Forecasting Stock Prices

Team: Michael Wallison (CIS 4190), Kaihao Liu (CIS 5190), Jiwon Kang (CIS 4190)
Project Mentor TA: Anusha Srikanthan

0) Code
Project folder: https://drive.google.com/drive/folders/1Rw-JZ7GJnul2L3P55PKYTgmM-VdkqzMY?usp=sharing

1) Abstract

In this project, we investigate the prediction of stock prices, a crucial task for investors and financial institutions in optimizing their investment strategies. We employ and evaluate various models and explore different ensemble and voting methods to enhance prediction accuracy. Prior work on the Kaggle dataset has used stacking to combine LSTM and GRU models; however, we aim to further ensemble other models to develop a more accurate prediction model for the test dataset. Our empirical results reveal that the most effective approach for stock price prediction is a standalone ARIMA model, outperforming other models and ensembles in terms of accuracy and reliability.

2) Introduction

In this project, we focus on predicting stock prices using machine learning models. Our primary dataset comprises the stock price data of Reliance Industries Ltd across 249 days (249 rows x 7 columns), with features such as Date, Open, High, Low, Close, Adj Close, and Volume. To corroborate the validity of the models employed, we use a secondary, larger dataset containing stock price data of Amazon.com Inc. The inputs to our machine learning models are the closing prices, while the outputs are the predicted stock prices.

Our implementation contributions involve utilizing various machine learning models, including Support Vector Regression (SVR), Random Forest (RF), and K-Nearest Neighbors (KNN). In addition, we also implemented time series models including Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA). We explore different ensemble and voting techniques, such as SVR + RF, ARIMA + GRU.

Evaluation: For our evaluation contributions, we assess the performance of each individual model and the ensemble combinations on both the primary dataset (Reliance Industries Ltd) and the secondary dataset (Amazon.com Inc), using the training and testing RMSE. This evaluation allows us to compare the accuracy and reliability of different models and ensembles, ultimately identifying the most effective approach for stock price prediction. Among our ensemble models, the voting-based ensemble comprising (ARIMA + LSTM) and (ARIMA + GRU) models delivers the best performance. However, the model with the best prediction accuracy and reliability was the standalone ARIMA model. The model with the worst prediction accuracy and reliability was the K-Nearest Neighbors model.

2) How We Have Addressed Feedback From the Proposal Evaluations

After receiving feedback on our project proposal and working closely with our mentor, we have significantly revised our project plans. We recognized that our initial plan lacked critical details, so we conducted a thorough analysis of various ensemble techniques to determine the most effective way to combine models for predicting stock prices. Moreover, we intend to integrate statistical time series models and the FB prophet model. Our objective is to provide a comprehensive evaluation of different models and ensembles, including the proposed approach, and to explain the reasons for the observed results, if possible.

3) Background

Our project builds upon the prior work of "Advanced Stock Pred using SVR, RFR, KNN, LSTM, GRU," which uses multiple machine learning algorithms for stock prediction. However, this study has some limitations, such as a small dataset and no time series model for stock data, which is a time series data. We also refer to the study "A Comparative Analysis of Machine Learning Algorithms in Stock Prediction," which compares the performance of several algorithms, including Linear Regression, KNN, FB Prophet, and LSTM, in predicting stock prices.

- Title: Advanced Stock Pred using SVR, RFR, KNN, LSTM, GRU
  - URL: https://www.kaggle.com/code/ysthehurricane/advanced-stock-pred-using-svr-rfr-knn-lstm-gru/notebook
- Title: A Comparative Analysis of Machine Learning Algorithms in Stock Prediction
  - URL: http://ieomsociety.org/proceedings/2021rome/800.pdf

4) Summary of Our Contributions

Our primary implementation contribution involves extending prior work by incorporating multiple models, with a particular focus on time series models, which are essential for capturing temporal dependencies and trends in financial data. We introduced ARIMA and SARIMA models and developed a novel approach for their prediction, overcoming the challenge of dividing time series data into training and testing sets. To further validate our findings, we applied the models to a larger and more popular stock dataset, Amazon.com Inc. Moreover, we conducted a thorough analysis, highlighting the reasons why time series models can outperform machine learning and neural network models in stock price prediction.

