

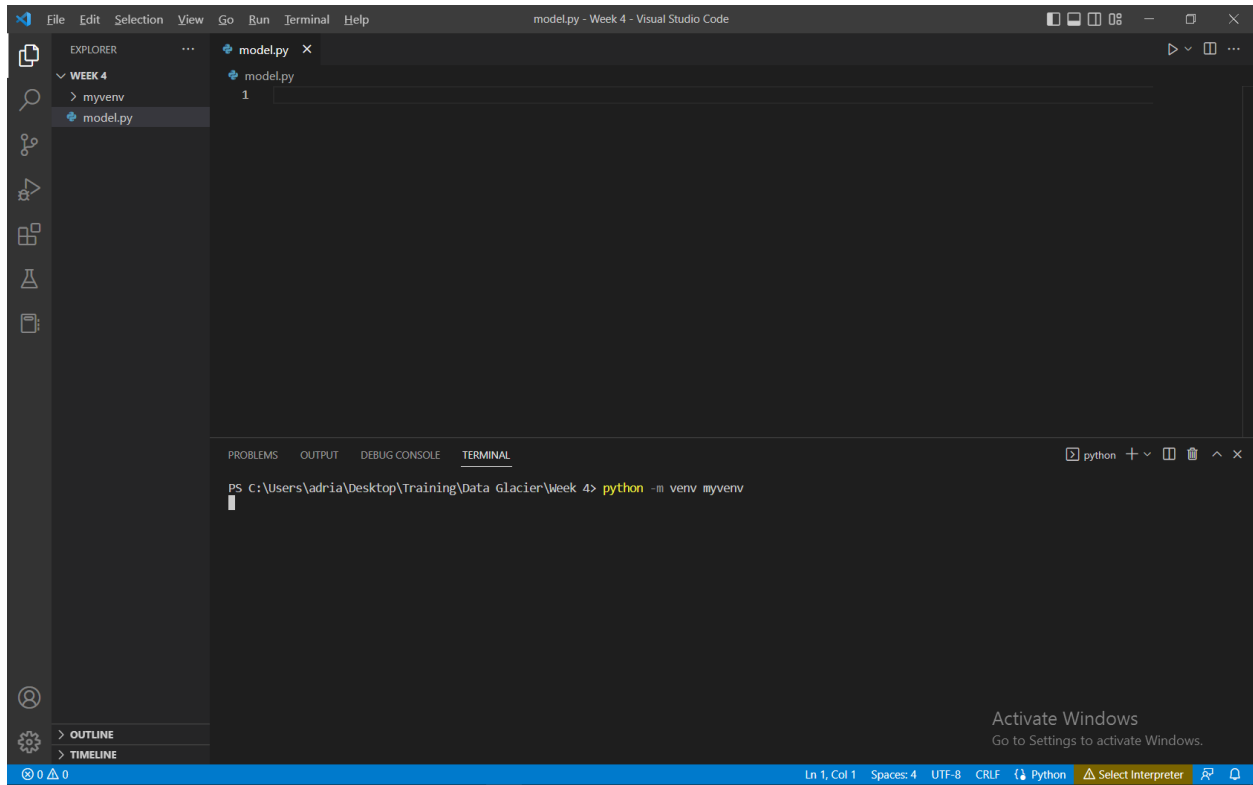
Name: Adrian Baysa

Batch Code: LISUM16

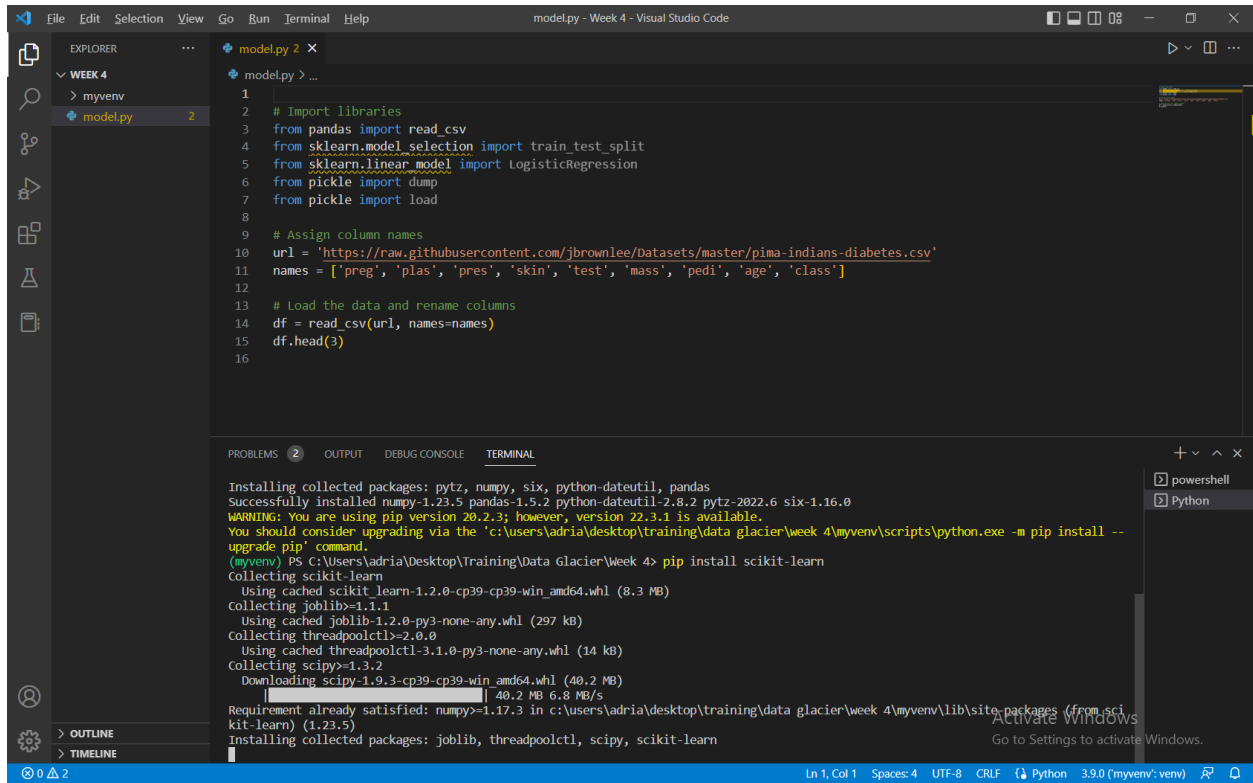
Submission Date: 13 December 2022

Submitted to: n/a

Create environment



Install libraries



The screenshot shows the Visual Studio Code interface with a Python file named `model.py` and its terminal output.

