UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF FINANCE AND BANKING

❖❖❖

END OF TERM REPORT

PACKAGED SOFTWARE APPLICATION

FOR FINANCE 1

Topic:

Using the ARIMA model to forecast the MANA token's value in the future.

Instructors: Ngô Phú Thanh

Student: Nguyễn Đức Minh Tấn

Student ID: K194141745

Ho Chi Minh City, January 25, 2021.

Table of contents.

1. Abstract.

Currently, cryptocurrency is becoming a new trend in the 4.0 technology era with the strong explosion of Blockchain applications. In addition to the famous and long-standing cryptocurrencies like Etherum (ETH), Bitcoin (BTC), or even Binance (BNB), the community is currently interested and highly appreciate the cryptocurrencies' potential of NFT or Metaverse projects. In which, Decentraland is one of the prominent coins in this field. In this report, the author uses the Auto-ARIMA model with data from 2017 to the end of 2021 to analyze and predict the development trend of this cryptocurrency. Research results show that the MANA have stationarity attribute and its' future value can strongly move positively.

Keyword: Cryptocurrency, Metaverse, ARIMA, forecasting.

2. Reason for choosing the topic.

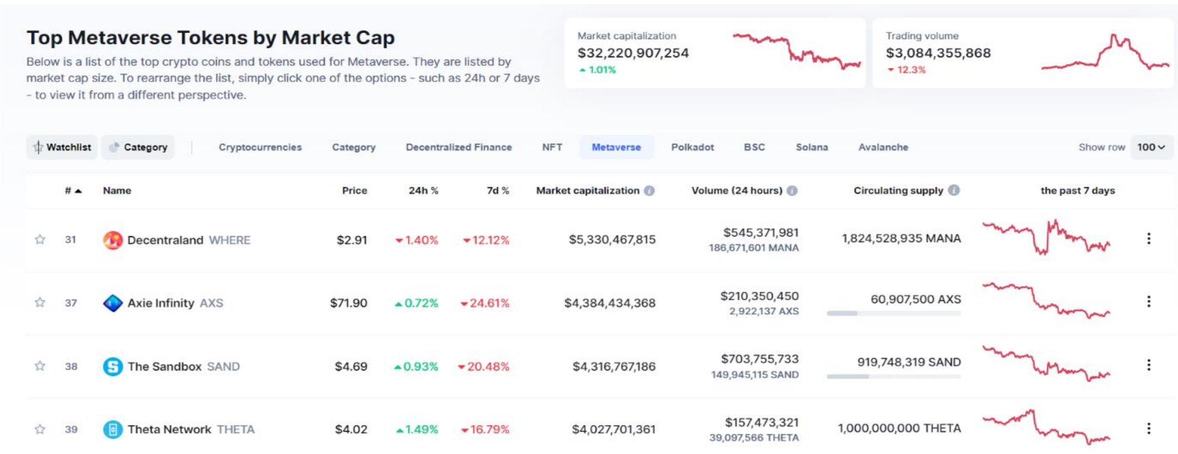
2.1. Reason for choosing the MANA asset.

During the Covid-19 pandemic, complicated developments have disadvantage affected on Vietnam's market economy relatively seriously. However, the investment trend in the field of securities Thu Hường (2021) and cryptocurrency market VTV Digital (2021) became the most popular investment trend during this period in our country. However, cryptocurrencies have just entered to the investment pofolio of customers in Vietnam, it has attracted a lot of attention from investors while there are still controversial opinions about it.

At this time, people gradually have to get used to interacting with each other through screens to ensure government directives, protect the health of themselves and their families. Thanks to that, the Decentraland project with the token MANA has been sought and invested by many people and is considered to have the most potential for development in the future cryptocurrency world. In addition, on October 28, 2021, billionaire Mark Zuckerberg - Facebook's CEO announced to change Facebook's name to Meta, this changes to focus on building a "Metaverse" world. The virtual field is expected to be the "new chapter" of the mobile Internet, which has reinforced the potential of this "virtual real estate" project (Nguyen Toan, 2021). Metaverse is a concept of a continuous, online 3D universe that combines various virtual spaces. Metaverse will allow users to meet, socialize, play, and work together in these 3D spaces.

Decentraland coin (MANA) works on Blockchain technology. Therefore, the operation of the Mana platform is entirely determined by the user. They can shop for land, items, pets, etc. on this platform. Lands on Mana has a permanent ownership value. You unleash your content creation on the land without any worries. Moreover, the content is completely dependent, from 3D to various games. Recently, a virtual real estate NFT of 259 parcels of land in Decentraland was sold for over $900,000 (Matthew De Saro, 2021). Gradually the virtual universes will interact with each other, players can buy and sell virtual goods from different games and universes on the markets. Hence, users can sell their virtual land in the world of Decentraland and use that money to buy game skins in the Fortnite world, for example. Otherwise, MANA token is the project having the largest market capital in the Metaverse field recently.

Source: Coinmarketcap



Top Metaverse Tokens by Market Cap

Below is a list of the top crypto coins and tokens used for Metaverse. They are listed by market cap size. To rearrange the list, simply click one of the options - such as 24h or 7 days - to view it from a different perspective.

