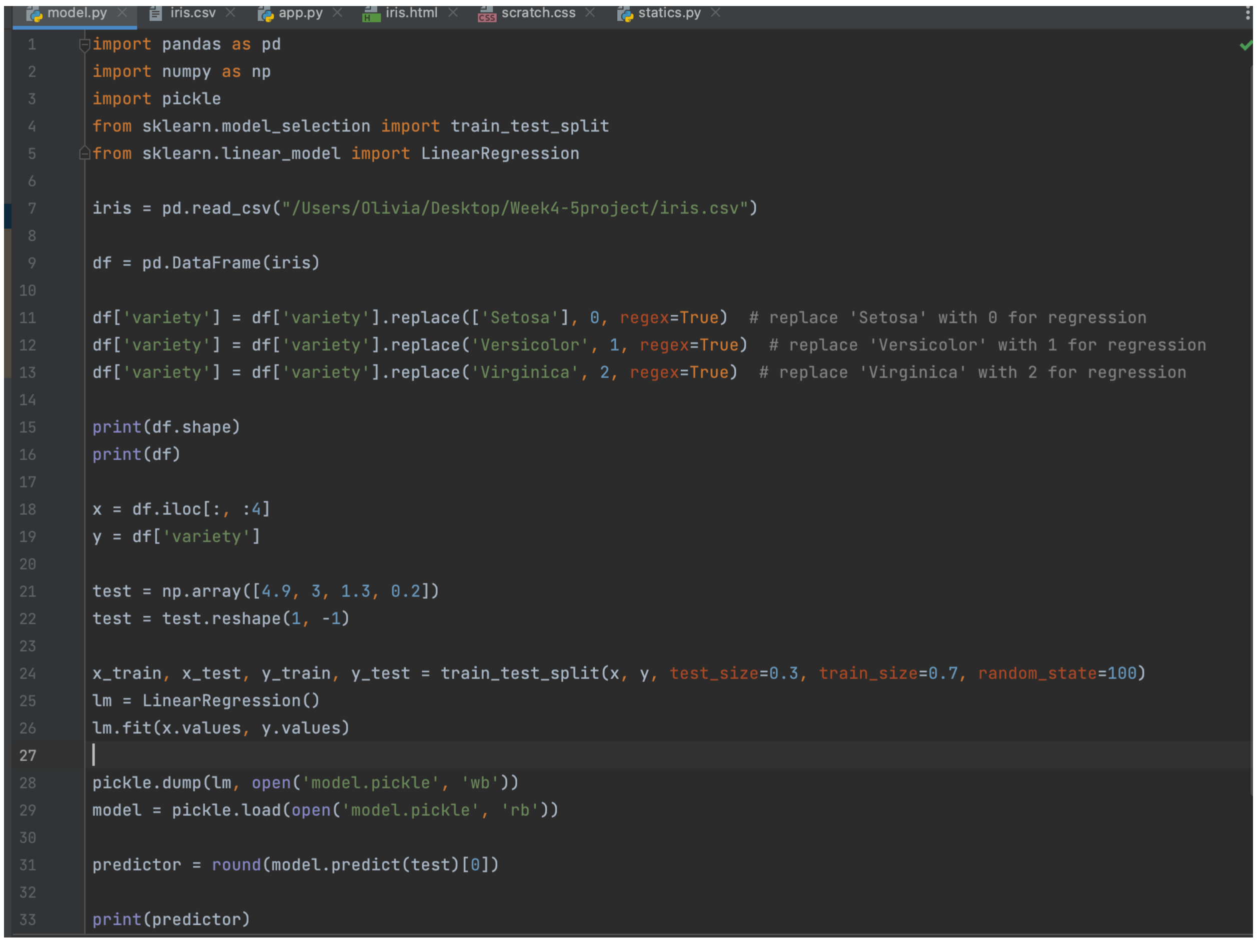
Name: Week 4: Data Intake on API
Report date: 1/9/2022
Internship Batch: LISUM16
Version: 1.0
Data intake by: Olivia Foster
Data intake reviewer: Data Glacier
Data storage location: https://github.com/LiviaNFoster/Week4-5

Data:
The data was pulled from a toy data set from the famous Iris dataset.

