



VIRTUAL INTERNSHIP

DATA SCIENCE

Batch Code: LISUM01

Week 5:

Cloud and API Deployment

VATSAL VINESH MANDALIA

27 July 2021

Index

A. Introduction to the assignment	3
B. Web-based Deployment on Heroku	4-8
C. API-based Deployment on Heroku	9-10
D. Conclusion	11

A. Introduction to the assignment

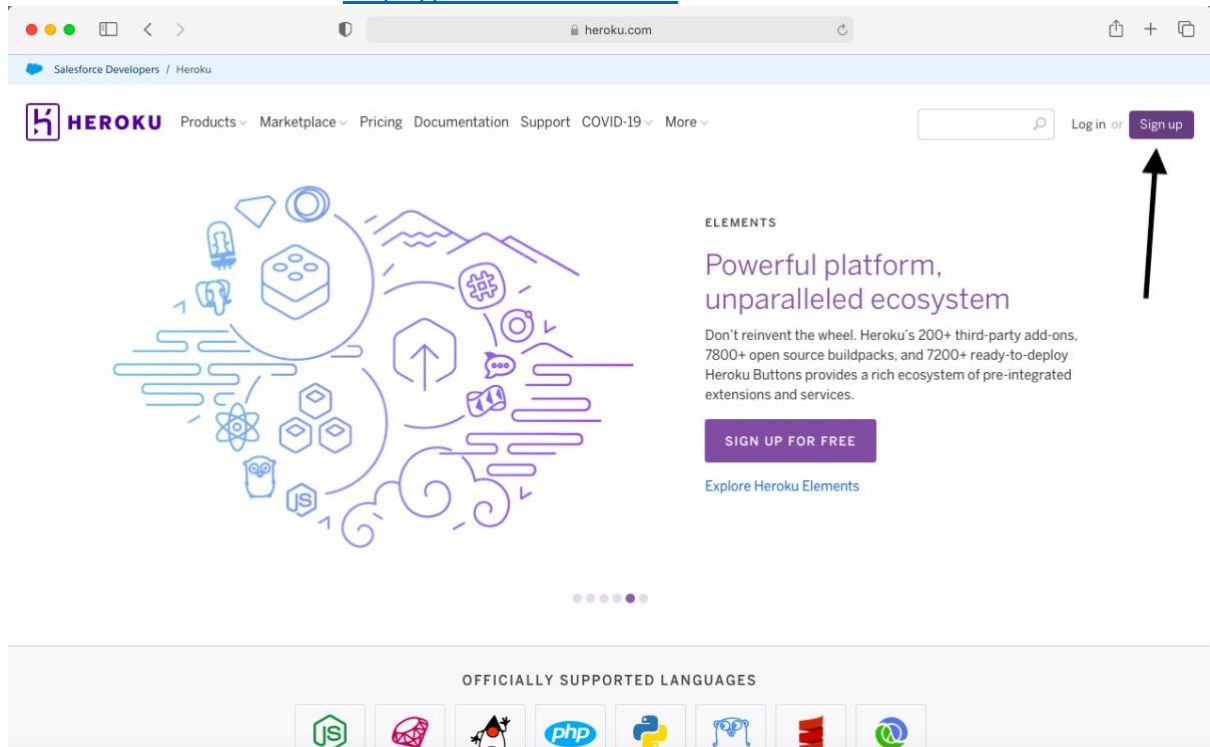
This assignment's goal is to carry out deployment of the flask app on Heroku using two ways: Web-based and API-based deployment.

Visual Studio Code is used to build the code with the programming language being Python. Safari is used as the web browser to test the deployment of the model.

