Data Glacier

Your Deep Learning Partner

VIRTUAL INTERNSHIP

DATA ANALYSIS

LISUM01

CLOUD AND API DEPLOYMENT

Payal Upadhyay

2021/07/08

Contents

Name: Payal Upadhyay

Batch Code: LISUM101

Submission Date: 2021/07/08

Submitted to: Data Glacier

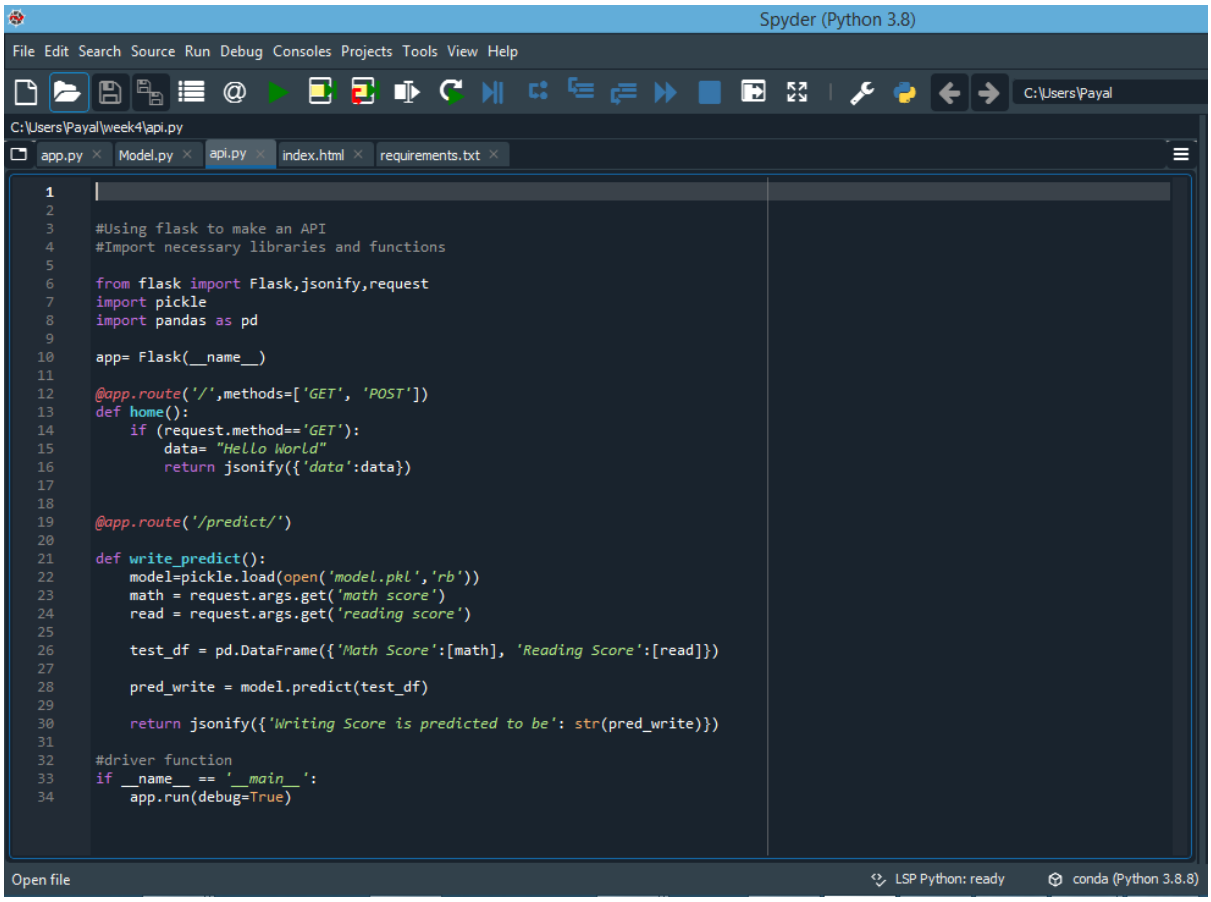
Introduction

I have used StudentsPerformance.csv (available on www.kaggle.com) to predict a student's writing score based on their math score as well as reading score. The data was split into a train set (70 %) and test set (30%). Linear Regression was used to make the prediction. The model was deployed on Postman and Heroku. Snapshots of files created and instructions on the process is provided below.