Python Script (`model.py`):

```
1  
2 # Import libraries  
3 from pandas import read_csv  
4 from sklearn.model_selection import train_test_split  
5 from sklearn.linear_model import LogisticRegression  
6 from pickle import dump  
7 from pickle import load  
8  
9 # Assign column names  
10 url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.csv'  
11 names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']  
12  
13 # Load the data and rename columns  
14 df = read_csv(url, names=names)  
15 df.head(3)  
16
```

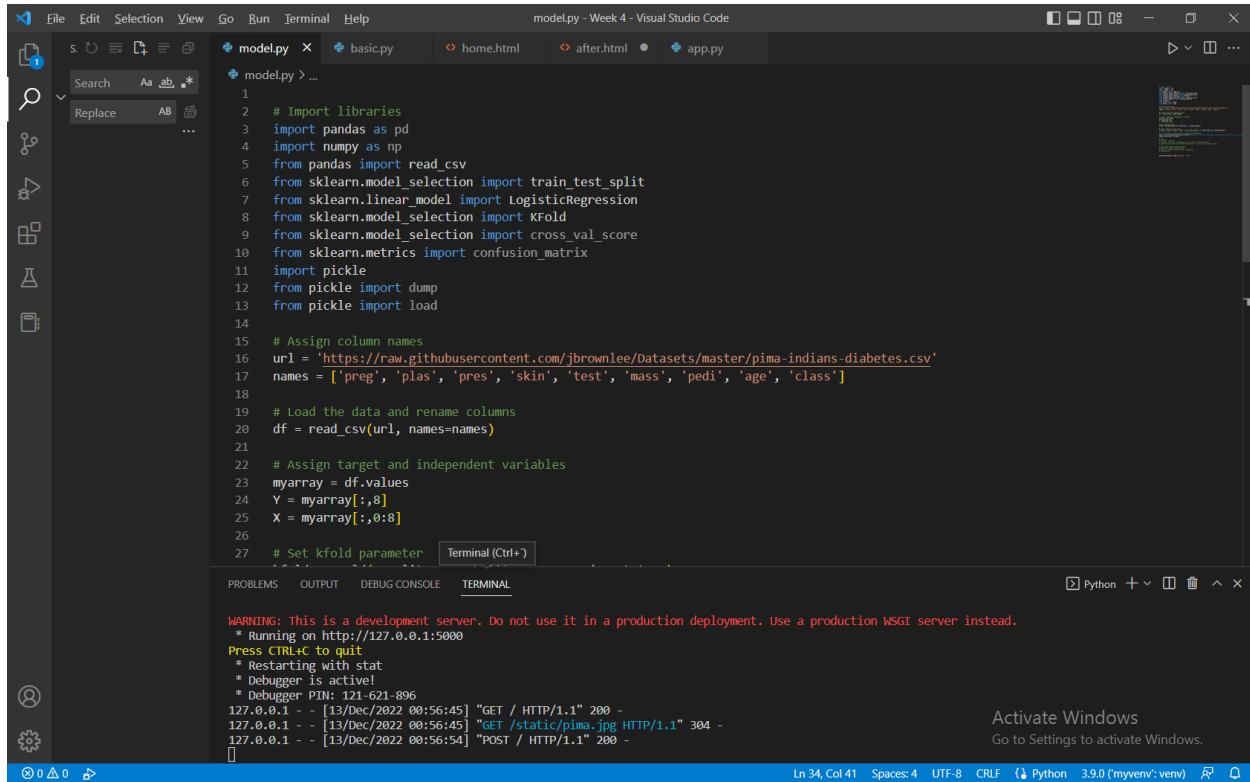
Terminal Output:

```
Installing collected packages: pytz, numpy, six, python-dateutil, pandas  
Successfully installed numpy-1.23.5 pandas-1.5.2 python-dateutil-2.8.2 pytz-2022.6 six-1.16.0  
WARNING: You are using pip version 20.2.3; however, version 22.3.1 is available.  
You should consider upgrading via the 'c:\users\adria\desktop\training\data glacier\week 4\myenv\scripts\python.exe -m pip install --  
upgrade pip' command.  
(myenv) PS c:\Users\adria\Desktop\Training\Data Glacier\Week 4> pip install scikit-learn  
Collecting scikit-learn  
  Using cached scikit_learn-1.2.0-cp39-cp39-win_amd64.whl (8.3 MB)  
Collecting joblib>=1.1.1  
  Using cached joblib-1.2.0-py3-none-any.whl (297 kB)  
Collecting threadpoolctl>=2.0.0  
  Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)  
Collecting scipy>=1.3.2  
  Downloading scipy-1.9.3-cp39-cp39-win_amd64.whl (40.2 MB)  
    40.2 MB 6.8 MB/s  
Requirement already satisfied: numpy>=1.17.3 in c:\users\adria\desktop\training\data glacier\week 4\myenv\lib\site-packages (from sci  
kit-learn) (1.23.5)  
Installing collected packages: joblib, threadpoolctl, scipy, scikit-learn
```

The status bar at the bottom indicates the file is at Line 1, Column 1, using UTF-8 encoding, CRLF line endings, and the Python 3.9.0 interpreter from the `myenv` virtual environment.

Create Model:

- Import libraries
- Download the dataset
- Identify X and y (target variable)
- Set kfold parameter
- Create train and test data
- Fit the model (Logistic Regression)
- Dump the model as a pickle file



The screenshot shows the Visual Studio Code interface. The main editor window displays a Python file named `model.py` with the following code:

```
1
2 # Import libraries
3 import pandas as pd
4 import numpy as np
5 from pandas import read_csv
6 from sklearn.model_selection import train_test_split
7 from sklearn.linear_model import LogisticRegression
8 from sklearn.model_selection import KFold
9 from sklearn.model_selection import cross_val_score
10 from sklearn.metrics import confusion_matrix
11 import pickle
12 from pickle import dump
13 from pickle import load
14
15 # Assign column names
16 url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.csv'
17 names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
18
19 # Load the data and rename columns
20 df = read_csv(url, names=names)
21
22 # Assign target and independent variables
23 myarray = df.values
24 Y = myarray[:,8]
25 X = myarray[:,0:8]
26
27 # Set kfold parameter
```

The terminal window at the bottom shows the output of a Python script running a web server:

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 121-621-896
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -
127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -
```

The status bar at the bottom indicates the file is at line 34, column 41, with 4 spaces, UTF-8 encoding, and CRLF line endings. The Python environment is 3.9.0 (myenv: venv).

