Tesla stock forecast

A case study by Mohammad Saleh

Tasks

1. Analyze the data.
2. Create a univariate time series forecast:
    a. Goal: Prediction of the “close values” for the period 15.06.21 - 29.06.21
    b. Compare at least two different approaches and evaluate the results.
3. Create a multivariate time series forecast:
    a. Create your own features to improve predictions
    b. Goal: Prediction of the “close values” for the period 15.06.21 - 29.06.21
    c. Compare at least two different approaches and evaluate the results.

Understanding the problem

Time series

Time series is a sequence of data that has some order usually with a time component in a set interval.

Data analysis

also known as exploratory data analysis (EDA) of a time series involves analyzing the data for its structure such as seasonality or nature of its temporal process.

Prediction/Forecasting

Prediction of future values of time series involves using past data to predict the future.

Project objective:

1. Univariate study for the Tesla stock market, using data and visualizations.
2. Multivariate study for the Tesla stock market, using external data and visualizations.

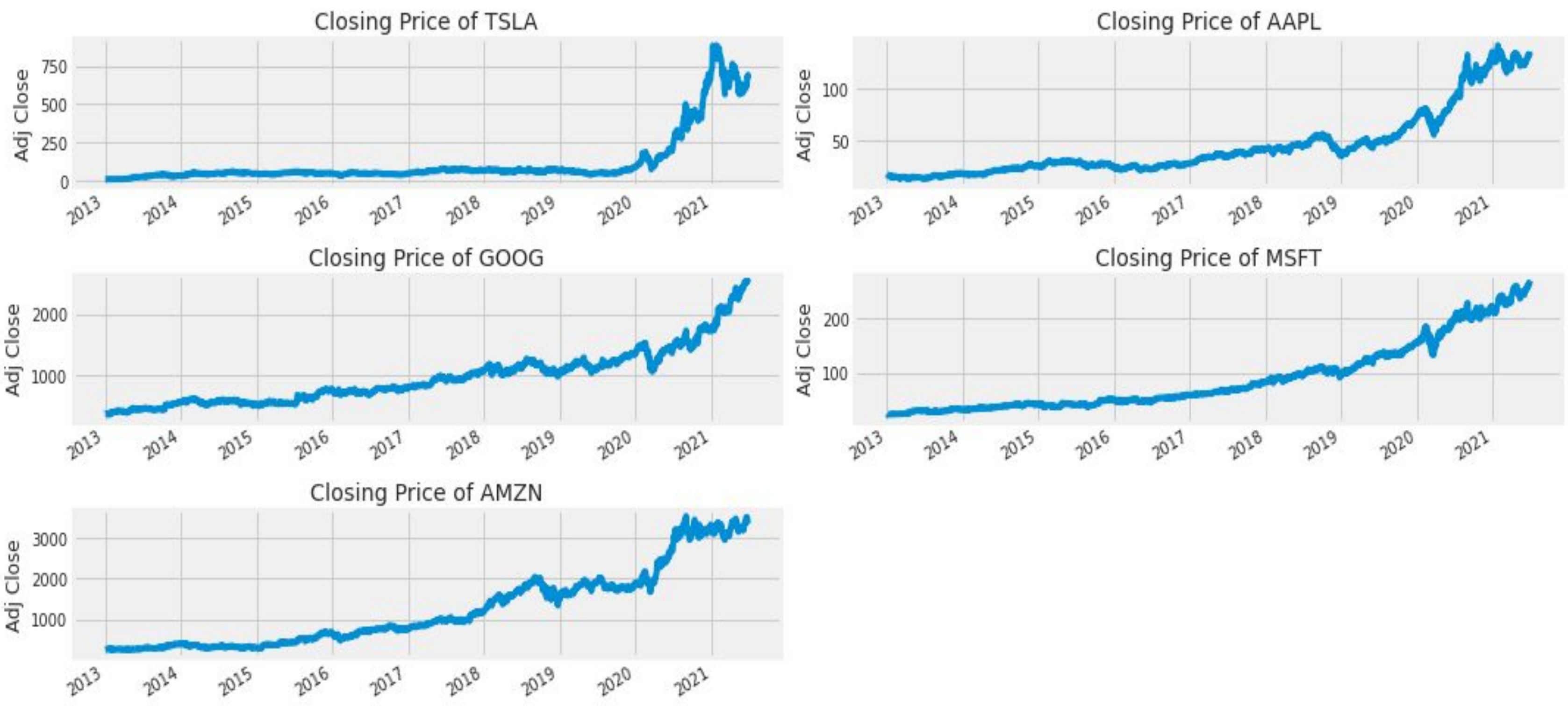
- *univariate: time series with a single observation per time increments.*
- *multivariate: time series that has more than one observation per time increments. These observations or time-dependent variables can capture the dynamic of multiple time series.*

Understanding the stock market

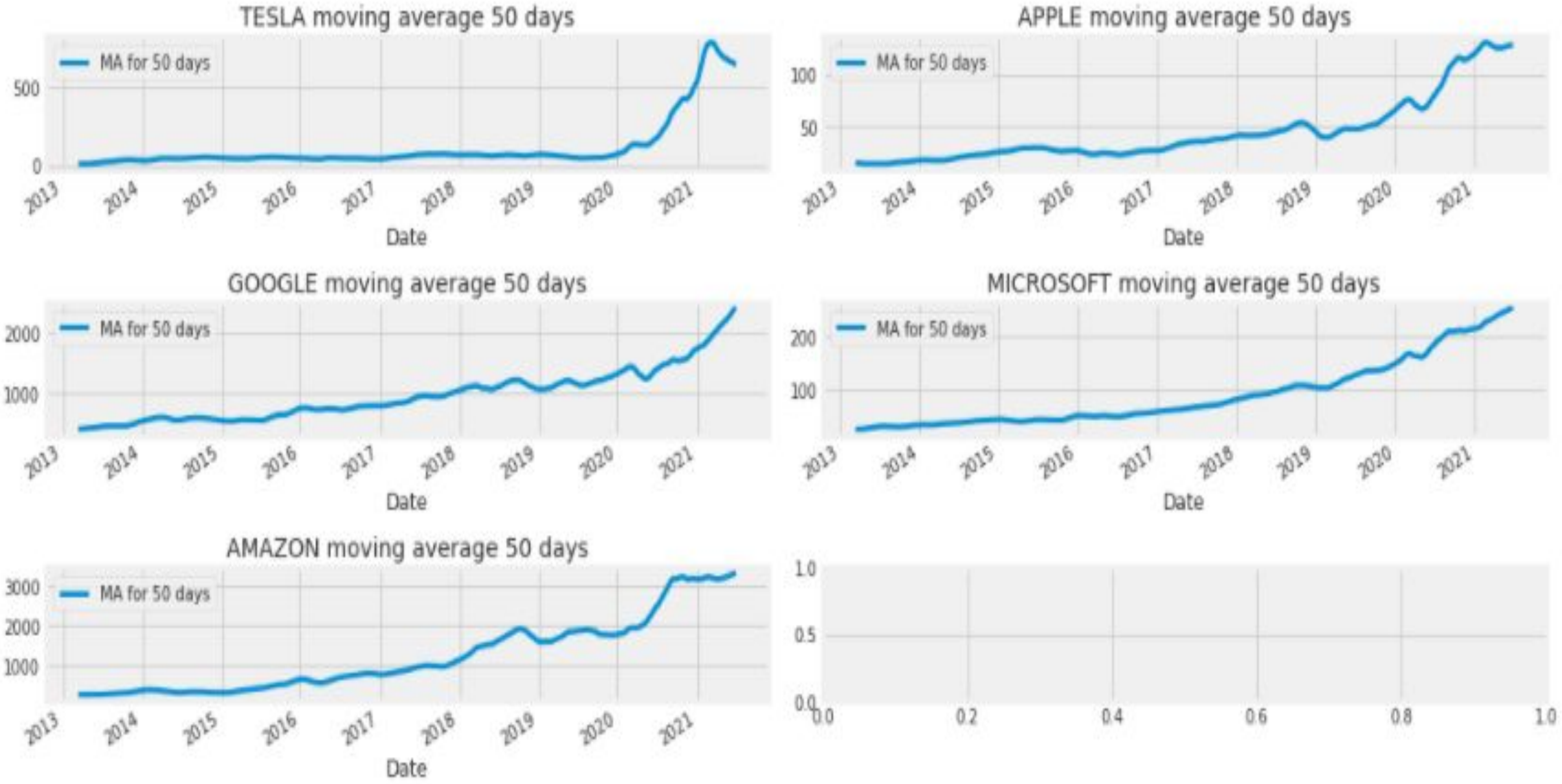
Part 1. Exploratory data analysis

1. What was the change in price of the stock over time?
2. What is the trend of the stock?
3. What was the moving average of the various stocks?
4. How is the seasonality trends of the Tesla stock?
5. What are the correlation between different stocks'?
6. How much value do we put at risk by investing in Tesla stock?
7. How can we attempt to predict future stock behavior?

