

Application of Flask with Python

Winner Walecha

Student, Department of IT MAIT(Rohini sector-22), affiliated to GGSIPU.

Submitted: 01-06-2021

Revised: 14-06-2021

Accepted: 16-06-2021

ABSTRACT: This paper presents an overview of flask and its usage with python, how does it work and why is it considered as a great choice by many developers. The paper explains the origin of flask and how it became popular with time. To see the practical application of flask it, a python model has been created with a mean absolute error of 11.22. The end point of the project was an API created using flask. Techniques used were Linear Regression, Lasso Regression and Random Forest Regressions which were optimised using gridsearchCV.

I. INTRODUCTION:

1.1 Python and Web Development:

Python has been ranked among the top 10 languages from the start of 2003. It has now become an essential part of programming world because of its following features:

1. Simplicity
2. Robustness
3. Flexibility
4. Speed and compatibility
5. Ease of use and **many more**

Many tech giants including Spotify, Pinterest, LinkedIn and Google base have python in their architecture. With the development of the Silicon Valley python has become an important aspect of programming. Not only in development, python can be used in Machine Learning, Robotics, Data Science and many more.

Python offers so much of the functionality through its Frameworks. A Framework is a collection or bundle of modules that provides a generic functionality which can be integrated or modified to create softwares which are application specific.

Flask is among many of the frameworks offered by python. For a complete web application Pyramid and Django can be handy. Similarly for visualization or prototyping any application Flask comes into play.

1.2 Flask:

The history of flask goes back in 2004, where a person named Armin Ronacher developed

flask as an April fool's joke. Flask was named as **Bottle** framework before.

Flask has become very popular among python developers. As the result it got second most stars on GitHub among other python Web-D Frameworks in October 2020. It was also voted as the most recognised web framework in Python Developers Survey (2018).

Flask is a web framework which is written in python. It is referred to as a microFramework because of its ability of not requiring any tools or libraries to function. It has no form validation or database abstraction layer. Further, it supports multiple extensions that can add functionality as if they were the part of flask itself.

It is included into small scale solutions, for example an app or an API, where a little functionality is the top priority.

Other Features of Flask:

1. Support for secure cookies
2. Unicode based
3. Restful request dispatching
4. Extensive Documentation
5. Integrated support for unit testing

II. IMPLEMENTATION:

For this project Anaconda distribution of python is used. The machine learning model used for the generation of flask predicts the salary of a fresher taking into account various parameters of the job offered. The model has been trained with the mean absolute error of 11.22 and is ready to be put into production.

The following steps were taken into account while model training:

1. Using a selenium web scraper, 1000 jobs in the field of data science has been scraped.

The jobs are visited one by one and the scraper picked up the following attributes from each of them:

- Job Title
- Salary Estimate
- Job Description
- Rating
- Location

- Competitors
 - Industry
 - Revenue
 - Type of Ownership
2. Here after collecting the data features were engineered from the text of each job description to quantify the value companies put on python, excel, aws and spark.

The numeric salary was parsed out and the rows without effective efficient data were removed.

Also, a column was added if the job was at the company's headquarters. The age of the company was also taken into account considering the year of foundation of company.

The job description was checked for the skills such as:

- Python
- R
- Excel
- AWS
- Spark

And separate columns were made for the same.

Job seniority and title for the job offered were also taken into the account.

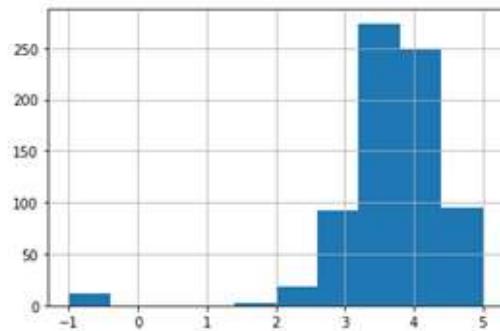


Figure 1 Rating Histogram

3. EDA was performed on the dataset and libraries such as Matplotlib and seaborn were used to obtain a graphical representation of the data. Dependencies among various variables were checked and the results were obtained.
4. For the process of Model Building the categorical variables were converted into dummy variables. Then the

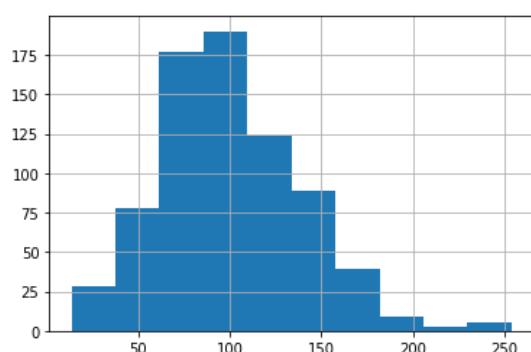


Figure 2 Average Salary Histogram

5. data is split into train and tests sets having test size of 20%.

