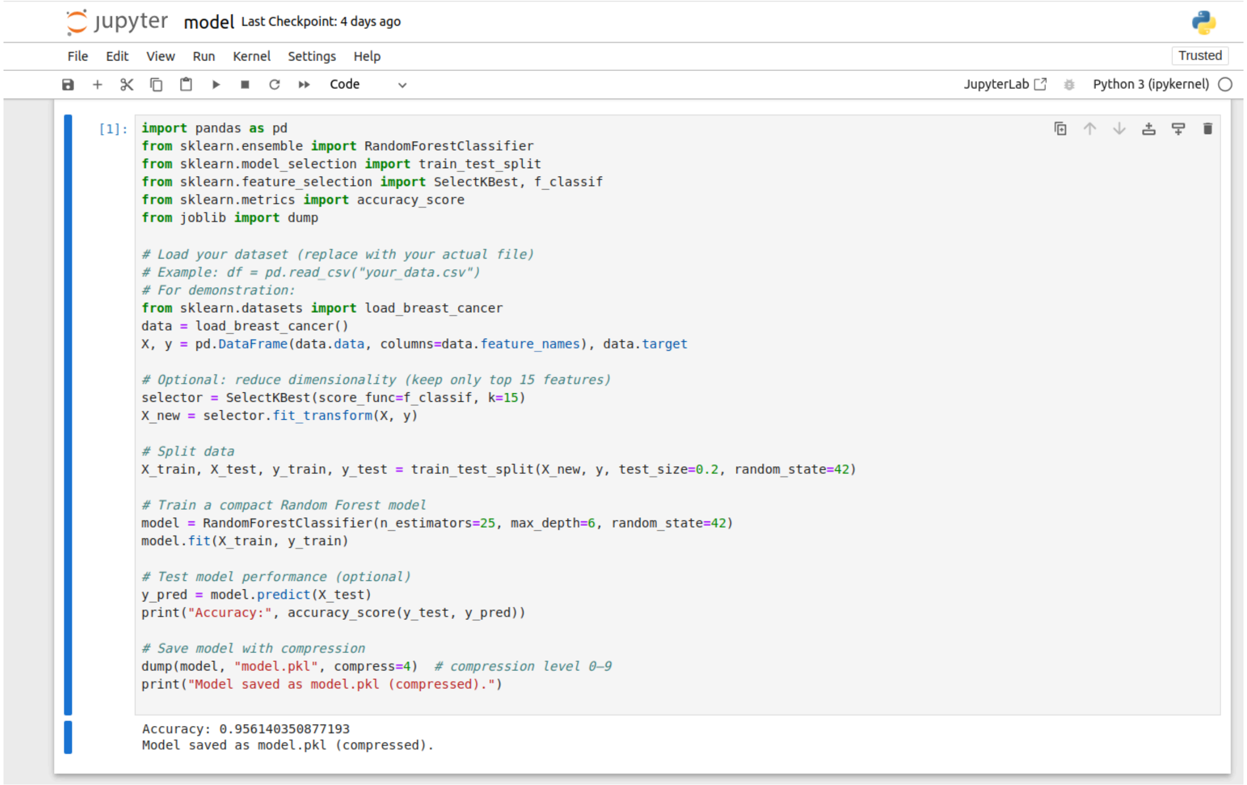
Name: Priyanjali Patel

Batch Code: LISUM45

Submission Date : 8 June 2025

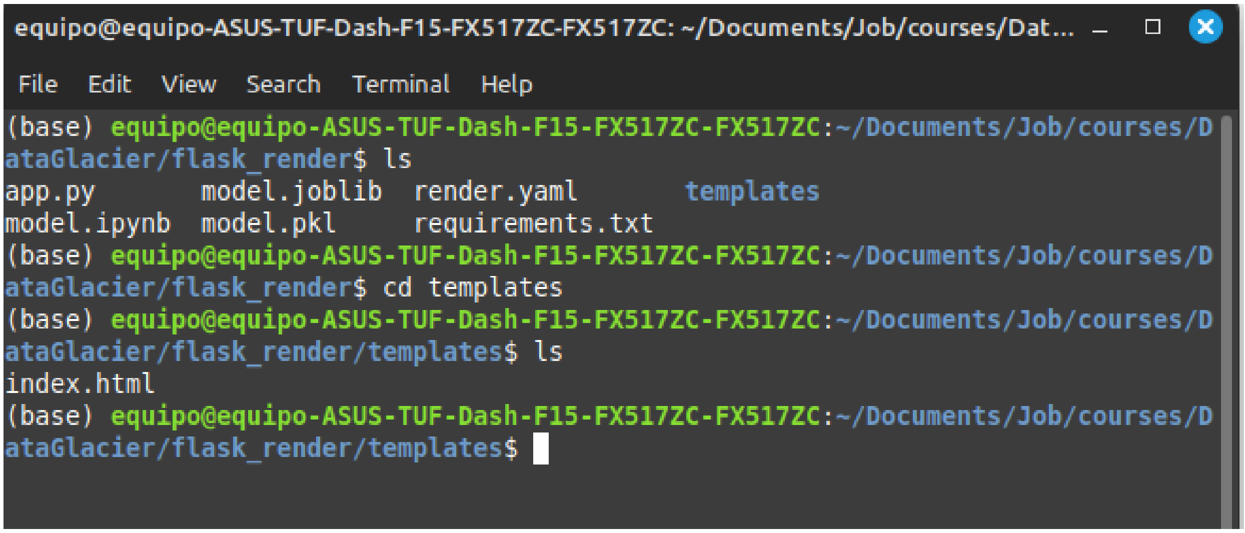
- **Prepare the model (Using Toy Data)**



```python
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.metrics import accuracy_score
from joblib import dump

# Load your dataset (replace with your actual file)
# Example: df = pd.read_csv("your_data.csv")
# For demonstration:
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X, y = pd.DataFrame(data.data, columns=data.feature_names), data.target

# Optional: reduce dimensionality (keep only top 15 features)
selector = SelectKBest(score_func=f_classif, k=15)
X_new = selector.fit_transform(X, y)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.2, random_state=42)

# Train a compact Random Forest model
model = RandomForestClassifier(n_estimators=25, max_depth=6, random_state=42)
model.fit(X_train, y_train)

# Test model performance (optional)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))

# Save model with compression
dump(model, "model.pkl", compress=4)  # compression level 0–9
print("Model saved as model.pkl (compressed).")
```