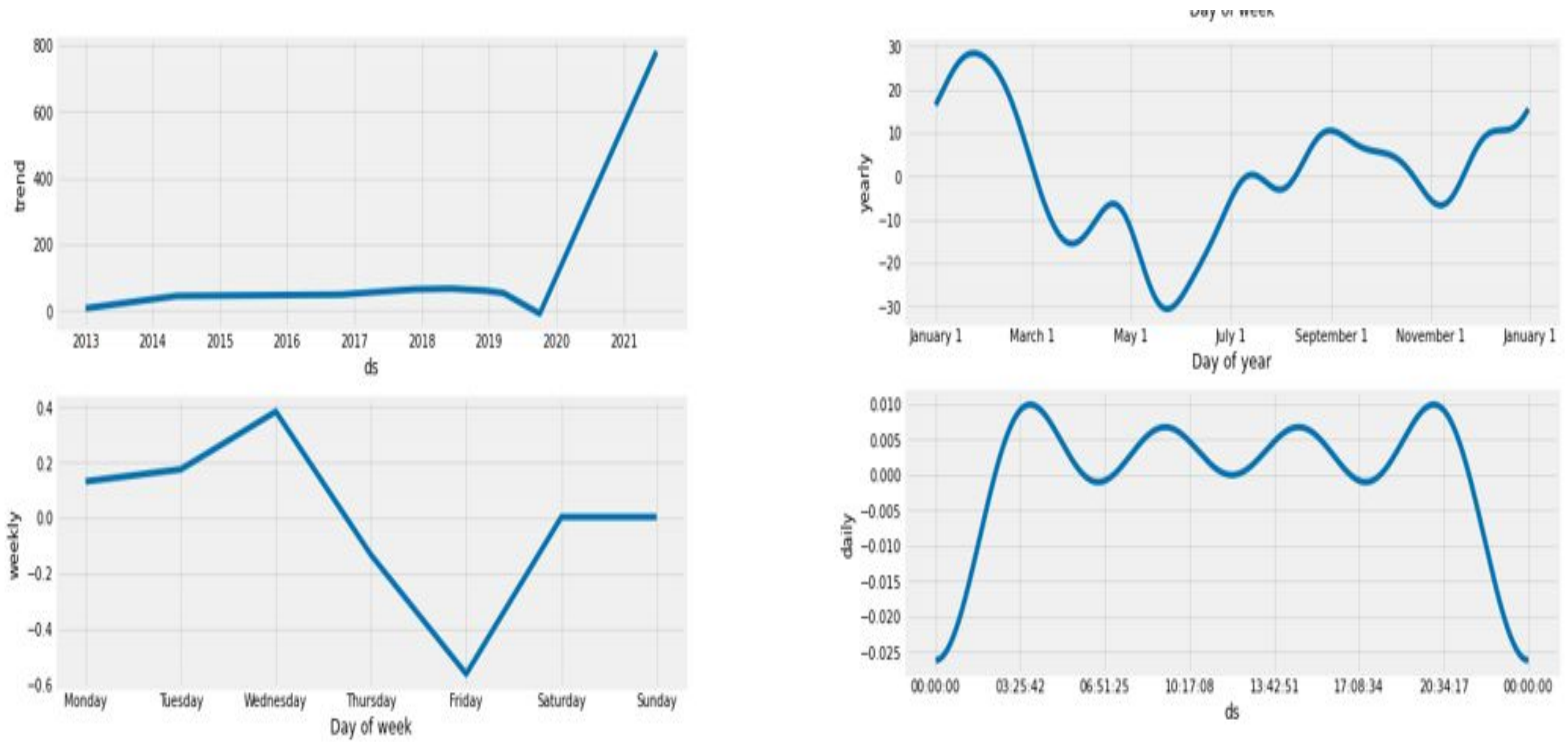
What was the change in price of tesla and other relevant stocks over time?



Moving average for 50 days of the previous trends

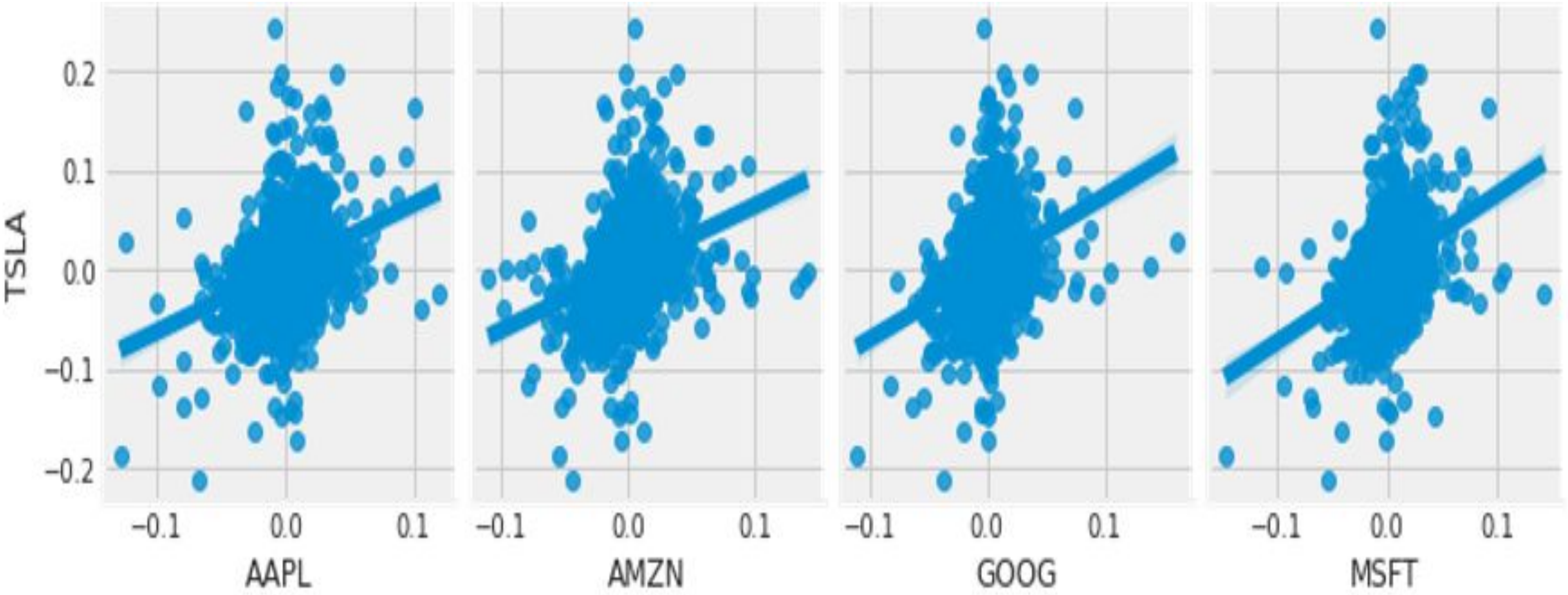


How is the seasonality trends of the Tesla stock?

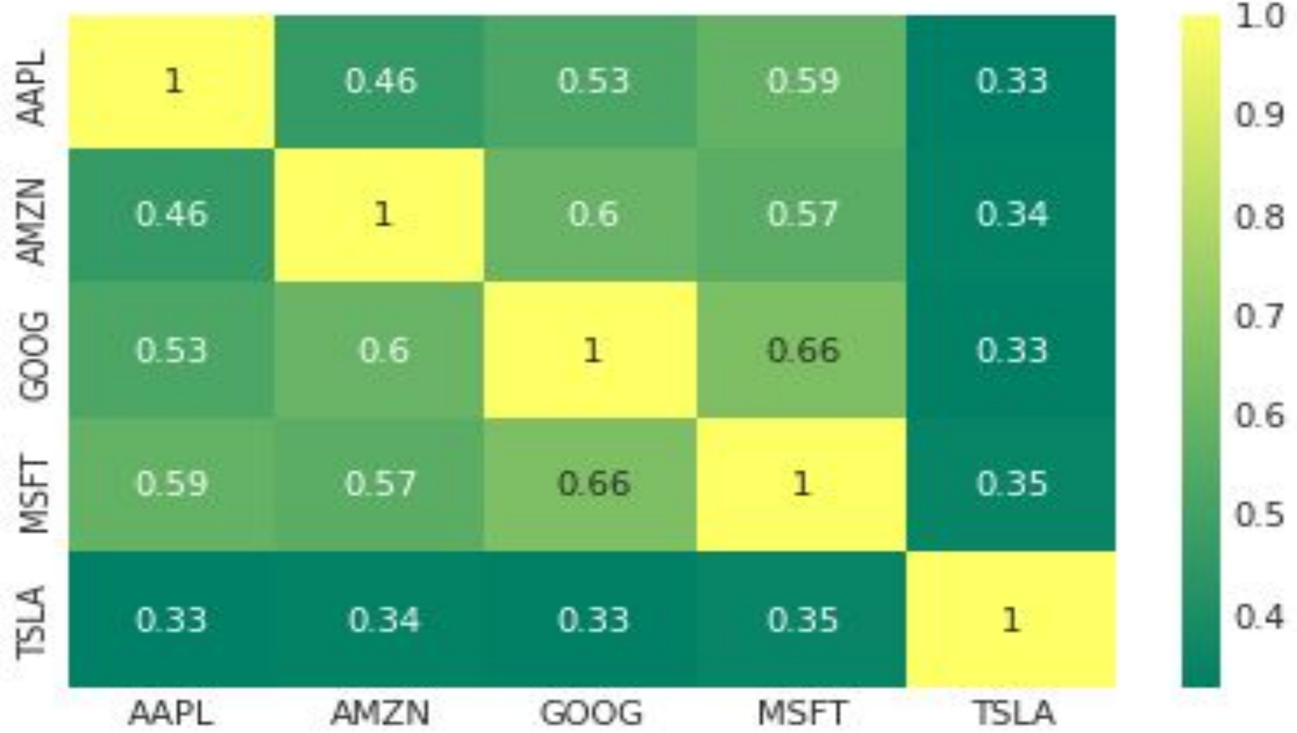


What are the correlation between different stocks ?

