



**Data Glacier**

Your Deep Learning Partner

# Cloud and API deployment

Batch code: LISP01

Submission date: 28 March, 2021

Submission to: Data Glacier

# Agenda

Environment creation  
Configuring Heroku  
Deployment into Heroku  
Deployment Test  
API Code  
API Deployment  
API Deployment Test

# Environment creation

Environment created

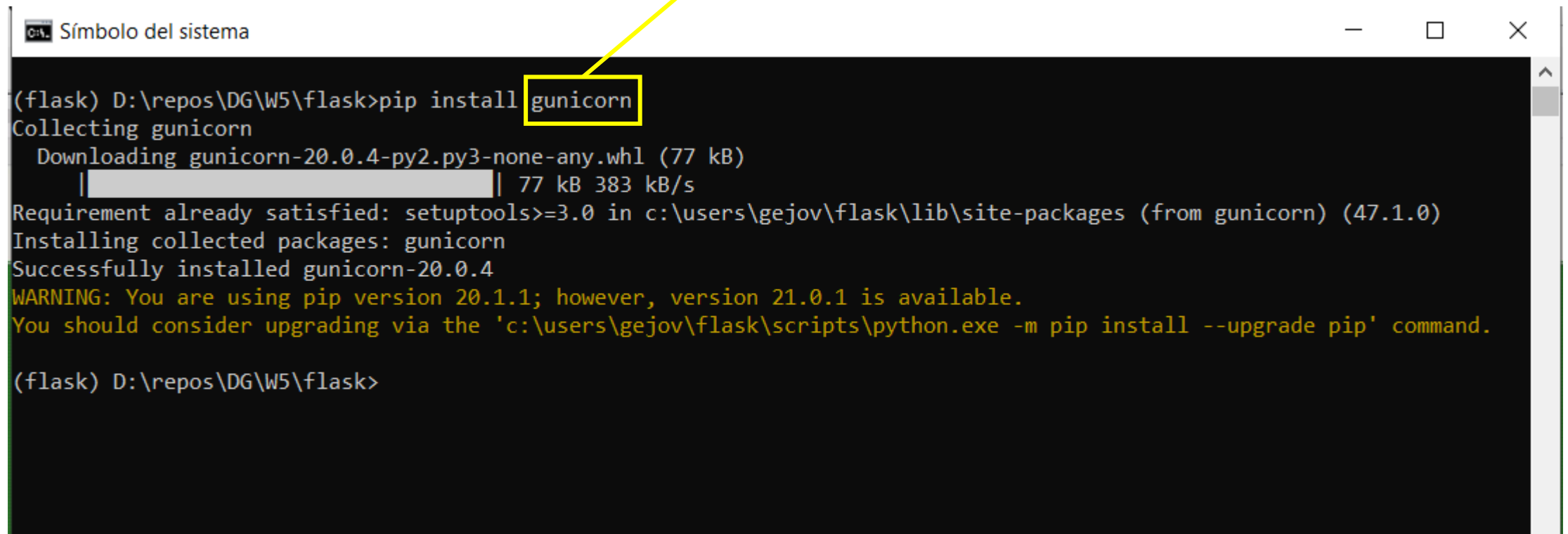


```
Símbolo del sistema
(flask) D:\repos\DG\W5\flask>
```

# Environment creation

Installing libraries

HTTP Server



```
Símbolo del sistema

(flask) D:\repos\DG\W5\flask>pip install gunicorn
Collecting gunicorn
  Downloading gunicorn-20.0.4-py2.py3-none-any.whl (77 kB)
    | 77 kB 383 kB/s
Requirement already satisfied: setuptools>=3.0 in c:\users\gejov\flask\lib\site-packages (from gunicorn) (47.1.0)
Installing collected packages: gunicorn
Successfully installed gunicorn-20.0.4
WARNING: You are using pip version 20.1.1; however, version 21.0.1 is available.
You should consider upgrading via the 'c:\users\gejov\flask\scripts\python.exe -m pip install --upgrade pip' command.

(flask) D:\repos\DG\W5\flask>
```