API based deployment (on Postman)

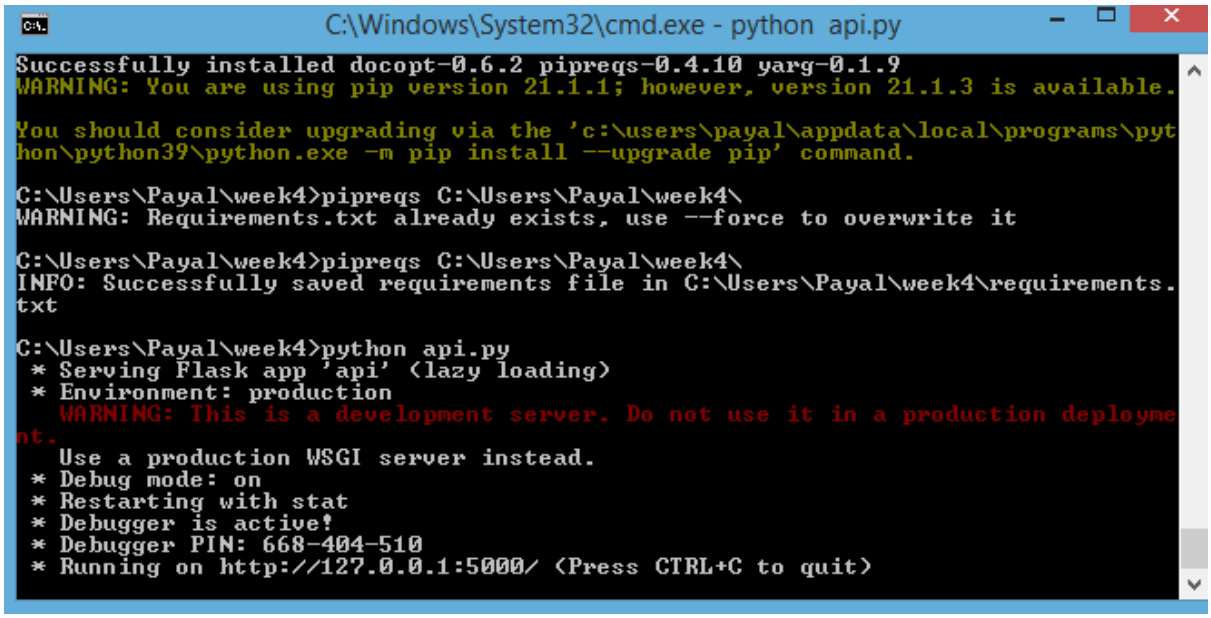
Aim: Predict the writing score based on math score and reading score

api.py



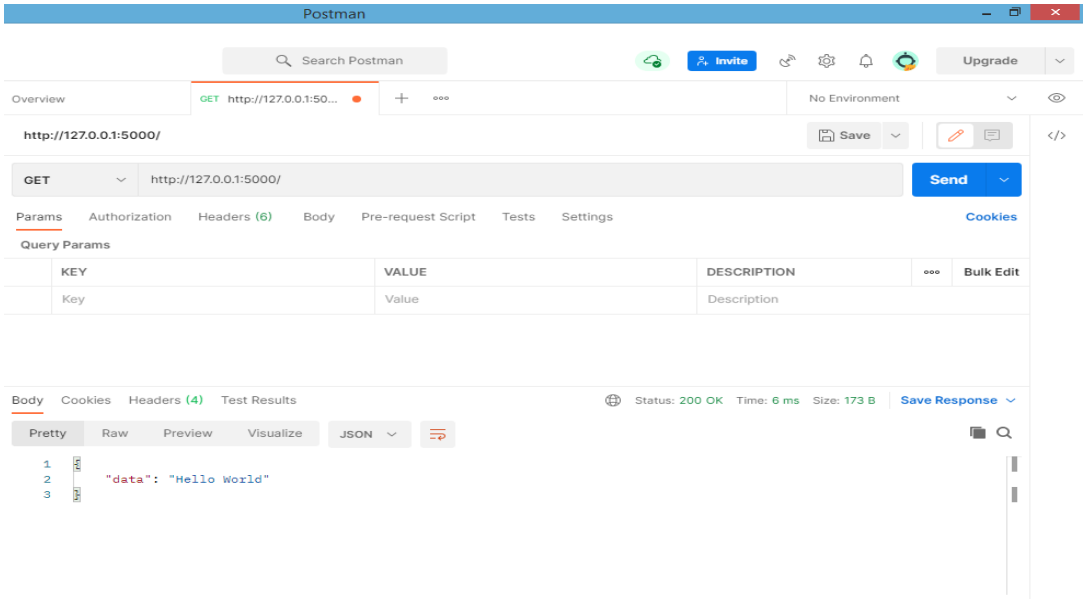

```python
#Using flask to make an API
#Import necessary libraries and functions

from flask import Flask,jsonify,request
import pickle
import pandas as pd

app= Flask(__name__)

@app.route('/',methods=['GET', 'POST'])
def home():
    if (request.method=='GET'):
        data= "Hello World"
        return jsonify({'data':data})


@app.route('/predict/')

def write_predict():
    model=pickle.load(open('model.pkl','rb'))
    math = request.args.get('math score')
    read = request.args.get('reading score')

    test_df = pd.DataFrame({'Math Score':[math], 'Reading Score':[read]})

    pred_write = model.predict(test_df)

    return jsonify({'Writing Score is predicted to be': str(pred_write)})

#driver function
if __name__ == '__main__':
    app.run(debug=True)
```

