5130Of2022 IWS1 Personal Project Report

Group Members

1. Nagarjuna Kocharla
2. Sahithi Nallani
3. Niharika Chundury

Introduction

At the beginning of the semester, me and my group mates decided on creating a react web app for crypto currency tracking and visualization, where a user would be able to enter his/her transactions, see them on the dashboard and have access to their total portfolio analysis, with total investment, profit/loss ratios and their portfolio outlook into the future, as stated in the project proposal. We have created the web application now that lets users add transactions to their portfolio, and a portfolio analysis button, where users will be able to see the visualization of his/her investments, in terms of profit/loss, total value.

Motivation/Problem and our Approach

The problems, we aim to address with our application are: -

1. There are apps for making investments, and there are apps like Coinstats to track investments, but per my knowledge and research, an app to track investments and analyze the same investment to decide(buy/sell/hold) is lacking. The root of my motivation for the project stems from this. As mentioned above, we aim to create an application that solve the problem of tracking and analysis within the same platform without having to go back and forth.

2. The first problem of designing the tracking system for cryptocurrency would be straight forward, the second part of providing portfolio analysis would be challenging, we would have to research about ML algorithms that would produce the best results against non-stationary data, an algorithm like random Forrest or SVM would help, but also our feature space has to be rich for ideal predictions, for crypto data, features can be 52 week high/low, trading volume, average open price, social media interaction volume(via sentiment analysis) and the output would be price(closing) on a time period in the future(60 day/90 day). We would have to add more features to make the model more robust, but this is a good starting point.

   We ended up going with LSTM or Long Short-Term Memory deep learning model for forecasting, the model was developed by my group mate Sahithi Nallani.

Tools/ Technologies Implemented

For the backend code, I used the flask API framework and build the endpoints with python. For the database, I build it using docker postgresql and docker compose as mentioned in my weekly reports and coingecko API for crypto live prices

I was responsible for building the backend/server-side code for the app. In doing so, I build the following API end points

1. get_transactions – the endpoint is used to retrieve transactions from the database, using a select query, the endpoints return a JSON response, which is used by the frontend to display transactions
2. post_transactions – the endpoint is used to add a transaction that the user wishes to add, I wrote this endpoint to check if the user is adding a buy or sell transaction and based on that check if the transaction is valid (so the user cannot add a transaction to sell 2 bitcoins when he only has 1).
3. get_details_coinwise – the endpoints gets the total coins, total investment in terms of equity and value and live price. I put a check in place where only transactions where status='active' are selected, and ones with 'delete' are ignored. The endpoint returns a JSON response which is used by the frontend code for the portfolio analysis function.

4. delete_transaction – the endpoint is used to remove a transaction, in a case that the user accidentally adds a transaction, I did not remove the transaction from the database, but set a flag called status and set it to delete if the user decides to remove the transaction from his portfolio.

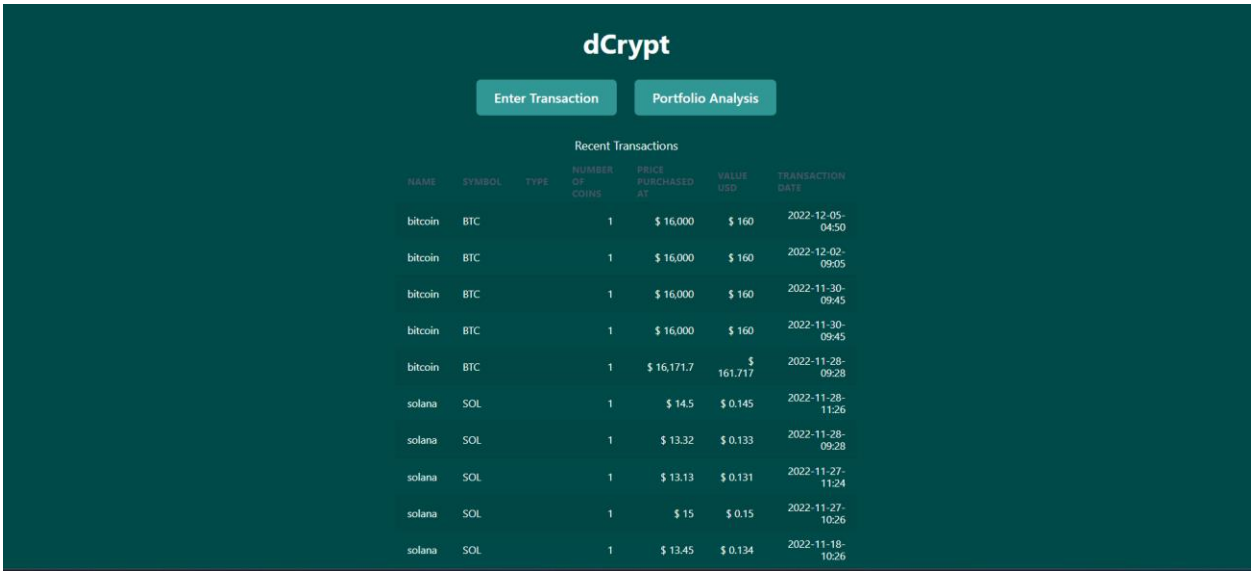
5. get_predictions – the endpoint runs the LSTM model to make bitcoin forecasts for time ranges, of 100 days before today's date to 100 days from todays date. The end point predicts the values which is in normalized form (0 to 1), scales it back to original value (USD) and creates a plot of the forecast with price as Y axis and time as X axis and return that plot and json response

