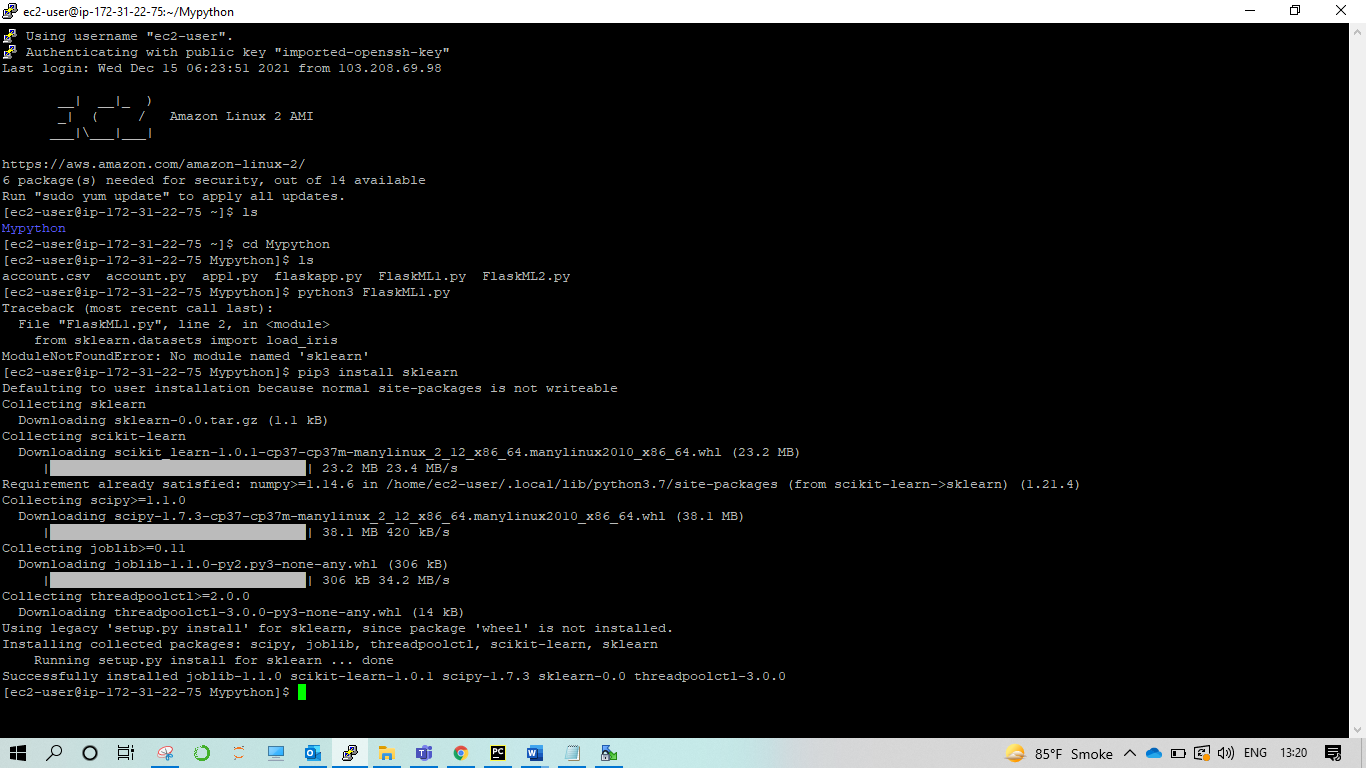
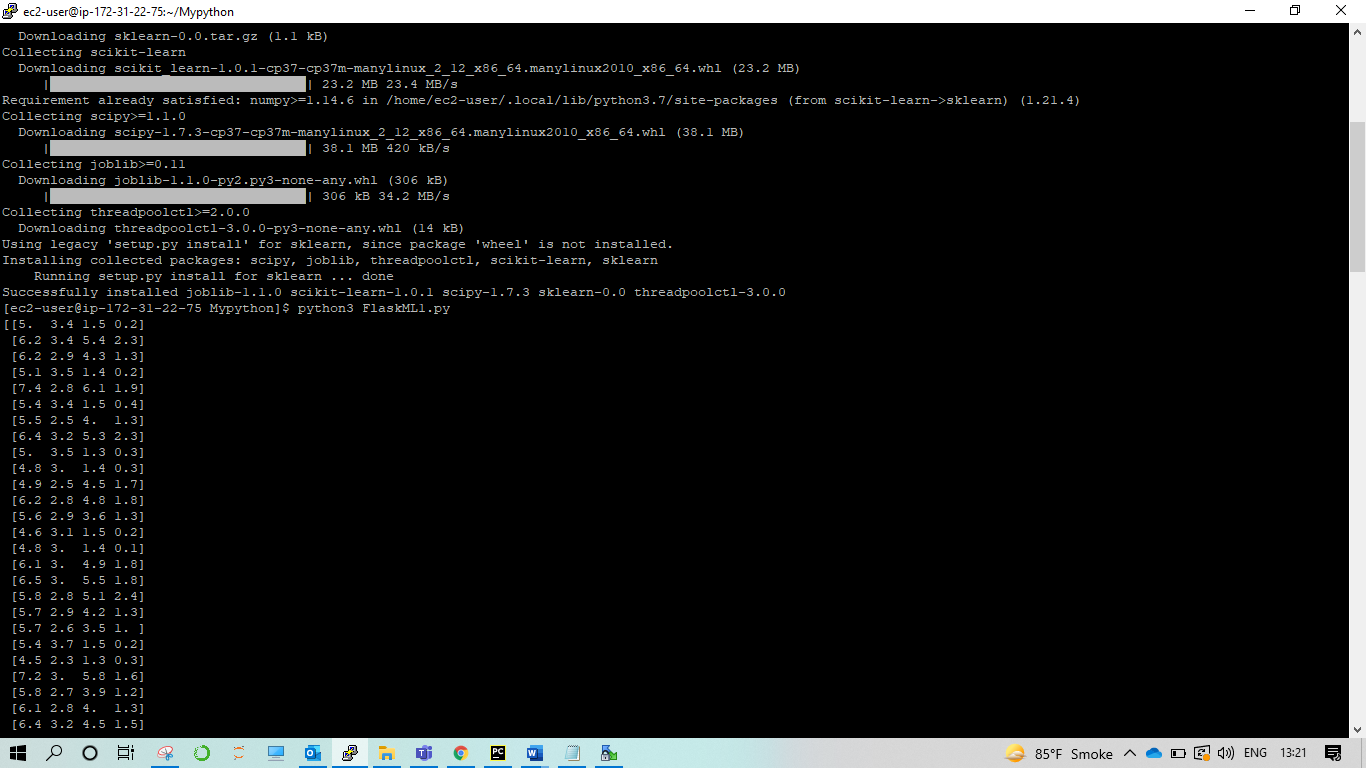


