**GOLD PRICE PREDICTION USING FBPROPHET**

**INTRODUCTION**

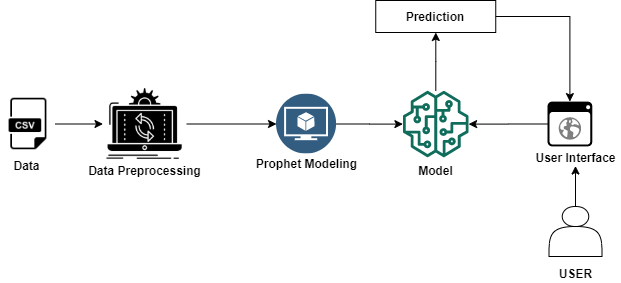
The demand for gold never ends. The gold rate trend shows that gold is one of the best investment plans. So it is wise to predict the trend of the gold rate. Different statistical models can be used for suitable modelling and prediction of data. This paper summarizes the idea of gold rate prediction using FBProphet. It is always visible that the price of gold follows a nonlinear nature. The prediction of prices is commanding for the proper financial and investment plans. The gold rate fluctuation can be modelled as an exponential curve. The dataset collected from the World Gold Council is used and results have shown that the proposed architecture is one of the best financial forecasting methods. Time series analysis is widely using for the applications. But as useful it is also challenging to forecast the correct projections, thus can’t be easily automated because of the underlying assumptions and factors. The analysts who produced accurate forecasts are also rare, and there is a big market available for them because it requires a substantial understanding of statistics and data analysis and has prior experience of producing time series forecasting.

Facebook open-sourced its time-series forecasting tool called Prophet in 2017 which produced accurate forecasts as produced by skilled analysts with a minimum amount of human efforts.

**SOLUTION**

We are creating an application to predict future close price of Gold by using FBprophet model. AS Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. This analysis is the best option for this kind of prediction because we are using the Previous history of Gold Close prices to predict future Close prices.

**ARCHITECTURE:**

****

**PROJECT WORK FLOW:**

* Data Collection.
* Data Pre-processing.
* Model Building.
* Application Building.

**LEARNING OUTCOMES:**

By the end of this project:

* You’ll be able to know the fundamental concepts of time series forecasting.
* Flask Application Development.

**MILSTONES:**

To accomplish this, complete all the milestones & activities listed below.

1. Installation of Pre-requisites.

* Installation of Anaconda IDE / Anaconda Navigator.
* Installation of Python packages.

1. Data Collection.

* Create or Collect the dataset.

1. Data Pre-processing.

* Importing of Libraries.
* Importing of Dataset & Visualisation.

1. Model Building.

* Fitting the prophet library.
* Cross validation of the model.
* Evaluation of the model.
* Save the model.

1. Application Development.

**PRE-REQUISITES:**

**Milestone 1: Installation of Pre-requisites**

To complete the project successfully, you need to install following software & packages:

**Activity 1: Install Anaconda IDE / Anaconda Navigator.**

* In order develop a solution to this problem statement, we need an environment to write and test the code.
* We use Anaconda IDE (Integrated Developing Environment).
* Refer to the below link to download & install Anaconda Navigator.

**Link:** [**https://www.youtube.com/watch?v=5mDYijMfSzs**](https://www.youtube.com/watch?v=5mDYijMfSzs)

**Activity 2:** **Installation of Python Packages**

* Follow the below steps to install the required libraries.
* Open Anaconda Navigator as administrator.
* Type “pip install pystan” and press enter.
* Type “conda install -c conda-forge fbprophet” and press enter.

The above steps allow to install the required packages.

With thiswe are done with the completion of milestone 1.

* To know more about the prophet library go through the reference links.

**Link 1**:[**https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/**](https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/)

**Link2:**[**https://towardsdatascience.com/a-quick-start-of-time-series-forecasting-with-a-practical-example-using-fb-prophet-31c4447a2274**](https://towardsdatascience.com/a-quick-start-of-time-series-forecasting-with-a-practical-example-using-fb-prophet-31c4447a2274)

**Milestone 2: Data Collection**

Now, the milestone-2 is all about creation or collection of dataset

Browse the below link to download the dataset.