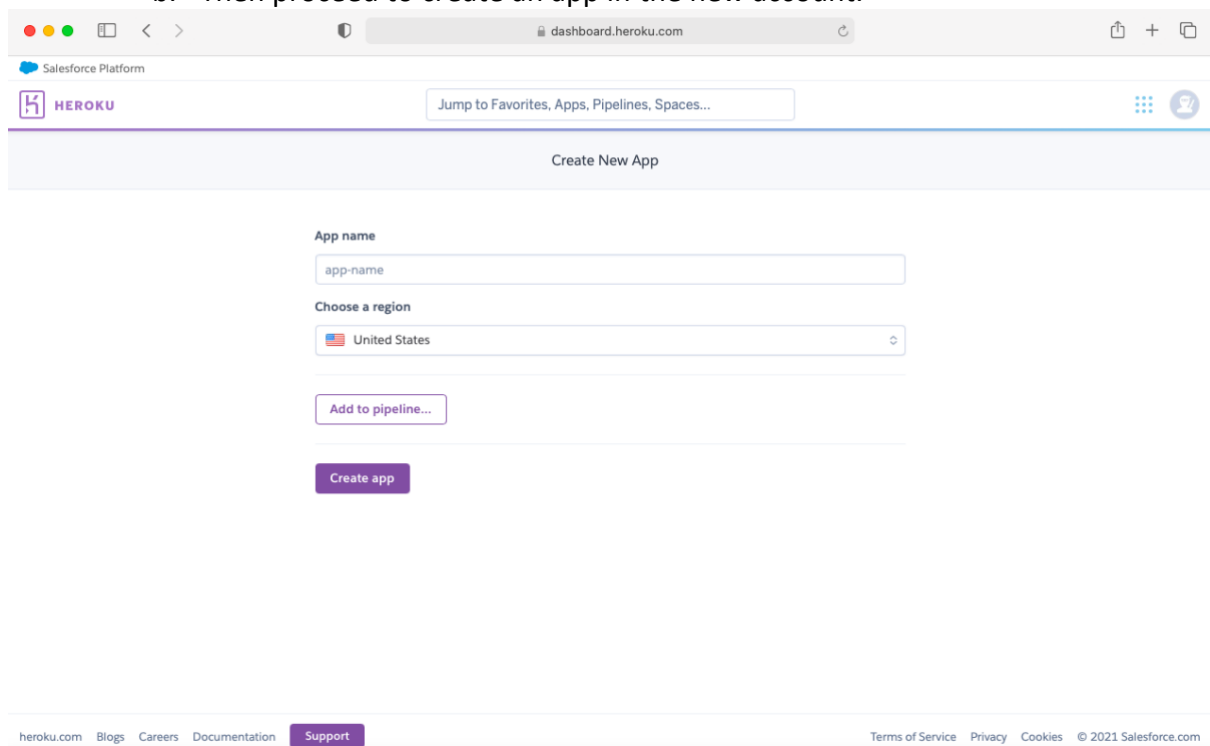
All the necessary files required to carry out the deployment process through the two methods are stored in a Github repository. The URL of the Github repository is given below, <https://github.com/Vatsal-2414/Heroku-Model-Deployment-Week5.git>

B. Web based deployment on Heroku

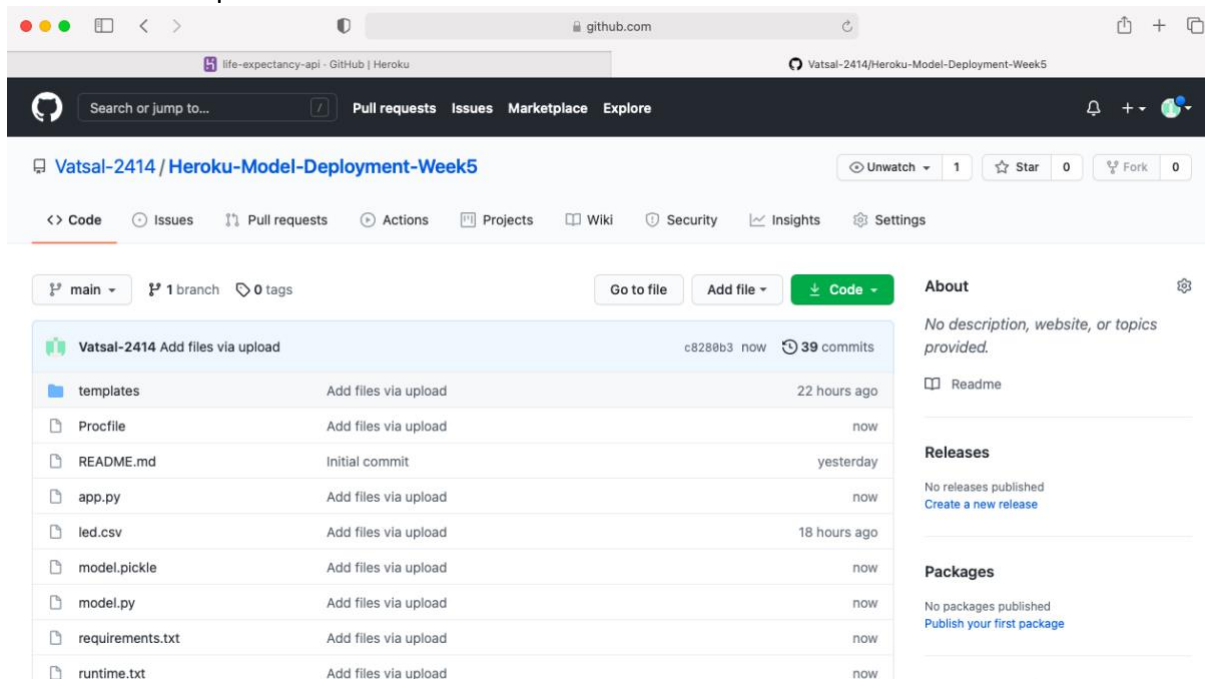
- a. To carry out web-based deployment, we need to first sign up and create an account at <https://www.heroku.com>.



