Project Report:
Forecasting the NIFTY50 Index

Group 11

Members:
Arqam Patel, 210194
Pulak Gautam, 210791
Akshat Singh Tiwari, 210094
Anuj Sarda, 210931
Tanmay Purohit, 211097

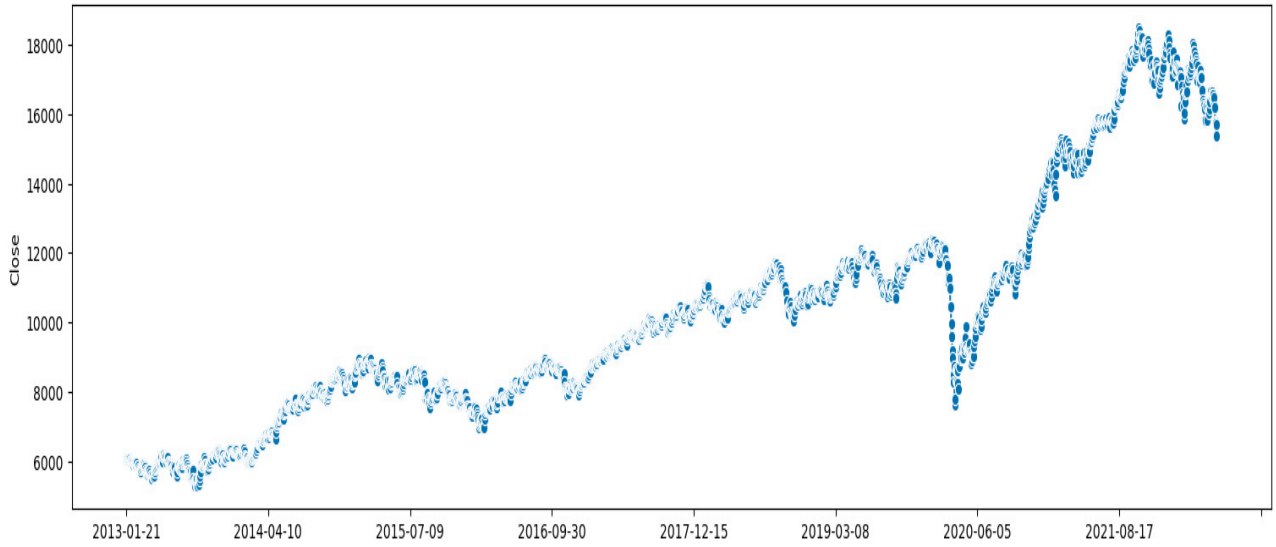
INDEX

Problem statement:
Our task is to predict the NIFTY50 close for the next two days, given the historical data for the previous 50 days. We explore and compare various techniques for forecasting.
We use the metrics MAE (mean absolute error), and directional accuracy to benchmark these various methods.

Data
We use data from Yahoo Finance, from 2013 to June 2023. We use 90% of the data for training and calculate our test metrics on the remaining 10%, keeping the parameters fixed. We have access to Open, Close, Low, High, Volume, and the respective adjusted values.

Exploratory analysis of the time series data

This Exploratory Data Analysis (EDA) aims to uncover patterns, trends, and important characteristics within the time series data. The analysis includes the application of differencing and the interpretation of Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to guide the exploration of stationarity.



Differencing was applied to the time series data to achieve stationarity. The following observations were made:

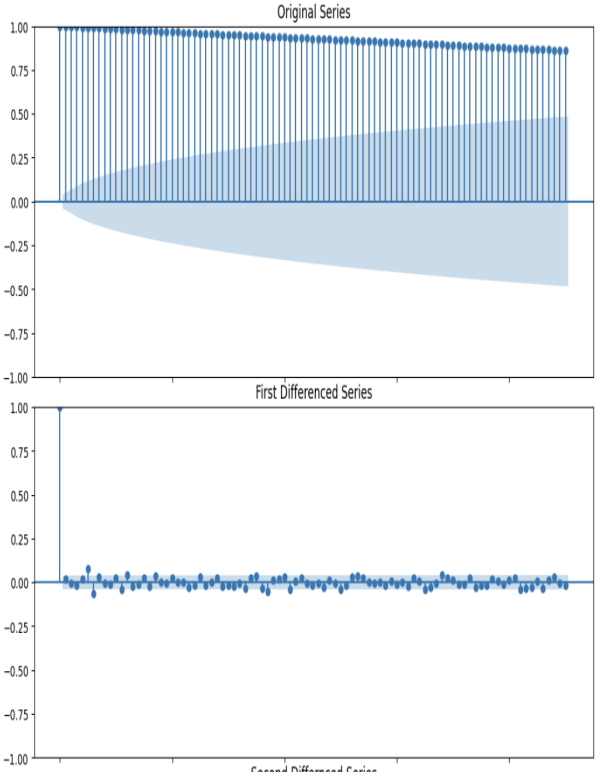
1st Differenced Series:

