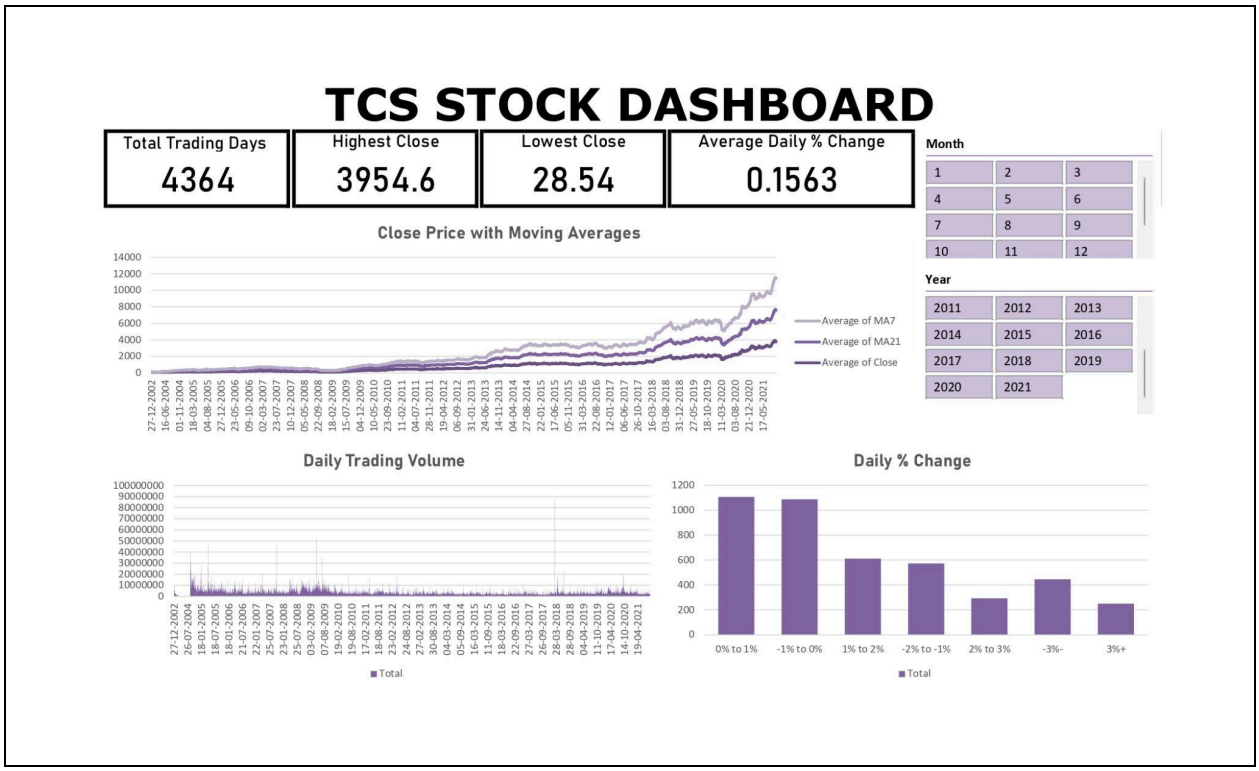
Project 3
Domain: Data Analyst

TCS Stock Data - Live and Latest



Abstract

This project focuses on analyzing and forecasting the historical stock data of Tata Consultancy Services (TCS), one of India's largest IT firms. Using Python (Pandas, Scikit-learn, TenserFlow) and Microsoft Excel, the analysis uncovers price trends, volatility patterns, and future stock behavior. A dynamic Excel Dashboard enhances interpretability through interactive KPIs, slicers, and visualizations.

The project delivers a full pipeline - from data cleaning and feature engineering to predictive modeling (Linear Regression and LSTM) and professional dashboarding. This solution helps in understanding past performance and assists in future stock movement estimation.

