perplexity

Stock Market Predictor App: Expert Development Roadmap & Feasibility Analysis

Executive Summary & Feasibility Assessment

Project Complexity Rating: 7/10 (Medium-High)
Success Probability: 70% for MVP, 40% for full-featured version
Recommended Approach: Focus on MVP first, then iterate

Your project idea is **ambitious but achievable** within 6 weeks if properly scoped. Based on my analysis of current technologies and your skill level, I recommend building a **Minimum Viable Product (MVP)** first, then expanding features incrementally.

Core Technology Stack Analysis

Backend & APIs

- **Web Framework**: Flask (lightweight, Python-friendly)[1] [2] [3]
- **Database**: SQLite for development, PostgreSQL for production[2] [3]
- **Stock Data**: yfinance (free), Alpha Vantage API (free tier: 500 calls/day)[4] [5] [6]
- **News Data**: NewsAPI (free tier: 1000 calls/day), Finnhub (free tier: 60 calls/minute)[7] [8]
- **ML Framework**: TensorFlow/Keras for LSTM models[9] [10] [11]

Cross-Platform GUI Options

Based on your Windows/Mac collaboration requirement:

- **Recommended: Flet** - Python-based, Flutter-powered, excellent for beginners[12]
- **Alternative: Kivy** - More mature but steeper learning curve[13] [14]
- **Advanced Option: Flask web app** - Accessible via browser on both platforms[15] [2]

Sentiment Analysis

- **VADER**: Superior for social media content, handles emojis and caps[16] [17]
- **TextBlob**: Simpler implementation, good for news articles[17] [16]

Detailed 6-Week Development Roadmap

Week	Tasks	Person A (Windows)	Person B (Mac)	Collaboration Tools
Week 1	Setup & Authentication	Flask app setup, user registration	Database design, login system	GitHub, shared repository [15] [18]
Week 2	Data Integration	Stock API integration (yfinance)	News API setup (NewsAPI)	API testing, data validation
Week 3	Sentiment Analysis	VADER sentiment implementation	News data processing pipeline	Code reviews, testing
Week 4	ML Model Development	LSTM model architecture	Data preprocessing, training	Model sharing via GitHub
Week 5	GUI & Integration	Flet/Kivy GUI development	Dashboard components	Cross-platform testing
Week 6	Testing & Deployment	Bug fixes, optimization	Documentation, deployment	Final integration testing

Required Learning Path & Time Investment

Week 1 Preparation (Before Starting)

- **Flask Fundamentals**: 15-20 hours[3] [2]
- **API Integration Basics**: 10 hours[5] [4]
- **Git/GitHub Collaboration**: 5 hours[18] [15]

Technical Skill Requirements by Week

1. **Flask + SQLite Authentication**: Medium difficulty[2] [3]
2. **API Integration**: Easy-Medium[6] [4] [5]
3. **Sentiment Analysis**: Medium[16] [17]
4. **LSTM Implementation**: High difficulty[19] [11] [9]
5. **Cross-platform GUI**: Medium-High[13] [12]

Recommended MVP Feature Set

Phase 1: Core MVP (Weeks 1-4)

✅ User authentication with SQLite database
✅ Basic stock dashboard with real-time prices
✅ Simple news sentiment analysis (VADER)
✅ Basic prediction display (linear regression first)
✅ Web-based interface (accessible on both platforms)

Phase 2: Advanced Features (Post-MVP)

☆ LSTM neural network implementation
☆ Advanced sentiment quantification
☆ Native desktop app (Flet/Kivy)
☆ Real-time WebSocket updates
☆ Portfolio tracking features

Critical Risk Mitigation Strategies

Technical Risks & Solutions

- **LSTM Complexity**: Start with simpler linear regression, upgrade later[20] [21]
- **API Rate Limits**: Implement caching, use multiple free APIs[4] [5]
- **Cross-platform Issues**: Use web-based approach initially, then native apps[12]
- **Data Quality**: Focus on major stocks (AAPL, MSFT, GOOGL) for consistency[22] [19]

Project Management Recommendations

- **Daily standups**: 15-minute progress sync
- **Weekly sprint reviews**: Assess progress and adjust scope
- **Version control**: Branching strategy for parallel development[15] [18]
- **Testing strategy**: Continuous testing throughout development

Essential Libraries & Dependencies

```
# Core requirements.txt
flask==2.3.3
flask-sqlalchemy==3.0.5
flask-login==0.6.3
pandas==2.0.3
numpy==1.24.3
yfinance==0.2.18
requests==2.31.0
vaderSentiment==3.3.2
textblob==0.17.1
tensorflow==2.13.0
scikit-learn==1.3.0
flet==0.10.3  # For cross-platform GUI
```

Collaboration Workflow Setup

Repository Structure

```
stock-predictor-app/
├── backend/
│   ├── app.py (Flask main)
│   ├── models.py (Database)
│   ├── api_handlers.py (Stock/News APIs)
│   └── ml_models.py (LSTM/Predictions)
├── frontend/
│   ├── gui_main.py (Flet/Kivy app)
│   └── templates/ (Web templates)
├── data/
├── tests/
└── docs/
```

Development Environment Setup

1. **Shared Development**: Use GitHub Codespaces or similar[15]
2. **Local Setup**: Virtual environments on both Windows/Mac
3. **API Keys Management**: Use environment variables, shared .env files
4. **Database Sync**: SQLite for development, shared via Git

Success Metrics & Milestones

Week-by-Week Success Criteria

- **Week 1**: Working authentication system
- **Week 2**: Real-time stock data display
- **Week 3**: Basic sentiment scores from news
- **Week 4**: Simple prediction model working
- **Week 5**: Cross-platform app launched
- **Week 6**: End-to-end system functional

