Advanced Machine Learning – Practicum 2-Time Series Analysis

[Tee Li Lin]

(Qns 1)

As there are no missing values in 'timestamp' column, there is no data pre-processing to be done for this column except change the data format into hours before resampling into days. Due to the high fluctuation in cryptocurrency's Closing price, thus will leave the dataset as it is. There are missing values in 'Target' column, thus there is imputation of values in this column using pandas interpolate function.

(Qns 2)

From Jupyter Notebook:

In [63]:
```
#extract bitcoin data
Bitcoin_df = raw_data_final[raw_data_final['Asset_ID'] == 1]
Bitcoin_data=Bitcoin_df.drop('Asset_ID',axis=1)
Bitcoin_data
```

Out[63]:

timestamp	Count	Open	High	Low	Close	Volume	VWAP	Target
2018-01-01 00:01:00	229.0	13835.194000	14013.800000	13666.11	13850.176000	31.550062	13827.062093	-0.014643
2018-01-01 00:02:00	235.0	13835.036000	14052.300000	13680.00	13828.102000	31.046432	13840.362591	-0.015037
2018-01-01 00:03:00	528.0	13823.900000	14000.400000	13601.00	13801.314000	55.061820	13806.068014	-0.010309
2018-01-01 00:04:00	435.0	13802.512000	13999.000000	13576.28	13768.040000	38.780529	13783.598101	-0.008999
2018-01-01 00:05:00	742.0	13766.000000	13955.900000	13554.44	13724.914000	108.501637	13735.586842	-0.008079
...	...	...	...	...	...	...	...	...
2021-09-20 23:56:00	1940.0	42983.780000	43001.850849	42878.26	42899.012857	56.850913	42935.489499	NaN
2021-09-20 23:57:00	2026.0	42904.197143	42932.000000	42840.16	42860.005714	80.993326	42879.576084	NaN
2021-09-20 23:58:00	1986.0	42859.385714	42887.500000	42797.20	42827.020000	65.677734	42844.090693	NaN
2021-09-20 23:59:00	4047.0	42839.012802	43042.160000	42818.10	43017.277143	138.335477	42935.761938	NaN
2021-09-21 00:00:00	2698.0	43009.961250	43048.510000	42961.64	43002.505000	128.206820	43011.414052	NaN

1956282 rows × 8 columns

In [74]:
```
#resample the sub datasets into days
df_bitcoin=Bitcoin_close.resample("D").mean()
df_ethereum=Ethereum_close.resample("D").mean()
df_dogecoin=Dogecoin_close.resample("D").mean()
df_Bitcoin=Bitcoin_target.resample("D").mean()
df_Ethereum=Ethereum_target.resample("D").mean()
df_Dogecoin=Dogecoin_target.resample("D").mean()
```

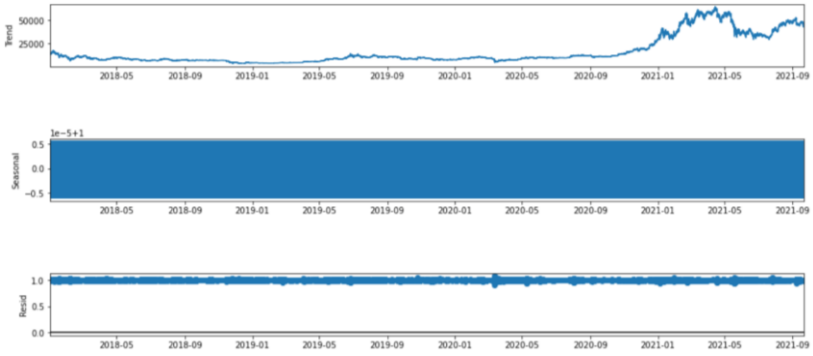
In [75]:
```
def split_data(data, percent):
  train_size = int(len(data) * percent)
  train, test = data[0:train_size], data[train_size:]
  return train, test
```

