

Name: Yusuf Aishat

Batch Code: LISUM37

Submission Date: 3rd October, 2024

Submitted to: Data Glacier

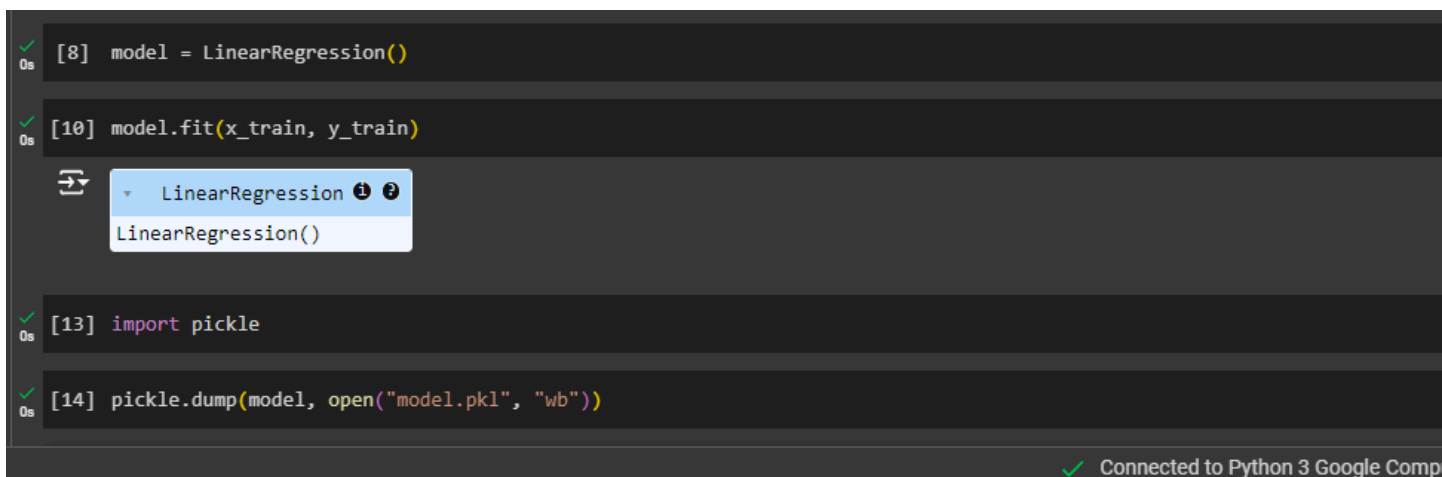
## STEP 1: CREATING AND SAVING THE MACHINE LEARNING MODEL

The Linear regression model was trained to predict student scores based on sleep and studying hours.

The model was then saved using Python's Pickle module.

```
[8] model = LinearRegression()

[10] model.fit(x_train, y_train)
```