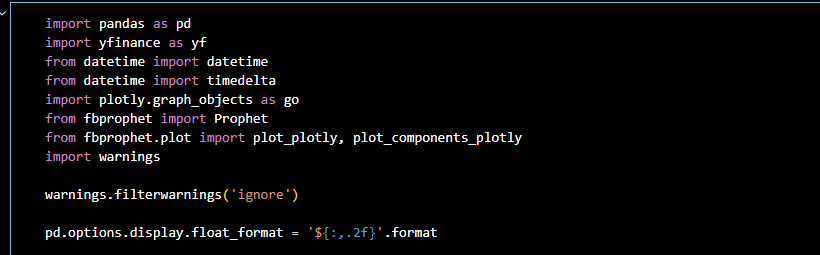
**Link:** [GC\_F Gold Futures — Stock Price and Discussion | Stocktwits](https://stocktwits.com/symbol/GC_F)

**Milestone 3: Data Pre-processing**

In this milestone, you need to complete all the below activities to build the model.

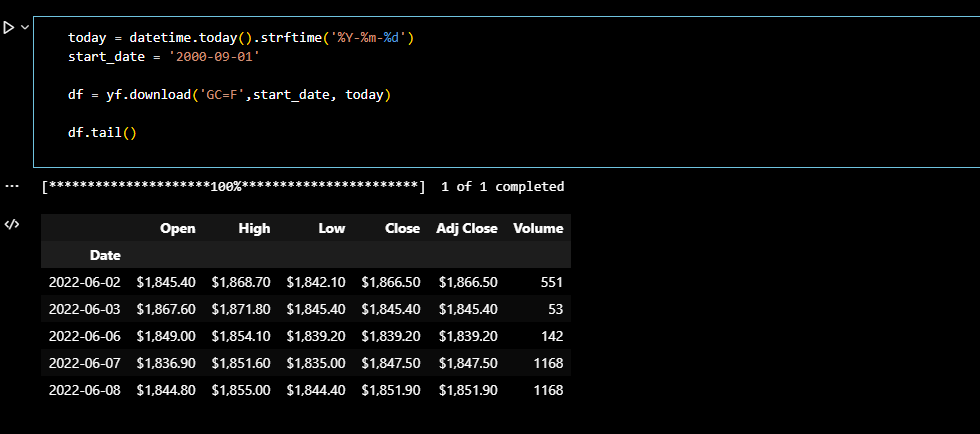
**Activity 1: Import Libraries**

Import the below essential libraries for data pre-processing and FBProphet forecasting, etc…

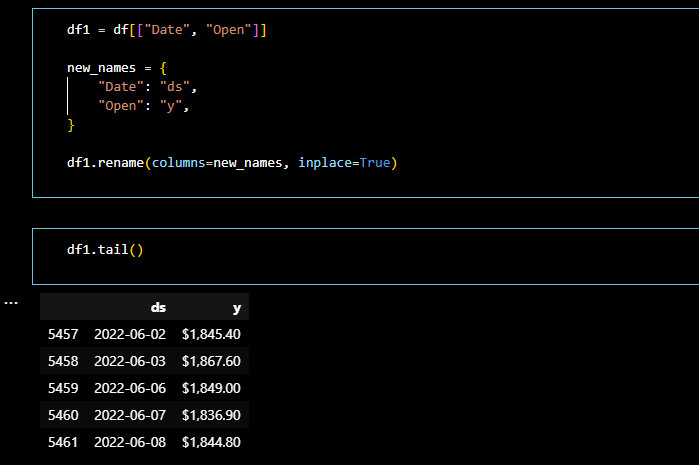


**Activity 2: Fetch Data & Visualisation**

* Fetching the data from Yahoo Finance using the below command & List the first five row of the data using head function.



* Selecting & Renaming columns accordingly to the prophet library integration for building the model.



* Plotting Open price, using plotly library.