- Run the api.py file to ensure no errors are encountered

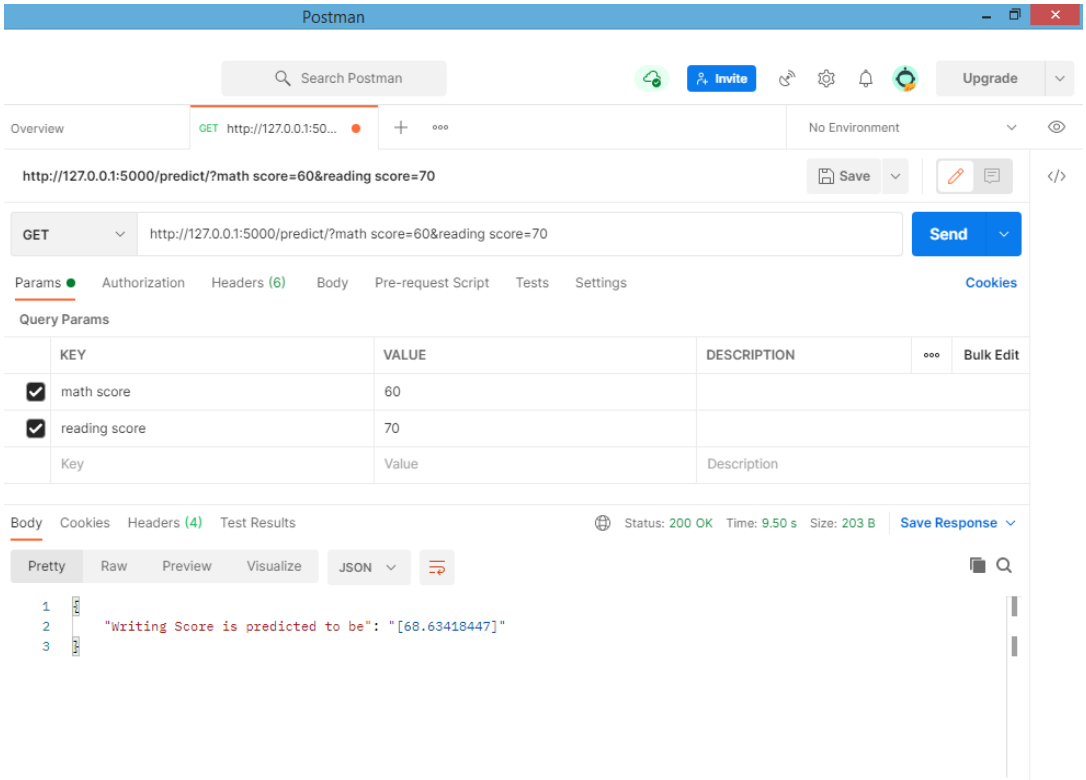


Postman

- Install Postman
- Create an account and Log in
- Click 'Create Request'
- Enter the URL from the command prompt and test