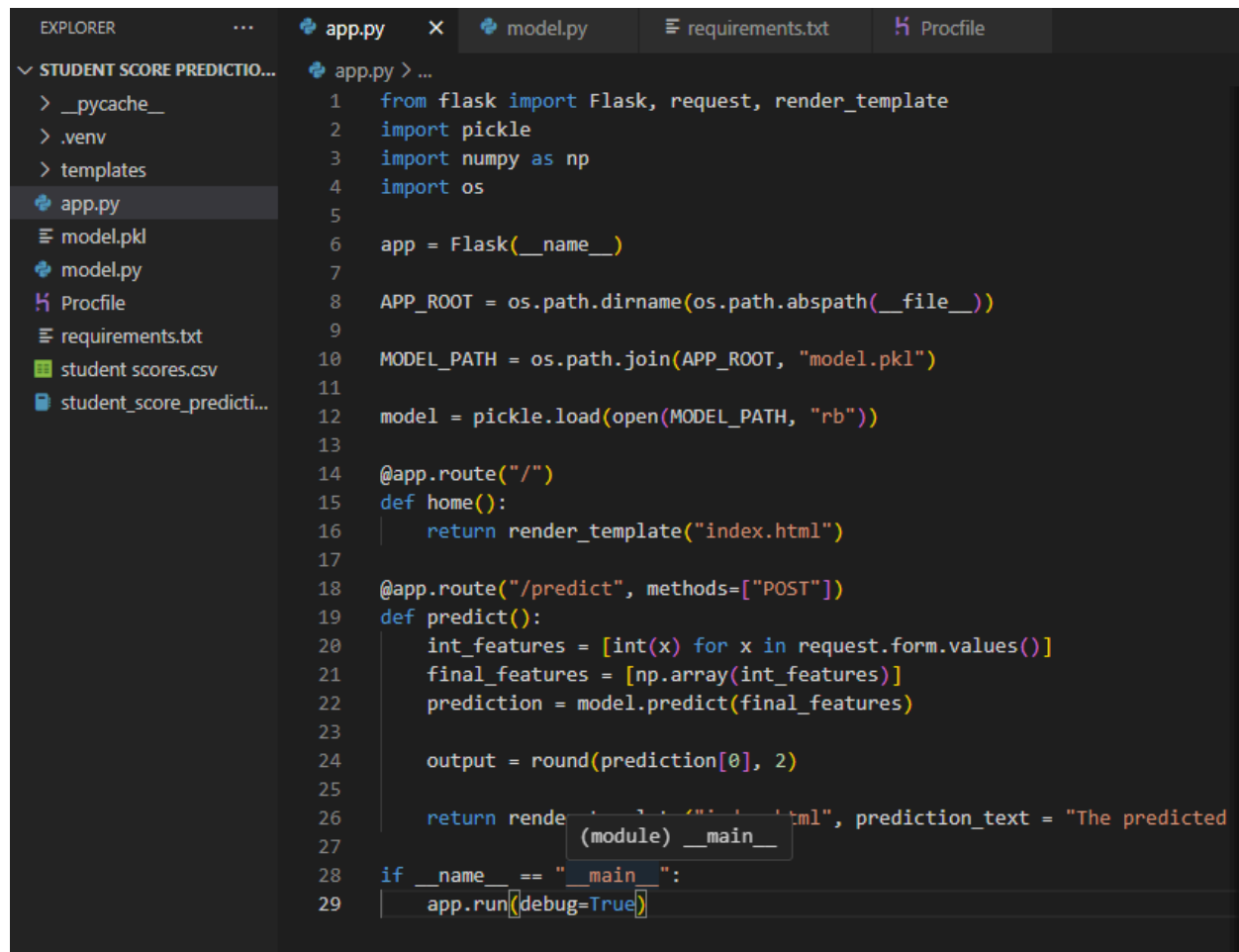
```
[13] import pickle

[14] pickle.dump(model, open("model.pkl", "wb"))
```