For evaluation contribution, we enhanced the assessment of model performance by introducing visualizations of evaluation metrics, allowing for more effective comparisons between models. We tested the models on both the primary dataset (Reliance Industries Ltd) and the secondary dataset (Amazon.com Inc). While prior work employed a variety of evaluation metrics without visual aids, we focused on selecting the most relevant ones, such as Root Mean Squared Error (RMSE), and provided visualizations to facilitate a comprehensive understanding of each model's performance.

5) Detailed Description of Contributions

5.1 Implementation Contributions

ARIMA and SARIMA

ARIMA (Autoregressive Integrated Moving Average) is a statistical modeling technique used to analyze and forecast univariate time series data. The model combines three components: Autoregression (AR), Differencing (I), and Moving Average (MA), which are represented by p, d, and q.

$$(1-\sum_{i=1}^{p}\phi_i L^i)(1-L)^d X_t = (1+\sum_{i=1}^{q}\theta_i L^i)\varepsilon_t$$

SARIMA (Seasonal Autoregressive Integrated Moving Average) is an extension of the ARIMA model that specifically addresses seasonality in time series data. It combines the ARIMA model components (Autoregression, Differencing, and Moving Average) with seasonal components to capture both the underlying patterns and the seasonality in the data.
The SARIMA model is specified by the parameters (p, d, q) for the non-seasonal ARIMA components and (P, D, Q, s) for the seasonal components.

$$\underbrace{(1-\phi_1 B)}_{\text{Non-seasonal AR(1)}}\ \underbrace{(1-\Phi_1 B^4)}_{\text{Seasonal AR(1)}}\ \underbrace{(1-B)}_{\text{Non-seasonal difference}}\ \underbrace{(1-B^4)}_{\text{Seasonal difference}}y_t \;=\; \underbrace{(1+\theta_1 B)}_{\text{Non-seasonal MA(1)}}\ \underbrace{(1+\Theta_1 B^4)}_{\text{Seasonal MA(1)}}e_t.$$

Our initial implementation of the ARIMA model using separate training and testing data yielded unsatisfactory results. We realized that by only using a subset of the available data, the model was unable to capture the full range of variation in the data and failed to generalize well to new, unseen data.

To address this issue, we adopted the rolling forecast method, also known as walk-forward validation. This approach fits an ARIMA model with the specified order for each data point in both the training and testing sets and makes one-step-ahead forecasts for each data point by fitting the model on all available historical data up to that point. The rolling forecast method is well-suited for time series data because it:

Preserves temporal order by using all available historical data up to the current point.
Adapts to changing dynamics by iteratively refitting the model for each new data point.
Focuses on one-step-ahead forecasting, a common objective in time series applications.
Provides a realistic and robust evaluation of the model's performance on unseen data.

Hence, we also employed this method for the SARIMA model to ensure accurate predictions and evaluations. We explored various combinations of p, d, and q values ranging from 0 to 3 to identify the best ARIMA model. The optimal model parameters were found to be (1,1,1).

Figure A: ARIMA Model for Reliance Data

After tuning the model, we picked the final parameters to be order = (1, 1, 1), seasonal_order = (0, 1, 1, 2).

Figure B: SARIMA Model for Reliance Data

FB Prophet

Facebook Prophet is a forecasting method that combines additive decomposition and state space modeling. The underlying model can be represented by the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

where:

1. y(t) is the predicted value at time t.
2. g(t) represents the trend component, which models non-periodic changes in the data.
3. s(t) is the seasonal component, accounting for periodic fluctuations (such as daily, weekly, and yearly patterns).
4. h(t) denotes the holiday component, capturing the effects of holidays and other special events.
5. epsilon_t is the error term, representing the unexplained variation in the data.

During our implementation of the FB Prophet model, we encountered several difficulties. We experimented with various combinations of parameters and evaluated the model's performance using MSE and MAE. However, the results showed a high MSE, indicating that the model was not performing well.

We attempted to optimize the hyperparameters.

