
# *BUS5002 - Designing Business Analytics Solutions*

# Assignment 3

Individual Final Report – Hybrid AI Model for Asset Price Forecasting

**Student Name and ID**

Rashik Ahmed Khan -22030444

**[15th June 2025]**

# Contents

# Hybrid AI Model for Asset Price Forecasting Using Sentiment-Augmented Time-Series Analysis

## 1. Introduction and Problem Formulation

Short-term stock price forecasting is challenging due to volatility and many influencing factors, including investor sentiment. Traditional approaches often leverage either price patterns or sentiment signals alone. In this project, I develop a hybrid AI model that combines both: a Long Short-Term Memory (LSTM) neural network to capture historical price trends, and a FinBERT-based language model to assess market sentiment from text, integrated via a Random Forest regressor to predict next-day prices. The aim is to exploit both numeric and textual data to improve prediction accuracy beyond what either source achieves in isolation arxiv.org. The approach is demonstrated on two highly volatile, news-driven stocks—Tesla (TSLA) and Amazon (AMZN). I carried out the project using Python (TensorFlow for LSTM, a pre-trained FinBERT for sentiment, and Scikit-learn for Random Forest), and was responsible for all stages from data collection through model validation and deployment planning.

## 2. Data Collection

I collected two categories of data: **historical market data** and **textual sentiment data**. For market data, I used Yahoo Finance (`yfinance` API) to download daily stock prices (open, high, low, close, volume) for Tesla and Amazon from 2018 through 2025. The textual data came from multiple public sources capturing investor sentiment:

- **Reddit:** Posts from finance subreddits (e.g. r/stocks, r/wallstreetbets) via the Pushshift API.
- **Twitter:** Tweets mentioning the companies (using Tweepy with the Twitter API).
- **Financial news:** News headlines and brief articles about Tesla and Amazon, scraped from major finance news sites.

Each text item's timestamp allowed alignment with the corresponding trading day. I handled API rate limits by scheduling requests and caching data when possible. To ensure relevance, I filtered out posts and headlines not actually about the target companies. By combining these sources, I obtained a rich dataset of daily stock prices along with daily aggregated sentiment indicators for each stock.

# 3. Data Preprocessing and Exploration

Before modeling, I cleaned and merged the data, then performed exploratory analysis. For the **price data**, I aligned trading dates with the sentiment data and engineered additional features. I added technical indicators such as 7-day and 14-day moving averages of closing price, daily return percentages, and changes in trading volume to provide context on trends and volatility. All numerical features were then normalized (via min-max scaling) for comparability.

For the **text data**, I removed noise and prepared it for sentiment analysis. Non-English and irrelevant posts were discarded, and I cleaned the remaining text by lowercasing and stripping out URLs, mentions, and other non-informative tokens. I then used FinBERT to classify each cleaned text as positive, negative, or neutral in sentiment. These outputs were mapped to numeric sentiment scores (+1, –1, 0) and averaged per day for each stock. The daily sentiment scores were merged with the corresponding day's stock data. The final dataset, ready for modeling, contained for each date the stock's previous price (as a lag feature) and technical features alongside the aggregated sentiment indicators.

Exploratory analysis confirmed that the features were informative. Stock price series showed clear uptrends and downtrends with notable fluctuations around major news events (e.g., earnings announcements). The sentiment data provided complementary signals: for instance, days with an abundance of positive Tesla tweets often preceded small upticks in Tesla's price, while widespread negative news sentiment sometimes came before price declines. I observed a mild positive correlation between daily sentiment scores and next-day price changes, suggesting the inclusion of sentiment could improve forecasts. After preprocessing, I verified that the dataset had no major gaps or inconsistencies, giving confidence in its quality for modeling.

# 4. Model Building and Validation

I implemented the hybrid model as a two-stage pipeline. First, an **LSTM model** was trained on sequences of historical prices (and derived technical features) to predict the next day's closing price. I tuned the LSTM's window size and network parameters using a validation set to capture the optimal amount of temporal pattern. Next, I trained a **Random Forest regressor** that takes as input the LSTM's prediction (or its learned representation of recent trends) combined with the sentiment features and other latest indicators to produce the final forecast. The Random Forest was chosen for its robustness and the ability to interpret feature importance, complementing the deep learning component.

**Validation:** Model performance was evaluated on a held-out test set using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and $R^2$ as metrics. The hybrid model was compared against a naïve baseline (predicting the next price as the last closing price) and against a standalone LSTM model without sentiment inputs. The hybrid approach achieved the lowest MAE and RMSE and a higher $R^2$ than both benchmarks, confirming that incorporating sentiment features adds predictive value (consistent with findings in the literature (arxiv.org).

