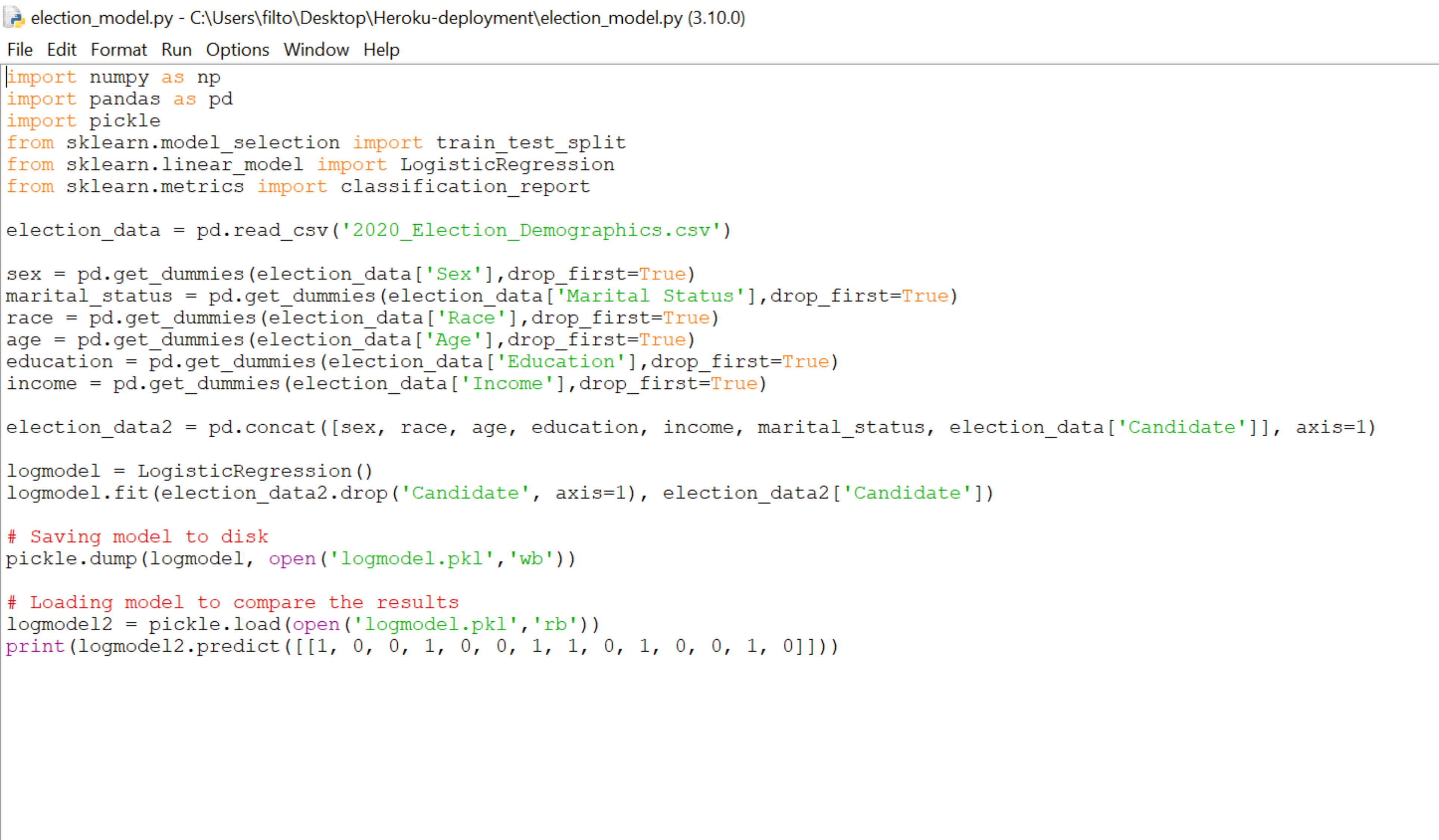