- daily_seasonality = False
- weekly_seasonality = True
- yearly_seasonality = True

The above combination can produce a reasonable shape, but we still encountered high MSE values. These challenges suggest that the FB Prophet model may not be the best fit for our stock prediction task.

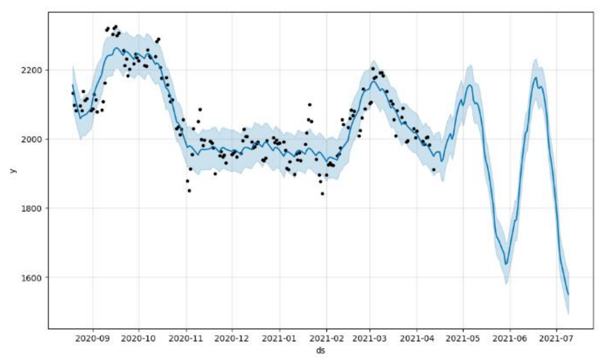


Figure C, D: FB prophet for Reliance Data

Overall, implementing the FB Prophet model for predicting stock prices was very challenging. As we did not achieve reasonably good performance, we excluded it from our final evaluation of performance across the many models we tested. However, we were still able to learn about the model's uses and limitations.

Ensembles

After thoroughly analyzing the performance of various models, we decided to combine ARIMA with GRU and LSTM, as ARIMA demonstrated the best performance for our dataset. In order to create a more accurate ensemble model, we opted for a voting approach.

The rationale behind this choice is that combining the strengths of the top-performing time series model (ARIMA) with the capabilities of deep learning models (GRU and LSTM) can potentially lead to improved predictions. The voting ensemble method allows each model to contribute its predictions, and the final forecast is determined by aggregating these predictions.

By incorporating the expertise of different models in this way, we aim to create a robust and accurate ensemble model for stock price prediction.

We chose to combine ARIMA with LSTM and GRU using a voting ensemble method because the ARIMA model had the best performance in predicting stock prices compared to LSTM and GRU. We did not use other ensemble methods for ARIMA because the rolling forecast method used in ARIMA creates a different model structure that makes it difficult to use other ensemble methods such as bagging or boosting.

We also combined SVR + RF for comparison purposes.

Figure E: ARIMA + GRU

Figure F: ARIMA + LSTM

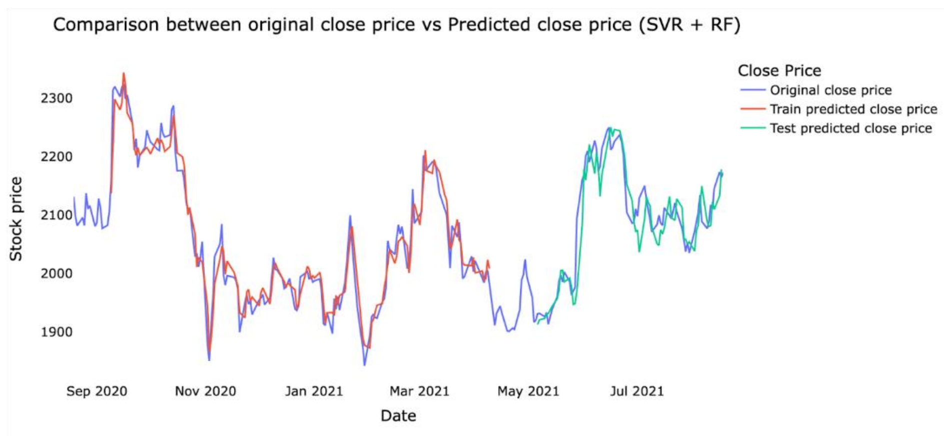


Figure G: SVR + RF

5.2 Evaluation Contribution

For dataset shifts, we opted to use the rolling forecast method for ARIMA and SARIMA models, as separate training and testing data had produced unsatisfactory results. We also experimented with different time periods for the datasets to test the robustness of the models. However, we did not explicitly explore distribution shifts as our focus was on identifying the most effective model and ensemble method for stock price prediction.