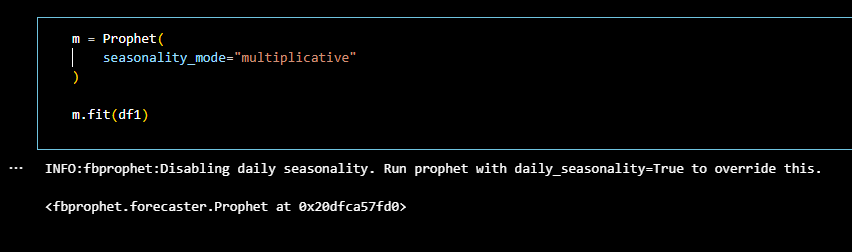
* With this , you have successfully completed the milestone-3

**Milestone 4: Model Building**

In this milestone, you will build the model using the prophet library.

**Activity 1: Fitting the prophet library**

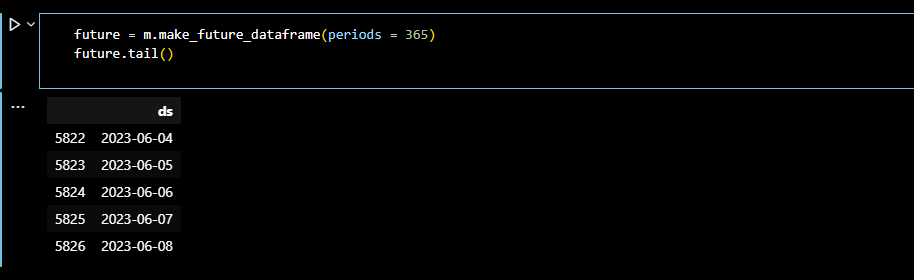
* Import the prophet library, create the instance and fit it to the dataset using the below commands.

­­­­

**Note:** It will take few minutes to fit the model.

* **Making Future Predictions**

The next step is to prepare our model to make future predictions. This is achieved using the Prophet.make\_future\_dataframe method and passing the number of days we’d like to predict in the future. We use the periods attribute to specify this. This also include the historical dates. We’ll use these historical dates to compare the predictions with the actual values in the ds column.