Market capitalization $32,220,907,254 ▲1.01%

Trading volume $3,084,355,868 ▼12.3%

Watchlist | Category | Cryptocurrencies | Category | Decentralized Finance | NFT | Metaverse | Polkadot | BSC | Solana | Avalanche | Show row 100

#	Name	Price	24h %	7d %	Market capitalization	Volume (24 hours)	Circulating supply	the past 7 days
31	Decentraland WHERE	$2.91	▼1.40%	▼12.12%	$5,330,467,815	$545,371,981 186,671,601 MANA	1,824,528,935 MANA	
37	Axie Infinity AXS	$71.90	▲0.72%	▼24.61%	$4,384,434,368	$210,350,450 2,922,137 AXS	60,907,500 AXS	
38	The Sandbox SAND	$4.69	▲0.93%	▼20.48%	$4,316,767,186	$703,755,733 149,945,115 SAND	919,748,319 SAND	
39	Theta Network THETA	$4.02	▲1.49%	▼16.79%	$4,027,701,361	$157,473,321 39,097,566 THETA	1,000,000,000 THETA	

With the potential internal resources available to this project, combined with the explosion of the appearance of the "Metaverse" world in the future, real estate transactions in Decentraland can be an investing trend like the real estate industry at this time.

2.2. Reason for choosing ARIMA model.

Autoregressive integrated moving average (ARIMA) is a popular model in technical analysis that is used for analyzing and predicting the future asset's values based on a series of past values Nguyễn Anh Phong (2020) which is suitable for my research reason. Besides, ARIMA makes use of lagged moving averages to smooth time series data. Therefore, they can prove inaccurate under certain market conditions, such as financial crises or periods of rapid technological change (Adan Hayes, 2021). However, there are also have some disadvantages such as the model still has to be done manually to determine the "p, d, q" parameters through the ACF/PACF chart (Nguyễn Anh Phong, 2020). Hence, these are the reasons mentioned above that answer the question of why this model is suitable and why the author chooses it for the purpose of analyzing the future value of MANA token through the OHLCV historical data that the author uses in this report.

3. Data.

3.1. Data Interpretation.

The Decentraland project was officially released in early 2015 and has been noticed by the cryptocurrency community. In June 2015, this coin started programming algorithms to accept Bitcoin transactions; and in March 2017, the land at the virtual reality platform Decentraland had exploded strongly in the community through 3D space sketching, under the connection and support of Blockchain technology. Thanks to that leverage, after a month of development, this Token has launched the feature using ERC20 technology that allows users to receive parcels of LAND land and interact with other lands (Esteban Ordano, 2021). Depending on this development, it becomes extremely easy to market the land according to the user's wishes. Based on this important event, along with the limitation of data, that I used MANA token’s OHLCV historical data from January 01, 2015 to December 10, 2021 from https://finance.yahoo.com via API protocol.

3.2. Data collection process.

- Step 1: Importing the yahoo.finance library into python.

```
import yfinance as yf
```

- Step 2: Reading MANA token data by using the command:

```
# Import data from yahoo.finance
df_mana = yf.download('MANA-USD',
                      start='2015-01-01',
                      end='2021-12-10',
                      progress=False)
```

With “MANA-USD” being the name of the cryptocuurency that author wants to download data, starting from January 1, 2015 to December 10, 2021.

3.3. Method of processing data.

Step 1: To make the study easy to do, the author has chosen the appropriate attribute to run the model. In this dataset, the author chooses “Adj Close” attribute.

```
# Select "Adj Close" attribute for calculation.
df_mana = pd.DataFrame(df_mana['Adj Close'])
```

Step 2: Checking if the DataFrame has dimensions containing missing data. After checking, the author realizes that there are no missing data in this dataset.

```python
# Check number of missing data in the datasets
print(f'The number of missing data:\n{df_mana.isnull().sum()}')
```

```
The number of missing data:
Adj Close    0
dtype: int64
```

Step 3: Converting the date data format to the business weekly data format to facilitate the calculation. The method used for this step is to use the *df_mana.resample('W-Fri').ffill()* statement.

```python
# Resample daily data into weekly data
df_mana = df_mana.resample('W-Fri').ffill()
```

Date	**Adj Close**
2017-11-10	0.012940
2017-11-17	0.012028
2017-11-24	0.015914
2017-12-01	0.016067
2017-12-08	0.050740
...	...
2021-11-12	3.297853
2021-11-19	4.290613

2021-11-26	4.639659
2021-12-03	3.908493
2021-12-10	3.464618

4. ARIMA model.

4.1. Model's analytical significance.

To be able to run the model, we need to import the necessary libraries into python such as the subpackages of statsmodels, pandas, numpy, matplotlib, etc.

```
# Import basic library to processing data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
import seaborn as sns
import yfinance as yf

plt.style.use('seaborn')
plt.rcParams['figure.dpi'] = 500
warnings.simplefilter(action='ignore', category=FutureWarning)
```

```
# Import statics model using for running model
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.stattools import kpss
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.stats.diagnostic import acorr_ljungbox
import scipy.stats as scs
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error
```

