

# Model Deployment on Heroku Cloud Platform

Name: Motamen MohammedAhmed

Batch Code: LISUM03

Submission Date: 19/09/2021

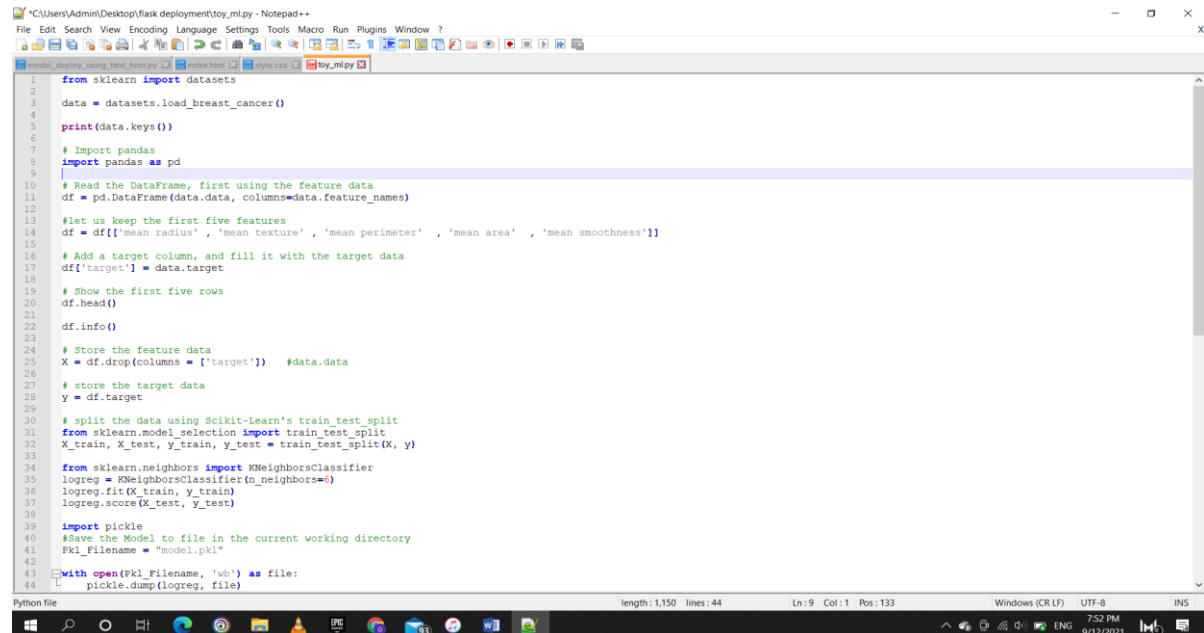
Submitted to: Data Glacier

## Introduction:

These are the steps required to deploy a machine learning model using Flask on a cloud platform (Heroku) instead of a local machine (your PC). Each step is provided with a screenshot.