MVP Definition of "Done"

✅ Users can register/login securely
✅ Dashboard shows favorite stocks with real prices
✅ News sentiment affects prediction scores
✅ Basic future price predictions displayed
✅ App works on both Windows and Mac

Final Recommendation

This project is absolutely feasible if you focus on building an MVP first. Your A-Level CS background provides a solid foundation, and the 6-week timeline is realistic for a functional prototype.[23] [24]

Key Success Factors:

1. Start with web-based interface (easier cross-platform)
2. Use pre-trained sentiment analysis (VADER/TextBlob) initially[17] [16]
3. Begin with simple ML models before attempting LSTM[21] [20]
4. Implement continuous integration from day one[18] [15]
5. Plan for post-project iterations and improvements

The project will provide excellent learning opportunities in full-stack development, API integration, machine learning, and collaborative software development - all valuable skills for your computer science degree and future career.

⁂

now give me an exact step by step roadmap on how i should learn everything needed for the development of the app. so i have decided to use kivy as the cross platform GUI option and i will not be using any web framework as this will be a desktop app for both mac and windows. i also do not know how to use APIs or have any knowledge on machine learning frameworks. so using this information and also including any python libraries and other things i need to learn create a detailed step by step roadmap in order.

Here is a precise, ordered learning roadmap to build the desktop app with **Kivy**, integrate **APIs**, and implement an **LSTM** forecaster, tailored for Windows and macOS without any web framework.[119] [120] [121] [122]

Roadmap order

- Kivy UI basics → SQLite for accounts → Stock/news APIs → VADER sentiment → TensorFlow/Keras LSTM forecasting → Desktop packaging with PyInstaller.[123] [121] [124] [125] [122] [126] [119]

Phase 0: Environment setup

- Install Python 3.x and create a virtual environment, then install Kivy using the official "Getting Started" guide and confirm with a "Hello World" app from Kivy Basics.[127] [120]
- Skim Kivy's Tutorials index to see the types of apps and patterns that will be used (widgets, layouts, animations, screen navigation).[119]
- Optional UI upgrade: plan to use KivyMD components later for material-style widgets once fundamentals are comfortable.[128]

Phase 1: Kivy fundamentals (UI, layouts, screens)

- Learn the Kivy app lifecycle, how the App class builds a root Widget, and how to structure a minimal GUI with labels/buttons.[120]
- Study widgets, layouts, and the KV language from the Kivy tutorials and a beginner-friendly guide to quickly prototype multi-screen layouts.[129] [119]
- Practice: build a multi-screen skeleton (Login, Dashboard, AI Insights) using ScreenManager and simple placeholders for content.[119]

Phase 2: Local data storage with SQLite (accounts and preferences)

- Learn the Python sqlite3 module: connecting, creating tables, inserting/selecting users, and committing transactions.[126]
- Design a minimal schema: users(id, username UNIQUE, password_hash, created_at), favorites(id, user_id, symbol), and basic CRUD to save/read favorite tickers.[126]
- Practice: implement a local "Sign up / Log in / Save favorites" flow with SQLite queries wired to Kivy input fields and buttons.[126]

Phase 3: Working with stock and news APIs (no prior API experience required)

- Fetch historical and current market data using yfinance's high-level Python interface, starting with Ticker().history() and yf.download() for multiple symbols.[121] [130]
- Understand free stock API rate limits using Alpha Vantage as a fallback for intraday endpoints and planning refresh cadence around 5 calls/min and 500/day (free tier).[131]
- Integrate a news API using NewsData.io's REST documentation to query by keyword, language, and date, and parse JSON responses for article titles, descriptions, and published dates.[124]
- Practice: build a data layer that can load a user's favorite tickers from SQLite, download latest prices and recent news headlines for those tickers, and store a cached snapshot locally.[121] [124]

Phase 4: News sentiment analysis with VADER

- Learn VADER's sentiment outputs (neg, neu, pos, compound) from the official docs and how to interpret the compound score.[125] [132]
- Use standard thresholds for the compound score for classification: compound > 0.05 positive, < -0.05 negative, else neutral.[133]
- Practice: compute per-article sentiment, then aggregate recent articles for a symbol into a rolling positive/negative ratio and a weighted average compound score.[125] [133]

Phase 5: Time-series forecasting with TensorFlow/Keras LSTM

- Follow a beginner LSTM time-series tutorial in TensorFlow/Keras to understand input shape, sequence windows, model definition, and MSE/Adam compilation.[122]
- Build datasets from yfinance price history (e.g., close price windows) and train a simple stacked LSTM → Dense(1) model for next-step prediction as an initial baseline.[122] [121]
- Practice: create a training script that loads historical data for a selected ticker, trains the LSTM, saves the model weights, and runs an inference to produce a next-day or next-interval forecast.[121] [122]

Phase 6: Wiring data+ML into Kivy

- Connect the SQLite-backed login and favorites UI with data loaders for price history and news so that a dashboard can list favorite tickers with latest prices and sentiment summaries.[120] [126]
- Add an "AI Insights" screen that triggers a prediction run using the saved LSTM model and displays the predicted direction with recent sentiment as context.[125] [122]
- Practice: add refresh buttons and modest periodic polling to update visible data without freezing the UI, keeping UI updates within the Kivy app lifecycle methods.[120]

Phase 7: Packaging for Windows and macOS

- Package the Kivy desktop app using PyInstaller, referencing known tips for bundling Kivy/KivyMD resources so the executable runs outside the dev environment.[123]
- Build separate executables on Windows and macOS machines, and test that SQLite files and any local model weights are found and loaded correctly by the packaged app.[123]
- Practice: produce a single-folder build with PyInstaller and verify that login, dashboard, sentiment, and prediction screens all function on both platforms.[123]