- b. Then proceed to create an app in the new account.



- c. A Github repository will need to be created with all the necessary files uploaded to it.



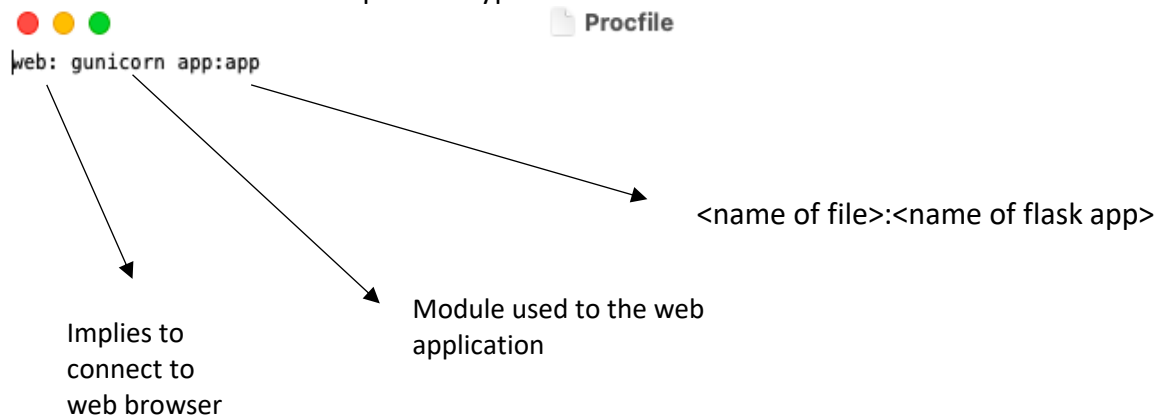
Along with the app.py, model.py and model.pickle files, we need to have three configuration files:

- requirements.txt – This contains the names and the corresponding versions of the modules required to deploy the flask application on the cloud.

```
Flask==2.0.1
Jinja2==3.0.1
numpy==1.21.1
pandas==1.3.1
scikit_learn==0.24.2
gunicorn==20.1.0
```

 requirements.txt

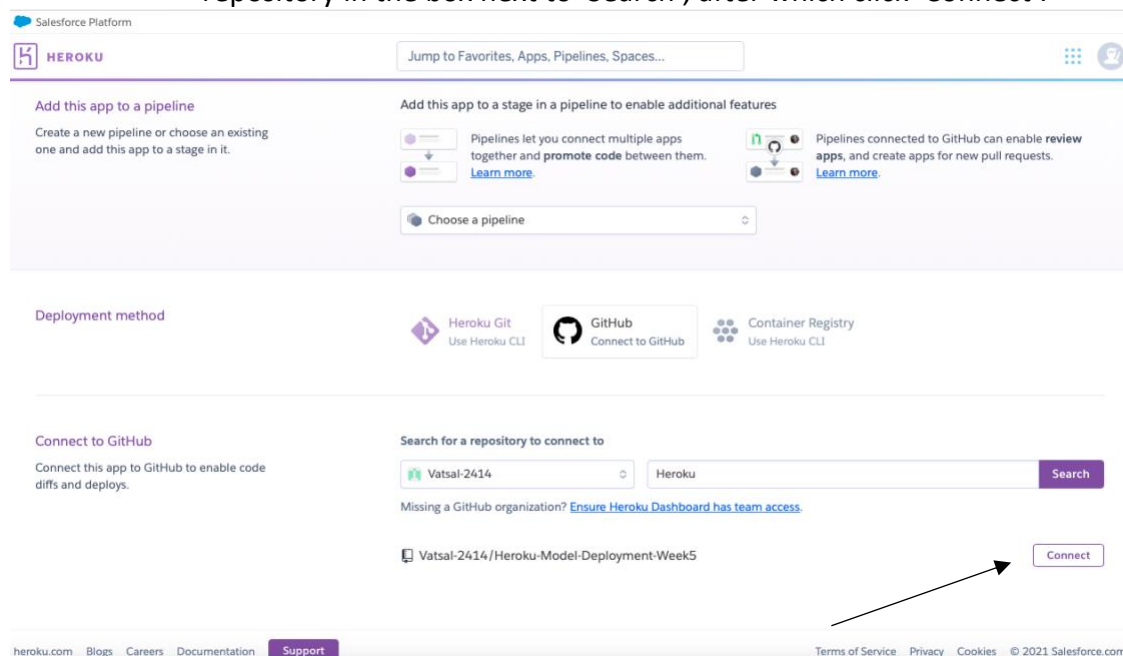
- Procfile – This file gives the process of how the Heroku app will run. It is plain text file with no extensions. The content written is shown below.