## Step 1:

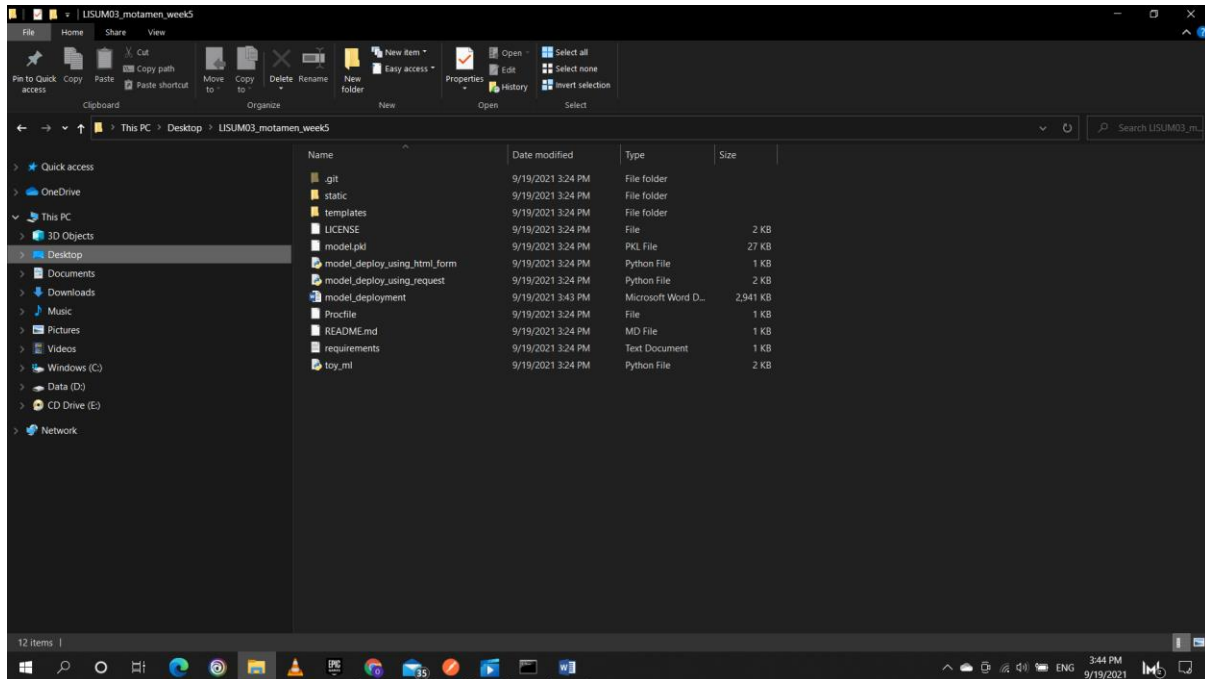
Find a toy dataset and build your model. We chose the breast cancer Wisconsin dataset.



```
1 from sklearn import datasets
2
3 data = datasets.load_breast_cancer()
4
5 print(data.keys())
6
7 # Import pandas
8 import pandas as pd
9
10 # Read the DataFrame, first using the feature data
11 df = pd.DataFrame(data.data, columns=data.feature_names)
12
13 #let us keep the first five features
14 df = df[['mean radius', 'mean texture', 'mean perimeter', 'mean area', 'mean smoothness']]
15
16 # Add a target column, and fill it with the target data
17 df['target'] = data.target
18
19 # Show the first five rows
20 df.head()
21
22 df.info()
23
24 # Store the feature data
25 X = df.drop(columns = ['target']) #data.data
26
27 # store the target data
28 y = df.target
29
30 # split the data using Scikit-Learn's train_test_split
31 from sklearn.model_selection import train_test_split
32 X_train, X_test, y_train, y_test = train_test_split(X, y)
33
34 from sklearn.neighbors import KNeighborsClassifier
35 logreg = KNeighborsClassifier(n_neighbors=4)
36 logreg.fit(X_train, y_train)
37 logreg.score(X_test, y_test)
38
39 import pickle
40 #Save the Model to file in the current working directory
41 Pkl_Filename = "model.pkl"
42
43 with open(Pkl_Filename, 'wb') as file:
44     pickle.dump(logreg, file)
```

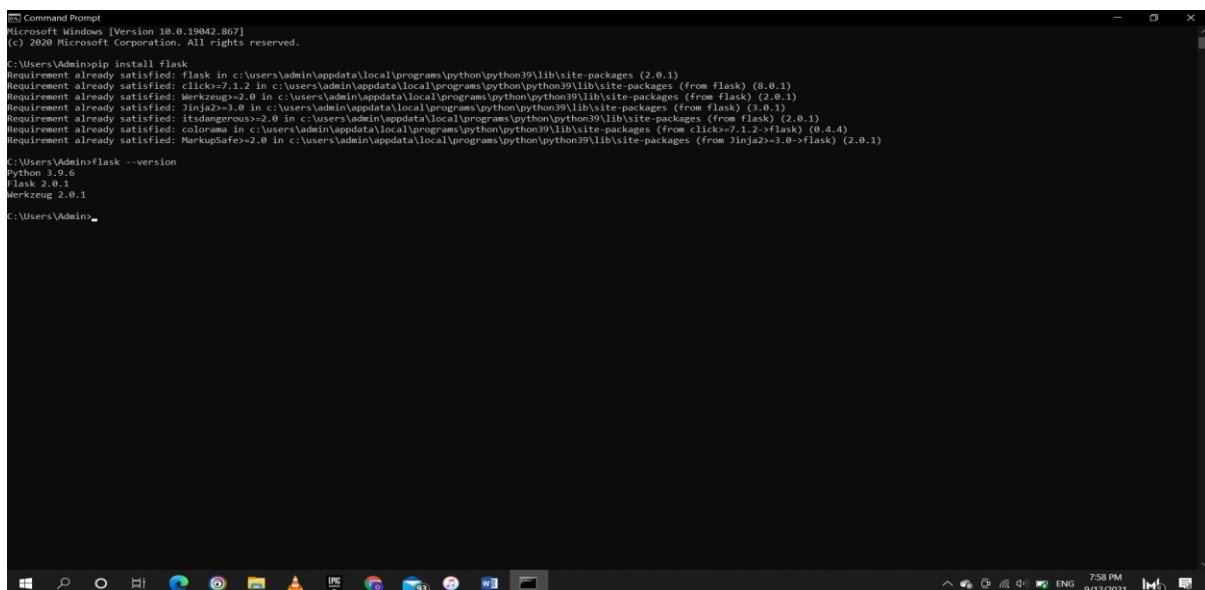
## Step 2:

Save the model using pickle module. The model is saved as model.pkl file.



## Step 3:

Install Flask and make sure it is installed successfully. Flask is the core for this deployment to work properly.



## Step 4:

Create a web app using Flask. The features are collected from the HTML request. Also, create a web app using the HTML form. We need both of them.

```
C:\Users\Admin\Desktop\flask deployment\model_deploy.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
model_deploy_using_html_form.py index.html style.css try_ml.py model_deploy.py
1 from flask import Flask, jsonify, request
2 import pickle
3 import pandas as pd
4
5 app = Flask(__name__)
6
7
8 @app.route('/', methods = ['GET', 'POST'])
9 def home():
10     if request.method == 'GET':
11         data = "Hello World"
12         return jsonify({'data':data})
13
14
15 @app.route('/predict/')
16 def tumor_predict():
17     model = pickle.load(open('model.pkl', 'rb'))
18     mean_radius = request.args.get('mean_radius')
19     mean_texture = request.args.get('mean_texture')
20     mean_perimeter = request.args.get('mean_perimeter')
21     mean_area = request.args.get('mean_area')
22     mean_smoothness = request.args.get('mean_smoothness')
23
24
25     test_df = pd.DataFrame({'mean_radius':[mean_radius], 'mean_texture':[mean_texture],
26                             'mean_perimeter':[mean_perimeter], 'mean_area':[mean_area], 'mean_smoothness':[mean_smoothness]})
27
28     tumor_pred = model.predict(test_df)
29     if tumor_pred == 1:
30         return jsonify({'result': 'tumor is benign, it will not spread'})
31     else:
32         return jsonify({'result': 'tumor is malignant, it will spread'})
33
34
35 if __name__ == '__main__':
36     app.run(port = 5000, debug = True)
```

Python file length: 1,137 lines: 36 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS 8:12 PM 9/12/2021

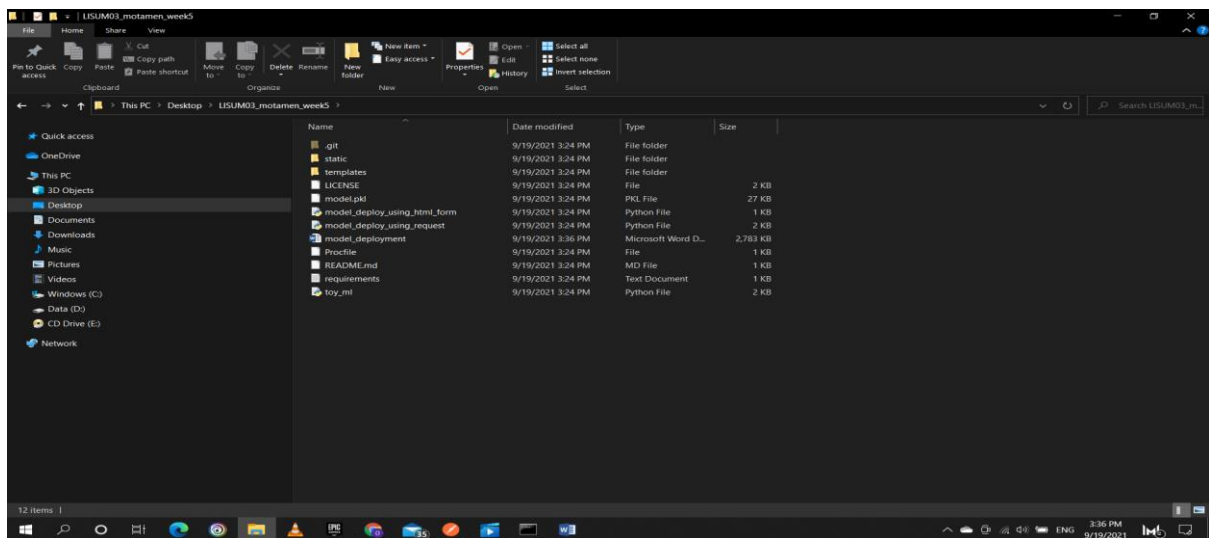
```
C:\Users\Admin\Desktop\UISUM03_motamen_week5\model_deploy_using_html_form.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
model_deploy_using_html_form.py index.html style.css try_ml.py new 1 model_deploy_using_request.py index.html new 2 Procfile model_deploy_using_request.py model_deploy_using_html_form.py
1 from flask import Flask, render_template, request
2 import pickle
3 import numpy as np
4
5 app = Flask(__name__)
6
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12
13 @app.route('/predict', methods = ['POST'])
14 def tumor_predict():
15
16     model = pickle.load(open('model.pkl', 'rb'))
17     int_features = [float(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19
20     tumor_pred = model.predict(final_features)
21     if tumor_pred == 1:
22         return render_template('index.html', prediction_text = "tumor is benign, it will not spread")
23     else:
24         return render_template('index.html', prediction_text = "tumor is malignant, it will spread")
25
26
27 if __name__ == '__main__':
28     app.run(debug = True)
```

