A CRYPTOCURRENCY ANALYSIS TOOL BASED ON SOCIAL METRICS

Bill Xu[1], Yu Sun[2]

[1]École Internationale de Montréal, 11 Cm. de la Côte-Saint-Antoine, Westmount, QC H3Y 2H

[2]California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Recent years have witnessed the dramatic popularity of cryptocurrencies, in which millions invest to join the cryptocurrency community or make financial gains [1]. Investors employ many ways to analyze a cryptocurrency, from a purely technical approach to a more utility-centred approach [2]. However, few technologies exist to help investors find cryptocurrencies with bright prospects through social metrics, an equally if not more important viewpoint to consider due to the importance of communities in the space. This paper proposes an application to evaluate cryptocurrencies based on social metrics by establishing scores and models with machine learning and other tools [3]. We verified the need for our application through surveys, applied it to test investment strategies, and conducted a qualitative evaluation of the approach. The results show that our tool benefits investors by providing them with a different lens to view cryptocurrencies and helps them make more thorough decisions.

KEYWORDS

Cryptocurrencies, Machine learning, Analysis, Application

1. INTRODUCTION

The rise in popularity of cryptocurrencies, like Bitcoin, Ethereum and Dogecoin, has attracted much attention from investors worldwide who want a piece of that pie [4]. However, the lack of accessible resources and analytical tools to help investors identify cryptocurrencies with bright prospects left many clueless. There are numerous cryptocurrencies in the market, and without a proper analytical tool, it is challenging for an investor to detect opportunities and make proper commitments [5]. To better understand our target audience and consumer opinion, we conducted a customer outreach survey on online forums and within our communities. In total, it received 20 responses. The majority of the respondents (70%) felt the need for a cryptocurrency analysis tool and stated that they would like to know the future direction of each cryptocurrency (80%). Furthermore, half of the respondents have trouble finding the right cryptocurrency to invest in, and 70% think a cryptocurrency's community is essential to its success. For these reasons, we decided to develop Retrospect, a free and easy-to-use app designed to provide users with the latest cryptocurrency data and analysis based on social metrics, such as Twitter activity or GitHub commits, to determine a crypto's quality and prospects [6]. This solution will allow investors to analyze each cryptocurrency through community activity and an overview of the cryptocurrency's predicted direction. In our survey, 50% of the respondents would use Retrospect as their cryptocurrency analysis tool, with the other half being undecided.

The most important tools for cryptocurrencies analysis in the current market can be easily separated into three categories: charting tools, such as Tradingview, market data tools, such as Coinmarketcap, research reporting platforms, like cryptoresearch.report, network statistics tools, like BitcoinVisuals, and news aggregators, like CryptoPanic. Despite their efficiency and quality, these tools share one major issue: they only provide information to investors and expect them to come to conclusions themselves. However, the market lacks tools that come up with analysis results for the investor, like Retrospect. The current tools assume that investors want to analyze everything and do the job alone, which is often not the case. The rare analytical tools currently on the market offering similar services to Retrospect either fail to implement a consistent model (due to them employing different analysts) or do not directly compete with Retrospect by putting little to no emphasis on social metrics to give out prospects. Furthermore, tools with a plethora of features using highly sophisticated algorithms fail to address fundamental investor concerns, and their complicated interface makes them even less attractive to regular people. These critical issues make the current cryptocurrency analysis apps unattractive to investors.

Our tool, as stated, is a free-to-use application providing cryptocurrency data and analysis based on social metrics. In this paper, our goal is to explain the functionality of our app and our process of determining the perfect model to fit our data and give us prediction results. There are some excellent features of Retrospect. First, the user interface is straightforward, making navigation the slightest concern for our users. Second, our app provides a RETRO-SCORE© for each cryptocurrency, guiding users to understand each cryptocurrency's social state better and helping them make better investment decisions. Third, we provide our users with a market view score, allowing them to determine whether the community is currently optimistic or pessimistic. Fourth, we help our users determine the predicted price movement of the cryptocurrency in the next twenty-four hours based on our model. Compared to most tools available, we help investors do their job. Compared to similar apps, our analytical focus and approach set us apart, and compared to sophisticated tools, our easy UI and features will attract more users [7]. These features are our strengths and will allow us to provide users with the best experience possible. Therefore, we believe that Retrospect has its place in the cryptocurrency world.