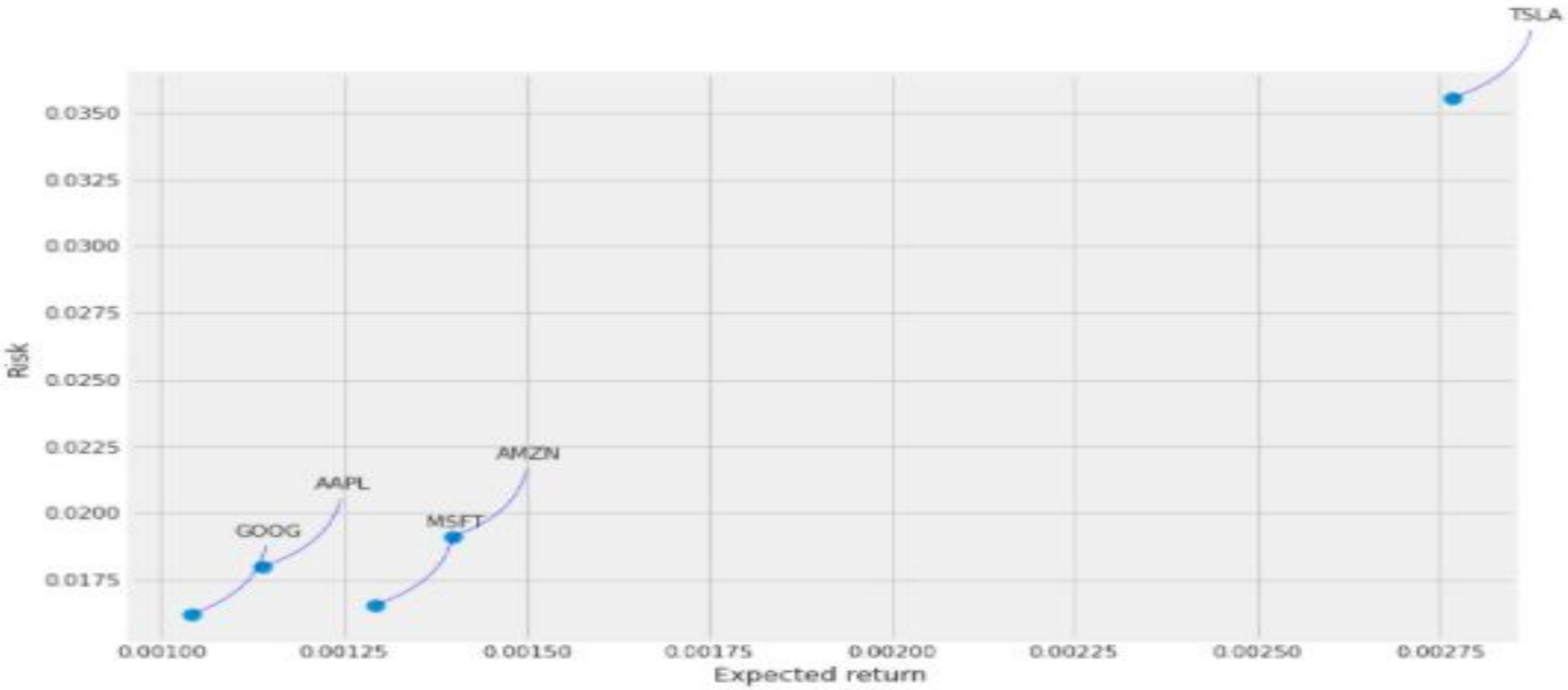
1. Scatter plot between tesla and different stocks



2. Heatmap plot between Tesla and different stocks



How much value do we put at risk by investing in Tesla stock?



How can we attempt to predict Tesla future stock behavior?

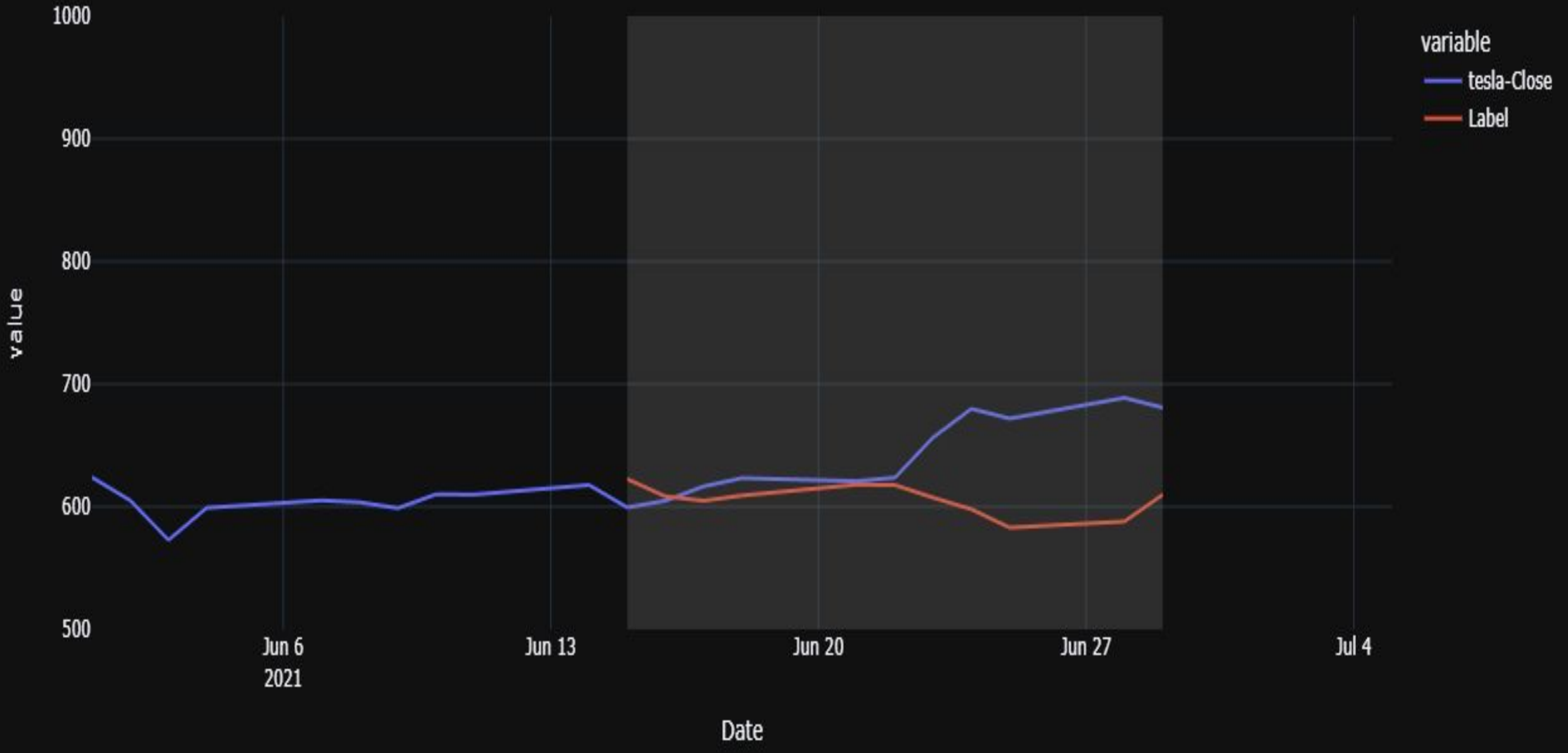
1. First approach: we try to predict the closing stock using a tweaked classical machine learning with time estimators.
2. Second approach: we handle the problem as univariate time series using Prophet and Arima
3. Third approach: Using multivariate time series using different models

First approach

treating problem as normal prediction methodology by filling future dates as our predictors and having Close value as our target.