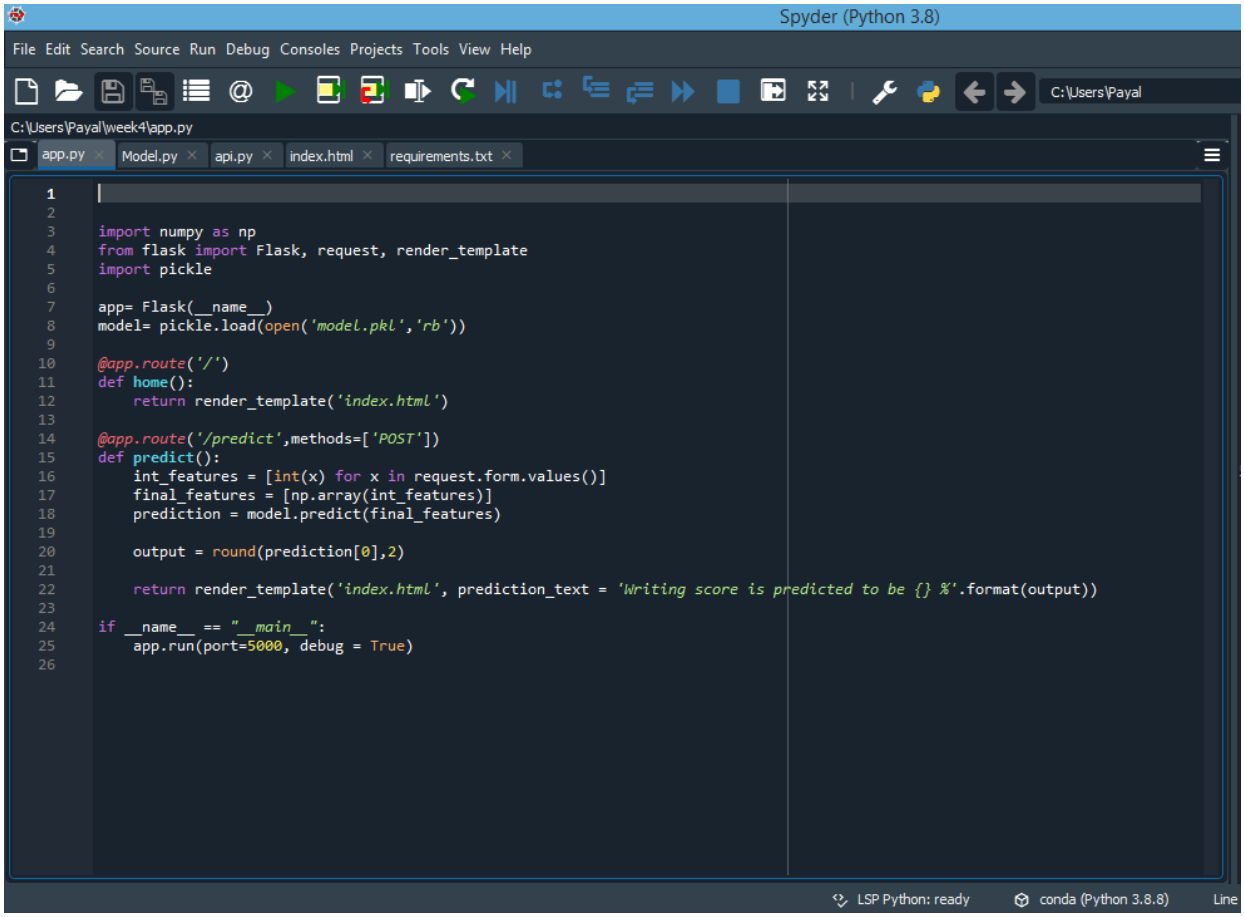


- Enter ‘predict\‘ to the url
- Enter parameters
- Click send to see output

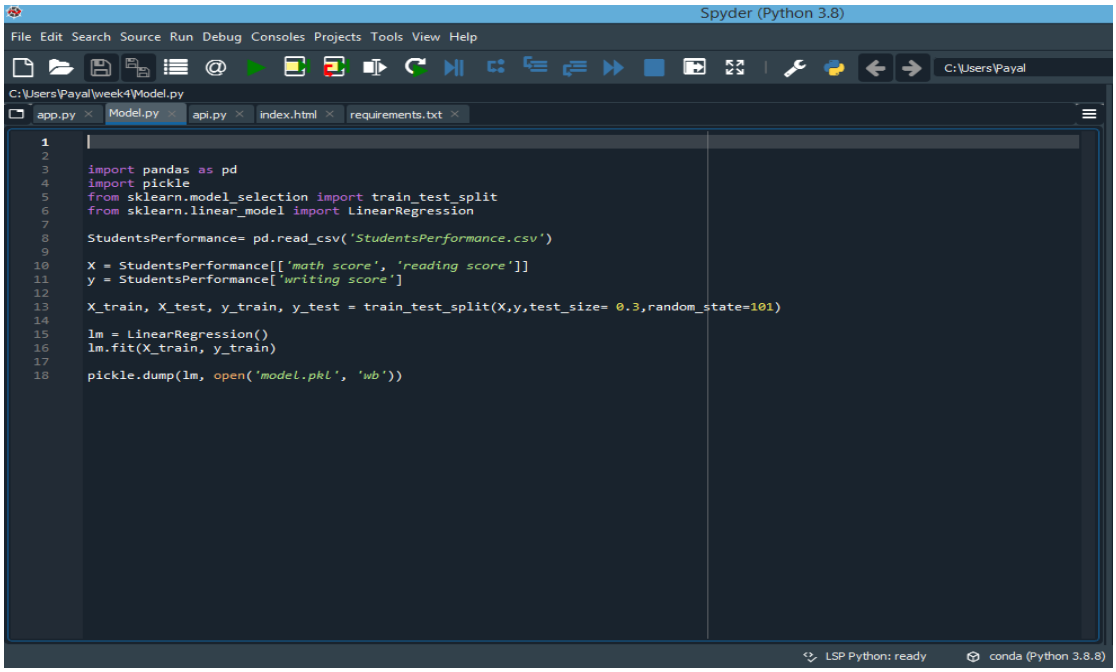


Cloud based deployment (Heroku)

app.py



Model.py



index.html

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Payal

C:\Users\Payal\week4\templates\index.html

app.py | Model.py | api.py | index.html | requirements.txt

```html
<!DOCTYPE html>
<html>

<head>

    <title> PREDICTING WRITING SCORE </title>



</head>

<body bgcolor="LightSalmon" text="Black" align="center">

        <div class="login">
        <h1 class="text centre"> PREDCTING THE WRITING SCORE </h1>

    <form action="{{url_for('predict')}}" method="post">

    <input type="text" name="math score" placeholder="Math Score" required="required" />
    <input type="text" name="reading score" placeholder="Reading Score" required="required" />


    <button type="submit"class="btn btn-primary btn-block btn-large"> <Font color = "DarkBlue"> <b>PREDICT</b> </Font> </button>

    </form>
    <br>
    <br>
    <b> {{ prediction_text }} </b>

    </div>

</body>
</html>
```