**pip3 install sklearn**





# load the iris dataset as an example

from sklearn.datasets import load\_iris

iris = load\_iris()

# store the feature matrix (X) and response vector (y)

X = iris.data

y = iris.target

#print(X)

#print(y)

# splitting X and y into training and testing sets

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

print(X\_train)

print(y\_train)

print(X\_test)

print(y\_test)

# training the model on training set

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=3)

knn.fit(X\_train, y\_train)

# making predictions on the testing set

y\_pred = knn.predict(X\_test)

print(y\_pred)

# comparing actual response values (y\_test) with predicted response values (y\_pred)

from sklearn import metrics

print("kNN model accuracy:", metrics.accuracy\_score(y\_test, y\_pred))

import pickle

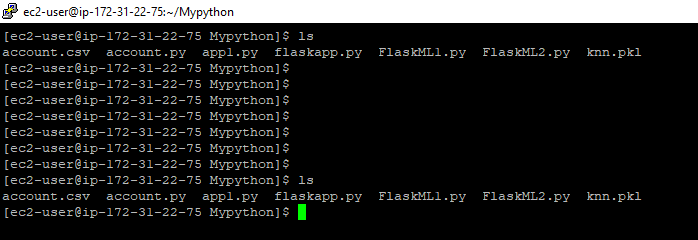
with open("knn.pkl",'wb') as model\_pkl:

pickle.dump(knn,model\_pkl)