Using XGboost with 10 folds:

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	22.1544	637.2410	25.2436	-11.7469	1.1022	0.5563
1	5.4029	42.0964	6.4882	-0.6697	0.1463	0.1182
2	4.5799	29.6263	5.4430	-0.7082	0.1121	0.0950
3	6.6143	54.3864	7.3747	-1.7069	0.1636	0.1582
4	13.9693	289.8435	17.0248	-1.1662	0.3195	0.2266
5	7.9148	105.2381	10.2586	-3.0487	0.1708	0.1213
6	6.8581	72.5214	8.5160	-1.2110	0.1316	0.1136
7	9.4374	143.7295	11.9887	-1.2487	0.2224	0.2061
8	129.5731	27609.8398	166.1621	-1.8537	1.2707	0.6053
9	364.9974	200738.2344	448.0382	-8.7161	1.2844	0.5248
Mean	57.1502	22972.2757	70.6538	-3.2076	0.4924	0.2725
SD	108.7794	59817.5728	134.0907	3.6339	0.4810	0.1944

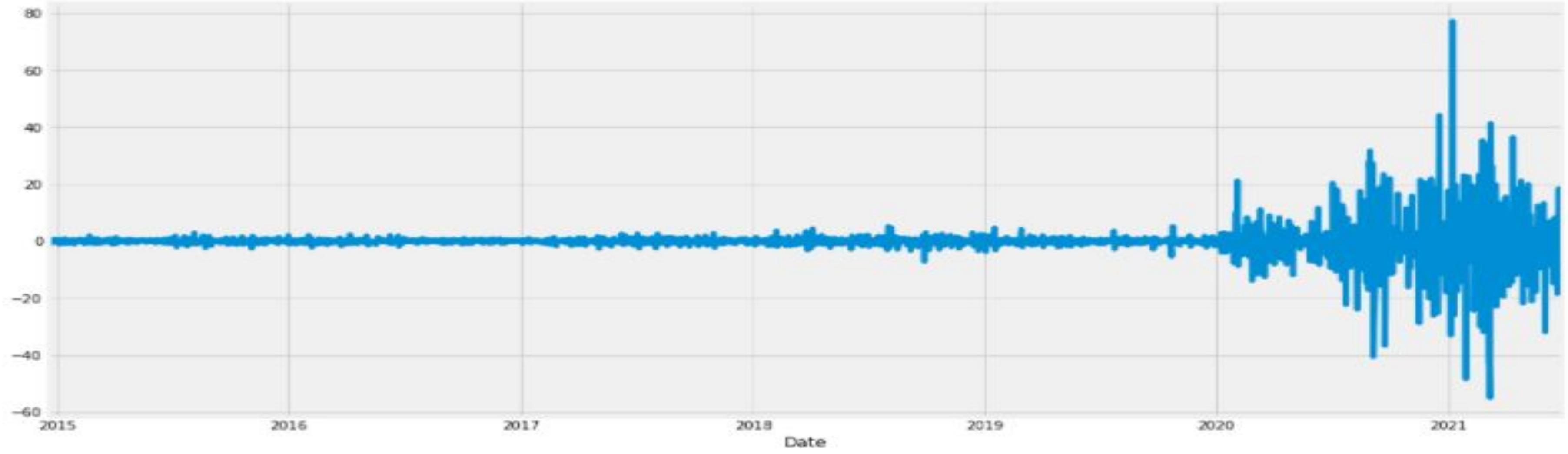


1000
900
800
700
600
500
value
Jun 6
2021
Jun 13
Jun 20
Jun 27
Jul 4
Date
variable
tesla-Close
Label

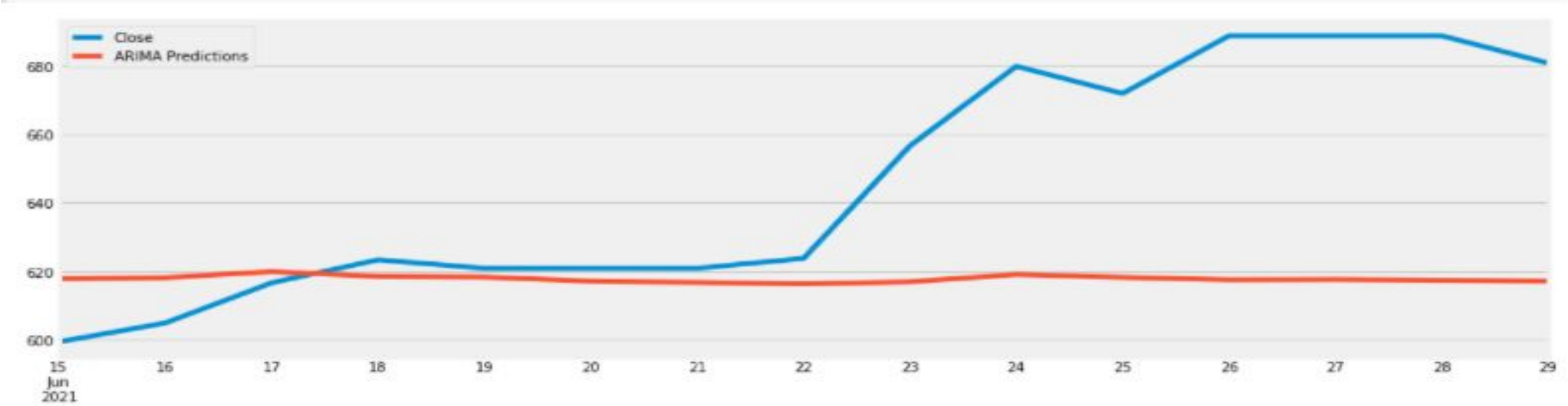
Second approach

Using Prophet and ARIMA modelling to get predictions

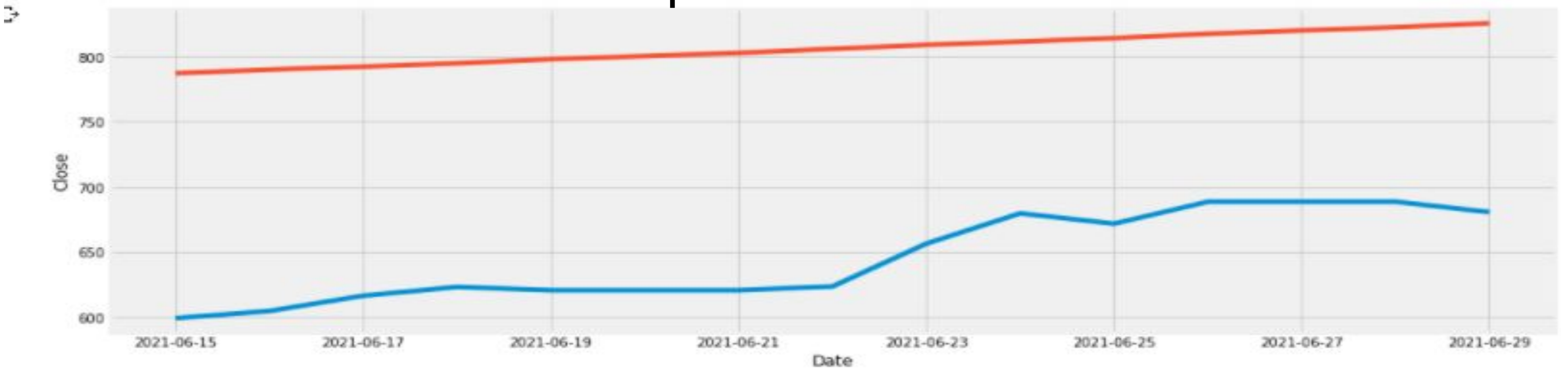
Limitations: The residuality after decomposition of trending, and seasonality starting from 2020 are varying as we see in graph