Introduction

Stock markets are highly volatile and influenced by a wide range of economic, sectoral, and behavioral factors. This project titled “TCS Stock Data - Live and Latest” explores TCS stock performance over the years, capturing insights through data analysis, visual storytelling, and forecasting models.

Through tools like Google Colab for scripting, Excel for dashboarding, and machine learning for modeling, the project reveals valuable patterns in price fluctuations, volume shifts, and trend movements.

Features

This end-to-end project covers the complete lifecycle of data analysis:

- ★ Loading and cleaning historical TCS stock data (CSV).
- ★ Handling missing values and formatting datetime fields.
- ★ Feature engineering (Daily % Change, Moving Averages, Lag Features).
- ★ Exploratory Data Analysis: Volatility, distribution, and volume analysis.
- ★ Interactive Excel Dashboards with slicers, charts, and KPIs.
- ★ Forecasting with Linear Regression and LSTM Neural Network.
- ★ Evaluation of model accuracy using MAE, RMSE, and visualization.
- ★ Saving models and exporting results for reuse.

Tools and Techniques Used

Tool	**Use**
Python (Google Colab)	Data processing, modeling (Linear Regression, LSTM), feature engineering
Excel	KPI summary, interactive charts, histogram, volume trend
Pandas / NumPy	Data wrangling and transformation

Matplotlib / Seaborn	Visualizing historical and predicted trends
Scikit-learn	Training and evaluating Linear Regression
TensorFlow	Building and training LSTM model

Machine Learning in the TCS Stock Data Analysis and Forecasting

This project applies two machine learning models —- **Linear Regression and LSTM (Long Short-Term Memory)** — to predict the future stock price of TCS based on historical patterns..

Both models serve different purposes:

- ★ Linear Regression offers a **simple, interpretable forecast**
- ★ LSTM offers a **deep learning-based sequence model** that learns temporal patterns over time.

1. Linear Regression

Linear Regression is a supervised learning algorithm used to model the relationship between one dependent variable (target) and one or more independent variables (features). It assumes a linear relationship and fits a best-fit straight line that minimizes the prediction error.

How It's Used in This Project:

Step	**Explanation**
Features (X)	Lagged Close prices (e.g., previous day's price), moving averages (e.g., MA7, MA21), and possibly % change
Target (Y)	Next day's closing price
Data Split	Train-test split to evaluate performance

Training	Model learns coefficients that best predict future closing prices
Evaluation	MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and R^2 score to assess accuracy

2. LSTM (Long Short-Term Memory) Neutral Network

LSTM is a type of Recurrent Neural Network (RNN) designed to handle sequential data and long-term dependencies. It's particularly suited for time series prediction because it retains memory of past inputs across time steps using special "gates" (input, forget, output).

How It's Used in This Project:

Step	Explanation
Input Sequence	Rolling window of past 60 days' closing prices (for scaled data)
Output	Predicted closing price for the next day
Preprocessing	Data is scaled between 0-1 using MinMaxScaler for better convergence
Model Architecture	Sequential LSTM layers with dropout (for regularization) and a dense output layer
Training	Epoch-based training using past sequences
Evaluation	Predicted values are inverse-transformed and compared to actual prices visually and numerically

By combining both models:

★ The **Linear Regression** model gives a fast, explainable forecast and helps benchmark performance

- ★ The **LSTM** model offers a more complex, pattern-aware approach suited for volatile stock data

Together, these models provide both insight and predictive capability, making this project valuable for analysis, traders, and business intelligence dashboards.

Conclusion

This project provided a comprehensive end-to-end analysis of **TCS Stock Data**, combining data cleaning, feature engineering, exploratory data analysis, machine learning, and dashboarding into a single unified workflow.

Through Python and Excel, we explored the behavior of TCS stock over time — analyzing key metrics such as daily percentage change, moving averages, and trading volume. Feature-rich transformations like features and rolling averages allowed us to understand short and long term trends in price movements.