# Environment creation

Saving libraries version to deployment

```
Símbolo del sistema

(flask) D:\repos\DG\W5\flask>pip freeze
click==7.1.2
Flask==1.1.2
gunicorn==20.0.4
itsdangerous==1.1.0
Jinja2==2.11.3
joblib==1.0.1
MarkupSafe==1.1.1
numpy==1.20.1
scikit-learn==0.23.2
scipy==1.6.1
threadpoolctl==2.1.0
Werkzeug==1.0.1
WTForms==2.3.3
```

```
Símbolo del sistema

(flask) D:\repos\DG\W5\flask>pip freeze > requirements.txt

(flask) D:\repos\DG\W5\flask>
```

# Configuring Heroku

## Creating User

The screenshot shows the Heroku dashboard interface. At the top, there's a navigation bar with the Heroku logo, a search bar containing "Jump to Favorites, Apps, Pipelines, Spaces...", and user profile icons. Below the navigation bar is a purple banner with the text "Welcome to Heroku" and "Now that your account has been set up, here's how to get started." The main content area features two primary actions: "Create a new app" and "Create a team". Below these, there's a section for language guides with icons for Node.js, Ruby, Java, PHP, Python, Go, Scala, and Clojure.

**HEROKU**

Jump to Favorites, Apps, Pipelines, Spaces...

**Welcome to Heroku**  
Now that your account has been set up, here's how to get started.

**Create a new app**  
Create your first app and deploy your code to a running dyno.  
[Create new app](#)

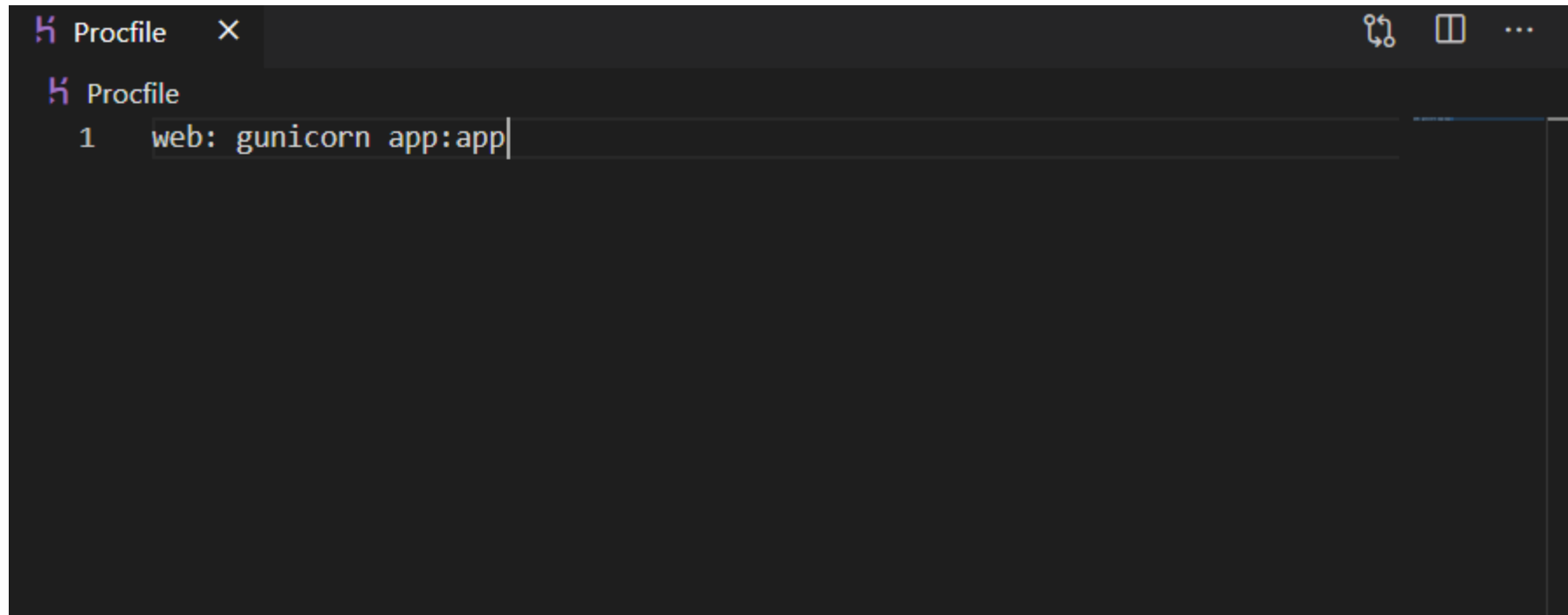
**Create a team**  
Create teams to collaborate on your apps and pipelines.  
[Create a team](#)