Python file length: 782 lines: 28 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS 4:24 PM 9/19/2021

## Step 5:

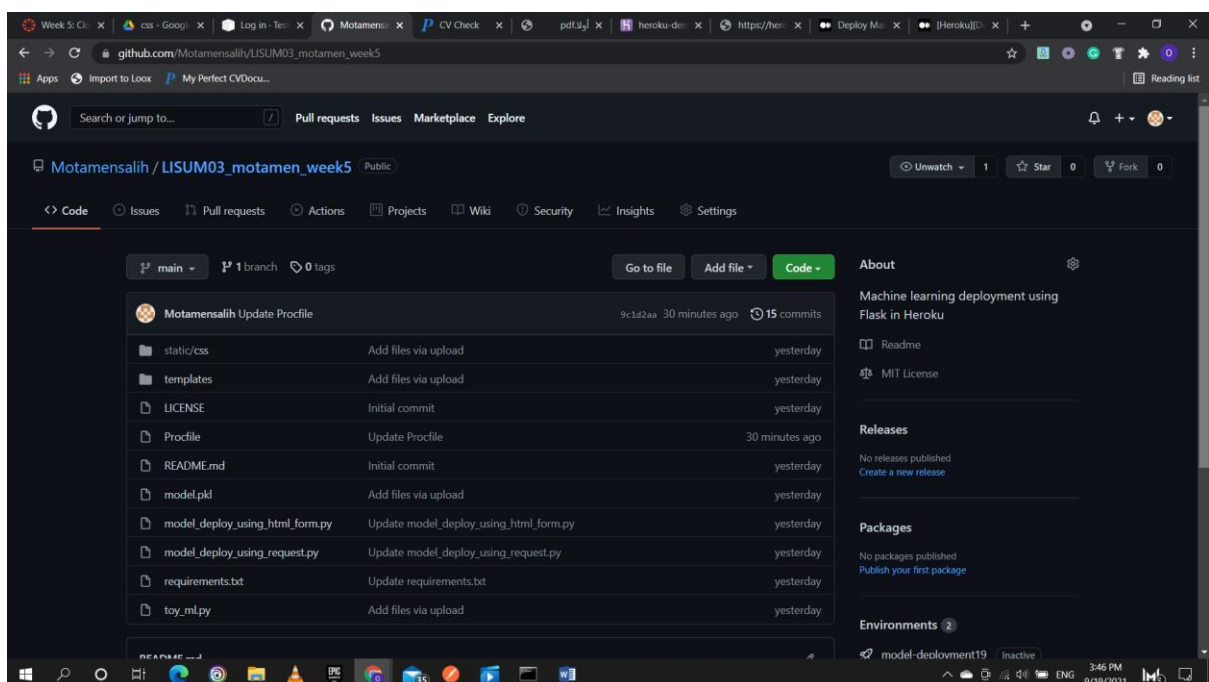
Create a file called Procfile and add it to the folder. This file tells the Heroku which process we are deploying. Add the following line to it. Don't add an extension to the file name.