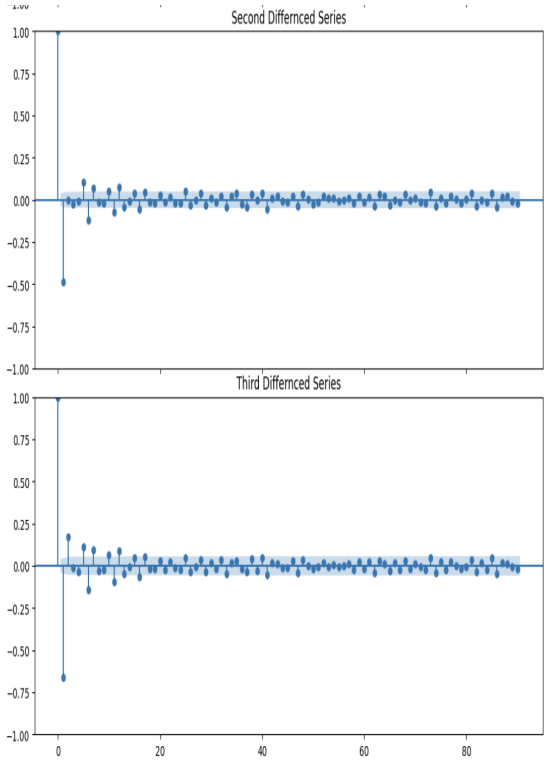
Interpretation: The 1st differenced series is close to stationary, indicating successful trend removal. Implication: A stationary series is conducive to more accurate modeling and forecasting.

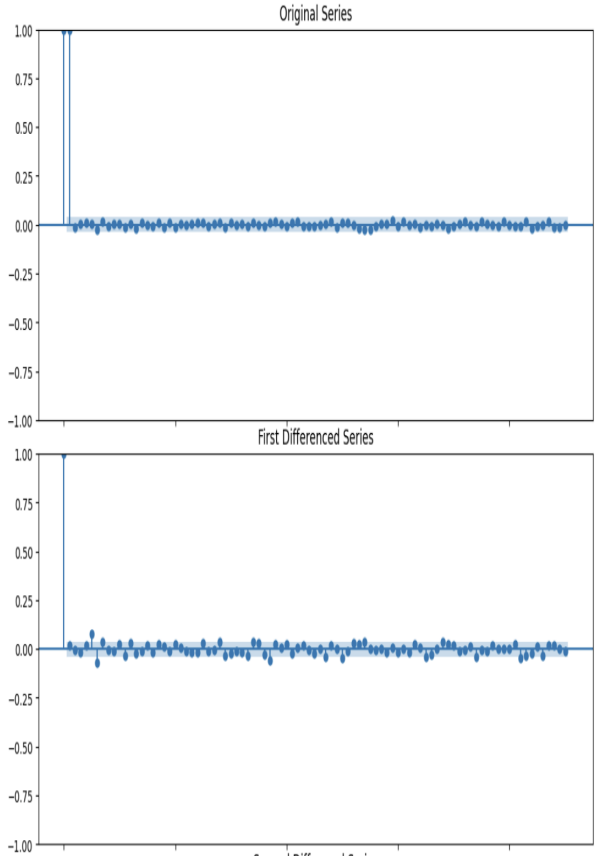
2nd Differenced Series:

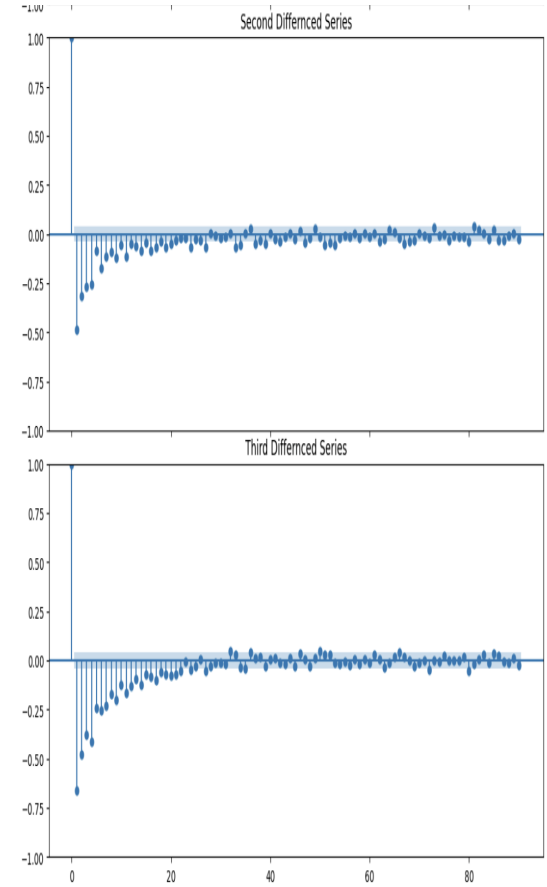
Interpretation: The 2nd differenced series is close to stationary, but a large negative PACF at lag 1 suggests potential over-differencing.Implication: Over-differencing may introduce unnecessary complexity, necessitating careful model selection

ACF Plots and PACF plots







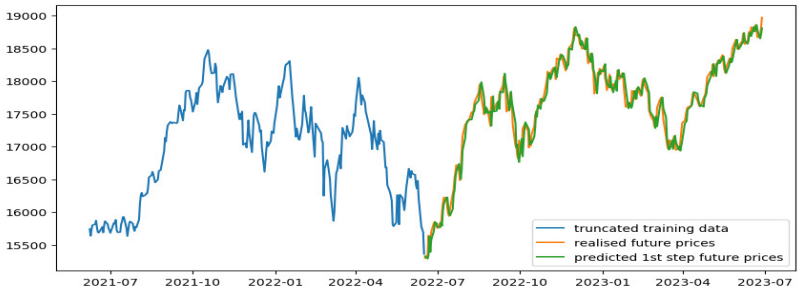


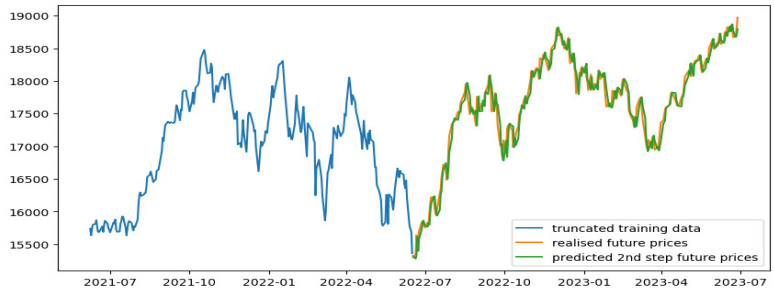
Univariate Time Series Models

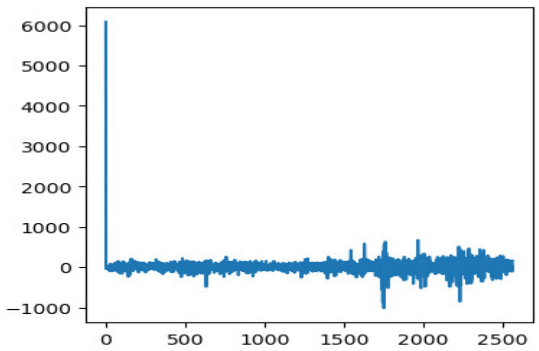
We test ARIMA models of several orders based on rough estimates from the observations of ACF and PACF plots. We test the hypothesis that (5,1,5) is the true order, and check against adjacent values.

