Future Stock Price Prediction for MAANG Companies

Architectural Decisions Document Template

1 Architectural Components Overview

1.1 Data Source

1.1.1 Technology Choice

The data source used for getting the recent stock prices for the MAANG companies is Yahoo Finance (Yahoo Finance). I have used their yfinance (yfinance) API to web scrap the recent stock price data for the required companies. The data includes the daily Opening, Highest, Lowest, Closing and Adjusted Closing prices of a company from its inception. The web scrapping of the data is shown in the ETL file. The DataFrame generated after the web scrapping of the data is shown in Figure 1, below:

Figure 1: DataFrame generated for the Amazon Stock Price

Date	Open	High	Low	Close	Adj Close	Volume
1997-05-15	0.121875	0.125000	0.096354	0.097917	0.097917	1443120000
1997-05-16	0.098438	0.098958	0.085417	0.086458	0.086458	294000000
1997-05-19	0.088021	0.088542	0.081250	0.085417	0.085417	122136000
1997-05-20	0.086458	0.087500	0.081771	0.081771	0.081771	109344000
1997-05-21	0.081771	0.082292	0.068750	0.071354	0.071354	377064000

1.1.2 Justification

The reason for using yfinance (yfinance) API was to get the most recent data directly from the online source.

1.2 Enterprise Data

1.2.1 Technology Choice

Enterprise Data was not used in this project.

1.2.2 Justification

Enterprise Data was not used in this project.

1.3 Streaming analytics

1.3.1 Technology Choice

This model is used to predict daily closing stock price prediction. The model would be taking daily stock prices from the Yahoo Finance directly using yfinance (yfinance) API and predict the next day closing price for the stock.

1.3.2 Justification

To get the daily updated data.

1.4 Data Integration

1.4.1 Technology Choice

This model is used to predict daily closing stock price prediction. The extracted data from the Yahoo Finance contains the daily Opening, Highest, Lowest, Closing and Adjusted Closing prices of the company from its inception. The Figure 2 below, shows the summary statistics for the extracted data for Amazon company's stock prices and the null values in each column. Similar statistics were generated for other MAANG companies.