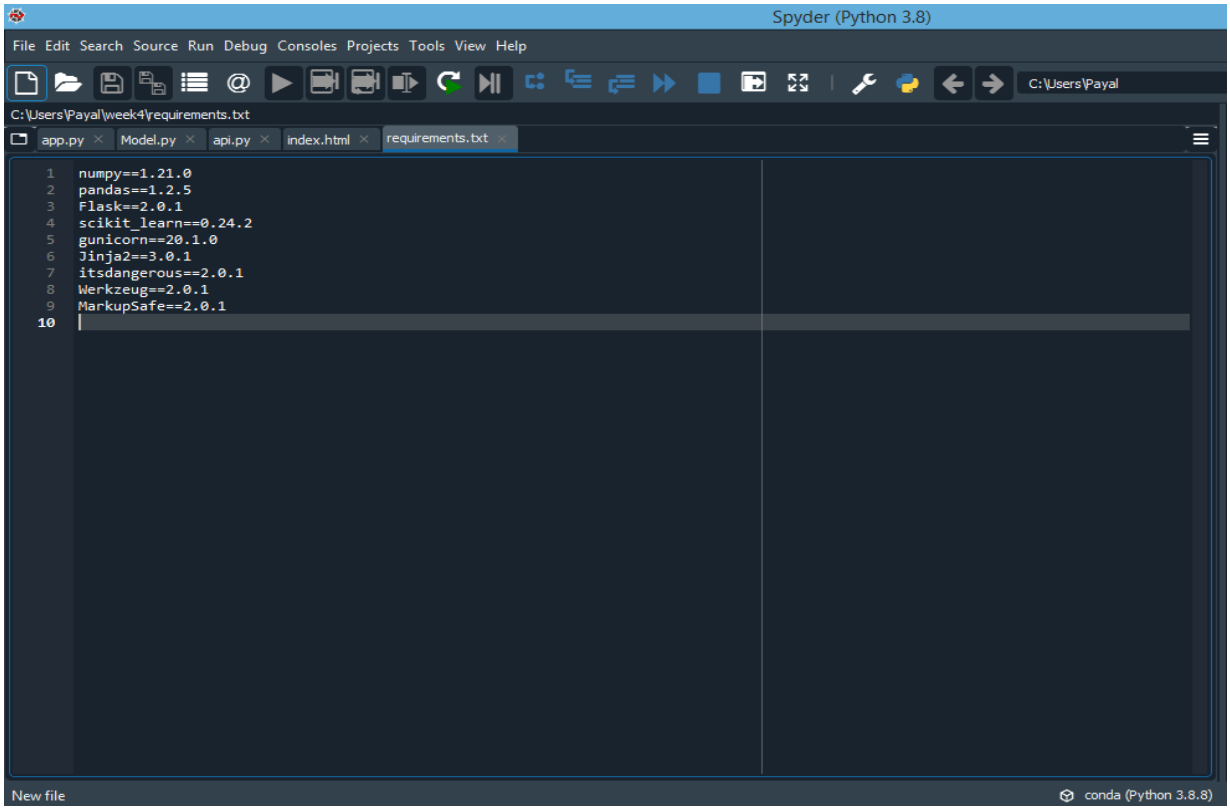
Open file | conda (Python 3.8.8)