Connected to Python 3 Google Comp

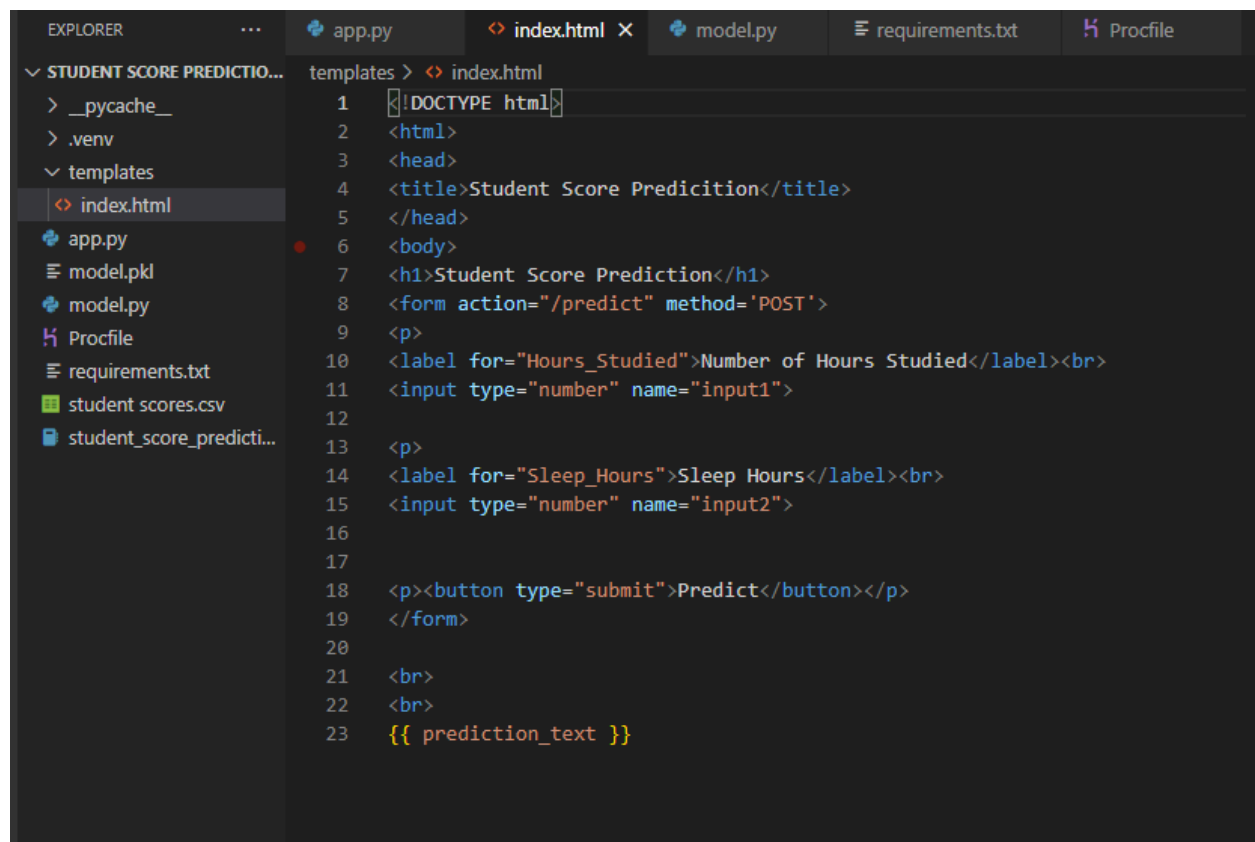
## STEP 2: CREATING THE FLASK APP

A new file `app.py` was created in the same directory as the model and the model was then loaded into the flask file.



```
1 from flask import Flask, request, render_template
2 import pickle
3 import numpy as np
4 import os
5
6 app = Flask(__name__)
7
8 APP_ROOT = os.path.dirname(os.path.abspath(__file__))
9
10 MODEL_PATH = os.path.join(APP_ROOT, "model.pkl")
11
12 model = pickle.load(open(MODEL_PATH, "rb"))
13
14 @app.route("/")
15 def home():
16     return render_template("index.html")
17
18 @app.route("/predict", methods=["POST"])
19 def predict():
20     int_features = [int(x) for x in request.form.values()]
21     final_features = [np.array(int_features)]
22     prediction = model.predict(final_features)
23
24     output = round(prediction[0], 2)
25
26     return render_template("index.html", prediction_text = "The predicted")
27
28 if __name__ == "__main__":
29     app.run(debug=True)
```