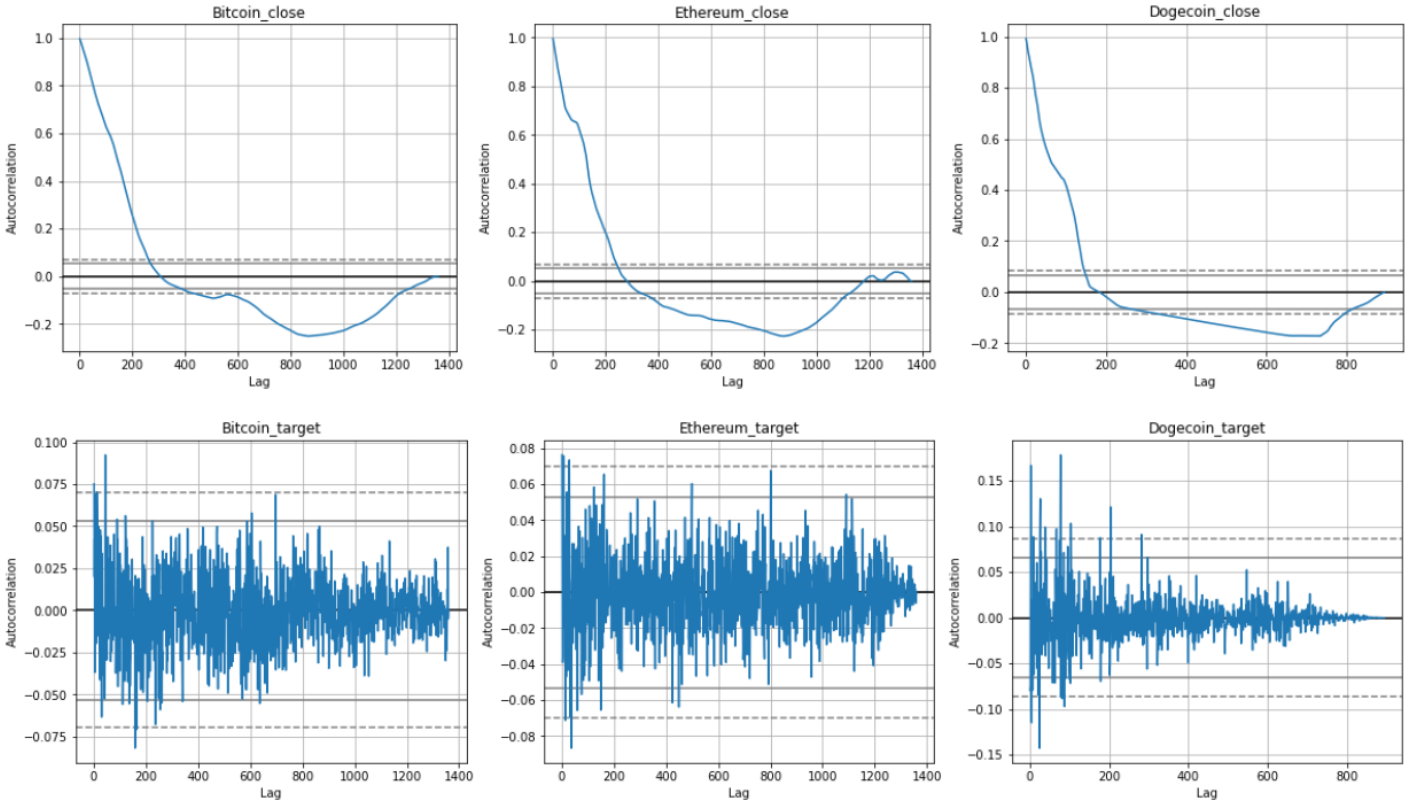
In [76]:
```
#Split the time series datasets for Close
percent=0.7
bitcoin_train_test=split_data(df_bitcoin, percent)
ethereum_train_test=split_data(df_ethereum, percent)
dogecoin_train_test=split_data(df_dogecoin, percent)
```

In [77]:
```
#Split the time series datasets for Target
percent=0.7
Bitcoin_train_test=split_data(df_Bitcoin, percent)
Ethereum_train_test=split_data(df_Ethereum, percent)
Dogecoin_train_test=split_data(df_Dogecoin, percent)
```

(Qns 3)

Based on the plots below, it seems that there is 'noise' in the target data for all 3 cryptocurrencies and for the close data, there is a big jump in values from 2021-1 onwards based on the seasonal_decompose plots. And the lowest autocorrelation for Bitcoin and Ethereum occurred at 800-1000 lags while for Dogecoin, it occurred at 600-800 lags.