In three application scenarios, we demonstrate how the above combination of techniques increases investors' ability to make better investment decisions and confidence. First, we show the usefulness of our approach by surveying our app testers to determine whether our application helps them solve some of their pains as investors: 1) Lack of analytical tools, 2) Hard to use the application, 3) Hard to understand analytical results. Second, we determined the model with the highest accuracy that will fit our data and give us predictions through a series of tests with different quantities of data. Third, we tested our model's "real" accuracy by back testing it and comparing the model's output with what happened.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges thatwe met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusions remarks, as well as points out this project's future work.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Establishing an Analytical Lens

It is challenging to distinguish ourselves from others in finance and cryptocurrency if I do not have a different approach to the problem. That is why one of this project's significant challenges was finding a unique and adequate approach to cryptocurrency analysis [8]. After considering the factors that make a cryptocurrency successful, like usefulness, adoptability, and scalability, we realized that another underutilized factor also plays an essential role in a cryptocurrency's price movement: the community. After sufficient research, I concluded that it was indeed possible to develop a thorough model of a cryptocurrency through social metrics by measuring different factors: Tweet count, Tweets polarity, and GitHub commits.

Furthermore, I remarked that only a few tools considered social metrics, with little emphasis on them. Therefore, I concluded that having a community-focused analytical view is the best choice for a complete and unique model.

2.2. Finding a suitable model for the Data

Another challenge was to find the right way to model the data. Indeed, with so many factors to consider in our final model (tweet count change, commits change, market cap rank, price, and price change 24h). During our experimentation phase, I struggled even to have a model with accuracy or R^2 higher than 0. After countless tests and research, I finally found a suitable model yielding satisfactory results. As shown in section 4, I tested a wide variety of models from the library scikit-learnvi, including linear and random forest regression. This was the biggest challenge I faced while creating this project because of its difficulty and essential role in Retrospect.

2.3. Building the Application

In addition, I also faced problems while building the application on Android Studio. Firstly, I struggled to make our application connect to our application backend server (AWS) due to problems with JSON reading. After fixing this, I also encountered issues with app formatting and functionalities like sort. Finally, there was an issue with the light theme not working on many devices. These issues kept me up at night multiple times to resolve them. Because of the time commitment put into fixing these simple yet complicated issues, the application-building process was also an uncomfortable ride.

3. SOLUTION

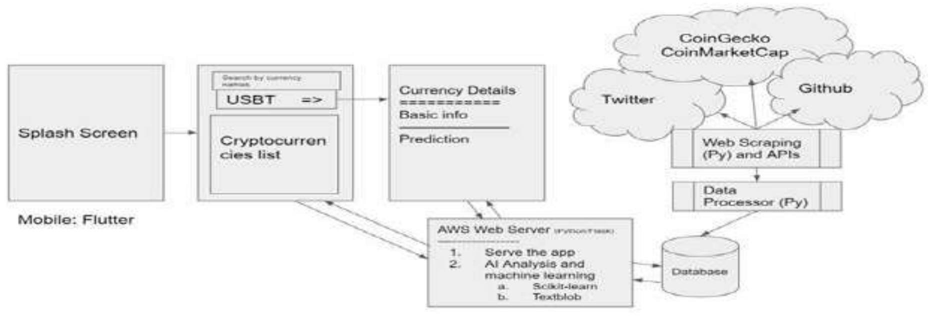
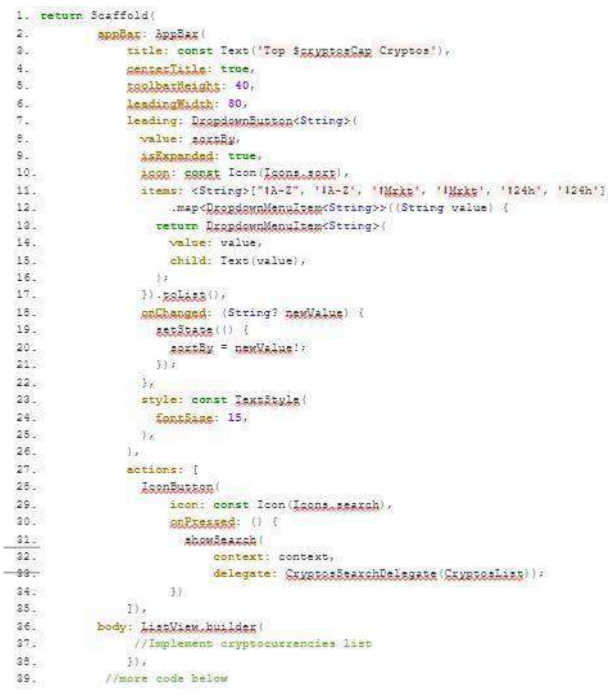


Figure 1. Overview of the solution

Retrospect is a free and easy-to-use app designed to provide users with the latest cryptocurrency data and analysis based on social metrics, such as Twitter activity or GitHub commits, to determine a crypto's quality and future prospects. Retrospect's home page consists of a list of the top 500 cryptocurrencies by market cap. For each cryptocurrency on that page, the user can view basic information, including price, 24 hours change, market cap, and volume. On the home page, users can also search for a specific cryptocurrency and sort the list by alphabetical order, market cap, or 24 hours change. Upon selecting a currency from the list, the user will be redirected to the cryptocurrency's details page. The user can access more information about the coin on that page, including 24 hours high and low, all-time high, symbol, total supply, and market cap rank. The user can then find cryptocurrency analysis in the bottom half of the page. This is where our AI analysis and machine learning come to play. Firstly, we have the RETRO-SCORE©, which determines the quality of a cryptocurrency (the higher, the better). Secondly, we have the market view score, which determines whether the crypto community is currently optimistic or pessimistic about the coin. Thirdly, we see the Tweets count change in the last seven days and GitHub activity change in the last seven days. Fourthly, we have the predicted change in the next 24h, telling the user the predicted price movement of the cryptocurrency in the next 24h. The predicted change ranges from very bearish to very bullish and is determined by our model.