Figure H: ARIMA result with separate training and testing data

As the ARIMA and SARIMA models were a key component of our implementation contributions, our experiments and evaluation metrics were designed to answer the key question of whether and by how much these new models–ARIMA and SARIMA–performed better than the rest of the models in terms of prediction accuracy and reliability.

In order to answer this question, we tested the performance of all of the models as measured by the Root Mean Squared Error on both the training and testing stock close data for both Reliance Industries Ltd and Amazon Inc.

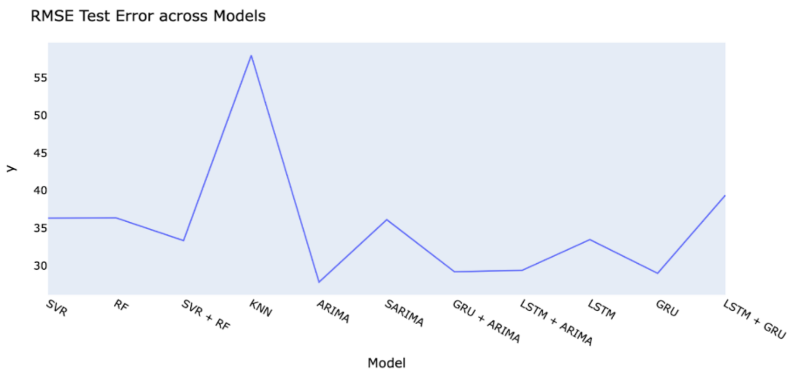


Figure I: Test RMSE Error Across Models for Reliance Industries Ltd

Figure J: Test RMSE Error Across Models for Amazon Inc

According to the figures summarizing our evaluation metrics, the ARIMA model outperformed all other models, including the ensembles, with the lowest testing RMSE for both datasets. The consistently superior performance of ARIMA suggests that it is a reliable and accurate predictor of time-series data. In contrast, the SARIMA model performed poorly in comparison to the ARIMA model, ranking 7th in prediction accuracy on the testing data for the Reliance Industries Ltd dataset.

As for the LSTM and GRU models, our results confirmed their known differences, with GRU outperforming LSTM in the Reliance Industries Ltd dataset, while LSTM outperformed GRU in the Amazon Inc dataset. This aligns with the idea that LSTM is better suited for larger datasets, while GRU is more effective with shorter datasets. On the other hand, models such as RFR, KNN, and SVR assume independence between data points and do not capture the temporal nature of stock price data, resulting in poor performance and inaccurate predictions.

It is worth noting that we encountered difficulties when implementing the FB Prophet model, and it produced very high MSE values, indicating poor performance. This, along with the consistently better performance of ARIMA, led us to focus on ARIMA for our ensemble approach, using voting to combine it with the LSTM and GRU models. Overall, our analysis suggests that ARIMA is a strong choice for stock price prediction, particularly for datasets with a significant temporal component.

The following table describes the training and testing RMSE for both companies.

	Reliance Training RMSE	Reliance Testing RMSE***	Amazon Training RMSE	Amazon Testing RMSE
ARIMA	44.302	27.845 (best)	2.1837	3.5975 (best)
GRU	41.676	29.019	2.0292	4.1628
GRU + ARIMA	43.949	29.223	2.1409	3.7007
LSTM + ARIMA	43.473	29.408	2.1173	3.6937
SVR + RF	25.055	33.359	3.3336	7.3566
LSTM	39.126	33.493	2.1472	3.7405
SARIMA	52.816 (worst)	36.138	2.5275	4.4062
SVR	37.275	36.343	6.3962 (worst)	8.6212
RF	15.944 (best)	36.386	0.78071 (best)	6.9115
LSTM + GRU	42.681	39.383	1.9645	3.6817
KNN	48.428	57.961 (worst)	2.4643	9.7223 (worst)

As shown in the table above, all models performed better on the Amazon Inc dataset compared to the Reliance Industries Ltd dataset. This could be since the Amazon Inc dataset is larger and less volatile compared to the Reliance Industries Ltd dataset.