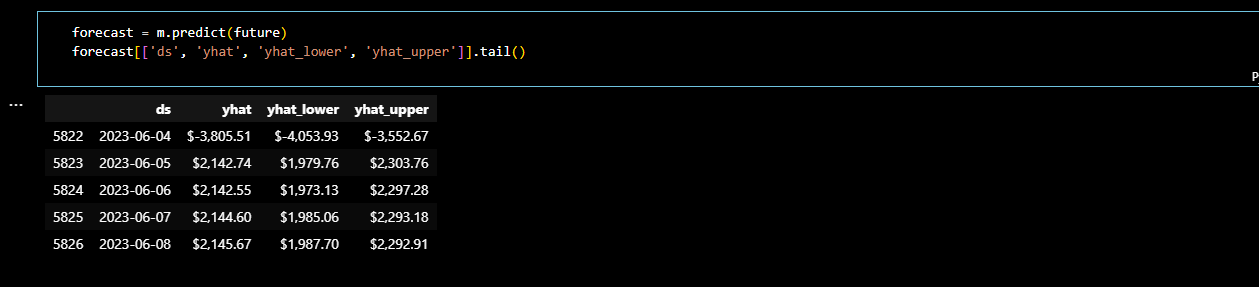


* **Obtaining the Forecasts**

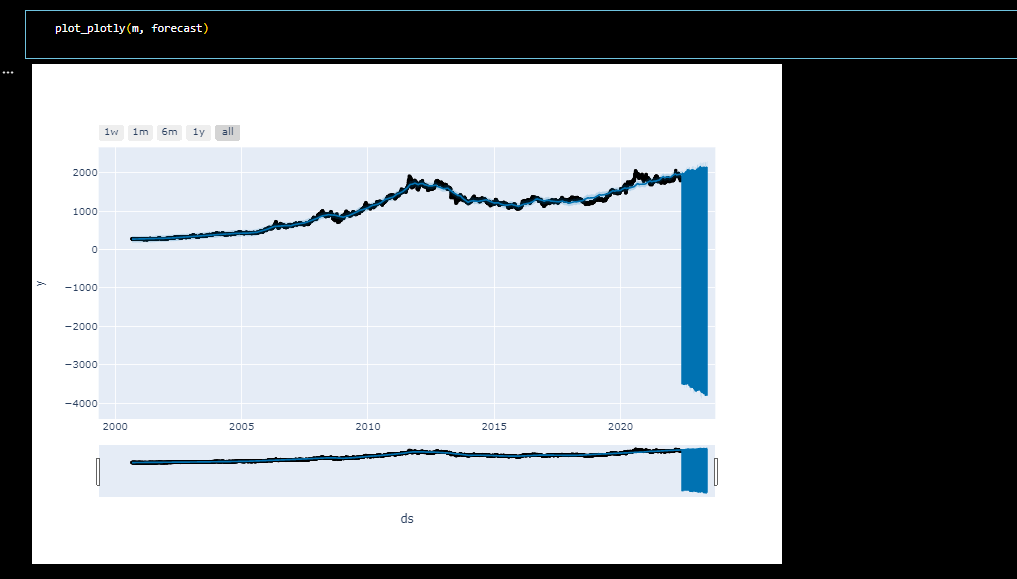
We use the predict method to make future predictions. This will generate a dataframe with an **yhat** column that will contain the predictions.

If we check the head for our forecast dataframe we’ll notice that it has very many columns. However, we are mainly interested in **ds, yhat, yhat\_lower and yhat\_upper**. **yhat** is our predicted forecast, **yhat\_lower** is the lower bound for our predictions and **yhat\_upper** is the upper bound for our predictions.

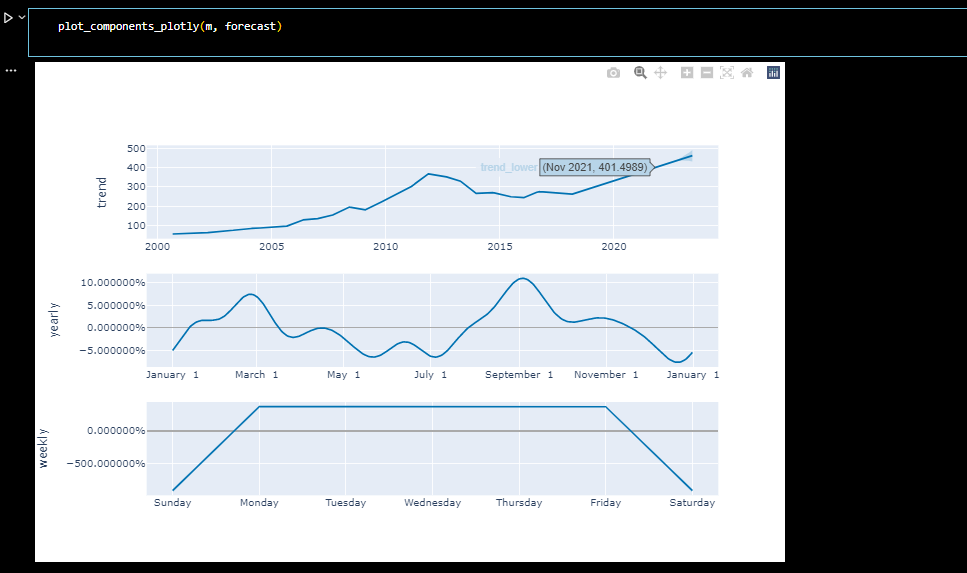
* Get the summary of the forecast using the below command.



* Plotting the forecast



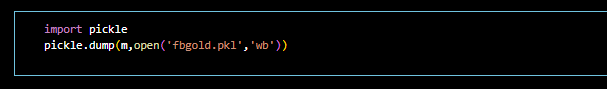
* Now, let’s visualize each component.



**Activity 3: Save the model.**

This is the final activity of this milestone, here you will be saving the model to integrate to the web application.

* Follow the commands to save your model.