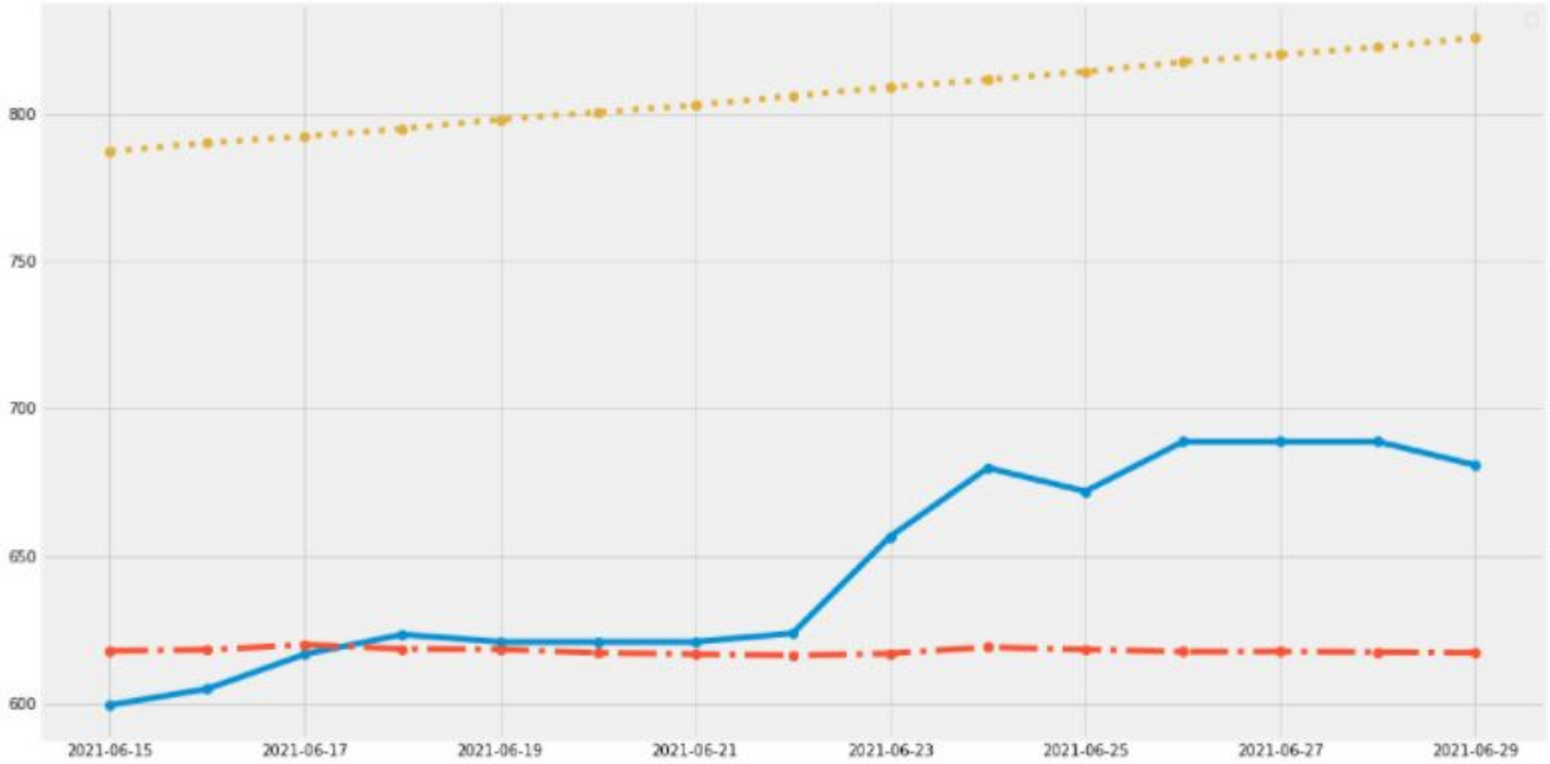
Arima predictions



Prophet predictions



ARIMA vs Prophet Comparison



The blue line represents Tesla close Data, red dashed-dotted line represents ARIMA Predictions and the yellow dotted line represents Prophet Predictions.

Evaluation Comparison

	Models	RMSE Errors	MSE Errors
0	ARIMA	43.271162	1872.393462
1	Prophet	162.000709	26244.229738

Values predictions comparison

	Close	Arima_pred	Prophet_Predictions
Date			
2021-06-15	599.359985	617.771436	787.251045
2021-06-16	604.869995	618.085558	790.086938
2021-06-17	616.599976	619.849097	792.359570
2021-06-18	623.309998	618.446755	794.914609
2021-06-19	620.830017	618.268457	798.062236
2021-06-20	620.830017	617.058106	800.459297
2021-06-21	620.830017	616.723678	802.887908
2021-06-22	623.710022	616.254642	805.992344
2021-06-23	656.570007	616.887264	809.051655
2021-06-24	679.820007	619.024759	811.532382
2021-06-25	671.869995	618.199161	814.272586
2021-06-26	688.719971	617.471682	817.574462
2021-06-27	688.719971	617.540369	820.086756
2021-06-28	688.719971	617.334317	822.583645
2021-06-29	680.760010	617.083318	825.701888

Summary

The objective of this article was to get the basic understanding of time series forecasting models such as ARIMA, and Prophet. From the experiment, we can see that SARIMAX model forecasting has better accuracy than the Prophet model forecasting. The RMSE for the SARIMAX model was around 43 while Prophet Model had RMSE of 162

SARIMAX model was the better model

Third approach

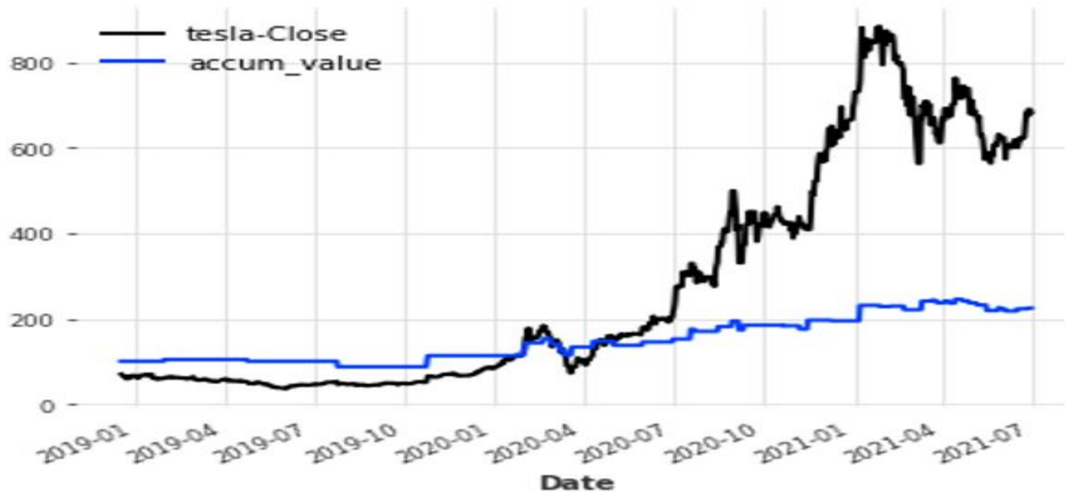
Using multivariate modeling with help of
extra outside sources features

To be able to have a covariate that can help us seeing future, I looked into two things:

1. first was different relevant stocks like technology stocks, or EV-cars competitors.
2. Second was looking into historical major events and milestones of Tesla, then filling a chronological table with this information.

Reference :
https://www.tradingview.com/symbols/NASDAQ-TSLA/history-timeline/#autopilot-saves-lives-says-nhtsa-2017-01-19

Applying EDA for the external features of Major events



We can see the trend is moving up and down with Tesla-Closing values

Correlation analysis

	tesla-Close	accum_value
tesla-Close	1.000000	0.963774
accum_value	0.963774	1.000000

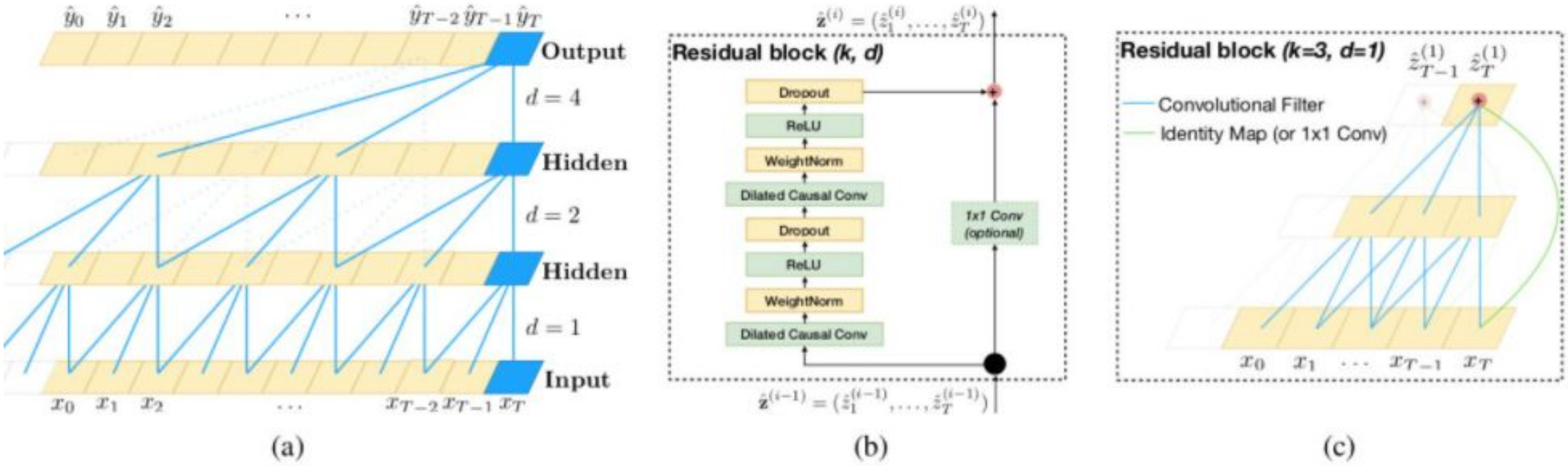
The correlation is very high between target and External features, which makes it a good candidate as a covariate feature

