01

INTERNSHIP REPORT

Pratham Agrawal
prathamagrawal1205@gmail.com

02

Task 1

Report

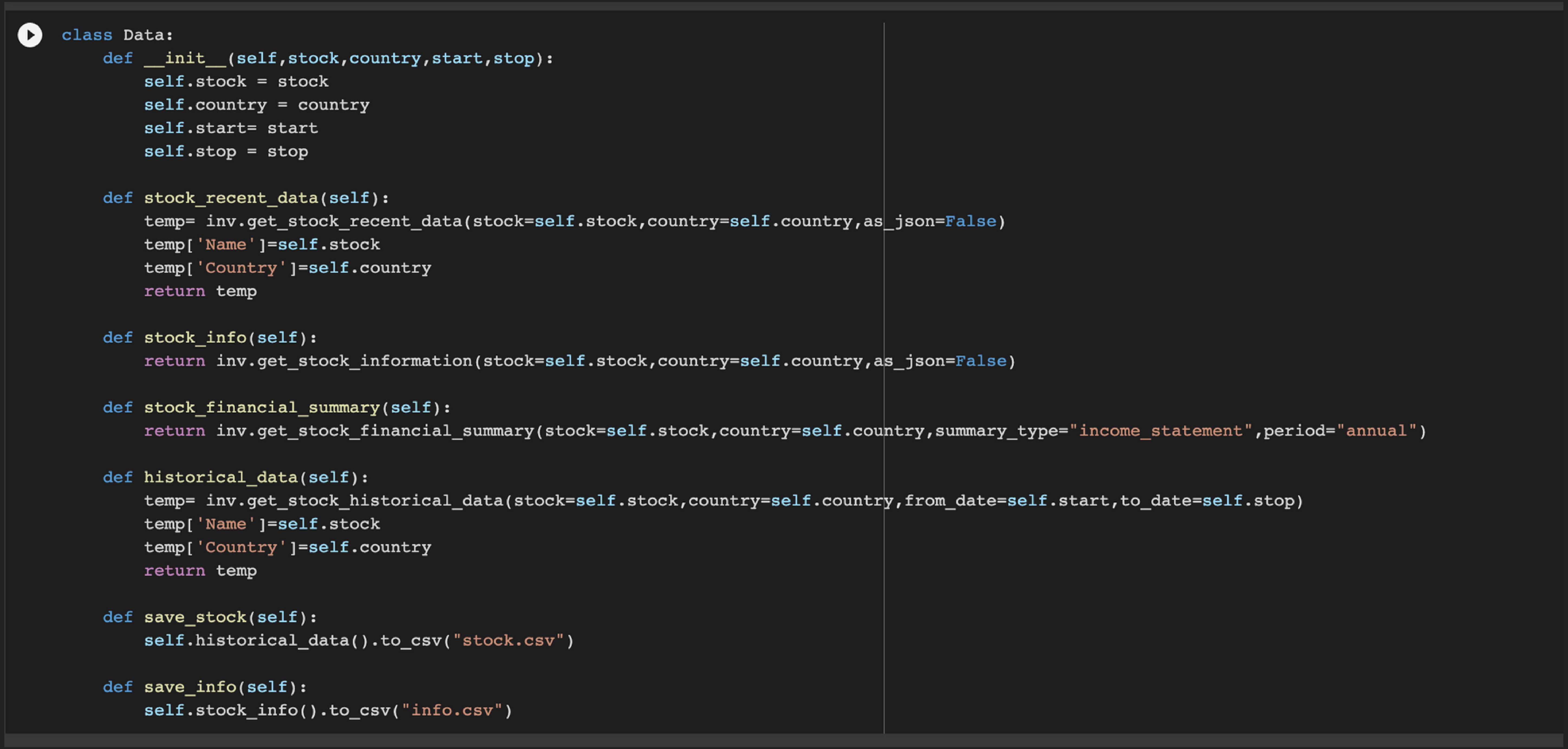
- Use any time series data from investing.com
- Target commodities price like: Oil, Natural Gas, Resin, or Metal Prices.
- I have selected to work with the option 2. LSTM model for daily price prediction.
- How could you improve your results with time and resources.
- Related Thinking and planning in a short report.

