Model Deployment using Flask

Name: Shalu Saroj

Batch code: LISUM30

Submission date: 08/13/2023

Submitted to: Data Glacier

Overview

- Deploying your basic machine learning model
- Learn how to use Flask to deploy a machine learning model into production
- Model deployment is a core topic in data scientist interviews – so start learning!

Abstract

This project has been written for the beginners of model deployment. With a simple linear regression example, a model was created on Spyder using Flask.

Table of Contents:

What is Model Deployment?

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. In this way, we turn the model we have created into a product. At the same time, we offer the product to the user side.

What is Flask?

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies, and several common framework-related tools. The only feature that distinguishes Flask from other frameworks is that it is very easy to use.

Installing Flask on your Machine

Installing Flask is simple and straightforward. I generally use pip installed.

```
# If you are using pip
$ pip install flask

# For Linux
$ sudo apt-get install python3-flask
```

If you want to work with the latest Flask code before it’s released, install or update the code from the master branch:

```
# Living on the edge
$ pip install -U https://github.com/pallets/flask/archive/master.tar.gz
```

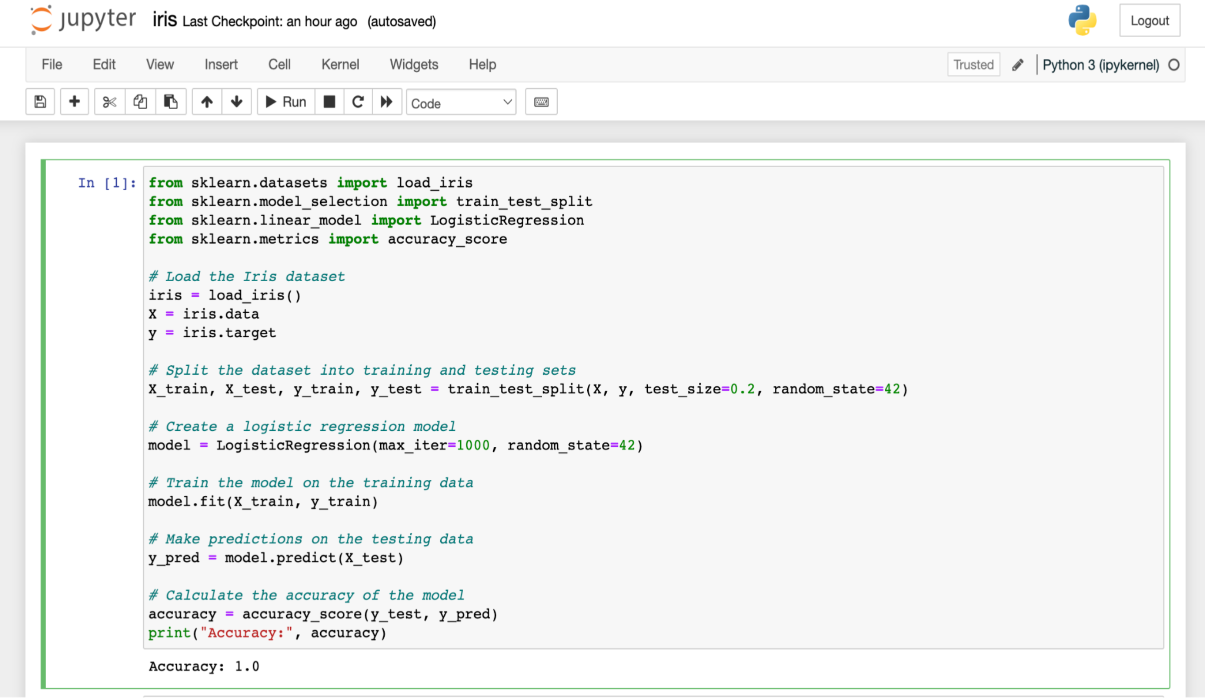
That's it. We are ready to deploying your machine learning model.

