#Author : Authors of Learn & Practice AI by Projects

#Purpose : Streamlit application with UI components to serve the user request for IRIS model's prediction request.

#import libraries

import streamlit as st

import joblib

import json

#model file name

joblib\_file = "Iris\_Classifer.jbl"

#load the model from the disk

classification\_model = joblib.load(joblib\_file)

#function to predict the class using given prameter values

def predict\_class(sepal\_length:float, sepal\_width:float, petal\_length:float, petal\_width:float):

#print the arguments' value on the command prompt

print("sepal\_length : {} , sepal\_width: {}, patal\_length : {}, petal\_width : {}".format(sepal\_length, sepal\_width, petal\_length, petal\_width))

#predict the class for the given arguemnt

result = classification\_model.predict([[sepal\_length, sepal\_width, petal\_length, petal\_width]])

# print the predicted result(json format)

print(json.dumps({"predicted-result": result[0]}))

# return the predicted result(json format)

return (json.dumps({"predicted-result": result[0]}))

def show\_UI():

#UI title

st.title("IRIS Flower Classification");

#text boxes for user input

sepal\_length = st.text\_input('septal-length',0.0)

sepal\_width = st.text\_input('septal-width', 0.0)

petal\_length = st.text\_input('petal-length', 0.0)

petal\_width = st.text\_input('petal-width', 0.0)

#button to call predict\_class() function to predict the result.

if st.button("Predict Class"):

predicted\_class = predict\_class(sepal\_length, sepal\_width, petal\_length, petal\_width)

#show the predicted result.

st.success(predicted\_class)

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

#call show\_UI function to show the UI

show\_UI()