===\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*========

FlaskML1.py

=====\*\*\*\*\*\*\*\*\*\*\*\*=========



Pickle file generated successfully…. 😊

**from flask import Flask,request**

**import numpy as np**

**import pickle**

**with open('knn.pkl','rb') as model\_file:**

**model=pickle.load(model\_file)**

**app=Flask(\_\_name\_\_)**

**@app.route('/predict')**

**def predict\_iris():**

**s\_length = request.args.get("s\_length")**

**s\_width = request.args.get("s\_width")**

**p\_length = request.args.get("p\_length")**

**p\_width = request.args.get("p\_width")**

**prediction=model.predict(np.array([[s\_length,s\_width,p\_length, p\_width]]))**

**return str(prediction)**

**if \_\_name\_\_=='\_\_main\_\_':**

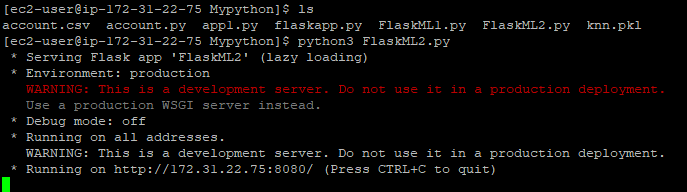
**app.run(host='0.0.0.0',port=8080)**

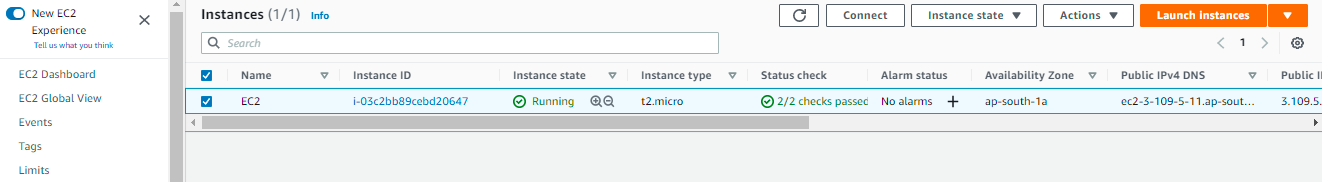
===\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*========

FlaskML2.py

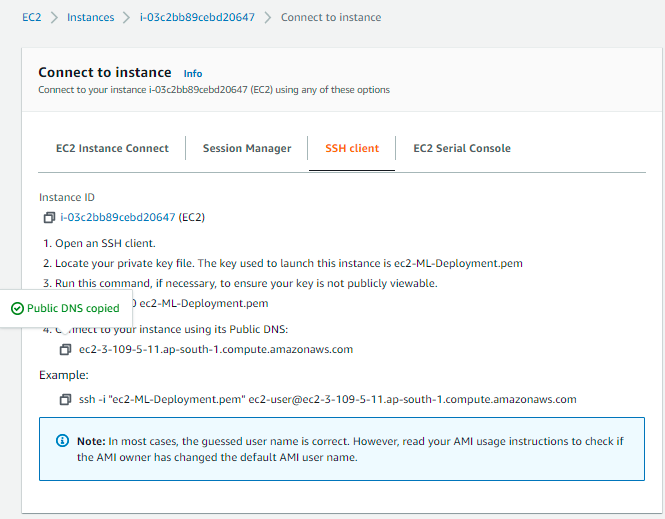
Above is FlaskML2 code………

=====\*\*\*\*\*\*\*\*\*\*\*\*=========

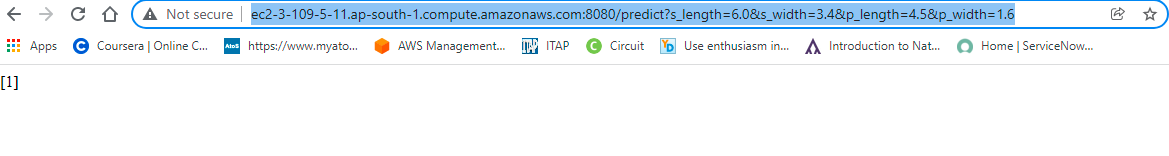




Click on Connect…. 😊



http://[ec2-13-127-234-4.ap-south-1.compute.amazonaws.com/predict?s\_length=6.0&s\_width=3.4&p\_length=4.5&p\_width=1.6](http://ec2-3-109-5-11.ap-south-1.compute.amazonaws.com:8080/predict?s_length=6.0&s_width=3.4&p_length=4.5&p_width=1.6)



Congratulations…. 😊