

# INTRO to DATA SCIENCE

## WEB DEVELOPMENT WITH FLASK/HEROKU

## LAST TIME:

I. BIG DATA

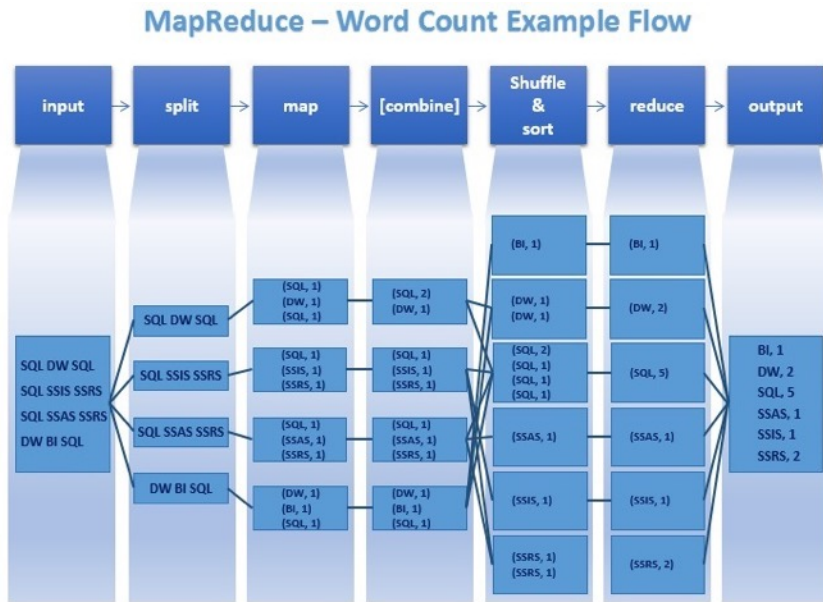
II. PROGRAMMING MODEL

III. IMPLEMENTATION DETAILS

IV. WORD COUNT EXAMPLE

## EXERCISE:

V. MAP-REDUCE USING PYTHON



# **QUESTIONS?**

**WHAT WAS THE MOST INTERESTING THING YOU LEARNT?**

**WHAT WAS THE HARDEST TO GRASP?**

**I. WHAT IS WEB DEVELOPMENT?**

**II. WHAT IS HEROKU / FLASK?**

**III. MVC**

**IV. DEPLOYING KNN**

- **UNDERSTAND MODEL - VIEW - CONTROLLER**
- **BE ABLE TO DEPLOY A MODEL ON HEROKU**
- **BE ABLE TO USE VIRTUAL ENVIRONMENTS**

---

**INTRO TO DATA SCIENCE**

---

# **WEB DEVELOPMENT**

Q: what is web development

A: The work involved with building and maintaining a live website



```
57 ga.type = 'text';
58 ga.async = true;
59 ga.src = ('https:' == document.location.protocol ? 'https://www' : 'http://www') + '.google-analytics.com/ga.js';
60 var s = document.getElementsByTagName('script')[0];
61 s.parentNode.insertBefore(ga, s);
62
63 ));
64 </script>
65 <?php
66 if (is_singular() && get_option('thread_comments')) {
67     wp_enqueue_script('comment-reply');
68 }
69 <?php wp_head(); ?>
70 </head>
71 <body <?php body_class(); ?>
72 <div id="header">
73     <div class="wrapper">
74         <h1>
75             <?php if (is_front_page() && Spaced < 2) : ?>
76                 " />
77             <?php else : ?>
78                 <a href="/" title="Root">" />
79             <?php endif; ?>
80         </h1>
81         <form id="search" method="get" action="/">
82             <div>
83                 <input accesskey="s" type="text" id="s" name="s" />
84                 <input type="submit" value="Find" />
85             </div>
86         </form>
87     </div>
88 </div>
```

## Two types of web development

### **Front-end:**

HTML/CSS, Responsive design

*Makes things pretty / easy to use*

### **Back-end:**

Many languages, Model View Controller, Databases

*Makes the site "work"*



Full-stack Development comprises of both front-end and back-end work

Part of being a web developer is knowing the technologies used:

**Web-framework**  
database and site code



**Deployment**  
“serve” the website  
so that other people can use it



---

**HEROKU / FLASK**

---

**HEROKU / FLASK**

Web Development is hard..

Which is why GA has several classes dedicated to it

We will use two very simple web development tools:  
**Heroku** and **Flask**

Did someone say Flask!?