First of all, author decomposes the MANA token price data series to understand the nature of the data more clearly and conveniently. Based on the results of the *Figure 1*, author can comment that the data had an overall strongly uptrend. The seasonal chart in *Figure 2* also shows that this data is seasonal, and the pattern of annual repetition is quite clear. From this, we can confirm that the adjusted close price data of the MANA token is not stationary.

```
# Update size and font for image
%matplotlib inline
plt.rcParams['figure.figsize'] = [20, 12]
plt.rcParams.update({'font.size': 20})

# Plot the data decay
decompose_results = seasonal_decompose(df_mana['Adj Close'],model='multiplicative')
decompose_results.plot()

plt.show()
```

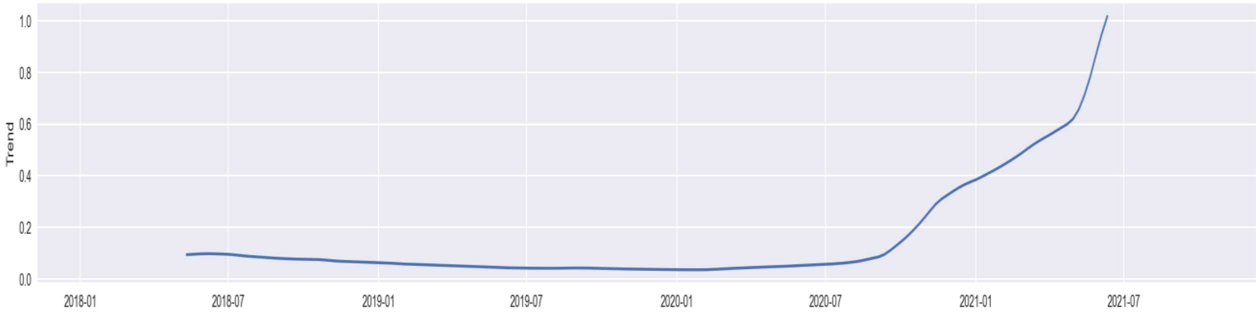


Figure 1. Decentraland price trend.

Figure 2. Decentraland price seasonal.

To check the stationarity of the above data set, we can use statistical tests such as the Augmented Dickey-Fuller Test (ADF), the Kwiatkowski–Phillips–Schmidt–Shin test (KPSS), ect.

- ADF test have the hypothesis that $\begin{cases} H_0\text{: time series is a not stationary} \\ H_1\text{: time series is a stationary} \end{cases}$

- KPSS test have the hypothesis that $\begin{cases} H_0\text{: time series is a stationary} \\ H_1\text{: time series is a not stationary} \end{cases}$

```python
# Function ADF test
def adf_test(x):

    stats_index = ['Test Statistic', 'p-value', '# of Lags Used', '# of Observations Used']

    test_adf = adfuller(x, autolag='AIC')
    results = pd.Series(test_adf[0:4], index = stats_index)

    for key, value in test_adf[4].items():
        results[f'Critical Value ({key})'] = value

    return results


# Function KPSS test
def kpss_test(x):

    stats_index = ['Test Statistic', 'p-value', '# of Lags']

    test_kpss = kpss(x)
    results = pd.Series(test_kpss[0:3], index = stats_index)

    for key, value in test_kpss[3].items():
        results[f'Critical Value ({key})'] = value

    return results
```

The author performed two tests on the adjust close price of the MANA token. The results show that this dataset is not stationary because the p-value of ADF is 0.996801, which geater than α = 0.05 (accept H_0: time series is a not stationary) and the p-value of the KPSS test is 0.01, which lower than α = 0.05 (reject H_0: time series is a stationary)

```python
# Test ADF with the adjust close price.
print('ADF test result of the adjust close price:')
adf_test(df_mana['Adj Close'])

# Test KPSS with the adjust close price.
print('KPSS test result of the adjust close price:')
kpss_test(df_mana['Adj Close'])
```

ADF test result of the adjust close price:

Test Statistic	*1.345641*
p-value	*0.996853*
# of Lags Used	*14.000000*
# of Observations Used	*199.000000*
Critical Value (1%)	*-3.463645*
Critical Value (5%)	*-2 .876176*
Critical Value (10%)	*-2.574572*

KPSS test result of the adjust close price:

Test Statistic	*0.759257*
p-value	*0.010000*
# of Lags	*15.000000*
Critical Value (10%)	*0.347000*
Critical Value (5%)	*0.463000*
Critical Value (2.5%)	*0.574000*
Critical Value (1%)	*0.739000*

To try to correct for errors in the data, the author calculates the first difference shown it in *Figure 3* and also use this data to retests ADF and KPSS.

```python
# Create a first difference column.
df_mana['diff'] = df_mana['Adj Close'].diff()
df_mana.dropna(inplace=True)
```