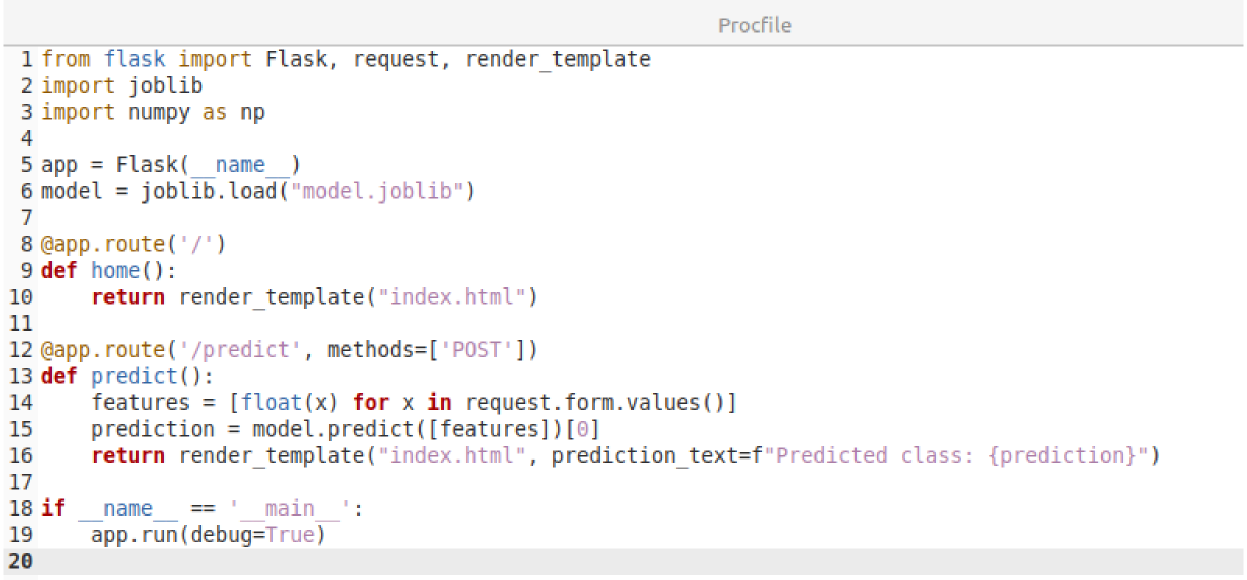
```
Accuracy: 0.956140350877193
Model saved as model.pkl (compressed).
```

- **Project Structure**


equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC: ~/Documents/Job/courses/Dat...
File Edit View Search Terminal Help

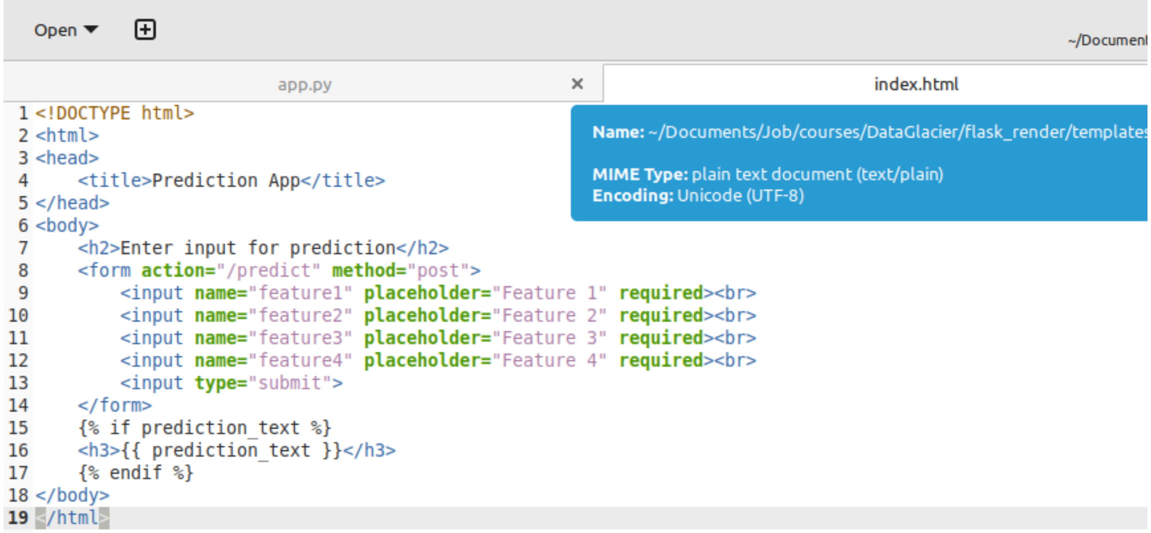
```
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/D
ataGlacier/flask_render$ ls
app.py        model.joblib  render.yaml        templates
model.ipynb   model.pkl     requirements.txt
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/D
ataGlacier/flask_render$ cd templates
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/D
ataGlacier/flask_render/templates$ ls
index.html
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/D
ataGlacier/flask_render/templates$ 
```

