

ML-Model-Flask-Deployment

This is a demo project to elaborate how Machine Learn Models are deployed on production using Flask API

Project Structure

This project has four major parts :

- model.py - This contains code for our Machine Learning model to predict employee salaries based on training data in 'hiring.csv' file.
- app.py - This contains Flask APIs that receives employee details through GUI or API calls, computes the predicted value based on our model and returns it.
- request.py - This uses requests module to call APIs already defined in app.py and displays the returned value.
- templates - This folder contains the HTML template to allow user to enter employee detail and displays the predicted employee salary.

```
jupyter model.py ✓ a few seconds ago
File Edit View Language

1 # Importing the Libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 import pickle
6
7 dataset = pd.read_csv('hiring.csv')
8
9 dataset['experience'].fillna(0, inplace=True)
10
11 dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)
12
13 X = dataset.iloc[:, :3]
14
15 #Converting words to integer values
16 def convert_to_int(word):
17     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
18                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
19     return word_dict[word]
20
21 X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))
22
23 y = dataset.iloc[:, -1]
24
25 #Splitting Training and Test Set
26 #Since we have a very small dataset, we will train our model with all available data.
27
28 from sklearn.linear_model import LinearRegression
29 regressor = LinearRegression()
30
31 #Fitting model with training data
32 regressor.fit(X, y)
33
34 # Saving model to disk
35 pickle.dump(regressor, open('model.pkl','wb'))
36
37 # Loading model to compare the results
38 model = pickle.load(open('model.pkl','rb'))
39 print(model.predict([[2, 9, 6]]))
```

```
jupyter hiring.csv ✓ a few seconds ago
File Edit View Language

1 experience,test_score,interview_score,salary
2 ,8,9,50000
3 ,8,6,45000
4 five,6,7,60000
5 two,10,10,65000
6 seven,9,6,70000
7 three,7,10,62000
8 ten,,7,72000
9 eleven,7,8,80000
```

jupyter app.py ✓ a few seconds ago

File Edit View Language

```
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     """
15     For rendering results on HTML GUI
16     """
17     int_features = [int(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
24
25 @app.route('/predict_api', methods=['POST'])
26 def predict_api():
27     """
28     For direct API calls through request
29     """
30     data = request.get_json(force=True)
31     prediction = model.predict([np.array(list(data.values()))])
32
33     output = prediction[0]
34     return jsonify(output)
35
36 if __name__ == "__main__":
37     app.run(debug=True)
```

jupyter request.py ✓ a few seconds ago

File Edit View Language

```
1 import requests
2
3 url = 'http://localhost:5000/predict_api'
4 r = requests.post(url, json={'experience':2, 'test_score':9, 'interview_score':6})
5
6 print(r.json())
```

Create the machine learning model, run model.py

```
jupyter Model.py (autosaved)
File Edit View Insert Cell Kernel Widgets Help
+ < > Run Code

In [1]: # Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle

dataset = pd.read_csv('hiring.csv')

dataset['experience'].fillna(0, inplace=True)

dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)

X = dataset.iloc[:, :3]

#Converting words to integer values
def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
                'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]

X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))

y = dataset.iloc[:, -1]

#Splitting Training and Test Set
#Since we have a very small dataset, we will train our model with all available data.

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with training data
regressor.fit(X, y)

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

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[2, 9, 6]]))

[53290.89255945]
```

This would create a serialized version of our model into a file model.pkl

fred-fird > Belajar > Deploy ML Model > Deployment-flask-master

Name	Date modified	Type	Size
.ipynb_checkpoints	12/1/2019 4:13 PM	File folder	
Deployment-flask-master	12/1/2019 4:09 PM	File folder	
static	6/16/2019 3:23 PM	File folder	
templates	6/16/2019 3:23 PM	File folder	
app.py	6/16/2019 3:23 PM	Python File	1 KB
hiring.csv	6/16/2019 3:23 PM	Microsoft Excel C...	1 KB
model.pkl	12/1/2019 4:14 PM	PKL File	1 KB
model.py	6/16/2019 3:23 PM	Python File	2 KB
Model.py.ipynb	12/1/2019 4:11 PM	IPYNB File	1 KB
README.md	6/16/2019 3:23 PM	MD File	2 KB
request.py	6/16/2019 3:23 PM	Python File	1 KB
untitled	12/1/2019 4:10 PM	File	0 KB