Retrospect's data comes from many different sources and is obtained through 1) web scraping with beautifulsoup4 and 2) APIs [9]. The data is then processed and stored in our Database, where we analyze the cryptocurrency data using scikit-learn and Textblob. Our app then directly obtains the data from our Firebase database from Amazon Web Services.

3.1. Flutter APP

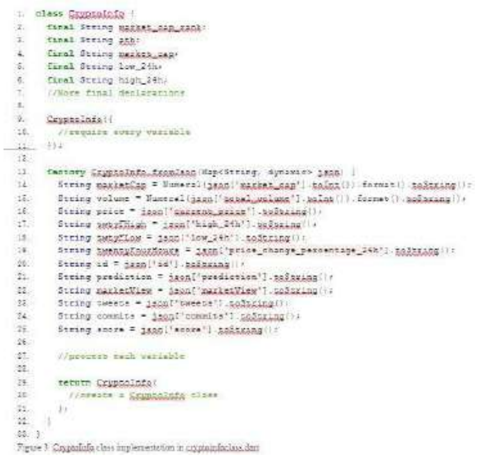


```dart
return Scaffold(
        appBar: AppBar(
            title: const Text('Top $cryptosCap Cryptos'),
            centerTitle: true,
            toolbarHeight: 40,
            leadingWidth: 80,
            leading: DropdownButton<String>(
              value: sortBy,
              isExpanded: true,
              icon: const Icon(Icons.sort),
              items: <String>["↑A-Z", '↓A-Z', '↑Mrkt', '↓Mrkt', '↑24h', '↓24h']
                  .map<DropdownMenuItem<String>>((String value) {
                return DropdownMenuItem<String>(
                  value: value,
                  child: Text(value),
                );
              }).toList(),
              onChanged: (String? newValue) {
                setState(() {
                  sortBy = newValue!;
                });
              },
              style: const TextStyle(
                fontSize: 15,
              ),
            ),
            actions: [
              IconButton(
                  icon: const Icon(Icons.search),
                  onPressed: () {
                    showSearch(
                        context: context,
                        delegate: CryptosSearchDelegate(CryptosList));
                  })
            ]),
        body: ListView.builder(
             //Implement cryptocurrencies list
            }),
      //more code below
```

Figure 2. Home page code snippet in main.dart

The flutter app was implemented in Android Studio using Dart. The flutter app is a summary of different files:

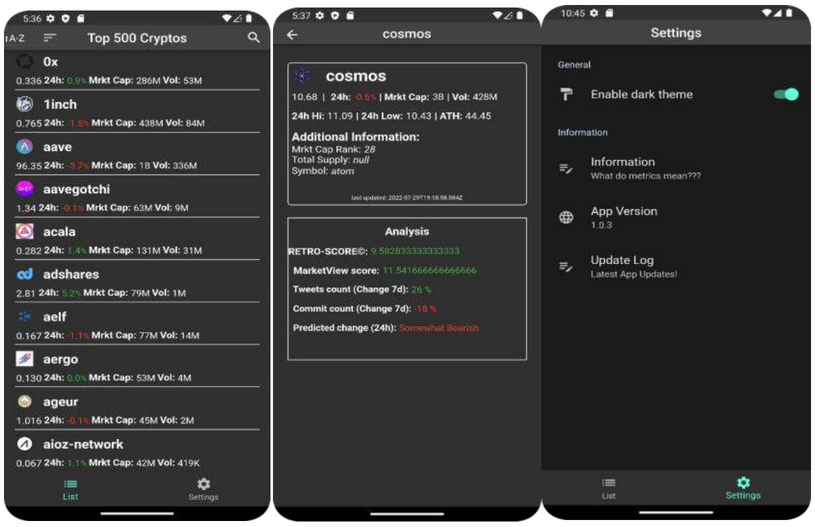
- main.dart contains the implementation for the home page with the cryptocurrencies list and the settings page. The application also fetches data from AWS in this file. Most of the app was implemented by myself, but I used GetX's for dark theme and a Settings_ui's settings template.
- detailspage.dart contains the code for the app's details page, which includes basic informationand the application's analysis results.
- cryptosearchdelegate.dart implements the search bar for the application.
- information.dart and updatelog.dart contain the code for the app's information and update logrespectively.
- cryptosinfoclass.dart declares the class used to convert passed data from AWS from JSON to ausable class.



```
class CryptoInfo {
  final String market_cap_rank;
  final String ath;
  final String market_cap;
  final String low_24h;
  final String high_24h;
  //More final declarations

  CryptoInfo({
    //require every variable
  });

  factory CryptoInfo.fromJson(Map<String, dynamic> json) {
    String marketCap = Numeral(json['market_cap'].toInt()).format().toString();
    String volume = Numeral(json['total_volume'].toInt()).format().toString();
    String price = json['current_price'].toString();
    String twtyFHigh = json['high_24h'].toString();
    String twtyFLow = json['low_24h'].toString();
    String twentyFourHours = json['price_change_percentage_24h'].toString();
    String id = json['id'].toString();
    String prediction = json['prediction'].toString();
    String marketView = json['marketView'].toString();
    String tweets = json['tweets'].toString();
    String commits = json['commits'].toString();
    String score = json['score'].toString();

    //process each variable

    return CryptoInfo(
      //create a CryptoInfo class
    );
  }
}
```