**Looking for help getting started with your language?**  
Get started by reading one of our language guides in the Dev Center

[Node.js](#) [Ruby](#) [Java](#) [PHP](#) [Python](#) [Go](#) [Scala](#) [Clojure](#)

# Configuring Heroku

Saving file for gunicorn

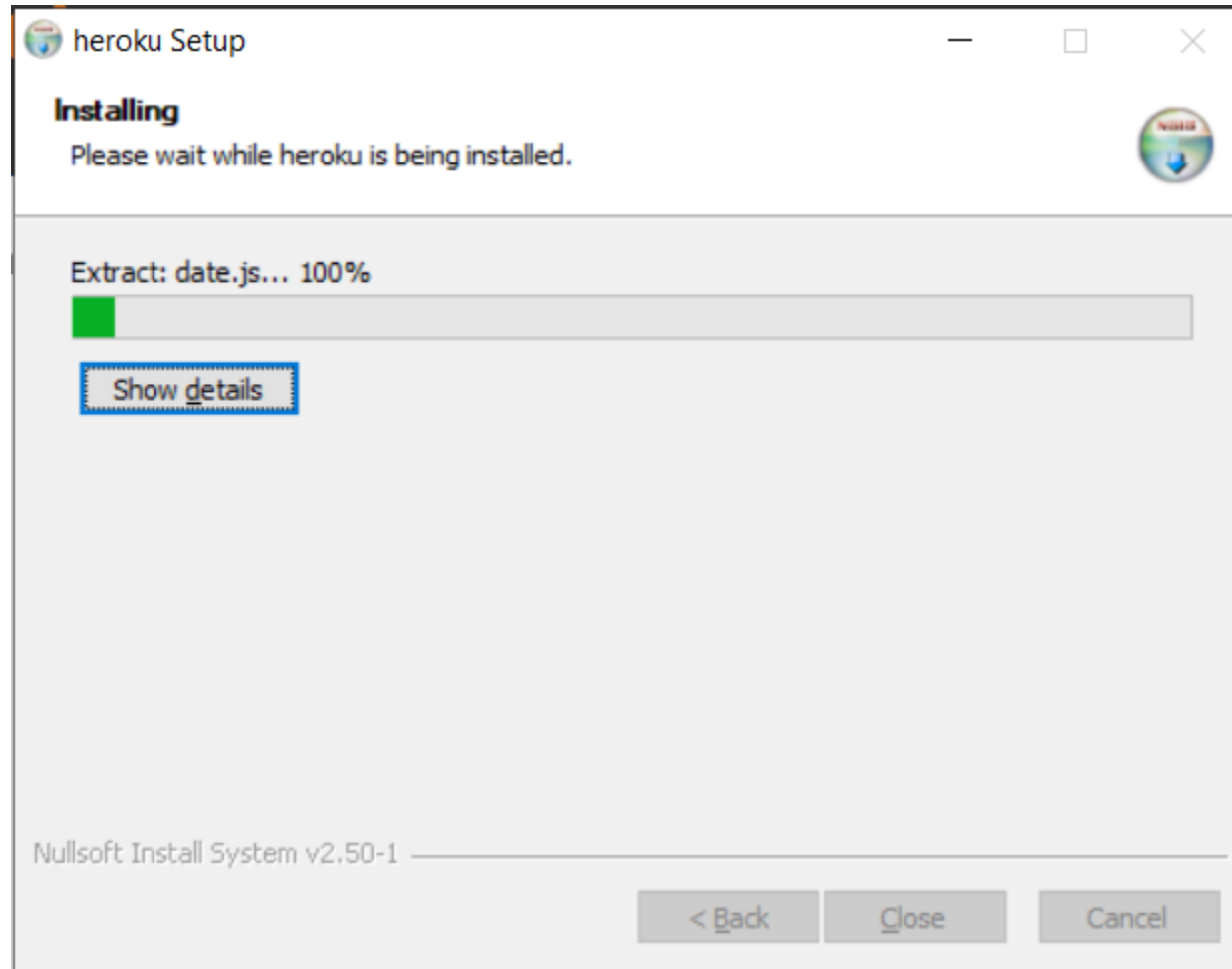


```
Procfile
1 web: gunicorn app:app
```

