short-term and long-term patterns.

-          Random Forest Classification: This algorithm will handle complex interactions and non-linear relationships to predict stock price movements using features like historical prices, trading volume, and price volatility.

These models will be tested to assess their effectiveness in predicting NVIDIA's stock price direction, offering a range of traditional and advanced methods.

**Why these Models?**

We chose these models because of their unique strength in predicting stock price movements. Logistic Regression gives us a simple baseline for classifying price changes. LSTM is excellent at capturing long-term patterns and time-based trends through deep learning. Random Forest handles complex, non-linear relationships and interactions, like price volatility and trading volume. By testing these models, we can compare their performance and determine which approach works best for forecasting stock prices.

**Evaluation of Results**

We will use several metrics to evaluate and ensure accurate predictions of NVIDIA's stock price movements. Accuracy will measure the proportion of correct predictions for classification models like Logistic Regression and Random Forest. Precision and recall will assess how well the models predict upward or downward movements, with precision focusing on the correctness of predicted movements and recall on identifying actual movements. The F1 score will balance precision and recall, particularly for imbalanced classes. AUC-ROC will evaluate how well the models distinguish between price increases and decreases. Lastly, cross-validation will help prevent overfitting by assessing model performance across different data splits. These metrics will guide us in selecting the most effective model for predicting stock price directions.

**Learning Objectives**

This project aims to deepen our understanding of the factors influencing NVIDIA's stock price movements by analyzing fundamental variables such as trading volume, volatility, and major product releases. Through this analysis, we seek to identify the elements that significantly drive stock price increases or decreases. Additionally, by comparing models like Logistic Regression, LSTM, and Random Forest, we aim to determine which algorithms are most effective for predicting stock price movements. The project will also help refine our skills in classification-based modeling and machine learning techniques, enabling us to enhance our predictive capabilities within financial markets. Ultimately, we expect to develop a model that accurately predicts stock price movements while providing insights into the behavior of NVIDIA's stock about technological advancements.

**Risks**

This project faces several risks that must be carefully managed. One critical risk is overfitting, where the model may perform well on historical data but needs to generalize to new, unseen data. To address this, we will employ cross-validation techniques and avoid using overly complex models. Another concern is data quality; inaccuracies or missing data in the historical stock dataset could lead to flawed predictions. We will use thorough data preprocessing and validation steps to mitigate this risk and improve the model's reliability.

**Ethical Concerns**

There are also ethical considerations to keep in mind for this project. One significant concern is market manipulation. The developed predictive models could be used for unethical purposes, such as manipulating stock prices. While this project is for educational purposes, it is essential to consider the broader implications of sharing predictive models. Additionally, data privacy becomes another concern if we decide to integrate sentiment analysis from social media or financial reports. We must ensure that any external data sources comply with privacy regulations to avoid misuse of personal information.

**Contingency Plan**

Suppose the initial project fails to deliver the expected results. In that case, we will pivot to a predictive analytics project using an NCAA dataset containing information on sports participation, revenues, expenses, and enrollment at various institutions. The alternative project will focus on predicting the financial performance of sports programs based on participation numbers and other relevant factors. Similar preprocessing steps will be applied, such as handling missing data and feature engineering and testing models like linear Regression and random forest to forecast revenue and expenses. This pivot will allow us to apply the same machine-learning techniques in a different domain.

**Exploratory Data Analysis**

After diving into the NVIDIA stock data, we believe the data will be insufficient to answer our main questions by itself. The dataset gives us ample coverage of key variables, like daily prices and trading volumes, which should provide insights into how the stock fluctuates around major product launches. However, incorporating external factors, like major technological releases (e.g., RTX or GTX series) and market sentiment, could give us a clearer picture of how outside influences impact stock prices. Adding event markers for significant product releases and analyzing how prices and volumes behave before and after these launches will help us capture the direct effects of those events. Market sentiment data sourced from news articles or social media could further enhance our analysis by offering a deeper look into how public perception impacts price movements. Sentiment analysis using NLP techniques could quantify this data and give us a more nuanced understanding.

For visualizations, we still think line charts and bar charts will be useful for showing trends over time, but candlestick charts could offer a more detailed view of stock price movements, particularly around key dates like product launches. Additionally, we plan to generate a correlation heatmap to analyze the relationships between variables like volume, volatility, and price movements against the new sentiment data. We could also include metrics like daily price volatility (high vs. low) to better understand stock fluctuations and how they relate to major events.

