

Supervisor: Sultanmurat Yeleu

Apple Stock price prediction using Python (Google Colab) and Facebook Prophet

(Kirill Lukyanchuk and Nygmetzhan Akhmetov IT-2105)

<https://github.com/Nygmet29/AP.git>
https://youtu.be/_eJq6bwK2is

- **Overview**
 - Introduction
 - Problem
 - Literature review
 - Current work
 - Data and methods
 - Information about the data
 - Description of ML models
 - Discussion
 - Critical review of results
- References

- **Structure**

- **Introduction**

- Problem

Apple Stock price prediction using Python (Google Colab) and Facebook Prophet - is the program written in Google Colab with imported Facebook Prophet library to predict prices. We decided to develop this project, since Stock Prices of world companies are inspected by millions of investors and program that will predict not only Apple's stock prices, but other companies' stock values (if we import history prices) will be quite effective application.

- Literature review (links that we used are at the end of the report)

From Nasdaq website we downloaded our dataset of Apple Inc. stock price in (.csv) format since 2018th year till 2025th.

- Current work

1) Firstly, we imported pandas library in order to analyze our dataset

```
[ ] #import pandas library
import pandas as pd
```

2) We wrote this code in order to import our dataset from PC

```
#import our dataset from PC
from google.colab import files
files.upload()
```

3) Then we created dataframe that copied all information from our dataset, after that we removed \$ sign from the values of "Close/Last" column, because pandas reads only the numeric values, and then inverted our list of dataset.

```
df = pd.read_csv('AAPL.csv')
df['Close/Last'] = df['Close/Last'].str.replace('$', '')

df['Close/Last'] = pd.to_numeric(df['Close/Last'])

df = df.iloc[::-1]
df
```

	Date	Close/Last	Volume	Open	High	Low
1258	02/20/2018	42.9625	134762640	\$43.0125	\$43.565	\$42.855
1257	02/21/2018	42.7675	149512280	\$43.2075	\$43.53	\$42.7525
1256	02/22/2018	43.1250	123815040	\$42.95	\$43.4875	\$42.9275
1255	02/23/2018	43.8750	135088200	\$43.4175	\$43.9125	\$43.385
1254	02/26/2018	44.7425	149414680	\$44.0875	\$44.8475	\$44.0525
...
4	02/13/2023	153.8500	62199010	\$150.952	\$154.26	\$150.92
3	02/14/2023	153.2000	61707570	\$152.12	\$153.77	\$150.86
2	02/15/2023	155.3300	65669250	\$153.11	\$155.5	\$152.88
1	02/16/2023	153.7100	68167940	\$153.51	\$156.33	\$153.3475
0	02/17/2023	152.5500	59144120	\$152.35	\$153	\$150.85

1259 rows × 6 columns

4) We renamed "Date" and "Close/Last" columns to "ds" and "y", because Facebook prophet reads and outputs in 'ds' and 'y' format.

```
df = df[['Date', 'Close/Last']]  
df = df.rename(columns={'Date': 'ds', 'Close/Last': 'y'})  
df
```

	ds	y
1258	02/20/2018	42.9625
1257	02/21/2018	42.7675
1256	02/22/2018	43.1250
1255	02/23/2018	43.8750
1254	02/26/2018	44.7425
...
4	02/13/2023	153.8500
3	02/14/2023	153.2000
2	02/15/2023	155.3300
1	02/16/2023	153.7100
0	02/17/2023	152.5500

1259 rows × 2 columns

5) We separated last 30 days and placed them into the array named "last30days_for_pred" in order to compare predicted values with the actual values from this array.

```
last30days_for_pred = df[len(df)-30:]
last30days_for_pred
```

	ds	y
29	01/06/2023	129.62
28	01/09/2023	130.15
27	01/10/2023	130.73
26	01/11/2023	133.49
25	01/12/2023	133.41
24	01/13/2023	134.76
23	01/17/2023	135.94
22	01/18/2023	135.21
21	01/19/2023	135.27
20	01/20/2023	137.87
19	01/23/2023	141.11
18	01/24/2023	142.53
17	01/25/2023	141.86
16	01/26/2023	143.96

6) Then we removed last 30 days from dataframe.

```
[ ] #remove last 30 days from dataframe
df = df[:-30]
df
```

	ds	y
1258	02/20/2018	42.9625
1257	02/21/2018	42.7675
1256	02/22/2018	43.1250
1255	02/23/2018	43.8750
1254	02/26/2018	44.7425
...
34	12/29/2022	129.6100
33	12/30/2022	129.9300
32	01/03/2023	125.0700
31	01/04/2023	126.3600
30	01/05/2023	125.0200