- runtime.txt – This gives the runtime version of Python used on the local machine



- After creating the app, click on the sub-heading 'Deploy'. In the deployment method section, select 'Connect to Github'. Write the name of the Github repository in the box next to 'Search', after which click 'Connect'.



- e. Two options appear for deploying either 'Automatic' or 'Manual' deploys. I opted for Manual deploy of the main branch of the repository.

The screenshot shows the Heroku dashboard for a Salesforce Platform application. The 'Automatic deploys' section is on the left, and the 'Manual deploy' section is on the right. In the 'Manual deploy' section, there is a 'Deploy a GitHub branch' heading, followed by a 'Choose a branch to deploy' dropdown menu set to 'main'. A black arrow points to the 'Deploy Branch' button next to the dropdown. Below the 'Deploy Branch' button, there is a table showing the deployment status of the application.

Step	Status
Receive code from GitHub	✓
Build main c19d19df	✓
Release phase	✓
Deploy to Heroku	✓

- f. The flask app is successfully deployed on Heroku.

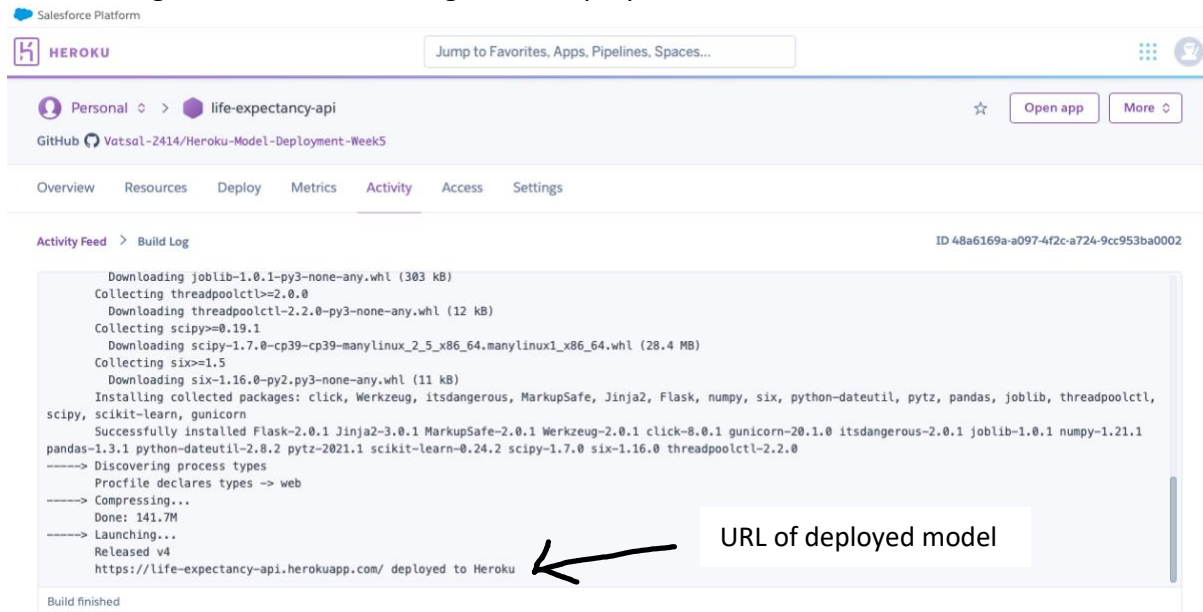
The screenshot shows the Heroku dashboard for a Salesforce Platform application. The 'Manual deploy' section is on the left, and the 'Automatic deploys' section is on the right. In the 'Manual deploy' section, there is a 'Deploy a GitHub branch' heading, followed by a 'Choose a branch to deploy' dropdown menu set to 'main'. A black arrow points to the 'Deploy Branch' button next to the dropdown. Below the 'Deploy Branch' button, there is a table showing the deployment status of the application.