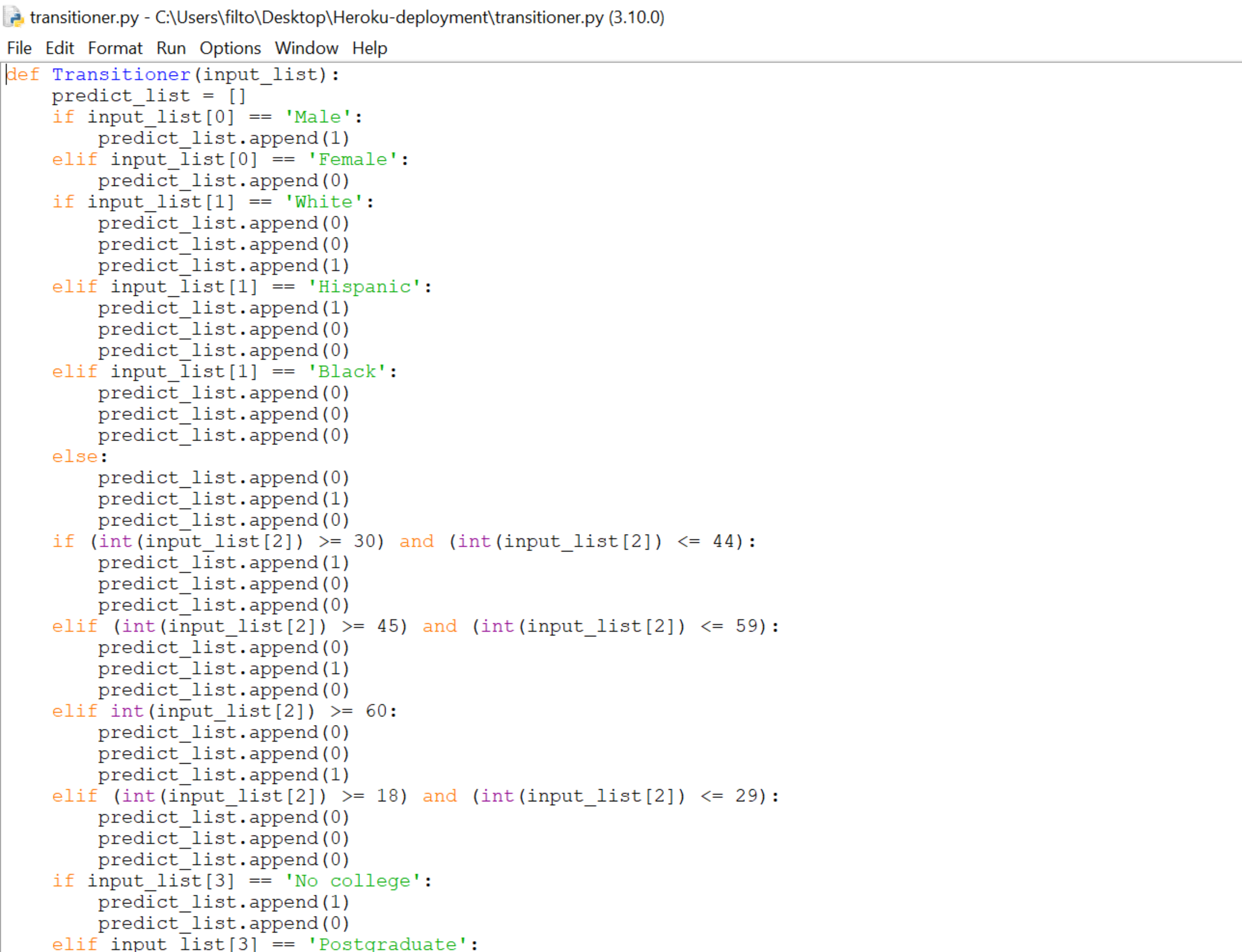

