Name: Zhanqiu Guo

Batch Code: LISUM19

Submission Date: 2023-3-20

Submitted To: https://github.com/Zhanqiu-Guo/Data-Glacier-Projects.git

Deploy Model:

app.py create_pdf.py

app.py > ...

```python
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression # importing Sklearn's logistic regression's module
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import numpy as np
from sklearn.linear_model import LogisticRegression
import joblib
import pandas as pd
from flask import Flask, jsonify, request, render_template

iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.25, random_state=42)
scale=StandardScaler()
X_train = scale.fit_transform(X_train)
model = LogisticRegression()
model.fit(X_train, y_train)

# Save the model to disk
filename = 'finalized_model.sav'
joblib.dump(model, filename)

app = Flask(__name__)

# Load the saved model
model = joblib.load('finalized_model.sav')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    dict = {0: "Iris-setosa", 1: "Iris-versicolor", 2: "Iris-virginica"}
    # Get input from user and make prediction using loaded model
    float_features = [float(x) for x in request.form.values()]
    final_features = [np.array(float_features)]
    prediction  = model.predict(final_features)
    output = round(prediction[0], 2)
    return render_template('index.html', prediction_text='The flower is most likely to be {}'.format(dict[output]))

if __name__ == '__main__':
    app.run(port=5000, debug=True)
    app.run(debug=True, host="0.0.0.0", port=8989)
```

Create PDF:

create_pdf.py

```python
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Image
from reportlab.lib.styles import getSampleStyleSheet
def create_pdf(name, batch_code, submission_date, submitted_to, image_paths):
    doc = SimpleDocTemplate("output.pdf", pagesize=letter)
    styles = getSampleStyleSheet()
    story = []

    ptext = f'Name: {name}'
    story.append(Paragraph(ptext, styles["Normal"]))
    story.append(Spacer(1, 12))

    ptext = f'Batch Code: {batch_code}'
    story.append(Paragraph(ptext, styles["Normal"]))
    story.append(Spacer(1, 12))

    ptext = f'Submission Date: {submission_date}'
    story.append(Paragraph(ptext, styles["Normal"]))
    story.append(Spacer(1, 12))

    ptext = f'Submitted To: {submitted_to}'
    story.append(Paragraph(ptext, styles["Normal"]))
    story.append(Spacer(1, 12))

    ptext = f'Deploy Model: '
    story.append(Paragraph(ptext, styles["Normal"]))
    story.append(Spacer(1, 12))
    im1 = Image(image_paths[0])
    im1._restrictSize(600,600)
    story.append(im1)

    ptext = f'Create PDF: '
    story.append(Paragraph(ptext, styles["Normal"]))
    story.append(Spacer(1, 12))
    im2 = Image(image_paths[1])
    im2._restrictSize(600,600)
    story.append(im2)

    doc.build(story)

create_pdf(
    "Zhanqiu Guo",
    "LISUM19",
    "2023-3-20",
    "https://github.com/Zhanqiu-Guo/Data-Glacier-Projects.git",
    ["Deploy_Model.png", "Create_PDF.png"]
)
```