- Run app.py and Model.py to make sure no errors are encountered

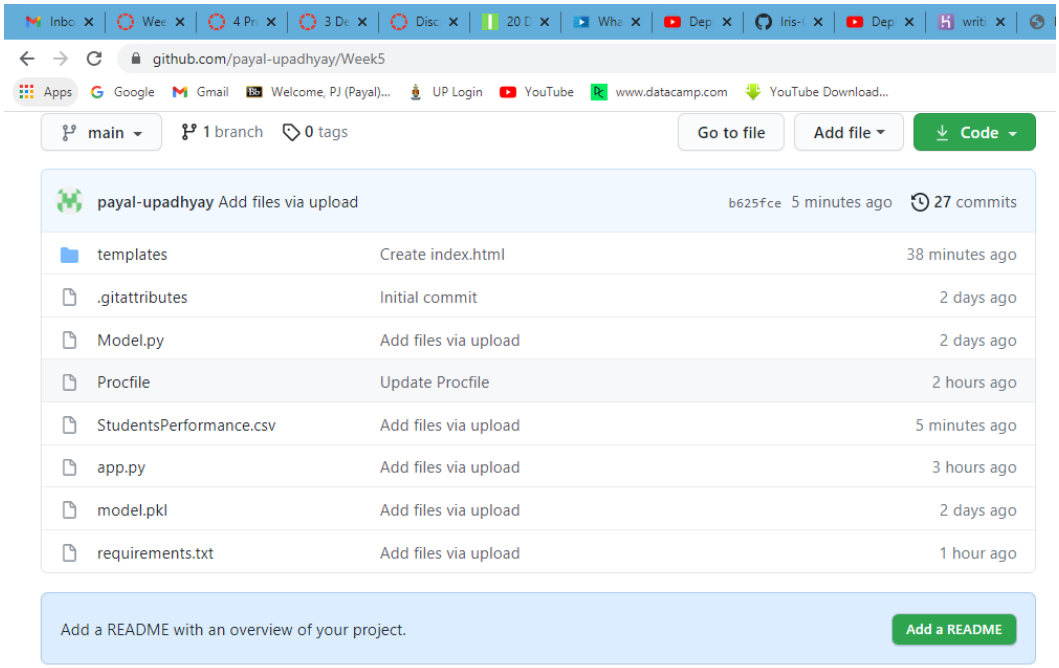
C:\Windows\System32\cmd.exe

```
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 668-404-510
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [05/Jul/2021 19:14:22] "GET /predict/?math%20score=60&reading%20sc
ore=70 HTTP/1.1" 200 -
127.0.0.1 - - [05/Jul/2021 19:16:00] "GET / HTTP/1.1" 200 -

C:\Users\Payal\week4>python app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 668-404-510
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

C:\Users\Payal\week4>python Model.py

C:\Users\Payal\week4>
```

requirements.txt

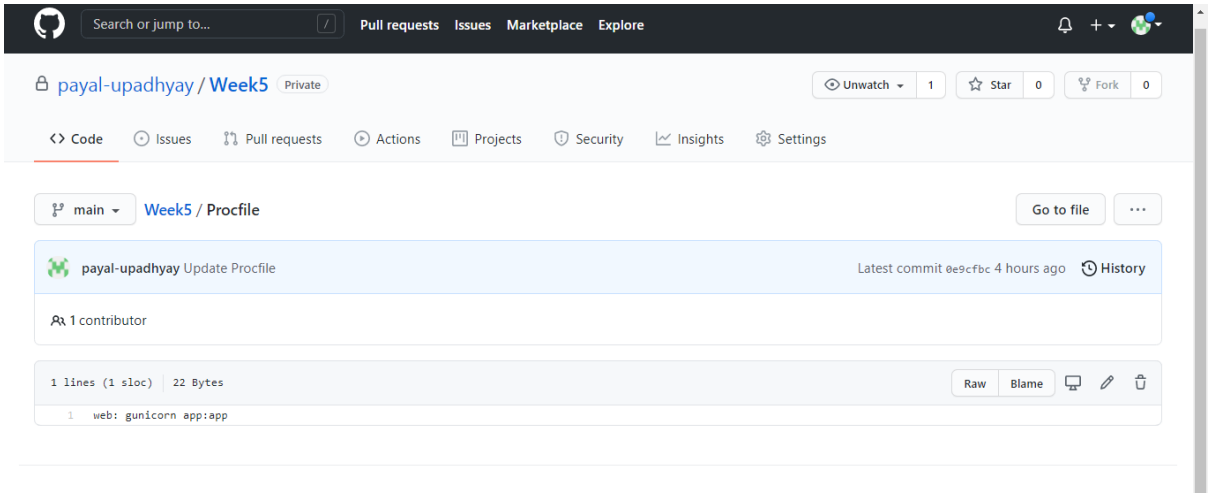


- Upload all the files to GitHub repository
- index.html file must be contained in the templates folder