election_model.py - C:\Users\filto\Desktop\Heroku-deployment\election_model.py (3.10.0)

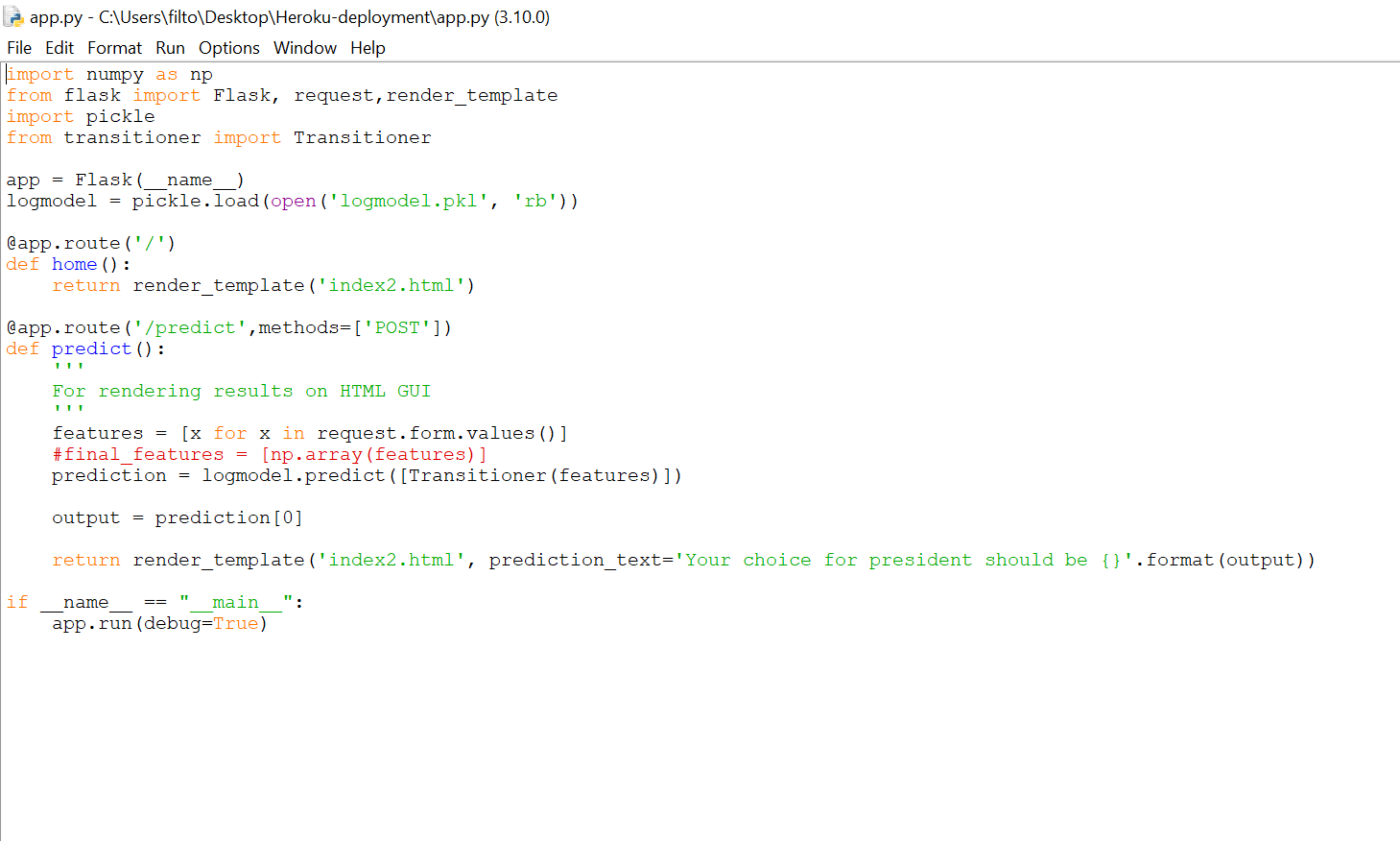
File Edit Format Run Options Window Help

```
import numpy as np
import pandas as pd
import pickle
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

election_data = pd.read_csv('2020_Election_Demographics.csv')

sex = pd.get_dummies(election_data['Sex'],drop_first=True)
marital_status = pd.get_dummies(election_data['Marital Status'],drop_first=True)
race = pd.get_dummies(election_data['Race'],drop_first=True)
age = pd.get_dummies(election_data['Age'],drop_first=True)
education = pd.get_dummies(election_data['Education'],drop_first=True)
income = pd.get_dummies(election_data['Income'],drop_first=True)

election_data2 = pd.concat([sex, race, age, education, income, marital_status, election_data['Candidate']], axis=1)

logmodel = LogisticRegression()
logmodel.fit(election_data2.drop('Candidate', axis=1), election_data2['Candidate'])

# Saving model to disk
pickle.dump(logmodel, open('logmodel.pkl','wb'))

# Loading model to compare the results
logmodel2 = pickle.load(open('logmodel.pkl','rb'))
print(logmodel2.predict([[1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0]]))
```