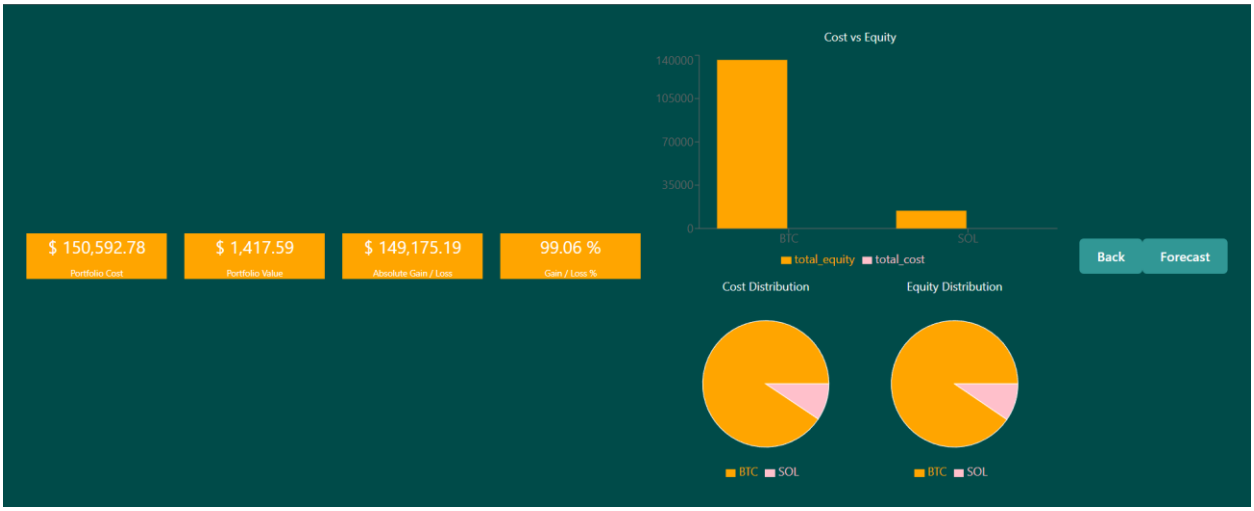
The database for the application was built using Docker, a containerized PostgreSQL database was created using docker compose. And the server-side infrastructure and code was connected to the database using psycopg python library to make PostgreSQL connection pools.

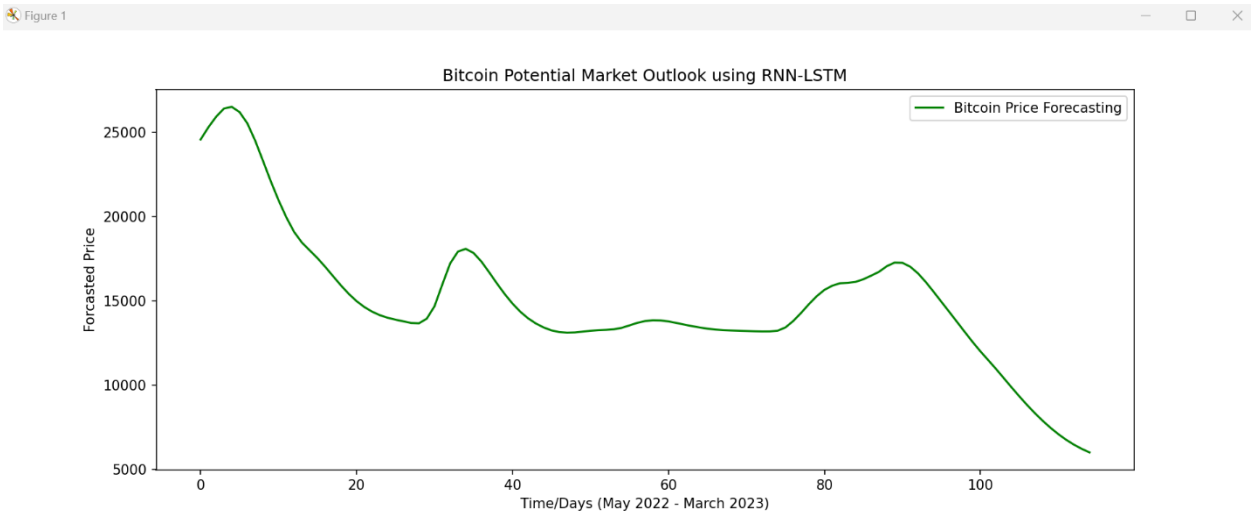
Update – For the database to be available seamlessly, I create a database and AWS and migrated copy of our data there, and changed the server code to connect to AWS database, I included both the connection strings in the code for clarity

