Indiana University Bloomington

## Time Series Analysis

## Final Project Part 3

### **Project title:** UFO sightings analysis

**Project team:** Anas Omary (individual project)

# Part 0. Introduction

Unidentified Flying Object (UFO) as a topic has gained traction in recent years, and has been an intriguing “spooky” topic for years that it became part of our culture with many Hollywood movies and series about UFOs, although it is part of our culture, the subject of UFOs has been stigmatized and laughed at in science communities and scientists have approached this subject with caution in fear of being called pseudo scientists or being described as tin foil hat conspiracy theorists.

**Objectives**

For this project I will follow a scientific approach in analyzing UFO related data to try to draw subjective conclusions on wither UFOs are real or not and where those sightings happen, For the pure fun of it!

**Project description**

**Usefulness**

In this project I will use shiny library to create an interactive dashboard that shows the history and progress of UFO sightings worldwide with multiple criteria filters, I haven’t found many visualizations on the web that tackles all aspects of UFO sightings and haven’t found any interactive dashboards, so I will make the dashboard interactive to give the ability for users to view data from different angles, stakeholders for this project are any person interested in the UFO subject.

**Dataset**

I will use UFO sightings dataset from data.world ((<https://data.world/timothyrenner/ufo-sightings>))

Origin: The National UFO Research Center ([NUFORC](http://www.nuforc.org/)) This dataset contains the UFO sightings collected worldwide from 1969 till 2022 for the purpose of investigating and analyzing those sightings, as far as data reliability the dataset seems reliable and a result of actual sightings reported by actual people across the world, the dataset includes attributes like observation date and time, location, duration, UFO shape and other attributes in the raw form as it is recorded on the NUFORC site which makes this dataset a good candidate for time series analysis.

**Communication and sharing**

This is an individual project where I will be working solo to create the UFO sightings interactive dashboard

**Github**

All project files can be found on github in the following repository:

https://github.com/ANASOMARY/TimeSeriesAnalysis\_FinalProject

# Part 1. Time Series Application Architecture

### Summary

The UFO analysis application is hosted entirely on google cloud platform, I created a new project named “TimeSeriesAnalysis” on GCP and under the project I created a storage bucket and a virtual machine that runs Linux ubuntu and did the needed firewall configurations for VM to allow connections to the deployed application, then installed python and the needed libraries like Dash, Tensorflow …etc. LSTM model and dataset are stored on a cloud bucket along with a backup of the web application app.py, and the application is deployed on a virtual machine on GCP that calls the dataset file which is a CSV dataset and the LSTM model which is used to predict time series, from the storage bucket.

The app.py application running on the virtual machine is a web application developed using python and dash, it connects directly to the CSV dataset that is located on a GCP storage bucket and is configured to be publicly available, the app.py calls a saved LSTM model which is the outcome of part 2 of this project where 3 models were trained and tested then finally I decided to use the LSTM model because it scored the highest, the LSTM model is saved on the same storage bucket (my\_lstm\_model.h5) , the connection security was not a big concern since the data is already publicly available and does not contain any sensitive information.

The user can reach the application by calling the URL (<http://34.125.118.242:8080/>)

The front-end layout is created using dash and HTML, I used “dash\_html\_components” library to create HTML elements, and used dash\_core\_components to create the UI controls, the UI consists of the following elements:

1. H1, H2, H3 headers: those are created using dash\_html\_components
2. Date range picker: this is a date range picker where the user will choose the start and end date for the period that will be analysed.
3. Drop down lists: there will be 3 drop-down lists where the user will choose the country, state, and city in which UFO sightings happened, whatever the user chooses will be reflected in the visualizations
4. Button: the submit button is created using HTML and will be used to fire the event which will redraw the visualizations based on the user choices.
5. 2 graph elements: I used dash to create two graph elements, one will show a bar-chart and the other one will show a time series visualization with prediction.

### Web app architecture

**Requesting data:** <https://storage.googleapis.com/tsa_final_project/tsa_final_project_folder/ufo_dataset.csv>

**Requesting LSTM model:**

gs://tsa\_final\_project/tsa\_final\_project\_folder/my\_lstm\_model.h

GCP Storage bucket

Application server (GCP Linux VM)