To extend this analysis, we implemented **machine learning models**:

- ★ **Linear Regression** offered a simple, interpretable forecast based on engineered features.
- ★ **LSTM (long Short-Term Memory)** provided a deep learning approach capable of capturing complex temporal dependencies in sequential price data.

The resulting forecasts were then exported and integrated into an **interactive Excel dashboard** designed for decision-makers. It includes KPIs, slicers, trend charts, volume breakdowns, and distribution visualizations — making it easy to explore historical patterns and predictive outcomes dynamically.

Overall, this project not only demonstrates technical proficiency in data analysis and modeling but also shows how to **bridge code and business insight** through a clean, interactive, and professional dashboard — reflecting real-world stock market analysis.

Project Link: https://github.com/Sa880-hue/TCS-Stock-Data---Live-and-Latest

Appendix

Appendix A: Python Code Snippets (Description Only)

Below is a high-level overview of the 7 phases and their corresponding Python code blocks. Actual code is included in the Colab notebook.

- **Phase 1: Data Loading and Cleaning**
  - ➔ Loaded `TCS_stock_history.csv` using Pandas
  - ➔ Converted `Date` column to datetime
  - ➔ Handled missing values and sorted data chronologically
- **Phase 2: Data Preprocessing & Feature Engineering**
  - ➔ Created new columns: `Daily_Change_Pct`, `MA7`, `MA21`, `Year`, `Month`
  - ➔ Added lag features (e.g., previous day's `Close`)
  - ➔ Generated bins and labels for histogram analysis
- **Phase 3: Exploratory Data Analysis (EDA)**
  - ➔ Visualized daily closing price and moving averages
  - ➔ Plotted volume trends and percent change histograms
  - ➔ Analyzed volatility and price distributions using Seaborn and Matplotlib
- **Phase 4: Linear Regression Modeling**
  - ➔ Selected features: `Lag_Close`, `MA7`, `MA21`, `Daily_Change_Pct`
  - ➔ Split data into training and test sets
  - ➔ Trained a Linear Regression model using Scikit-learn
  - ➔ Evaluated predictions using MAE, RMSE, and R^2 score
- **Phase 5: LSTM Neural Network Modeling**
  - ➔ Scaled `Close` prices using MinMaxScaler
  - ➔ Created rolling windows of 60 days as input sequences
  - ➔ Built LSTM layers using Keras (tensorFlow backend)
  - ➔ Trained the model and generated predictions
  - ➔ Inverse-transformed predicted values and evaluated visually and numerically

- **Phase 6: Predictions & Evaluation**
  - ➔ Combined actual vs predicted values for both models
  - ➔ Generated visual plots of model performance over time
  - ➔ Evaluated long-term vs short-term forecasting behavior
- **Phase 7: Export & Dashboard Preparation**
  - ➔ Exported cleaned dataset and model predictions to CSV/Excel
  - ➔ Saved models as .pkl (Linear) and .h5 (LSTM)
  - ➔ Prepared Excel-ready file for dashboard integration

Appendix B: Excel Formula List

Column	Formula	Purpose
Daily_Change_Pct	=(Close - Open) / Open * 100	Calculates daily percent change
Change-Bin	=FLOOR (M2, 1)	Create integer bin for histogram
Bin_Label	=IF (N2 >= 3, "3%+", IF (N2 <= -3, "-3%-", N2 & "% to " & (N2+1) & "%"))	Label each bin range
MA7	=AVERAGE(L2:L8) or =AVERAGE([@Close]: [Close])	7-day moving average
MA21	=AVERAGE(L2:L22)	21-day moving average
Year	=YEAR(Date)	Extract year
Month	=TEXT(Date, "MMMM")	Extract full month name

Appendix C: Data Description

File Name	Description
TCS_stock_history.csv (Primary)	Daily open, high, low, close, volume data
TCS_stock_info.csv	Metadata about the TCS stock
TCS_stock_action.csv	Events like stock splits, dividends, bonuses

This appendix supports the main report and provides clarity for technical reviewers and analysts reviewing the project.