(Qns 4,5)

Moving averages is a series of averages, calculated from historic data.

Moving Average equation:

$$Y_t = \alpha + \epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + .. + \phi_q\epsilon_{t-q}$$

Auto-regression is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step.

Arima equation:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p}\epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + .. + \phi_q\epsilon_{t-q}$$

Below are the ARIMA's optimised hyperparameters for 3 cryptocurrencies Close and target:

```
Results for Dogecoin Close:
best p,d,q =(0, 1, 0), RMSE value =0.024593205119807164, lowest MAPE value =0.05896
277047840079

Results for Ethereum Close:
best p,d,q =(1, 1, 0), RMSE value =88.59611718183469, lowest MAPE value =0.03133092
677381613

Results for Bitcoin Close:
best p,d,q =(0, 2, 1), RMSE value =1345.2126989185801, lowest MAPE value =0.0237288
30103463116

Results for Dogecoin Target:
best p,d,q =(0, 0, 0), RMSE value =0.0013294429042115168, lowest MAPE value =1.2934
447047212214

Results for Ethereum Target:
best p,d,q =(0, 0, 0), RMSE value =0.00030312246305027663, lowest MAPE value =1.032
3851752239994

Results for Bitcoin Target:
best p,d,q =(0, 0, 0), RMSE value =0.00028932887047023224, lowest MAPE value =1.002
0300260829107
```

(Qns 6)

Dogecoin Close: Gradient Boosting Regression model

```
The best parameters across ALL searched params:
 {'learning_rate': 0.01, 'max_depth': 7, 'max_features': 7, 'min_samples_leaf': 3,
'min_samples_split': 40, 'n_estimators': 1000, 'subsample': 0.8}

 RMSE using Gradient Boosting Regression model: 3.491722739058418e-05

 MAPE using Gradient Boosting Regression model: 0.008920027242506245
```

Ethereum Close: Decision Tree Regression model

```
The best parameters across ALL searched params:
 {'learning_rate': 0.01, 'max_depth': 7, 'max_features': 7, 'min_samples_leaf': 3,
'min_samples_split': 40, 'n_estimators': 1000, 'subsample': 0.8}

 RMSE using Decision Tree Regression model: 83.82735447964059

 MAPE using Decision Tree Regression model: 0.0920470823266285
```

Bitcoin Close: Random Forest Regression model

```
The best parameters across ALL searched params:
 {'bootstrap': False, 'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf':
1, 'min_samples_split': 2, 'n_estimators': 100}

 RMSE using Random Forest Regression model: 0.5331402114189215

 MAPE using Random Forest Regression model: 1.9509352856908415e-05
```

Dogecoin Target: Random Forest Regression model

```
The best parameters across ALL searched params:
 {'bootstrap': True, 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf':
8, 'min_samples_split': 12, 'n_estimators': 100}

 RMSE using Random Forest Regression model: 0.0004375554579413572

 MAPE using Random Forest Regression model: 1.1108827057243031
```

Ethereum Target: Random Forest Regression model

```
The best parameters across ALL searched params:

 {'bootstrap': True, 'max_depth': 60, 'max_features': 'sqrt', 'min_samples_leaf':
8, 'min_samples_split': 12, 'n_estimators': 100}

 RMSE using Random Forest Regression model: 0.00015200489456382478

 MAPE using Random Forest Regression model: 1.496278011537673
```

Bitcoin Target: Random Forest Regression model

```
The best parameters across ALL searched params:

 {'bootstrap': True, 'max_depth': 40, 'max_features': 'sqrt', 'min_samples_leaf':
8, 'min_samples_split': 5, 'n_estimators': 400}

 RMSE using Random Forest Regression model: 0.00015352299928651867

 MAPE using Random Forest Regression model: 1.335318074126454
```