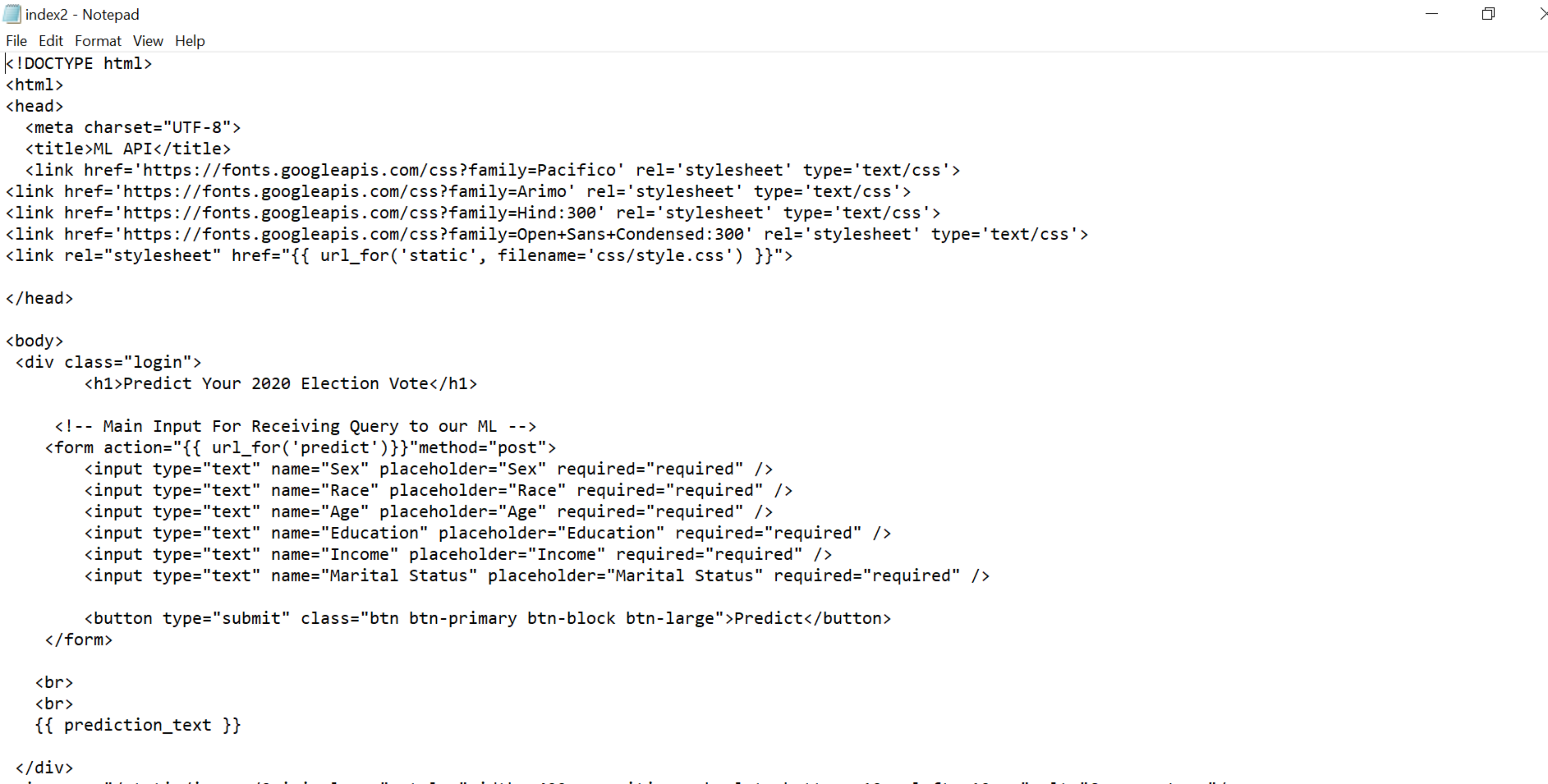


transitioner.py - C:\Users\filto\Desktop\Heroku-deployment\transitioner.py (3.10.0)