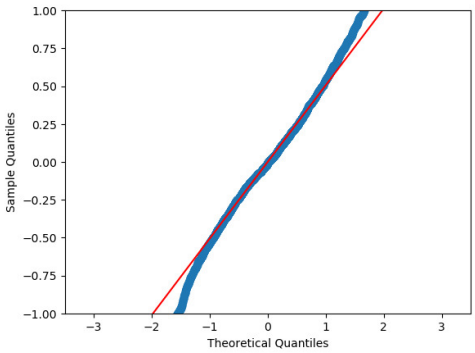
ARIMA Order	AIC	BIC	R^2 Score	MAE	Directional Accuracy
(5,1,5)	31645.024	31709.358	0.970950	103.738816	0.501961
(4,1,4)	31661.553	31714.190	0.971072	104.914693	0.478431
(6,1,6)	31648.796	31724.827	0.970978	103.680435	0.505882

The ARIMA(5,1,5) model exhibits a low directional accuracy, despite having a decent Mean Absolute Error (MAE). This indicates that the current approach may not be suitable, as the model's directional predictions are comparable to randomly choosing to either bid up or down.
Additionally, upon examining the Q-Q plot of the residuals for this model, we observe that they closely adhere to a normal distribution. This suggests that there may be limitations in modelling further from a classical time series perspective.









Feature engineering based methods

Feature engineering involves using domain specific transformations of raw data to create more informative features for modelling.

Technical indicators as features

We use common technical indicators from Investopedia. Some of them are:

1. Bollinger Bands (`Upper_Band`, `Lower_Band`):
   Measure volatility by plotting a set of trendlines two standard deviations away from a simple moving average of the stock's price.
2. Moving Average Convergence Divergence (`MACD`):
   A trend-following momentum indicator that shows the relationship between two moving averages of a stock's price.
3. Commodity Channel Index (`CCI`):
   A versatile indicator that can be used to identify a new trend or warn of extreme conditions when a stock is overbought or oversold.
4. Rate of Change (`ROC`):
   A momentum oscillator that measures the percentage change in price from one period to the next, indicating the speed at which the price is changing.
5. Stochastic Oscillator (`SO%K`):
   A momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time.
6. Force Index (`ForceIndex1`, `ForceIndex20`):
   An indicator that uses price and volume to assess the power behind a price move, with the 1-day Force Index using one period of data and the 20-day Force Index using twenty periods.

Data augmentation

We make use of the fact that the BSE Sensex is a very similar and highly correlated, we use the historical data of Sensex, after normalising it, to augment our training set