1. app.py



```python
from flask import Flask, request, render_template
import joblib
import numpy as np

app = Flask(__name__)
model = joblib.load("model.joblib")

@app.route('/')
def home():
    return render_template("index.html")

@app.route('/predict', methods=['POST'])
def predict():
    features = [float(x) for x in request.form.values()]
    prediction = model.predict([features])[0]
    return render_template("index.html", prediction_text=f"Predicted class: {prediction}")

if __name__ == '__main__':
    app.run(debug=True)

```

2. templates/index.html

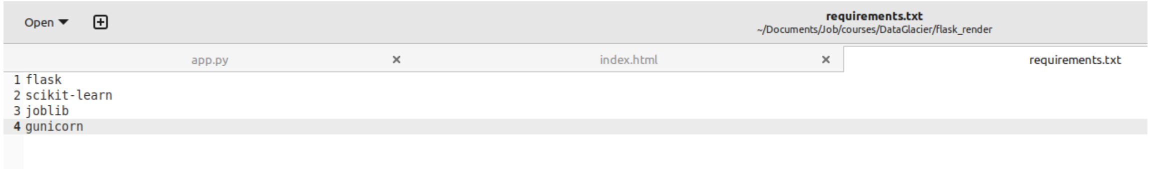


Name: ~/Documents/Job/courses/DataGlacier/flask_render/templates

MIME Type: plain text document (text/plain)
Encoding: Unicode (UTF-8)

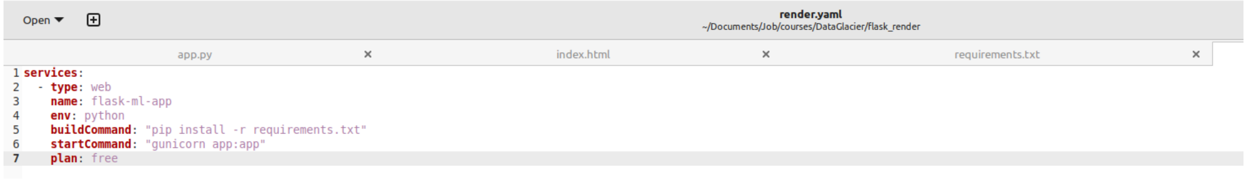
```
<!DOCTYPE html>
<html>
<head>
    <title>Prediction App</title>
</head>
<body>
    <h2>Enter input for prediction</h2>
    <form action="/predict" method="post">
        <input name="feature1" placeholder="Feature 1" required><br>
        <input name="feature2" placeholder="Feature 2" required><br>
        <input name="feature3" placeholder="Feature 3" required><br>
        <input name="feature4" placeholder="Feature 4" required><br>
        <input type="submit">
    </form>
    {% if prediction_text %}
    <h3>{{ prediction_text }}</h3>
    {% endif %}
</body>
</html>
```