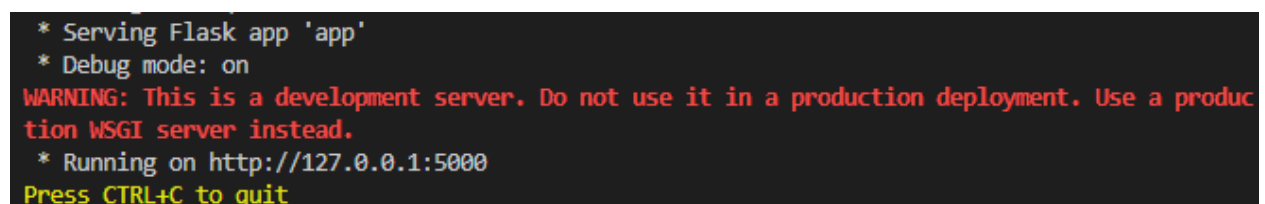
**STEP 3:** CREATING an HTML page that contains a form to fill out values that will be sent as input to the model and will return the results when the “Predict” button is clicked



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Student Score Prediction</title>
5 </head>
6 <body>
7 <h1>Student Score Prediction</h1>
8 <form action="/predict" method='POST'>
9 <p>
10 <label for="Hours_Studied">Number of Hours Studied</label><br>
11 <input type="number" name="input1">
12
13 <p>
14 <label for="Sleep_Hours">Sleep Hours</label><br>
15 <input type="number" name="input2">
16
17
18 <p><button type="submit">Predict</button></p>
19 </form>
20
21 <br>
22 <br>
23 {{ prediction_text }}
```

**STEP 4:** Running python app.py in the terminal and getting the URL:

<http://127.0.0.1:5000>