Suggested 6-week learning cadence

- Week 1: Kivy app lifecycle, KV language, multi-screen scaffold, and a working UI skeleton.[129] [119] [120]
- Week 2: SQLite integration for accounts and favorites, with working sign-up/login and persistence.[126]

- Week 3: Stock data via yfinance and news via NewsData.io, shown on the dashboard with basic refresh logic.[124] [121]
- Week 4: VADER sentiment analysis on recent news with per-ticker aggregates and classification thresholds.[133] [125]
- Week 5: TensorFlow/Keras LSTM training on historical prices and a basic forecast surfaced in the app.[122] [121]
- Week 6: Polish and package with PyInstaller, test on Windows/macOS, and prepare a short demo.[123]

Libraries and tools to learn (in order)

- Kivy (core UI, KV language, ScreenManager) for the cross-platform desktop GUI.[119] [120]
- KivyMD (optional) to speed up building polished material-style components.[128]
- sqlite3 (Python standard library) for local user accounts and favorites.[126]
- yfinance for market data retrieval directly in Python.[130] [121]
- Alpha Vantage REST API basics and rate limits for intraday or fallback data planning.[131]
- NewsData.io for a simple news API workflow and JSON parsing.[124]
- VADER sentiment for rule-based sentiment and interpretable compound scores.[133] [125]
- TensorFlow/Keras LSTM for time-series forecasting and model training/inference.[122]
- PyInstaller for packaging Kivy apps as desktop executables.[123]

Concrete practice milestones

- Build and navigate a three-screen Kivy app (Login, Dashboard, AI Insights) with static placeholder data.[120] [119]
- Implement SQLite-backed sign-up/login and favorites management, verified by closing and reopening the app.[126]
- Display price charts and latest prices for favorite tickers using yfinance, and list recent news headlines from a news API query.[121] [124]
- Attach VADER sentiment to each headline and show per-ticker aggregate sentiment with a labeled indicator.[125] [133]
- Train a small LSTM on one ticker's history and display the next prediction with a simple confidence or error metric.[121] [122]
- Package the app for both platforms with PyInstaller and confirm functionality on fresh machines.[123]

Notes on scope and expectations

- yfinance is a convenient educational tool for downloading market data from Yahoo Finance, making it suitable for prototypes and learning integrations.[121]
- Alpha Vantage free-tier limits necessitate batching and spacing requests when fetching more frequent intraday data in a desktop app.[131]
- VADER's compound thresholds provide a fast, baseline sentiment classifier; advanced NLP can be explored later if time remains.[133] [125]
- A simple LSTM following a beginner tutorial is sufficient for demonstrating end-to-end forecasting in six weeks, with future tuning as an extension.[122]

If this exact order is followed—Kivy UI → SQLite → market/news APIs → VADER sentiment → TensorFlow LSTM → packaging—the app can be implemented with a realistic learning curve for beginners to APIs and ML on Windows and macOS.[124] [120] [125] [122] [121] [126] [123]

⁂

okay this is good, now building upon this give me a nice neat table and a list containing everything i need to learn (libraries, frameworks, applications, technologies, etc...) with the columns: name, learning time, good resources to learn, what its used for.

Here is a consolidated learning plan presented as a neat **table** plus an ordered **checklist**, covering libraries, frameworks, tools, and what each is for, with time estimates and reliable resources.[219] [220]

Learning table

Name	Learning time	Good resources to learn	What it's used for
Kivy (core + KV language)	10–15 hours	Kivy Tutorials and Basics in official docs [220] [221]	Cross-platform Python GUI, widgets, layouts, KV language, ScreenManager for multi-screen apps [220] [221]
KivyMD (optional)	4–6 hours	Quick KivyMD tutorial video walkthrough [222]	Material Design components for faster, modern UI inside Kivy apps [222]
sqlite3 (Python stdlib)	6–8 hours	SQLite with Python tutorial (queries, tables, CRUD) [223]	Local user accounts, favorites, and settings persistence (no server needed) [223]
yfinance	4–6 hours	yfinance docs and API reference [224] [225]	Historical and recent market data (prices, OHLCV) directly in Python [224] [225]
Alpha Vantage (optional)	3–4 hours	Free tier limits and usage notes [226]	Intraday/fallback stock data and indicators with API key and rate-limit awareness [226]

Name	Learning time	Good resources to learn	What it’s used for
NewsData.io (news API)	4–6 hours	NewsData.io documentation (REST filtering, JSON fields) [227]	Fetch company news headlines/descriptions/dates for sentiment analysis [227]
VADER sentiment	3–5 hours	VaderSentiment docs + example guide [228] [229]	Fast rule-based sentiment on headlines; compound score aggregation per ticker [228] [229]
TensorFlow/Keras (LSTM)	12–20 hours	Beginner LSTM time-series tutorial in TensorFlow/Keras [230]	Sequence modeling for next-step price forecasting and demo predictions [230]
pandas	8–12 hours	pandas official documentation (User Guide + API) [219]	DataFrames for cleaning, joining, windowing, and preparing ML datasets [219]
NumPy	8–12 hours	NumPy absolute basics for beginners [231]	Efficient arrays, numerical ops, and shaping tensors for ML models [231]
Matplotlib	6–10 hours	Matplotlib documentation (Users guide + API) [232] [233]	Static charts for price history, sentiment summaries, and model outputs [232] [233]
requests	4–6 hours	Real Python: Requests guide (HTTP/JSON patterns) [234]	Calling REST APIs (news, fallback market APIs), headers, auth, JSON parsing [234]
python-dotenv	1–2 hours	Dotenv usage tutorial (env vars, API keys) [235]	Keep API keys/config out of code, load from .env into environment [235]
PyInstaller (with Kivy)	3–5 hours	Packaging Kivy/KivyMD desktop apps with PyInstaller [236]	Build distributable desktop executables for Windows and macOS [236]
Git (core)	6–10 hours	Official Git docs and command reference [237] [238]	Version control: branching, commits, merges, and collaboration workflow [237] [238]
GitHub workflow	4–6 hours	GitHub Docs: Hello World and getting started [239] [240]	Pull requests, code reviews, issues, and teaming on a shared repo [239] [240]
virtualenv/venv	2–3 hours	virtualenv user guide (activation, isolation) [241]	Isolated Python environments to keep dependencies clean per machine [241]