Figure 2: Summary Statistics for Amazon Company with null values in each column

Description of data for Amazon Company

	Open	High	Low	Close	Adj Close	Volume
count	6498.000000	6498.000000	6498.000000	6498.000000	6498.000000	6.498000e+03
mean	31.425786	31.802941	31.010145	31.412629	31.412629	1.427673e+08
std	48.031573	48.593805	47.401699	47.994680	47.994680	1.402839e+08
min	0.070313	0.072396	0.065625	0.069792	0.069792	9.744000e+06
25%	1.995125	2.025000	1.957500	2.000125	2.000125	6.901152e+07
50%	6.399250	6.497000	6.304750	6.417000	6.417000	1.060610e+08
75%	38.288124	38.524001	38.026374	38.312751	38.312751	1.609865e+08
max	187.199997	188.654007	184.839493	186.570496	186.570496	2.086584e+09

```
Null values in the data for Amazon Company
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

As our model is future stock closing price prediction, we have to transform the data such that past timestamp data of closing price is assigned as features and future data is assigned as label. The length of past timestamps data based on which the future data is predicted is decided by us. The length of past timestamps data decided by us in this project is 10.

Therefore, the final dataset contains 10 features having time series data of past 10 days and the target data is the data for 11th day and so on. As there were no null values in the data, **we do not require any data imputation**. As we were using a Neural Network model a **MinMax scaler** was used to scale the data. Figure 3 below shows the DataFrame for the final transformed data.

Figure 3: Final Transformed Data for Amazon

Date	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7	Feature8	Feature9	Feature10	Target
1997-07-03	0.000031	0.000035	0.000028	0.000031	0.000031	0.000031	0.000025	0.000039	0.000032	0.000052	0.000138
1997-07-07	0.000035	0.000028	0.000031	0.000031	0.000031	0.000025	0.000039	0.000032	0.000052	0.000138	0.000162
1997-07-08	0.000028	0.000031	0.000031	0.000031	0.000025	0.000039	0.000032	0.000052	0.000138	0.000162	0.000243
1997-07-09	0.000031	0.000031	0.000031	0.000025	0.000039	0.000032	0.000052	0.000138	0.000162	0.000243	0.000249
1997-07-10	0.000031	0.000031	0.000025	0.000039	0.000032	0.000052	0.000138	0.000162	0.000243	0.000249	0.000313
...	...	...	...	...	...	...	...	...	...	...	...
2023-03-06	0.520803	0.506755	0.513243	0.513404	0.500964	0.502358	0.504879	0.493833	0.493619	0.508471	0.502305
2023-03-07	0.506755	0.513243	0.513404	0.500964	0.502358	0.504879	0.493833	0.493619	0.508471	0.502305	0.501232
2023-03-08	0.513243	0.513404	0.500964	0.502358	0.504879	0.493833	0.493619	0.508471	0.502305	0.501232	0.503216
