

Data Intake Report

Name: Elias Sharonov
Report date: 07/05/2024
Internship Batch: LISUM34
Submitted to: GitHub

Task:

1. Select any toy data (simple data).

The Iris Dataset

This data sets consists of 3 different types of irises' (Setosa, Versicolour, and Virginica) petal and sepal length, stored in a 150x4 numpy.ndarray

The rows being the samples and the columns being: Sepal Length, Sepal Width, Petal Length and Petal Width.

The below plot uses the first two features. See [here](#) for more information on this dataset.

```
# Code source: Gaël Varoquaux  
# Modified for documentation by Jaques Grobler  
# License: BSD 3 clause
```

Loading the iris dataset




```
from sklearn import datasets  
  
iris = datasets.load_iris()
```

Scatter Plot of the Iris dataset

```
import matplotlib.pyplot as plt  
  
_, ax = plt.subplots()  
scatter = ax.scatter(iris.data[:, 0], iris.data[:, 1], c=iris.target)  
ax.set(xlabel=iris.feature_names[0], ylabel=iris.feature_names[1])  
_ = ax.legend(  
    scatter.legend_elements()[0], iris.target_names, loc="lower right", title="Classes"  
)
```

2. Save the model

```
C: > Users > ilyav > source > repos > DataGlacier Stuff > DataGlacier Week 4 > Week4.py > ...
1  from sklearn.datasets import load_iris
2  from sklearn.model_selection import train_test_split
3  from sklearn.linear_model import LogisticRegression
4  import joblib
5
6  iris = load_iris()
7  X, y = iris.data, iris.target
8
9  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
10
11 model = LogisticRegression(max_iter=200)
12 model.fit(X_train, y_train)
13
14 joblib.dump(model, 'iris_model.pkl')
```

 FlaskDeployment.py	07/05/24 20:10	Python source file	2 KB
 iris_model.pkl	07/05/24 19:29	PKL File	1 KB
 Week4.py	07/05/24 19:46	Python source file	1 KB

3. Deploy the model on flask (web app)

```
from flask import Flask, request
import joblib
import numpy as np
from sklearn.datasets import load_iris

iris = load_iris()

model = joblib.load('iris_model.pkl')

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        sepal_length = float(request.form['sepal_length'])
        sepal_width = float(request.form['sepal_width'])
        petal_length = float(request.form['petal_length'])
        petal_width = float(request.form['petal_width'])

        prediction = model.predict(np.array([[sepal_length, sepal_width, petal_length, petal_width]]))

        species = iris.target_names[prediction[0]]

        return f'<h1>Predicted Iris Species: {species}</h1>'

    return '''
    <form method="post">
        <title> Iris Classifier </title>
        <h1> Iris Classifier </h1>
        <label for="sepal_length">Sepal Length:</label>
        <input type="text" id="sepal_length" name="sepal_length"><br><br>
        <label for="sepal_width">Sepal Width:</label>
        <input type="text" id="sepal_width" name="sepal_width"><br><br>
        <label for="petal_length">Petal Length:</label>
        <input type="text" id="petal_length" name="petal_length"><br><br>
        <label for="petal_width">Petal Width:</label>
        <input type="text" id="petal_width" name="petal_width"><br><br>
        <input type="submit" value="Run the Model">
    </form>
    '''

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

Classifier

Predicted Iris Species: virginica