Current Progress and Results

We as a team are happy with our progress so far, and our app is doing what we intend, below are the images of our app, in its current state







Flow/Functionality of our Application

When a user opens our app, he will be greeted with the home page which contains 2 buttons,

1. Enter Transaction
2. Portfolio Analysis

Enter Transaction Button

The user can enter his/her transaction either buy or sell buy clicking on the button and entering the name of the coin (bitcoin,solana etc), symbol (BTC,SOL etc), type (buy or sell) purchase price, transaction date, total values and value in usd. And the transaction is saved in our database and is display below the add transaction and portfolio analysis buttons as seen above.

Portfolio Analysis Button

When a user clicks on this button, he/she can see the total investment in all coins aggregated, total investment in each individual coin, the profit/loss ratio, total current value in usd and total cost, both in numerical and visually using bar charts and pie charts. These metrics were aggregated using backend code, where I queried the database for sum of coins, value_usd, total_cost and grouped it by coin name/symbol and returned the values as a json response. The JSON response was fetched by the frontend react Js code to display the values.

When the user is on the portfolio analysis page, he has another option via the forecast button

Forecast Button

On the Portfolio Analysis page, the user will be able to click on the forecast button, below is the process involved in making the forecasts

Model Training/Building

My group mate Sahithi Nallani has developed a Recurrent neural network model using LSTM or long short-term memory, to forecast potential bitcoin prices into the future. The model was trained using metrics like previous day/s open, close, volume, high, low, adj close. The data for these values was downloaded using Yfinance or yahoo finance historic data API from 2014 up until the model build was done, so dec 1. And the data was scrapped and converted into a pandas dataframe, and the columns within the dataframe were normalized (0 to 1 range) for training the RNN model. Once Sahithi completed training and testing the model, it was packaged using model.save.

LSTM model usage

After the model was packaged, I loaded it in the backend code. And made predictions from may 2022 to now () date + 100 days. I had issues with datetime as the model was not accepting the datatime I gave as input, so I had to experiment with different datetime formats, the now() + 100 days was achieved using datetime timedelta. Once the predictions were made, I scaled them back to original bitcoin values using scaler functions and plotted a graph with time as x axis and price as y axis. And this forecast plot was fetched by frontend react code by hitting the get_predictions API endpoint.

When the user clicks on the forecast button, the lstm model gets to work, makes forecast and the user will be able to use the forecast plot.

Conclusion and Future Work

Cryptocurrency has taken off and become a multi trillion-dollar industry in recent years. The investor base has exponentially increase in recent years, and applications like robin hood made it easy. Given the fast-paced nature of crypto, an application to track and visualize one’s investments is essential, along with having an outlook on value of the crypto. Me and my group mates, have created a web app, where users can track and see visualizations of their portfolio along with a button to generate future potential outlook. While we are proud of our work, we intend to make improvements to the app. The potential improvements are

1. The LSTM model must be trained with more rigorous data, like historic indicators like 52 weeks high/low, hashrate, blockchain related data and sentiment analysis to gauge market sentiment
2. Making a native version of the application would be a good idea and improvements
3. Have a function where the user can give consent to the application, to link itself to his/her robin hood or investment account, so all transactions made can be automatically pulled, and analyzed. But extensive security features must be in place to execute this.

Potential Improvements to be made (between Friday Dec 2 submission and Project Show and Tell on Dec 05)

We are left with the following things to complete (now completed)

1. Sometimes, when I click on the back button, and open portfolio analysis it takes time or fails to load. Must investigate this bug and find/fix it
2. We are thinking of changing the background of the app to teal color, right now we just went with the basic black color. Also thinking of changing the visualization(charts) color to something that matches teal background
3. In portfolio analysis, we wish to add a forecast for bitcoin, the model is already built by, and the endpoint is created, we are looking for the ideal place to display it on the application

Update – The above improvements have been made, I created the endpoint for getting predictions and the forecast has been included in the portfolio analysis via forecast button as seen above. And the background was also changed. The above pictures are updated to represent the most recent status of our application.

Distribution of Work

Group members

1.Nagarjuna Kocharla

2.Sahithi Nallani

3.Niharika Chundury

Nagarjuna Kocharla

I was responsible for writing the backend code for the app, all the code written by me is under backend directory, apart from prediction_model.py, with instructions and comments in the code

Sahithi Nallani

was responsible for building and testing the LSTM model to predict bitcoin future price, and the code is under backend in prediction_model.py

Niharika Chundury

was responsible for building and testing frontend code for the app, and the code is under frontend.

References

1. https://flask.palletsprojects.com/en/2.2.x/
2. https://www.psycopg.org/
3. https://www.postgresql.org/docs/current/errcodes-appendix.html
4. https://hub.docker.com/_/postgres
5. https://auth0.com/blog/developing-restful-apis-with-python-and-flask/