2023-03-09	0.513404	0.500964	0.502358	0.504879	0.493833	0.493619	0.508471	0.502305	0.501232	0.503216	0.494262
2023-03-10	0.500964	0.502358	0.504879	0.493833	0.493619	0.508471	0.502305	0.501232	0.503216	0.494262	0.486112

1.4.2 Justification

1. Summary Statistics to get the statistics of each column in the data.
2. No data imputation because no null values were present.
3. Organization of target data based on timestamps because of time series prediction.
4. MinMax scaler to scale the data, as the model used for prediction is a neural network model.

1.5 Data Repository

1.5.1 Technology Choice

Data repository used for this project is the IBM cloud object storage.

1.5.2 Justification

1. Massive Scalability
2. Reduced Complexity
3. Searchability
4. Resiliency
5. Cost efficiency

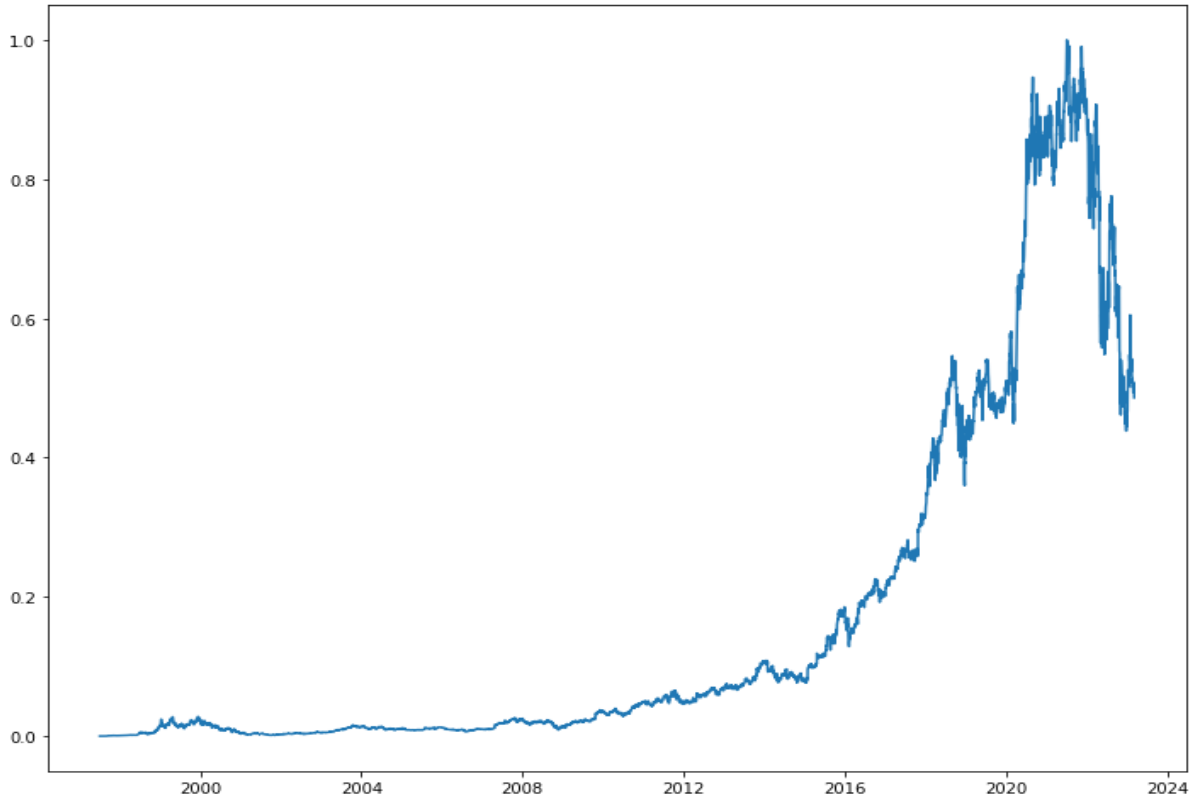
1.6 Discovery and Exploration

1.6.1 Technology Choice

Python packages used for this project are Tensorflow, Keras, Scikit-learn, Pandas, Numpy, and Matplotlib. And, coding was done on Jupyter notebooks. The time series data of the

companies is plotted using the Matplotlib package. Figure 4 below, shows the scaled actual time series data for closing stock price for Amazon.

Figure 4: Scaled actual time series plot for closing stock price for Amazon



1.6.2 Justification

All the packages used are the state of the art.

1.7 Actionable Insights

1.7.1 Technology Choice

The model used in this project to predict the future price of the MAANG companies is Stateful LSTM model, with the final sequence output given to a dense layer and taking a unit output. The states of the LSTM were reset at the end of each epoch. The metrics used to evaluate the model performance are the mean absolute error (MAE), mean squared error (MSE), and R2 score. The final LSTM model is compared with the benchmark XGBoost model trained using Grid Search over a set of hyperparameters.

Table 1 below, shows the performance of the stateful LSTM model on train, test and validation data for each MAANG company.

Table 1: Stateful LSTM model performance

Company	Mean Absolute Error Train	Mean Absolute Error Val	Mean Absolute Error Test	Mean Squared Error Train	Mean Squared Error Val	Mean Squared Error Test	R2 Score Train	R2 Score Val	R2 Score Test
Amazon	0.008395	0.007295	0.028893	0.000073	0.000103	0.001199	0.982167	0.979215	0.943841

Apple	0.008801	0.006974	0.018277	0.000078	0.000055	0.000628	0.914197	0.976146	0.989179
Google	0.007533	0.007606	0.028573	0.000064	0.000113	0.001239	0.994125	0.983882	0.918329
