Max Kimball

MSDS434

Final Project Reflection

Crypto Sentiment Analysis Project

I developed a cloud-based sentiment analysis service that monitors and predicts cryptocurrency market sentiment. This application analyzes Telegram messages to determine market sentiment and provides forecast predictions for future trends. Some of the current capabilities include sentiment analysis of messages related to crypto in 2021. This also includes interactive visualizations of sentiment trends, 3–7-day sentiment forecasting, an automated deployment pipeline, and basic monitoring and health checks. It will be expanded to include real-time sentiment analysis and correlation of price data.

The key advantages of the app are that the FastAPI provides excellent performance, along with the docker containerization and automated CI/CD which ensures consistent reliable deployment. The CI/CD capabilities make it easy for other developers to manage the application and provide updates simultaneously. The application provides excellent user experience by using a simple API interface and visual charts for intuitive trend analysis with quick response times. The sentiment data can easily be analyzed using Redshift to query results such as analyzing the average sentiment over a year.

The main disadvantages of the application include limited historical data. There are only 3 months of historical data in 2021 and a lot of it was reduced after cleaning the data. There is currently no real-time data being collected or forecasted. The sentiment categories only include positive, negative, neutral, or mixed. There could have been more features added such as real-time price data to make better correlations. There is only basic error handling, limited test coverage, and basic monitoring implementation.

For this application, there are many recommended short term, medium term, and long term improvements that can be made. During the short term, real-time data collection can be implemented from Twitter, Reddit, Telegram, and news articles. A longer history of data would make the model a lot more accurate, and it would be able to forecast data over a longer period of time. Medium term improvements include adding support for more cryptocurrencies, creating a user-friendly web interface, improving forecasting accuracy, adding alerts for significant sentiment changes, A/B testing deep learning models, and making the application more secure. Long term improvements after 6 months could be intergrading with difference trading platforms aside from Binance, adding a retraining pipeline for machine learning models, implementing an advanced analytics dashboard in Tableau, and adding more market data correlation analysis. It could also be expanded to analyze more market data such as stocks.

There could also be a lot more tools added to improve the application. The Twitter API v2 can be used for real-time data, Apache Kafka for stream processing, and MongoDB for more flexible data storage. The infrastructure can also be improved by moving to AWS EC2 or Kubernetes for better container organization. AWS Elastic Beanstalk can be used to simply deploying and managing the application instead of just hosting it on an EC2 instance. This could handle tasks like capacity provisioning, load balancing, and auto-scaling. Some ML/AI improvements can be made, such as implementing BERT for better sentiment analysis. The model can be hosted on SageMaker endpoint and frequently be trained on new data for better results.

Some of the technical risks associated with the application include data quality degradation, limited API rate, scaling issues, and decline of model accuracy. The business risks include API service disruptions, data privacy and security issues, regulatory compliance, and competition from similar services.

This application provides a solid foundation for sentiment analysis but requires improvements to reach its full potential. The proposed roadmap and improvements can balance immediate needs with long-term scalability and reliability. With the proposed improvements, it could cost $300-400 a month by using enhanced AWS services, additional data storage, and API usage costs. The next steps would be to prioritize real-time data collection, implementing comprehensive monitoring and alerts, adding user authentication, and improving forecasting accuracy.