The larger dataset allows for more historical data to be used in training the model, which in turn can improve the model's accuracy. Additionally, the more stable and consistent market of the Amazon Inc dataset may have contributed to the better performance of the models. Conversely, the Reliance Industries Ltd dataset may have had greater fluctuations in price, which could have made it more difficult for the models to accurately predict future prices.

6) Compute/Other Resources Used

We referred to the book "Forecasting: Principles and Practice (2nd ed)" by Rob Hyndman and George Athanasopoulos for guidance on implementing ARIMA and SARIMA models. We also obtained the stock market data, including the Amazon data, from the Kaggle datasets repository: https://www.kaggle.com/datasets/paultimothymooney/stock-market-data

7) Conclusions We think ARIMA is the best model for stock prediction because it is specifically designed for time series data and can capture the linear relationships between variables. Additionally, ARIMA can handle non-stationary time series data by differentiating the data, which makes it more suitable for stock price prediction than non-time series models such as LSTM and GRU.

The ensemble of ARIMA with LSTM and GRU did not outperform ARIMA alone because these models focus more on capturing non-linear relationships, which may not be as relevant in the context of stock price prediction. Furthermore, the ensemble of ARIMA with LSTM and GRU may introduce more complexity and overfitting, leading to worse performance.

In hindsight, we encountered a few roadblocks during the project. One was that the Amazon data we collected was too large to execute, so we had to choose a shorter time period starting from 2017 March 1st.

Another was that implementing FB Prophet was harder than expected due to an installation problem. We also found that the model did not perform as well as we had hoped, despite trying various combinations of hyperparameters. Additionally, feature engineering was outside the scope of our project, and as we worked, we focused more on the performance of the ensemble methods specifically. Overall, the project idea evolved as we worked on it, with changes made to the dataset and modeling techniques used.

For future work, we recommend exploring more advanced ensemble methods and incorporating more fundamental analysis indicators. In terms of ethical considerations, the ability to develop models that can accurately predict stock prices could have significant impacts on the financial market and the wider economy. Therefore, we would ensure that our methods for producing these models are transparent and unbiased. We would work to ensure that no preexisting inequalities are perpetuated by our models. Additionally, scaling up this work to predict models for more datasets (that would be updated daily with the stock market) could cost very significant computational resources.

8) Roles of team members (1-2 sentences each):
Michael Wallison (CIS 4190): extensively explored the FB Prophet model, conducted hyperparameter tuning, and reviewed relevant literature to evaluate the model's effectiveness in comparison to other models.
Kaihao Liu (CIS 5190): established a code base for our project. Studied and implemented ARIMA, SARIMA, and their relevant ensembles, for both the Amazon Inc and Reliance Industries Ltd datasets.
Jiwon Kang (CIS 4190): worked on other ensemble models. Evaluated and compared models and created data visualizations to gain insights into the behavior of the different models on the stock price data.

Citation

Hyndman, Rob J, and George Athanasopoulos. "Forecasting: Principles and Practice (2nd Ed)." *8.9 Seasonal ARIMA Models*, https://otexts.com/fpp2/seasonal-arima.html.

Lendave, Vijaysinh. "LSTM vs GRU in Recurrent Neural Network: A Comparative Study." *Analytics India Magazine*, 14 Dec. 2021, https://analyticsindiamag.com/lstm-vs-gru-in-recurrent-neural-network-a-comparative-study/.

Midway report

Project Midway Report (1 submission per group, add all members in your group)

Graded

Group

Kaihao Liu
Jiwon Kang
Michael Wallison
View or edit group

Total Points

7 / 7 pts

Question 1

Evaluation Question [select all pages] — 7 / 7 pts

✓ + 1 pt	Does the report follow the provided template including the 4-page limit (excluding exempted portions), with reasonable responses to all questions?
✓ + 2 pts	Has feedback from the last round been effectively addressed?
✓ + 1 pt	Has the team identified a clear topic and viable new target contribution, as per the project specifications provided in class?
✓ + 1 pt	Has the team moved in a non-trivial way towards their target contribution?
✓ + 2 pts	Has a clear and systematic work plan been formulated for the remaining weeks?