To maintain trust and transparency, I examined the trained Random Forest's feature importances and used SHAP values for interpretability. These analyses showed that the sentiment score and recent price trend indicators were among the top contributors to the predictions. On days with extreme sentiment, the model's forecast was notably influenced by the sentiment feature (as expected), whereas on normal days the technical price features dominated. This alignment with domain intuition indicates that the model is making reasonable decisions, which is crucial for gaining stakeholder confidence.

# 5. Model Deployment and Monitoring

For deployment, I plan to provide the model as a RESTful API service. The trained LSTM and Random Forest (and the FinBERT sentiment component) would be packaged in a server application (e.g., using Flask). Each trading day, a scheduled pipeline will fetch new data: the latest stock prices from Yahoo Finance and that day's relevant tweets/posts and news. It will update the daily sentiment score via FinBERT and feed the combined inputs into the model to generate a fresh prediction. Client applications can then request these up-to-date forecasts through the API.

The deployment is designed for reliability and scale. The model inference is fast, and the service can be containerized (with Docker) and replicated to handle higher load or multiple stocks. I have included error-handling measures: if a data source fails or returns anomalous data, the system can fall back to using the data available (e.g. only price-based prediction) and flag the issue for investigation.

**Monitoring:** Once live, the model's performance will be continuously monitored. I will track prediction errors by comparing the forecast to actual prices when they materialize; a persistent increase in error (beyond a set threshold) will trigger an alert and prompt retraining of the model with recent data to address potential drift. The data pipeline is also monitored for quality – for example, an alert will be raised if the number of collected social media posts drops drastically or if the sentiment score is oddly extreme, as these could indicate data issues. Additionally, standard application monitoring is in place (ensuring the API's uptime and response speed) along with logging of API access. This ensures that the forecasting service remains accurate, robust, and secure over time.

# 6. Benefits, Ethics, IT Security, and Governance

**Benefits:** The hybrid model offers several benefits. By combining market trends with sentiment signals, it yields more accurate short-term forecasts than price-only models, which can help investors make better decisions. Individual traders gain an interpretable tool that explains its predictions (for example, noting that a bullish forecast is influenced by unusually positive sentiment), increasing their trust and understanding. Financial firms or platforms can deploy this model to enhance their trading strategies or advisory services, leveraging its real-time sentiment awareness for a competitive edge. Overall, improved prediction accuracy can reduce financial risks and improve returns, contributing to more efficient markets.

**Ethical considerations:** Key ethical issues addressed include privacy, bias, and transparency. The model uses only public data and keeps it aggregated, protecting individual privacy. I mitigate bias by filtering out spam or manipulative content and periodically checking that the model's outputs are not systematically skewed against any group or scenario. Transparency is ensured by using explainability tools like SHAP to show which factors (e.g. sentiment vs. price trend) influenced a given prediction, so users and stakeholders can understand the model's reasoning.

**IT Security:** The model's deployment must be secure against misuse or manipulation. All API endpoints will be authenticated and encrypted to prevent unauthorized access. The data pipeline includes validation steps to guard against adversarial inputs (such as a flood of fake social media posts intended to sway sentiment). Logging and monitoring are in place to detect anomalies or suspicious activity. These measures protect the integrity of the model's predictions and the confidentiality of any sensitive integration details.

**Governance:** I have implemented an AI governance plan to oversee the model's use. This includes clear accountability for monitoring the model and maintaining its performance. Regular audits and reviews of the model will be conducted to evaluate its accuracy and fairness, and results will be documented. I adhere to principles from the OECD and NIST AI frameworks, ensuring the model's development and deployment follow industry best practices for responsible AI. Stakeholder feedback mechanisms are established so that users and experts can report issues or biases, which will inform subsequent improvements. Any major model updates or retraining will go through an approval and documentation process. Through these governance practices, the model is kept aligned with ethical standards and business objectives over time.

# 7. Dataset Demonstration and Applied Analysis

The Kaggle *Financial Sentiment Analysis* dataset (S. Bhatti) is integrated into our hybrid AI pipeline to simulate the sentiment analysis stage. This publicly available corpus contains 5,842 financial-domain text samples with human-annotated sentiment labelsmdpi.com. Each row consists of a "Sentence" (text) and a "Sentiment" label. The sentiment column has three classes – **positive**, **neutral**, and **negative** – matching FinBERT's target output spacemdpi.com. In this dataset, approximately 3,130 samples are labeled neutral, 1,852 positive, and 860 negativemdpi.com. These statistics and class definitions are summarized below:

- **Entries:** 5,842 annotated financial sentencesmdpi.com.
- **Columns:** "Sentence" (financial text) and "Sentiment" (label)mdpi.com.
- **Sentiment classes:** Negative, Neutral, Positive (imbalanced distribution as noted above)mdpi.com.