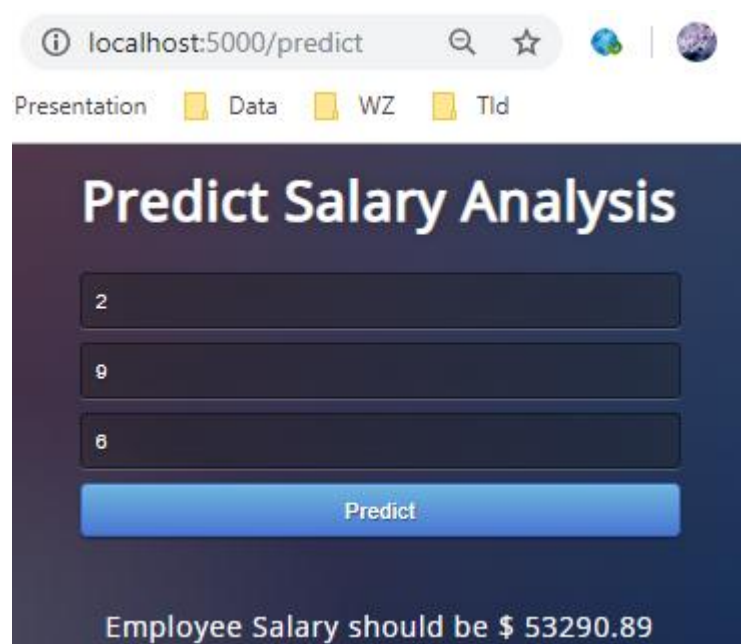
Run app.py using 'python app.py' command to start Flask API

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\Hp> cd Belajar/Deploy-ML-Model/Deployment-flask-master
(base) PS C:\Users\Hp\Belajar\Deploy-ML-Model\Deployment-flask-master> python app.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: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 216-715-217
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

By default, flask will run on port 5000.

Navigate to URL <http://localhost:5000>

You should be able to view the homepage as below :



localhost:5000/predict

Presentation Data WZ Tld

Predict Salary Analysis

2

9

6

Predict

Employee Salary should be \$ 53290.89

Enter valid numerical values in all 3 input boxes and hit Predict.

Send direct POST requests to Flask API using Python's inbuilt request module

Run python request.py command to send the request with some pre-population values.

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\Hp> cd Belajar/Deploy-ML-Model\Deployment-flask-master
(base) PS C:\Users\Hp\Belajar\Deploy-ML-Model\Deployment-flask-master> python request.py
53290.892559447646
(base) PS C:\Users\Hp\Belajar\Deploy-ML-Model\Deployment-flask-master>
```

Hosting the flask app

flask app hosted on the localhost can't be shared with others since it's "local". Host the script to a free python hosting website called pythonanywhere by following a few simple steps:

- Sign in for pythonanywhere account.
- Add a new web app
 - I Choose flask and python version 3.7. After creating the web app, we will get a URL that points to flask endpoint. By default, [username].pythonanywhere.com.
 - My URL: fredfird.pythonanywhere.com
- Install dependencies

this ML model using various external libraries such as sklearn, numpy, pandas etc. Install all of them in your pythonanywhere environment. To do so, open a new bash console and install libraries using pip. Make sure to install libraries with pip using the user option because we don't get the superuser rights.