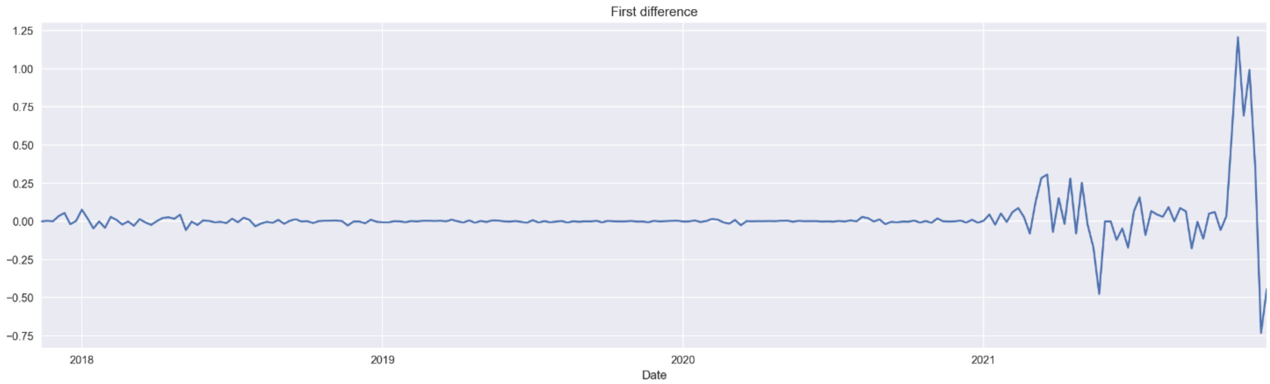


Figure 3. Decentraland price and the first difference chart.

After performing the first difference, we can clearly see the volatility of the first difference line around 0. This can partly help us understand that this data set stops at the first difference. To be more certain of that, we continue to test ADF and KPSS on the first difference dataset to give the most accurate and objective comments.

```python
# Test ADF with the first-order difference.
print('ADF test result of the first-order difference:')
adf_test(df_mana['diff'])

# Test KPSS with the first-order difference.
print('KPSS test result of the first-order difference:')
kpss_test(df_mana['diff'])
```

ADF test result of the first difference:		***KPSS test result of the first difference:***	
Test Statistic	*-3.130495*	*Test Statistic*	*0.346925*
p-value	*0.024377*	*p-value*	*0.100000*
# of Lags Used	*13.000000*	*# of Lags*	*15.000000*
# of Observations Used	*199.000000*	*Critical Value (10%)*	*0.347000*
Critical Value (1%)	*-3.463645*	*Critical Value (5%)*	*0.463000*
Critical Value (5%)	*-2.876176*	*Critical Value (2.5%)*	*0.574000*
Critical Value (10%)	*-2.574572*	*Critical Value (1%)*	*0.739000*

In the first difference data and α = 0.05, with the ADF test, we get that p-value is 0.024377 which lower than α (reject H_0: time series is a not stationary). For the KPSS test, we get the result that p-value is 0.1 which geater than α (accept H_0: time series is a stationary). After testing, we obtain the results of the first difference series of the data set to ensure stationarity. Thus, we can conclude that the parameter "d" is 1.

Thirdly, to run the ARIMA model, we must determine the two other important parameters are "p" and "q" through the graphs of PACF and ACF, respectively. We draw ACF/PACF histogram with a delay is 25 and α = 0.05 on the first difference characteristic of the dataset, and the results are shown in *Figure 4*.

```
# Specify parameters
n_lags = 25
significance_level = 0.05

# Update size and font for image
%matplotlib inline
plt.rcParams['figure.figsize'] = [22, 18]
plt.rcParams.update({'font.size': 20})

# Draw plot
fig, ax = plt.subplots(2, figsize=(16, 8))

# Plot ACF
plot_acf(df_mana['diff'], ax = ax[0], lags = n_lags,
         alpha = significance_level, c= 'green')

# Plot PACF
plot_pacf(df_mana['diff'], ax = ax[1], lags = n_lags,
          alpha = significance_level,c= 'red')

plt.show()
```

According to *Figure 4*, we can see that in the Autocorrelation charts, the flags gradually decrease from 0 to 2 and drop suddenly at the 3rd flag point. At the Partial Autocorrelation chart, the flags gradually decrease from 0 to 1 and suddenly decrease at the 2nd flag point. Hence, we can choose the "p" parameter from 0 to 2, the "q" parameter from 0 to 3 and the "d" parameter is 1 as defined above. Thus, there will be many ARIMA models that can be used in this case, such as ARIMA (1,1,1), ARIMA (2,1,1), ARIMA (2,1, 2), ARIMA (2,1,3), ARIMA (1,1,2), ARIMA (1,1,3), ect. However, to determine the most appropriate model to forecast the price of MANA, the author chooses a test method to select the model with the lowest AIC.