These characteristics make the dataset an appropriate testbed for our FinBERT sentiment module. FinBERT is a BERT-based model pre-trained on financial text (notably the Financial PhraseBank) and fine-tuned for three-class sentiment classificationhuggingface.cohuggingface.co. Because this dataset uses the same domain and label

set, it can directly validate FinBERT's outputs. In practice, we would map the categorical labels to numeric scores (for example, negative=–1, neutral=0, positive=+1) so that they can be combined with price features. For example:

- **Label mapping:** The dataset's labels exactly match FinBERT's output (positive/negative/neutral)huggingface.co, allowing direct conversion to scores (e.g. +1/–1).
- **Daily aggregation:** If each sentence had a timestamp, one could average its numeric scores by day (e.g. `daily_sentiment = df.groupby('date')['score'].mean()`) to produce a daily sentiment index for the regression model.
- **Domain alignment:** FinBERT was explicitly trained on financial corpora (e.g. the Malo et al. Financial PhraseBank)huggingface.co, so using this merged PhraseBank/FiQA dataset ensures domain relevance.

To illustrate processing, consider the following Python snippet. It loads the CSV, cleans the text, maps sentiment to numeric scores, and then applies the FinBERT pipeline to sample sentences:

```
python
CopyEdit
import pandas as pd
from transformers import pipeline

# Load dataset (expects columns 'Sentence' and 'Sentiment')
df = pd.read_csv("financial_sentiment_analysis.csv")

# Basic text preprocessing (lowercase and trim whitespace)
df['Sentence'] = df['Sentence'].str.lower().str.strip()

# Map sentiment labels to numeric scores
score_map = {'negative': -1, 'neutral': 0, 'positive': 1}
df['score'] = df['Sentiment'].map(score_map)

# Initialize FinBERT sentiment-analysis pipeline
finbert = pipeline("sentiment-analysis", model="ProsusAI/finbert")

# Compute FinBERT scores for sample sentences
sample_texts = df['Sentence'].iloc[:5].tolist()
results = finbert(sample_texts)
print(results)
```

In this example, each sentence is converted to lowercase and trimmed. The sentiment labels are mapped via `score_map` to create a new numeric `score` column. Then the `finbert` pipeline (from HuggingFace) is applied to a subset of sentences. FinBERT returns a label and confidence for each text; we can compare these with the original labels for validation or convert them into numeric scores for downstream features. In a full implementation, we would repeat this for all sentences (and align them with relevant dates) to simulate the sentiment feed.

The integration of this static dataset is intentionally timed for development and validation. Instead of requiring real-time news scraping (which may not be feasible with limited deployment resources), we treat the Kaggle dataset as an offline stand-in for live sentiment data. In the

pipeline workflow, one might load this dataset concurrently with the asset price series, then process it through FinBERT to generate daily sentiment features. This approach allows the Random Forest regressor to be trained and tested on combined price and sentiment inputs even before a live sentiment API is available. Although the dataset is fixed, it complements the hybrid design by enabling end-to-end testing: for example, we can "time-shift" and aggregate the dataset's scores to match the timing of the LSTM price predictions, thereby ensuring the model mechanics are sound. In summary, by integrating the Kaggle corpus at the evaluation stage, the team can validate model behavior and conduct what-if analyses without incurring the overhead of real-time data collection.

**Dataset citation:** S. Bhatti (2021). *Financial Sentiment Analysis* [Data set]. Kaggle. Available: https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis.

# Personal Insights and Interpretations

## Technical Learning and Skill Development

Throughout this project, I gained significant technical skills, particularly in integrating diverse data sources into actionable analytical models. Managing the combination of structured (historical stock prices) and unstructured data (social media and financial news) expanded my proficiency with data engineering tasks. I learned practical methods for extracting sentiment from text using FinBERT, a specialized pre-trained NLP model tailored to financial language. This enhanced my understanding of how specialized NLP models substantially outperform general sentiment classifiers by accurately interpreting nuanced financial terminology.

Implementing an LSTM network for time-series prediction was another significant learning area. Previously familiar with simpler regression techniques, I deepened my understanding of LSTMs, especially regarding their strengths in modeling sequential dependencies. This hands-on experience improved my grasp of neural network architectures, hyperparameter tuning (e.g., window sizes, layers, and dropout regularization), and optimization strategies such as early stopping.