Setting up the Project WorkFlow

1. Model Building
2. Save the model and setup app
3. Webpage Template
4. Predict class and send results

Build Machine Learning Model

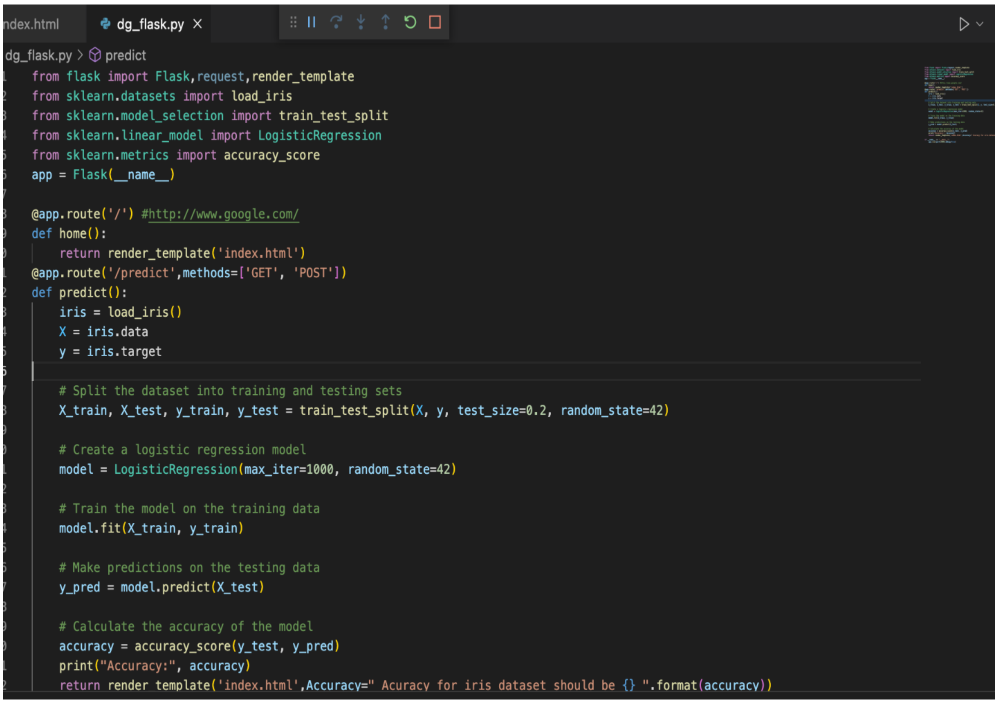
I prefer to work on Jupyter Notebook. — Our dataset has 25 rows and 2 columns. Let's take a look at what our dataset actually looks like. To do this, use the head() method:



```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a logistic regression model
model = LogisticRegression(max_iter=1000, random_state=42)

# Train the model on the training data
model.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

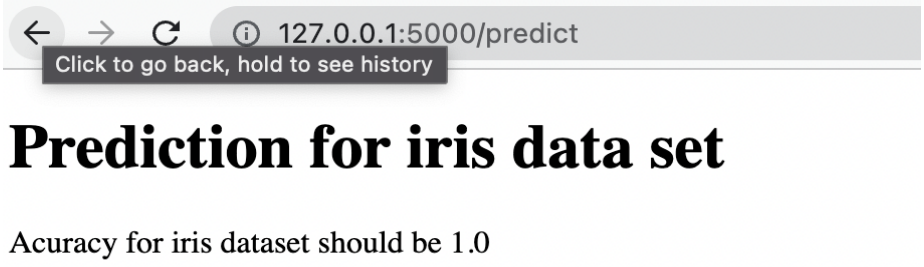
# DEPLOYING ON FLASK:



```python
from flask import Flask,request,render_template
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
app = Flask(__name__)

@app.route('/') #http://www.google.com/
def home():
    return render_template('index.html')
@app.route('/predict',methods=['GET', 'POST'])
def predict():
    iris = load_iris()
    X = iris.data
    y = iris.target

    # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Create a logistic regression model
    model = LogisticRegression(max_iter=1000, random_state=42)

    # Train the model on the training data
    model.fit(X_train, y_train)

    # Make predictions on the testing data
    y_pred = model.predict(X_test)

    # Calculate the accuracy of the model
    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy)
    return render_template('index.html',Accuracy=" Acuracy for iris dataset should be {} ".format(accuracy))
```

Working of the Deployed Model

We have successfully started the Flask server! Open your browser and go to this address – http://127.0.0.1[redacted]1080/. You will see that the Flask server has rendered the default template.



Prediction for iris data set

Acuracy for iris dataset should be 1.0