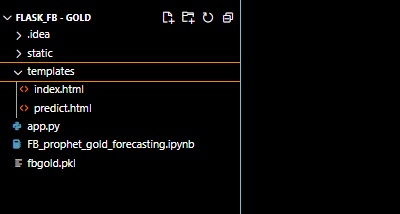


* With this, you have successfully completed the milestone-4.

**Milestone 5: Application Building**

In this milestone, you will be developing an html page and integrate the saved model.

* In order to process proceed with this milestone, arrange all your project files in the below format.



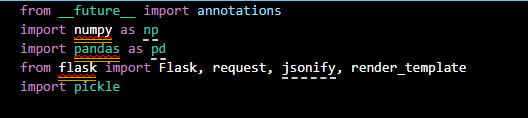
* All the above files will be used to develop a flask application.
* In the static folder we will be storing all the flask background images.
* In the templates, you will be storing all the rendering files and html pages.
* The gold\_fb.pkl is the saved model file
* In the app.py, you will be writing a code for flask integration

**Activity 1: Importing of Libraries and routing the html pages**

We will be using python for server-side scripting. Let’s see step by step process for writing backend code.

Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of current module (\_\_name\_\_).

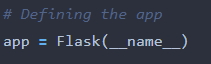
* Import the following libraries

****

* Rendering to html page

Here we will be using declared constructor to route to the html page which we have created earlier.

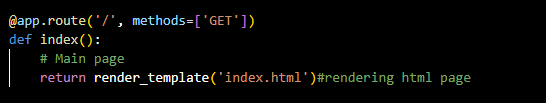
In the above example, ‘/’ URL is bound with home.html function. Hence, when the home page of web server is opened in browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using **POST** Method.



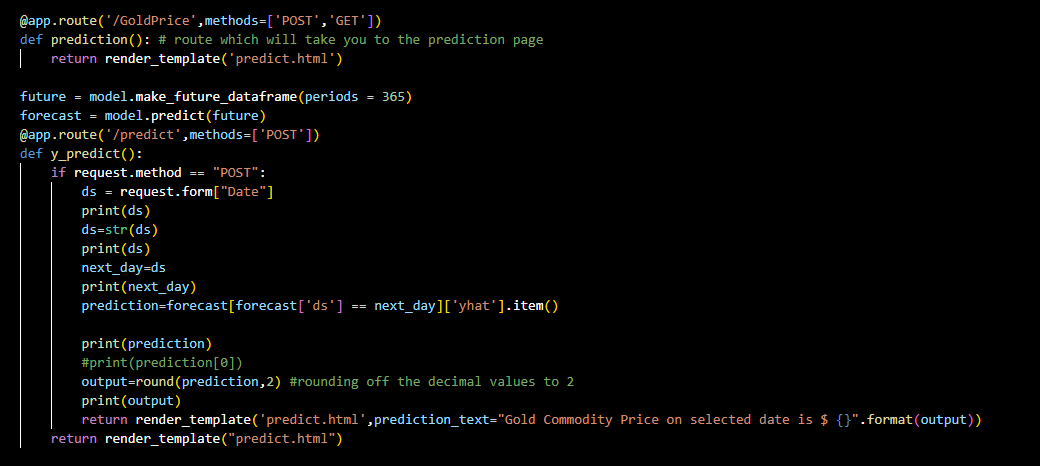
* Loading the save model using pickle library.



* Rendering of home page html.

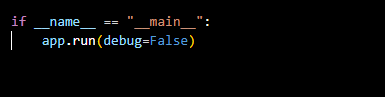
****

* Routing the prediction to the home page.



* Calling of Main Function

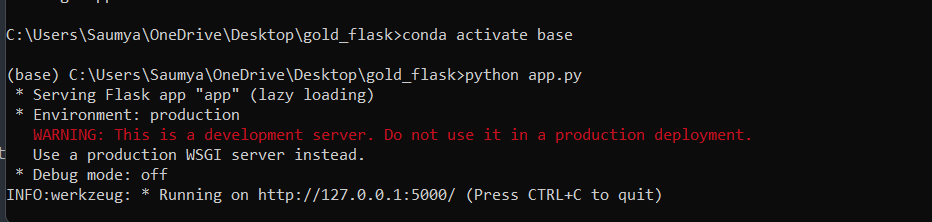
This is used to run the application in local host.



