# Stock Market Price Prediction Report

## Project Title: SENSEX Price Forecasting using ARIMA, SARIMA, and LSTM Models

### Author: Lokesh Garg

### Date: June 2025

## 1. Executive Summary

This project focuses on forecasting the **SENSEX** index from **2008 to 2025** using statistical and deep learning techniques, including **ARIMA**, **SARIMA**, and **LSTM**. These models were evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The LSTM model achieved the best performance, indicating its superior capability in capturing non-linear market trends. This report highlights the model development process, comparative performance, visualizations, limitations, and future improvements.

## 2. Introduction

Stock market forecasting is crucial for investors and financial analysts. The **SENSEX**, representing 30 of the largest and most actively traded stocks on the Bombay Stock Exchange, is a major indicator of the Indian stock market’s health. Given its volatility and trend-based patterns, effective prediction requires robust modeling techniques.

### Objectives:

* Forecast future SENSEX values using historical daily closing prices.
* Compare the performance of ARIMA, SARIMA, and LSTM models.
* Assess each model using quantitative metrics and visual insights.

## 3. Data Collection and Preprocessing

* **Source**: Yahoo Finance
* **Time Period**: January 2008 – June 2025
* **Attributes Used**: Date, Closing Price
* **Frequency**: Daily

### Preprocessing Steps:

* Missing values filled using forward fill method.
* Date converted to datetime object and set as index.
* Used MinMaxScaler to scale prices for LSTM (range 0–1).
* Applied first-order differencing to remove trend for ARIMA and SARIMA.
* ADF (Augmented Dickey-Fuller) test conducted to check stationarity.

### Stationarity Test:

The ADF test result yielded a test statistic lower than the critical value at a 5% significance level, confirming that differencing made the series stationary.

### Seasonal Decomposition:

Seasonal decomposition of time series showed weekly seasonality patterns, which were addressed in SARIMA modeling.

### Visualization:

A time-series plot of the SENSEX closing prices from 2008 to 2025 illustrates multiple bullish and bearish phases, global financial crises, and post-pandemic recovery.

## 4. Model Architectures

### 4.1 ARIMA Model

* Model trained on differenced data.
* Parameters selected through iterative testing.
* Final Order: **(p=2, d=1, q=2)**
* Residual diagnostics performed to validate assumptions.

### 4.2 SARIMA Model

* Seasonality incorporated with weekly cycles.
* Final Order: **(p=2, d=1, q=2)(P=2, D=1, Q=2, s=7)**
* Provided significant improvement over ARIMA by capturing recurring weekly patterns.

### 4.3 LSTM Model (Deep Learning)

* Data transformed into sequences using sliding windows.
* Input shape: 60 previous timesteps to predict next value.

from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import LSTM, Dropout, Dense  
  
model = Sequential()  
model.add(LSTM(units=128, return\_sequences=True, input\_shape=(X\_train.shape[1], 1)))  
model.add(Dropout(0.3))  
model.add(LSTM(units=64, return\_sequences=False))  
model.add(Dropout(0.2))  
model.add(Dense(1))  
  
model.compile(optimizer='adam', loss='mean\_squared\_error')  
model.fit(X\_train, y\_train, epochs=25, batch\_size=32, validation\_split=0.1)

## 5. Evaluation Metrics

| Model | RMSE | MAE |
| --- | --- | --- |
| ARIMA | 12124.75 | 9965.74 |
| SARIMA | 3517.34 | 2769.64 |
| LSTM | 948.30 | 731.86 |

### Interpretation:

* **LSTM** demonstrated exceptional performance in both RMSE and MAE, confirming its effectiveness in capturing complex temporal dependencies.
* **SARIMA** outperformed ARIMA by addressing seasonality.
* **ARIMA** was the least accurate, likely due to lack of seasonal modeling and inability to learn non-linear relationships.

## 6. Visual Comparisons

Plots included in analysis:

* Line Chart: Actual vs Predicted Prices for each model
* Residual Plot: Error distribution for ARIMA and SARIMA
* LSTM Loss Curve: Training vs Validation loss over epochs
* Feature Scaling and Inverse Transform Impact

## 7. Limitations

* Classical models (ARIMA/SARIMA) assume linearity and may fail during extreme market shifts.
* LSTM requires extensive hyperparameter tuning and larger datasets.
* No inclusion of macroeconomic indicators (e.g., interest rates, GDP) or qualitative signals (e.g., news sentiment).

## 8. Future Work

* Incorporate external regressors (e.g., Nifty index, global indices, inflation rates).
* Integrate sentiment analysis from financial news and social media.
* Explore hybrid models combining ARIMA/SARIMA with LSTM.
* Experiment with advanced models like GRU, Bi-LSTM, and Transformer-based Time Series Models.
* Build an interactive dashboard using Streamlit for real-time forecasting.

## 9. Conclusion

This project demonstrates the evolution of time-series forecasting, from traditional models like ARIMA and SARIMA to deep learning approaches like LSTM. The superior performance of LSTM highlights the growing importance of data-driven, neural approaches for financial predictions. While classical models provide a solid baseline, modern methods offer enhanced accuracy and adaptability in volatile market environments.

## 10. References

* Yahoo Finance for SENSEX data
* TensorFlow and Keras Documentation
* Statsmodels Python Library for ARIMA/SARIMA
* Time Series Analysis by Chatfield
* Deep Learning for Time Series Forecasting by Jason Brownlee

## Appendix

* Data Preprocessing Code Snippets
* ADF Test Output and Interpretation
* Seasonal Decomposition Charts
* Residual Plots and Model Diagnostics