```
pip install flask flask_cors jsonify numpy pandas
```

```
Downloading https://files.pythonhosted.org/packages/63/5d/44962520aa458852b6d40ac553dd1431f636/ab31f39ee/650b5/136d9a/jsonify-0.5.1.tar.gz
Building wheels for collected packages: jsonify
Building wheel for jsonify (setup.py) ... done
Created wheel for jsonify: filename=jsonify-0.5-cp27-none-any.whl size=1563 sha256=69a95f1de4c1374b8ff739abcb43dca11e7e8c0419612244ed6595623e1f327
4
Stored in directory: /home/fredfird/.cache/pip/wheels/ba/51/8a/136c19b604769b6cb4946fc3402a7068d86fe1f43f8824e2bf
Successfully built jsonify
Installing collected packages: jsonify
ERROR: Could not install packages due to an EnvironmentError: [Errno 13] Permission denied: '/usr/local/lib/python2.7/dist-packages/jsonify-0.5.dist-info'
Consider using the '--user' option or check the permissions.
07:06 ~/mysite $ pip install flask flask_cors jsonify numpy pandas
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Looking in links: /usr/share/pip-wheels
Requirement already satisfied: flask in /usr/local/lib/python2.7/dist-packages (1.1.1)
Collecting flask_cors
Using cached https://files.pythonhosted.org/packages/78/38/e68b11daa5d613e3a91e4bf3da76c94ac9ee0d9cd515af9c1ab80d36f709/Flask_Cors-3.0.8-py2.py3-none-any.whl
Processing /home/fredfird/.cache/pip/wheels/ba/51/8a/136c19b604769b6cb4946fc3402a7068d86fe1f43f8824e2bf/jsonify-0.5-cp27-none-any.whl
Requirement already satisfied: numpy in /usr/local/lib/python2.7/dist-packages (1.16.5)
Requirement already satisfied: pandas in /usr/local/lib/python2.7/dist-packages (0.24.2)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python2.7/dist-packages (from flask) (1.1.0)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python2.7/dist-packages (from flask) (2.10.3)
Requirement already satisfied: click>=5.1 in /usr/local/lib/python2.7/dist-packages (from flask) (7.0)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python2.7/dist-packages (from flask) (0.16.0)
Requirement already satisfied: Six in /usr/local/lib/python2.7/dist-packages (from flask-cors) (1.13.0)
Requirement already satisfied: python-dateutil>=2.5.0 in /usr/local/lib/python2.7/dist-packages (from pandas) (2.8.0)
Requirement already satisfied: pytz>=2011k in /usr/local/lib/python2.7/dist-packages (from pandas) (2019.3)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/lib/python2.7/dist-packages (from Jinja2>=2.10.1->flask) (0.23)
Installing collected packages: flask-cors, jsonify
ERROR: Could not install packages due to an EnvironmentError: [Errno 13] Permission denied: '/usr/local/lib/python2.7/dist-packages/Flask_cors'
Consider using the --user option or check the permissions.
07:07 ~/mysite $
```

- Upload the files

Inside the default folder — /mysite/ you need to upload your complete folder. You can do it either using the files page on the website or using the bash console by using wget command to download your files.

```
/home/fredfird/mysite/flask_app.py
1 # Create API of ML model using flask
2
3 ...
4 This code takes the JSON data while POST request an performs the prediction using loaded model and returns
5 the results in JSON format.
6 ...
7
8 # Import libraries
9 from flask import Flask, request, jsonify
10 from sklearn.externals import joblib
11 import pandas as pd
12
13 app = Flask(__name__)
14
15 # Load the model
16 model = joblib.load(open('/home/fredfird/mysite/model.pkl', 'rb'))
17
18 @app.route('/api/', methods=['POST'])
19 def predict():
20     # Get the data from the POST request.
21     data = request.get_json(force=True)
22
23     # convert data into dataframe
24     data.update((x, [y]) for x, y in data.items())
25     data_df = pd.DataFrame.from_dict(data)
26
27     # predictions
28     prediction = model.predict(data_df)
29
30     # Take the first value of prediction
31     output = {'prediction': int(prediction[0])}
32
33     return jsonify(output)
34
```

pythonanywhere Dashboard Consoles **Files** Web Tasks Databases

Warning You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).

/home/fredfird/ Open Bash console here **0% full** – 736.0 KB of your 512.0 MB quota

Directories

Enter new directory name New directory

[_pycache_](#)

Files

Enter new file name, eg hello.py New file

flask_app.py 2019-12-02 07:11 852 bytes

model.pkl 2019-12-02 07:51 546 bytes

Upload a file

100MB maximum size

- Reload the web app endpoint will now act as an API to facilitate other applications.

pythonanywhere Dashboard Consoles Files **Web** Tasks Databases

Warning You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).

[fredfird.pythonanywhere.com](#) Add a new web app

Configuration for fredfird.pythonanywhere.com

Reload:

Reload fredfird.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details](#).

Normal Basic Auth Digest Auth OAuth 1.0 No environment ▼

http://fredfird.pythonanywhere.com/predict_api POST ▼ URL params

form-data x-www-form-urlencoded raw Text ▼

```

1 {
2   "experience":2, "test_score":9, "interview_score":6
3 }

```

Send Preview Add to collection

Body Cookies (3) Headers (4) STATUS 200 OK TIME 322 ms

Pretty Raw Preview JSON XML

```

1 53290.892559447646

```