Overall, the project report addressed most of the previous feedback and has been well drafted. The study proposes focusing on evaluating the performance of ensemble methods on stock predictions and also plans to use ARIMA models to incorporate time series analysis. There has been significant progress towards target contribution. The only suggestion is to focus a bit more on writing an introduction that captures the overall motivation. Drop by my office hours or email for clarifications.

Forecasting Stock Prices

Team: Michael Wallison (CIS 4190), Kaihao Liu (CIS 5190), Jiwon Kang (CIS 4190)
Project Mentor TA: Anusha

1) Introduction

Set up the problem:
We will be training on the stock price data across 249 days for Reliance Industries Ltd (250 rows x 7 columns) whose features are Date (the day of interest), Open (the opening price of that stock), High (the highest intraday stock price), Low (the lowest intraday stock price), Close (the closing stock price), Adj Close (the adjusted closing stock price), and Volume (the number of shares traded).

Although the prior work associated with the Kaggle dataset has used the one ensemble method of stacking in combining the LSTM and GRU models into one sequential neural network model, we aim to ensemble the models further using a variety of methods such as bagging, boosting, and stacking in order to arrive at a model that is more accurately predicts on the test dataset than the models in the prior work.

This is a clear topic, well done. One additional suggestion would be to specify the input and output predictions of the ensemble

Implementation:
We have divided the responsibility among team members to compare different subsets of models using various ensemble methods. Additionally, one team member will be focused on incorporating new models into the comparison set and evaluating the performance of each model. This distribution of tasks will allow us to thoroughly compare different models and ensembles, as well as investigate the effectiveness of incorporating new models into our analysis.

It's useful to specify the way the work is divided. Additionally, it would be good to provide details on how you would go about implementing bagging and boostiing.

Evaluation:
Given that our objective is to predict the stock closing prices, we will use a minimum of three performance metrics, including RMSE, MSE, and MAE.

Evaluation can also include what you would expect to see from using such performance metrics. How would the met-...

Expand

2) How We Have Addressed Feedback From the Proposal Evaluations

We have dived deeper into prior work and realized that existing studies only combined LSTM and GRU models, and did not provide detailed explanations on why LSTM and GRU are considered the best options. Additionally, they only used one ensemble method, which is adding them as different layers in the network.

Please cite papers when making statements about the lack of existing studies

We want to conduct a thorough comparison of various ensemble methods, such as bagging, boosting, and stacking, to determine the most effective approach for combining models for stock price prediction.

ℸHi

2

The ensemble methods and the time series models are both great ideas.

In addition, we plan to incorporate statistical time series models, such as ARIMA and ARCH, into our study to explore whether there are alternative options that may yield better results.

We will also try models like FB Prophet to see their performances.

We will also provide a comprehensive evaluation of the performance of different models and ensembles, including the proposed approach. If possible, we will try to dive deeper into different algorithms and explain the reasons behind the observed results in detail.

3) Prior Work We are Closely Building From

The prior work we are using comes from projects related to stock market prediction, and specifically gauging which algorithms work best.

.

A. Title: Advanced Stock Pred using SVR, RFR, KNN, LSTM, GRU
   URL:
   https://www.kaggle.com/code/ysthehurricane/advanced-stock-pred-using-svr-rfr-knn-lstm-gru/notebook
   Relevance to Project: In this paper, the authors used SVR, RFR, KNN, LSTM, and GRU as machine learning algorithms for stock prediction.
B. Title: A Comparative Analysis of Machine Learning Algorithms in Stock Prediction
   URL: http://ieomsociety.org/proceedings/2021rome/800.pdf
   Relevance to Project: In this paper, the authors used Linear Regression, KNN, FB Prophet, and LSTM as machine learning algorithms for stock prediction.

4) Contributions

4.1 Implementation Contributions

We are planning to do Option 1, implementing and systematically comparing a wide variety of approaches. We will utilize ensembling and feature engineering to aid our research.