7) Here we installed fbprophet library in order to forecast prices

```
# installation of pystan
pip install pystan==2.19.1.1

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pystan==2.19.1.1
  Downloading pystan-2.19.1.1-cp38-cp38-manylinux1_x86_64.whl (62.6 MB)
    62.6/62.6 MB 11.3 MB/s eta 0:00:00
Requirement already satisfied: Cython!=0.25.1,>=0.22 in /usr/local/lib/python3.8/dist-packages (from pystan==2.19.1.1) (0.29.33)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.8/dist-packages (from pystan==2.19.1.1) (1.21.6)
Installing collected packages: pystan
  Attempting uninstall: pystan
    Found existing installation: pystan 3.3.0
    Uninstalling pystan-3.3.0:
      Successfully uninstalled pystan-3.3.0
  Successfully installed pystan-2.19.1.1

# installation of Facebook prophet
pip install fbprophet

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting fbprophet
  Using cached fbprophet-0.7.1.tar.gz (64 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: Cython>=0.22 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (0.29.33)
Requirement already satisfied: cmdstanpy==0.9.5 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (0.9.5)
Requirement already satisfied: pystan>=2.14 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (2.19.1.1)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (1.21.6)
Requirement already satisfied: pandas>=1.0.4 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (1.3.5)
Requirement already satisfied: matplotlib>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (3.2.2)
Requirement already satisfied: LunarCalendar>=0.0.9 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (0.0.9)
Requirement already satisfied: convertdate>=2.1.2 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (2.4.0)
Requirement already satisfied: holidays>=0.10.2 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (0.19)
Requirement already satisfied: setuptools-git>=1.2 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (1.2)
Requirement already satisfied: python-dateutil>=2.8.0 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (2.8.2)
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.8/dist-packages (from fbprophet) (4.64.1)
Requirement already satisfied: pymeeus<1,>=0.3.13 in /usr/local/lib/python3.8/dist-packages (from convertdate>=2.1.2->fbprophet) (0.3.13)
Requirement already satisfied: korean-lunar-calendar in /usr/local/lib/python3.8/dist-packages (from holidays>=0.10.2->fbprophet) (1.0.1)
```

■ Data & Methods

8) Here we fitted our Prophet object, then we created predicted future dataframe for the next 2 years and implemented it to forecast object.

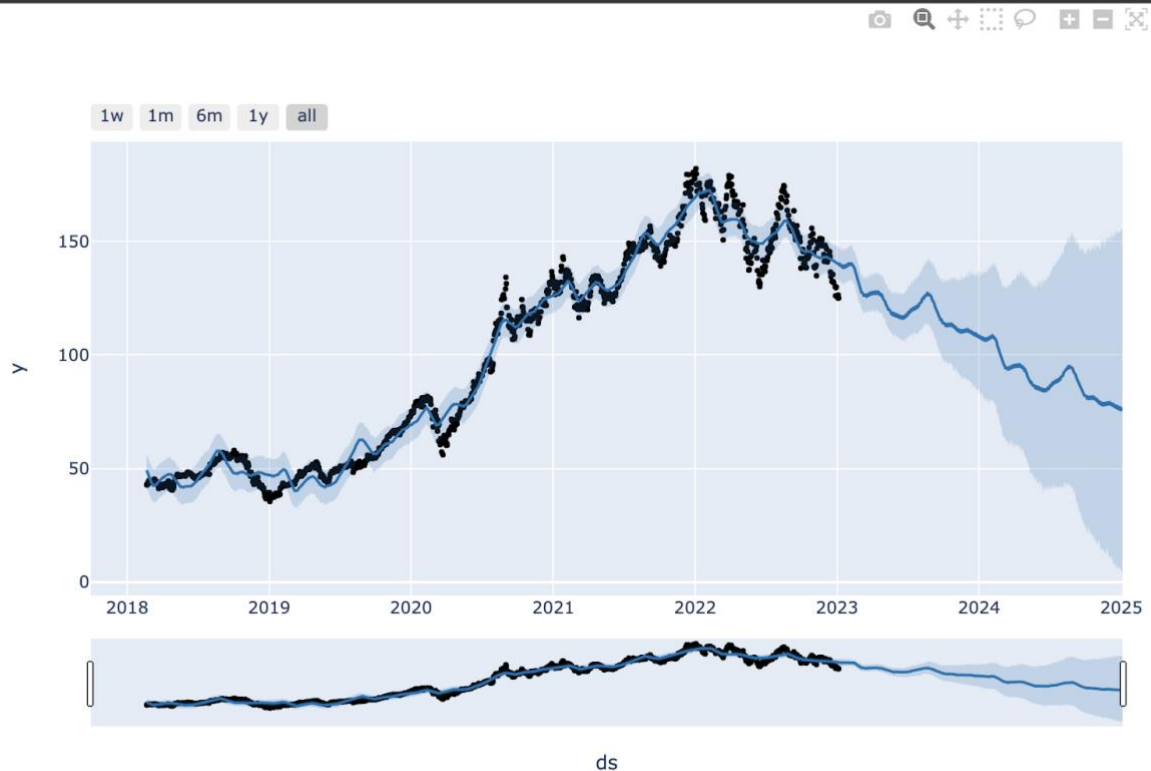
```
[ ] # here prophet library imported, object of prophet, future object is created and predicted and model is trained
from fbprophet import Prophet
fbp = Prophet(daily_seasonality = True)
fbp.fit(df)
future = fbp.make_future_dataframe(periods=730)
forecast = fbp.predict(future)
```

In the given code the object of Prophet (fbp) fitted all data from 'df' which is dataframe that consists dataset copy. Then future object trained dataset for the next 730 days according given dataset, and forecast function made a predict of future object.