web: gunicorn model\_deploy\_using\_request:app



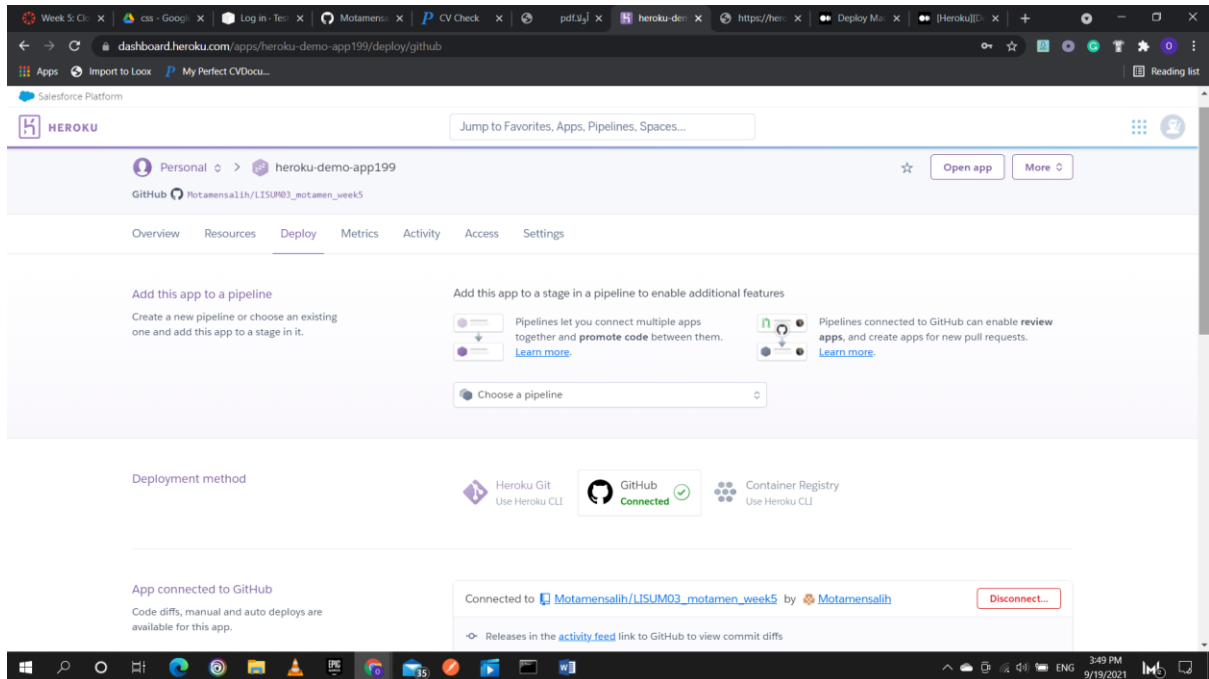
## Step 6:

Upload all the required files to a GitHub repository.



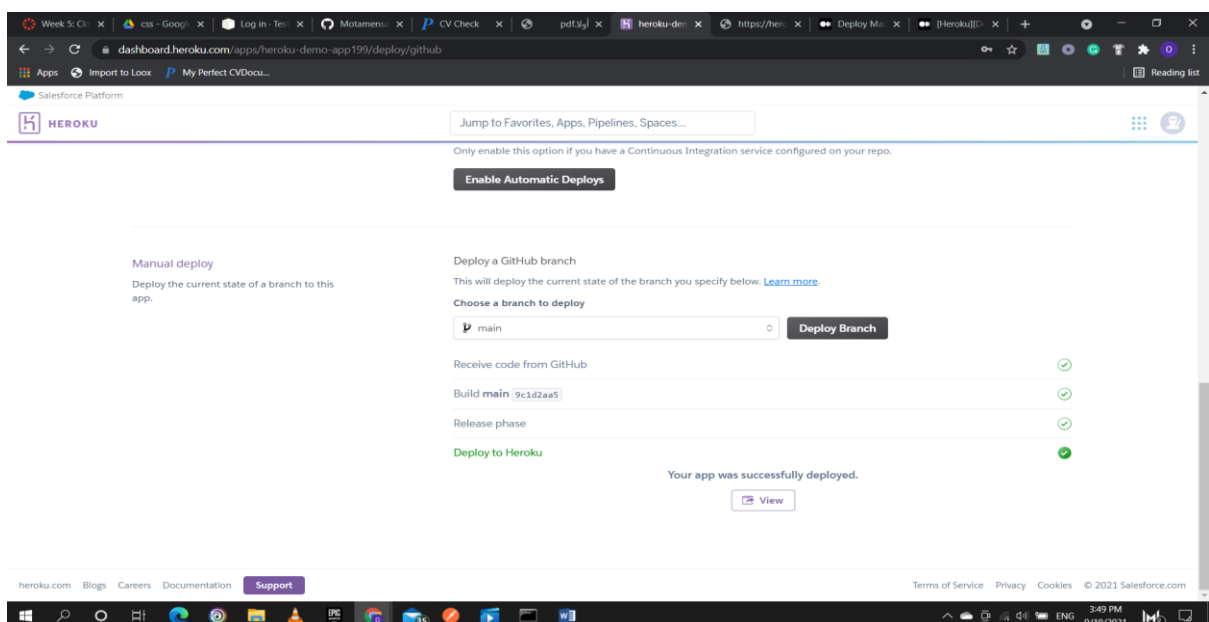
## Step 7:

Sign up on Heroku. Then, create a new app and link the GitHub repository to that app.



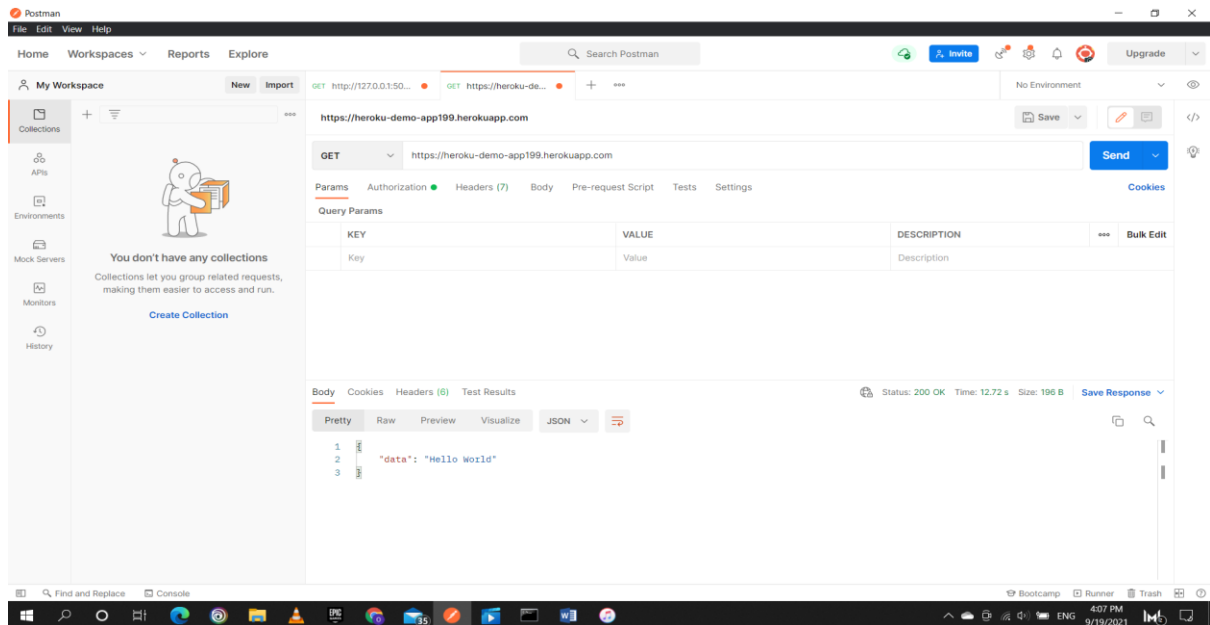
## Step 8:

Deploy the model on Heroku by pressing **deploy branch**. If everything went fine, you will see a message stating that your app was successfully deployed.