While the data covers most of our needs, marking specific event dates for product launches will be a key adjustment to help us analyze how the stock behaves before and after these milestones. Along with this, we are considering creating lagged features (e.g., previous day’s close or volume) to account for time-dependent relationships in the data. This will help capture the typical lag effects seen in stock market behavior.

The models we picked (Logistic Regression, LSTM, and Random Forest) still feel like the right choice for predicting stock movements. However, we may tweak our evaluation metrics depending on how the data behaves. If we find a heavy imbalance (like mostly upward price movements), precision, recall, and F1 score might provide a more accurate reflection of model performance, rather than relying solely on accuracy.

Overall, we are still aligned with our original expectations. NVIDIA’s tech breakthroughs, especially the AI-driven RTX chips, seem to have a noticeable impact on stock prices. However, we need to ensure these events are truly driving price movements and not just coinciding with broader market or economic trends.

**Conclusion**

This project aims to predict NVIDIA's stock price movements by leveraging predictive analytics models, such as Logistic Regression, LSTM, and Random Forest Classification. By focusing on how technological innovations, like the release of NVIDIA's AI-powered RTX chips, have influenced investor behavior, we seek to uncover patterns in stock market reactions. The project will evaluate short- and long-term trends using key metrics such as accuracy, precision, recall, F1 score, and AUC-ROC. These evaluations will guide us toward identifying the most effective model for predicting stock price movements.

We have carefully considered risks such as overfitting and data quality, with plans for cross-validation and robust data preprocessing. Ethical concerns, especially around market manipulation and data privacy, are also addressed, ensuring the models remain responsibly used. Suppose the stock price prediction project does not yield satisfactory results. In that case, we are prepared to pivot to a backup project focused on predicting the financial performance of NCAA sports programs, applying similar machine learning techniques.

Overall, this project will enhance our understanding of stock market dynamics and technological advancements, refining our forecasting techniques and providing actionable insights for financial analysts and investors.

**Milestone 2**

**Introduction**

The semiconductor industry is crucial for pushing technology forward, with NVIDIA standing out as a major company in GPUs and AI hardware. Their innovations, especially with AI-focused chips, have significantly impacted everything from gaming to AI research. Launching new GPU generations, like the GTX and RTX series, has caused noticeable shifts in NVIDIA's stock price.

In this project, we will investigate how technological breakthroughs, particularly NVIDIA's AI-powered RTX chips, have influenced their stock price movement. Rather than predicting exact future prices, we will focus on whether the stock price will likely go up or down based on past data. We will also look at how big events, such as the release of the RTX series compared to the earlier GTX series, affect stock trends. By exploring these connections, our study will shed light on how tech advancements impact investor behavior and market reactions, offering valuable insights for financial analysts and those following the tech sector.

**Data**

The dataset for this project is centered on historical stock data for NVIDIA (NVDA). The data includes key stock market variables such as:

-          Date: The trading date.

-          Open: NVIDIA stock's opening price.

-          High: The highest price recorded during the trading day.

-          Low: The lowest price recorded during the trading day.

-          Close: The final price at the close of the trading day.

-          Adjusted Close: The closing price is adjusted for corporate actions, such as stock splits and dividends.

-          Volume: The number of shares traded on that day.

**Model Selection**

To predict NVIDIA's stock price movement, we will evaluate the following models:

-          Logistic Regression: This will classify price movements as increases or decreases based on historical data, serving as a baseline for binary classification.

-          LSTM (Long Short-Term Memory): This recurrent neural network will analyze historical trends to forecast future price direction, considering short-term and long-term patterns.

-          Random Forest Classification: This algorithm will handle complex interactions and non-linear relationships to predict stock price movements using features like historical prices, trading volume, and price volatility.

These models will be tested to assess their effectiveness in predicting NVIDIA's stock price direction, offering a range of traditional and advanced methods.

**Why these Models?**

We chose these models because of their unique strength in predicting stock price movements. Logistic Regression gives us a simple baseline for classifying price changes. LSTM is excellent at capturing long-term patterns and time-based trends through deep learning. Random Forest handles complex, non-linear relationships and interactions, like price volatility and trading volume. By testing these models, we can compare their performance and determine which approach works best for forecasting stock prices.