Ordered learning checklist

- Learn Kivy app lifecycle, KV language, widgets, layouts, and ScreenManager; build a 3-screen skeleton (Login, Dashboard, AI Insights). [220] [221]
- Wire SQLite for Sign-Up/Login and favorites; practice table creation, inserts, selects, and updates from the Kivy UI. [223]
- Pull market data with yfinance (history, multi-ticker download) and cache results; understand when a fallback API like Alpha Vantage is needed due to rate limits. [224] [226]

- Fetch company news from NewsData.io; parse JSON responses and map articles to tickers and recent time windows.[234] [227]
- Apply VADER to headlines; derive per-ticker compound score aggregates and simple positive/negative classifications.[228] [229]
- Use pandas and NumPy to clean/merge price and sentiment features; prepare sliding windows for sequence modeling.[231] [219]
- Implement a basic TensorFlow/Keras LSTM for next-step forecasting; save and reuse the model for in-app inference.[230]
- Visualize price charts and predictions with Matplotlib inside the app's views for clarity and debugging.[233] [232]
- Externalize API keys/config with python-dotenv and environment variables for safe local development.[235]
- Package and test desktop executables on Windows and macOS with PyInstaller; confirm data paths and model loading work.[236]
- Collaborate with Git and GitHub: branch, commit, PR, review, and merge using an agreed branching strategy.[238] [239]

If this table and checklist are followed in order, the necessary skills will be acquired to build the Kivy desktop app with accounts, market/news integrations, VADER sentiment, an LSTM forecaster, and cross-platform packaging.[220] [224] [228] [230] [236]

⁂

1. https://ijsrem.com/download/stock-market-prediction-compliance-using-ai-and-ml/
2. https://www.geeksforgeeks.org/python/how-to-add-authentication-to-your-app-with-flask-login/
3. https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-flask-login
4. https://dev.to/williamsmithh/top-5-free-financial-data-apis-for-building-a-powerful-stock-portfolio-tracker-4dhj
5. https://marketstack.com
6. https://site.financialmodelingprep.com/developer/docs
7. https://finlight.me/blog/news-apis-for-developers-in-2025
8. https://www.intelligenthq.com/best-5-news-data-apis-in-2025/
9. https://archives.journal-grail.science/index.php/2710-3056/article/view/13
10. https://www.semanticscholar.org/paper/56ea5fee05545ca37abcc20314700e6d6927ce68
11. https://www.geeksforgeeks.org/nlp/stock-price-prediction-project-using-tensorflow/
12. https://flet.dev
13. https://www.linkedin.com/pulse/cross-platform-mobile-app-development-python-abdullah-shakir-cp4af
14. https://dev.to/mr_nova/flutter-vs-react-native-vs-kivy-best-framework-for-lightweight-app-development-5dbc

15. https://www.linkedin.com/pulse/continuously-deploying-flask-app-from-github-repository-eurico-paes-3ymlf
16. https://www.youtube.com/watch?v=V858y9L_RJM
17. https://spotintelligence.com/2022/12/16/sentiment-analysis-tools-in-python/
18. https://dev.to/sudo_anuj/automating-flask-app-deployment-with-docker-github-actions-8gh
19. https://bcpublication.org/index.php/BM/article/view/3109
20. https://www.geeksforgeeks.org/machine-learning/stock-price-prediction-using-machine-learning-in-python/
21. https://ijircst.org/DOC/66-stock-price-prediction-using-python-in-machine-learning.pdf
22. https://github.com/Py-Fi-nance/Stock-Price-Forecasting-with-Machine-Learning
23. https://appcost.ai/blog/software-development-timeline-guide
24. https://monday.com/blog/project-management/project-timeline/
25. https://publications.eai.eu/index.php/sis/article/download/4446/2704
26. https://drpress.org/ojs/index.php/fcis/article/download/10208/9933
27. https://www.mdpi.com/2227-7390/11/3/590/pdf?version=1675665621
28. https://www.ccsenet.org/journal/index.php/ijef/article/download/75705/42155
29. https://www.youtube.com/watch?v=94PlBzgeq90
30. https://drlee.io/advanced-stock-pattern-prediction-using-lstm-with-the-attention-mechanism-in-tensorflow-a-step-by-143a2e8b0e95
31. https://dev.to/adeleke123/from-flask-app-to-cicd-pipeline-with-github-actions-docker-hub-aoa
32. https://cswsolutions.com/blog/posts/2023/september/formulating-a-realistic-software-development-timeline-example/
33. https://www.youtube.com/watch?v=CbTU92pbDKw
34. https://www.honeybadger.io/blog/flask-github-actions-continuous-delivery/
35. https://kvytechnology.com/blog/software/estimate-development-time/
36. https://www.tensorflow.org/tutorials/structured_data/time_series
37. https://github.com/orgs/community/discussions/68841
38. https://stepmediasoftware.com/blog/shape-up-methodology/
39. https://www.machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
40. https://github.com/pallets/flask
41. https://www.fuselio.com/blog/the-6-week-sme-software-development-mvp-journey-from-planning-to-post-launch-iteration
42. https://www.kaggle.com/code/faressayah/stock-market-analysis-prediction-using-lstm
43. https://stackoverflow.com/questions/77883901/how-to-use-flask-python-in-github-actions-to-serve-a-web-app
44. https://ijarsct.co.in/Paper22534.pdf
45. https://ieeexplore.ieee.org/document/10031901/
46. https://www.ewadirect.com/proceedings/aemps/article/view/18814
47. https://www.jsr.org/hs/index.php/path/article/view/6070