```
14
15 # Assign column names
16 url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.csv'
17 names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
18
19 # Load the data and rename columns
20 df = read_csv(url, names=names)
21
22 # Assign target and independent variables
23 myarray = df.values
24 Y = myarray[:,8]
25 X = myarray[:,0:8]
26
27 # Set kfold parameter
28 kfold = KFold(n_splits=10, shuffle=True, random_state=7)
29
30 # Create test and train split
31 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.33, random_state=7)
32
33 # Fit the model on train dataset using LogisticRegression
34 model = LogisticRegression(max_iter=1000)
35 model.fit(X_train, Y_train)
36
37 # Score
38 # scoring = 'accuracy'
39 # results = cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
40 # print("Accuracy Mean and Standard Deviation:", results.mean(), results.std())
41
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on http://127.0.0.1:5000

Press CTRL+C to quit

* Restarting with stat

* Debugger is active!

* Debugger PIN: 121-621-896

127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -

127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -

```
27 # Set kfold parameter
28 kfold = KFold(n_splits=10, shuffle=True, random_state=7)
29
30 # Create test and train split
31 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.33, random_state=7)
32
33 # Fit the model on train dataset using LogisticRegression
34 model = LogisticRegression(max_iter=1000)
35 model.fit(X_train, Y_train)
36
37 # Score
38 # scoring = 'accuracy'
39 # results = cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
40 # print("Accuracy Mean and Standard Deviation:", results.mean(), results.std())
41
42 # Test Predict and confusion_matrix
43 # predicted = model.predict(X_test)
44 # matrix = confusion_matrix(Y_test, predicted)
45 # print(matrix)
46
47
48 pickle.dump(model, open('pima.pkl', 'wb'))
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on http://127.0.0.1:5000

Press CTRL+C to quit

* Restarting with stat

* Debugger is active!

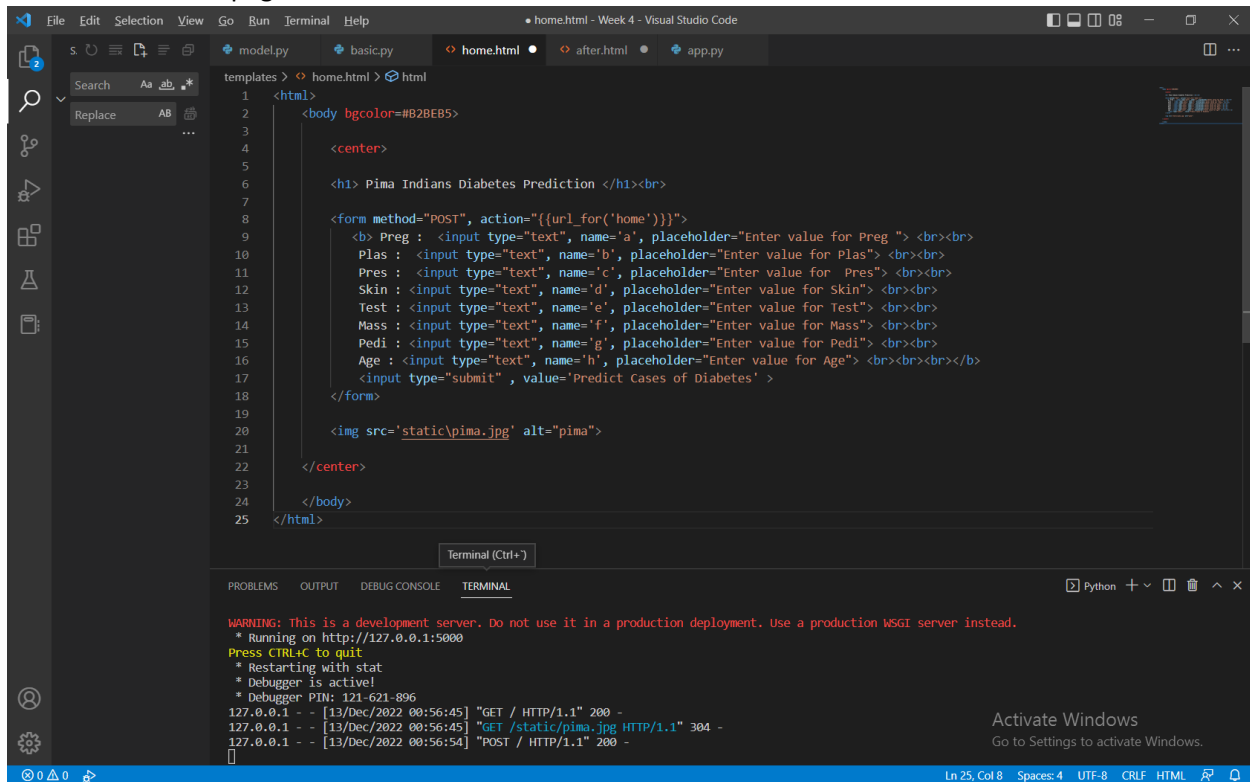
* Debugger PIN: 121-621-896

127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -

127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -

Create home.html page

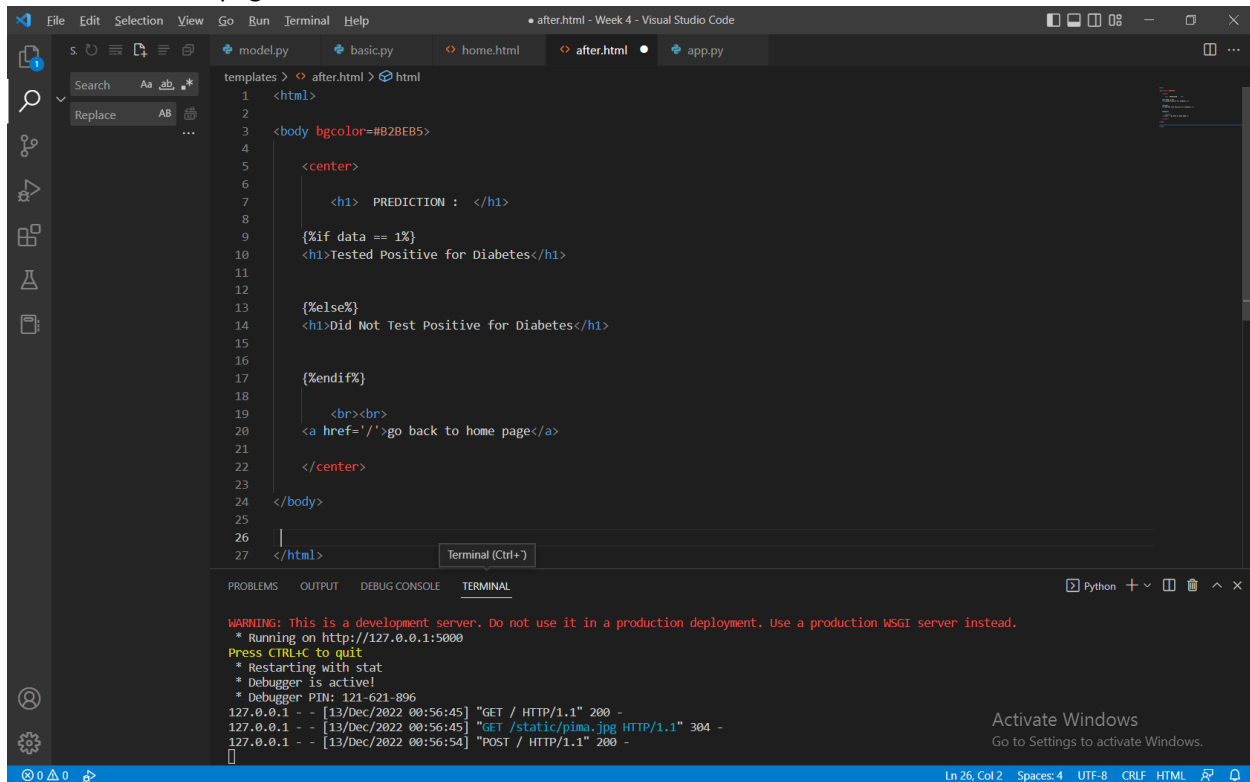


The screenshot shows the Visual Studio Code editor with the 'home.html' file open. The file contains HTML code for a web page titled 'Pima Indians Diabetes Prediction'. The code includes a form with input fields for 'Preg', 'Plas', 'Pres', 'Skin', 'Test', 'Mass', 'Pedi', and 'Age', along with a 'submit' button. The form is styled with a background color of #B2BEB5 and is centered. Below the form, there is an image placeholder for 'pima.jpg'. The terminal at the bottom shows a warning message about using a development server and a list of log messages indicating the server is running on http://127.0.0.1:5000.