# Flask

web development,  
one drop at a time

Did someone say Flask!?



# Flask

web development,  
one drop at a time

Flask is a micro-web-framework based entirely in python

## **What does that mean?**

It means we can write the entire backend in Python!

Did someone say Heroku!?



Heroku is a Salesforce company that lets us deploy our websites easily

**What does that mean?**

We use heroku to rent servers to host the website

One thing that stays constant over all technologies is the idea of the

**Model View Controller paradigm**



---

**MODEL VIEW CONTROLLER**

---

**MODEL VIEW CONTROLLER**

---

## MODEL VIEW CONTROLLER

---

Model View Controller:

---

## MODEL VIEW CONTROLLER

---

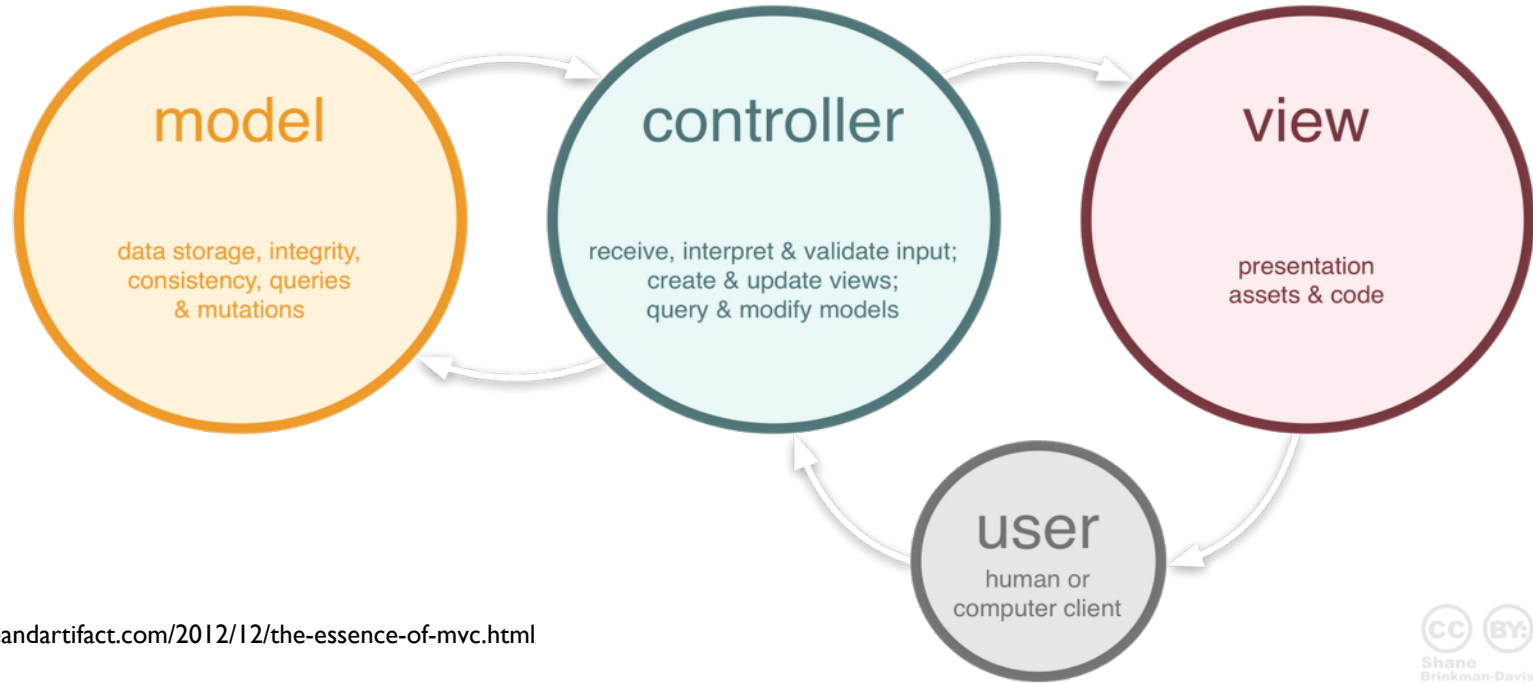
Model View Controller:

Is a way of life

## MODEL VIEW CONTROLLER

Model View Controller:

But actually it's a software design pattern specifically for web apps



---

## MODEL VIEW CONTROLLER

---

Model View Controller:

### **Model**

- Responsible for managing the data
- It's a database essentially!