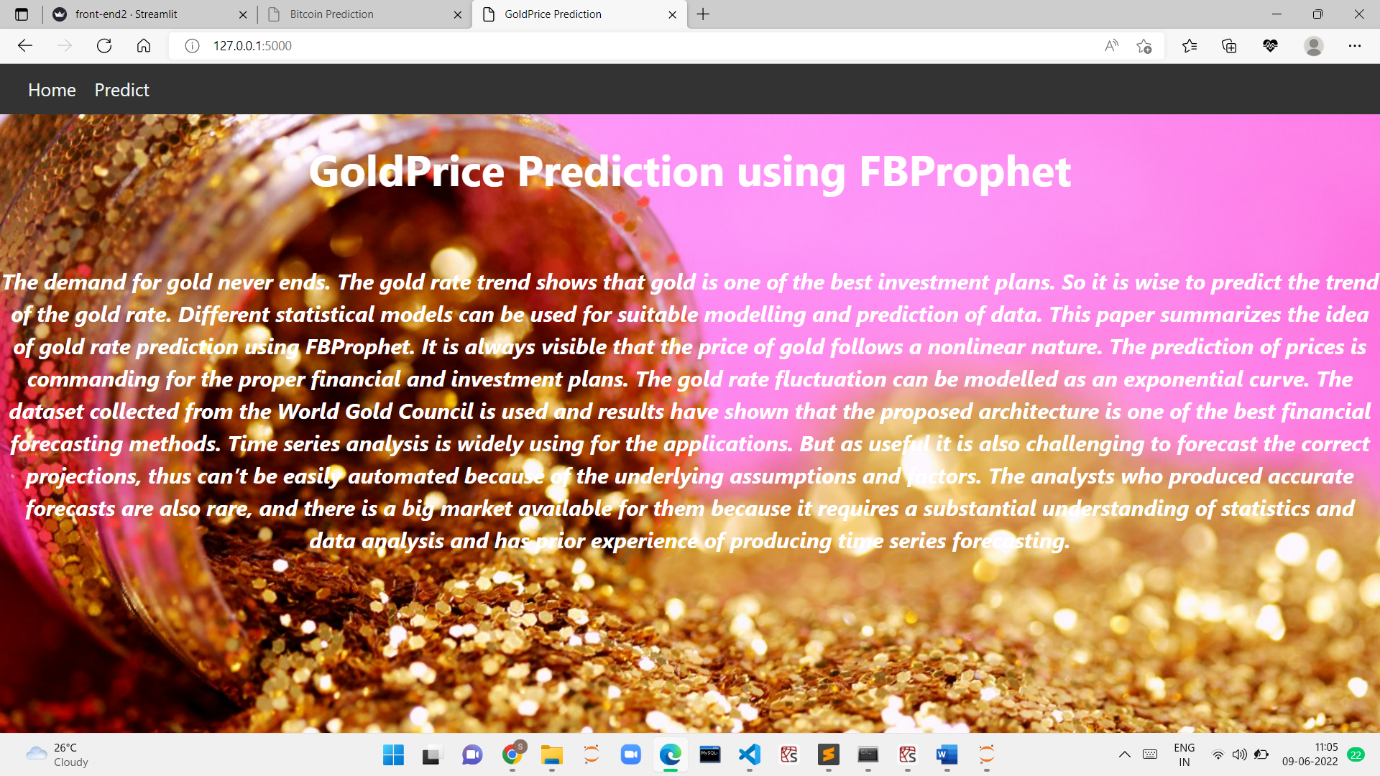
Note: Ensure that, you have written the code properly without indentation errors.