```
* Serving Flask app 'app'
* Debug mode: on
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
```

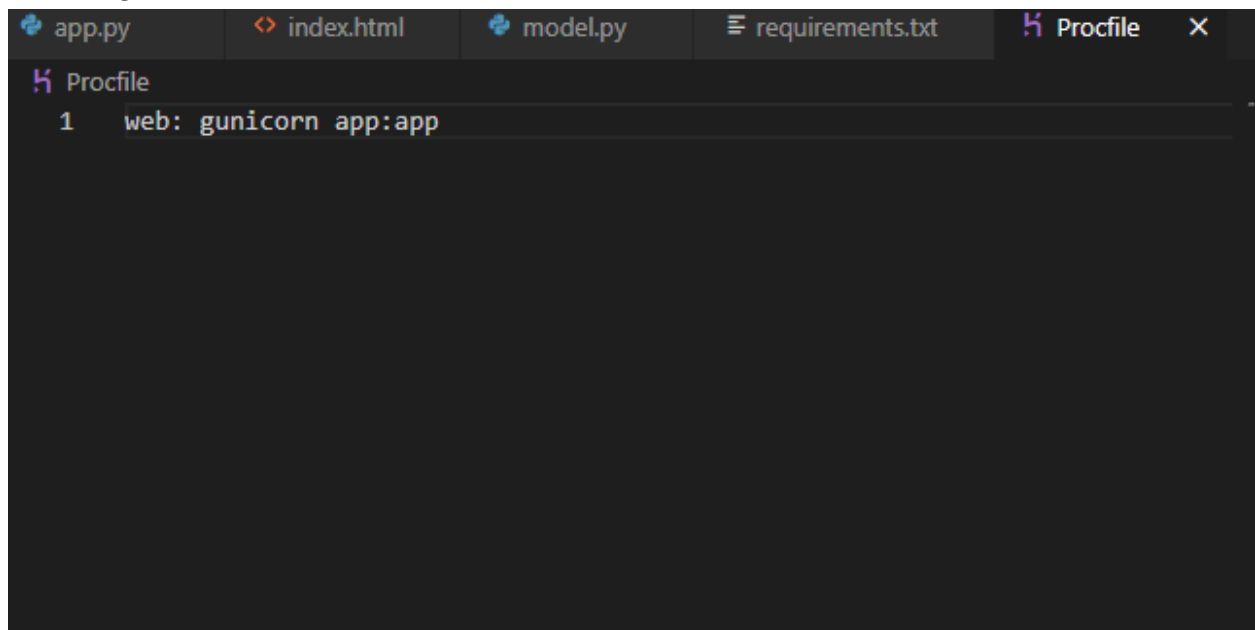
## STEP 5: Opening the URL



## DEPLOYING ON HEROKU

### STEP 6:

Creating a Procfile



## STEP 7:

### Committing code into a repository on GitHub

The screenshot shows the GitHub interface for a repository named 'student-score-prediction' owned by 'aishatyusuf'. The repository is public and has 1 watcher, 0 forks, and 0 stars. The main branch is 'main'. The repository contains several files and folders: '\_\_pycache\_\_', 'templates', 'LICENSE' (initial commit yesterday), 'Procfile', 'README.md' (created 30 minutes ago), 'app.py', 'model.pkl', 'model.py', 'requirements.txt', 'student\_scores.csv', and 'student\_score\_prediction.ipynb'. The README file is selected, showing its content: 'student score prediction'. The right sidebar displays repository statistics and sections for 'About', 'Releases', 'Packages', 'Deployments' (with 1 deployment named 'student-score-prediction-demo' from 19 minutes ago), and 'Languages'.

Repository: **student-score-prediction** (Public)

Owner: **aishatyusuf**

Navigation: [Code](#) | [Issues](#) | [Pull requests](#) | [Actions](#) | [Projects](#) | [Wiki](#) | [Security](#) | [Insights](#) | [Settings](#)

Repository Stats: [Pin](#) | [Unwatch](#) (1) | [Fork](#) (0) | [Star](#) (0)

Branch: **main** | [Go to file](#) | [Code](#)

File/Folder	Commit Message	Time
<b>aishatyusuf</b>	Add files via upload	9f6777c · 28 minutes ago
<b>__pycache__</b>	Add files via upload	28 minutes ago
<b>templates</b>	Add files via upload	28 minutes ago
<b>LICENSE</b>	Initial commit	yesterday
<b>Procfile</b>	Add files via upload	28 minutes ago
<b>README.md</b>	Create README.md	30 minutes ago
<b>app.py</b>	Add files via upload	28 minutes ago
<b>model.pkl</b>	Add files via upload	28 minutes ago
<b>model.py</b>	Add files via upload	28 minutes ago
<b>requirements.txt</b>	Add files via upload	28 minutes ago
<b>student_scores.csv</b>	Add files via upload	28 minutes ago
<b>student_score_prediction.ipynb</b>	Add files via upload	28 minutes ago

**About**

No description, website, or topics provided.

[Readme](#)

[MIT license](#)

[Activity](#)

0 stars

1 watching

0 forks

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Deployments** (1)

[student-score-prediction-demo](#) (19 minutes ago)

**Languages**

student score prediction

## STEP 8:

### Creating app on Heroku and connecting to the repository

Salesforce Platform

HEROKU

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

Personal > student-score-prediction-demo

☆ Open app More

GitHub aishatyusuf/student-score-prediction

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  
Connected**

Container Registry  
Use Heroku CLI

**App connected to GitHub**  
Code diffs, manual and auto deploys are available for this app.

Connected to [aishatyusuf/student-score-prediction](#) by [aishatyusuf](#) [Disconnect...](#)

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

## STEP 9:

### Successful Deployment.

URL: <https://student-score-prediction-demo-b8861ec7bbca.herokuapp.com/>

deployed to this app.

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

Receive code from GitHub

✓

Build main 9f6777cd

✓

Release phase

✓

Deploy to Heroku

✓

Your app was successfully deployed.

View

heroku.com

Blogs

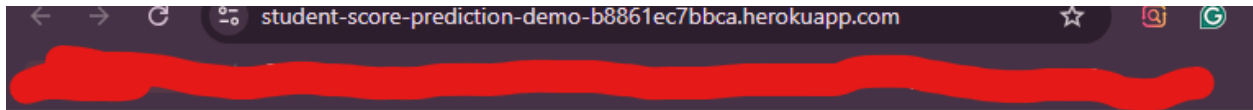
Careers

Documentation

Support

Terms of Service

Privacy



# Student Score Prediction

Number of Hours Studied

Sleep Hours