### **View**

- Presents the data / app
- Responsible for design / user experience

### **Controller**

- Responds to user input and performs operations based on it
- Eg. User inputs a number of neighbors and the controller trains the model

## MODEL VIEW CONTROLLER

---

Model View Controller:

### Model

- Responsible for managing the data
- It's a database essentially!

### View

- Presents the data / app
- Responsible for design / user experience

### Controller

- Responds to user input and performs operations based on it
- Eg. User inputs a number of neighbors and the controller trains the model

#### QUESTION:

Which ones are handled by

Back end developers?

Front end?

## MODEL VIEW CONTROLLER

---

Model View Controller:

### **Model (Backend)**

- Responsible for managing the data
- It's a database essentially!

### **View (Frontend)**

- Presents the data / app
- Responsible for design / user experience

### **Controller (Backend / Frontend)**

- Responds to user input and performs operations based on it
- Eg. User inputs a number of neighbors and the controller trains the model

#### **QUESTION:**

Which ones are handled by

Back end developers?

Front end?

---

**DEPLOYING KNN**

---

**DEPLOYING KNN**



# Sample Flask App

[https://github.com/ghego/iris\\_calculator](https://github.com/ghego/iris_calculator)

Notice we have:

1. Models
2. Views (called templates)
3. Controller (controller.py)

# Sample Flask App

[https://github.com/ghego/iris\\_calculator](https://github.com/ghego/iris_calculator)

Go ahead and clone it

NOT IN YOUR OTHER GIT  
REPOSITORY 😊

# Sample Flask App

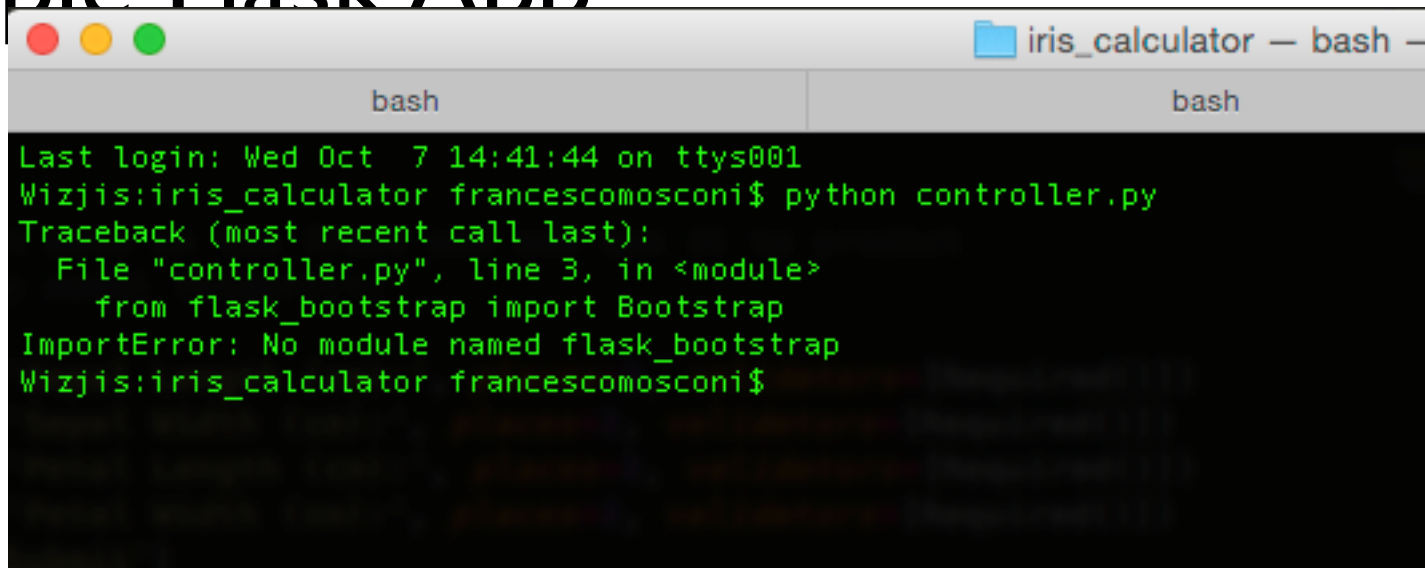
[https://github.com/ghego/iris\\_calculator](https://github.com/ghego/iris_calculator)

