Salary Prediction ML Model Deployment

Author:

Dominic Samo, BSc Computer Science and Physics, Certified Data Scientist and Software Engineer

Environment Used To Code and Deploy Flask ML Model and Web App

- Anaconda Navigator 2.3.2
    - Spyder IDE 5.4.3
- Ubuntu 22.04 LTS

Flask

- Flask ML Model and Web App bundled together
- Model

Extra Package Installed

CORS

- It is a Flask extension for handling Cross Origin Resource Sharing (CORS), making cross-origin AJAX possible
- $ `pip install -U flask-cors`
- Flask will be used for handling all API requests

Directory Tree

```
├── app.py
├── Data
│   └── hiring.csv
├── model.pkl
├── model.py
├── README.md
├── requirements.txt
├── static
│   ├── css
│   │   └── style.css
│   └── favicon.ico
├── templates
│   └── index.html
└── testapi.py
```

Requirements file

Contains all the packages required for running the Flask application.

Data file

CSV file used for training an ML model.

ML Model

Python file used for training and exporting the ML model to be used for predicting salaries from user inputs.

View function and Routes functions file

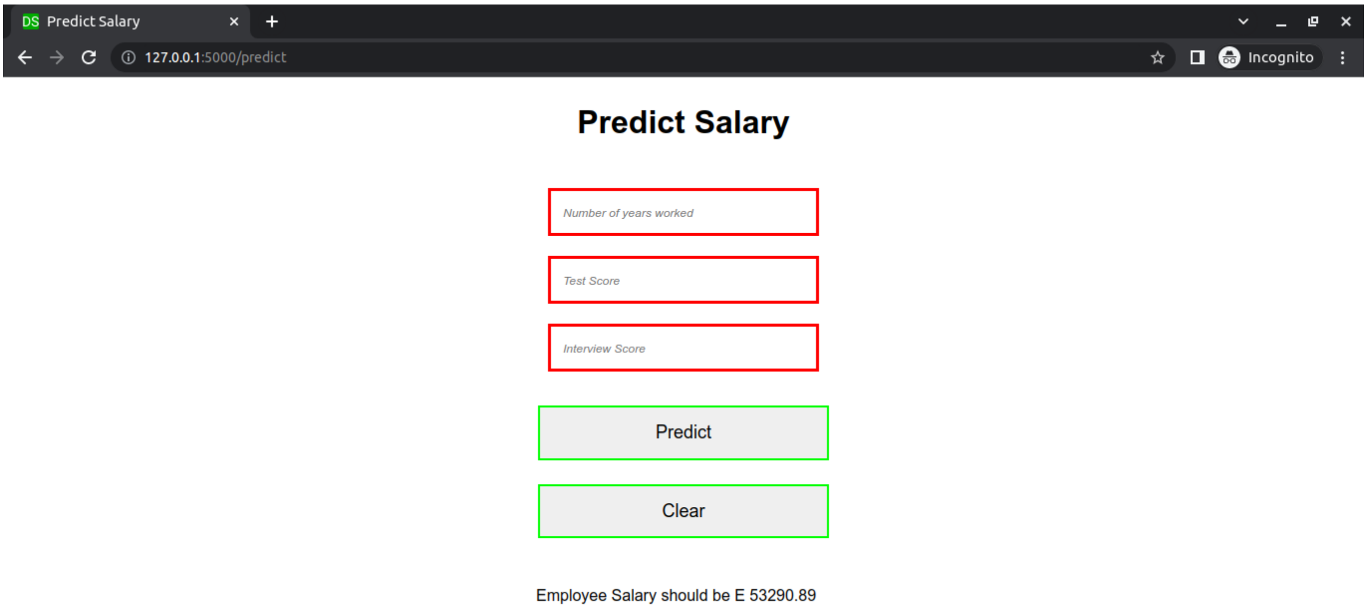
- View function renders the home html file which can be accessed from a web browser
- Routes functions handle POST requests.
    - The first route function `predict()` is for rendering the salary prediction on the view function
    - The second one `predict_api()` is for handling POST requests from any client that requsts a prediction

Running the Flask Application

- To run the Flask application, one can clone this current repository.
- Then one can open from any environment such as VS Code or Spyder IDE provided all the required packages are installed.
- Open the directory containing `app.py` and then start the Flask application by typing:
    - $ `python app.py` in a Terminal/Command Line
- The application will run locally on the URL `http://127.0.0.1:5000/`
- Copy the URL and paste it in your favorite web browser where a window similar to `Prediction from Flask Web App` below will be opened.
- Enter the required numbers in input boxes and then click `Predict`.
- A prediction will be shown.