TCN models

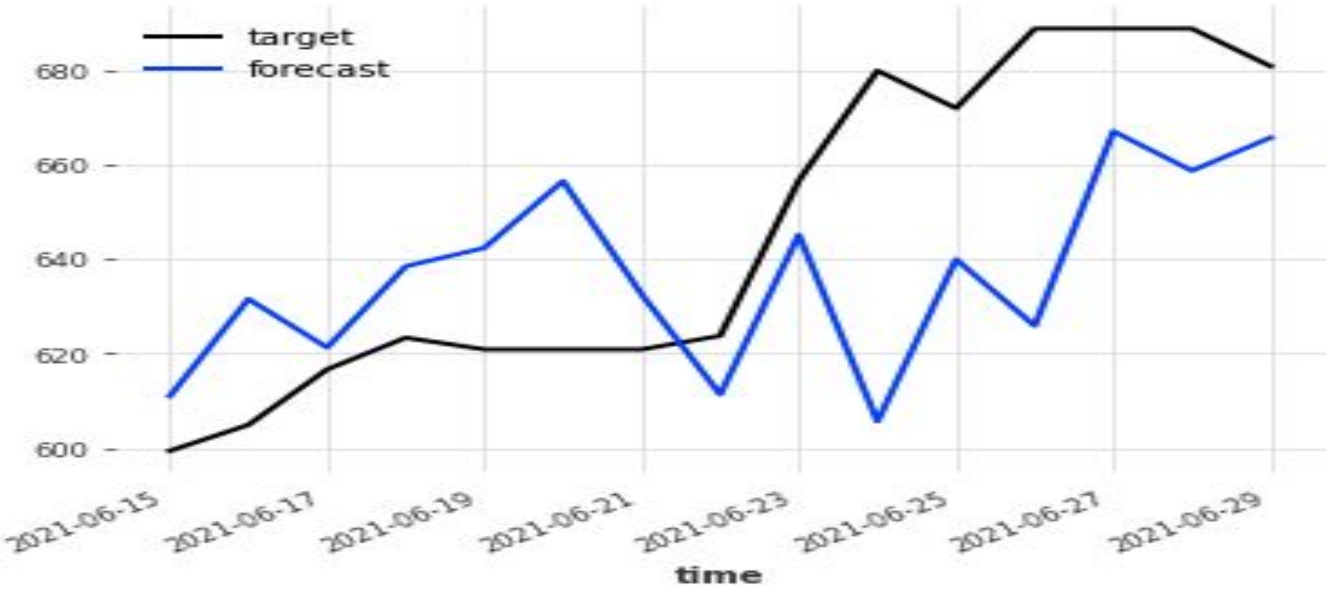
Temporal Convolutional Networks, or simply TCN, is a variation of Convolutional Neural Networks for sequence modelling tasks, by combining aspects of RNN and CNN architectures.

TCN is based upon two principles:

- the fact that the network produces an output of the same length as the input, and
- the fact that there can be no leakage from the future into the past. To accomplish the first point, the TCN uses a 1D fully-convolutional network (FCN) architecture.



TCN predictions vs Truth



TCN values vs Truth

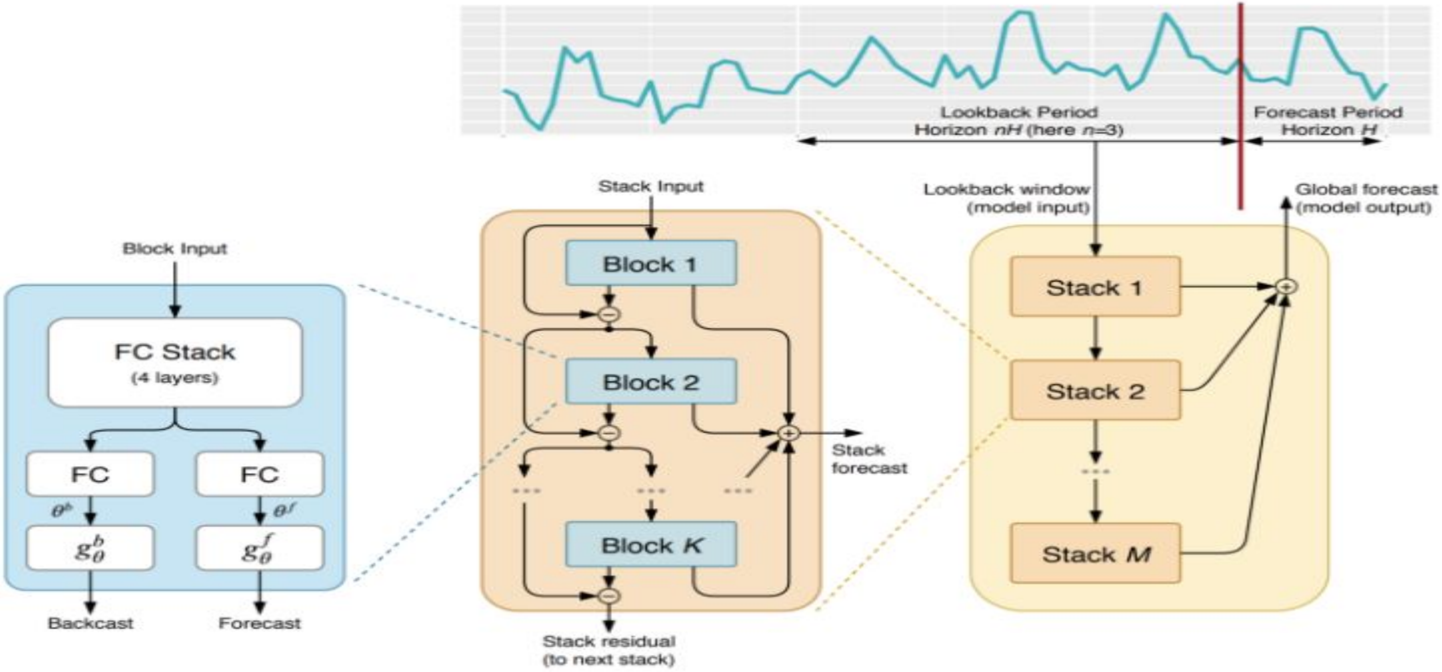
component	Label	tesla-Close
time		
2021-06-15	610.935648	599.359985
2021-06-16	631.486867	604.869995
2021-06-17	621.229285	616.599976
2021-06-18	638.414857	623.309998
2021-06-19	642.312689	620.830017
2021-06-20	656.455396	620.830017
2021-06-21	632.507817	620.830017
2021-06-22	611.303664	623.710022
2021-06-23	645.091970	656.570007
2021-06-24	605.602202	679.820007
2021-06-25	639.884807	671.869995
2021-06-26	625.823887	688.719971
2021-06-27	666.985626	688.719971
2021-06-28	658.653636	688.719971
2021-06-29	665.648219	680.760010

Evaluation Comparison

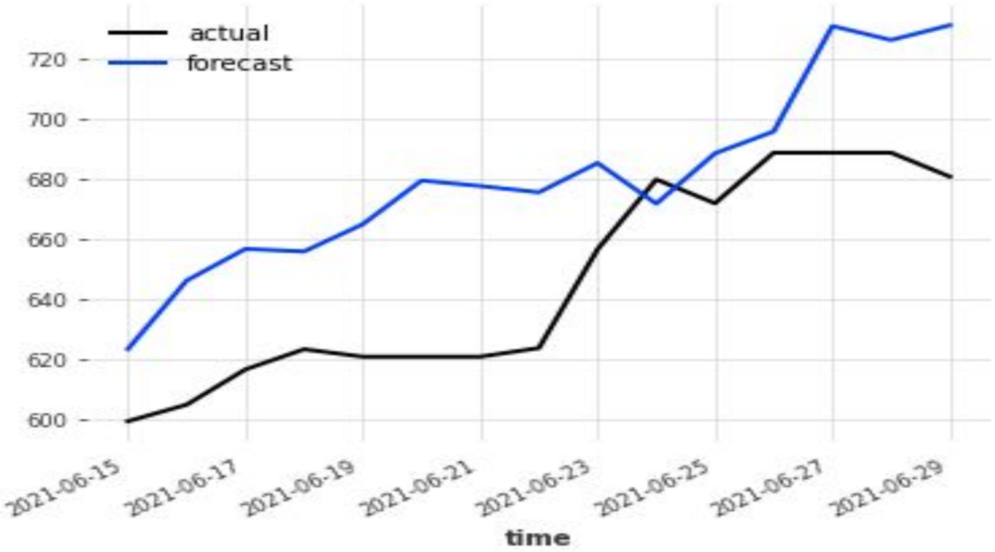
```
TCN model MAPE is: 4%
TCN model MAE is: 25.77
TCN model RMSE is: 32
```

N-BEATS modeling

N-BEATS is **a state-of-the-art model that shows the potential of pure DL architectures in the context of the time-series forecasting**. It outperforms well-established statistical approaches on the M3, and M4 competitions.