03

Work Flow

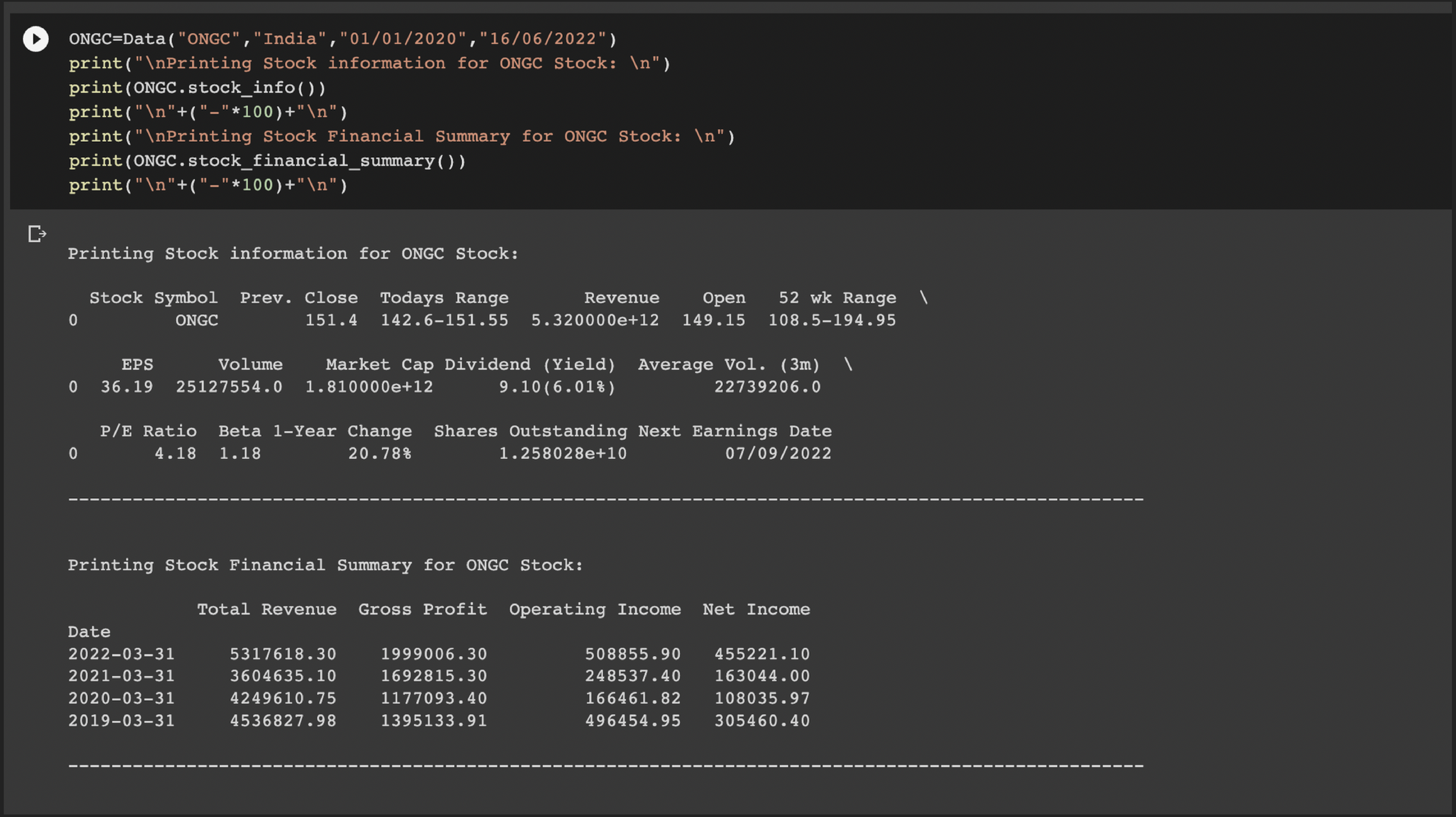
1. Data Extraction using the investpy module which extracts data from investing.com
2. Data such as stock information, open-close, high-low values for a particular day, and historical data is been extracted using **python classes and OOPS.**
3. **5 Stocks** have been considered in this task
4. Data is **preprocessed using some python libraries.**
   a. **dropping of necessary columns,**
   b. **Indexing data according to dates**
   c. **Scaling of data before fitting the data.**
5. Building an **LSTM** model using Keras libraries.
6. Fitting and tuning the model to get the best accuracy.