Figure 3 CryptoInfo class implementation in cryptoinfoclass.dart

Figure 3. CryptoInfo class implementation in cryptoinfoclass.dart



5:36

A-Z Top 500 Cryptos

0x
0.336 24h: 0.9% Mrkt Cap: 286M Vol: 53M

1inch
0.765 24h: -1.5% Mrkt Cap: 438M Vol: 84M

aave
96.35 24h: -3.7% Mrkt Cap: 1B Vol: 336M

aavegotchi
1.34 24h: -0.1% Mrkt Cap: 63M Vol: 9M

acala
0.282 24h: 1.4% Mrkt Cap: 131M Vol: 31M

adshares
2.81 24h: 5.2% Mrkt Cap: 79M Vol: 1M

aelf
0.167 24h: -1.1% Mrkt Cap: 77M Vol: 14M

aergo
0.130 24h: 0.0% Mrkt Cap: 53M Vol: 4M

ageur
1.016 24h: -0.1% Mrkt Cap: 45M Vol: 2M

aioz-network
0.067 24h: 1.1% Mrkt Cap: 42M Vol: 419K

List Settings

5:37

cosmos

cosmos
10.68 | 24h: -0.6% | Mrkt Cap: 3B | Vol: 428M
24h Hi: 11.09 | 24h Low: 10.43 | ATH: 44.45

Additional Information:
Mrkt Cap Rank: *28*
Total Supply: *null*
Symbol: *atom*

last updated: 2022-07-29T19:18:58.084Z

Analysis
RETRO-SCORE©: 9.582833333333333
MarketView score: 11.541666666666666
Tweets count (Change 7d): 26 %
Commit count (Change 7d): -18 %
Predicted change (24h): Somewhat Bearish

10:45

Settings

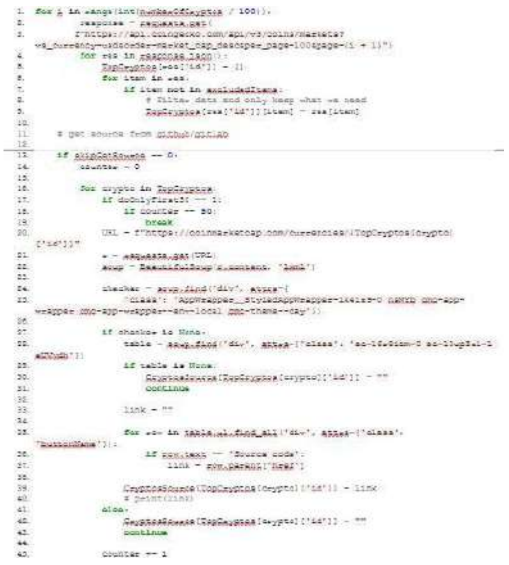
General
Enable dark theme

Information
Information
What do metrics mean???
App Version
1.0.3
Update Log
Latest App Updates!

