A

Project Report on

Stock Price Prediction WebApp

Submitted in partial fulfillment of completion of the course

Advanced Diploma in IT, Networking and Cloud

Submitted by:

Himanshu Prajapati

Mohit Maurya

Surajit Ghosh

Mahender Kumar Bharduwaj

Under Guidance of:

Ram Vaddadi (IBM)

Ankur Mishra (Edunet)

Year 2024

Abstract:

The Stock Price Prediction WebApp is a cutting-edge online platform designed to forecast stock prices with accuracy and reliability. Leveraging advanced machine learning algorithms and real-time market data, this web application provides users with a comprehensive and intuitive interface to analyze and predict stock prices. The webapp's robust architecture integrates data visualization, predictive modeling, and user-friendly features, enabling users to make informed investment decisions. By harnessing the power of data science and web development, this innovative webapp aims to revolutionize the field of stock market prediction, offering a valuable tool for investors, analysts, and financial professionals alike.

Acknowledgement

I would like to extend my sincere gratitude to the entire TECH TITANS team for their unwavering support and guidance throughout the development of the Stock Price Prediction WebApp. Their expertise and dedication have been instrumental in shaping this project into a cutting-edge platform for stock market prediction.

I would also like to acknowledge the contributions of various open-source libraries, frameworks, and tools that have been utilized in the development of this webapp. Their availability and community support have been invaluable in bringing this project to life.

Furthermore, I would like to thank the users who have provided feedback and suggestions during the testing phase of the webapp. Their input has been crucial in refining the platform and ensuring its usability and effectiveness.

Lastly, I would like to acknowledge the vast amount of data and resources available online, which have been instrumental in training and testing the machine learning models used in this webapp. The accuracy and reliability of the predictions made by this platform are a testament to the power of data science and the importance of leveraging real-time market data.

Special Thanks

TECH TITANS team for their technical expertise and guidance

Open-source libraries and frameworks for their contributions to the development of the webapp

Users who provided feedback and suggestions during the testing phase

Online resources and data providers for their role in training and testing the machine learning models.

Team Composition and Workload Division

- **HIMANSHU PRAJAPATI** - Financial analytics, database, data science and machine learning.
- **MOHIT MAURYA -** Software development, Error Handling and authentication.
- **SURAJIT GHOSH** - Focused on data analysis and front-end development.
- **MAHENDER KUMAR BHARDUWAJ** - Develop connections in digital marketing and user engagements.

- ## Introduction to Problem

  Develop an accurate stock price prediction system using deep learning models and deploy it in a production environment using Streamlit

- ## Literature Review

  StockAI is an innovative platform designed to provide insights and analytics for stock market investments. Using advanced algorithms and machine learning models, StockAI helps users make informed decisions by analyzing historical data, predicting market trends, and offering actionable insights. Our mission is to democratize stock market intelligence and make it accessible to both novice and experienced investors. With StockAI, you can track stock performance, analyze market trends, and receive personalized investment recommendations.

- ## Proposed Solution

  1. The system should be able to provide real-time predictions and be scalable to handle a large number of users.
  2. The system should be able to integrate with additional data sources such as news, social media, and economic indicators to enhance its performance.

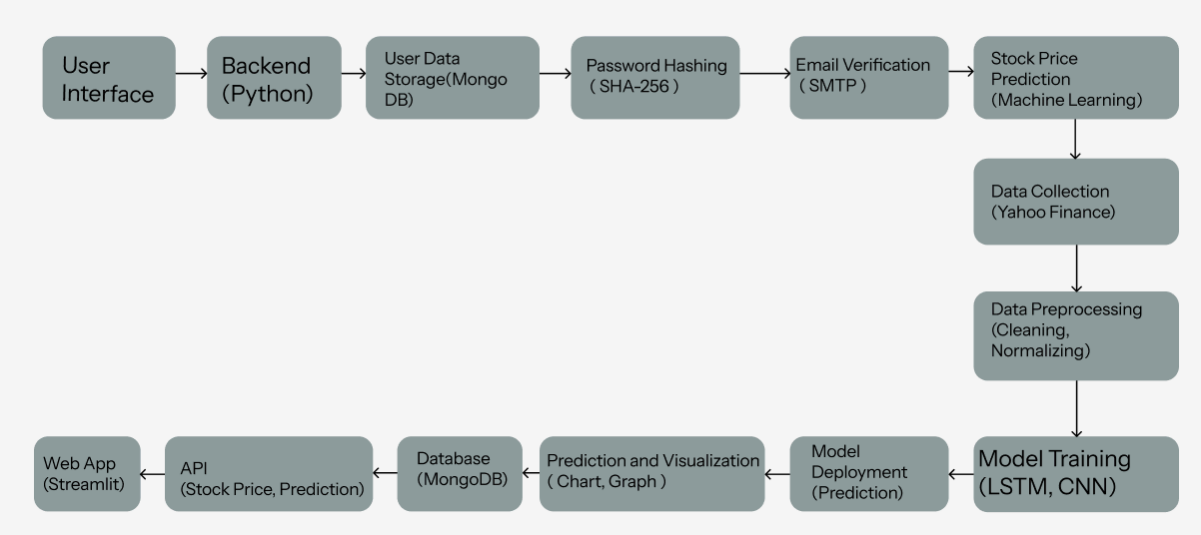
- ## Requirements

5. Technology Stack: TensorFlow, Python, Matplotlib, Seaborn and Streamlit.
6. Hardware: CPU, Memory, Storage and GPU.
7. Software: Operating System, Python, TensorFlow, Streamlit, Matplotlib and Seaborn.
8. Deployment Environment: Streamlit Cloud Platform and Database(Mongo Client).

9. User Requirements

The developed solution is a complex deep learning model that provides predictions of future stock prices. However, it has several limitations, including dependence on high-quality data, model complexity, lack of interpretability, and potential overfitting. Additionally, it may not incorporate domain-specific knowledge, handle rare events, or integrate real-time data. To improve the solution, future developments should focus on incorporating domain knowledge, handling rare events, integrating real-time data, and improving explainability and transparency.

10. Design Documentation



11. Implementation Details

Explore the stocks with the highest growth potential for the next quarter. Stay ahead of the market trends!

12. Testing

The testing strategy for the developed solution includes unit testing, integration testing, system testing, acceptance testing, performance testing, and security testing. A comprehensive set of test cases will be developed to cover various scenarios, and a combination of real-world and synthetic data will be used for testing. Automated testing tools will be used to streamline testing, and manual testing will be used to validate usability and performance. A testing schedule will be developed to ensure adequate testing time and resources.

13. Deployment

Deploy the trained deep learning model into a production environment using Streamlit, where it can analyze real-time stock price data and provide automated stock price predictions for users.

14. Future Scope

- Handling Rare Events: Developing techniques to handle rare events and outliers in the data to improve the solution's robustness and accuracy.
- Scalability and Performance Optimization: Optimizing the solution's performance and scalability to handle a large number of users and high volumes of data.
- Includes indices, mutual funds and etf.
- Create mobile application.

15. Conclusion

In conclusion, a comprehensive testing strategy is crucial to ensure the developed solution meets user requirements and performs as expected. By implementing a structured testing approach, including unit testing, integration testing, system testing, acceptance testing, performance testing, and security testing, we can validate the solution's functionality, performance, and reliability. The testing strategy outlined above provides a thorough framework for testing, including test cases, test data, testing tools, and a testing schedule. By following this strategy, we can identify and address any defects or issues early on, ensuring the solution is robust, efficient, and meets user expectations.

Appendix A

Project Code