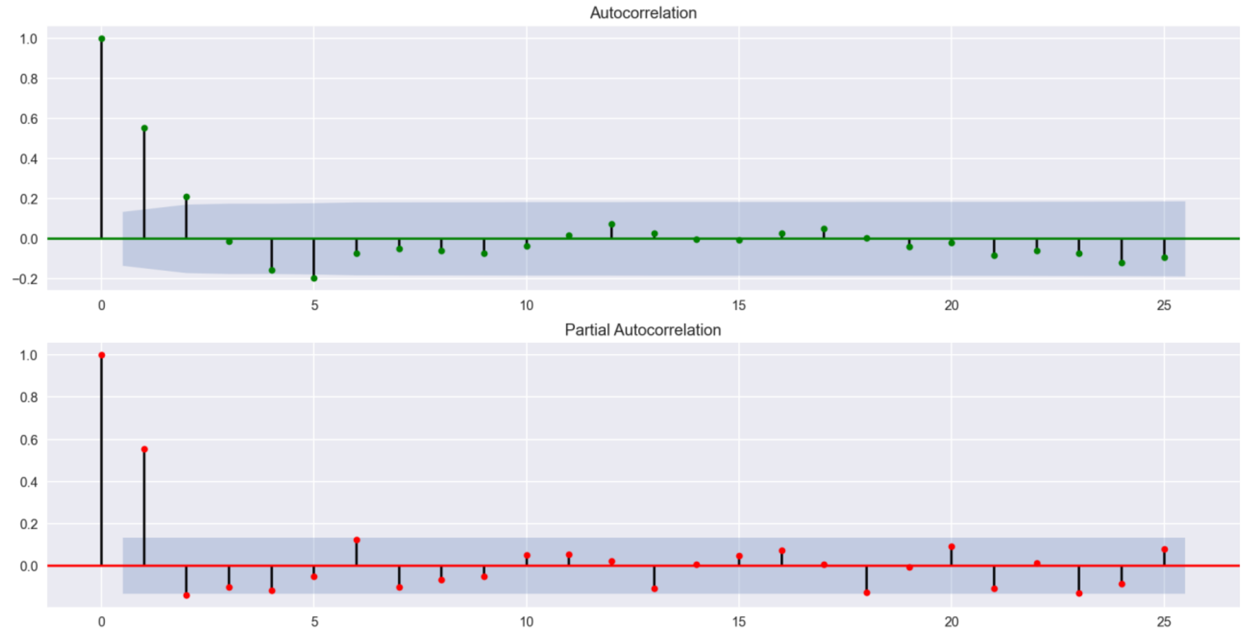


Figure 4. Autocorrelation test graph of MANA token.

After performing tests in the various parameters "p" and "q" have been scoped above. We can conclude that the parameters "p" is 2 and "q" is 3 with the model used **ARIMA (2, 1, 3)** because this model has the lowest AIC index **(AIC = -292.911)**, which is better than the other ARIMA model (Nguyễn Anh Phong, 2020).

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Adj Close   No. Observations:                  213
Model:                 ARIMA(2, 1, 3)   Log Likelihood                 152.455
Date:                Sun, 23 Jan 2022   AIC                           -292.911
Time:                        12:18:43   BIC                           -272.771
Sample:                    11-17-2017   HQIC                          -284.771
                         - 12-10-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.8486      0.068     12.468      0.000       0.715       0.982
ar.L2         -0.7031      0.048    -14.601      0.000      -0.798      -0.609
ma.L1         -0.2736      0.096     -2.860      0.004      -0.461      -0.086
ma.L2          0.6861      0.043     16.017      0.000       0.602       0.770
ma.L3          0.2831      0.066      4.309      0.000       0.154       0.412
sigma2         0.0138      0.001     22.955      0.000       0.013       0.015
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):              3830.11
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):              55.60   Skew:                             1.70
Prob(H) (two-sided):                  0.00   Kurtosis:                        23.54
```

The model summary provides a lot of information. The table in the middle is the coefficients table where the values under “coef” are the weights of the respective terms. The model gives quite good results when the regression coefficients are all highly statistically significant, that is, all of the column “P>|z|” is less than 0.05. From that, we can understand that the estimated coefficients are all significant, which means that the independent variables have an impact on the dependent variable explained by an equation $y = f(x)$. Thus, the model above is accepted.

In order to use the future value prediction results, we need to check whether the residuals of the model are appropriate.

```
def arima_diagnostics(resids, n_lags=25):

    # Create subplots
    fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)

    # Calculate resids
    r = resids
    resids = (r - np.nanmean(r)) / np.nanstd(r)
    resids_nonmissing = resids[~(np.isnan(resids))]

    # residuals over time
    sns.lineplot(x=np.arange(len(resids)), y=resids, ax=ax1)
    ax1.set_title('Standardized residuals')

    # Distribution of residuals
    x_lim = (-1.96 * 2, 1.96 * 2)
    r_range = np.linspace(x_lim[0], x_lim[1])
    norm_pdf = scs.norm.pdf(r_range)

    # Plot distribution
    sns.distplot(resids_nonmissing, hist=True, kde=True,
                 norm_hist=True, ax=ax2)
    ax2.plot(r_range, norm_pdf, 'g', lw=2, label='N(0,1)')
    ax2.set_title('Distribution of standardized residuals')
    ax2.set_xlim(x_lim)
    ax2.legend()