48. https://al-kindipublisher.com/index.php/jbms/article/view/8609
49. https://www.mdpi.com/2673-4591/56/1/254
50. https://www.mdpi.com/2227-7390/11/5/1130
51. https://ijai.iaescore.com/index.php/IJAI/article/view/23989
52. https://bcpublication.org/index.php/BM/article/download/3098/3051
53. https://www.techscience.com/iasc/v37n2/53230
54. https://www.mdpi.com/2297-8747/25/3/53/pdf
55. http://www.hrpub.org/download/20210630/UJAF15-12222396.pdf
56. https://arxiv.org/ftp/arxiv/papers/2004/2004.01497.pdf
57. https://www.techrxiv.org/doi/full/10.36227/techrxiv.16640197.v1
58. https://www.ijfmr.com/papers/2024/1/12383.pdf
59. https://dx.plos.org/10.1371/journal.pone.0319775
60. https://www.preprints.org/manuscript/202003.0256/v1/download
61. https://annals-csis.org/proceedings/rice2022/drp/pdf/02.pdf
62. https://www.youtube.com/watch?v=1O_BenficgE
63. https://sproutsocial.com/insights/sentiment-analysis-tools/
64. https://www.tigerdata.com/learn/time-series-analysis-and-forecasting-with-python
65. https://www.reddit.com/r/webdev/comments/151zk8y/is_there_any_free_stock_market_api_that_allows/
66. https://assemblyai.com/blog/best-apis-for-sentiment-analysis
67. https://python.plainenglish.io/how-i-built-an-ai-model-that-trades-the-stock-market-while-i-sleep-4501f338d89f
68. https://finnhub.io
69. https://newsapi.ai/blog/best-news-api-comparison-2025
70. https://www.sciencedirect.com/science/article/pii/S2590291124000615
71. https://www.alphavantage.co
72. https://www.edenai.co/post/best-sentiment-analysis-apis
73. https://polygon.io
74. https://finnhub.io/docs/api/news-sentiment
75. https://goldncloudpublications.com/index.php/irjaem/article/view/614
76. https://ijies.net/final-docs/final-pdf/1065.pdf
77. https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0011824500003476
78. https://www.semanticscholar.org/paper/f75662aa67e6c2ffa1337368ebf1be6924cdd7c1
79. https://dl.acm.org/doi/10.1145/3442555.3442561
80. https://www.ijrte.org/portfolio-item/A75800512123/
81. https://resources.today/en/13INOR420.html
82. https://www.semanticscholar.org/paper/02f9c4923bd7563b363ed4dbc367c9445db57b97
83. https://ieeexplore.ieee.org/document/10667693/
84. https://www.ijraset.com/best-journal/the-polyglots-playground-navigating-kmp-flutter-and-react-native-in-cross-platform-ecosystems

85. https://www.ej-eng.org/index.php/ejeng/article/download/2740/1221
86. https://dergipark.org.tr/en/download/article-file/1419794
87. https://www.mdpi.com/2078-2489/15/10/614
88. https://www.e3s-conferences.org/10.1051/e3sconf/202560100022
89. https://ejournal.bsi.ac.id/ejurnal/index.php/ijcit/article/download/9680/pdf
90. https://www.techscience.com/ueditor/files/csse/TSP_CSSE-42-2/TSP_CSSE_22519/TSP_CSSE_22519.pdf
91. https://www.mdpi.com/1424-8220/21/10/3324/pdf
92. https://www.matec-conferences.org/articles/matecconf/pdf/2018/91/matecconf_eitce2018_03044.pdf
93. https://arxiv.org/abs/2409.11667
94. https://ejournal.undiksha.ac.id/index.php/janapati/article/download/60629/26975
95. https://www.freecodecamp.org/news/how-to-setup-user-authentication-in-flask/
96. https://www.netguru.com/blog/python-sentiment-analysis-libraries
97. https://kivy.org
98. https://flask.palletsprojects.com/en/stable/tutorial/database/
99. https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair
100. https://codewave.com/insights/cross-platform-app-development-languages-overview/
101. https://stackoverflow.com/questions/41323397/flask-login-with-sqlite3
102. https://www.bairesdev.com/blog/best-python-sentiment-analysis-libraries/
103. https://www.yesitlabs.com/top-programming-languages-for-mobile-app-development-in-2024/
104. https://www.youtube.com/watch?v=gbTLL6htiPg
105. https://www.ishir.com/blog/122160/the-best-8-python-js-frameworks-for-mobile-development-how-to-use-them.htm
106. https://gist.github.com/jironghuang/24e0577e58844882604c0013407bf606
107. https://www.reddit.com/r/flask/comments/5ldblw/help_with_flasklogin_and_sqlite3/
108. https://ieeexplore.ieee.org/document/11034828/
109. https://ieeexplore.ieee.org/document/9596513/
110. https://bcpublication.org/index.php/BM/article/view/4398
111. https://www.matec-conferences.org/10.1051/matecconf/202338101017
112. https://www.irjmets.com/uploadedfiles/paper//issue_8_august_2023/43978/final/fin_irjmets1693926535.pdf
113. https://www.ewadirect.com/proceedings/ace/article/view/11083
114. https://ijsrem.com/download/stock-market-price-prediction-using-machine-learning-and-deep-learning/
115. http://www.ijcte.org/vol12/1267-G1758.pdf
116. https://aemps.ewapublishing.org/media/9bd1ed43a76d422c97c69dcca2dc3b4e.marked.pdf
117. https://www.ijert.org/research/stock-market-analysis-using-lstm-in-deep-learning-IJERTV9IS040649.pdf
118. https://bcpublication.org/index.php/BM/article/download/3109/3062
119. https://kivy.org/doc/stable/tutorials-index.html