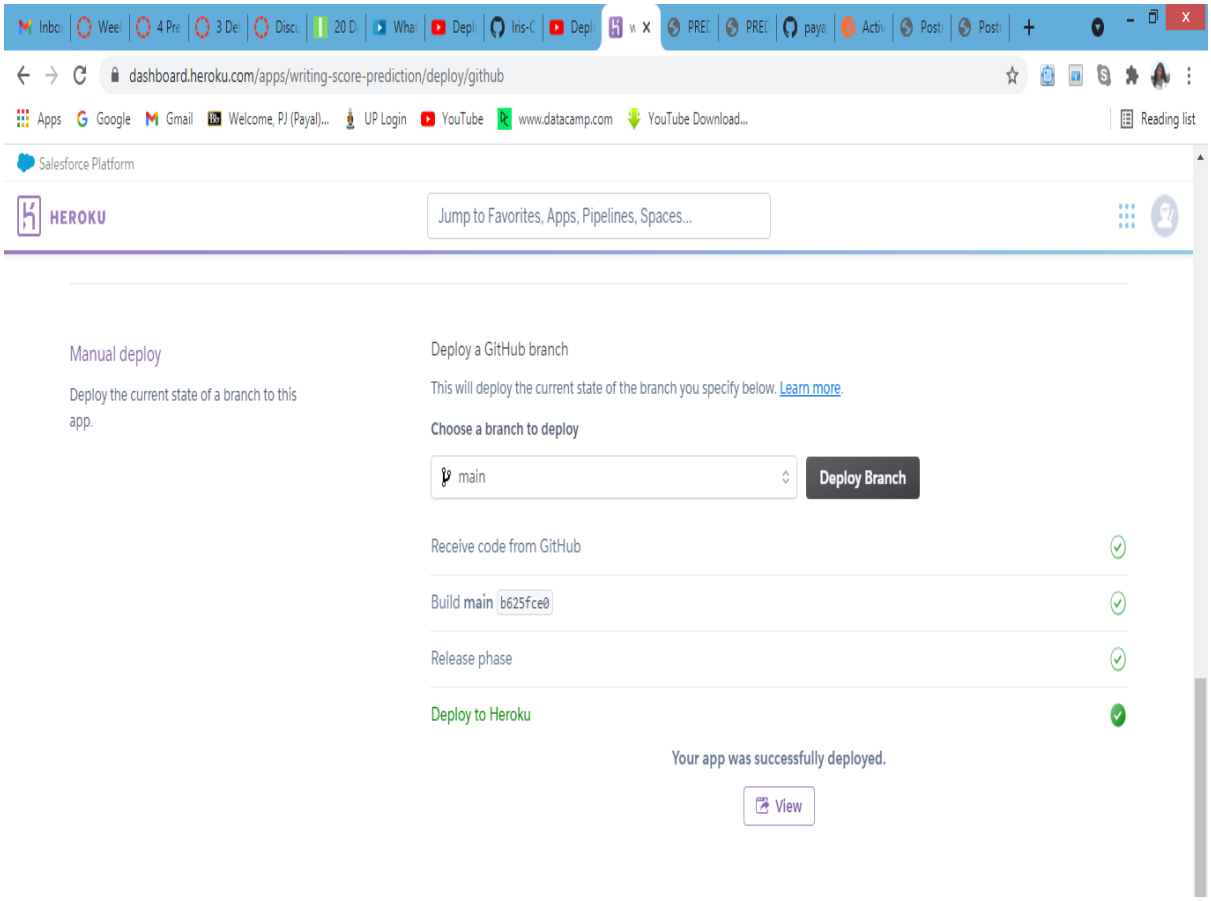
Procfile

- Create a **Procfile** (Procfile is a mechanism for declaring what commands are run by your application's dynos on the Heroku platform.)