To run locally, go to root and run

**python controller.py**

Go to <http://127.0.0.1:5000/>

# Sample Flask App

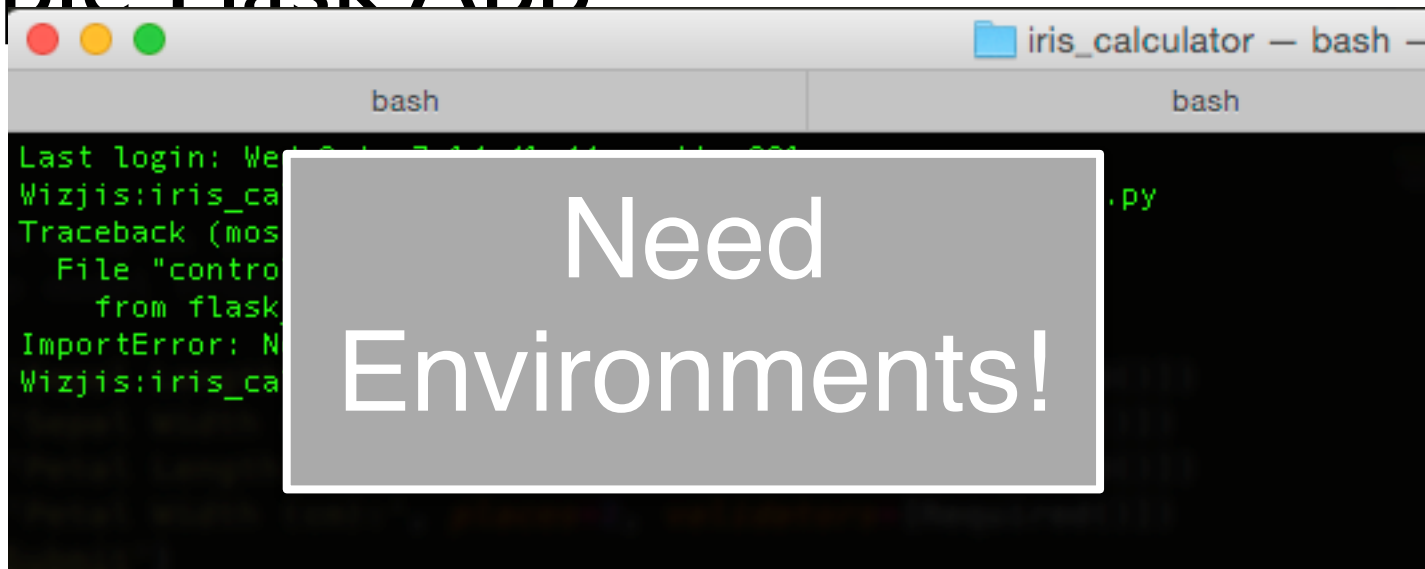


A terminal window titled 'iris\_calculator — bash' with two tabs labeled 'bash'. The terminal output shows a successful login followed by an attempt to run 'python controller.py'. This results in a 'Traceback (most recent call last):' error. The error indicates that in the file 'controller.py', line 3, there is an attempt to import 'Bootstrap' from 'flask\_bootstrap', but an 'ImportError' occurs because 'No module named flask\_bootstrap'.

```
Last login: Wed Oct 7 14:41:44 on ttys001
Wizjis:iris_calculator francescosconioni$ python controller.py
Traceback (most recent call last):
  File "controller.py", line 3, in <module>
    from flask_bootstrap import Bootstrap
ImportError: No module named flask_bootstrap
Wizjis:iris_calculator francescosconioni$
```

Go to <http://127.0.0.1:5000/>

# Sample Flask App



A terminal window titled 'iris\_calculator — bash' is shown. The terminal output includes the following text:

```
Last login: Wed Oct 26 11:11:11 UTC 2011
Wizjis:iris_calculator$ python app.py
Traceback (most recent call last):
  File "control.py", line 1, in <module>
    from flask import Flask
ImportError: No module named flask
Wizjis:iris_calculator$
```

Overlaid on the terminal is a grey rectangular box with a white border containing the text:

Need Environments!

Go to <http://127.0.0.1:5000/>

## Aside: Python environments

A python project has dependencies  
e.g....

## Aside: Python environments

A python project has dependencies

e.g....

pandas

numpy

scikit-learn

Q: How does one keep track of dependencies and make sure that a program runs on a different platform?



Q: How does one keep track of dependencies and make sure that a program runs on a different platform?