120. https://kivy.org/doc/stable/guide/basic.html
121. https://ranaroussi.github.io/yfinance/
122. https://pieriantraining.com/tensorflow-lstm-example-a-beginners-guide/
123. https://dev.to/ngonidzashe/using-pyinstaller-to-package-kivy-and-kivymd-desktop-apps-2fmj
124. https://newsdata.io/documentation
125. https://vadersentiment.readthedocs.io/en/latest/
126. https://www.tutorialspoint.com/sqlite/sqlite_python.htm
127. https://kivy.org/doc/stable/gettingstarted/intro.html
128. https://www.youtube.com/watch?v=gW4byuP97K4
129. https://www.geeksforgeeks.org/python/kivy-tutorial/
130. https://ranaroussi.github.io/yfinance/reference/index.html
131. https://www.fintut.com/alpha-vantage-api-limits/
132. https://github.com/cjhutto/vaderSentiment
133. https://www.geeksforgeeks.org/python/python-sentiment-analysis-using-vader/
134. https://talent500.com/blog/time-series-forecasting-with-long-short-term-memory-lstm-networks-in-tensorflow/
135. http://pubs.asha.org/doi/10.1044/2022_LSHSS-21-00151
136. https://dl.acm.org/doi/10.1145/3613904.3642752
137. https://journals.lww.com/00024665-200703000-00006
138. https://pubs.aip.org/jasa/article/155/4/2603/3283092/Conducting-high-quality-and-reliable-acoustic
139. https://rrp.nipne.ro/2024/AN76903.pdf
140. https://pubs.asha.org/doi/10.1044/2024_PERSP-24-00179
141. https://www.semanticscholar.org/paper/a1846d7e7ffdc614f706043609e15fd4706d5f11
142. https://arxiv.org/abs/2409.06752
143. https://journal.iaimnumetrolampung.ac.id/index.php/ji/article/view/3240
144. https://wellcomeopenresearch.org/articles/8-419/v1
145. https://kivy.org/doc/master/tutorials-index.html
146. https://kivy.readthedocs.io
147. https://pypi.org/project/kivymd/0.104.0/
148. https://kivyschool.com/pyinstaller-instructions/
149. https://kivy.org/doc/stable/guide-index.html
150. https://kivymd.readthedocs.io
151. https://stackoverflow.com/questions/51515471/creating-an-exe-file-for-windows-using-mac-for-my-kivy-app
152. https://www.youtube.com/watch?v=7gtkrrDJHM0
153. https://github.com/kivymd/KivyMD
154. https://kivy.org/doc/stable/guide/packaging.html
155. https://kivy.org/doc/stable/
156. https://app.readthedocs.org/projects/kivymd/

157. https://kivy.org/doc/stable/guide/packaging-osx.html
158. https://realpython.com/mobile-app-kivy-python/
159. https://www.ijisrt.com/intelligent-clinical-documentation-harnessing-generative-ai-for-patientcentric-clinical-note-generation
160. http://catalyst.nejm.org/doi/10.1056/CAT.23.0404
161. https://academic.oup.com/jamia/article/31/4/975/7606586
162. https://arxiv.org/abs/2402.16667
163. https://dl.acm.org/doi/10.1145/3616855.3635739
164. https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2810364
165. https://dl.acm.org/doi/10.1145/3531146.3533231
166. https://arxiv.org/abs/2308.00675
167. https://journals.lww.com/10.5435/JAAOS-D-23-00474
168. http://www.emerald.com/jd/article/80/7/346-363/1235682
169. https://pypi.org/project/yfinance/
170. https://algotrading101.com/learn/yfinance-guide/
171. https://docs.phidata.com/tools/yfinance
172. https://python-yahoofinance.readthedocs.io/en/latest/api.html
173. https://apipark.com/technews/ekqZjDgp.html
174. https://newsapi.org/docs
175. https://python-yahoofinance.readthedocs.io/en/latest/
176. https://stackoverflow.com/questions/79352540/alpha-vantage-daily-api-rate-limit-reached-shown-every-day
177. https://newsapi.ai/documentation?tab=introduction
178. https://www.interactivebrokers.com/campus/ibkr-quant-news/yfinance-library-a-complete-guide/
179. https://www.alphavantage.co/premium/
180. https://newsapi.org
181. https://www.youtube.com/watch?v=j0sBKAB75oc
182. https://www.alphavantage.co/support/
183. https://www.thenewsapi.com/documentation
184. https://github.com/ranaroussi/yfinance/issues/2376
185. https://www.worldscientific.com/doi/abs/10.1142/9789811215094_0005
186. https://www.ijraset.com/best-journal/integrating-textblob-and-vader-for-dynamic-sentiment-analysis-a-guibased-approach-with-emotion-visualization-and-confidence-assessment
187. https://ojs.bonviewpress.com/index.php/jdsis/article/view/2441
188. https://www.semanticscholar.org/paper/0f4eba3d1785aac3427299a8e775342d2247485f
189. https://doiserbia.nb.rs/Article.aspx?ID=1820-02142400040K
190. https://journal.upgris.ac.id/index.php/asset/article/view/14897
191. https://dl.acm.org/doi/10.1145/3638584.3638675
192. https://link.springer.com/10.1007/s11096-024-01803-0