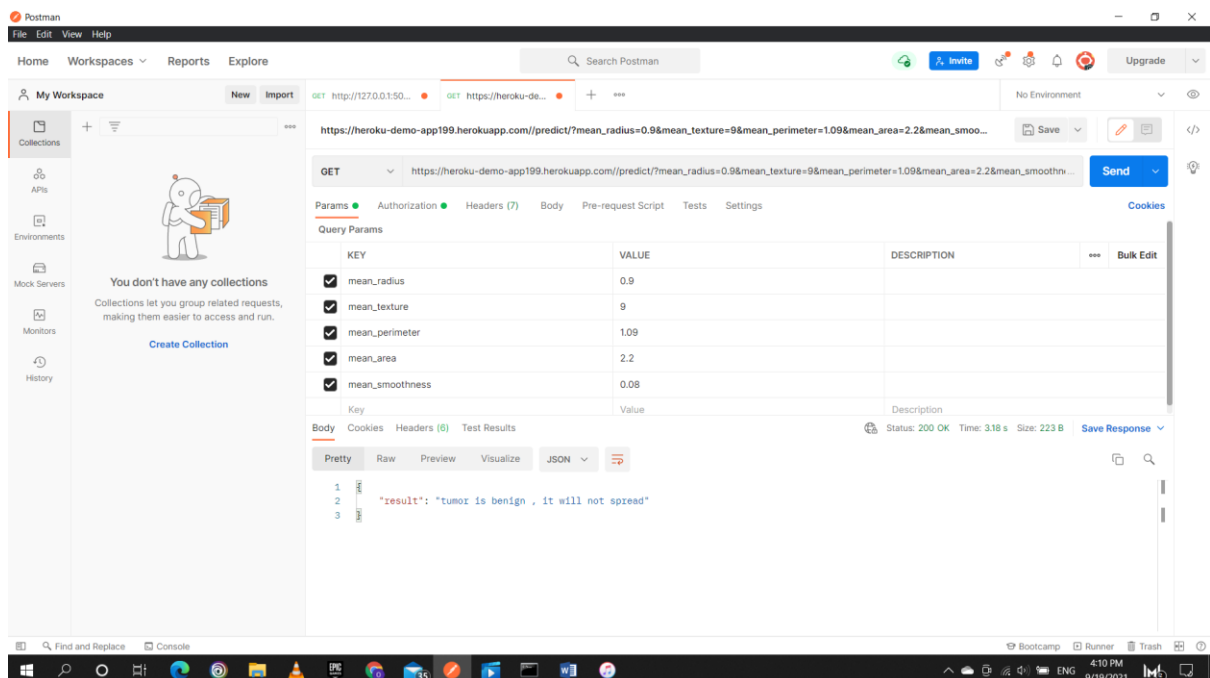


## Step 9:

Copy the given link. Then, open Postman and paste it. you will see the hello world message. This is the (‘/’) endpoint.



To predict using the model, modify the URL to include (‘/predict/’) endpoint and add the features as key-value pairs.