Models we are currently comparing and parameters we are considering tuning:

- SVR (currently kernel= 'rbf', C= 1e2, gamma= 0.1):
- RFR (n_estimators = 100, random_state = 0):
- KNN (n_neighbors = 15)
- LSTM (three layers, each shape = 32)
- GRU (4 layers, each shape = 32)
- LSTM + GRU (2 each, all shape = 32)
- ARIMA (tried p = [0,3], d = [0,3], and q = [0,3]
- FB Prophet (not set up yet)
- Linear Regression (not set up yet)

The ensembles will be chosen according to their individual performance on the dataset and the diversity of their predictions.Currently considering:

- Linear Regression + Support Vector Regression + K-Nearest Neighbors

C

Hide

- LSTM + GRU + FB Prophet
- Linear Regression + Support Vector Regression + K-Nearest Neighbors

4.2 Evaluation Contribution

Our main research objectives include identifying the optimal model for stock price prediction and exploring the factors that contribute to its effectiveness. In the event that we are unable to determine the ideal model, we aim to investigate the limiting factors that affect the predictive performance.

This sounds like a really good idea

Our approach will consist of evaluating and comparing the performance of various baseline models. We will consider models used in previous research and incorporate additional features for model training. The prior work only used past closing prices to predict future closing prices, and we aim to expand the feature set to improve the model's performance.

Why are these good metrics to use for this problem? Cite references that used this on the dataset

Given that our objective is to predict the stock closing prices, we will use a minimum of three performance metrics, including RMSE, MSE, and MAE.

We are planning to incorporate additional datasets that contain stock price data over time. The current dataset we are working with only covers the stock price data for Reliance Industries Ltd over a period of 249 days, and may not be r Incomplete sentence ader stock market.

However, we also understand that

As our data is based on time series, we plan to use cross-validation, rolling window analysis, or sliding window analysis to handle dataset shifts. Currently, we are focusing on testing cross-validation as a method to handle dataset shifts.

Preliminary results:
Current Code URL:
https://colab.research.google.com/drive/1UedqtXHcGd9gmXj7OutUUfE8E7EzkWQR?usp=sharing

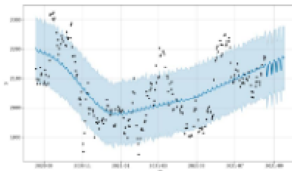
EDA:
We ran some exploratory data analysis on the dataset. **EDA examples are attached at the end.**

After conducting a preliminary analysis, we identified several limitations with the current database. Firstly, the database only includes data from a single company in India, which may not be representative of the overall stock market. Additionally, the database only covers a one-year period, which can restrict the effectiveness of statistical time series models that rely heavily on seasonal trends. To address these limitations, we are exploring the possibility of incorporating additional databases to enhance the scope of our study.

This sounds like a good idea to consider but please make sure that this is feasible within the semester. It sounds like a lot to do apart from the ensemble analysis. I would advise caution

Models:
We have successfully run the dataset using several models, partially using the work from prior sources.
Example of model performances are attached at the end.
We attempted to use the ARIMA model, but the results were not satisfactory as we had anticipated. This is likely due to the limited size of the dataset, which is insufficient for conducting a robust time series analysis.
Sample attached at the end.

We are currently working toward testing using other models.
In progress: fbprophet



Datashifts:
Currently, we have only applied cross-validation on Linear Regression and found that it did not result in poor model performance. However, as time series data is continuous, we aim to investigate the impact of data shifts on other models and evaluate their performance accordingly.

5) Risk Mitigation Plan

We have identified the risk of not finding a better implementation/algorithm than the current "best" one (LSTM + GRU) mentioned in our prior research. To mitigate this risk and produce a thorough project report, we plan to conduct a deeper investigation into this algorithm. We aim to determine its strengths and weaknesses and understand why it outperforms other models in forecasting stock prices. We will also examine if its performance deteriorates when predicting over longer periods of time. Although previous research has not answered these questions, we believe that our comprehensive analysis will enable us to produce a valuable report. Even if our conclusion remains the same, the in-depth research will provide a more nuanced understanding of the algorithm and its application in stock price prediction.

This is a good plan. Stop by my office hours if you would like to discuss more.