5. requirements.txt



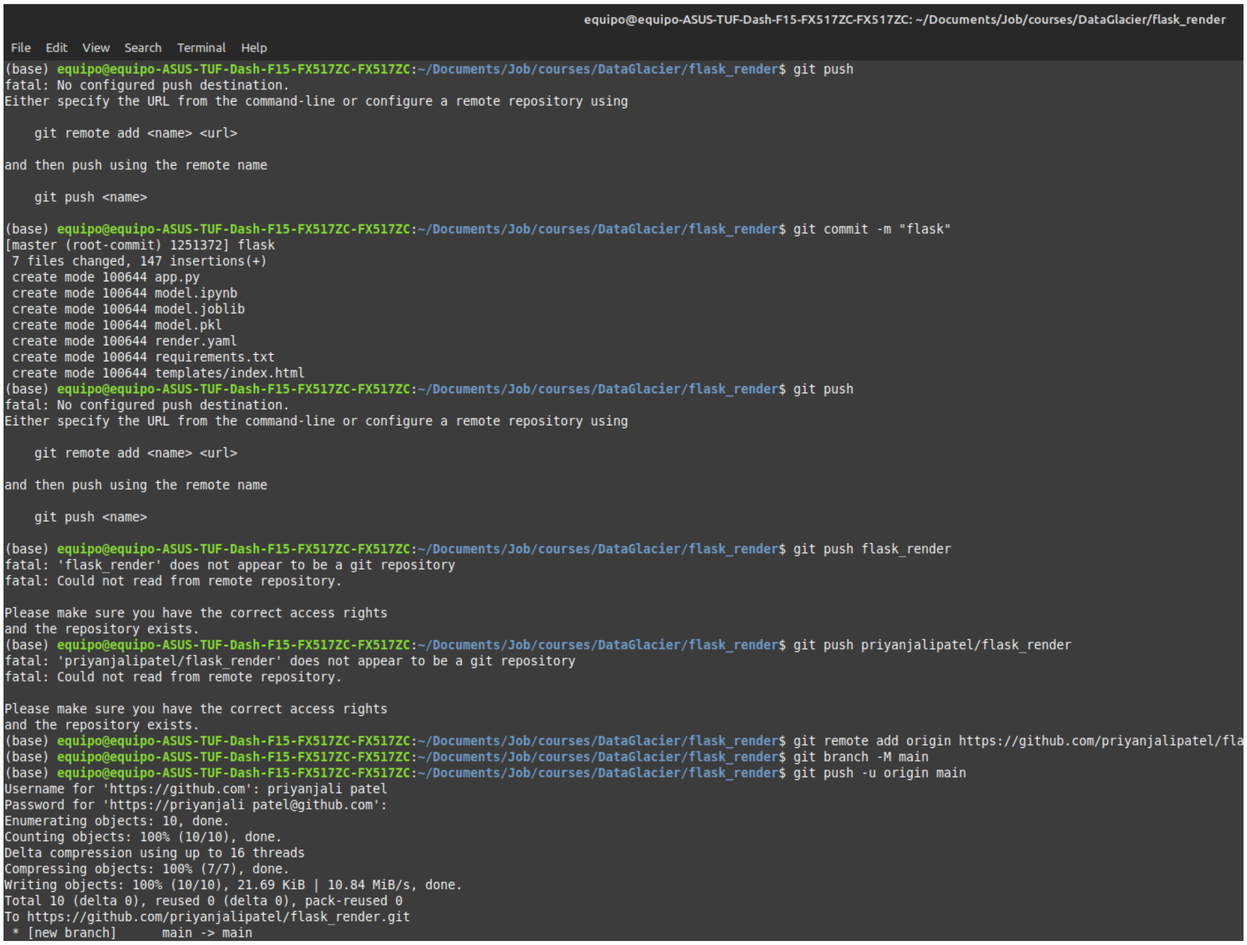
```
flask
scikit-learn
joblib
gunicorn
```

6. render.yaml



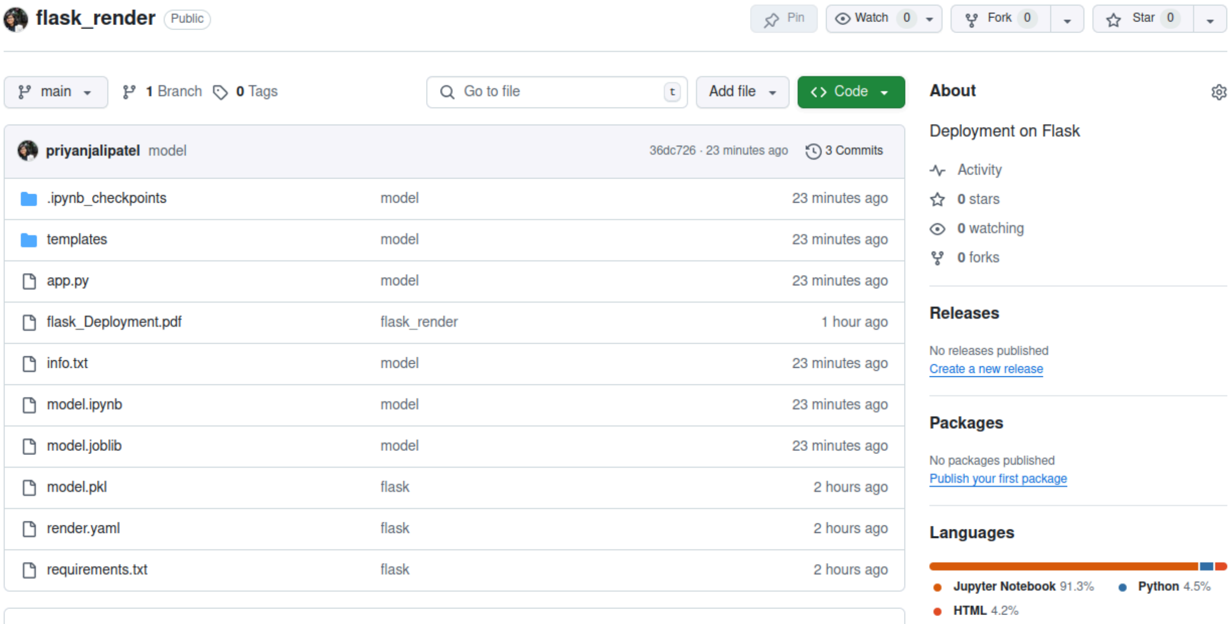
```
services:
  - type: web
    name: flask-ml-app
    env: python
    buildCommand: "pip install -r requirements.txt"
    startCommand: "gunicorn app:app"
    plan: free
```