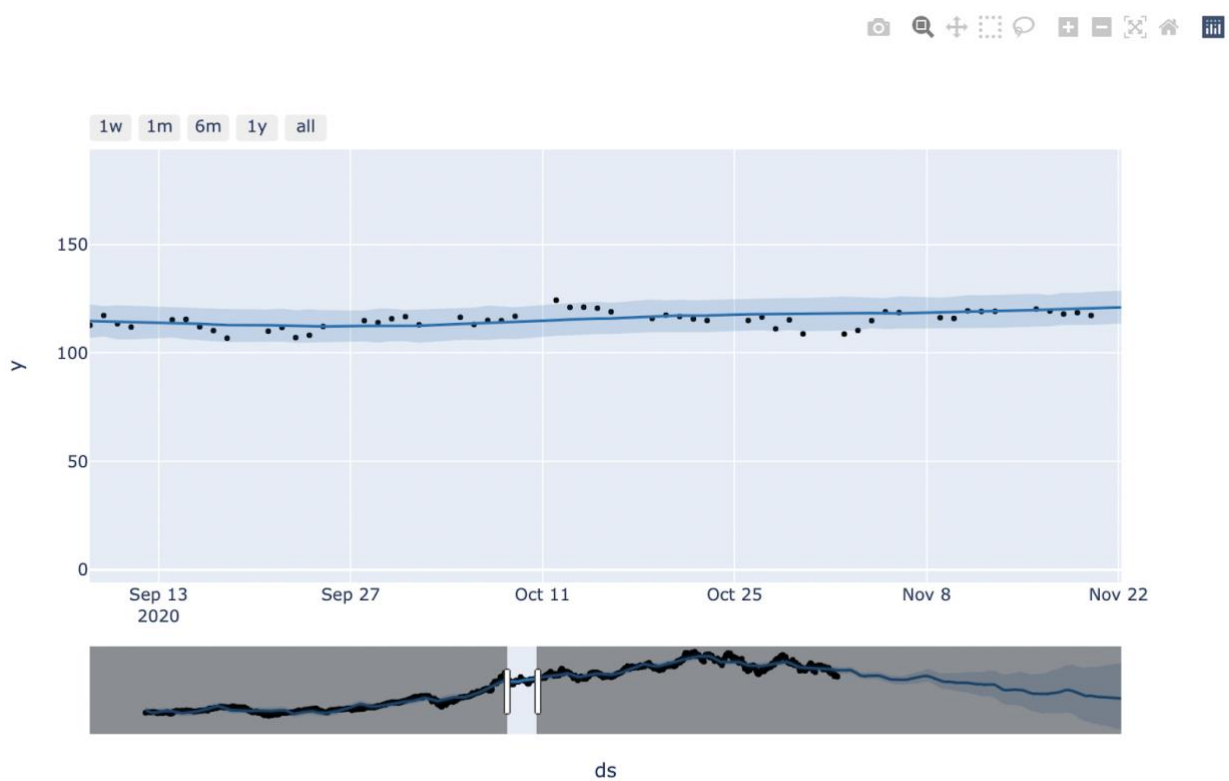
▪ Results

9) Written code shows the graph of dataframe with predicted values, where blue line is predicted value and black dots are actual values from dataset

```
# plot from fbrprophet is used to show the graph
from fbprophet.plot import plot_plotly
plot_plotly(fbp, forecast)
```



In these graphs you can see how the stock price is changing every day, week, month, year and you also can choose a period of time and see how the stock price changed during a certain period of time.



10) here we can predict any value from 2018 till 2025, it shows our predicted value

```
[ ] # here we can predict any value from 2018 till 2025
    forecast[forecast.ds == '02/14/2023']['yhat']

1268      138.452488
Name: yhat, dtype: float64
```

11) Here you can see the actual value and if you compare with the predicted, you can see the difference

```
# here we can output the actual value in order to compare and check the difference
last30days_for_pred[last30days_for_pred.ds == '02/14/2023']['y']

3      153.2
Name: y, dtype: float64
```

As you can see, our predicted value was “138.452488”, but actual value was “153.2”. So the difference is 14.747512.

▪ Discussion

When we saw the graph, we thought that predicted value will be higher, but it dramatically decreased. Also, as we said in the 5th step, we put away some array with 30 days stock price information, and then deleted it from whole dataframe in order to check the correctness ratio, and difference between predicted price and actual value was quite more than normal. And, surprisingly, our “future” object modeled predicted values for whole dataset from 2018 till 2025 years.

References

- Website from where we downloaded the dataset of Apple stock price for the last 5 years (2018-2023) - <https://www.nasdaq.com/market-activity/stocks/aapl/historical>
- Website from where we acknowledged how to connect Facebook Prophet library to our Google Colab - <https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/#:~:text=To%20use%20Prophet%20for%20forecasting,type%20of%20seasonality%2C%20and%20more.>
- From StackOverflow we got aware how to invert the dataset list - <https://stackoverflow.com/questions/10933838/how-to-read-a-csv-file-in-reverse-order-in-python>