List Settings

Figure 4. Flutter app overview

3.2. Data Fetching



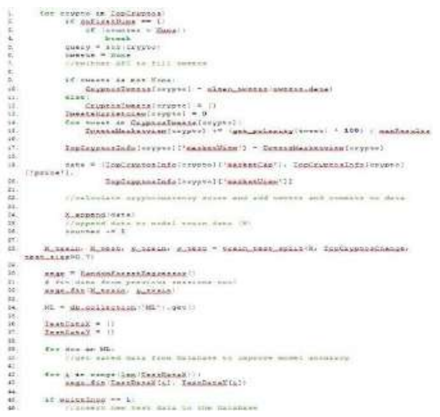
```
for i in range(int(numberOfCryptos / 100)):
        response = requests.get(
        f"https://api.coingecko.com/api/v3/coins/markets?
vs_currency=usd&order=market_cap_desc&per_page=100&page={i + 1}")
        for res in response.json():
            TopCryptos[res["id"]] = {}
            for item in res:
                if item not in excludedItems:
                    # Filter data and only keep what we need
                    TopCryptos[res["id"]][item] = res[item]

    # get source from github/gitlab

    if skipGetSource == 0:
        counter = 0

        for crypto in TopCryptos:
            if doOnlyFirst50 == 1:
                if counter == 50:
                    break
            URL = f"https://coinmarketcap.com/currencies/{TopCryptos[crypto]
['id']}"
            r = requests.get(URL)
            soup = BeautifulSoup(r.content, 'html')

            checker = soup.find('div', attrs={
                'class': 'AppWrapper__StyledAppWrapper-ik4lx5-0 gSNYb cmc-app-
wrapper cmc-app-wrapper--env-local cmc-theme--day'})

            if checker is None:
                table = soup.find('div', attrs={"class": 'sc-16r8icm-0 sc-10up5z1-1
eUVvdh'})
                if table is None:
                    CryptosSource[TopCryptos[crypto]['id']] = ""
                    continue

                link = ""

                for row in table.ul.find_all('div', attrs={'class':
'buttonName'}):
                    if row.text == 'Source code':
                        link = row.parent['href']

                CryptosSource[TopCryptos[crypto]['id']] = link
                # print(link)
            else:
                CryptosSource[TopCryptos[crypto]['id']] = ""
                continue

            counter += 1
```

Figure 5. Python code snipped to get data from CoinGecko API and webscrape CoinMarketCapfor source code

The data fetching code is written in python and uses many libraries to extract data. The process is as follows:

1) Use CoinGecko's API to get individual cryptocurrency data
2) Webscrape CoinMarketCap to obtain the source code of each coin by forcing the URL (adding the crypto name at the end of https://www.coinmarketcap.com/currecencies/{cryptoname here})
3) Using the newly acquired source code to obtain GitHub commit count for each cryptocurrencyusing their API
4) Using Twitter's API to obtain the tweet count of each cryptocurrency
5) Storing all obtained data on Google Firebase Firestore's Database [10]

3.3. Data analysis



```
    for crypto in TopCryptos:
        if doFirstNine == 1:
            if (counter > Nine):
                break
        query = str(crypto)
        tweets = None
        //twitter API to fill tweets

        if tweets is not None:
            CryptosTweets[crypto] = clean_tweets(tweets.data)
        else:
            CryptosTweets[crypto] = []
        TweetsSentiment[crypto] = 0
        for tweet in CryptosTweets[crypto]:
            TweetsSentiment[crypto] += (get_polarity(tweet) * 100) / maxTweets

        TopCryptosInfo[crypto]['marketView'] = TweetsSentiment[crypto]

        data = [TopCryptosInfo[crypto]['marketCap'], TopCryptosInfo[crypto]
['price'],
                TopCryptosInfo[crypto]['marketView']]

        //calculate cryptocurrency score and add tweets and commits to data

        X.append(data)
        //append data to model train data (X)
        counter += 1

    X_train, X_test, y_train, y_test = train_test_split(X, TopCryptosChange,
test_size=0.7)

    regr = RandomForestRegressor()
    # fit data from previous sessions too!
    regr.fit(X_train, y_train)

    ML = db.collection('ML').get()

    TestDataX = []
    TestDataY = []

    for doc in ML:
        //get saved data from Database to improve model accuracy

    for i in range(len(TestDataX)):
        regr.fit(TestDataX[i], TestDataY[i])

    if writeInDB == 1:
        //insert new test data in the Database
```

Figure 6. Python code

The data analysis code is also written in python. Let us view each part individually:

1) Get Tweet count and Commit count data from Database and save it under each cryptocurrencyas a percentage change compared to the last seven days.
2) Obtain 10 tweets from each cryptocurrency using Twitter's API
   a. clean them (using Word Net Lemmatizer from scikit-learn)
   b. get the polarity of these 10 tweets and calculate the score from -100 to 100 (market view score)
3) Calculate the RETRO-SCORE© for each cryptocurrency:

   $R=0.5\times MV+0.24\times T+0.24\times C+0.2\times(501-MC)$

   Where:

   R: RETRO-SCORE©
   MV: market view score
   T: Tweets change % 7d
   C: Commits change % 7d
   MC: market cap rank

4) Append each cryptocurrency's market cap rank, price, market view score, tweets count % change last seven days, and commits count % change last seven days as a list to the Random Forest Regressor model's X-axis.
5) Append each cryptocurrency's price change % 24h to the model's Y-axis
6) Obtain train data for the model
7) Fit newly obtained data and past data into Random Forest Regressor
8) Save new train data to the Database
9) Plug current data into the model to obtain predictions
10) Save model results to Database for each cryptocurrency

3.4. AWS and Flask Server

The Flask Server is the simplest part. Every time a user would make a request through AWS, the server would simply return each cryptocurrency's data and analysis from the Database as a JSON to the user through port 5000. If the Database is currently updating, the server will return{"refreshing_data":"please wait"}, and the app will wait for the Database to complete to execute. Furthermore, data fetch, and analysis is implemented in the AWS server script and run every twohours to put the user to date.