Step	Status
Receive code from GitHub	✓
Build main c19d19df	✓
Release phase	✓
Deploy to Heroku	✓

Your app was successfully deployed.

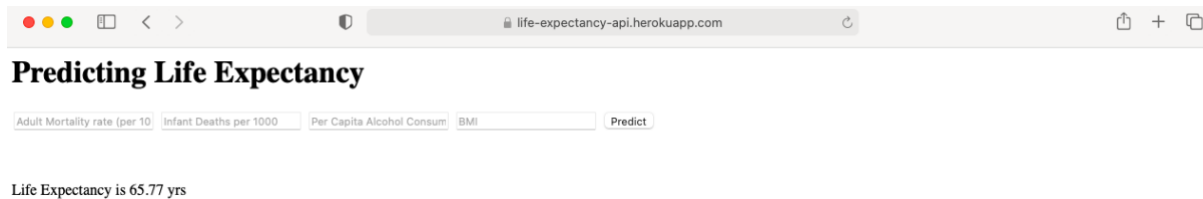
[View](#)

g. Here is the build log of the deployment.



```
Downloading joblib-1.0.1-py3-none-any.whl (303 kB)
Collecting threadpoolctl>=2.0.0
Downloading threadpoolctl-2.2.0-py3-none-any.whl (12 kB)
Collecting scipy>=0.19.1
Downloading scipy-1.7.0-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.whl (28.4 MB)
Collecting six>=1.5
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: click, Werkzeug, itsdangerous, MarkupSafe, Jinja2, Flask, numpy, six, python-dateutil, pytz, pandas, joblib, threadpoolctl,
scipy, scikit-learn, gunicorn
Successfully installed Flask-2.0.1 Jinja2-3.0.1 MarkupSafe-2.0.1 Werkzeug-2.0.1 click-8.0.1 gunicorn-20.1.0 itsdangerous-2.0.1 joblib-1.0.1 numpy-1.21.1
pandas-1.3.1 python-dateutil-2.8.2 pytz-2021.1 scikit-learn-0.24.2 scipy-1.7.0 six-1.16.0 threadpoolctl-2.2.0
----> Discovering process types
Procfile declares types -> web
----> Compressing...
Done: 141.7M
----> Launching...
Released v4
https://life-expectancy-api.herokuapp.com/ deployed to Heroku
Build finished
```

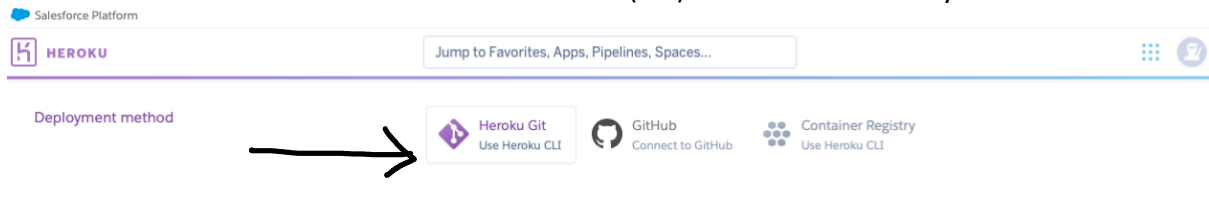
h. The URL above of the deployed model is pasted on the browser. The model is checked if it predicts the Life Expectancy for the values of the features entered.



The flask application containing the Linear Regression model is successfully deployed on Heroku through the web.

C. API-based app deployment using Heroku Command Line Interface

- a. Inside the Heroku app 'life-expectancy-api', the 'Heroku Git' deployment method is selected. To carry out deployment through this method, the Heroku Command Line Interface (CLI) is installed on the system.