7. Push to GitHub



equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC: ~/Documents/Job/courses/DataGlacier/flask_render
File Edit View Search Terminal Help

```
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>

(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git commit -m "flask"
[master (root-commit) 1251372] flask
 7 files changed, 147 insertions(+)
 create mode 100644 app.py
 create mode 100644 model.ipynb
 create mode 100644 model.joblib
 create mode 100644 model.pkl
 create mode 100644 render.yaml
 create mode 100644 requirements.txt
 create mode 100644 templates/index.html
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>

(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git push flask_render
fatal: 'flask_render' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git push priyanjalipatel/flask_render
fatal: 'priyanjalipatel/flask_render' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git remote add origin https://github.com/priyanjalipatel/fla
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git branch -M main
(base) equipo@equipo-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Documents/Job/courses/DataGlacier/flask_render$ git push -u origin main
Username for 'https://github.com': priyanjali patel
Password for 'https://priyanjali patel@github.com':
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 21.69 KiB | 10.84 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/priyanjalipatel/flask_render.git
 * [new branch]      main -> main
```



priyanjalipatel model 36dc726 · 23 minutes ago 3 Commits

.ipynb_checkpoints	model	23 minutes ago
templates	model	23 minutes ago
app.py	model	23 minutes ago
flask_Deployment.pdf	flask_render	1 hour ago
info.txt	model	23 minutes ago
model.ipynb	model	23 minutes ago
model.joblib	model	23 minutes ago
model.pkl	flask	2 hours ago
render.yaml	flask	2 hours ago
requirements.txt	flask	2 hours ago

About

Deployment on Flask

Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

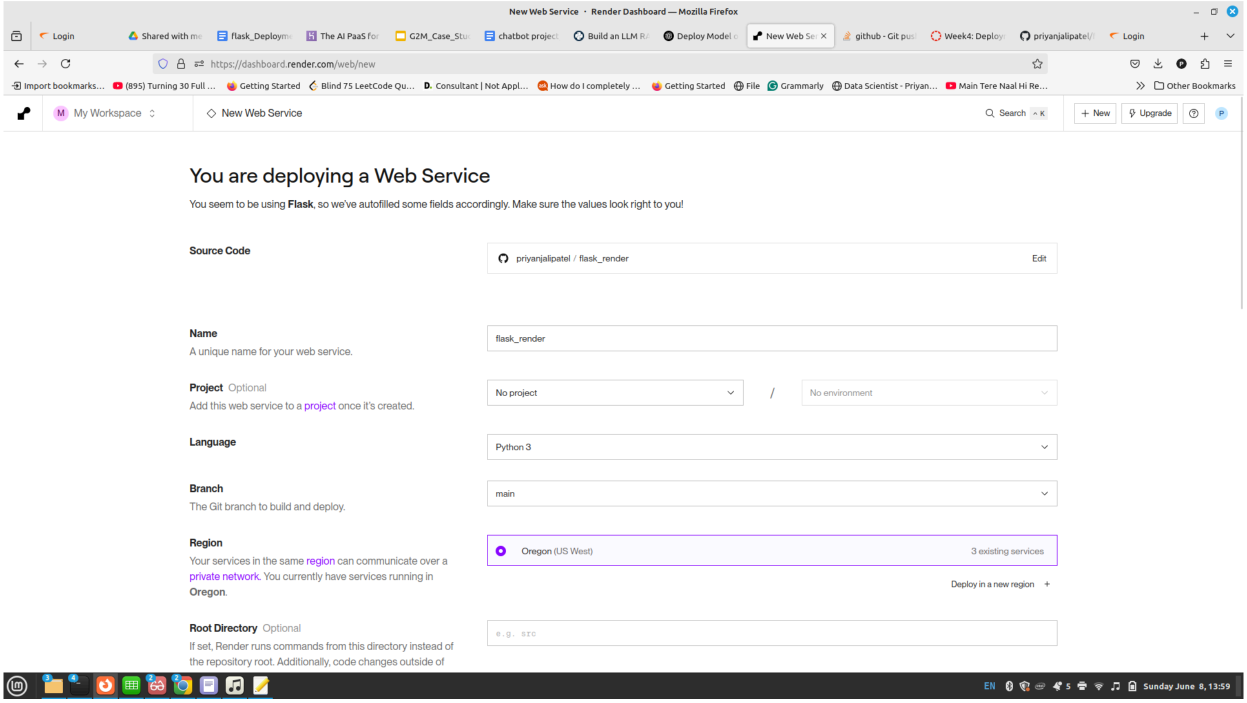
Languages

Jupyter Notebook 91.3% Python 4.5%
HTML 4.2%

- **Deploy on Render (https://dashboard.render.com/)**

Use settings:

- Environment: Python
- Build Command: pip install -r requirements.txt
- Start Command: gunicorn app:app
- Free Plan



You are deploying a Web Service

You seem to be using **Flask**, so we've autofilled some fields accordingly. Make sure the values look right to you!

