APPLdaily

What it does

APPLdaily is a web-application, aimed to help predict future stock prices based on News, News Sentiment and Historic Trends.

How We built it

We based our prediction on three parameters :

1. News
2. News and its sentiment
3. Historic Trends

News

Prediction using News Trend and sentiment

News and its sentiment

Prediction using News Trend and sentiment

Historic Trends

For the historic trends we used, **Neural Prophet** (Pytorch based version of **Facebook Prophet**). With rigourous tweaking with the parameters we were able to accomplish the foreseeable predictions till **April 22, 2022**.

```
from neuralprophet import NeuralProphet
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from dateutil.relativedelta import relativedelta
from pandas_datareader.data import DataReader
start = datetime.now() - relativedelta(years=12)
end = datetime.now()

data = DataReader('AAPL', data_source='yahoo', start=start, end=end)
df = data.reset_index()

df = df[["Date","Close"]] # select Date and Price
# Rename the features: These names are NEEDED for the model fitting
df = df.rename(columns = {"Date":"ds","Close":"y"}) #renaming the columns of the
dataset

m = NeuralProphet(growth="discontinuous",  # Determine trend types: 'linear',
'discontinuous', 'off'
                  changepoints=None, # list of dates that may include change points
```

```
(None -> automatic )
                n_changepoints=5,
                changepoints_range=0.8,
                trend_reg=0,
                trend_reg_threshold=True,
                yearly_seasonality=True,
                weekly_seasonality=True,
                daily_seasonality=True,
                seasonality_mode="additive",
                seasonality_reg=0,
                n_forecasts=500,
                n_lags=1000,
                num_hidden_layers=4,
                d_hidden=512,     # Dimension of hidden layers of AR-Net
                ar_sparsity=0.01,  # Sparcity in the AR coefficients
                learning_rate=0.001,
                epochs=400,
                loss_func="Huber",
                normalize="auto",  # Type of normalization ('minmax', 'standardize',
'soft', 'off')
                impute_missing=True,
)

metrics = m.fit(df, freq="D")
future = m.make_future_dataframe(df, periods=500, n_historic_predictions=len(df))
forecast = m.predict(future)
```

How you can access it

The application can be accessed from **Heroku Deployment** **ApplDaily** as well as **Google Colab Notebook** **ApplDaily**. Deployment on **Heroku** might sometime crash because of **Heroku's** runtime limitation of providing **~1GB** of RAM, which we recognised to be the threshold of our news sentiment analysis model.

In order for the user to have a smooth experience while using our application, we used **Ngrok** server based on Google Colab Notebook, which exposes the localhost link generated by the Streamlit application to global server's, using which a user can easily access our application.

Heroku Deployment

Our application is successfully deployed on **Heroku** but due to heroku's free-tier limitations, you might not be ablw to successfully execute the predictions, and the app might crash. [Heroku Deployment]

Google Colab Deployment

Running all the cells specified in the notebook, will finally result in a **Ngrok** link, where you can access your application because any further difficulties.

```
    from colab_everything import ColabStreamlit
    ColabStreamlit('/content/app.py') # streamlit app path
```

Example for the same is as follows : [Ngrok Server Link]

Challenges we ran into

- Kaggle dataset error
- LSTM Learning model
- **Deployment**
    - While deploying at Heroku, we recognised the problem of it being a failure

Accomplishments that we are proud of

Successfull deployment of the application at Heroku and at Google Colab using Ngrok Server.

What's next for the project

- A Fully Functional Web App deployed on Heroku or any other Platform-as-a-service (PaaS) client.
- Active Learning model for more accuracte predictions over time.

Cheers to the team

Developer	Name
	Rushank Savant
	Bhavya Goel