Prediction line with NBEATS model



NBEATS values vs Truth

component	Label	tesla-Close
time		
2021-06-15	623.408147	599.359985
2021-06-16	646.183932	604.869995
2021-06-17	656.697504	616.599976
2021-06-18	655.850648	623.309998
2021-06-19	664.923479	620.830017
2021-06-20	679.460944	620.830017
2021-06-21	677.635518	620.830017
2021-06-22	675.544687	623.710022
2021-06-23	685.280159	656.570007
2021-06-24	671.750072	679.820007
2021-06-25	688.492669	671.869995
2021-06-26	695.787317	688.719971
2021-06-27	730.848015	688.719971
2021-06-28	726.215096	688.719971
2021-06-29	731.133424	680.760010

Evaluation Comparison

```
NBEATS model MAPE is: 6%
NBEATS model MAE is: 35.99
NBEATS model RMSE is: 39
```

Multivariate LSTM model using tensorflow framework

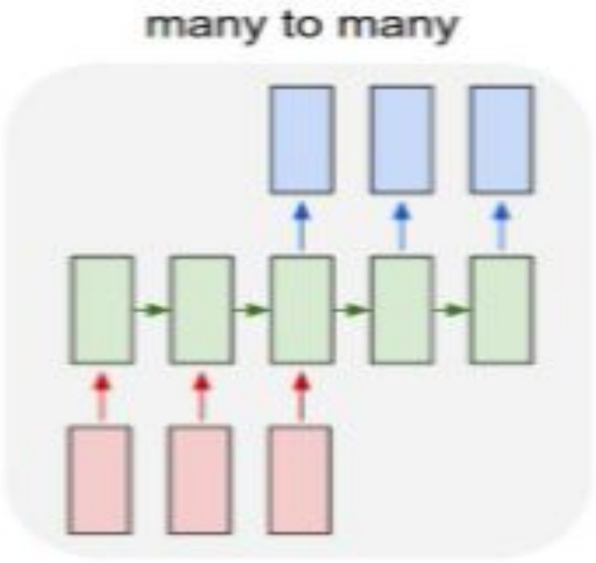
1. Number of features: 5
2. Features are :
   a. Tesla close value.
   b. Tesla volume.
   c. Nasdaq close value.
   d. Nasdaq volume.
   e. Tesla events trends.
3. Models architecture used were:
   a. E1D1 ==> Sequence to Sequence Model with one encoder layer and one decoder layer.
   b. E2D2 ==> Sequence to Sequence Model with two encoder layers and two decoder layers.

Many to many LSTM model

Many-to-Many

Many-to-Many sequence learning can be used for machine translation where the input sequence is in some language, and the output sequence is in some other language. It can be used for Video Classification as well, where the input sequence is the feature representation of each frame of the video at different time steps.

Encoder-Decoder network is commonly used for many-to-many sequence tasks. Here encoder-decoder is just a fancy name for a neural architecture with two LSTM layers.

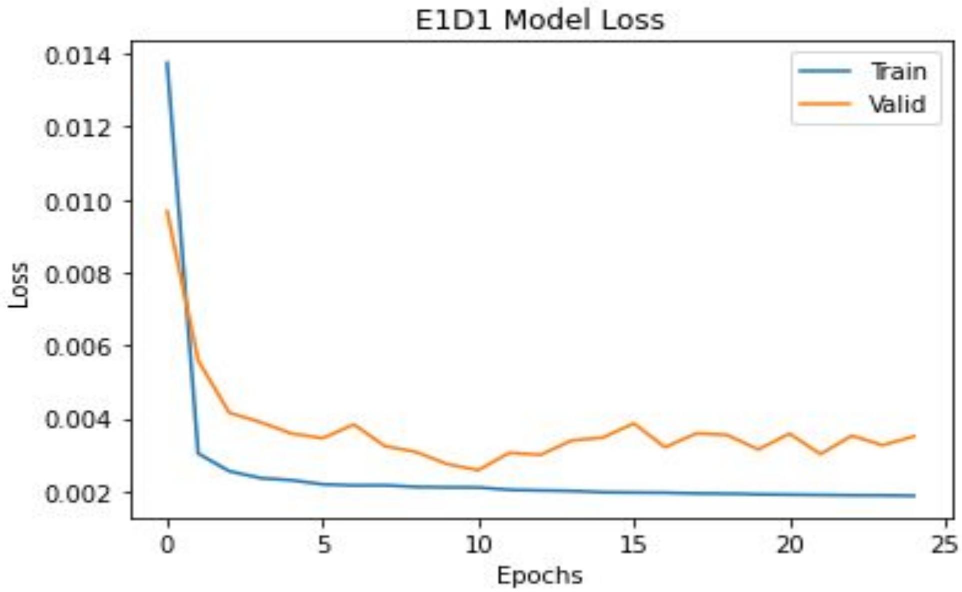


Model one

(Sequence to Sequence Model with one encoder layer and one decoder layer.)

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 35, 5)]      0           []

 lstm (LSTM)                    [(None, 100),        42400       ['input_1[0][0]']
                                 (None, 100),
                                 (None, 100)]

 repeat_vector (RepeatVector)   (None, 15, 100)      0           ['lstm[0][0]']

 lstm_1 (LSTM)                  (None, 15, 100)      80400       ['repeat_vector[0][0]',
                                                                  'lstm[0][1]',
                                                                  'lstm[0][2]']

 time_distributed (TimeDistribu  (None, 15, 5)       505         ['lstm_1[0][0]']
 ted)

==================================================================================================
Total params: 123,305
Trainable params: 123,305
Non-trainable params: 0
```

TRAIN/VALIDATION LOSS CURVE

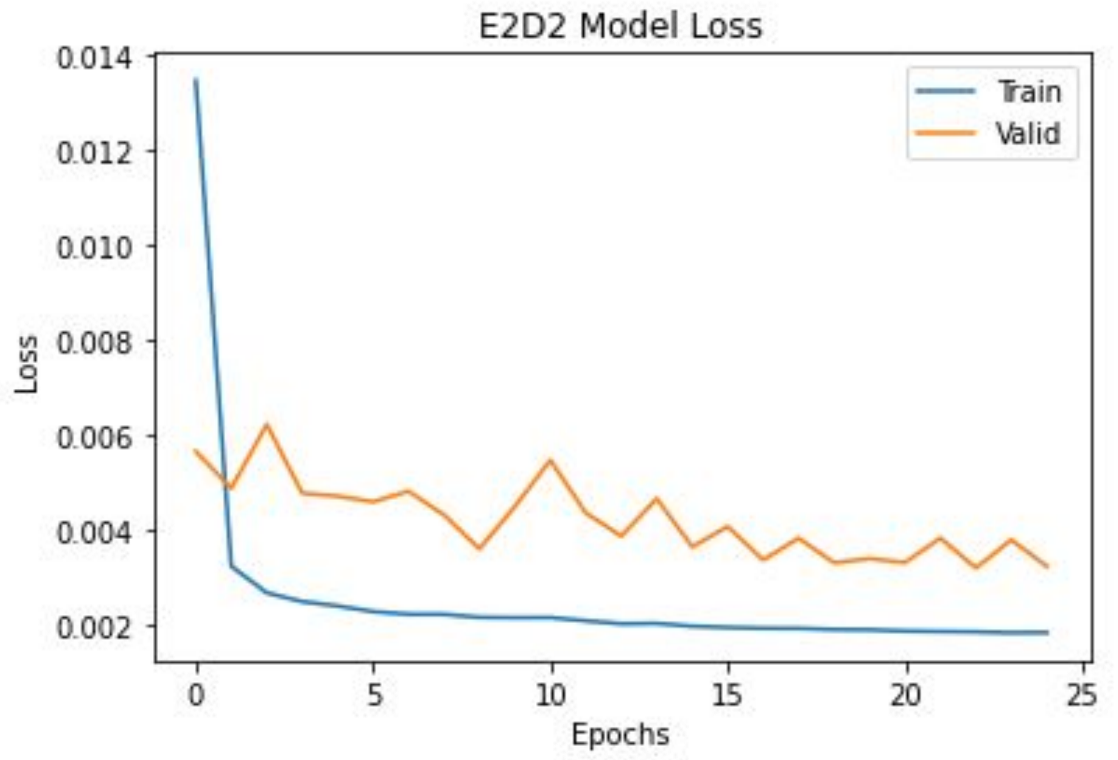


Model two

(Sequence to Sequence Model with two encoder layers and two decoder layers.)