**Activity 2: Running of flask Application**

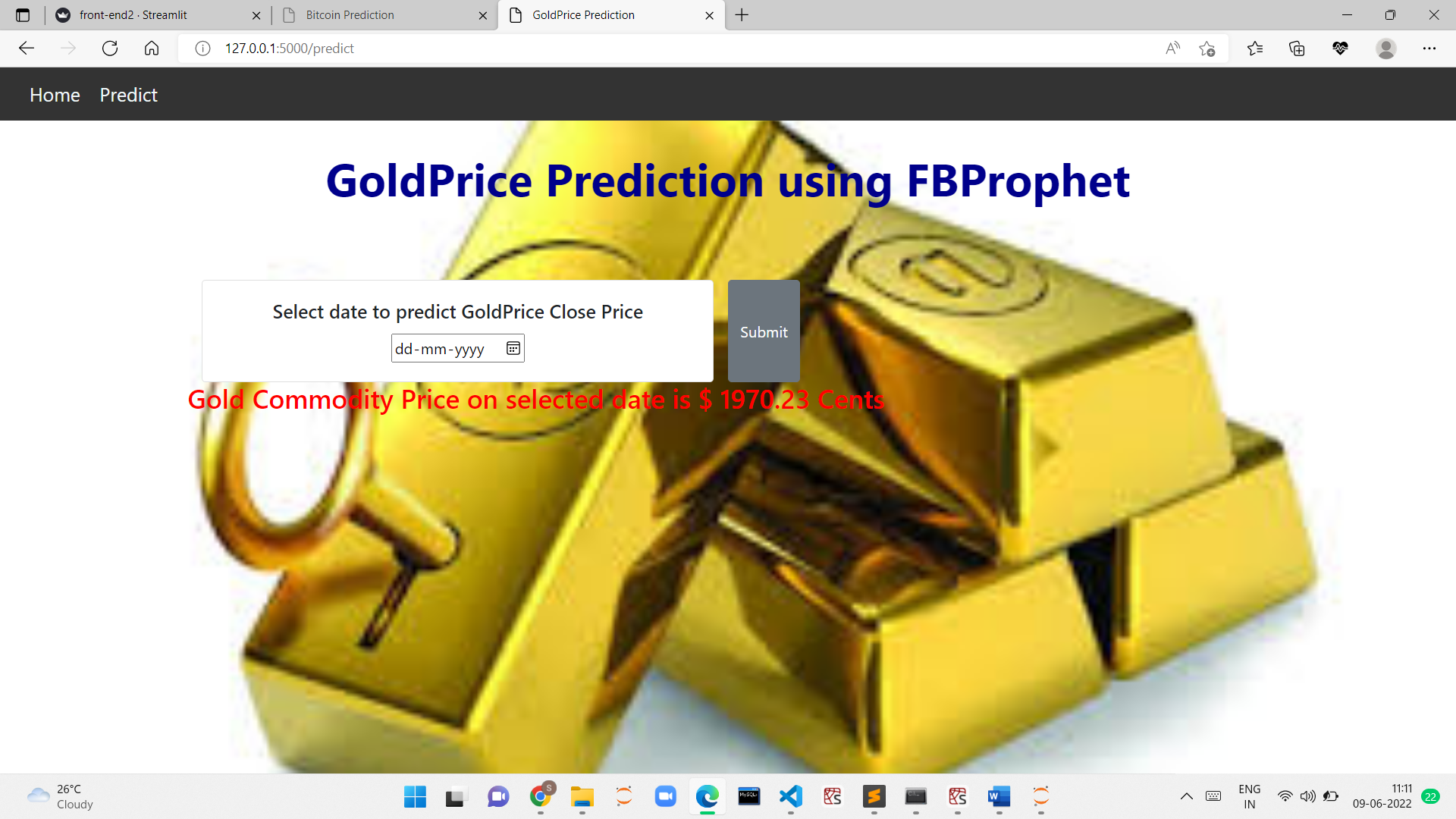
* Open anaconda prompt from start menu.
* Navigate to the folder where your app.py resides.
* Now type “python app.py” command.
* It will show the local host where your app is running on [**http://127.0.0.1.5000/**](http://127.0.0.1.5000/)
* Copy that local host URL and open that URL in browser. It does navigate you to the where you can view your web page.



* Your UI will look like



* Select the date you would like to predict and click on submit.
* The output prediction will be like



* Check the predictions for the different inputs.
* With this you have successfully completed the project.