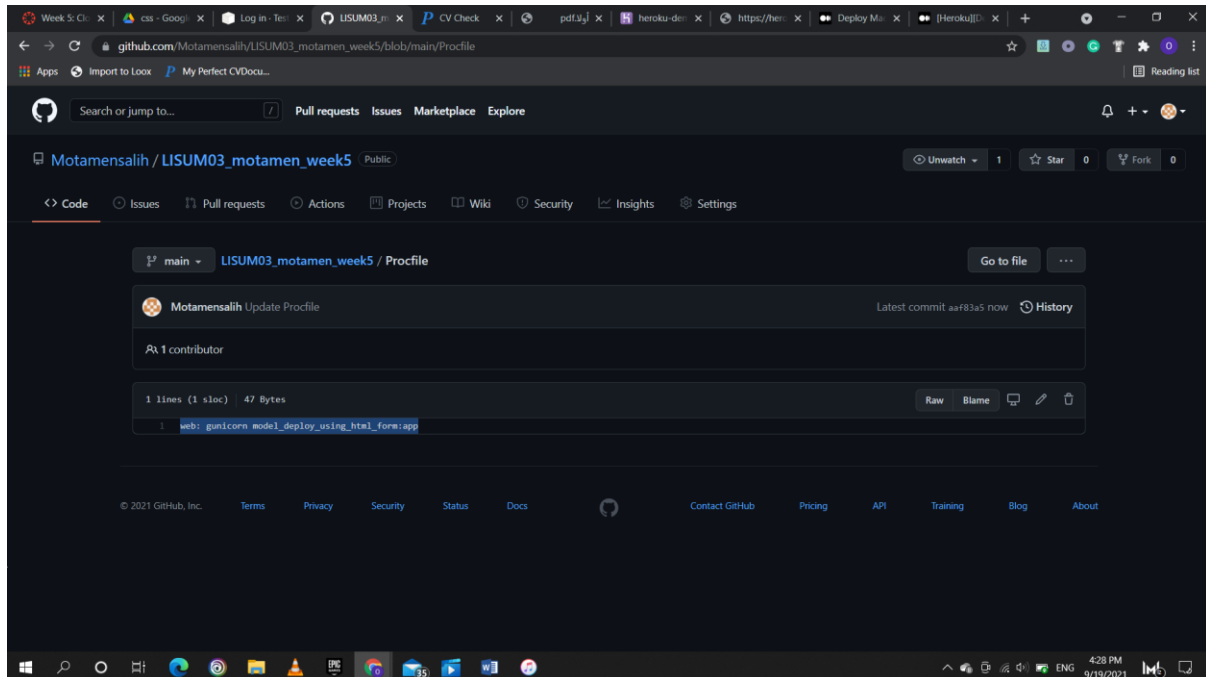


## Deployment on Heroku Using Web App:

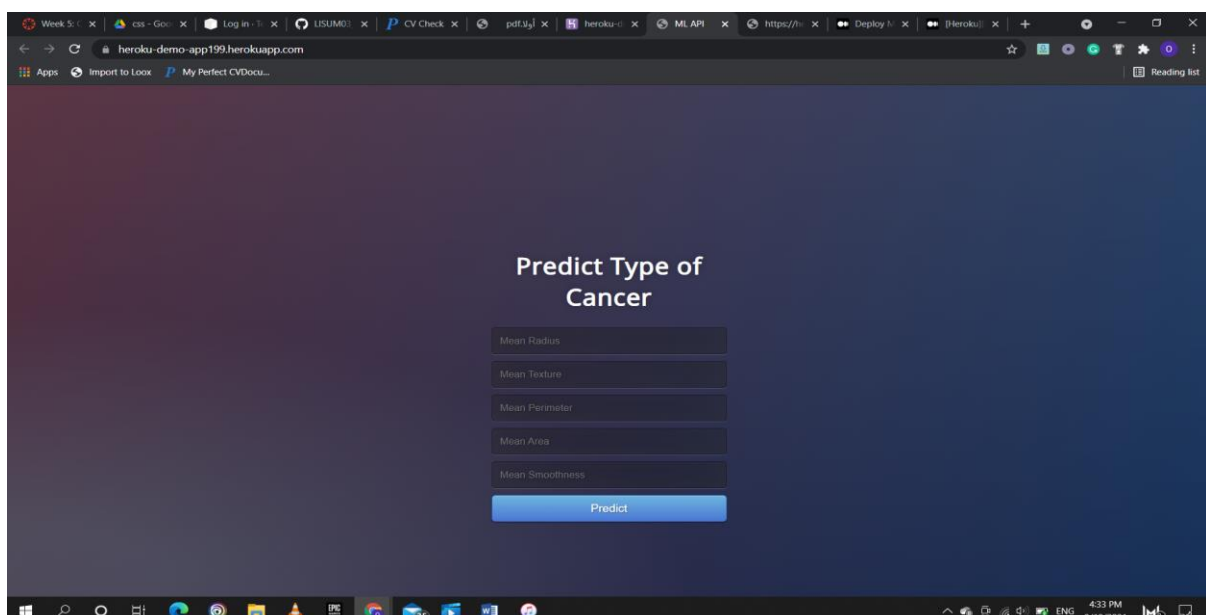
We can deploy the model using HTML forms. In this way, the model will receive the features from the form fields.

What we need to change is the Procfile only. Inside the GitHub repository, change its line to the following:

web: gunicorn model\_deploy\_using\_html\_form:app



Now, deploy the model again using the created app on Heroku and view it. You will see the following HTML page.



Enter the values and press predict to show the result of prediction.

The screenshot shows a web browser window with the URL `heroku-demo-app199.herokuapp.com/predict`. The page has a dark blue gradient background. At the top center, the text "Predict Type of Cancer" is displayed in white. Below this, there are five input fields, each with a light blue border and a dark blue background, containing the text "Mean Radius", "Mean Texture", "Mean Perimeter", "Mean Area", and "Mean Smoothness" respectively. A blue "Predict" button is located below these fields. At the bottom of the page, the text "tumor is benign , it will not spread" is displayed in white. The browser's address bar and tabs are visible at the top, and the Windows taskbar is visible at the bottom.

Predict Type of Cancer

Mean Radius

Mean Texture

Mean Perimeter

Mean Area

Mean Smoothness

Predict

tumor is benign , it will not spread