A: Environments!

Q: How does one keep track of dependencies and make sure that a program runs on a different platform?

A: Environments!

Go ahead and read here:

<http://conda.pydata.org/docs/using/envs.html>

[http://pip.readthedocs.org/en/stable/  
reference/pip\\_freeze/](http://pip.readthedocs.org/en/stable/reference/pip_freeze/)

How do I create a new clean environment that only contains the following packages:

- python
- numpy1.8.1
- scikit-learn0.15.2
- scipy0.14.0

How do I create a new clean environment that only contains the following packages:

- python
- numpy1.8.1
- scikit-learn0.15.2
- scipy0.14.0

```
conda create -n test_env python numpy=1.8.1 scikit-learn=0.15.2 scipy=0.14.0
```

How do I activate the environment?

How do I de-activate the environment?

How do I delete the environment?

**How do I activate the environment?**

```
source activate test_env
```

**How do I de-activate the environment?**

```
source deactivate
```

**How do I delete the environment?**

```
conda remove -n test_env --all
```

# Let's get back: Sample Flask App

[https://github.com/ghego/iris\\_calculator](https://github.com/ghego/iris_calculator)

## NOTE:

You may not have the required modules to run it right now..

If not, run

```
pip install -r  
requirements_clean.txt
```

To run locally, go to root and run

**python controller.py**

Go to <http://127.0.0.1:5000/>



We have two forms

The top form **trains the model**

The bottom form **predicts incoming data**

When we submit the data....

We have two forms

The top form **trains the model**

The bottom form **predicts incoming data**

When we submit the data....

**QUESTION:**

Which part of MVC  
handles the input?


We have two forms

The top form **trains the model**

The bottom form **predicts incoming data**

When we submit the data the **controller** handles it!

## Please fill out the form below!

Number of Neighbors:  

Paramter 2 (Optional)

Paramter 3 (Optional)

Paramter 4 (Optional)

Sepal Length (cm):

Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

**Note:**

Note we included 3 optional parameters in case you need to train on more than just one!

The machine learning model lives in the **model** folder  
not to be confused with the model in MVC

It is pickled...



The machine learning model lives in the **model** folder  
not to be confused with the model in MVC

It is pickled...

You know, the standard mechanism  
for serializing an object.

Essentially we can transform a  
python object into a file!



You can pickle anything!!

1. sklearn models
2. Jsons!
3. Strings!
4. Your own models



# Comprehensive Step by Step

1. Sign up for Heroku [here](#)
2. Create a new app (make sure Heroku toolbelt is installed)
3. Clone our flask app [here](#)
  1. Change and test at will
4. Run the command: **heroku git:remote -a <APP>**
5. Install the custom build back for scipy and numpy
  1. `heroku config:set BUILDPACK_URL=https://github.com/thenovices/heroku-buildpack-scipy --app <APP>`
  2. Run the command above in the root of your app (with the toolbelt installed)
6. Use as normal git repository:
  1. Git add, commit, etc...
  2. Git push heroku master (instead of origin)
7. Amaze people with your live KNN



# 1. SIGN UP FOR HEROKU

Self-explanatory?

<http://heroku.com>

## 2. CREATE A NEW APP (MAKE SURE HEROKU TOOLBELT IS INSTALLED)

Not self-explanatory

<https://toolbelt.heroku.com/>

Type into your console:

`heroku login`

## 3. CLONE OUR FLASK APP [HERE](#)

Self-explanatory?

Now you can run it locally!! (remember run `python controller.py`)