Source Code: priyanjalipatel / flask_render Edit

Name: flask_render
A unique name for your web service.

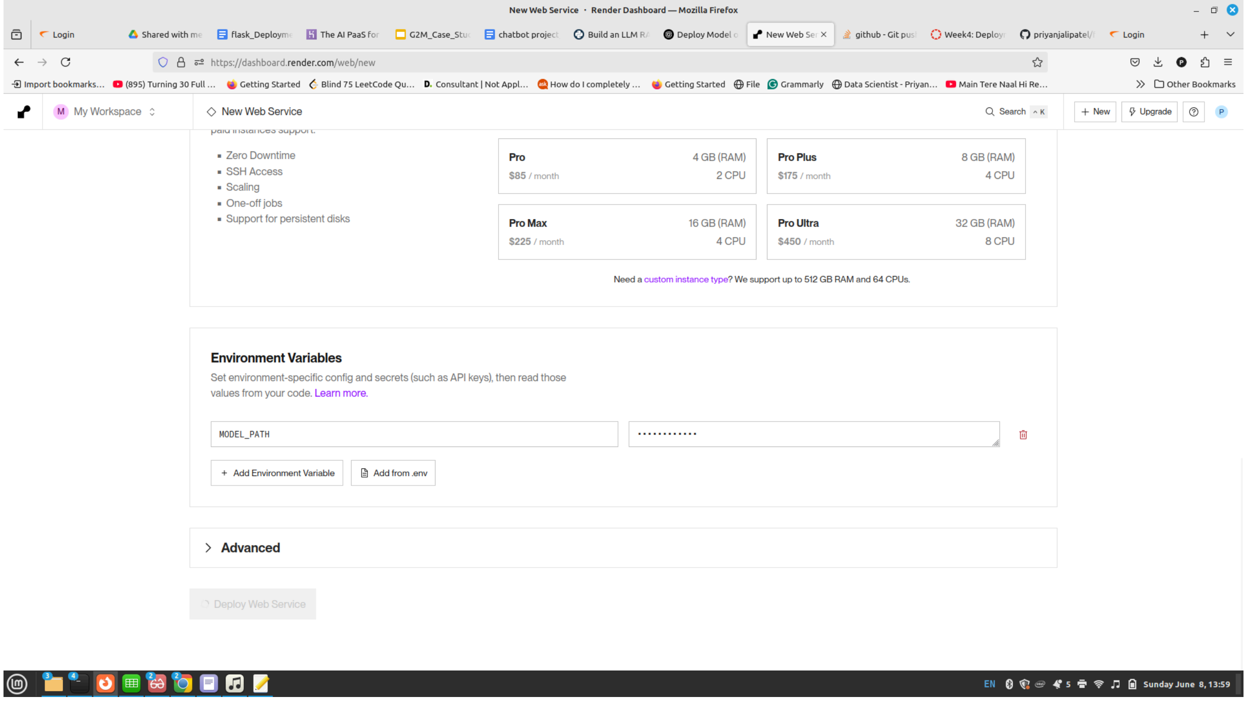
Project Optional: No project / No environment
Add this web service to a project once it's created.

Language: Python 3

Branch: main
The Git branch to build and deploy.

Region: ● Oregon (US West) 3 existing services
Your services in the same region can communicate over a private network. You currently have services running in **Oregon**.
Deploy in a new region +

Root Directory Optional: e.g. src
If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of



paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

Pro 4 GB (RAM) $85 / month 2 CPU	**Pro Plus** 8 GB (RAM) $175 / month 4 CPU
Pro Max 16 GB (RAM) $225 / month 4 CPU	**Pro Ultra** 32 GB (RAM) $450 / month 8 CPU