The image shows a code editor window with a dark theme. The title bar at the top reads 'Procfile' with a close button on the right. The editor content shows a file named 'Procfile' with a single line of code: '1 web: gunicorn app:app'. The line number '1' is in the left margin, and the text is highlighted with a light blue selection background. The cursor is at the end of the line.

# Configuring Heroku

Installing Heroku CLI






# Configuring Heroku

## Creating Heroku App



```
(flask) D:\repos\DG\W5\flask>heroku create
» Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
Creating app... !
! Invalid credentials provided.
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/18d23448-a5cc-4803-bd73-c1c49516283d?requestor=SFMyNTY.g
2gDbQAAAA4x0TAuMjMyLjc0LjIxOG4GAE1HfWF4AWIAAVGA.YxOC4aadhHoLxP0S9xfwb2fSAfSwTxZV0xtIfHJY10A
Logging in... done
Logged in as sunstreaker.cynos@gmail.com
Creating app... done, ⚙ fathomless-citadel-82794
https://fathomless-citadel-82794.herokuapp.com/ | https://git.heroku.com/fathomless-citadel-82794.git
```


# Configuring Heroku


Created App into Heroku

 **HEROKU**


Jump to Favorites, Apps, Pipelines, Spaces...


 


 Personal ↕


 **Welcome to Heroku**  
Now that your account has been set up, here's how to get started.


[Show next steps](#)


 Filter apps and pipelines

 fathomless-citadel-82794

**Gerson Orihuela**  
sunstreaker.cynos@gmail.com

 Account settings

 Notifications

 Sign out

heroku-20 • United States ☆

# Deployment into Heroku

## Commands for deployment

The screenshot displays the Heroku CLI documentation page. The header includes the Heroku logo, a search bar with the text "Jump to Favorites, Apps, Pipelines, Spaces...", and a user profile dropdown menu. The main content area is divided into three columns. The left column, titled "Deploy using Heroku Git", explains that users can use git in the command line or a GUI tool. The middle column, titled "Install the Heroku CLI", provides instructions on downloading and installing the CLI, including a link to the Heroku CLI page and a note about logging in to create a new SSH public key. Below this, it shows the command `$ heroku login`. The right column, titled "Create a new Git repository", explains how to initialize a git repository in a new or existing directory, showing the commands `$ cd my-project/`, `$ git init`, and `$ heroku git:remote -a fathomless-citadel-82794`. Below this, it shows the commands `$ git add .`, `$ git commit -am "make it better"`, and `$ git push heroku master`. A blue callout box with a key icon states: "You can now change your main deploy branch from 'master' to 'main' for both manual and automatic deploys, please follow the instructions [here](#)." The bottom section, titled "Existing Git repository", explains how to add the heroku remote for existing repositories, showing the command `$ heroku git:remote -a fathomless-citadel-82794`. The user profile dropdown menu on the right shows the user's name "Gerson Orihuela", email "sunstreaker.cynos@gmail.com", and links for "Account settings", "Notifications", and "Sign out".

**HEROKU** Jump to Favorites, Apps, Pipelines, Spaces...

**Deploy using Heroku Git**  
Use git in the command line or a GUI tool to deploy this app.

**Install the Heroku CLI**  
Download and install the [Heroku CLI](#).  
If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

**Create a new Git repository**  
Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a fathomless-citadel-82794
```

**Deploy your application**  
Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

**Existing Git repository**  
For existing repositories, simply add the `heroku` remote

```
$ heroku git:remote -a fathomless-citadel-82794
```

**Gerson Orihuela**  
sunstreaker.cynos@gmail.com

- Account settings
- Notifications
- Sign out

# Deployment into Heroku

Following prompts