```
1 <html>
2 <body bgcolor=#B2BEB5>
3
4 <center>
5
6 <h1> Pima Indians Diabetes Prediction </h1><br>
7
8 <form method="POST", action="{{url_for('home')}}">
9 <b> Preg : <input type="text", name='a', placeholder="Enter value for Preg "> <br><br>
10 Plas : <input type="text", name='b', placeholder="Enter value for Plas"> <br><br>
11 Pres : <input type="text", name='c', placeholder="Enter value for Pres"> <br><br>
12 Skin : <input type="text", name='d', placeholder="Enter value for Skin"> <br><br>
13 Test : <input type="text", name='e', placeholder="Enter value for Test"> <br><br>
14 Mass : <input type="text", name='f', placeholder="Enter value for Mass"> <br><br>
15 Pedi : <input type="text", name='g', placeholder="Enter value for Pedi"> <br><br>
16 Age : <input type="text", name='h', placeholder="Enter value for Age"> <br><br><br></b>
17 <input type="submit" , value='Predict Cases of Diabetes' >
18 </form>
19
20 <img src='static/pima.jpg' alt="pima">
21
22 </center>
23
24 </body>
25 </html>
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 121-621-896
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -
127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -

Create after.html page

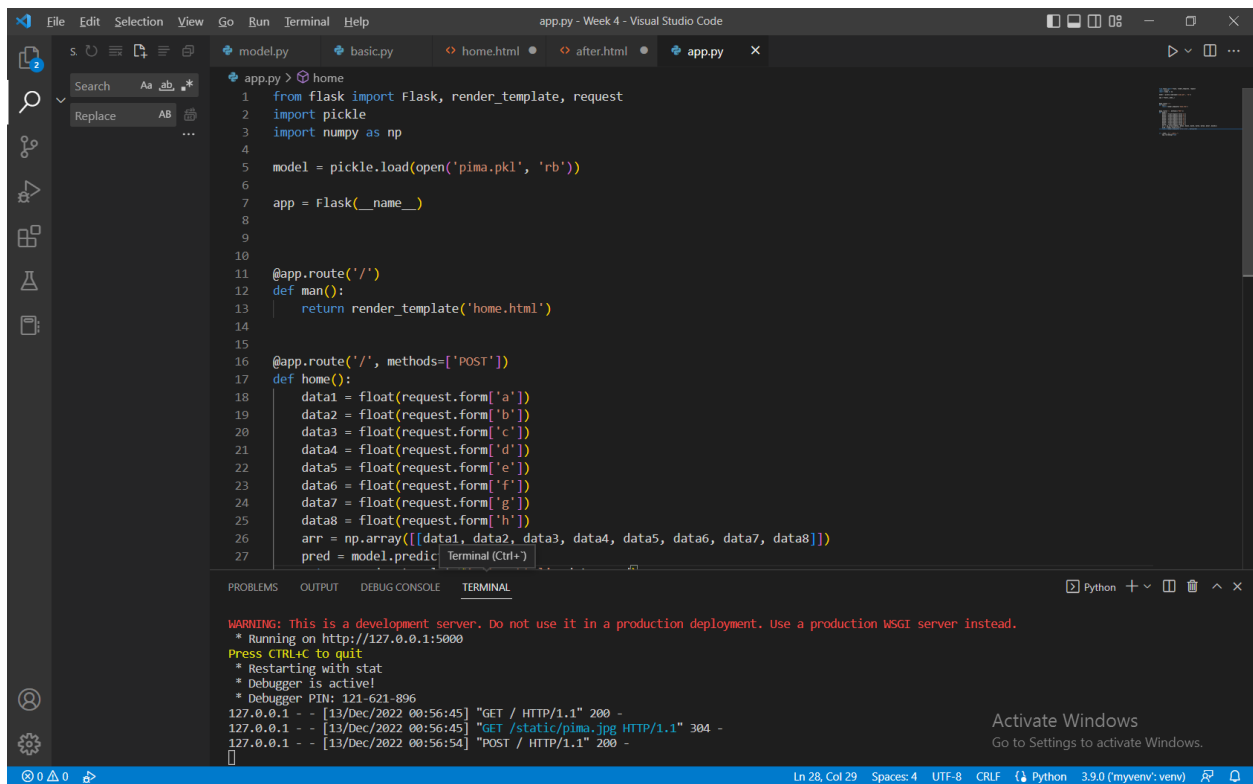


The screenshot shows the Visual Studio Code editor with the 'after.html' file open. The file contains HTML code for a web page titled 'PREDICTION :'. The code includes a conditional statement that checks if the data is 1% and displays 'Tested Positive for Diabetes' or 'Did Not Test Positive for Diabetes'. The page is styled with a background color of #B2BEB5 and is centered. Below the text, there is a link to go back to the home page. The terminal at the bottom shows a warning message about using a development server and a list of log messages indicating the server is running on http://127.0.0.1:5000.

```
1 <html>
2
3 <body bgcolor=#B2BEB5>
4
5 <center>
6
7 <h1> PREDICTION : </h1>
8
9 {%if data == 1%}
10 <h1>Tested Positive for Diabetes</h1>
11
12
13 {%else%}
14 <h1>Did Not Test Positive for Diabetes</h1>
15
16
17 {%endif%}
18
19 <br><br>
20 <a href='/'>go back to home page</a>
21
22 </center>
23
24 </body>
25
26
27 </html>
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 121-621-896
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -
127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -

Create app.py

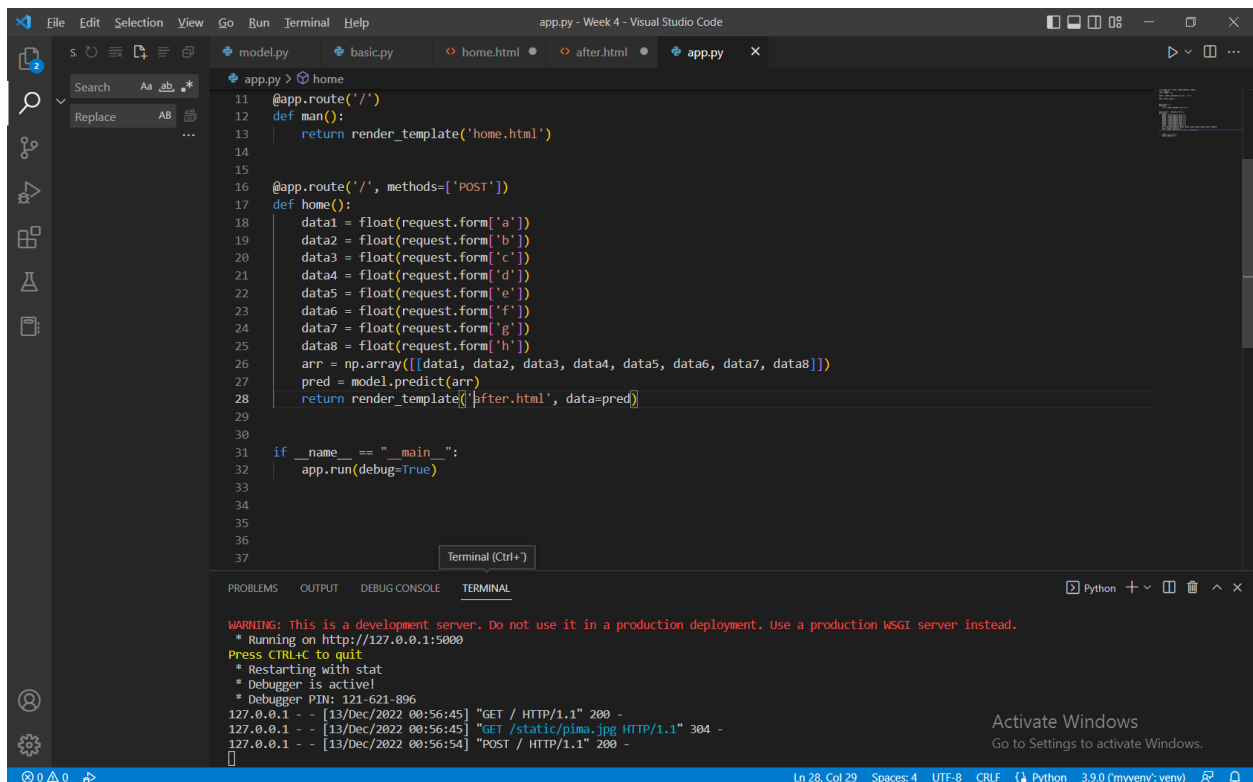


The screenshot shows the Visual Studio Code editor with a file named `app.py` open. The code is a Flask application that loads a pre-trained model and serves a home page. The terminal shows the application running on `http://127.0.0.1:5000` and receiving several requests.

```
1 from flask import Flask, render_template, request
2 import pickle
3 import numpy as np
4
5 model = pickle.load(open('pima.pkl', 'rb'))
6
7 app = Flask(__name__)
8
9
10
11 @app.route('/')
12 def man():
13     return render_template('home.html')
14
15
16 @app.route('/', methods=['POST'])
17 def home():
18     data1 = float(request.form['a'])
19     data2 = float(request.form['b'])
20     data3 = float(request.form['c'])
21     data4 = float(request.form['d'])
22     data5 = float(request.form['e'])
23     data6 = float(request.form['f'])
24     data7 = float(request.form['g'])
25     data8 = float(request.form['h'])
26     arr = np.array([data1, data2, data3, data4, data5, data6, data7, data8])
27     pred = model.predict(arr)
```

Terminal output:

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 121-621-896
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -
127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -
```



The screenshot shows the Visual Studio Code editor with the `app.py` file updated. The code now includes a `home` route that takes form data, predicts the outcome, and renders the `after.html` template. The terminal shows the application running and receiving requests.

```
11 @app.route('/')
12 def man():
13     return render_template('home.html')
14
15
16 @app.route('/', methods=['POST'])
17 def home():
18     data1 = float(request.form['a'])
19     data2 = float(request.form['b'])
20     data3 = float(request.form['c'])
21     data4 = float(request.form['d'])
22     data5 = float(request.form['e'])
23     data6 = float(request.form['f'])
24     data7 = float(request.form['g'])
25     data8 = float(request.form['h'])
26     arr = np.array([data1, data2, data3, data4, data5, data6, data7, data8])
27     pred = model.predict(arr)
28     return render_template('after.html', data=pred)
29
30
31 if __name__ == "__main__":
32     app.run(debug=True)
33
34
35
36
37
```

Terminal output:

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 121-621-896
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Dec/2022 00:56:45] "GET /static/pima.jpg HTTP/1.1" 304 -
127.0.0.1 - - [13/Dec/2022 00:56:54] "POST / HTTP/1.1" 200 -
```

Test app.py

Search re... x Datasets/... x Home Pa... x Untitled4... x Create ve... x deploy-m... x machine l... x python -... x 127.0.0.1... x +

127.0.0.1:5000

Pima Indians Diabetes Prediction

Preg :

Plas :

Pres :


Skin :

Test :

Mass :

Pedi :

Age :



Activate Windows
Go to Settings to activate Windows.

Enter values:

Search re... x Datasets/... x Home Pa... x Untitled4... x Create ve... x deploy-m... x machine l... x python -... x 127.0.0.1... x +

127.0.0.1:5000

Pima Indians Diabetes Prediction

Preg :

Plas :

Pres :


Skin :

Test :

Mass :

Pedi :

Age :

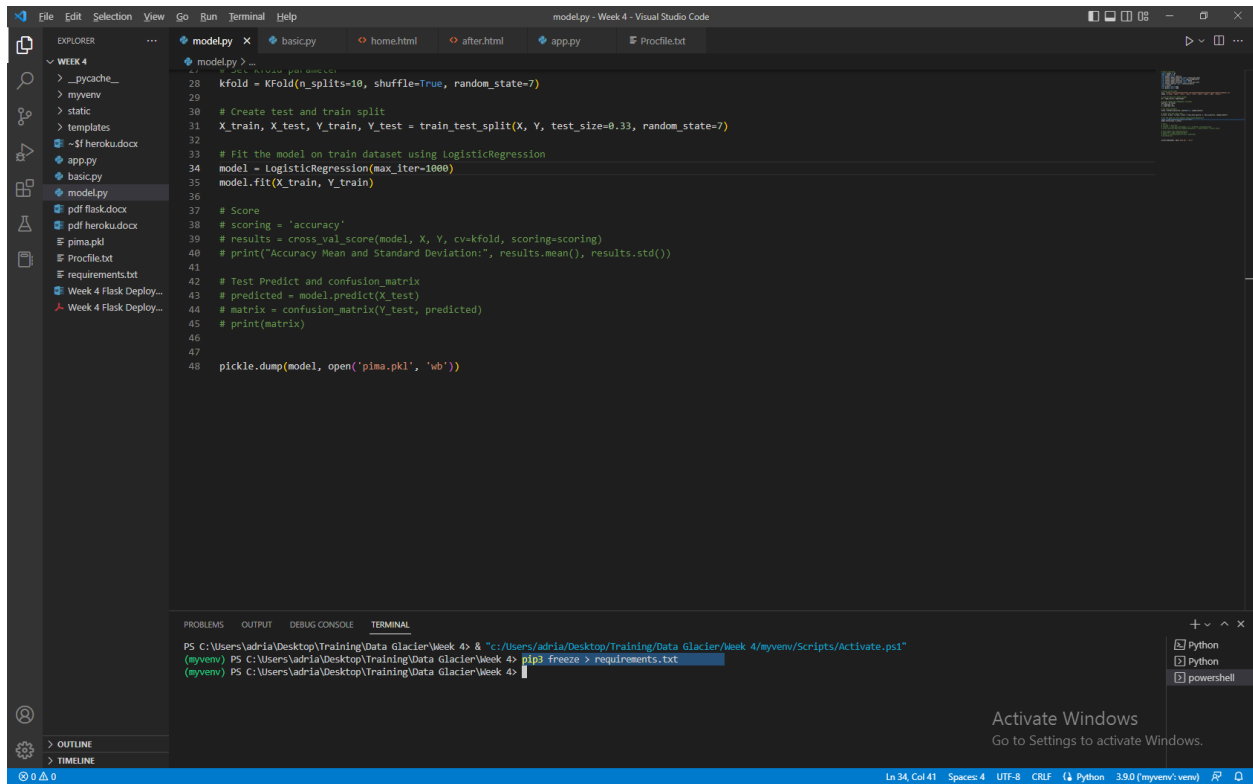


Activate Windows
Go to Settings to activate Windows.

See prediction results:



Create requirement.txt file



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files for 'WEEK 4', including `requirements.txt`. The main editor displays a Python script `model.py` with the following code:

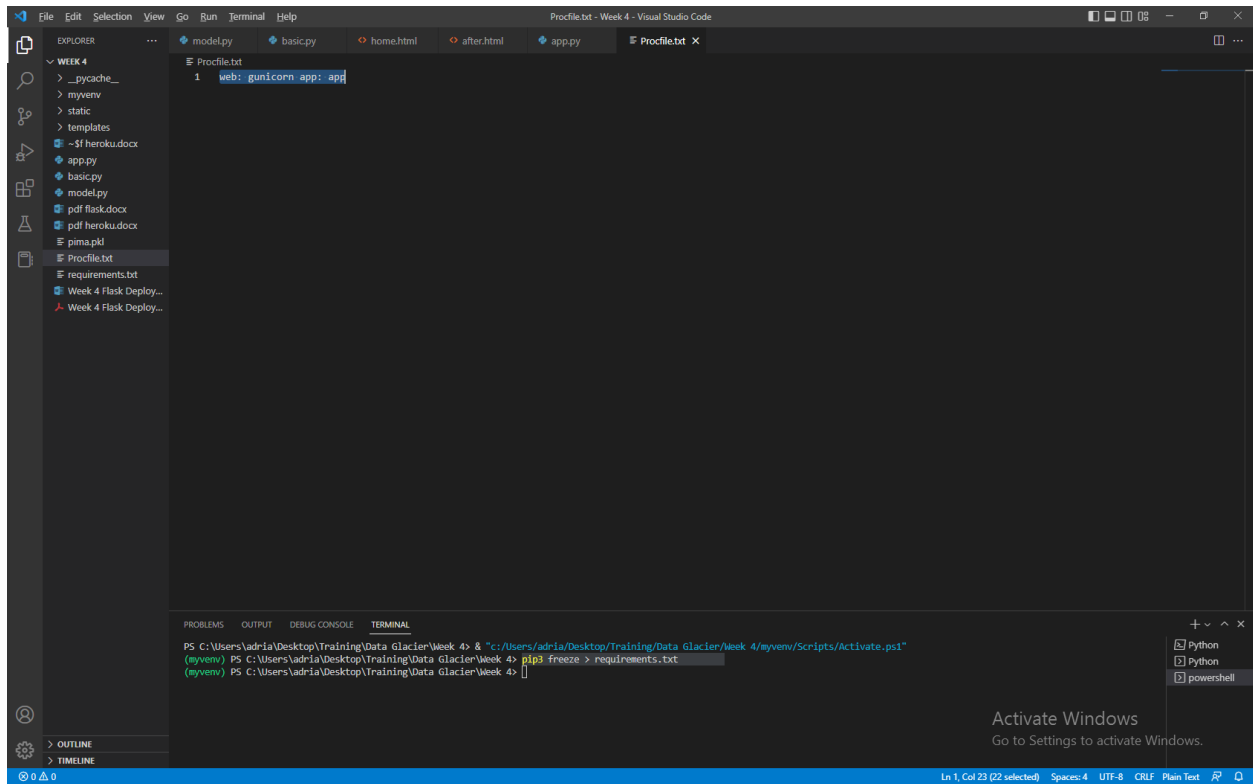
```
28 kfold = KFold(n_splits=10, shuffle=True, random_state=7)
29
30 # Create test and train split
31 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.33, random_state=7)
32
33 # Fit the model on train dataset using LogisticRegression
34 model = LogisticRegression(max_iter=1000)
35 model.fit(X_train, Y_train)
36
37 # Score
38 # scoring = 'accuracy'
39 # results = cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
40 # print("Accuracy Mean and Standard Deviation:", results.mean(), results.std())
41
42 # Test Predict and confusion_matrix
43 # predicted = model.predict(X_test)
44 # matrix = confusion_matrix(Y_test, predicted)
45 # print(matrix)
46
47
48 pickle.dump(model, open('pima.pkl', 'wb'))
```

The bottom panel shows the TERMINAL with the following commands and output:

```
PS C:\Users\adria\Desktop\Training\Data Glacier\Week 4> & "C:/Users/adria/Desktop/Training/Data Glacier/Week 4/myenv/Scripts/Activate.ps1"
(myenv) PS C:\Users\adria\Desktop\Training\Data Glacier\Week 4> pip3 freeze > requirements.txt
(myenv) PS C:\Users\adria\Desktop\Training\Data Glacier\Week 4>
```

An 'Activate Windows' watermark is visible in the bottom right corner of the terminal area.

Create Profile



Create repository in Github. Upload files and commit

The screenshot shows a GitHub repository page for 'agbaysa/dataglacier_week5'. The repository is public and has 0 stars, 1 watching, and 0 forks. The main branch is 'main' with 1 branch and 0 tags. The repository contains 8 commits. The files listed are: static, templates, Procfile, README.md, app.py, basic.py, model.py, pima.pkl, and requirements.txt. The README.md file is selected, showing the repository name 'dataglacier_week5' and the description 'Week 5 Cloud and API deployment'. The right sidebar shows the 'About' section with a README link, 'Releases' section with a 'Create a new release' link, 'Packages' section with a 'Publish your first package' link, and 'Environments' section with four inactive environments: pima-diabetes-api, pima-diabetes-api3, pima-api1, and pima-diabetes-api2. The 'Languages' section shows Python at 99.1% and Procfile at 0.9%.

agbaysa / dataglacier_week5 Public

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file + Code +

agbaysa Add files via upload fcf8a1c 1 minute ago 8 commits

static	Add files via upload	1 minute ago
templates	Add files via upload	1 minute ago
Procfile	Update Procfile	8 minutes ago
README.md	Initial commit	7 hours ago
app.py	Add files via upload	1 hour ago
basic.py	Add files via upload	1 hour ago
model.py	Add files via upload	1 hour ago
pima.pkl	Add files via upload	1 hour ago
requirements.txt	Update requirements.txt	39 minutes ago

README.md

dataglacier_week5

Week 5 Cloud and API deployment

About

Week 5 Cloud and API deployment

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Environments 4

- pima-diabetes-api Inactive
- pima-diabetes-api3 Inactive
- pima-api1 Inactive
- pima-diabetes-api2 Inactive

Languages

Activate Windows

Go to Settings to activate Windows.

Python 99.1% Procfile 0.9%

Login to Heroku. Select “New” and “Create New App”

The screenshot shows the Heroku dashboard in a web browser. The address bar displays 'dashboard.heroku.com/apps'. The page header includes the Heroku logo and a search bar with the text 'Jump to Favorites, Apps, Pipelines, Spaces...'. Below the header, there is a navigation bar with a 'Personal' dropdown menu and a 'New' button. A blue notification banner states: 'Free Heroku Postgres, free Heroku Data for Redis', and free Heroku Dynos are no longer available. If you have apps using these resources, you must upgrade to our [new low-cost](#) or other paid plans to ensure your apps continue to run and retain your data. To recover your data, [contact Heroku Support](#) as soon as possible. Eligible students can apply for platform credits through our [Heroku for GitHub Students program](#). [Learn more](#)

Below the notification, there is a search bar labeled 'Filter apps and pipelines'. A list of apps is displayed, with the first app being 'herofoo1'. To the right of the app name, it says 'heroku-20 - United States' and a star icon.

At the bottom of the page, there is a footer with links to 'heroku.com', 'Blogs', 'Careers', 'Documentation', and 'Support'. On the right side of the footer, there is a 'Terms of Service' link, a 'Privacy' link, a 'Cookies' link, and a copyright notice '© 2022 Salesforce.com'. A 'Activate Windows' watermark is visible in the bottom right corner.

Create app name

The screenshot shows the Heroku dashboard for the 'pima-diabetes-api' app. The browser address bar shows the URL 'dashboard.heroku.com/apps/pima-diabetes-api/deploy/heroku-git'. The Heroku logo is in the top left, and a search bar is in the top right. The navigation bar includes 'Personal' and 'pima-diabetes-api'. The main content area is divided into two columns. The left column is titled 'Add this app to a pipeline' and contains instructions on how to create a new pipeline or choose an existing one. The right column is titled 'Add this app to a stage in a pipeline to enable additional features' and contains instructions on how to connect multiple apps together and promote code between them. Below these instructions is a 'Choose a pipeline' dropdown menu. The 'Deployment method' section is visible, showing options for 'Heroku Git', 'GitHub', and 'Container Registry'. The 'Deploy using Heroku Git' section is expanded, showing instructions on how to use Git in the command line or a GUI tool to deploy this app. The 'Install the Heroku CLI' section is also visible, showing instructions on how to download and install the Heroku CLI. The 'Create a new Git repository' section is visible, showing instructions on how to initialize a git repository in a new or existing directory. The 'Deploy your application' section is visible, showing instructions on how to commit your code to the repository and deploy it to Heroku using Git. An 'Activate Windows' watermark is visible in the bottom right corner.

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

Personal > pima-diabetes-api

Overview Resources **Deploy** Metrics Activity Access Settings

Add this app to a pipeline
Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features
Pipelines let you connect multiple apps together and **promote code** between them. [Learn more](#)
Pipelines connected to GitHub can enable **review apps** and create apps for new pull requests. [Learn more](#)

Choose a pipeline

Deployment method

Heroku Git
Use Heroku CLI

GitHub
Connect to GitHub

Container Registry
Use Heroku CLI

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 pima-diabetes-api
```

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

Activate Windows
Go to Settings to activate Windows.

Connect to Github

Heroku

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

Personal > pima-diabetes-api

Overview Resources Deploy Metrics Activity Access Settings

Add this app to a pipeline
Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features
Pipelines let you connect multiple apps together and promote code between them. [Learn more](#)
Pipelines connected to GitHub can enable review apps, and create apps for new pull requests. [Learn more](#)

Choose a pipeline

Deployment method

Heroku Git Use Heroku CLI | GitHub Connect to GitHub | Container Registry Use Heroku CLI