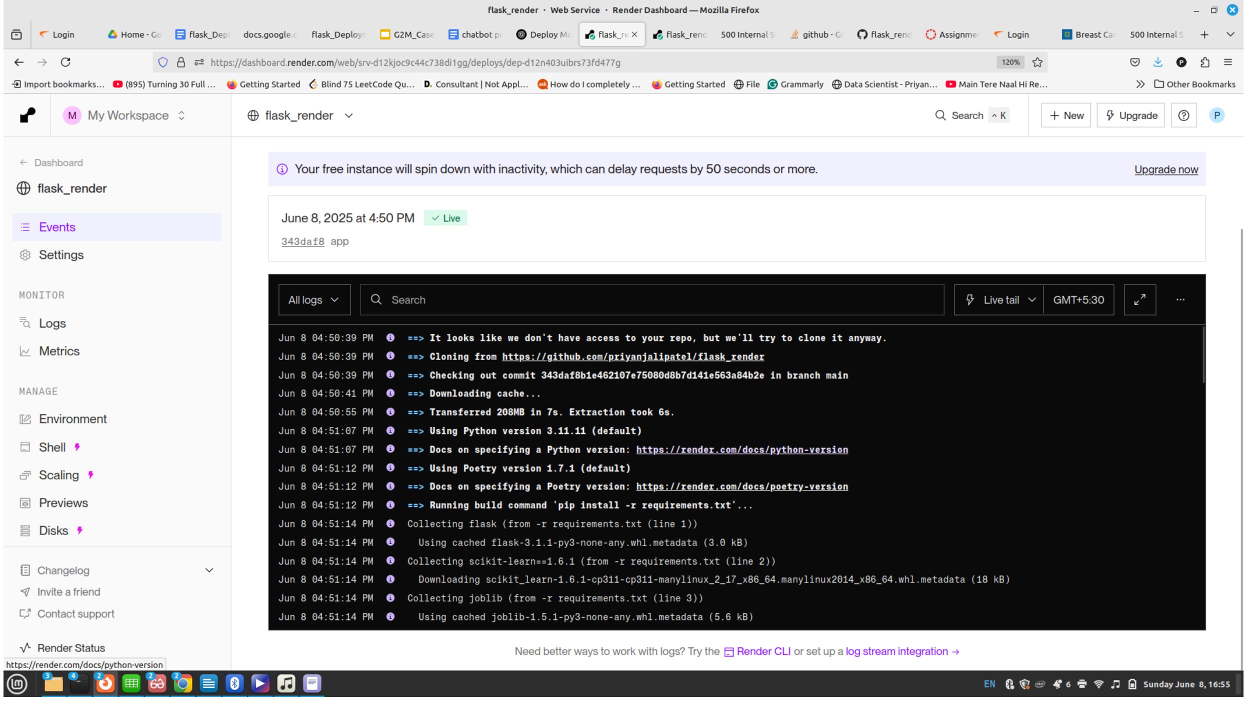
Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. Learn more.

MODEL_PATH	••••••••••••

Advanced



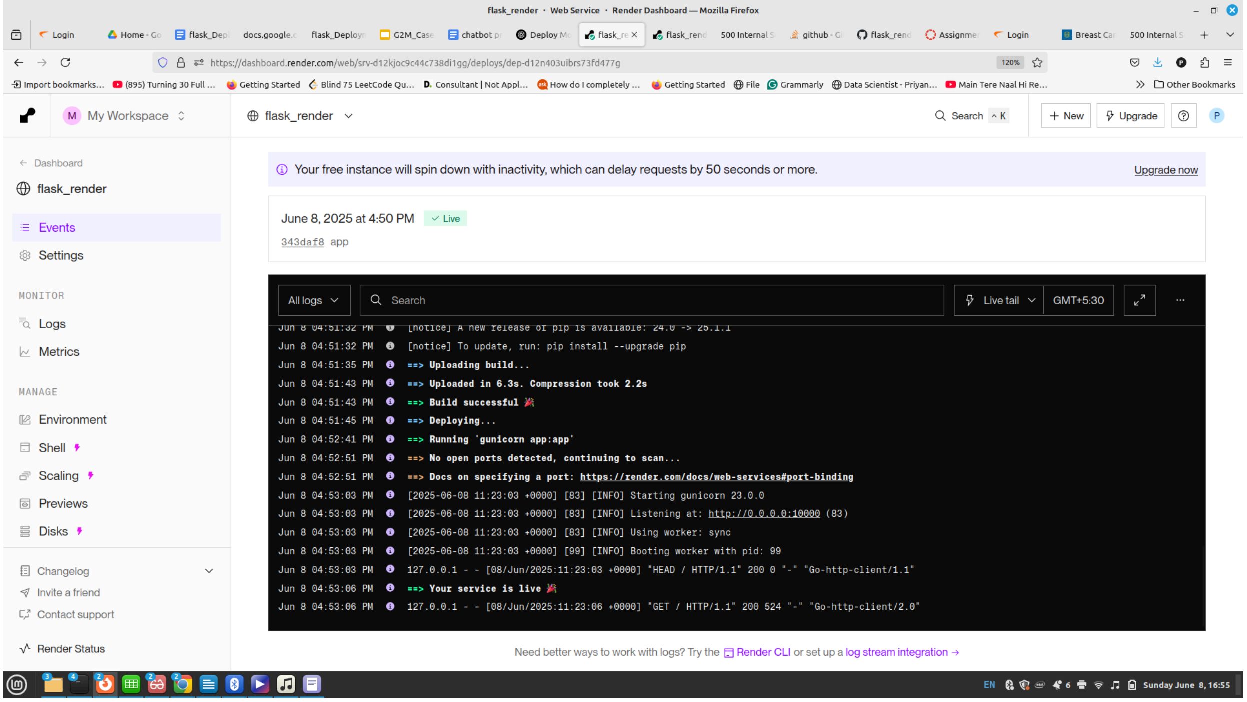
Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. Upgrade now