```
(flask) D:\repos\DG\W5\flask>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/a2c35554-7098-4759-870a-a42792569112?requestor=SFMyNTY.g
2gDbQAAAA4xOTAuMjMyLjc0LjIxOG4GAA3qhWF4AWIAAVGA.W1RU52GUcr1pwjGK-lwNwGFrSAc0Ns2TiJOx2wv_5yE
Logging in... done
Logged in as sunstreaker.cynos@gmail.com

(flask) D:\repos\DG\W5\flask>git init
Initialized empty Git repository in D:/repos/DG/W5/flask/.git/

(flask) D:\repos\DG\W5\flask>heroku git:remote -a fathomless-citadel-82794
set git remote heroku to https://git.heroku.com/fathomless-citadel-82794.git

(flask) D:\repos\DG\W5\flask>git add .

(flask) D:\repos\DG\W5\flask>git commit -am "make it better"
[master (root-commit) 4db0f08] make it better
8 files changed, 231 insertions(+)
create mode 100644 Procfile
create mode 100644 app.py
create mode 100644 model/finalized_model.sav
create mode 100644 requirements.txt
create mode 100644 static/css/123.jpg
create mode 100644 static/css/html.jpg
create mode 100644 static/css/styles.css
create mode 100644 templates/registrarse.html
```

# Deployment into Heroku

Following prompts

Done deployment