Data Extraction



```
class Data:
    def __init__(self,stock,country,start,stop):
        self.stock = stock
        self.country = country
        self.start= start
        self.stop = stop

    def stock_recent_data(self):
        temp= inv.get_stock_recent_data(stock=self.stock,country=self.country,as_json=False)
        temp['Name']=self.stock
        temp['Country']=self.country
        return temp

    def stock_info(self):
        return inv.get_stock_information(stock=self.stock,country=self.country,as_json=False)

    def stock_financial_summary(self):
        return inv.get_stock_financial_summary(stock=self.stock,country=self.country,summary_type="income_statement",period="annual")

    def historical_data(self):
        temp= inv.get_stock_historical_data(stock=self.stock,country=self.country,from_date=self.start,to_date=self.stop)
        temp['Name']=self.stock
        temp['Country']=self.country
        return temp

    def save_stock(self):
        self.historical_data().to_csv("stock.csv")

    def save_info(self):
        self.stock_info().to_csv("info.csv")
```

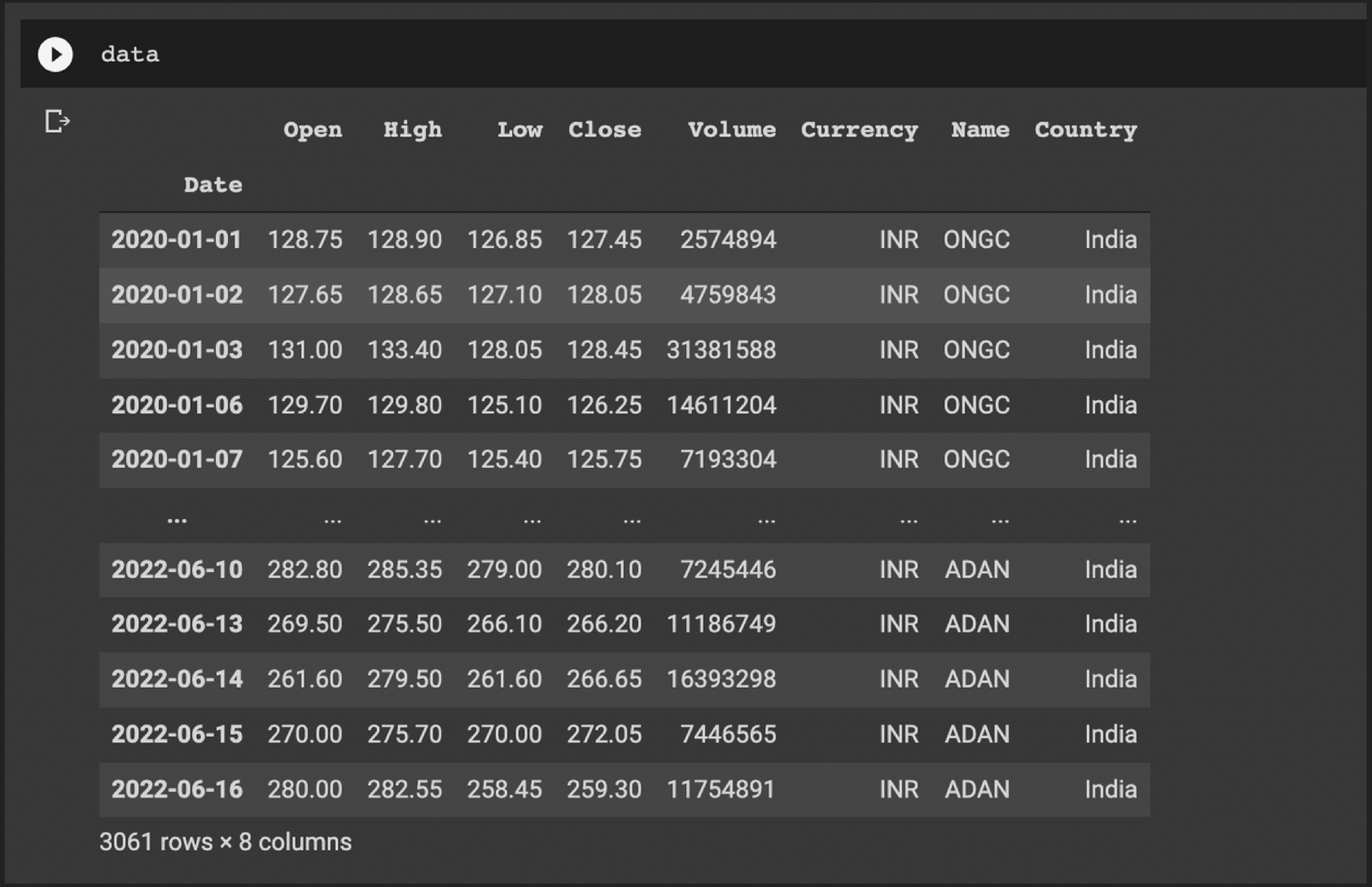
Data Example for a stock



```python
ONGC=Data("ONGC","India","01/01/2020","16/06/2022")
print("\nPrinting Stock information for ONGC Stock: \n")
print(ONGC.stock_info())
print("\n"+("-"*100)+"\n")