193. https://www.mdpi.com/2227-9709/11/2/24
194. https://ieeexplore.ieee.org/document/10933201/
195. https://www.mdpi.com/2227-9709/11/2/24/pdf?version=1713874475
196. https://arxiv.org/pdf/2307.14311.pdf
197. http://thesai.org/Downloads/Volume12No7/Paper_30-LSTM_VADER_and_TF_IDF_based_Hybrid_Sentiment.pdf
198. https://shmpublisher.com/index.php/joscex/article/download/193/139
199. http://thesai.org/Downloads/Volume14No3/Paper_20-An_Ensemble_Multi_Layered_Sentiment_Analysis_Model.pdf
200. https://arxiv.org/ftp/arxiv/papers/2209/2209.12604.pdf
201. https://www.mdpi.com/2673-4001/3/2/19/pdf?version=1653648330
202. https://turcomat.org/index.php/turkbilmat/article/download/2375/2083
203. https://www.tandfonline.com/doi/pdf/10.1080/24751839.2021.1874252?needAccess=true
204. http://ijece.iaescore.com/index.php/IJECE/article/download/17498/13057
205. https://www.nltk.org/_modules/nltk/sentiment/vader.html
206. https://pypi.org/project/vaderSentiment/
207. https://hex.tech/templates/sentiment-analysis/vader-sentiment-analysis/
208. https://www.tensorflow.org/tutorials/structured_data/time_series
209. https://docs.python.org/3/library/sqlite3.html
210. https://www.nltk.org/api/nltk.sentiment.vader.html
211. https://www.youtube.com/watch?v=94PIBzgeq90
212. https://docs.python.org/3.9/library/sqlite3.html
213. https://pythonprogramming.net/sentiment-analysis-python-textblob-vader/
214. https://www.machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
215. https://www.geeksforgeeks.org/python/python-sqlite/
216. https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair
217. https://www.geeksforgeeks.org/deep-learning/time-series-forecasting-using-tensorflow/
218. https://www.sqlite.org/docs.html
219. https://pandas.pydata.org/docs/
220. https://kivy.org/doc/stable/tutorials-index.html
221. https://kivy.org/doc/stable/guide/basic.html
222. https://www.youtube.com/watch?v=gW4byuP97K4
223. https://www.tutorialspoint.com/sqlite/sqlite_python.htm
224. https://ranaroussi.github.io/yfinance/
225. https://ranaroussi.github.io/yfinance/reference/index.html
226. https://www.fintut.com/alpha-vantage-api-limits/
227. https://newsdata.io/documentation
228. https://vadersentiment.readthedocs.io/en/latest/

229. https://www.geeksforgeeks.org/python/python-sentiment-analysis-using-vader/
230. https://pieriantraining.com/tensorflow-lstm-example-a-beginners-guide/
231. https://numpy.org/devdocs/user/absolute_beginners.html
232. https://matplotlib.org
233. https://matplotlib.org/stable/index.html
234. https://realpython.com/python-requests/
235. https://www.geeksforgeeks.org/python/using-python-environment-variables-with-python-dotenv/
236. https://dev.to/ngonidzashe/using-pyinstaller-to-package-kivy-and-kivymd-desktop-apps-2fmj
237. https://git-scm.com/doc
238. https://git-scm.com/docs/git
239. https://docs.github.com/get-started/quickstart/hello-world
240. https://docs.github.com
241. https://virtualenv.pypa.io/en/latest/user_guide.html
242. https://arxiv.org/abs/2203.08491
243. https://www.semanticscholar.org/paper/fc294d70337105961dd577b0d772fa9ff5f5812d
244. https://www.semanticscholar.org/paper/8b7a0a6c944027a3d919d486394ef18a508b1bbb
245. https://link.springer.com/10.1007/s10619-022-07418-6
246. https://joss.theoj.org/papers/10.21105/joss.03714
247. http://conference.scipy.org/proceedings/scipy2021/pdfs/fred_reiss.pdf
248. https://www.qeios.com/read/GFSQFL/pdf
249. https://www.frontiersin.org/articles/10.3389/fped.2021.746639/pdf
250. https://academic.oup.com/bioinformatics/article-pdf/32/21/3363/7889719/btw422.pdf
251. https://www.mdpi.com/1422-0067/21/4/1476/pdf
252. http://joss.theoj.org/papers/10.21105/joss.00279
253. https://arxiv.org/pdf/2001.00888.pdf
254. https://pmc.ncbi.nlm.nih.gov/articles/PMC11515437/
255. http://conference.scipy.org/proceedings/scipy2019/pdfs/eric_ma.pdf
256. https://arxiv.org/abs/1604.06783
257. https://pandas.pydata.org
258. https://www.w3schools.com/python/pandas/default.asp
259. https://devdocs.io/pandas~0.25/
260. https://www.reddit.com/r/learnpython/comments/xj5l2i/how_to_understand_the_pandas_documentation/
261. https://pandera.readthedocs.io
262. https://wiki.python.org/moin/NumPy
263. https://docs.python.org/3/library/urllib.request.html
264. https://docs.pandas-ai.com
265. https://github.com/numpy/doc
266. https://requests.readthedocs.io