4. EXPERIMENT

4.1. Experiment 1

Retrospect solves the major pains cryptocurrency investors experience: 1) Lack of analytical tools, 2) Hard to use the application, and 3) Hard to understand analytical results. Our solution is a free-to-use cryptocurrency analysis tool that provides investors with a cryptocurrency quality score and market view score while predicting future price movements for the currency. Retrospect's user interface is smooth and easy to use, thanks to its design and easy navigation. Furthermore, our analytical results are shown in a fashion that even people with no prior experience in cryptocurrency investing can understand. To validate these claims, we have conducted a survey of our early testers to get their opinion of the product.

The results of the tester survey are as follows: 80% of respondents agree that the user interface is easy and straightforward to use (20% strongly agree), 80% of respondents think that accessing a cryptocurrency is simple, 80% of respondents agree that the analysis is easy to read and understand, and 100% of respondents think that the overall experience is smooth. The results of this survey further demonstrate that Retrospect can solve current investors' problems.

4.2. Experiment 2: Establishing the best model

Predicting data involves having a well and functional model. To find the ideal model to fit our data, I have conducted tests on different models to obtain the best possible model accuracy. The models I have tested are Linear Regression, Random Forest Regression, Lasso Regression, Elastic Net Regression, and Stochastic Gradient Descent Regression. I have conducted 4 tests in total. The first 2 tests are the model accuracy (R2 since we are using regressions) of each model on the first data set (no stored previous data was plugged into the model). The next test is the model accuracy (R2) of each model on the first data and past stored data (5 packages of saved data from 5 different periods). We can observe that past data help improve the model's accuracy.

The final 2 tests are the model accuracy of each (R2) of each model on the first data and stored data (10 packages).

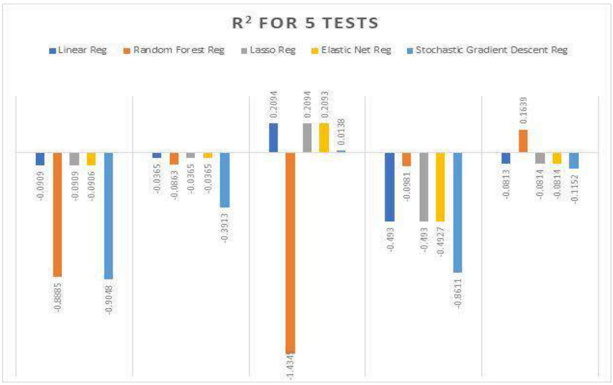
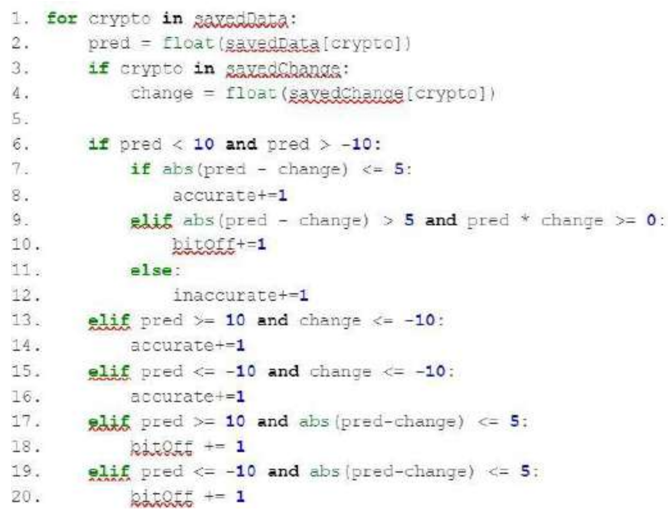


Figure 7. R^2 of each model

The reader should be as surprised as I am. We observe that Linear Regression, Random Lasso Regression, and Elastic Net Regression are more accurate when little data is used. The same case applies to a moderate amount of data to fit a model. However, when more test data is loaded into the model (2000+ test data from different packages from different times), Random Forest Regression seems to hold up better. The Stochastic Gradient Descent Regression is the worst performing model overall. Since we will be conducting large-scale machine learning models with a large amount of train data, Random Forest Regression is our best choice as model. However, if we were to analyze a smaller set of data, Linear, Lasso, and Elastic Net regressions seem to be a better fit.