## 4. RUN THE COMMAND: HEROKU CREATE

At the root of the directory

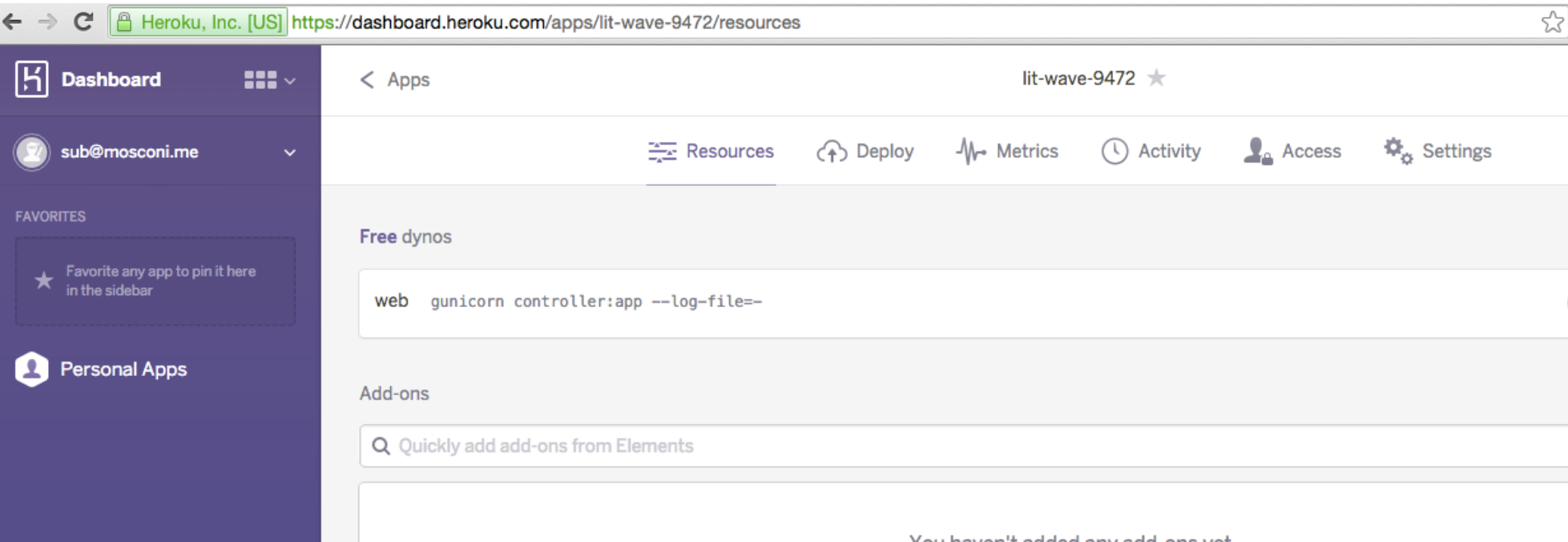
This adds a new git “remote”

Essentially a new place to push 😊

Check this by running *git remote -v*

You should see origin and heroku

## 4. RUN THE COMMAND: HEROKU CREATE



The screenshot shows the Heroku dashboard for the application 'lit-wave-9472'. The browser address bar shows the URL <https://dashboard.heroku.com/apps/lit-wave-9472/resources>. The left sidebar contains the 'Dashboard' link, the user 'sub@mosconi.me', and a 'FAVORITES' section with a tip to pin apps. The main content area has a top navigation bar with links for 'Apps', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The 'Resources' tab is active, displaying 'Free dynos' and a single dyno running the command `web gunicorn controller:app --log-file=-`. Below this is an 'Add-ons' section with a search bar and the text 'Quickly add add-ons from Elements'. At the bottom, a message states 'You haven't added any add-ons yet'.

Heroku, Inc. [US] <https://dashboard.heroku.com/apps/lit-wave-9472/resources>

Dashboard sub@mosconi.me

FAVORITES

★ Favorite any app to pin it here in the sidebar

Personal Apps

< Apps lit-wave-9472 ★

Resources Deploy Metrics Activity Access Settings

Free dynos

web gunicorn controller:app --log-file=-

Add-ons

🔍 Quickly add add-ons from Elements

You haven't added any add-ons yet

---

## 5. INSTALL THE CUSTOM BUILD BACK FOR SCIPY AND NUMPY

Not self-explanatory

<https://github.com/thenovices/heroku-buildpack-sciipy>

**Run:**

```
heroku config:set BUILDPACK_URL=https://github.com/thenovices/heroku-buildpack-sciipy - -  
app <APP>
```

**At the root of the directory**

---

## 6. USE AS NORMAL GIT REPOSITORY

Self-explanatory?

`git add .`

`git commit -m "I am a genius"`

`Git push heroku master`

## 6. USE AS NORMAL GIT REPOSITORY

Self-explanatory?

`git add .`

`git commit -m "I am a genius"`

`Git push heroku master`

**Note:**

It is installing a bunch of modules because of the requirements.txt file



# 7. AMAZE PEOPLE WITH YOUR LIVE KNN

Self-explanatory!!!!





**Note:**

Your unique website will have your app name instead of fm-iris

<https://fm-iris.herokuapp.com/>

# WHAT NOW?!?!?!1?

**Put in your own machine learning model!**

- 1. Build your model else where**
- 2. Load it into the model folder manually**