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 pima-diabetes-api
```

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

Activate Windows
Go to Settings to activate Windows.

<https://dashboard.heroku.com/apps/pima-diabetes-api/deploy/github>

Heroku

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

Personal > pima-diabetes-api

Overview Resources Deploy Metrics Activity Access Settings

Add this app to a pipeline
Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features
Pipelines let you connect multiple apps together and promote code between them. [Learn more](#)
Pipelines connected to GitHub can enable review apps, and create apps for new pull requests. [Learn more](#)

Choose a pipeline

Deployment method

Heroku Git Use Heroku CLI | GitHub Connect to GitHub | Container Registry Use Heroku CLI

Connect to GitHub
Connect this app to GitHub to enable code diffs and deploys.

View your code diffs on GitHub
Connect your app to a GitHub repository to see commit diffs in the activity log.

Deploy changes with GitHub
Connecting to a repository will allow you to deploy a branch to your app.

Automatic deploys from GitHub
Select a branch to deploy automatically whenever it is pushed to.

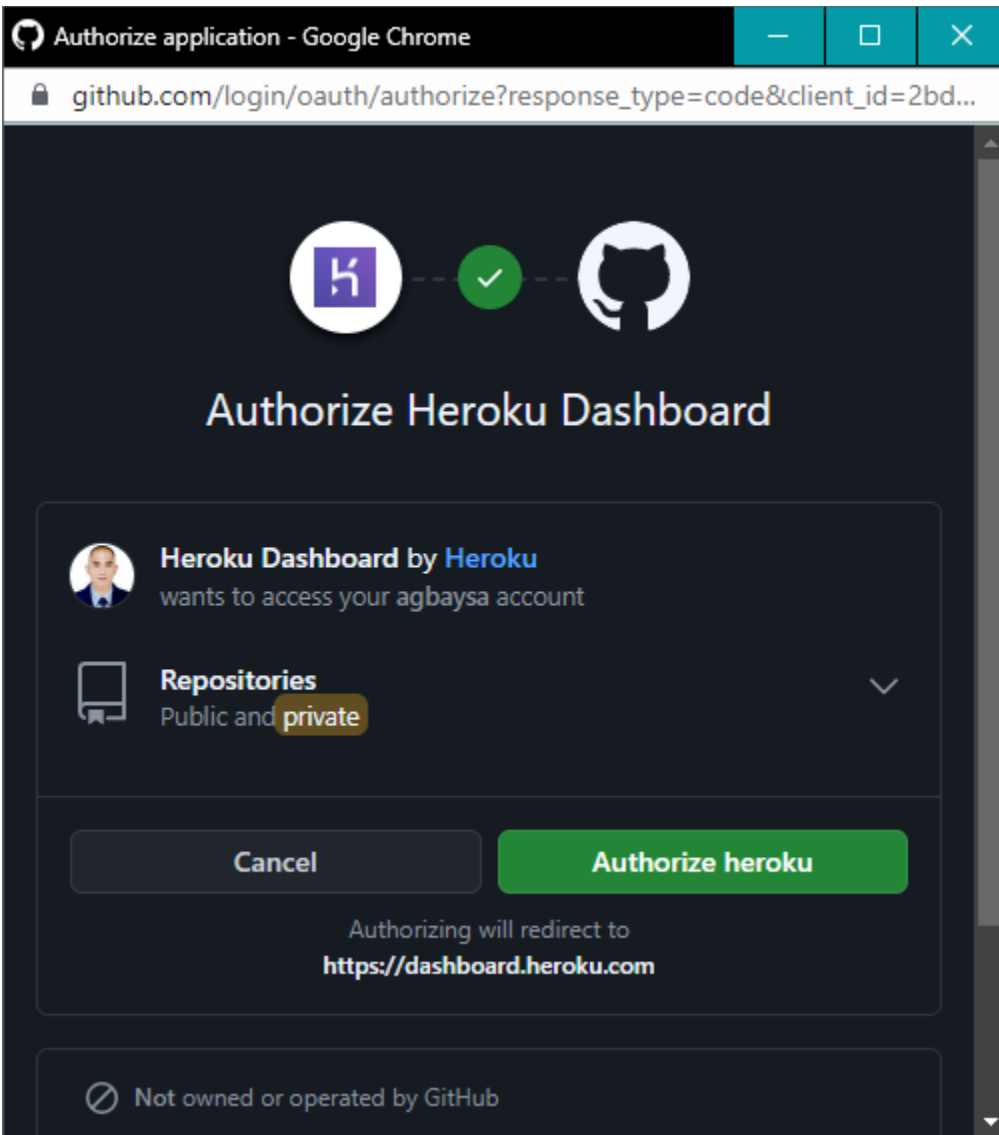
Create review apps in pipelines
Pipelines connected to GitHub can enable review apps, and create apps for new pull requests. [Learn more](#)

[Connect to GitHub](#)

Activate Windows
Go to Settings to activate Windows.

heroku.com Blogs Careers Documentation [Support](#)

Terms of Service Privacy Cookies © 2022 Salesforce.com



Select repository and connect

The screenshot shows the Heroku dashboard for the application 'pima-diabetes-api'. The 'Deployment method' section is active, showing options for Heroku Git, GitHub, and Container Registry. The 'Connect to GitHub' section is expanded, displaying a search bar and a list of repositories. The repository 'agbaysa/dataglacier_week5' is selected, and the 'Connect' button is highlighted.

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

agbaysa repo-name Search

Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

agbaysa / VC	Connect
agbaysa / Python-Code	Connect
agbaysa / portfolio	Connect
agbaysa / barnacles	Connect
agbaysa / sklearn	Connect
agbaysa / books	Connect
agbaysa / dataglacier_week5	Connect
agbaysa / eggnog	Connect
agbaysa / 1vb_scoring_io_052722	Connect
agbaysa / mdtacles	Connect
agbaysa / mudtacles	Connect
agbaysa / github-slideshow	Connect
agbaysa / heroku	Connect
agbaysa / pedal	Connect

Deploy branch

Heroku

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

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to agbaysa/dataglacier_week5 by agbaysa

Releases in the activity feed link to GitHub to view commit diffs

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

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

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. Deploys happen automatically: be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more

Choose a branch to deploy

main

Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more

Choose a branch to deploy

main

Deploy Branch

Activate Windows

Go to Settings to activate Windows.

heroku.com Blogs Careers Documentation Support

Terms of Service Privacy Cookies © 2022 Salesforce.com

View the deployed app and test

Browser tabs: pima-api1 - GitHub | https://pima-api1.h... | The Procfile | Heroku | agbaysa/dataglac... | [Heroku][Deploy]... | Add-ons - Heroku E... | Logging | Heroku D... | python - H14 error |

Address bar: pima-api1.herokuapp.com

Pima Indians Diabetes Prediction

Preg :

Plas :

Pres :


Skin :

Test :

Mass :

Pedi :

Age :



Activate Windows
Go to Settings to activate Windows.

Test and see prediction

Pima Indians Diabetes Prediction

Preg :

Plas :

Pres :


Skin :

Test :

Mass :

Pedi :

Age :



Activate Windows
Go to Settings to activate Windows.

PREDICTION :

Tested Positive for Diabetes

[go back to home page](#)

Activate Windows
Go to Settings to activate Windows.