Recursive feature elimination

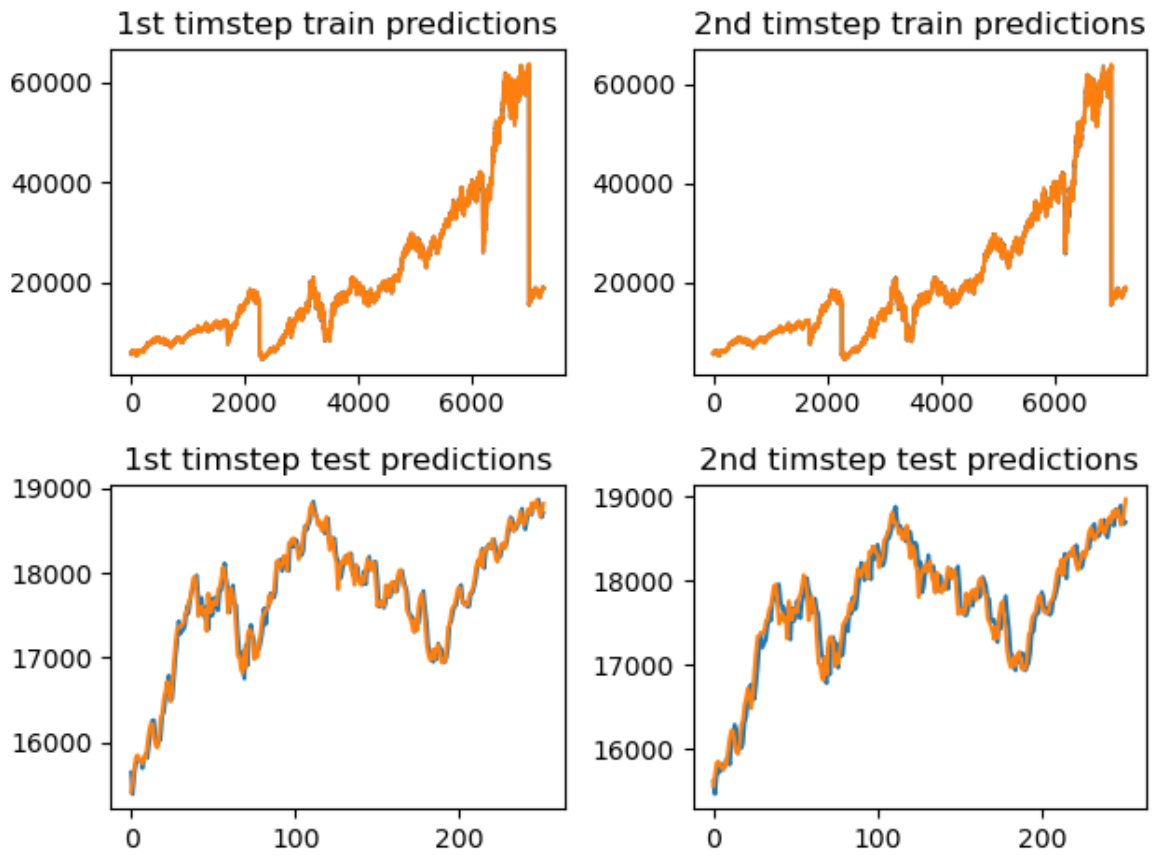
Recursive Feature Elimination (RFE) is a feature selection method that fits a model and removes the weakest feature (or features) at each iteration, which is considered the least important for predicting the target variable. It recursively builds models and eliminates a small number of features at each iteration that are deemed the least important until the specified number of features is reached. This process helps in identifying a subset of features that contribute most to the prediction variable or output in which you are interested. By

doing this, RFE helps improve the model's performance by eliminating redundant or irrelevant data that may negatively impact the model's accuracy.

Model Type	Train R^2	Train MSE	Train MAE	Test R^2	Test MSE	Test MAE	ADA
Linear Regression, lag = 50	0.999550	95811.80	204.26	0.951802	25770.08	125.63	0.626984
Ridge Regression, lag = 50	0.999547	96452.76	203.49	0.952345	25479.13	124.81	0.607143
LGBM Regression, lag = 50	0.999879	25883.45	118.74	0.947140	28285.70	134.78	0.531746
Linear Regression with RFE, lag = 50	0.999490	108751.78	209.66	0.954324	24432.52	121.75	0.587302
Ridge Regression with RFE, lag = 50	0.999486	105909.82	206.32	0.954919	24112.01	121.55	0.599206

We can see that the LGBM model, which is a gradient boosting and decision tree based algorithm commonly used for tabular datasets, highly overfits. Linear and ridge regression on the other hand, perform reasonably well both in sample and out of sample. There is an improvement in MAE after we eliminate features, but the directional accuracy drops as a result.

The results of the best model we obtained are:



State Space models

State space models are a framework used in various fields such as control theory, signal processing, statistics, and machine learning.. These models describe the evolution of a system over time by representing it in terms of states, observations, and control inputs. State space models are widely applied in diverse areas due to their flexibility and ability to handle complex dynamic systems.

SpaceTime:

A new state-space time series architecture for efficient sequence modelling which employs deep learning techniques.

A deep architecture that uses structured state-spaces for more effective time-series modeling. Space Time is a standard multi-layer encoder-decoder sequence model, built as a stack of repeated layers that each parametrize multiple SSMs. We designate the last layer as the “decoder”, and prior layers as “encoder” layers. Each encoder layer processes an input time series sample as a sequence-to-sequence map. The decoder layer then takes the encoded

sequence representation as input and outputs a prediction(for classification) or sequence(for forecasting).