```
(flask) D:\repos\DG\W5\flask>git push heroku master
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 24.16 MiB | 1.05 MiB/s, done.
Total 20 (delta 4), reused 0 (delta 0), 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: -----> Installing python-3.6.13
remote: -----> Installing pip 20.1.1, setuptools 47.1.1 and
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: Collecting click==7.1.2
remote:   Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
remote: Collecting Flask==1.1.2
remote:   Downloading Flask-1.1.2-py2.py3-none-any.whl (95 kB)
remote: Collecting gunicorn==20.0.4
remote:   Downloading gunicorn-20.0.4-py2.py3-none-any.whl (130 kB)
remote: Collecting itsdangerous==1.1.0
remote:   Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
remote: Collecting Jinja2==2.11.3
remote:   Downloading Jinja2-2.11.3-py2.py3-none-any.whl (125 kB)
remote: Collecting joblib==1.0.1
remote:   Collecting numpy==1.19.5
remote:     Downloading numpy-1.19.5-cp36-cp36m-manylinux2010_x86_64.whl (14.8 MB)
remote:   Collecting scikit-learn==0.23.2
remote:     Downloading scikit_learn-0.23.2-cp36-cp36m-manylinux1_x86_64.whl (6.8 MB)
remote:   Collecting scipy==1.5.4
remote:     Downloading scipy-1.5.4-cp36-cp36m-manylinux1_x86_64.whl (25.9 MB)
remote:   Collecting threadpoolctl==2.1.0
remote:     Downloading threadpoolctl-2.1.0-py3-none-any.whl (12 kB)
remote:   Collecting Werkzeug==1.0.1
remote:     Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
remote:   Collecting WTForms==2.3.3
remote:     Downloading WTForms-2.3.3-py2.py3-none-any.whl (169 kB)
remote:   Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask, gunicorn, joblib,
remote:     numpy, threadpoolctl, scipy, scikit-learn, WTForms
remote:   Successfully installed Flask-1.1.2 Jinja2-2.11.3 MarkupSafe-1.1.1 WTForms-2.3.3 Werkzeug-1.0.1 click-7.1.2
remote:     gunicorn-20.0.4 itsdangerous-1.1.0 joblib-1.0.1 numpy-1.19.5 scikit-learn-0.23.2 scipy-1.5.4 threadpoolctl-2.1.0
remote: -----> Discovering process types
remote:   Procfile declares types -> web
remote: -----> Compressing...
remote:   Done: 125.4M
remote: -----> Launching...
remote:   Released v3
remote:   https://fathomless-citadel-82794.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/fathomless-citadel-82794.git
* [new branch]      master -> master
```

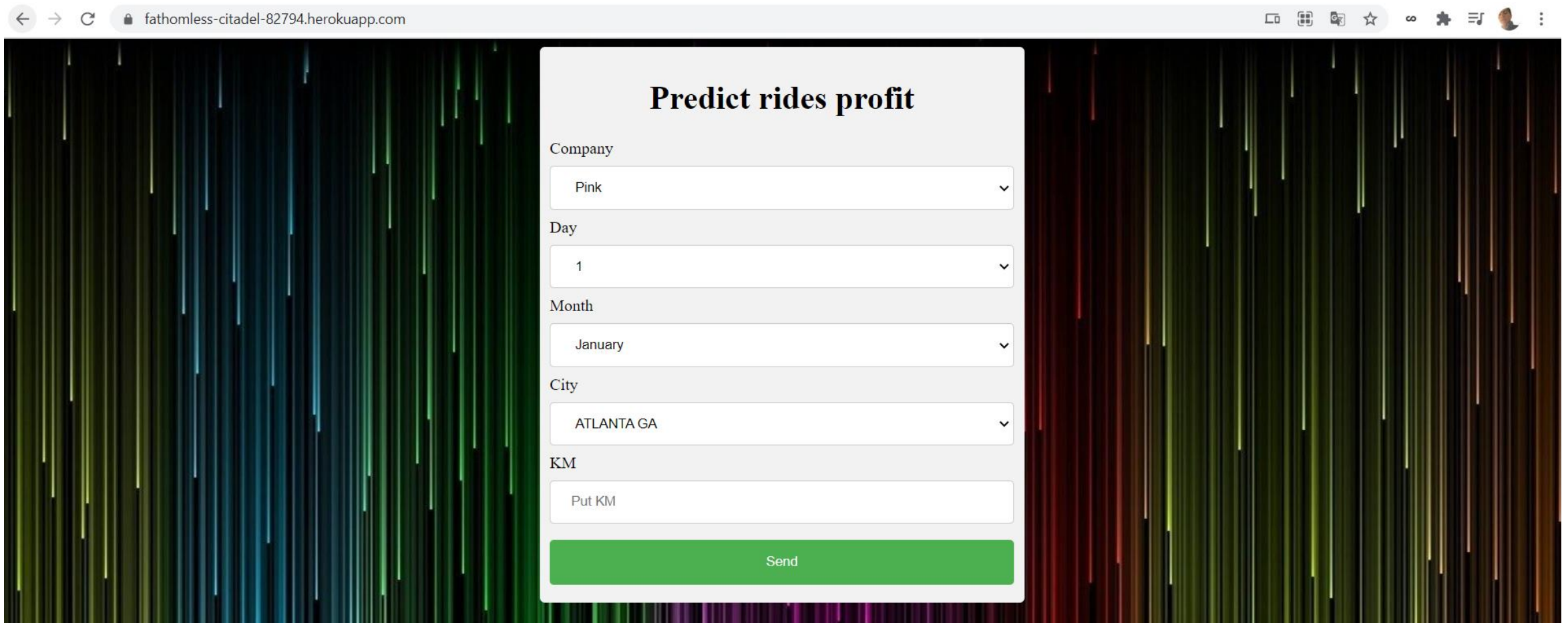
# Deployment Test

URL for Implemented Model

<https://fathomless-citadel-82794.herokuapp.com/>

# Deployment Test

Inserting URL



A screenshot of a web browser displaying a form titled "Predict rides profit". The browser's address bar shows the URL "fathomless-citadel-82794.herokuapp.com". The form is centered on a dark background with vertical light streaks. It contains five dropdown menus for "Company" (selected: Pink), "Day" (selected: 1), "Month" (selected: January), and "City" (selected: ATLANTA GA), followed by a text input field for "KM" (containing "Put KM"). A green "Send" button is at the bottom.

← → ↻ 🔒 fathomless-citadel-82794.herokuapp.com

## Predict rides profit

Company

Pink ▼

Day

1 ▼

Month

January ▼

City

ATLANTA GA ▼

KM

Put KM

Send

# Deployment Test

## Testing App

← → ↻ fathomless-citadel-82794.herokuapp.com/upload

**Predict rides profit**

Company

Pink ▾

Day

1 ▾

Month

January ▾

City

ATLANTA GA ▾

KM

Put KM

Send

**The profit should be: \$285.09**



# API Code

## Coding Flask API

Loaded model ←

Input data ←

Prediction ←