print("\nPrinting Stock Financial Summary for ONGC Stock: \n")
print(ONGC.stock_financial_summary())
print("\n"+("-"*100)+"\n")
```

```
Printing Stock information for ONGC Stock: 

  Stock Symbol  Prev. Close  Todays Range       Revenue    Open   52 wk Range  \
0         ONGC        151.4  142.6-151.55  5.320000e+12  149.15  108.5-194.95   

     EPS      Volume    Market Cap Dividend (Yield)  Average Vol. (3m)  \
0  36.19  25127554.0  1.810000e+12      9.10(6.01%)         22739206.0   

   P/E Ratio  Beta 1-Year Change  Shares Outstanding Next Earnings Date  
0       4.18  1.18        20.78%        1.258028e+10          07/09/2022  

----------------------------------------------------------------------------------------------------


Printing Stock Financial Summary for ONGC Stock: 

            Total Revenue  Gross Profit  Operating Income  Net Income
Date                                                                  
2022-03-31     5317618.30    1999006.30          508855.90   455221.10
2021-03-31     3604635.10    1692815.30          248537.40   163044.00
2020-03-31     4249610.75    1177093.40          166461.82   108035.97
2019-03-31     4536827.98    1395133.91          496454.95   305460.40

----------------------------------------------------------------------------------------------------
```