    # Q-Q plot
    qq = sm.qqplot(resids_nonmissing, line='s', ax=ax3)
    ax3.set_title('Q-Q plot')

    # ACF plot
    plot_acf(resids, ax=ax4, lags=n_lags, alpha=0.05)
    ax4.set_title('ACF plot')

    return fig
```

From the graph of *Figure 5*,we can confirm some things such as the residuals are normally distributed through the histogram of the distribution of the residuals. The residual errors seem to fluctuate around a mean of zero and have a uniform variance shown in the Standardized residual subplot. The Q-Q plot shows that mostly the dots fall perfectly in line with the red line. Any significant deviations would imply the distribution is skewed. Through the residual autocorrelation (ACF), we conclude that the residuals are stable because the residual errors are not autocorrelated. Overall, the model seems to be a good fit, thus we can use it for the above prediction results.

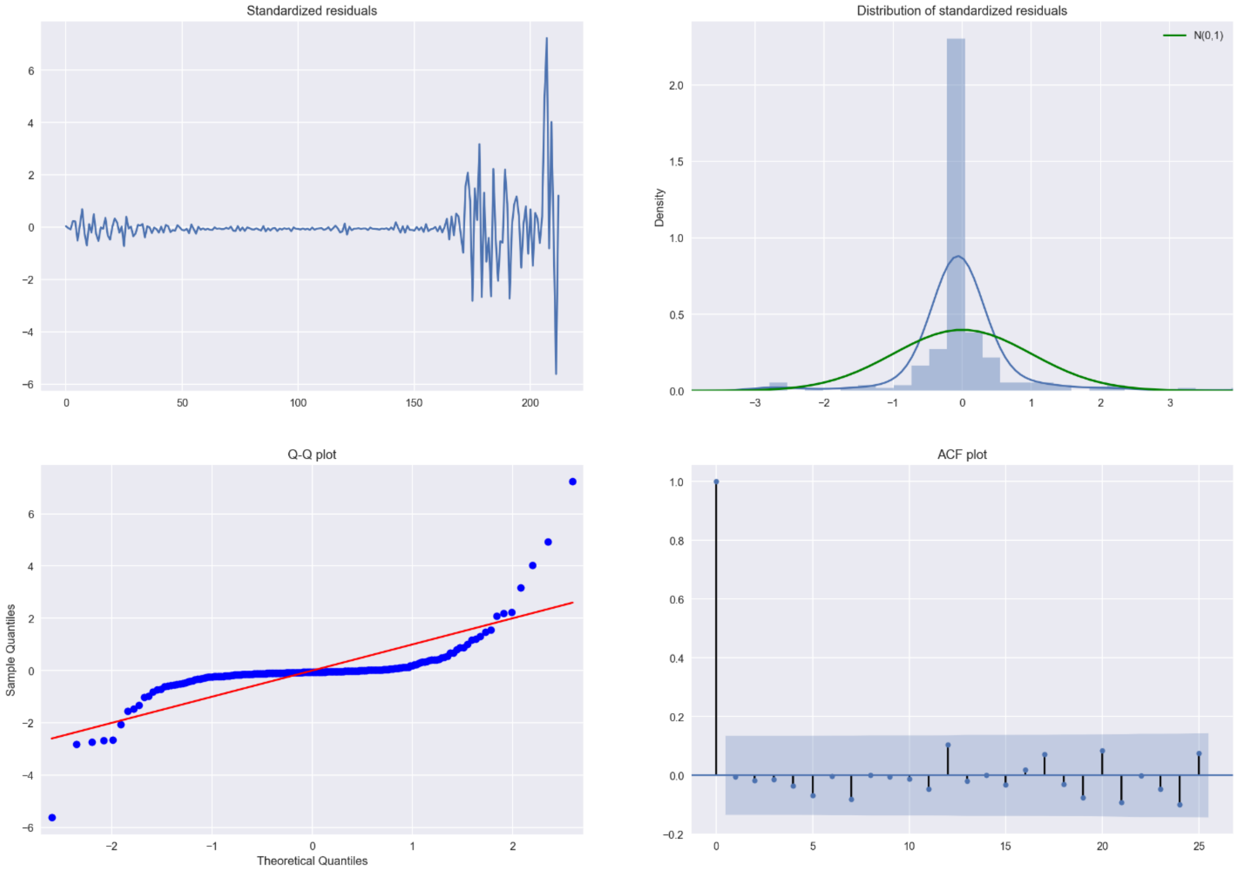


Figure 5. Arima diagnostics based on resids chart.

The author tests the accuracy of MANA's price forecasting tool against its closing price data set to see if the model is accurate and the forecast is true to reality.

```
# Forecast length
n_forecasts = len(data_forecast)

# ARIMA forecast restults
arima_pred = arima.predict(n_forecasts)

# Concatenate forecast data
arima_pred = pd.DataFrame(arima_pred)
arima_pred.columns = ['Predict']

# Real price
actual = df_mana["Adj Close"][6:]
```

Then, the author creates a DataFrame between the actual price and the predicted price for easy observation.