```
____________________________________________________________________________________________
Model: "model_1"
____________________________________________________________________________________________
 Layer (type)                   Output Shape          Param #     Connected to
============================================================================================
 input_2 (InputLayer)           [(None, 35, 5)]       0           []

 lstm_2 (LSTM)                  [(None, 35, 100),     42400       ['input_2[0][0]']
                                 (None, 100),
                                 (None, 100)]

 lstm_3 (LSTM)                  [(None, 100),         80400       ['lstm_2[0][0]']
                                 (None, 100),
                                 (None, 100)]

 repeat_vector_1 (RepeatVector)  (None, 15, 100)      0           ['lstm_3[0][0]']

 lstm_4 (LSTM)                  (None, 15, 100)       80400       ['repeat_vector_1[0][0]',
                                                                   'lstm_2[0][1]',
                                                                   'lstm_2[0][2]']

 lstm_5 (LSTM)                  (None, 15, 100)       80400       ['lstm_4[0][0]',
                                                                   'lstm_3[0][1]',
                                                                   'lstm_3[0][2]']

 time_distributed_1 (TimeDistri  (None, 15, 5)        505         ['lstm_5[0][0]']
 buted)

============================================================================================
Total params: 284,105
Trainable params: 284,105
Non-trainable params: 0
```

TRAIN/VALIDATION LOSS CURVE



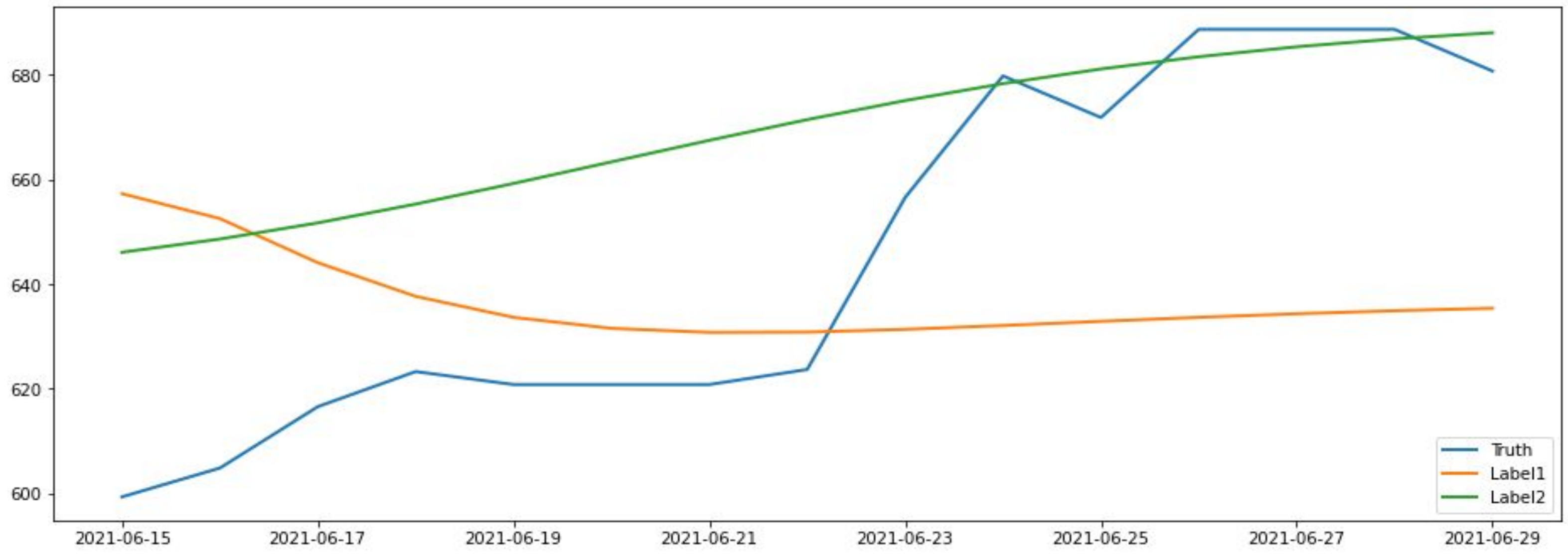
LSTM model values vs Truth

Date	tesla-Close	Label1	Label2
2021-06-15	599.359985	656.609375	670.043579
2021-06-16	604.869995	676.165039	675.748901
2021-06-17	616.599976	683.791260	678.459106
2021-06-18	623.309998	688.951660	680.256897
2021-06-19	620.830017	693.623596	682.378662
2021-06-20	620.830017	698.225708	685.027893
2021-06-21	620.830017	702.661316	687.998169
2021-06-22	623.710022	706.740417	691.000000
2021-06-23	656.570007	710.321106	693.795654
2021-06-24	679.820007	713.338074	696.233276
2021-06-25	671.869995	715.788635	698.239990
2021-06-26	688.719971	717.714294	699.802795
2021-06-27	688.719971	719.178406	700.948853
2021-06-28	688.719971	720.254700	701.726135
2021-06-29	680.760010	721.015930	702.193237

Evaluation Comparison

	LSTM1	LSTM2
MAPE	5.19	4.07
MAE	33.91	25.34
RMSE	39.00	31.00

Prediction lines with LSTM1 and LSTM2 models



Blue indicates the True data, Orange indicates the first prediction data, Green indicates the second prediction data.

Summary of the
four multivariate models

Comparison of models predictions

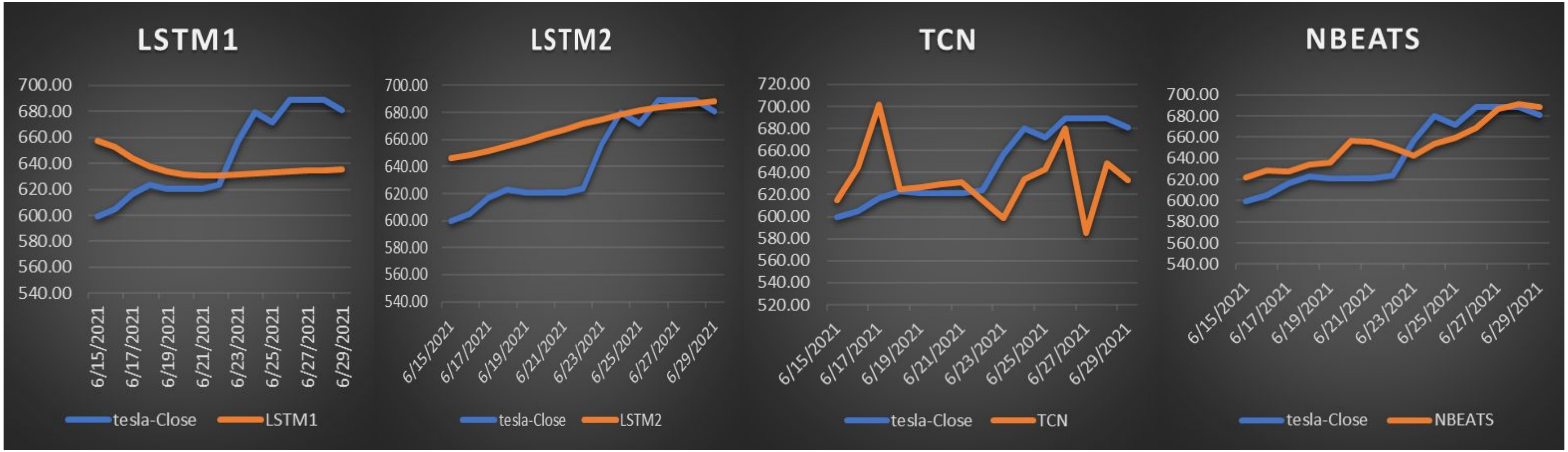
Date	tesla-Close	LSTM1	LSTM2	TCN	NBEATS
6/15/2021	599.36	657.31	646.11	615.00	622.42
6/16/2021	604.87	652.55	648.64	644.80	628.87
6/17/2021	616.60	644.12	651.73	701.51	627.39
6/18/2021	623.31	637.68	655.32	624.52	634.48
6/19/2021	620.83	633.67	659.26	626.93	635.85
6/20/2021	620.83	631.57	663.39	629.68	656.92
6/21/2021	620.83	630.80	667.51	631.24	655.84
6/22/2021	623.71	630.86	671.46	614.64	650.08
6/23/2021	656.57	631.38	675.10	598.21	642.30
6/24/2021	679.82	632.11	678.35	633.63	654.11
6/25/2021	671.87	632.91	681.14	642.63	659.34
6/26/2021	688.72	633.68	683.48	680.48	668.78
6/27/2021	688.72	634.36	685.39	585.46	686.86
6/28/2021	688.72	634.94	686.90	648.36	691.10
6/29/2021	680.76	635.41	688.07	632.83	688.28

Evaluation metrics

	LSTM1	LSTM2	TCN	NBEATS
MAPE	5.19%	4.07%	5.18%	2.79%
MAE	33.91	25.34	33.98	17.71
RMSE	39	31	45	20

With comparison of the different multivariate models we find that N-BEATS model was having the best evaluation metrics

Visualization of predictions



Summary

After choosing Deep learning approaches, we find out that the best approach was by adapting external data, and using multivariate time series approach, as **NBEATS, LSTM** and **TCN** had overall better MAE, MAPE, and RMSE.

Recommendation is to try further use an ensemble between these models.

Outlook

The usage of past covariates was effective in the multivariate deep learning models, we would look further into implementing future covariates that we can predict already like certain political events, or even weather forecast, combined with domain experts will surely enhance the type and quality of our effective data.