<http://34.125.118.242:8080/>

Requesting UFO data and LSTM model

UFO data, LSTM model

Ufo\_dataset.csv

My\_lstm\_model.h5

User

**App.py**

Python

Dash

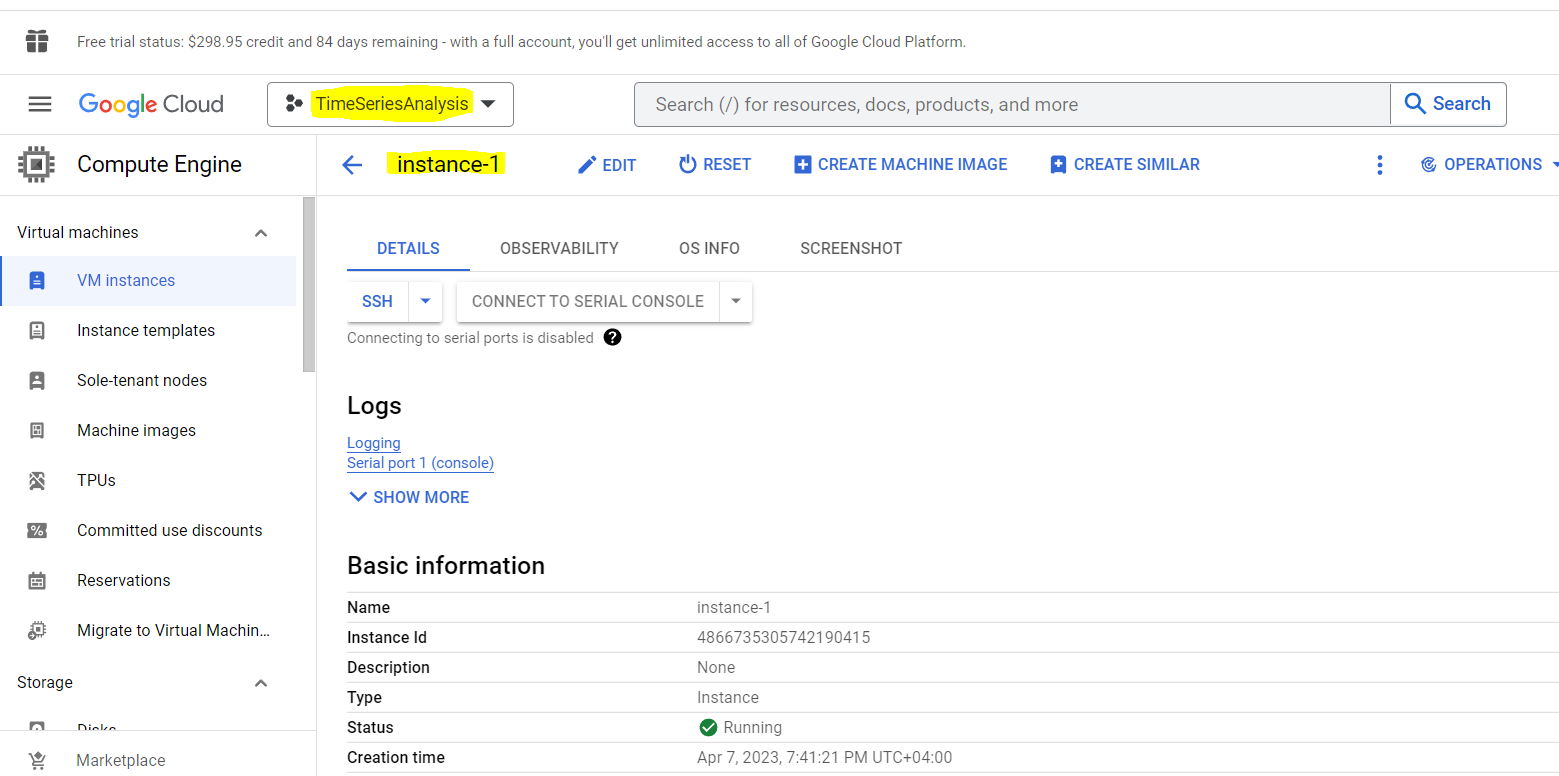
HTML

### Environment preparation

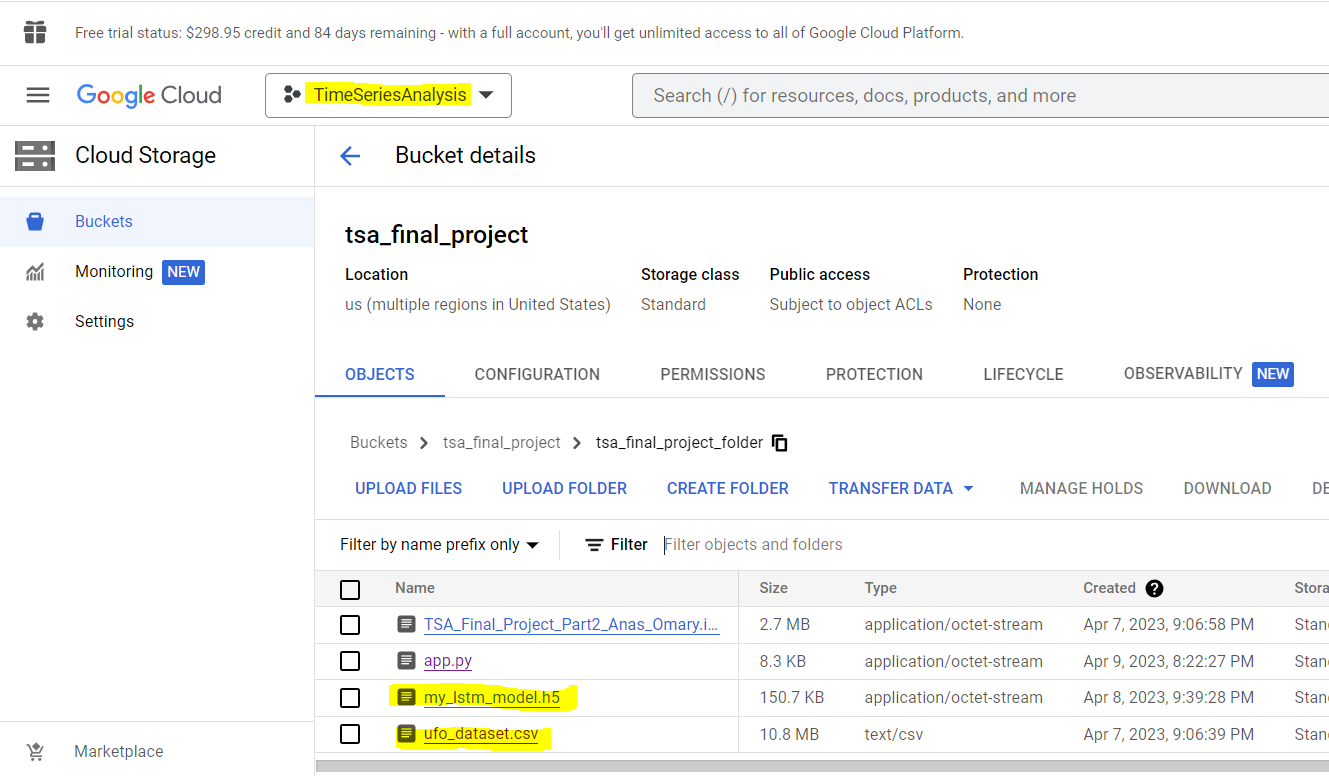
In this step I prepared the environment for the application deployment:

**On GCP I created a new project and under the project I created a storage bucket and a VM**:

* VM: I created a VM “instance-1” and changed the configuration of the VM to allow HTTP traffic and applied firewall rule to allow connections on port 8080:



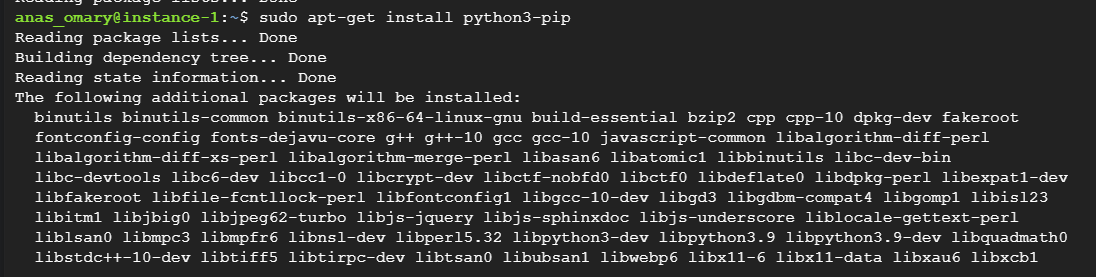
* I then created a storage bucket “tsa\_final\_project” and then created a folder inside the bucket “tsa\_final\_project\_folder” and uploaded the dataset “ufo\_dataset.csv” and the LSTM model “my\_lstm\_model.h5”:



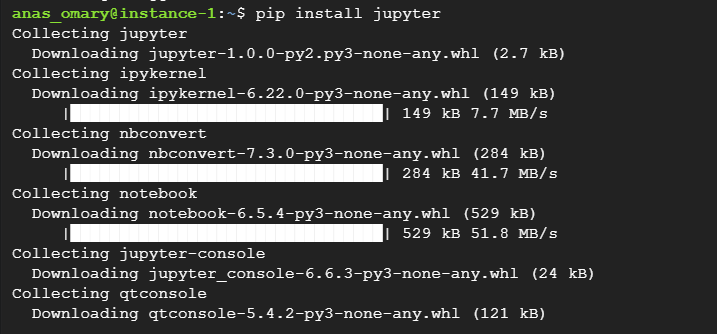
### App deployment

**I prepared the VM and deployed app.py:**

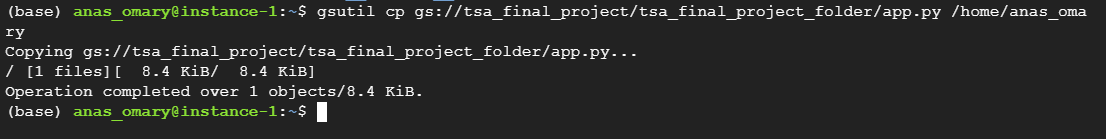
* Installed python:



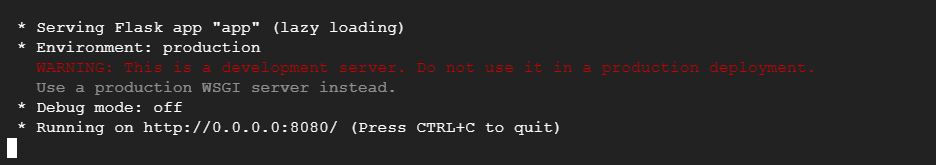
* Installed Jupyter:



* Copied the application file “app.py” from the bucket into the VM:

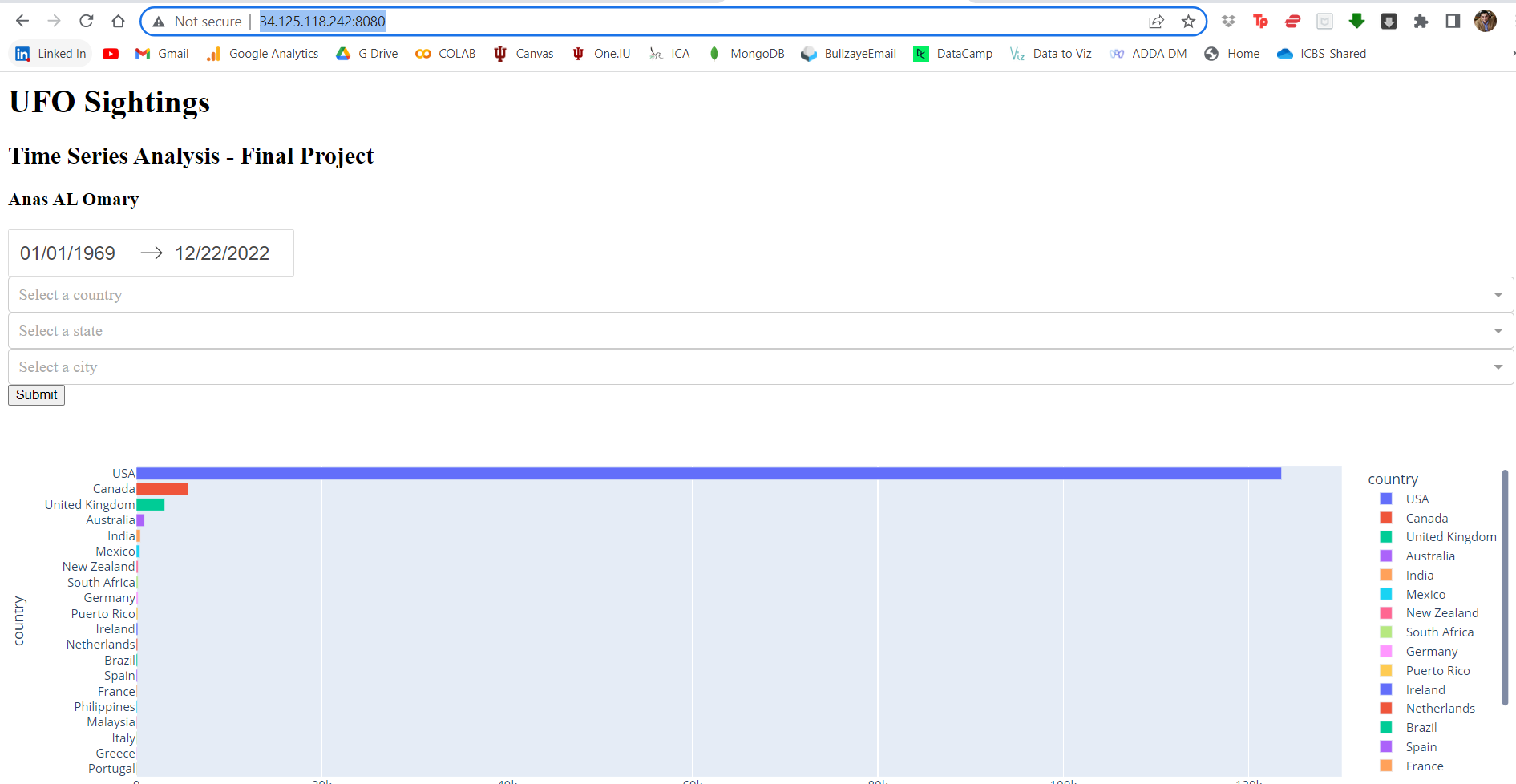


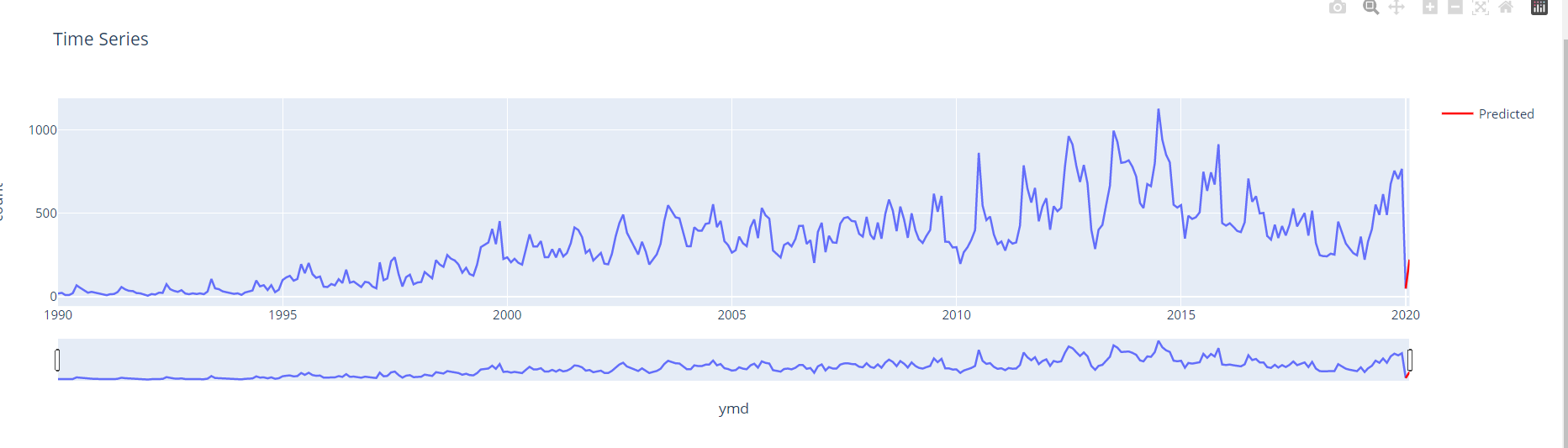
* Deployed the app.py using the command $ python app.py, I have configured app.py to run in production mode on port 8080