Prediction from Flask Web App

The Flask web app running on a browser showing a prediction. It is using the `predict()` function.



Predict Salary

Number of years worked

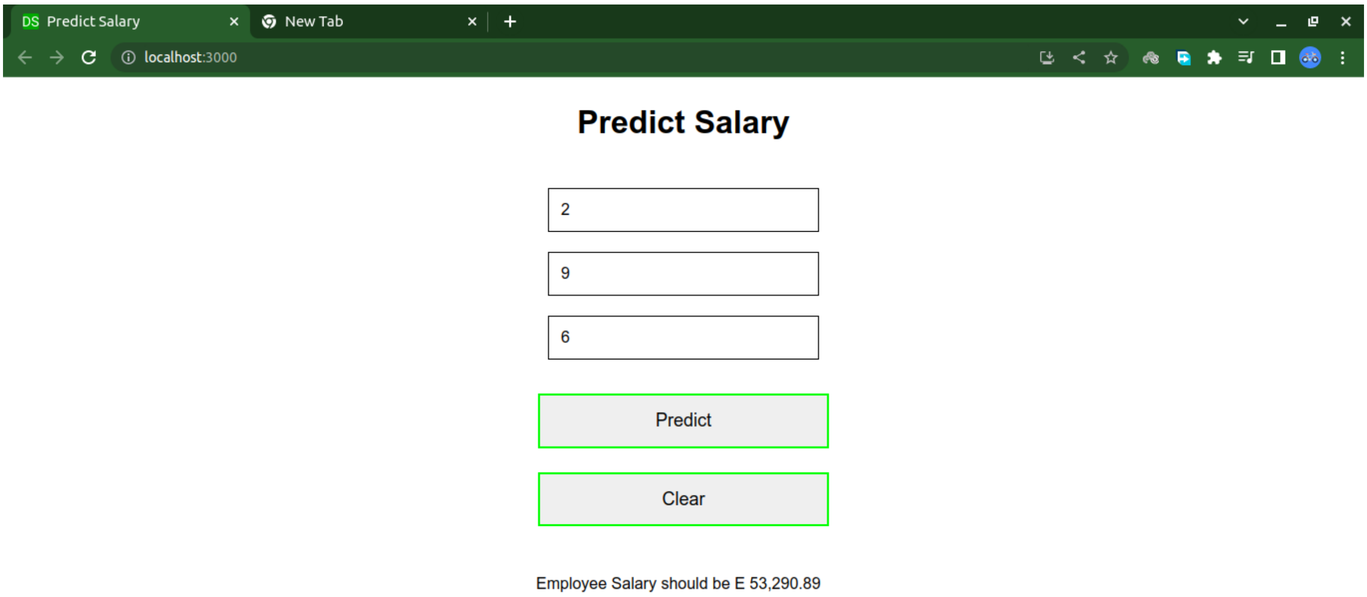
Test Score

Interview Score

Employee Salary should be E 53290.89

Prediction from React Web App

- A React web app running on a browser showing a prediction. It is using the `predict_api()` function.
- Click here to go to the React web app's repository and documentation.
- The app is running from `http://127.0.0.1:3000`