267. https://github.com/pandas-dev/pandas
268. https://numpy.org/doc/
269. https://pypi.org/project/requests/
270. https://www.pandadoc.com
271. https://www.w3schools.com/python/numpy/default.asp
272. https://www.w3schools.com/python/module_requests.asp
273. https://docs.python.org
274. https://academic.oup.com/bioinformatics/article/36/7/2272/5671693
275. https://doi.curvenote.com/10.25080/CTJE0023
276. https://arxiv.org/abs/2412.04478
277. https://journals.ametsoc.org/view/journals/bams/103/10/BAMS-D-21-0125.1.xml
278. https://ieeexplore.ieee.org/document/9678696/
279. http://biorxiv.org/lookup/doi/10.1101/2021.10.08.463725
280. https://www.semanticscholar.org/paper/ac8f81b7d179fe9f6b1c29349f5f31259ca6d485
281. https://zenodo.org/record/889212
282. https://www.ijisrt.com/intelligent-clinical-documentation-harnessing-generative-ai-for-patientcentric-clinical-note-generation
283. https://dl.acm.org/doi/10.1145/3616855.3635739
284. https://joss.theoj.org/papers/10.21105/joss.00024.pdf
285. https://www.aclweb.org/anthology/2020.acl-main.328.pdf
286. http://joss.theoj.org/papers/10.21105/joss.00250
287. http://openresearchsoftware.metajnl.com/articles/10.5334/jors.ai/galley/26/download/
288. http://arxiv.org/pdf/1910.00279.pdf
289. https://drpress.org/ojs/index.php/fcis/article/download/10367/10083
290. https://onlinelibrary.wiley.com/doi/pdfdirect/10.1002/imt2.115
291. https://genominfo.org/upload/pdf/gi-22079.pdf
292. http://conference.scipy.org/proceedings/scipy2017/pdfs/lindsay.pdf
293. https://pmc.ncbi.nlm.nih.gov/articles/PMC10919210/
294. https://www.w3schools.com/python/matplotlib_intro.asp
295. https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html
296. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html
297. https://www.youtube.com/watch?v=vUSFLs8w_dg
298. https://github.com/matplotlib/matplotlib
299. https://sagemaker.readthedocs.io/en/stable/sagemaker.sklearn.html
300. https://saurabh-kumar.com/python-dotenv/reference/
301. https://www.reddit.com/r/learnpython/comments/1f6tqkx/why_is_the_matplotlib_documentation_so_terrible/
302. https://scikit-learn.org
303. https://pypi.org/project/python-dotenv/

304. https://pypi.org/project/matplotlib/
305. https://devdocs.io/scikit_learn/
306. https://github.com/theskumar/python-dotenv
307. https://devdocs.io/matplotlib/
308. https://github.com/scikit-learn/scikit-learn
309. https://www.dotenv.org/docs/
310. https://www.w3schools.com/python/matplotlib_pyplot.asp
311. https://onlinelibrary.wiley.com/doi/10.1002/smr.2662
312. https://onlinelibrary.wiley.com/doi/10.1002/pds.5740
313. https://ecp.ep.liu.se/index.php/clarin/article/view/727
314. https://journal.uinsi.ac.id/index.php/syamil/article/view/4639
315. https://www.semanticscholar.org/paper/7845cfed08a3c58f367d5a4993248a09940e461a
316. https://scipost.org/10.21468/SciPostPhys.3.2.013
317. https://f1000research.com/articles/9-632/v3
318. https://www.semanticscholar.org/paper/7563f22b61f5fd9d9f39a90c19d21dc4c9893016
319. http://www.tandfonline.com/doi/full/10.1080/01639269.2015.1062586
320. https://www.semanticscholar.org/paper/ee4d8b7d5c9f0a2f09310ff8830d5e2be48dfbd2
321. https://onlinelibrary.wiley.com/doi/pdfdirect/10.1029/2021EA001797
322. https://arxiv.org/abs/2205.14100
323. https://arxiv.org/abs/2208.01317
324. https://pmc.ncbi.nlm.nih.gov/articles/PMC4718703/
325. https://www.theoj.org/joss-papers/joss.00029/10.21105.joss.00029.pdf
326. https://dx.plos.org/10.1371/journal.pbio.3003029
327. https://dl.acm.org/doi/pdf/10.1145/3639478.3640025
328. https://dl.acm.org/doi/pdf/10.1145/3611643.3616288
329. https://pmc.ncbi.nlm.nih.gov/articles/PMC4945047/
330. https://linkinghub.elsevier.com/retrieve/pii/S0164121220300522
331. https://git-scm.com/docs
332. https://docs.github.com/en/get-started/using-git/about-git
333. https://www.kernel.org/pub/software/scm/git/docs/git.html
334. https://docs.github.com/en/get-started/learning-to-code/getting-started-with-git
335. https://realpython.com/python-virtual-environments-a-primer/
336. https://www.gitbook.com
337. https://docs.github.com/en/get-started
338. https://docs.python.org/3/library/venv.html
339. https://git-scm.com/docs/gittutorial
340. https://github.com/Azure-Samples/html-docs-hello-world
341. https://docs.python.org/3/tutorial/venv.html

342. https://www.atlassian.com/git
343. https://github.com/Azure-Samples/python-docs-hello-world
344. https://www.w3schools.com/python/python_virtualenv.asp
345. https://www.w3schools.com/git/
346. https://www.epj-conferences.org/10.1051/epjconf/201921406035
347. https://www.semanticscholar.org/paper/31c2f75ab2dad89789b3147722fd734f1d1527f1
348. https://ieeexplore.ieee.org/document/10486768/
349. http://medrxiv.org/lookup/doi/10.1101/2024.05.08.24307084