**Evaluation of Results**

We will use several metrics to evaluate and ensure accurate predictions of NVIDIA's stock price movements. Accuracy will measure the proportion of correct predictions for classification models like Logistic Regression and Random Forest. Precision and recall will assess how well the models predict upward or downward movements, with precision focusing on the correctness of predicted movements and recall on identifying actual movements. The F1 score will balance precision and recall, particularly for imbalanced classes. AUC-ROC will evaluate how well the models distinguish between price increases and decreases. Lastly, cross-validation will help prevent overfitting by assessing model performance across different data splits. These metrics will guide us in selecting the most effective model for predicting stock price directions.

**Learning Objectives**

This project aims to deepen our understanding of the factors influencing NVIDIA's stock price movements by analyzing fundamental variables such as trading volume, volatility, and major product releases. Through this analysis, we seek to identify the elements that significantly drive stock price increases or decreases. Additionally, by comparing models like Logistic Regression, LSTM, and Random Forest, we aim to determine which algorithms are most effective for predicting stock price movements. The project will also help refine our skills in classification-based modeling and machine learning techniques, enabling us to enhance our predictive capabilities within financial markets. Ultimately, we expect to develop a model that accurately predicts stock price movements while providing insights into the behavior of NVIDIA's stock about technological advancements.

**Risks**

This project faces several risks that must be carefully managed. One critical risk is overfitting, where the model may perform well on historical data but needs to generalize to new, unseen data. To address this, we will employ cross-validation techniques and avoid using overly complex models. Another concern is data quality; inaccuracies or missing data in the historical stock dataset could lead to flawed predictions. We will use thorough data preprocessing and validation steps to mitigate this risk and improve the model's reliability.

**Ethical Concerns**

There are also ethical considerations to keep in mind for this project. One significant concern is market manipulation. The developed predictive models could be used for unethical purposes, such as manipulating stock prices. While this project is for educational purposes, it is essential to consider the broader implications of sharing predictive models. Additionally, data privacy becomes another concern if we decide to integrate sentiment analysis from social media or financial reports. We must ensure that any external data sources comply with privacy regulations to avoid misuse of personal information.

**Contingency Plan**

Suppose the initial project fails to deliver the expected results. In that case, we will pivot to a predictive analytics project using an NCAA dataset containing information on sports participation, revenues, expenses, and enrollment at various institutions. The alternative project will focus on predicting the financial performance of sports programs based on participation numbers and other relevant factors. Similar preprocessing steps will be applied, such as handling missing data and feature engineering and testing models like linear Regression and random forest to forecast revenue and expenses. This pivot will allow us to apply the same machine-learning techniques in a different domain.

**Conclusion**

This project aims to predict NVIDIA's stock price movements by leveraging predictive analytics models, such as Logistic Regression, LSTM, and Random Forest Classification. By focusing on how technological innovations, like the release of NVIDIA's AI-powered RTX chips, have influenced investor behavior, we seek to uncover patterns in stock market reactions. The project will evaluate short- and long-term trends using key metrics such as accuracy, precision, recall, F1 score, and AUC-ROC. These evaluations will guide us toward identifying the most effective model for predicting stock price movements.

We have carefully considered risks such as overfitting and data quality, with plans for cross-validation and robust data preprocessing. Ethical concerns, especially around market manipulation and data privacy, are also addressed, ensuring the models remain responsibly used. Suppose the stock price prediction project does not yield satisfactory results. In that case, we are prepared to pivot to a backup project focused on predicting the financial performance of NCAA sports programs, applying similar machine learning techniques.

Overall, this project will enhance our understanding of stock market dynamics and technological advancements, refining our forecasting techniques and providing actionable insights for financial analysts and investors.

Sources

HRTERHRTER. (2024). \*Nvidia stock historical data\* [Data set]. Kaggle.

<https://www.kaggle.com/datasets/programmerrdai/nvidia-stock-historical-data>

*Nvidia announces financial results for second quarter fiscal 2025.* NVIDIA Newsroom. (2024,

August 29). <https://nvidianews.nvidia.com/news/nvidia-announces-financial-results-for->second-quarter-fiscal-2025