Meta	0.007951	0.030599	0.015897	0.000109	0.001318	0.000575	0.996665	0.870148	0.981586
Netflix	0.008531	0.011462	0.019239	0.000081	0.000249	0.000709	0.994139	0.980316	0.985008

Table 2 below, shows the performance of the XGBoost model on train, test and validation data for each MAANG company.

Table 2: XGBoost model performance

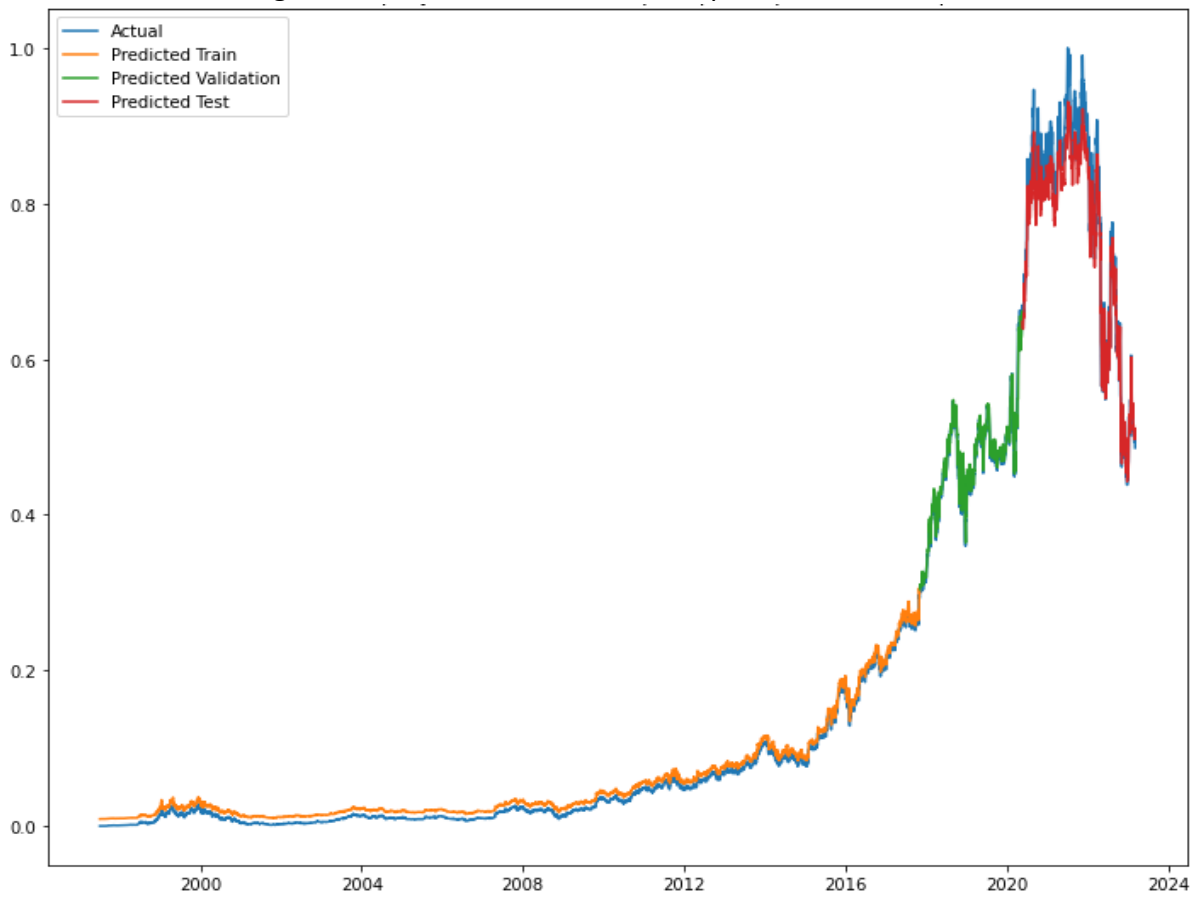
Company	Mean Absolute Error Train	Mean Absolute Error Test	Mean Squared Error Train	Mean Squared Error Test	R2 Score Train	R2 Score Test
Amazon	0.000435	0.293438	0.000000	0.121234	0.999909	-2.451219
Apple	0.000166	0.264855	0.000000	0.143941	0.999893	-0.945502
Google	0.000614	0.257838	0.000001	0.102264	0.999936	-1.766176
Meta	0.004520	0.066191	0.000044	0.009997	0.998754	0.820247
Netflix	0.000417	0.091933	0.000000	0.020096	0.999968	0.376608

We can make the following observations by comparing Table 1 and Table 2:

1. XGBoost does well on the train data but is unable do well on the test data as it do not have the ability to capture sequence information and the data in the test data is not from the same distribution which is a requirement for the XGBoost model.
2. Stateful LSTM is able to do equally well on the test data and is able to capture the sequence information.

Figure 4 below shows the plot for scaled actual daily stock prices along with the model predicted stock prices for Amazon on test, train and Validation data.

Figure 4: Actual vs. Predicted stock prices for Amazon



1.7.2 Justification

1. LSTM stateful model is used because it can capture the sequential information and is widely used in time series use cases.
2. States were reset after the end of each epoch because it could keep the state information throughout one epoch and one run over the whole data.
3. XGBoost is considered as a benchmark model as it is a very powerful ML algorithm, which can capture non-linear interactions as well.

1.8 Applications / Data Products

1.8.1 Technology Choice

This model can be used to predict the future stock prices for MAANG companies and help the investors make decisions in buying stocks. This model if trained on other company's data can be used to predict their future stocks price as well.

1.8.2 Justification

Model performance based on related metrics is very good and can help investors to make decisions.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

No compliance regulations are applied on using this data.

1.9.2 Justification

As the data used in this project is publicly available, no compliance regulations are applied on using this data.