```
volatility_pre = pd.DataFrame({'Real price': list(actual),
                               'Prediction price': list(arima_pred.Predict)},
                               index= arima_pred.index)
```

Date	Real Price	Prediction Price
2017-12-29	0.089106	0.072741
2018-01-05	0.165527	0.077205
2018-01-12	0.184358	0.203259
2018-01-19	0.137736	0.211668
2018-01-26	0.137317	0.115625
...	...	...
2021-11-12	3.297853	3.384903
2021-11-19	4.290613	3.812323
2021-11-26	4.639659	4.697180
2021-12-03	3.908493	4.557039
2021-12-10	3.464618	3.316007
2017-12-29	0.089106	0.072741
2018-01-05	0.165527	0.077205
2018-01-12	0.184358	0.203259
2018-01-19	0.137736	0.211668
2018-01-26	0.137317	0.115625
2021-12-03	3.908493	4.557039
2021-12-10	3.464618	3.316007

Next, the author visualizes the actual price and the predicted priceand the results are presented in the *Figure 6.*

```
# Plot forecast results
%matplotlib inline
plt.rcParams['figure.figsize'] = [20, 14]
plt.rcParams.update({'font.size': 20})

fig, ax = plt.subplots(1)
ax = sns.lineplot(data = actual, label = 'Actual')
ax.plot(arima_pred.Predict, label = 'ARIMA(2,1,3)')
ax.set(title="MANA token - actual vs predicted", xlabel = 'Date', ylabel = 'Price')

plt.legend()

plt.show()
```

We can see that the forecast line that the ARIMA model (2,1,3) gives is relatively accurate compared to the actual price line. In addition, indicators like RMSE = 0.12, MAE = 0.05 and MAPE = 17.01% have also reflected that.

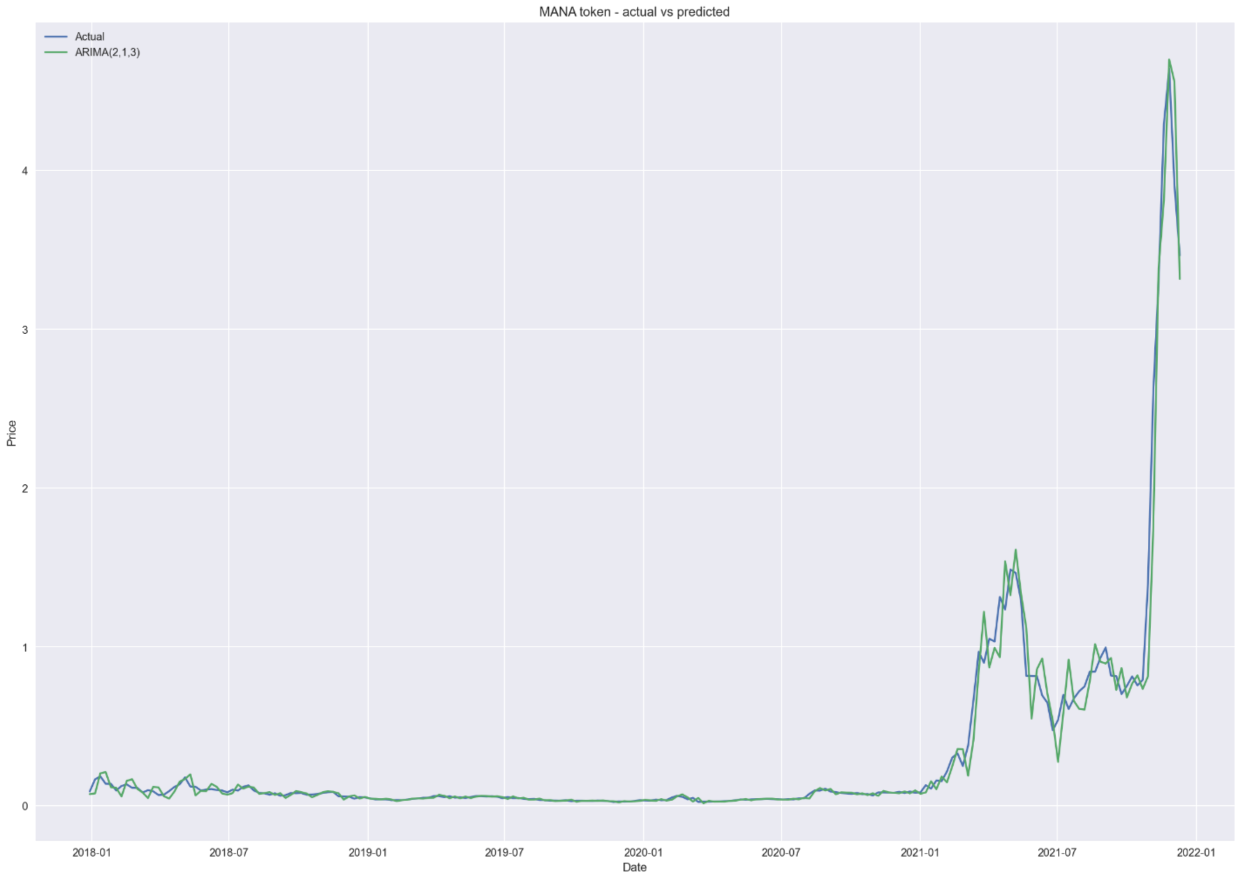


Figure 6. Predicted and actual price.

After determining the appropriate model after the testing steps, the author makes a prediction of the future price of the MANA token and use the predicted price to compare the volatility to the actual price. The author download the MANA's data from December 17, 2021 to the end of January 20, 2022 from yahoo.finance and tranfer it weekly data to the forecast model.