### App Testing

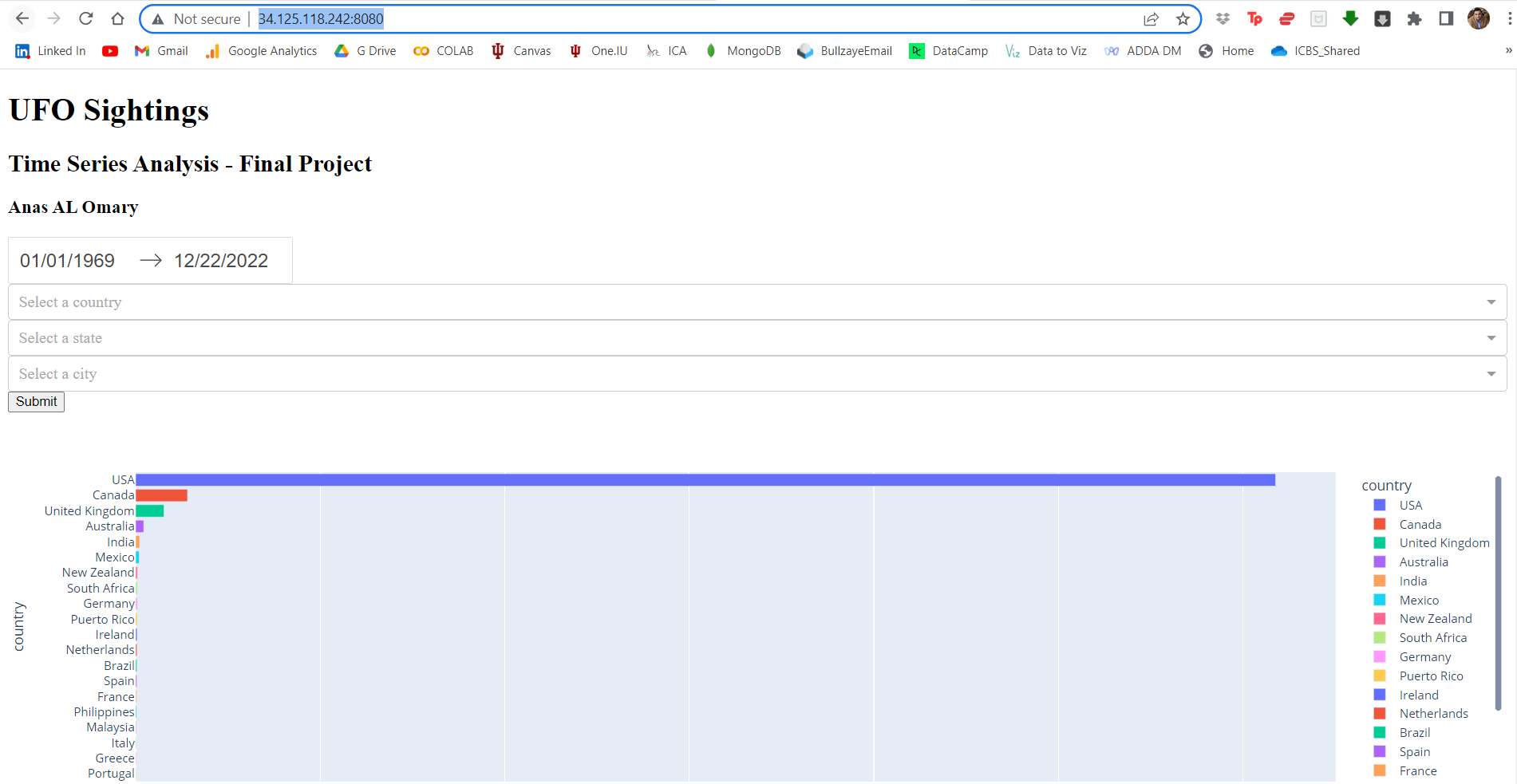
**I tested the app:** after preparing the environment and deploying app.py, I tested the application to make sure it is accessible and runs fine:

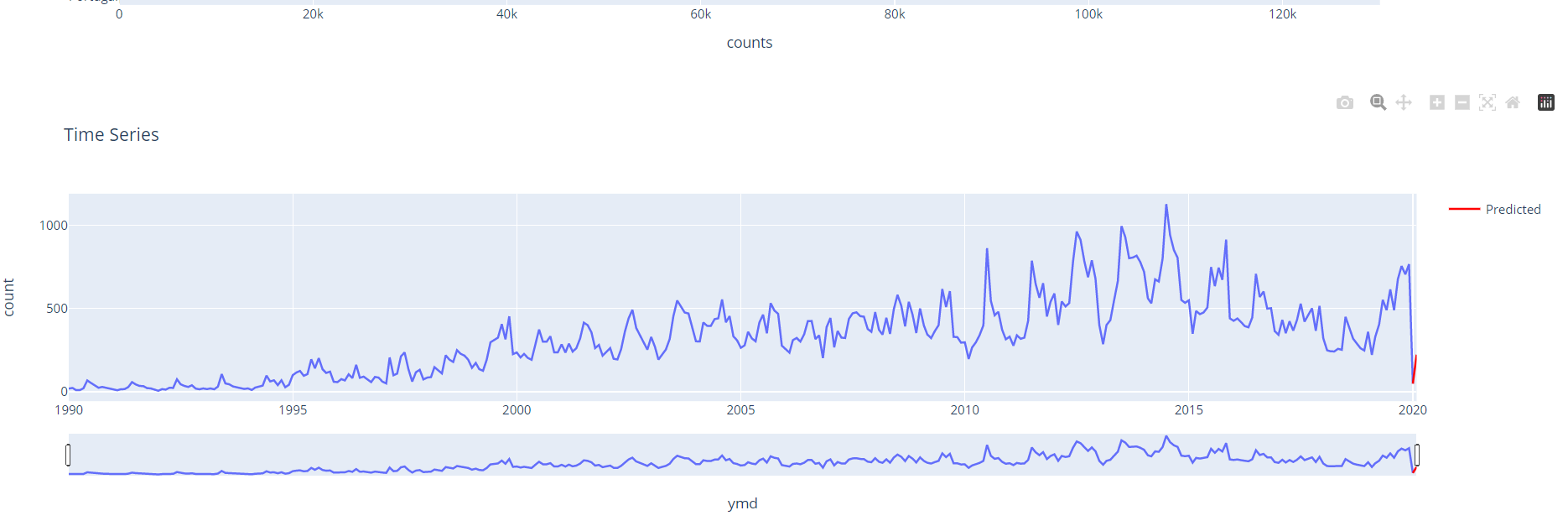




# Part 2. App Layout

I have already created the web app and deployed it on GCP following is the application:



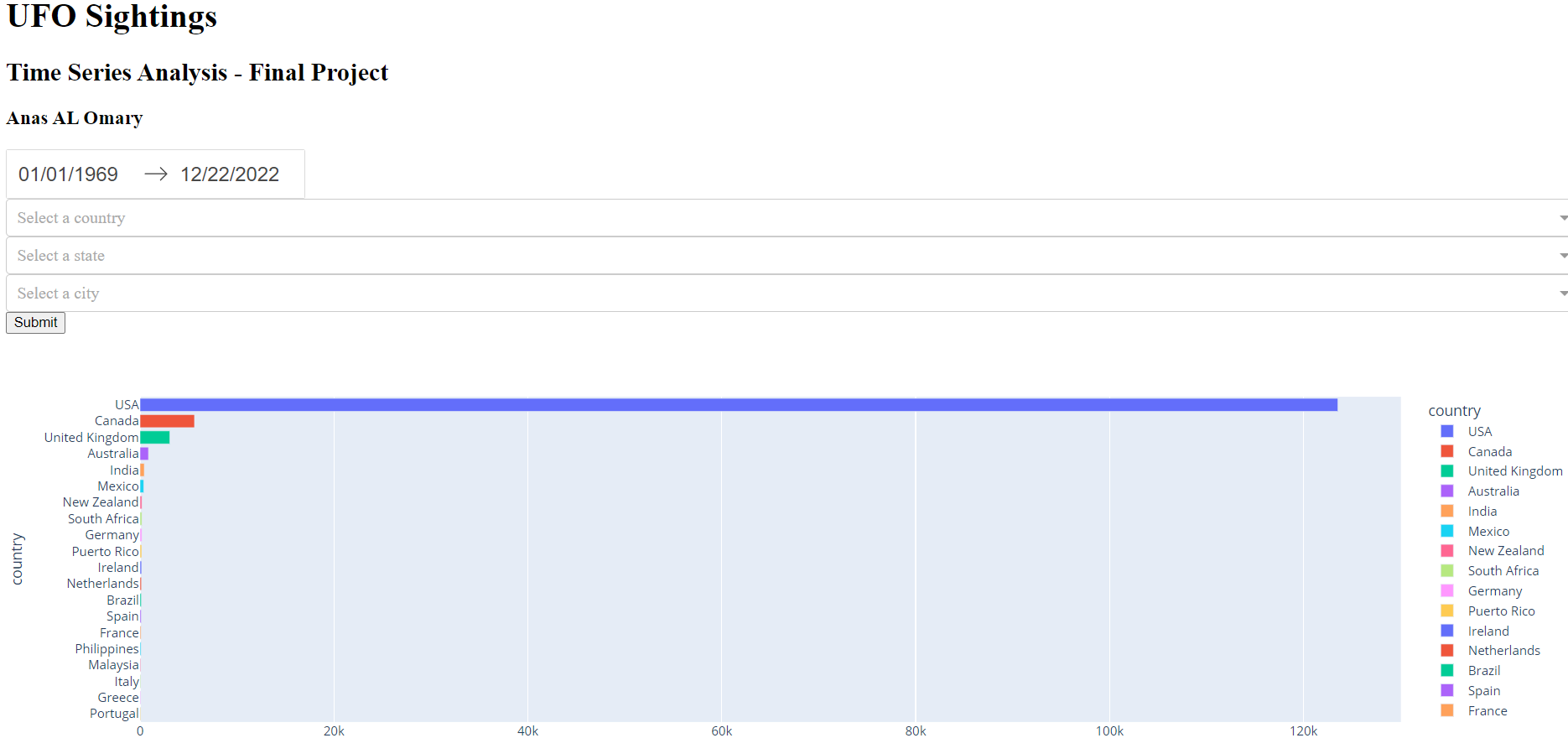


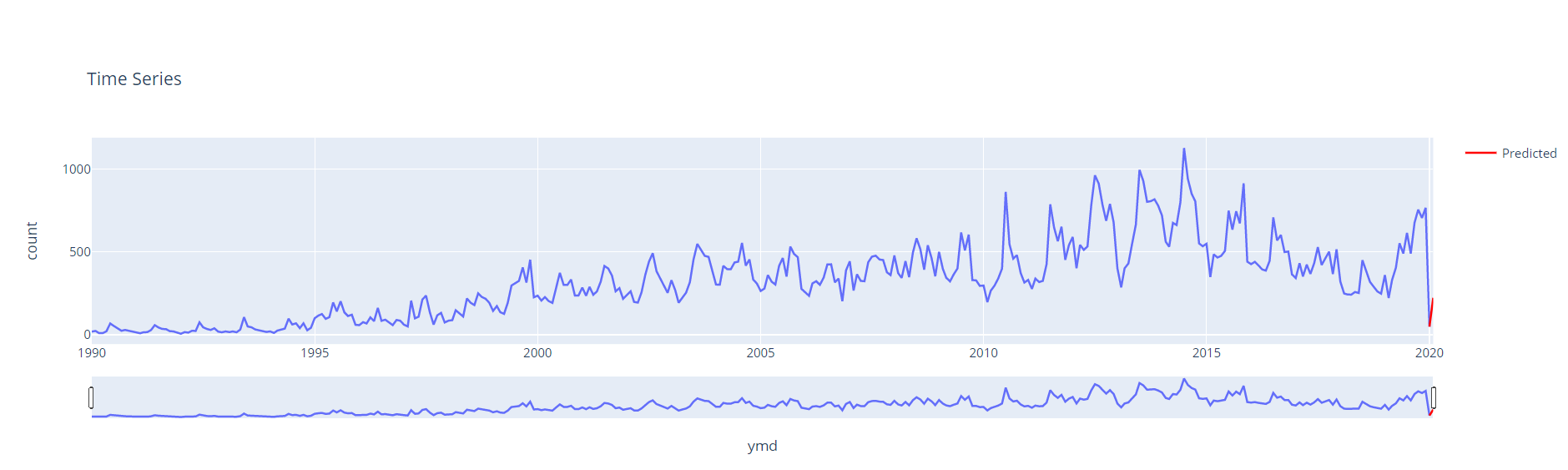
The user calls the application using the URL (http://34.125.118.242:8080/ ) then the web server serves the application app.py, the app consists of one page which has a header, date range picker, three drop-down lists and two graph elements:



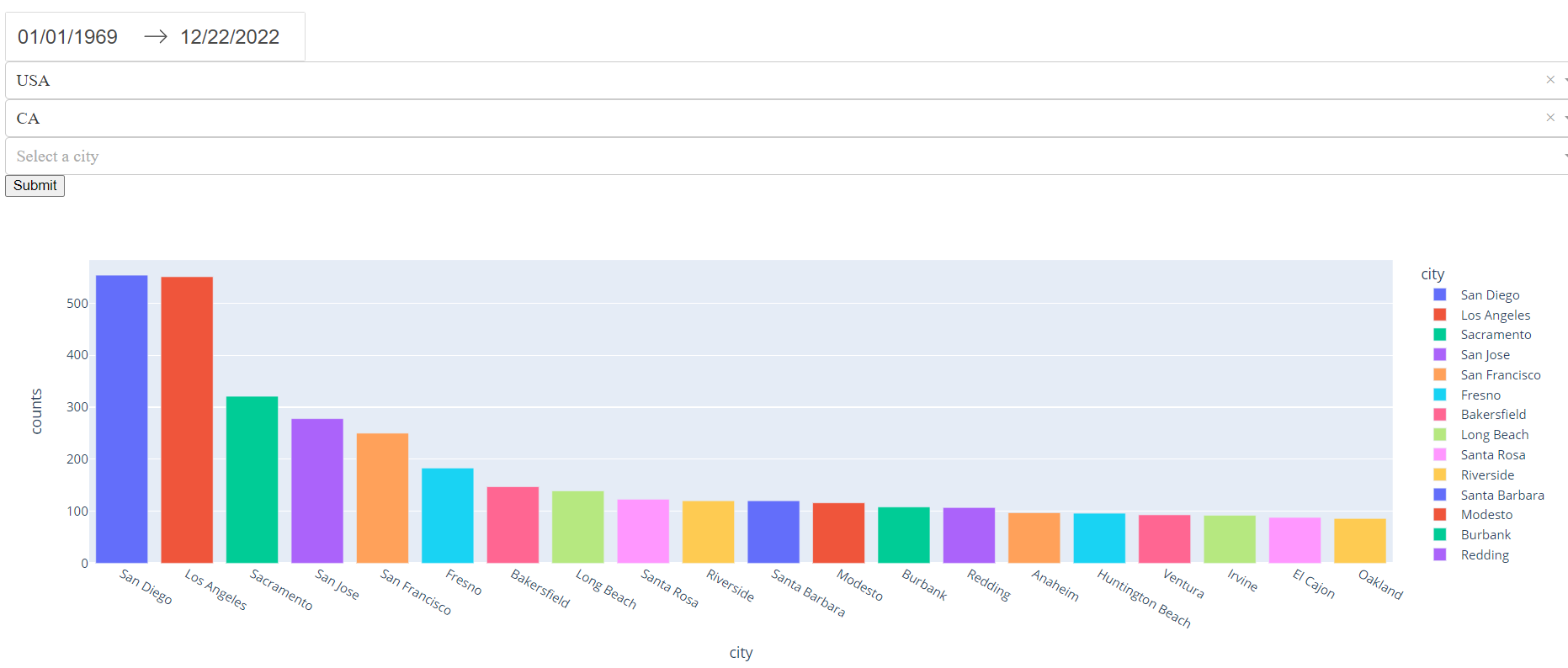
When the app first loads, it shows the date range picker with date period between January first 1969 and December 22 2022, then it loads data into the three drop down lists.

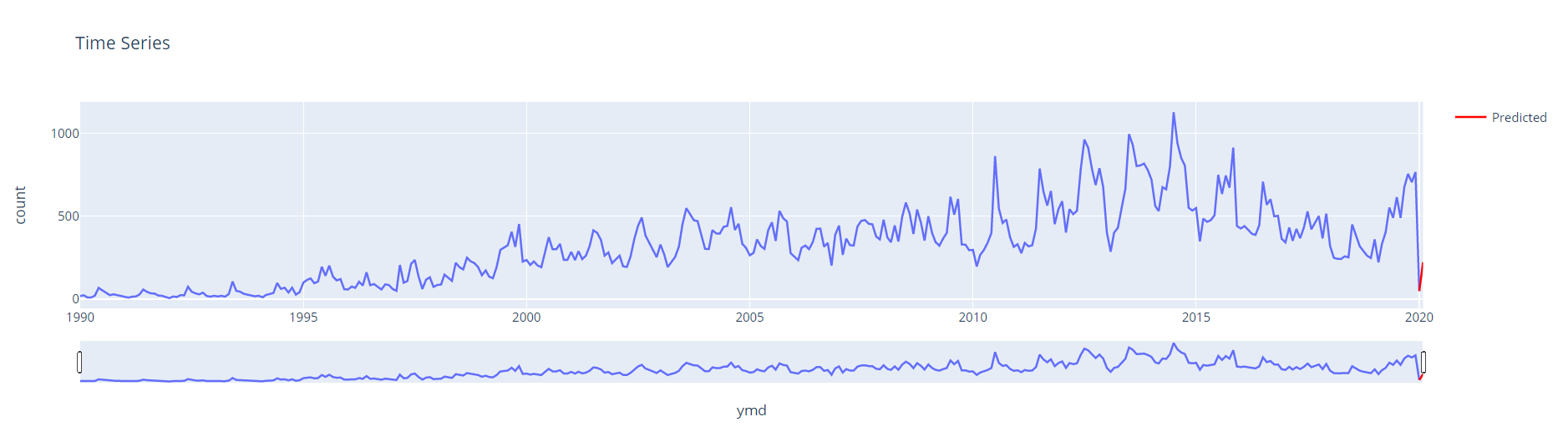
By default, the app loads the entire time series data from January first 1969 till December 22 2022 showing observations worldwide.





The user can filter by a specific time period, country, state and city, then click on submit, this will apply the filter on the bar-chart accordingly, then it will apply the filter on the time series visualization and recalculate predictions.





Following code shows how I am loading the LSTM model which I saved in part 2 of the project, then I reshape the data to be suitable for the model then I generate predictions and include it in the time series visualization:



# Part 3. Team Work

This is an individual project, and while I was designing the application I decided to go ahead and build the app locally on my machine, then I ended up deploying it on GCP, in the next phase, I will test with different design themes and further test the application and validate the predictions.