June 8, 2025 at 4:50 PM ✓ Live

343daf8 app

```
Jun 8 04:50:39 PM  ==> It looks like we don't have access to your repo, but we'll try to clone it anyway.
Jun 8 04:50:39 PM  ==> Cloning from https://github.com/priyanjalipatel/flask_render
Jun 8 04:50:39 PM  ==> Checking out commit 343daf8b1e462107e75080d8b7d141e563a84b2e in branch main
Jun 8 04:50:41 PM  ==> Downloading cache...
Jun 8 04:50:55 PM  ==> Transferred 208MB in 7s. Extraction took 6s.
Jun 8 04:51:07 PM  ==> Using Python version 3.11.11 (default)
Jun 8 04:51:07 PM  ==> Docs on specifying a Python version: https://render.com/docs/python-version
Jun 8 04:51:12 PM  ==> Using Poetry version 1.7.1 (default)
Jun 8 04:51:12 PM  ==> Docs on specifying a Poetry version: https://render.com/docs/poetry-version
Jun 8 04:51:12 PM  ==> Running build command 'pip install -r requirements.txt'...
Jun 8 04:51:14 PM  Collecting flask (from -r requirements.txt (line 1))
Jun 8 04:51:14 PM    Using cached flask-3.1.1-py3-none-any.whl.metadata (3.0 kB)
Jun 8 04:51:14 PM  Collecting scikit-learn==1.6.1 (from -r requirements.txt (line 2))
Jun 8 04:51:14 PM    Downloading scikit_learn-1.6.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (18 kB)
Jun 8 04:51:14 PM  Collecting joblib (from -r requirements.txt (line 3))
Jun 8 04:51:14 PM    Using cached joblib-1.5.1-py3-none-any.whl.metadata (5.6 kB)
```

Need better ways to work with logs? Try the Render CLI or set up a log stream integration →



Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. Upgrade now

June 8, 2025 at 4:50 PM ✓ Live

343daf8 app

```
Jun 8 04:51:32 PM  [notice] A new release of pip is available: 24.0 -> 25.1.1
Jun 8 04:51:32 PM  [notice] To update, run: pip install --upgrade pip
Jun 8 04:51:35 PM  ==> Uploading build...
Jun 8 04:51:43 PM  ==> Uploaded in 6.3s. Compression took 2.2s
Jun 8 04:51:43 PM  ==> Build successful 🎉
Jun 8 04:51:45 PM  ==> Deploying...
Jun 8 04:52:41 PM  ==> Running 'gunicorn app:app'
Jun 8 04:52:51 PM  ==> No open ports detected, continuing to scan...
Jun 8 04:52:51 PM  ==> Docs on specifying a port: https://render.com/docs/web-services#port-binding
Jun 8 04:53:03 PM  [2025-06-08 11:23:03 +0000] [83] [INFO] Starting gunicorn 23.0.0
Jun 8 04:53:03 PM  [2025-06-08 11:23:03 +0000] [83] [INFO] Listening at: http://0.0.0.0:10000 (83)
Jun 8 04:53:03 PM  [2025-06-08 11:23:03 +0000] [83] [INFO] Using worker: sync
Jun 8 04:53:03 PM  [2025-06-08 11:23:03 +0000] [99] [INFO] Booting worker with pid: 99
Jun 8 04:53:03 PM  127.0.0.1 - - [08/Jun/2025:11:23:03 +0000] "HEAD / HTTP/1.1" 200 0 "-" "Go-http-client/1.1"
Jun 8 04:53:06 PM  ==> Your service is live 🎉
Jun 8 04:53:06 PM  127.0.0.1 - - [08/Jun/2025:11:23:06 +0000] "GET / HTTP/1.1" 200 524 "-" "Go-http-client/2.0"
```

Need better ways to work with logs? Try the Render CLI or set up a log stream integration →

https://flask-render-msai.onrender.com

After clicking the URL to open the app in the browser, there is an index.html form for entering model inputs.

Test it with values to confirm it returns malignant or benign.

P.S. : I am getting an error here .

- ## Summary

Deployment Summary: Breast Cancer Prediction Web App

Project Title: Breast Cancer Tumor Classifier

Model: Random Forest Classifier using top 15 features Dataset: Breast Cancer Wisconsin Diagnostic Dataset (sklearn.datasets.load_breast_cancer)

Deployment Type: Web Application (HTML + Flask) + API Hosting Platform: Render (Free Cloud Hosting)

Steps Followed

1. Data Preparation:
   - Loaded the breast cancer dataset from scikit-learn.
   - Selected the top 15 most relevant features using SelectKBest with f_classif.
2. Model Building:
   - Trained a compact RandomForestClassifier with n_estimators=25, max_depth=6.
   - Evaluated accuracy on test data (approx. ~94–96%).

    - Saved the model as model.joblib using joblib.dump().
3. Web App Development:
    - Created a Flask app (app.py) to load the saved model and take 15 inputs via a form.
    - Designed a user-friendly HTML form (index.html) for the input interface.
    - Rendered prediction output directly on the webpage (malignant or benign).
4. API Integration:
    - Exposed a POST endpoint /predict to receive form data and return prediction results dynamically.
5. Deployment:
    - Used Render.com (free tier) to deploy the web app.
    - Created requirements.txt, Procfile, and pushed the project to GitHub.
    - Linked GitHub repository to Render for auto-deployment.

Outcome

- A live, browser-accessible breast cancer classifier that accepts 15 feature values and predicts whether a tumor is malignant or benign.
- Fully functional on free cloud infrastructure (Render).
- Project includes both web form interface and API logic.