State-space models for time series: We build on the discrete-time state-space model(SSM), which maps observed inputs uk to hidden states xk, before projecting back to observed output yk.

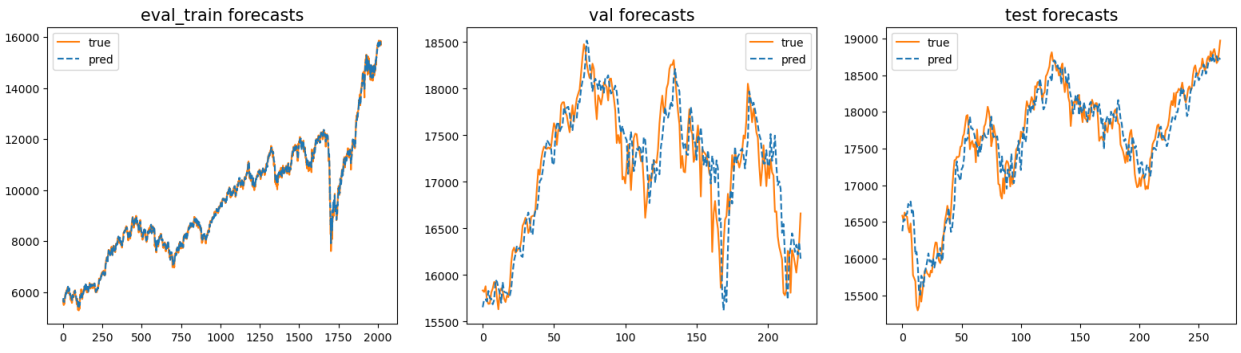
$$x_{k+1} = \boldsymbol{A}x_k + \boldsymbol{B}u_k$$
$$y_k = \boldsymbol{C}x_k + \boldsymbol{D}u_k$$

where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, $\boldsymbol{B} \in \mathbb{R}^{d\times m}$, $\boldsymbol{C} \in \mathbb{R}^{m'\times d}$, and $\boldsymbol{D} \in \mathbb{R}^{m'\times m}$. For now, we stick to *single-input single-output* conventions where $m, m' = 1$, and let $\boldsymbol{D} = 0$. To model time series in the single SSM setting, we treat $\boldsymbol{u}$ and $\boldsymbol{y}$ as copies of the same process, such that

$$y_{k+1} = u_{k+1} = \boldsymbol{C}(\boldsymbol{A}x_k + \boldsymbol{B}u_k)$$

We can thus learn a time series SSM by treating A,B,C as black-box parameters in a neural net layer,i.e. , by updating A,B,C via gradient descent s.t. With input uk and state xk at time-step k, the previous equation predicts y(k+1) that matches the next time-step sample yk+1 = uk+1 .

These are the final results obtained using SpaceTime architecture.



Final Results and Discussion

We list the performances of the various models in terms of the metrics. We also consider several naive benchmarks to compare our models with, to ensure that the forecasting capabilities of our models are significant.

Model	**R^2**	**MAE**	**ADA**
X(t) = X(t-1)	0.999	79.5169	0
X(t) = X(t-1) + 10	0.999	79.35	0.536
X(t) = X(t-1) + e	0.999	79.35	0.50
ARIMA(5,1,5)	0.971	103.73	0.502
Space-Time	0.932	184.22	0.521
Linear regression	0.951	125.63	**0.627**

Only MAE is a bad indicator of the success of a model since it is unable to convey directional information, which is pivotal in any financial use case. This is because in quantitative finance, forecasting is ultimately used to determine the magnitude and direction of the daily position in a financial asset. While profit can be more efficiently produced (i.e. higher return on same amount of capital) if we manage to forecast the magnitude of daily change/returns correctly, getting a better forecast in absolute terms is meaningless if we do not capture the direction of the movement.

Thus, considering that we have achieved significant improvement over. We can conclude that our modelling efforts did yield significant results in forecasting the movements of the index, even without directly optimising over the metric of directional accuracy.

References:

1. Zhang, M., Saab, K. K., Poli, M., Dao, T., Goel, K., & Ré, C. (2023). Effectively Modeling Time Series with Simple Discrete State Spaces. arXiv preprint arXiv:2303.09489