**NIFTY 50 Stock Market Analysis & Prediction**

**📊 Project Overview**

This project involves analyzing historical NIFTY 50 stock market data and predicting daily market movements using machine learning. The project is divided into two parts:

1. A **Jupyter Notebook** (PDS\_PythonProject.ipynb) for data exploration, cleaning, and model building.
2. A **Streamlit App** (app.py) that provides a user-friendly interface for analysis and prediction.

**📁 Files Included**

* NIFTY\_50.csv: Historical stock data.
* PDS\_PythonProject.ipynb: Notebook with EDA and ML model.
* app.py: Basic version of the Streamlit web app.
* streamlit\_app\_improved.py: Enhanced version of the Streamlit app.
* NIFTY50\_Presentation.pptx: Project presentation.

**🧪 Jupyter Notebook Highlights (PDS\_PythonProject.ipynb)**

**✅ Steps Performed:**

* **Data Loading & Cleaning**:
  + Parsed dates
  + Handled missing values with mean/forward fill (no row drops)
* **Feature Engineering**:
  + Daily\_Movement: Up (1) or Down (0) based on Close vs Open
  + Price\_Range, Volatility\_Level
* **Exploratory Data Analysis**:
  + Summary statistics, trend plots, heatmaps, categorical breakdowns
* **Model Training**:
  + Random Forest Classifier
  + Train/Test split with evaluation: Accuracy ~90%, precision, recall

**🌐 Streamlit App Highlights (streamlit\_app\_improved.py)**

**🔧 Features:**

* Upload your own CSV file
* Date range selector
* Interactive visualizations (price trends, volume, heatmaps)
* Live market movement predictor with user inputs

**🧠 Behind the Scenes:**

* The app loads the model trained in the notebook
* Accepts user input for new prediction
* Displays prediction (Market Up or Down)
* Plots trends and statistics for uploaded data

**👨‍💻 How to Run the App**

1. Install dependencies:

pip install streamlit pandas scikit-learn matplotlib seaborn

1. Launch the app:

streamlit run streamlit\_app\_improved.py

**📌 Future Improvements**

* Use advanced models like XGBoost or LSTM
* Time-series forecasting
* Integrate real-time stock data using APIs
* Add model comparison dashboard in app

**📬 Contact**

* GitHub: [Your GitHub link]
* Email: [Your email]