4.3. Experiment 3: Model Accuracy

We have established the model for our cryptocurrency prediction analysis, which is a Random Forest Regression. We have chosen it thanks to its ability to maintain a higher R2 with more data being plugged into it. Now, we will conduct an analysis to verify the model's accuracy in "real world application." To do this, we will save the model's prediction analysis for one day and check the next day if the 24h price change corresponds to the change predicted by the model. We will check if the price change corresponds to the prediction tier our model has established (very bearish < -10%, bearish < -5%, somewhat bearish < 0%, neutral = 0%, somewhat bullish > 0%, bullish > 5%, very bullish > 10%). We will be conducting this experiment 5 times at irregular time intervals. Figure 9 shows the experiment's results. The model is accurate if the price change is in the tier the model has placed. The model is a bit off if the price change and the predictions are both positive or negative, and the price change is not in the right tier, but 5-10% off (e.g., the model placed it somewhat bullish, but it is bullish). The model is inaccurate if it does not satisfy the 2 conditions above.



```python
for crypto in savedData:
    pred = float(savedData[crypto])
    if crypto in savedChange:
        change = float(savedChange[crypto])

    if pred < 10 and pred > -10:
        if abs(pred - change) <= 5:
            accurate+=1
        elif abs(pred - change) > 5 and pred * change >= 0:
            bitOff+=1
        else:
            inaccurate+=1
    elif pred >= 10 and change <= -10:
        accurate+=1
    elif pred <= -10 and change <= -10:
        accurate+=1
    elif pred >= 10 and abs(pred-change) <= 5:
        bitOff += 1
    elif pred <= -10 and abs(pred-change) <= 5:
        bitOff += 1
```

Figure 8. Python code to evaluate each output's accuracy

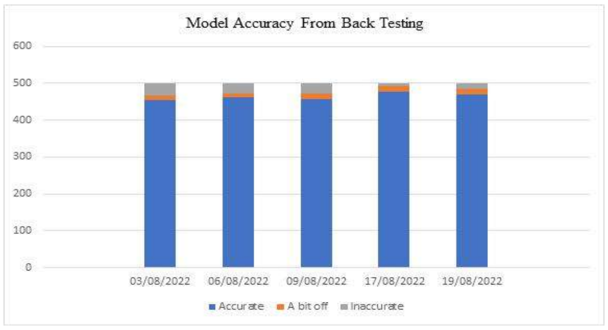


Figure 9. Model Accuracy

After analyzing the results, we conclude that the real accuracy of our model is around 92.8%, which is almost 6x higher than the model's R2 and enough to justify its use.

From Experiment 1, we have successfully concluded that our app help solve the problems our potential users might have: tool availability, usability, and understandability through a survey for our initial testers. From Experiment 2, we have established the ideal model for our needs, Random Forest Regression, thanks to its ability to remain accurate with more data plugged into it. Although the accuracy of our prediction is not top-notch, we offset the margin of error by a significant amount by indicating whether we are bullish or bearish on the currency instead of indicating the predicted % change, which is the result of our data. That is why the "real" model accuracy is so high, as shown in Experiment 3. This concludes the Evaluation part.

The first related work is written by Johan Bollen, Huina Mao, and Xiao-Jun Zeng and tries to predict the stock market using Twitter mood. Using different libraries for sentiment analysis, like Opinion Finder and a mathematical model, the team concluded that a relationship exists between sentiment on Twitter and the price change, stating that sentiment is reflected after 3 to 4 days. Like my work, our model takes in the sentiment of tweets for a financial product to make predictions. However, our model vastly differs, and my model takes in more variables than just tweets. This work supports the idea that a relationship exists between financial instruments and people's reactions.

The second related work, written by Ross C. Phillips and Denise Gorse, discusses cryptocurrency price drivers using a Wavelet coherence analysis. This work seeks to "demonstrate how factor relationships are prone to strengthen and weaken their correlation with price as cryptocurrency goes through different market regimes." In this work, the authors fetched data from 2018 and back to each cryptocurrency's creation date and only analyzed four cryptocurrencies. In my work, I only used the most recent data to make the model more sensitive to the present. Both works take into consideration social metrics, but M. C. Phillips and M. Gorse's work considers Reddit activity while my work considers Twitter activity. Other metrics used in the related work include Wikipedia and Google searches. The objective of our works is also different. The related works consider the change in the correlation of cryptocurrency price drivers during different market regimes, while my work considers the future price change of the cryptocurrency.

The third related work is written by Stuart Colianni, Stephanie Rosales, and Michael Signorotti and aims to create a trading algorithm using Twitter sentiment analysis. The algorithm did surprisingly well by reporting an accuracy of 95% day-to-day and 76.23% hour-to-hour. Like my work, this work considers tweets polarity as a factor to plug in the model. However, instead of predicting the percentage change for the cryptocurrency, the related work's model, logistic regression, only predicts whether the cryptocurrency will increase or decrease in the next hour or day. Also, only data from Bitcoin was collected, so the model may not be a fit for every cryptocurrency. Nevertheless, the accuracy of the model is still impressive.