```
# import data used for forcasting

data_forecast = yf.download('MANA-USD',
                            start='2021-12-17',
                            end='2022-01-20',
                            progress=False)

data_forecast = data_forecast['Adj Close'].resample('W-Fri').ffill()
```

```
# Forecast length
n_forecasts = len(data_forecast)

# ARIMA forecast restults
arima_forcase = arima.forecast(n_forecasts)

# Concatenate forecast data
arima_forcase = pd.DataFrame(arima_forcase)
arima_forcase.columns = ['arima_forcase']
```

After that, the author creates a DataFrame having the predicted price and the actual prices.

```
volatility = pd.DataFrame({'Real price': list(data_forecast),
                           'Prediction price': list(arima_forcase.arima_forcase)},
                           index= data_forecast.index)
```

Date	Real Price	Prediction Price
2021-12-17	3.072275	3.100168
2021-12-24	3.505177	3.021332
2021-12-31	3.270671	3.252766
2022-01-07	3.051883	3.504594
2022-01-14	2.955072	3.555564
2022-01-21	2.735672	3.421748

```
# Plot forecast results
%matplotlib inline
plt.rcParams['figure.figsize'] = [20, 14]
plt.rcParams.update({'font.size': 20})

fig, ax = plt.subplots(1)
ax = sns.lineplot(data = data_forecast, label = 'Actual')
ax.plot(arima_forcase.arima_forcase, label = 'ARIMA(2,1,3)')
ax.set(title="MANA token - actual vs predicted", xlabel = 'Date', ylabel = 'Price')

plt.legend()

plt.show()
```

Figure 7. Forcasting MANA token.

After performing the model, the author obtained the results that the predicted price line has predicted the downtrend of the actual price line in the short-term and the volatility can be accepted, which be shown in *Figure 7.* This strong volatility is due to the fact that BTC has had a sharp drop in price, making the participants' psychology of the cryptocurrency market are nervous, leading to Altcoins being adversely affected as well. The difference between the actual value and the forecast value through the RMSE index is 0.46, the MAE index is 0.38 and the MAPE index is 12.58%, three of them are quite good.

4.2. Conclusion.

To be able to implement the ARIMA model to predict the future value of MANA tokens in particular and many economic and financial chains in general, which are five steps. Almost all economic and financial series in level form are nonstationary, therefore, we must use the difference to consider their stationarity. We can test for stationarity through one of two tests, ADF and KPSS, or both to find out the parameter "d". Then, we continue to determine the remaining two parameters "p" and "q" through the ACF/PACF chart. After defining the model, it is necessary to consider the relevance of the regression results. Then it is necessary to test the regression results before predicting the model. Finally, we visualize the future value predicted from the model which checks for deviations from the actual value.

After completing the ARIMA model for the MANA token, we can see that the token is stationary at the first difference, the econometric indicators all give statistically significant results. In addition, according to the forecast of the model ARIMA (3, 1, 2) shows that the price of this token will tend to decrease in the short term although the Decentraland project is currently the trend of the future with the introduction of Metaverse. The model not only has partly accurately predicted the movement of this token but also show that the volatility compared to actual data maybe accepted even through the cryptocurrency products often have a high degree of risk and high price fluctuations which accord to Phan Le Thanh Long, Director of AFA Research & Education (Thắng, 2021). However, due to the limitation of the predicted dataset, the results have not yet been given the highest accuracy. At the same time, we can apply machine learning to better forecast high-risk assets like MANA tokens in particular and cryptocurrencies in general.

5. References.

Adan Hayes. (2021). Autoregressive Integrated Moving Average (ARIMA). Retrieved from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

Esteban Ordano, A. M., Yemel Jardi, Manuel Araoz. (2021). Decentraland White paper. Retrieved from https://decentraland.org/whitepaper.pdf

Matthew De Saro. (2021). Most Expensive Virtual Plot of Land Ever Sells for $900,000 on Decentraland. Retrieved from https://finance.yahoo.com/news/most-expensive-virtual-plot-land-185000712.html
Nguyễn Anh Phong, N. T. C., Phan Huy Tâm, Ngô Phú Thanh. (2020). *Sách tham khảo "Ứng dụng Python trong tài chính"*: NXB Đại học Quốc Gia Tp.HCM.
Nguyen Toan. (2021). What is Decentraland (MANA)? Everything you need to know about MANA Token. Retrieved from https://coin98insights.com/what-is-decentraland-mana
Thắng, Q. (2021). Nhà đầu tư tiền mã hóa đối mặt những rủi ro nào? Retrieved from https://zingnews.vn/nha-dau-tu-tien-ma-hoa-doi-mat-nhung-rui-ro-nao-post1218711.html
Thu Hường. (2021). Thị trường chứng khoán Việt Nam năm 2020 và triển vọng năm 2021. Retrieved from http://consosukien.vn/thi-truong-chung-khoan-viet-nam-nam-2020-va-trien-vong-nam-2021.htm
VTV Digital. (2021). Đầu tư vào tiền ảo năm 2021 nhiều hơn 8 năm trước cộng lại. Retrieved from https://vietnamnet.vn/vn/kinh-doanh/tai-chinh/dau-tu-vao-tien-ao-nam-2021-tang-manh-802968.html

- Link Github: https://github.com/BrianNguyen2001/ARIMA-Model_MANA-Token