```
from flask import Flask, jsonify, request
import numpy as np
import pickle

app = Flask(__name__)

filename = 'model/finalized_model.sav'
model = pickle.load(open(filename, 'rb'))

@app.route('/', methods=['POST'])
def create_store():
    request_data = request.get_json()
    km=request_data['km']
    company=request_data['company']
    city=request_data['city']
    day=request_data['day']
    month=request_data['month']

    print(type(km),type(company),type(city),type(day),type(month))

    int_features = [float(km),int(month),int(day),int(company),int(city)]
    final_features = [np.array(int_features)]
    prediction =model.predict(final_features)

    output = round(prediction[0],2)
    print(output)

    return jsonify({'output':output})

app.run(port=5000)
```

# API Code

## API Test in local

The screenshot displays a REST client interface for testing a local API. The request is a POST to `http://127.0.0.1:5000/`. The body contains a JSON object with input data. The response is a 200 OK status with a JSON object containing the prediction output.

**URL** ← `POST http://127.0.0.1:5000/`

**Input data** ← 

```
{
  "km": "100",
  "company": "1",
  "city": "0",
  "day": "1",
  "month": "1"
}
```

**Prediction** ← 

```
{
  "output": 401.79
}
```

flask\_1 / http://127.0.0.1:5000/ Save Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Body Cookies Headers (4) Test Results 200 OK 148 ms 163 B Save Response

Pretty Raw Preview Visualize JSON

# API Deployment

Following prompts

```
(flask_rest) D:\repos\DG\pW5\flask_res>heroku create
Creating app... done, ⬢ intense-springs-49493
https://intense-springs-49493.herokuapp.com/ | https://git.heroku.com/intense-springs-49493.git

(flask_rest) D:\repos\DG\pW5\flask_res>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/d5438fd4-69d3-4737-8f2b-3e131b15edb3?requestor=SFMyNTY.
2gDbQAAAA4x0TAuMjMyLjc0LjIyNm4GANO6wnp4AWIAAVGA.Miqe03aH87yaQ4SvXc8Bs3HPpRlnCuh403DYXEcnP5I
Logging in... done
Logged in as sunstreaker.cynos@gmail.com

(flask_rest) D:\repos\DG\pW5\flask_res>git init
Reinitialized existing Git repository in D:/repos/DG/pW5/flask_res/.git/

(flask_rest) D:\repos\DG\pW5\flask_res>heroku git:remote -a intense-springs-49493
set git remote heroku to https://git.heroku.com/intense-springs-49493.git

(flask_rest) D:\repos\DG\pW5\flask_res>git add .

(flask_rest) D:\repos\DG\pW5\flask_res>git commit -am "make it better"
[master ac60401] make it better
1 file changed, 2 insertions(+), 1 deletion(-)
```

# API Deployment

Following prompts

```
(flask_rest) D:\repos\DG\pW5\flask_rest>git push heroku master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 23.83 MiB | 1.31 MiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), 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: -----> Installing python-3.6.13
remote: -----> Installing pip 20.1.1, setuptools 47.1.1 and wheel 0.34.2
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote:      Collecting click==7.1.2
remote:      Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
remote:      Collecting Flask==1.1.2
remote:      Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
remote:      Collecting gunicorn==20.1.0
remote:      Downloading gunicorn-20.1.0.tar.gz (370 kB)
```

# API Deployment

URL for Implemented API

<https://intense-springs-49493.herokuapp.com/>

# API Deployment Test

API Test in server Heroku

The screenshot displays a REST client interface with a POST request to `https://intense-springs-49493.herokuapp.com/`. The request body is a JSON object with the following data: `{ "km": "100", "company": "1", "city": "0", "day": "1", "month": "1" }`. The response is a JSON object: `{ "output": 401.79 }`. Green arrows and boxes highlight the URL, input data, and prediction.

**URL** ← `POST https://intense-springs-49493.herokuapp.com/`

**Input data** ← `{  
 "km": "100",  
 "company": "1",  
 "city": "0",  
 "day": "1",  
 "month": "1"  
}`

**Prediction** ← `{  
 "output": 401.79  
}`

# Thank You