Then 3 different models were exploited namely, Multiple Linear Regression, Lasso Regression and

Random Forest. These models were further evaluated using Mean Absolute Error.

The results were as follows.

The Random Forest model stood apart when compared to the other approaches on the test and validation sets.

Method Used	Mean Absolute Error
Random Forest	11.22
Linear Regression	18.86
Ridge Regression	19.67

Table 1. Comparison Of Methods

6. Using Flask for productionization.

Python versions 2.6 and above require installing flask

It can be installed by writing the following command in the terminal:

Conda install -c anaconda flask

page in a webserver is opened in for example, Chrome, the string “HELLO” will be displayed accordingly.

```
1  from flask import Flask
2  app = Flask(__name__)    # Flask constructor
3
4
5  @app.route('/')
6  def hello():
7      return 'HELLO'
8
9  if __name__=='__main__':
10     app.run()
```

Figure 3 Checking the installation

2.1 Project Structure:

This project has mainly 4 files:

1. Model.py:

This file contains the trained model to predict the salary based on company's description and other attributes.

2. App.py:

This contains flask API which receives an array of attributes in form of 0 and 1 through API calls, generates the estimated salary based on the model and returns it.

3. Request.py:

This uses python's request module to generate request to call API defined in app.py.

It send the array of attributes and prints the returned value from the server.

4. Data_input.py:

This file contains the array of attribute or the input data which is to be fed into the model.

This file is imported in request.py and is sent along with the request made.

📁 .ipynb_checkpoints	04-06-2021 20:45	File folder	
📁 __pycache__	05-06-2021 23:37	File folder	
📁 models	04-06-2021 20:41	File folder	
🐍 app	04-06-2021 21:13	Python Source File	1 KB
🐍 data_input	04-06-2021 20:47	Python Source File	2 KB
📄 Procfile	04-06-2021 20:26	File	0 KB
📄 requirements	04-06-2021 20:23	Text Document	2 KB
🐍 rq	04-06-2021 21:13	Python Source File	1 KB
🐍 wsgi	04-06-2021 21:13	Python Source File	1 KB

Figure 4 Folder Structure and files.

- Model.py:

```
1 app = Flask(__name__)
2 @app.route('/predict', methods=['GET'])
3 def predict():
4
5     request_json = request.get_json()
6     x = request_json['input']
7     x_in = np.array(x).reshape(1, -1)
8
9     model = load_models()
10    prediction = model.predict(x_in)[0]
11
12    response = json.dumps({'response': prediction})
13    return response, 200
14
15 if __name__ == '__main__':
16     application.run(debug = True)
```

Figure 5 Generating Response

In the model.py file the data along with the model is imported and the data is thus fed into the model. The result is sent as the response and can be seen in the terminal when the request.py file is run.

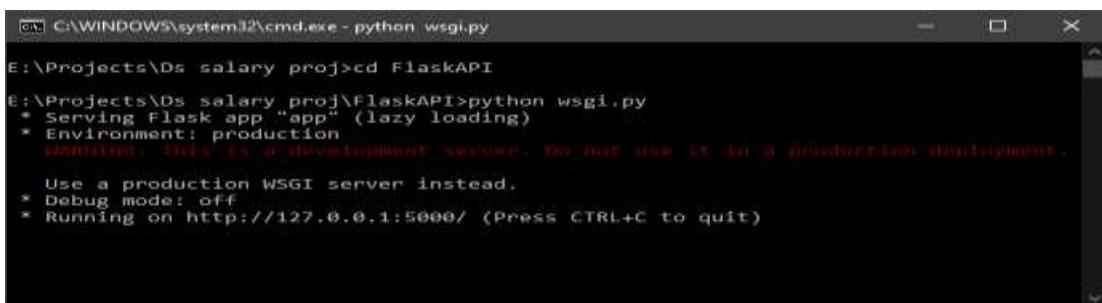
- Request.py

```
1 import requests
2 from data_input import data_in
3
4 URL = 'http://127.0.0.1:5000/predict'
5 headers = {"Content_Type": "application/json"}
6 data = {"input": data_in}
7
8 r = requests.get(URL, headers = headers, json = data)
9
10 print(r.json())
```