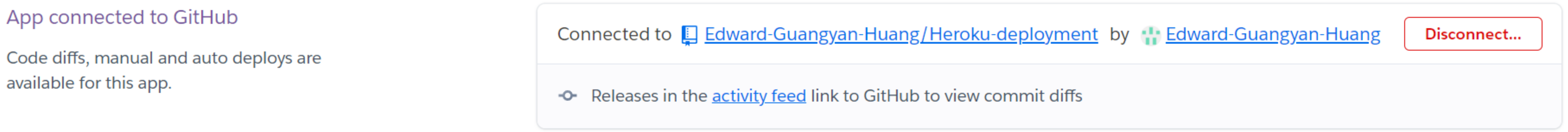
```python
def Transitioner(input_list):
    predict_list = []
    if input_list[0] == 'Male':
        predict_list.append(1)
    elif input_list[0] == 'Female':
        predict_list.append(0)
    if input_list[1] == 'White':
        predict_list.append(0)
        predict_list.append(0)
        predict_list.append(1)
    elif input_list[1] == 'Hispanic':
        predict_list.append(1)
        predict_list.append(0)
        predict_list.append(0)
    elif input_list[1] == 'Black':
        predict_list.append(0)
        predict_list.append(0)
        predict_list.append(0)
    else:
        predict_list.append(0)
        predict_list.append(1)
        predict_list.append(0)
    if (int(input_list[2]) >= 30) and (int(input_list[2]) <= 44):
        predict_list.append(1)
        predict_list.append(0)
        predict_list.append(0)
    elif (int(input_list[2]) >= 45) and (int(input_list[2]) <= 59):
        predict_list.append(0)
        predict_list.append(1)
        predict_list.append(0)
    elif int(input_list[2]) >= 60:
        predict_list.append(0)
        predict_list.append(0)
        predict_list.append(1)
    elif (int(input_list[2]) >= 18) and (int(input_list[2]) <= 29):
        predict_list.append(0)
        predict_list.append(0)
        predict_list.append(0)
    if input_list[3] == 'No college':
        predict_list.append(1)
        predict_list.append(0)
    elif input_list[3] == 'Postgraduate':
```


app.py - C:\Users\filto\Desktop\Heroku-deployment\app.py (3.10.0)
File Edit Format Run Options Window Help

```python
import numpy as np
from flask import Flask, request,render_template
import pickle
from transitioner import Transitioner

app = Flask(__name__)
logmodel = pickle.load(open('logmodel.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index2.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    features = [x for x in request.form.values()]
    #final_features = [np.array(features)]
    prediction = logmodel.predict([Transitioner(features)])

    output = prediction[0]

    return render_template('index2.html', prediction_text='Your choice for president should be {}'.format(output))

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



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
        <h1>Predict Your 2020 Election Vote</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="Sex" placeholder="Sex" required="required" />
        <input type="text" name="Race" placeholder="Race" required="required" />
        <input type="text" name="Age" placeholder="Age" required="required" />
        <input type="text" name="Education" placeholder="Education" required="required" />
        <input type="text" name="Income" placeholder="Income" required="required" />
        <input type="text" name="Marital Status" placeholder="Marital Status" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>
```