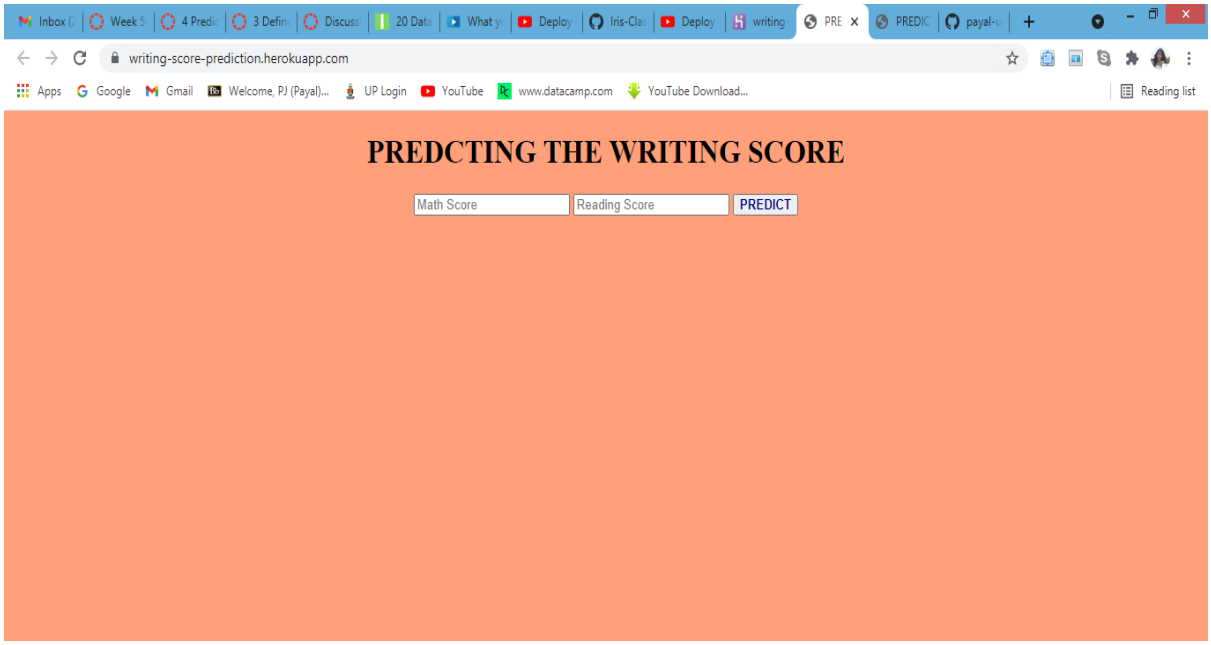


Heroku

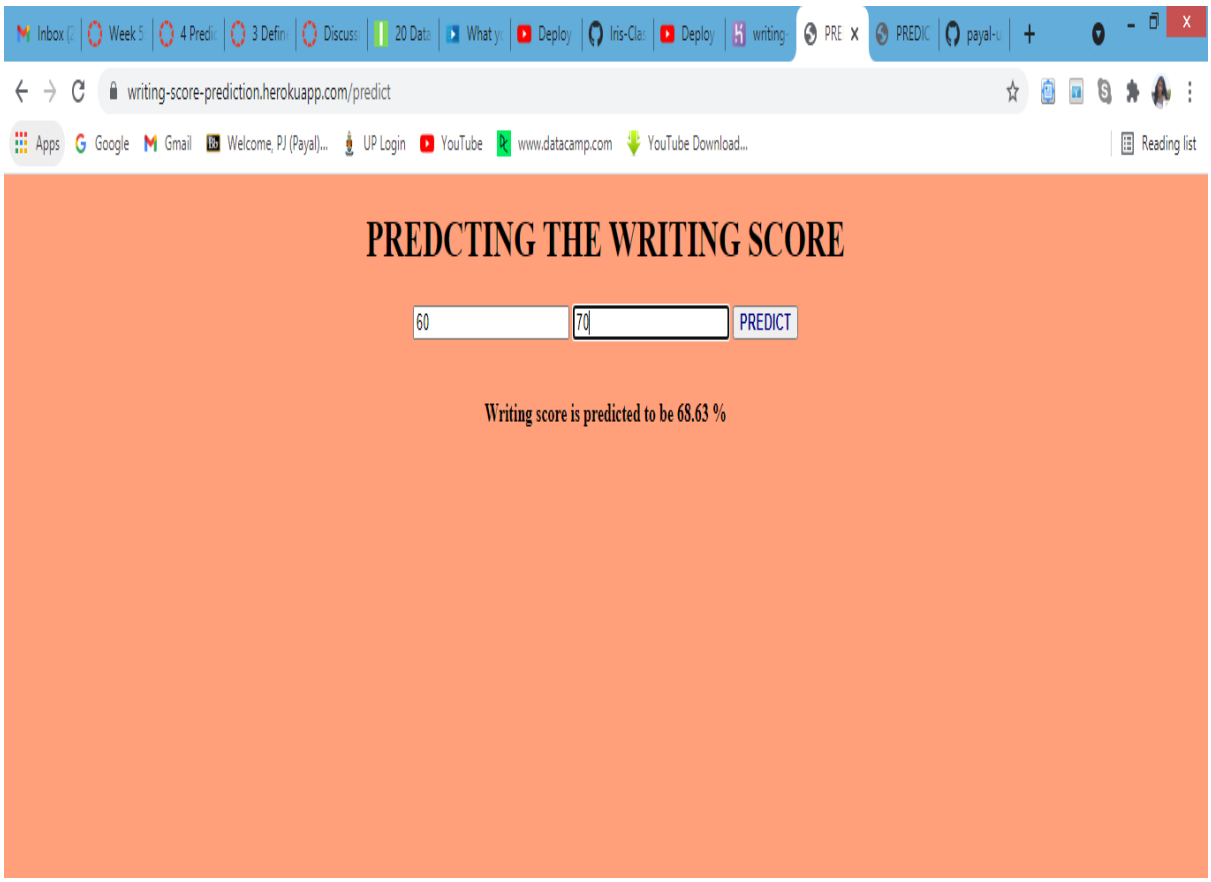
- Create and log into Heroku account
- Select ‘create new app’ and name your app
- Enter repository name and connect GitHub repository to Heroku
- Click ‘Deploy Branch’ in order to deploy



- After successful deployment click on view to view the webpage



- Test the webpage by inputting parameters



View on: https://writing-score-prediction.herokuapp.com/

Conclusion

After inputting a few scores I realized that the writing score that is predicted is more correlated with the reading score and there is very little correlation between writing score and math score.