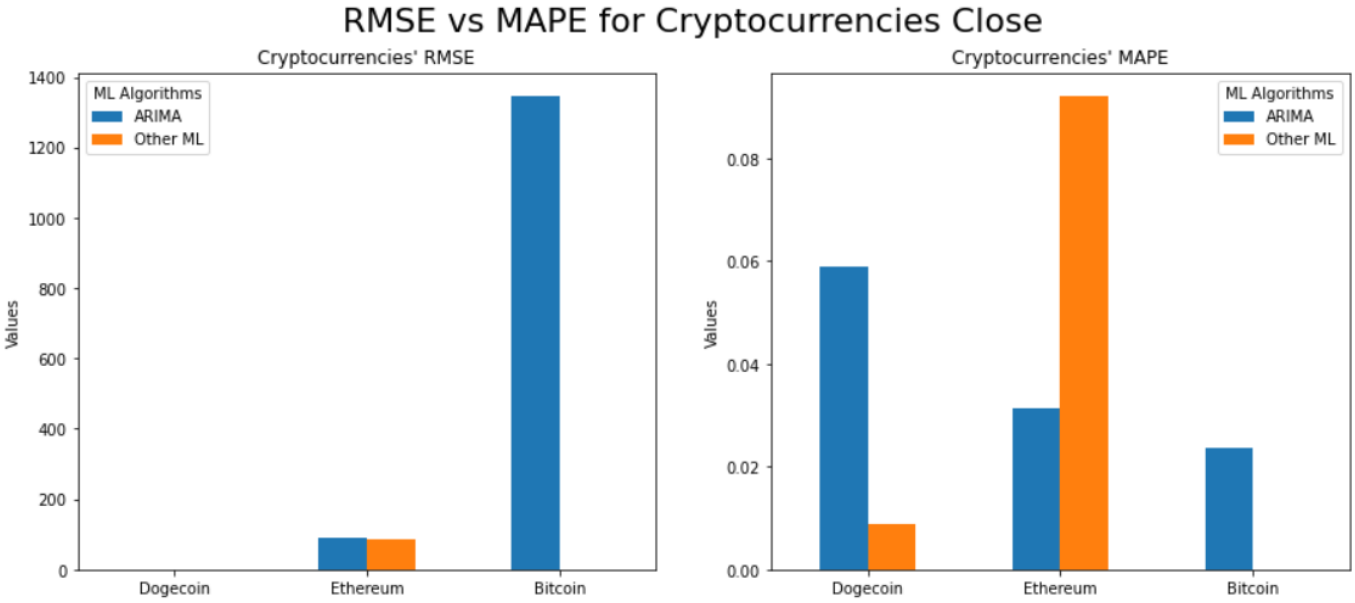
Overall result for 3 cryptocurrencies Close:

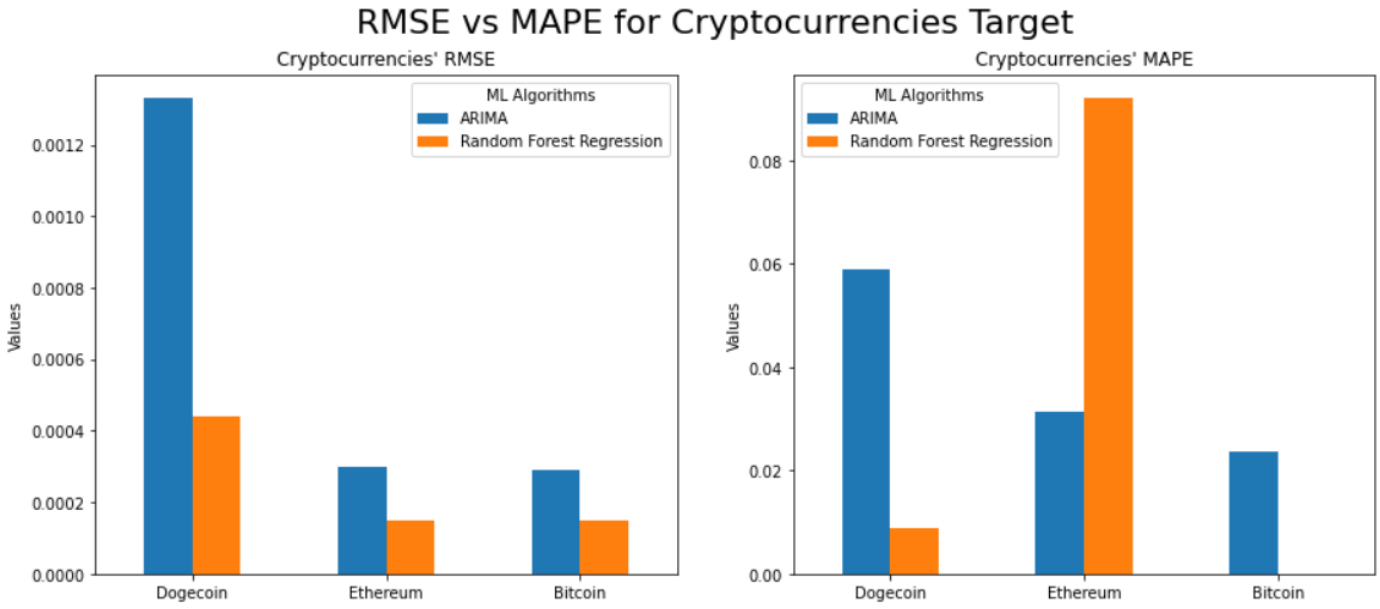
	ARIMA		Other ML	
	RMSE	MAPE	RMSE	MAPE
Close				
Dogecoin	0.02459	0.05896	0.00003	0.00892
Ethereum	88.59612	0.03133	83.82735	0.09205
Bitcoin	1345.21270	0.02373	0.53314	0.00002