Predict Salary

2

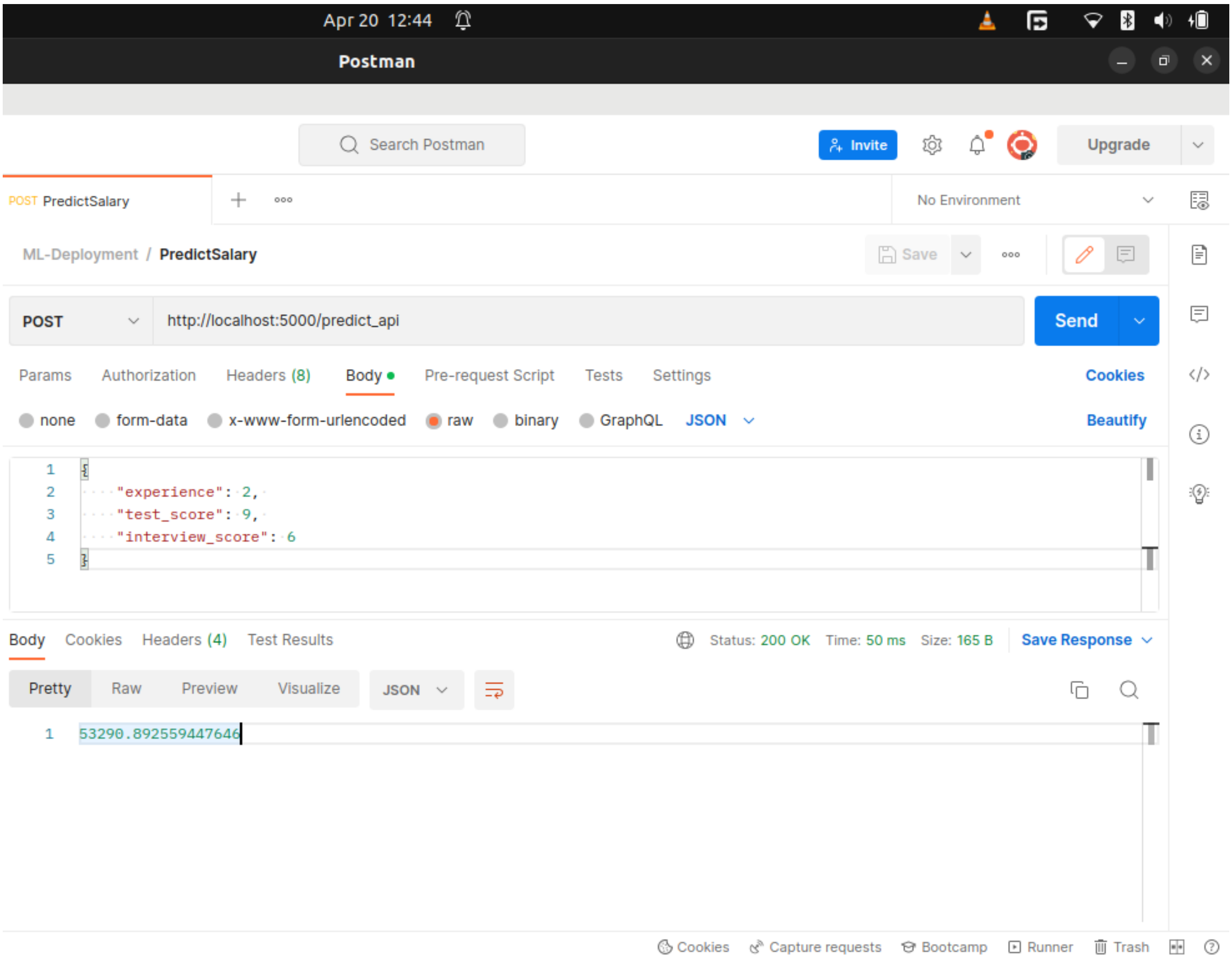
9

6

Employee Salary should be E 53,290.89

Prediction from Postman

- Any API platform can be used to test the `predict_api()` function
- The image below shows the prediction from an API POST request using Postman.



POST http://localhost:5000/predict_api

```
{
    "experience": 2,
    "test_score": 9,
    "interview_score": 6
}
```

Status: 200 OK Time: 50 ms Size: 165 B

```
53290.892559447646
```

Attribution

For inspiration I used Deploy Machine Learning Model using Flask by Krish Naik

---

Let's Connect On LinkedIn

LinkedIn - https://www.linkedin.com/in/dominic-samo-754014187/