5. RELATED WORK

Yuanyi Chen presented a system to remote control Android Mobile Phone by using a computer [11]. In the paper, the author explained that developers need to understand and identify the relationship between the four components, active page, service, content provider and broadcast receiver, of the Android system in order to create a remote control application. Also, the author described how the remote control system works from PC device to the server and mobile device. Our application has a similar approach. We use Wifi to send the information from one device to the other. However, our app uses another mobile device as a controller instead of PC, which makes the controller device be more efficient at the time to help another user since most of the people carry the mobile device.

Sørensen H. et al presented a wireless system to share screens in video calls [12]. They proposeda system that can share both digital content as well as physical artifacts in a video call. Our app is similar to this system, our system mirroring the screen in real time. However, our system is notfor a video call and not only for screen share; it also provides remote control.

Bi L. et al proposed a system to remote control power point play in computers without installing any program in mobile devices [13]. It uses Java Native Interface (JNI) technology to control the windows system's function. In our research, we use Android Studio that uses JNI in order to compile our code. As different from this paper, we control the screen of other mobile devices to provide help instead of remote control power point play.

6. CONCLUSIONS

In this paper, I have proposed an application to help investors analyze cryptocurrencies and conduct a thorough analysis of their chosen currency. This application solves the major pains investors currently have with the crypto space: lack of analytical tools and difficulty in using current tools. I have shown a demand for my app through online surveys and by explaining the recent surge in popularity of cryptocurrencies. I have outlined the challenges I faced during the creation of Retrospect. I have explained how Retrospect functions and how I built it using python 3, dart, and flutter through Android Studio, Google Firebase, AWS, and different python packages [14]. I have also conducted three experiments to validate the app's utility and model's accuracy. The experiment results show that the app effectively solves the challenges investors are facing. Furthermore, it shows the app's usefulness thanks to its prediction accuracy. Retrospect will be a free-to-use app listed on the google play store and the apple store that will be available to any country.

Although Retrospect solved cryptocurrency investors' problems, its model's R^2 still has a long way to go to become accurate. Indeed, it would be optimal if the model's accuracy never reaches the negative ground. By improving the model accuracy, we can come to better conclusions and simplify investors' jobs even further. Therefore, I aim for 99% accuracy. Furthermore, optimizations can be made to the app's model processing speed. Due to Twitter's API hard rate cap, the model database takes 15 minutes to update. If we break through this time barrier, the app can update more often than two hours.

In the future, I plan to improve the model's accuracy by trying out more models or my mapping my data differently. Furthermore, I could find a solution to Twitter's API cap and not wait for 15 to get my data ready [15]. I also plan to improve the app's user interface and features to make it a better experience. These are the elements I wish to work on in the future.

REFERENCES

[1] Chan, Stephen, et al. "A statistical analysis of cryptocurrencies." Journal of Risk and Financial Management 10.2 (2017): 12.
[2] Phillip, Andrew, Jennifer SK Chan, and Shelton Peiris. "A new look at cryptocurrencies." Economics Letters 163 (2018): 6-9.
[3] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.
[4] Vranken, Harald. "Sustainability of bitcoin and blockchains." Current opinion in environmental sustainability 28 (2017): 1-9.
[5] Qadan, Mahmoud, David Y. Aharon, and Ron Eichel. "Seasonal and calendar effects and the price efficiency of cryptocurrencies." Finance Research Letters 46 (2022): 102354.
[6] Kalliamvakou, Eirini, et al. "The promises and perils of mining github." Proceedings of the 11th working conference on mining software repositories. 2014.
[7] Lenz, Eva Maria, and Ian D. Wilson. "Analytical strategies in metabonomics." Journal of proteome research 6.2 (2007): 443-458.
[8] Alexander, Carol, and Michael Dakos. "A critical investigation of cryptocurrency data and analysis." Quantitative Finance 20.2 (2020): 173-188.
[9] Glez-Peña, Daniel, et al. "Web scraping technologies in an API world." Briefings in bioinformatics

15.5 (2014): 788-797.
[10] Chatterjee, Nilanjan, et al. "Real-time communication application based on android using Google firebase." Int. J. Adv. Res. Comput. Sci. Manag. Stud 6.4 (2018).
[11] Chen, Yuanyi. "Research on Application System of Remote-Control Computer of Android Mobile Phone." Journal of Physics: Conference Series. Vol. 1992. No. 2. IOP Publishing, 2021.
[12] Sørensen, Henrik, et al. "Wireless smart phone mirroring in video calls." IFIP Conference on Human-Computer Interaction. Springer, Cham, 2015.
[13] Bi, Lingyan, et al. "Design and application of remote control system using mobile phone with JNI interface." 2008 International Conference on Embedded Software and Systems Symposia. IEEE, 2008.
[14] Esmaeel, Hana R. "Apply android studio (SDK) tools." International Journal of Advanced Research in Computer Science and Software Engineering 5.5 (2015).
[15] Pfeffer, Jürgen, Katja Mayer, and Fred Morstatter. "Tampering with Twitter’s sample API." EPJ Data Science 7.1 (2018): 50.