- b. After which, we will need to login to the Heroku account in the local system. The below command is executed in the terminal.

```
$ heroku login
```

- c. The Github repository 'Heroku-Model-Deployment-Week5' is cloned onto local system.

```
Vatsals-MacBook-Air:~ vatsalmandalia$ git clone https://github.com/Vatsal-2414/Heroku-Model-Deployment-Week5.git
Cloning into 'Heroku-Model-Deployment-Week5'...
remote: Enumerating objects: 114, done.
remote: Counting objects: 100% (114/114), done.
remote: Compressing objects: 100% (102/102), done.
remote: Total 114 (delta 39), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (114/114), 206.43 KiB | 528.00 KiB/s, done.
Resolving deltas: 100% (39/39), done.
Vatsals-MacBook-Air:~ vatsalmandalia$ cd Heroku-Model-Deployment-Week5/
Vatsals-MacBook-Air:Heroku-Model-Deployment-Week5 vatsalmandalia$
```

- d. To make the process simple, I created another app on Heroku with the name 'life-expectancy-cli-api' from the local system.

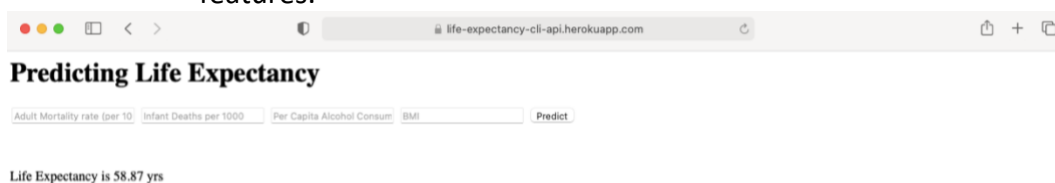
```
/vatsals-MacBook-Air:Heroku-Model-Deployment-Week5 vatsalmandalia$ heroku create life-expectancy-cli-api
Creating life-expectancy-cli-api... done
https://life-expectancy-cli-api.herokuapp.com/ | https://git.heroku.com/life-expectancy-cli-api.git
```

- e. The main branch of the Git repository 'Heroku-Model-Deployment-Week5' is pushed to Heroku. This starts the deployment of the code.

```
Vatsals-MacBook-Air:Heroku-Model-Deployment-Week5 vatsalmandalia$ git push heroku main
Enumerating objects: 114, done.
Counting objects: 100% (114/114), done.
Delta compression using up to 4 threads
Compressing objects: 100% (63/63), done.
Writing objects: 100% (114/114), 206.43 KiB | 103.21 MiB/s, done.
Total 114 (delta 39), reused 114 (delta 39), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Python app detected
remote: -----> Using Python version specified in runtime.txt
remote: -----> Installing python-3.9.6
remote: -----> Installing pip 20.2.4, setuptools 47.1.1 and wheel 0.36.2
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: Collecting Flask==2.0.1
```

```
Heroku-Model-Deployment-Week5 — -bash — 116x24
remote: Downloading scipy-1.7.0-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.whl (28.4 MB)
remote: Collecting joblib>=0.11
remote: Downloading joblib-1.0.1-py3-none-any.whl (303 kB)
remote: Collecting six>=1.5
remote: Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
remote: Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, Flask, numpy, six,
python-dateutil, pytz, pandas, threadpoolctl, scipy, joblib, scikit-learn, gunicorn
remote: Successfully installed Flask-2.0.1 Jinja2-3.0.1 MarkupSafe-2.0.1 Werkzeug-2.0.1 click-8.0.1 gunicorn-
20.1.0 itsdangerous-2.0.1 joblib-1.0.1 numpy-1.21.1 pandas-1.3.1 python-dateutil-2.8.2 pytz-2021.1 scikit-learn-0.24
.2 scipy-1.7.0 six-1.16.0 threadpoolctl-2.2.0
remote: -----> Discovering process types
remote: Procfile declares types -> web
remote:
remote: -----> Compressing...
remote: Done: 141.7M
remote: -----> Launching...
remote: Released v3
remote: https://life-expectancy-cli-api.herokuapp.com/ deployed to Heroku
remote: Verifying deploy... done
To https://git.heroku.com/life-expectancy-cli-api.git
* [new branch] main -> main
Vatsals-MacBook-Air:Heroku-Model-Deployment-Week5 vatsalmandalia$ heroku open
```

- f. By typing the command – 'heroku open' in the terminal, the flask app opens in the browser. The model is tested by inputting certain values for the four features.



Finally, the flask application is also successfully deployed using Heroku Git.

D. Conclusion

The flask app was successfully deployed on the open-source cloud platform Heroku through the two ways- Web and API based.