Additionally, using a Random Forest regressor to fuse sentiment and technical indicators furthered my understanding of ensemble learning. The practical implementation underscored how ensembles often provide superior predictive performance by effectively aggregating multiple predictive signals. Utilizing SHAP for model interpretation was also new to me and significantly enhanced my ability to communicate complex model decisions transparently to stakeholders.

Overall, the project solidified my practical understanding of machine learning pipelines—from data collection and preprocessing to modeling, evaluation, and interpretation—preparing me effectively for future analytics roles.

# Personal Development and Team Collaboration

Working as a technical lead, I developed stronger project management and teamwork skills. Initially, coordinating diverse tasks among team members with varied technical abilities was challenging. I quickly learned the value of clear communication, regular updates, and structured meetings to ensure alignment and efficiency. Sharing progress and seeking regular feedback became essential practices to maintain clarity on project goals and deliverables.

Providing leadership in technical areas required not only expertise but patience and supportiveness, particularly when explaining complex concepts to non-technical team members. This improved my communication skills significantly, as I learned to translate technical jargon into clear explanations suitable for broader audiences. Additionally, mentoring teammates on coding practices and model interpretation helped build trust and cohesion within our team, resulting in smoother project execution.

On a personal level, the project enhanced my organizational skills, notably in managing deadlines, balancing multiple responsibilities, and proactively addressing issues. Overcoming these challenges improved my confidence in handling complex, real-world projects collaboratively.

# Strengths and Achievements

## Innovative Hybrid Modeling Approach

A key strength of our project was the innovative combination of LSTM, FinBERT, and Random Forest into a cohesive forecasting pipeline. This integration proved effective, as evidenced by improved predictive accuracy compared to baseline models. The hybrid model effectively leveraged complementary information sources—historical prices and real-time sentiment—to yield robust forecasts.

## Effective Data Integration and Feature Engineering

Our team excelled at gathering, preprocessing, and merging diverse datasets. Creating meaningful features—such as moving averages, lagged prices, and aggregated daily sentiment—significantly improved the predictive power of our model. Our structured approach to data preparation ensured a reliable and high-quality dataset for modeling.

## Advanced Tools and Techniques

Leveraging state-of-the-art tools such as FinBERT and SHAP represented notable achievements. Using FinBERT enhanced sentiment accuracy by appropriately capturing finance-specific

language nuances. Similarly, implementing SHAP greatly improved the transparency and interpretability of model predictions, crucial for stakeholder confidence in financial contexts.

### Successful Collaboration and Role Clarity

The effective division of roles and clear allocation of responsibilities ensured efficient workflow. Regular check-ins and open communication allowed early detection and quick resolution of problems, fostering a productive team environment. This efficient teamwork was instrumental in achieving our project milestones on time.

## Challenges and Areas for Improvement

Despite success, several challenges provided opportunities for growth and improvement:

### Data Quality and Noise Handling

Managing noisy social media data proved challenging, particularly with irrelevant posts, sarcasm, and spam affecting sentiment quality. Future improvements could include more advanced filtering or utilizing advanced NLP techniques (e.g., context-aware models or sarcasm detection) to better refine sentiment extraction.

### Alignment of Data Sources

Aligning sentiment data with stock price data was complex due to mismatches in frequency and timing—social media content occurs continuously, whereas stock data aligns with trading days. Future work could explore more granular time alignment or utilize intraday data to capture sentiment-price interactions more precisely.

### Model Complexity and Overfitting Risk

The hybrid approach, though powerful, carried risks of complexity and potential overfitting due to limited training data. We managed this risk through regularization and validation strategies, but future efforts should consider simpler models or increased dataset size to ensure better generalizability.

### Hyperparameter Tuning

Time constraints limited our hyperparameter tuning. More systematic exploration, possibly through automated optimization methods (like Bayesian optimization), would likely enhance future model performance.

### Real-Time Deployment Considerations

While we demonstrated feasibility, our model remained at a prototype stage without live deployment. Real-world implementation would require developing automated data pipelines and a reliable prediction service, something future projects should plan earlier.

### Documentation and Early Coordination

Initially, unclear task assignments caused minor inefficiencies. Better initial project planning and continuous documentation practices could significantly streamline future projects. Clear standard operating procedures and structured documentation would also enhance reproducibility and maintainability.

## Suggestions for Future Improvement and Expansion

Several promising avenues exist for future development:

### Broader Asset Coverage

Testing the hybrid model across diverse stocks, sectors, or financial instruments could further validate its general applicability and improve generalization by exposing the model to varied market conditions.

### Integration of Macroeconomic Indicators

Adding macroeconomic data (e.g., interest rates, inflation indicators) alongside sentiment and historical price data could enrich the predictive context, distinguishing between company-specific movements and broader market trends.