Final Data (for 5 stocks):



```
data
```

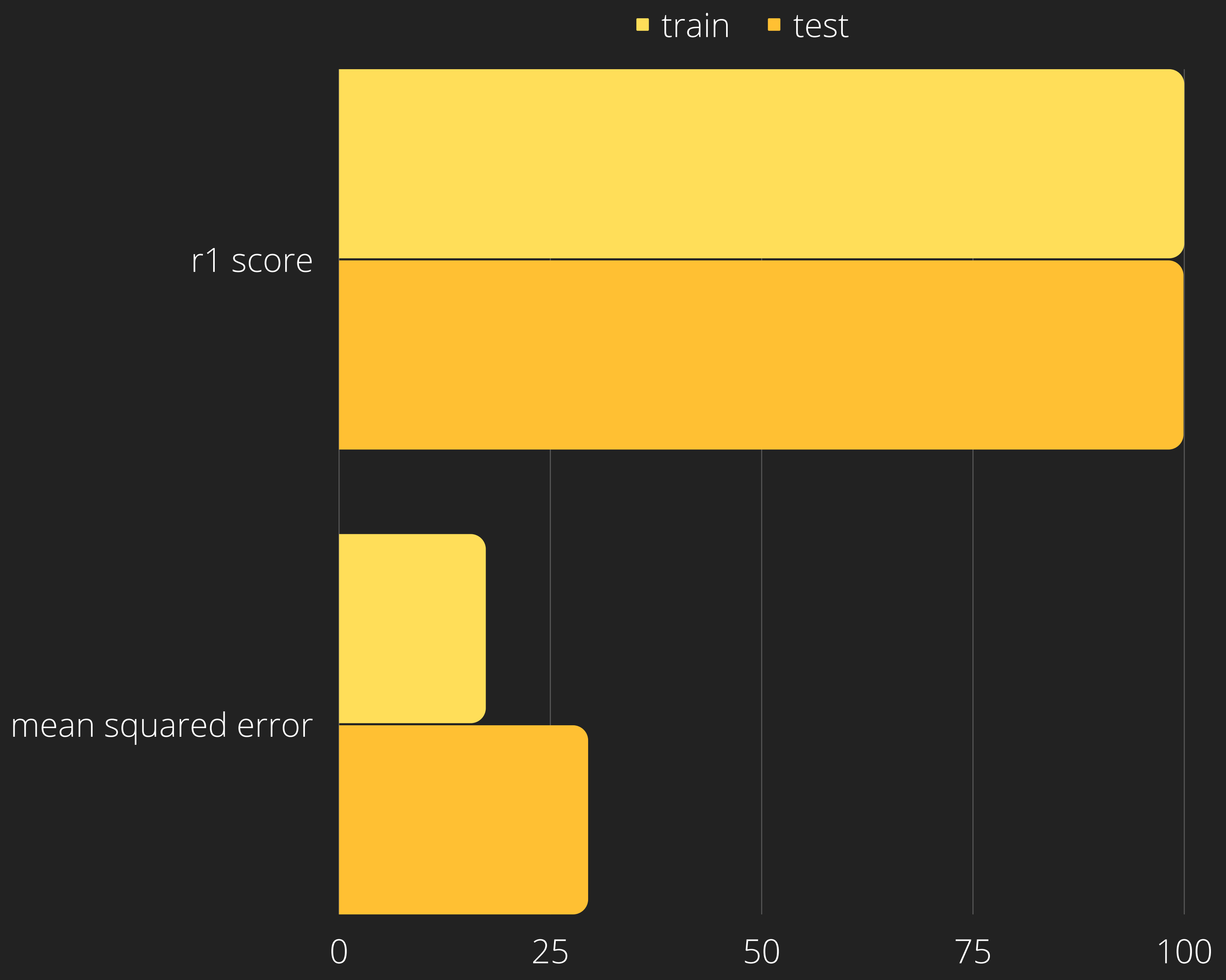
Date	Open	High	Low	Close	Volume	Currency	Name	Country
2020-01-01	128.75	128.90	126.85	127.45	2574894	INR	ONGC	India
2020-01-02	127.65	128.65	127.10	128.05	4759843	INR	ONGC	India
2020-01-03	131.00	133.40	128.05	128.45	31381588	INR	ONGC	India
2020-01-06	129.70	129.80	125.10	126.25	14611204	INR	ONGC	India
2020-01-07	125.60	127.70	125.40	125.75	7193304	INR	ONGC	India
...	...	...	...	...	...	...	...	...
2022-06-10	282.80	285.35	279.00	280.10	7245446	INR	ADAN	India
2022-06-13	269.50	275.50	266.10	266.20	11186749	INR	ADAN	India
2022-06-14	261.60	279.50	261.60	266.65	16393298	INR	ADAN	India
2022-06-15	270.00	275.70	270.00	272.05	7446565	INR	ADAN	India
2022-06-16	280.00	282.55	258.45	259.30	11754891	INR	ADAN	India