Figure 6 Generating Request and printing response

III. RESULTS:

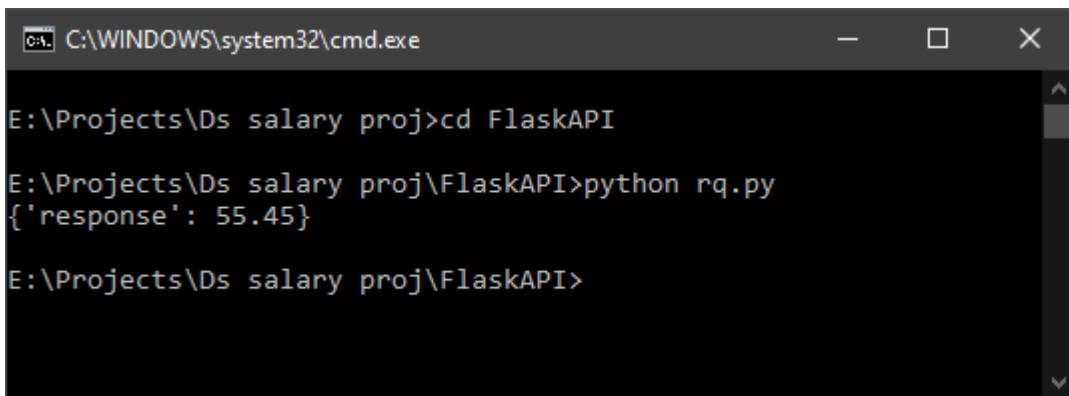
The following results were obtained:



```
C:\WINDOWS\system32\cmd.exe - python wsgi.py
E:\Projects\Ds salary proj>cd FlaskAPI
E:\Projects\Ds salary proj\FlaskAPI>python wsgi.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 7 Starting the Server

When the app.py (wsgi.py) file was run, the server was started at the url: ‘<http://127.0.0.1:5000>’ In the second terminal the rq.py (request.py) file was run which fed the data to this server and printed the response generated by this server. Below is the snippet:



```
C:\WINDOWS\system32\cmd.exe
E:\Projects\Ds salary proj>cd FlaskAPI
E:\Projects\Ds salary proj\FlaskAPI>python rq.py
{'response': 55.45}
E:\Projects\Ds salary proj\FlaskAPI>
```

Figure 8 Getting Response

The above response showed that the salary that the applicant can expect based on the data fed is: 55.45 thousand dollars.

IV. CONCLUSIONS:

As we all know that web development is a complex process of structuring content with dynamic data transactions. For maintaining such complexity technologies such as python and Flask are more useful. Such technologies are also helps to create more user friendly interface for data fetch from WWW. This paper can be conclude as Python can be used for making web more powerful, fast and efficient with the help of Flask Template Engine. Keeping in mind the usability and effective performance of flask, variety of interactive web apps can be created.

REFERENCES:

- [1]. <https://dl.acm.org/doi/abs/10.1145/2500499>
- [2]. <https://onlinelibrary.wiley.com/doi/abs/10.1111/jbl.12010>
- [3]. https://www.researchgate.net/publication/332276688_Data_science_big_data_and_statistics
- [4]. R Weihs, C., Ickstadt, K. Data Science: the impact of statistics. *Int J Data Sci Anal* 6, 189–194 (2018).
- [5]. Wu, X., Kumar, V., Ross Quinlan, J. et al. Top 10 algorithms in data mining. *Knowl Inf Syst* 14, 1–37 (2008). <https://doi.org/10.1007/s10115-007-0114-2>
- [6]. Annals of Data Science, Yong Shi
- [7]. Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing 100190, China.
- [8]. <https://core.ac.uk/download/pdf/55305148.pdf>
- [9]. <https://www.geeksforgeeks.org/python-introduction-to-web-development-using-flask/>
- [10]. <https://www.stxnext.com/python-vs-other-programming-languages/>
- [11]. <https://uwsgi-docs.readthedocs.io/en/latest/WSGIquickstart.html>
- [12]. Dr.Bhoomi Gupta, Winner Walecha, Data Science Salary Estimator
- [13]. <http://www.ijmtst.com/volume6/issue12/59.IJMTST0612210.pdf>