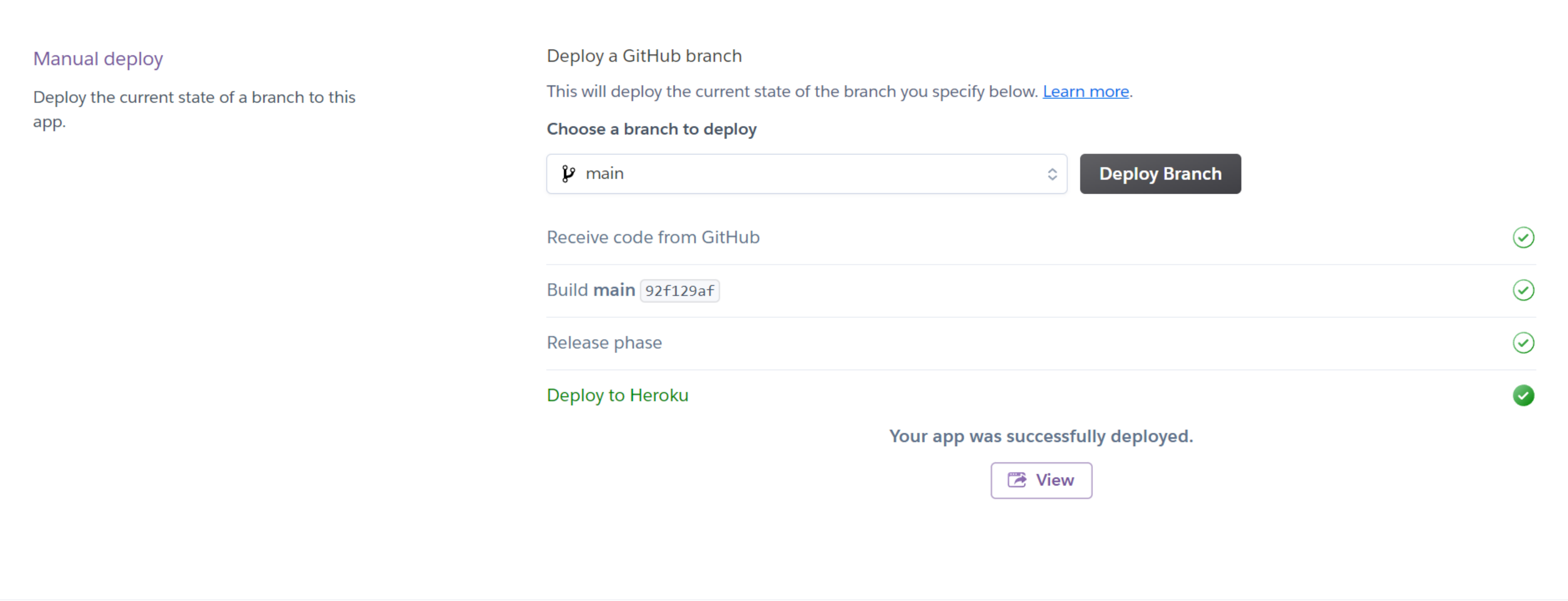

App connected to GitHub

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

Connected to Edward-Guangyan-Huang/Heroku-deployment by Edward-Guangyan-Huang

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

**NOTE: I was having problems providing my payment information, so I used my friend Edward's Github account to complete this activity**



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 ✓

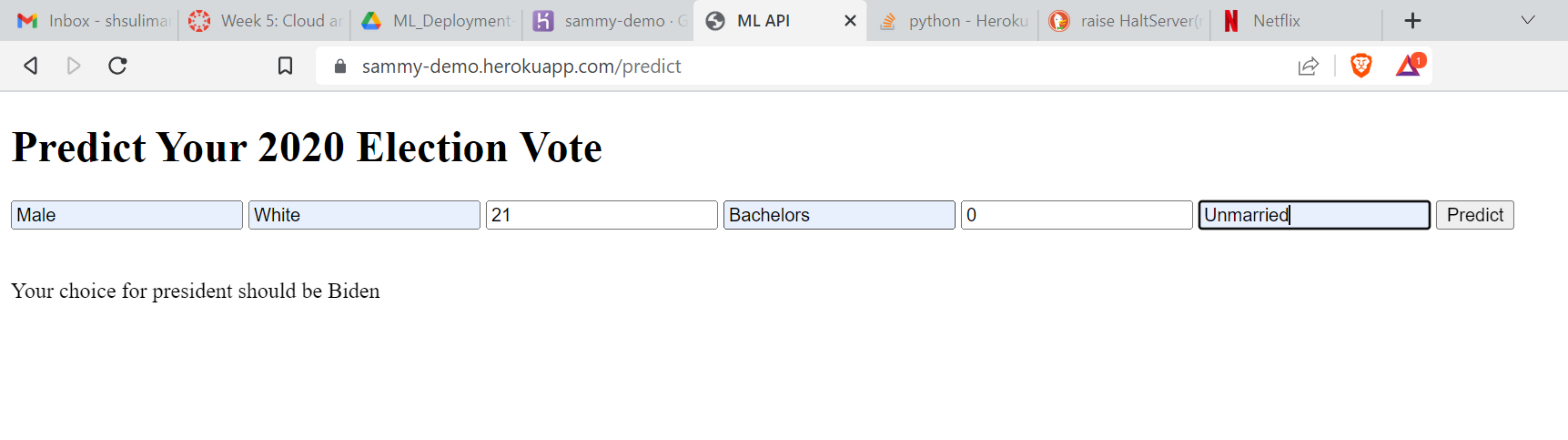
Build **main** `92f129af` ✓

Release phase ✓

Deploy to Heroku ✓

**Your app was successfully deployed.**

View



# Predict Your 2020 Election Vote

| | | | | | | |
|---|---|---|---|---|---|---|
| Male | White | 21 | Bachelors | 0 | Unmarried | Predict |

Your choice for president should be Biden