3061 rows × 8 columns

07

2.95

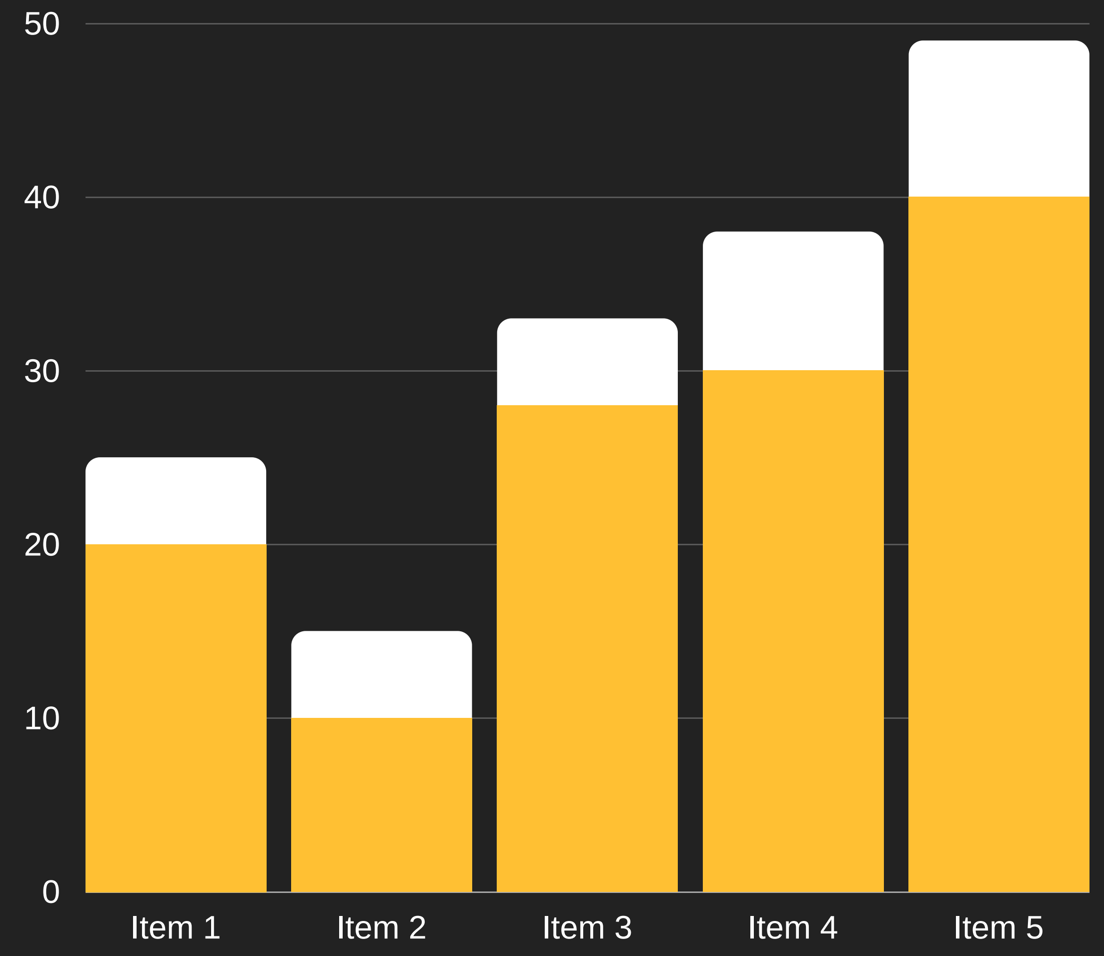
mean squared error after model training



How can results be improved ?

An LSTM model accuracy can be increased by feeding input data which is rich and quite backdated.
Feeding more data, transforming the data and feature selection can be the three ways, the model's accuracy can be improved.

With time, the combination of models or stacking can be a good way to improve the performance of the model.



Links:

1. **Colab File:** https://colab.research.google.com/drive/1qJkaI-zc9m0hIv_T0VG3QgZLdPTqqm-Y#scrollTo=g5-aUvKcuUls
2. **Python file:** https://github.com/prathamagrawal/Taiyo_task/blob/main/Task1/dataextraction.py

Task 2

Report

- Use World Bank Projects Dataset
- Binary Classifier
    - building a binary classifier to predict the probability whether a project will be closed or canceled/distressed

Work Flow

a. Data downloaded from the world bank projects website

b. Preprocessing of data includes removing null values and unnecessary columns

c. Converting all the strings values to numeric values

d. Feature selection using Principle Component Analysis (PCA)

e. Building an ML/ Neural Network model for diagnosis.

f. Training and Testing

Links:

Colab File:
https://colab.research.google.com/drive/1nrhtBIwGBpsM-y5hUFGIgCfFz4odrPXP#scrollTo=xGF0gIFrnuLh