Zhuoming Yang
Batch Code:
LISUM14

Submission Date Oct 28 2022

Submitted to: github

Import csv file “iris”
import sklearn package – standardize data and training model
make pickle file of our model

```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import pickle
# load the csv file
df = pd.read_csv("iris.csv")


print(df.head())
# select independent and dependent variable
X = df[["Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width"]]
y = df["Class"]
# train test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Feature scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train, y_train)

# make pickle file of our model
pickle.dump(rf_classifier, open("model.pkl", "wb"))
```

Save the model
Deploy the model on flask
create app.py
running on
http://127.0.0.1:5000

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
# create flask app
app = Flask(__name__)

# load the pickle model
model = pickle.load(open("model.pkl","rb"))

@app.route("/")
def Home():
    return render_template("index.html")

@app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    return render_template("index.html", prediction_text = "The flowers species is {}".format((prediction)))

if __name__ == "__main__":
    app.run(debug=True)
```