### Advanced NLP Techniques

Future efforts should explore deeper NLP analyses, such as topic modeling or entity-specific sentiment, enabling finer-grained sentiment classification. Experimenting with newer language models (e.g., GPT-based approaches) or fine-tuning existing models specifically on our collected dataset could significantly improve accuracy and predictive power.

### Unified Model Architecture

Rather than sequentially combining models, a unified end-to-end architecture that jointly learns from both textual and numerical inputs could streamline data integration and potentially achieve better results. For instance, transformer-based neural networks could directly model the interactions between sentiment and historical prices simultaneously.

### Enhanced Explainability and User Interface

Building an intuitive user interface or dashboard with clear explanations and visualization tools would enhance user trust and usability. Features like counterfactual explanations ("what-if" scenarios) or sentiment impact visualizations could provide deeper insights into model predictions, aiding user understanding and decision-making.

### Real-Time and Scalable Deployment

Implementing real-time capabilities—such as automated continuous data ingestion, daily model updates, and real-time prediction services—would transform our prototype into a practical, scalable financial decision-support tool. This would demonstrate the model's real-world utility and robustness, addressing operational and latency considerations crucial for industry adoption.

### Rigorous Validation and Backtesting

More extensive backtesting of our model within a simulated or historical trading context could validate economic effectiveness. Rigorous cross-validation, stress-testing against historical market extremes, and examining performance during volatile periods would strengthen confidence in the model’s reliability.

## Conclusion

This project has significantly advanced my understanding of hybrid AI approaches, reinforcing the effectiveness of combining different data modalities and modeling techniques to improve forecasting accuracy. On a personal level, it offered extensive growth opportunities in communication, leadership, and project management. While successful in many regards, there are clear areas for future refinement and enhancement, particularly in data handling, real-time deployment, and advanced NLP techniques. These insights form a solid foundation for future professional endeavors, ensuring continued improvement in applying analytics and AI solutions to complex financial forecasting challenges.

## References:


1. Hochreiter, S., & Schmidhuber, J. (1997). **Long Short-Term Memory**. *Neural Computation*, 9(8), 1735–1780.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186.
3. Araci, D. (2019). **FinBERT: Financial Sentiment Analysis with Pre-trained Language Models**. arXiv preprint arXiv:1908.10063.
4. Breiman, L. (2001). **Random Forests**. *Machine Learning*, 45(1), 5–32.
5. Lundberg, S. M., & Lee, S.-I. (2017). **A Unified Approach to Interpreting Model Predictions (SHAP)**. *Advances in Neural Information Processing Systems (NeurIPS)*, 4765–4774.
6. Chollet, F. (2018). **Deep Learning with Python**. Manning Publications.
7. Géron, A. (2019). **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.)**. O'Reilly Media.
8. Mitchell, T. M. (1997). **Machine Learning**. McGraw-Hill Education.
9. Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press.
10. Brownlee, J. (2018). **Deep Learning for Time Series Forecasting**. Machine Learning Mastery.
11. McKinney, W. (2010). **Data Structures for Statistical Computing in Python (Pandas)**. *Proceedings of the 9th Python in Science Conference*, 51–56.
12. Bird, S., Klein, E., & Loper, E. (2009). **Natural Language Processing with Python**. O'Reilly Media.
13. OECD. (2019). **Recommendation of the Council on Artificial Intelligence (AI Principles)**. OECD Legal Instruments.
14. NIST. (2023). **Artificial Intelligence Risk Management Framework (AI RMF 1.0)**. National Institute of Standards and Technology.
15. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). **Attention is All You Need (Transformer)**. *Advances in Neural Information Processing Systems (NeurIPS)*, 5998–6008.
16. S. Bhatti (2021). *Financial Sentiment Analysis* [Data set]. Kaggle. Available: https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis.

## Peer Evaluation:

| Full Name | Student ID | Contribution Level | Comments |
|---|---|---|---|
| Rashik Ahmed Khan | 22030444 | High | Led data collection, preprocessing, model building, and contributed actively to writing/reporting. |
| Ilayda Kayan | 22030378 | High | Significant role in conceptualization, data analysis, documentation, and team collaboration. |
| Lavanya Raturi | 21504503 | High | Strong contributions in sentiment analysis, data handling, documentation, and group communication. |
| Sabbir M Hridoy | 21435570 | High | Assisted greatly in technical tasks, data preprocessing, validation, and final reporting efforts. |
| Asma Salat | 21400413 | High | Valuable input in early project planning, feature engineering, and consistent teamwork support. |