Overall result for 3 cryptocurrencies Target:

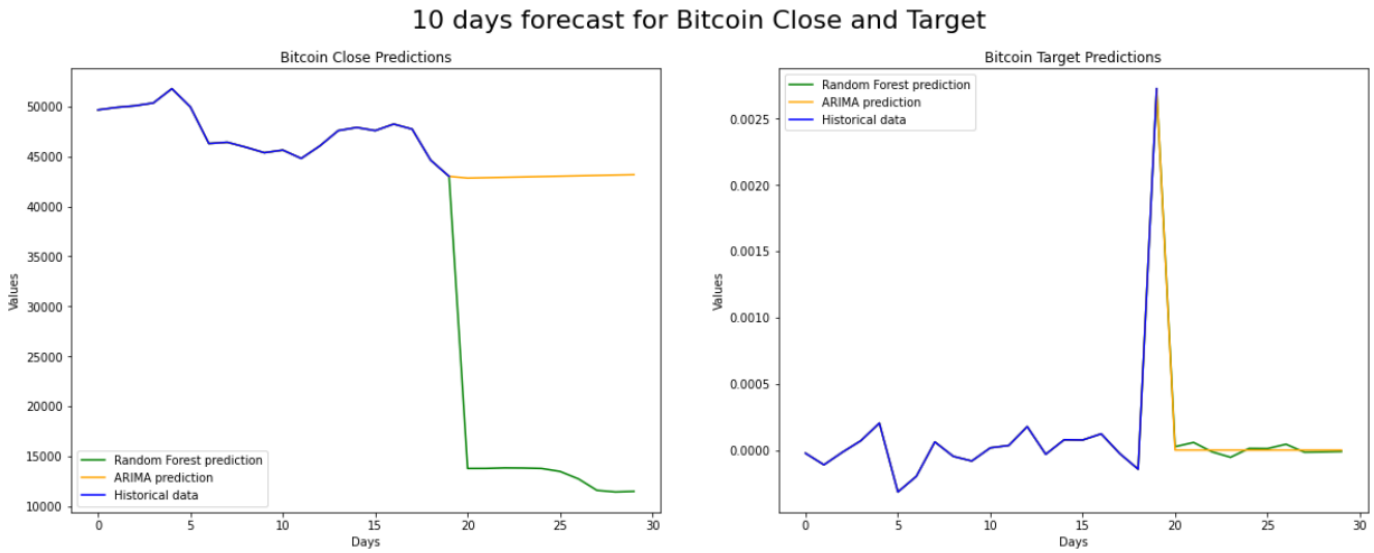
	ARIMA		Random Forest Regression	
	RMSE	MAPE	RMSE	MAPE
Target				
Dogecoin	0.00133	1.29344	0.00044	1.11088
Ethereum	0.00030	1.03239	0.00015	1.49628
Bitcoin	0.00029	1.00203	0.00015	1.33532

(Qns 7)





(Qns 8)



Based on the graph above, it seems that Random Forest model is not suitable for predicting this bitcoin dataset using time series. ARIMA is the much suitable model to predict time series data.