Stock Price Prediction 📈

Project: Stock-Price-Prediction

Author: Abhishek Shukla — @abhii0402

Overview

A Python-based project that collects historical stock data, performs exploratory data analysis and feature engineering, trains time-series and ML models (e.g., ARIMA, Prophet, LSTM, RandomForest), and provides scripts/notebooks for predicting future stock prices. The repository is suitable for learning, experimentation, and baseline comparisons.

Key Features

- Download historical stock data (Yahoo Finance / `yfinance`).
- Visualizations and exploratory data analysis (EDA).
- Feature engineering (rolling means, volatility, returns, technical indicators).
- Multiple modelling approaches: classical time-series (ARIMA/Prophet), deep learning (LSTM), and ML regressors (RandomForest, XGBoost).
- Train / validate / test pipelines with backtesting and walk-forward validation.
- Metrics & model comparison (RMSE, MAE, MAPE, R^2).
- Jupyter notebooks and modular scripts for reproducibility.

Repo Structure (suggested)

```
Stock-Price-Prediction/
├─ data/                    # raw and processed datasets (not committed large
files)
├─ notebooks/               # exploratory notebooks and experiments
│  ├─ 01_data_collection.ipynb
│  ├─ 02_eda.ipynb
│  ├─ 03_feature_engineering.ipynb
│  └─ 04_modeling_and_results.ipynb
├─ src/                     # modular code
│  ├─ data.py
│  ├─ features.py
│  ├─ models.py
│  ├─ train.py
│  └─ evaluate.py
├─ requirements.txt
├─ environment.yml          # optional conda env
```

```
├─ README.md
└─ LICENSE
```

Getting Started

Prerequisites

- Python 3.8+
- pip (or conda)

Install

```
# using pip
python -m venv venv
source venv/bin/activate   # macOS / Linux
venv\Scripts\activate    # Windows
pip install -r requirements.txt
```

`` **(example)**

```
pandas
numpy
matplotlib
seaborn
scikit-learn
tensorflow
keras
xgboost
statsmodels
prophet
yfinance
jupyterlab
joblib
```

Quick Usage

1. Download data for a ticker (e.g., AAPL) and save to `data/`:

```
python src/data.py --ticker AAPL --start 2015-01-01 --end 2024-12-31 --out data/
AAPL.csv
```

1. Run feature engineering:

```
python src/features.py --in data/AAPL.csv --out data/AAPL_features.csv
```

1. Train a model (example LSTM):

```
python src/train.py --config configs/lstm_config.yaml
```

1. Evaluate & plot results:

```
python src/evaluate.py --model outputs/lstm_best.pkl --test data/AAPL_test.csv
```

Notebooks

Open the notebooks in `notebooks/` to reproduce EDA, visualizations, and model experiments. Notebooks include step-by-step explanations and plots.

Modeling Notes

- **Data split**: Use time-based splitting (train / validation / test) — avoid random shuffling for time-series.
- **Scaling**: Apply MinMax or StandardScaler fit on training set only.
- **Feature lagging**: Use lag features (t-1, t-2...) and rolling windows for moving averages and volatility.
- **Backtesting**: Implement walk-forward validation to simulate realistic forecasting.

Evaluation Metrics

Common metrics provided:

- RMSE (Root Mean Squared Error)
- MAE (Mean Absolute Error)
- MAPE (Mean Absolute Percentage Error)
- R^2 (Coefficient of determination)

Tips & Tricks

- Try differencing or log transforms for non-stationary series before ARIMA.
- For neural networks, tune sequence length, batch size, and epochs; use early stopping.
- Combine models with simple ensembling (averaging or weighted blending) for better stability.

Dataset & Sources

- Historical prices: Yahoo Finance via `yfinance`.
- Optionally add fundamentals or alternative data (sentiment, news, macro indicators) to improve predictions.

Contributing

Contributions are welcome! If you'd like to contribute:

1. Fork the repo.
2. Create a feature branch: `git checkout -b feat/my-feature`
3. Commit your changes and push.
4. Open a pull request with a clear description.

Please follow the coding style and include tests for new modules where applicable.

License

This project is available under the **MIT License**. See `LICENSE` for details.

Contact

Abhishek Shukla — GitHub: @abhii0402

If you'd like this README in Hinglish or want a shorter/longer version, or a ready-to-upload `README.md` file with badges and images — I can update it. 🚀