



Data Glacier

Your Deep Learning Partner

Name: Manoj Nagaraja

Batch code: LISUM19

Submission date: 03/04/2023

Submitted to: Week 5: Cloud and API deployment

Github : <https://github.com/ManojN7270/Cloud-and-API-Deployment.git>

```
import flask
import pickle
```

```
from sklearn import datasets
import pandas as pd
data=datasets.load_iris()
df = pd.DataFrame(data.data,columns=data.feature_names)
df['target']=data['target']
```

```
X=df.loc[:,df.columns!="target"] #Let the feature dataframe contain every column of df, except the value we are predicting
y=df.loc[:,df.columns=="target"].values.ravel() #Let the target dataframe contain only the value we are predicting
```

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.25, shuffle=True)
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X, y)
pred=knn.predict(X_test)
acc=accuracy_score(y_test, pred)
```

```
from joblib import dump, load
dump(knn, 'model.joblib')
```

```
['model.joblib']
```

```
l_m = load('model.joblib')
```

```
print(l_m)
```

```
KNeighborsClassifier(n_neighbors=3)
```

Select data (iris dataset), create and save a simple model (knn classifier):

```
1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # In[ ]:
5 import numpy as np
6 from flask import Flask, request, render_template
7 import joblib
8 from joblib import load
9 from sklearn.neighbors import KNeighborsClassifier
10 import os
11 images_folder=os.path.join('static', 'images')
12 app=Flask(__name__)
13 app.config['UPLOAD_FOLDER'] = images_folder
14 model=load('model.joblib')
15
16 @app.route('/')
17 def home():
18     return render_template('index.html')
19 @app.route('/predict', methods=['POST'])
20 def predict():
21     features=[float(x) for x in request.form.values()]
22     final_features=[np.array(features)]
23     prediction=model.predict(final_features)
24     pred_round=round(prediction[0])
25     output=""
26     if pred_round==0:
27         output+="Setosa"
28         file = os.path.join(app.config['UPLOAD_FOLDER'], 'setosa.jpg')
29     elif pred_round==1:
30         output+="Versicolor"
31         file = os.path.join(app.config['UPLOAD_FOLDER'], 'versicolor.jpg')
32     else:
33         output+="Virginica"
34         file = os.path.join(app.config['UPLOAD_FOLDER'], 'virginica.jpg')
35
36     return render_template('index.html', prediction_text='This iris flower is {}'.format(output),
37                           iris=file
38                           )
39 if __name__ == "__main__":
40     app.run(port=5000, debug=True, use_reloader=False)
41 |
```

Predict the type of iris flower (Setosa, Versicolor, or Virginica) using a K nearest neighbors classifier (k=3)

This iris flower is Setosa



The app is working as intended.

Each type of prediction was tested using Postman:

Overview | POST https://predict3ir... | No Environment

https://predict3iris.herokuapp.com/predict | Save | Edit | Comments

POST | https://predict3iris.herokuapp.com/predict | Send

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings | Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	sepal_length	4.5			
<input checked="" type="checkbox"/>	sepal_width	2.3			
<input checked="" type="checkbox"/>	petal_length	1.3			
<input checked="" type="checkbox"/>	petal_width	0.3			

Body | Cookies | Headers (6) | Test Results | Status: 200 OK | Time: 100 ms | Size: 1.22 KB | Save Response

Pretty | Raw | Preview | Visualize

Predict the type of iris flower (Setosa, Versicolor, or Virginica) using a K nearest neighbors classifier (k=3)

Sepal Length(cm)

Sepal Width(cm)

Petal Length(cm)

Petal Width(cm)

Predict

This iris flower is Setosa

Bootcamp | Runner | Trash

POST | https://predict3iris.herokuapp.com/predict | Send

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings | Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	sepal_length	5.6			
<input checked="" type="checkbox"/>	sepal_width	2.5			
<input checked="" type="checkbox"/>	petal_length	3.9			
<input checked="" type="checkbox"/>	petal_width	1.1			

Body | Cookies | Headers (6) | Test Results | Status: 200 OK | Time: 409 ms | Size: 1.22 KB | Save Response

Pretty | Raw | Preview | Visualize

Predict the type of iris flower (Setosa, Versicolor, or Virginica) using a K nearest neighbors classifier (k=3)

Sepal Length(cm)

Sepal Width(cm)

Petal Length(cm)

Petal Width(cm)

Predict

POST

https://predict3iris.herokuapp.com/predict

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	sepal_length	5.7			
<input checked="" type="checkbox"/>	sepal_width	2.5			
<input checked="" type="checkbox"/>	petal_length	5			
<input checked="" type="checkbox"/>	petal_width	2			

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 435 ms

Size: 1.22 KB

Save Response

Pretty

Raw

Preview

Visualize

Predict the type of iris flower (Setosa, Versicolor, or Virginica) using a K nearest neighbors classifier (k=3)

Sepal Length(cm)

Sepal Width(cm)

Petal Length(cm)

Petal Width(cm)

Predict

This iris flower is Virginica

Bootcamp

Runner

Trash