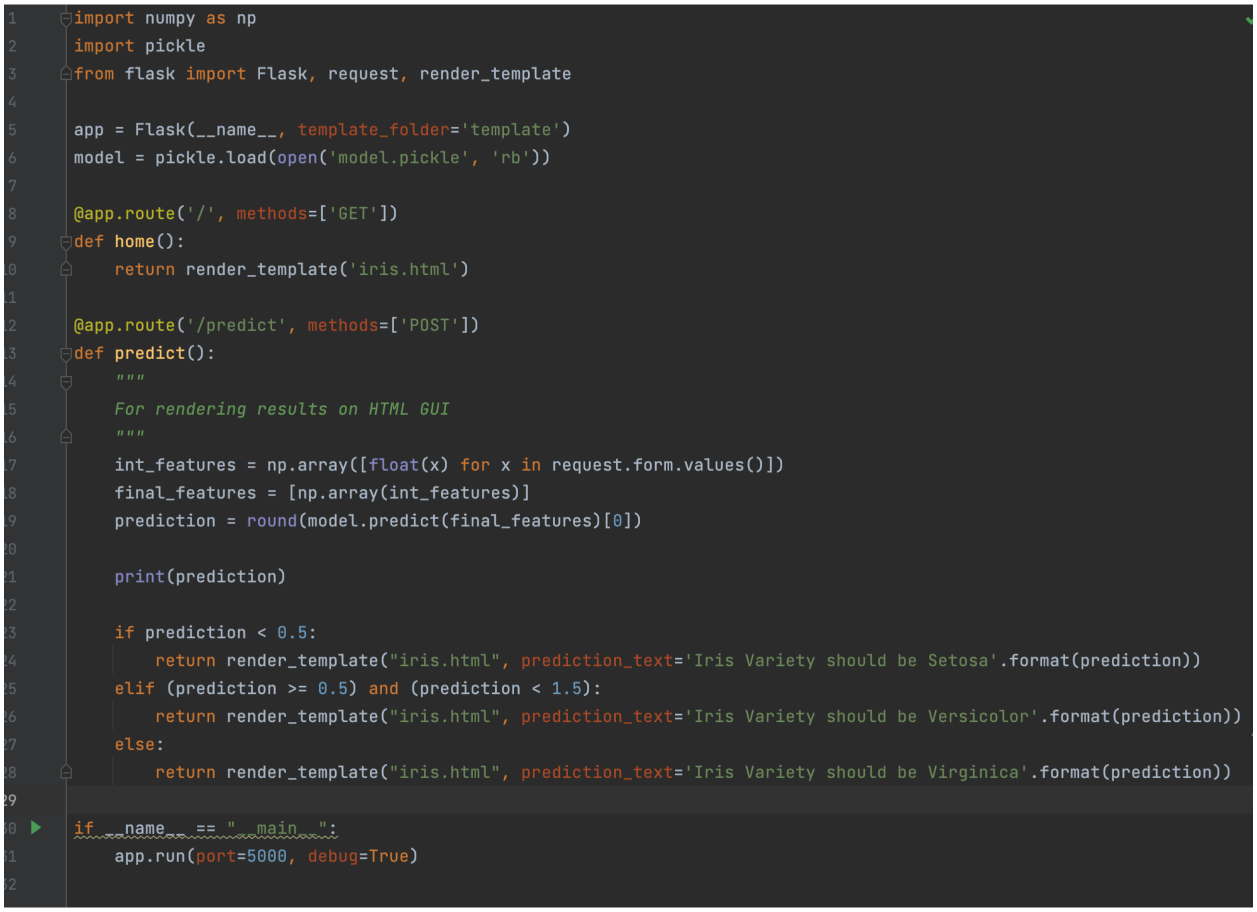
Total number of observations	150
Total number of files	1
Total number of features	5
Base format of the file	csv
Size of the data	3,975 bytes

Model.py:



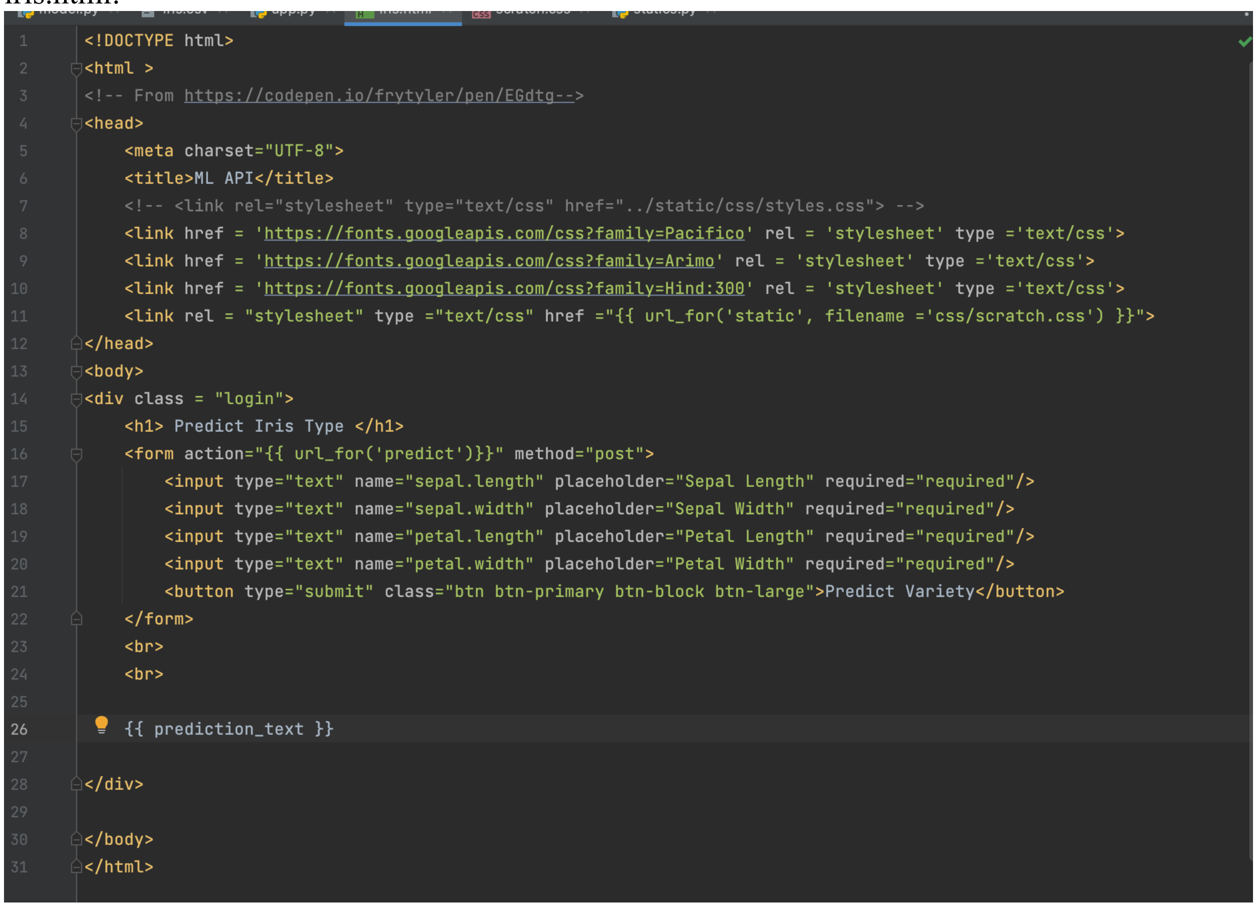
```python
import pandas as pd
import numpy as np
import pickle
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

iris = pd.read_csv("/Users/Olivia/Desktop/Week4-5project/iris.csv")

df = pd.DataFrame(iris)

df['variety'] = df['variety'].replace(['Setosa'], 0, regex=True)  # replace 'Setosa' with 0 for regression
df['variety'] = df['variety'].replace('Versicolor', 1, regex=True)  # replace 'Versicolor' with 1 for regression
df['variety'] = df['variety'].replace('Virginica', 2, regex=True)  # replace 'Virginica' with 2 for regression

print(df.shape)
print(df)

x = df.iloc[:, :4]
y = df['variety']

test = np.array([4.9, 3, 1.3, 0.2])
test = test.reshape(1, -1)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, train_size=0.7, random_state=100)
lm = LinearRegression()
lm.fit(x.values, y.values)

pickle.dump(lm, open('model.pickle', 'wb'))
model = pickle.load(open('model.pickle', 'rb'))

predictor = round(model.predict(test)[0])

print(predictor)
```

app.py:



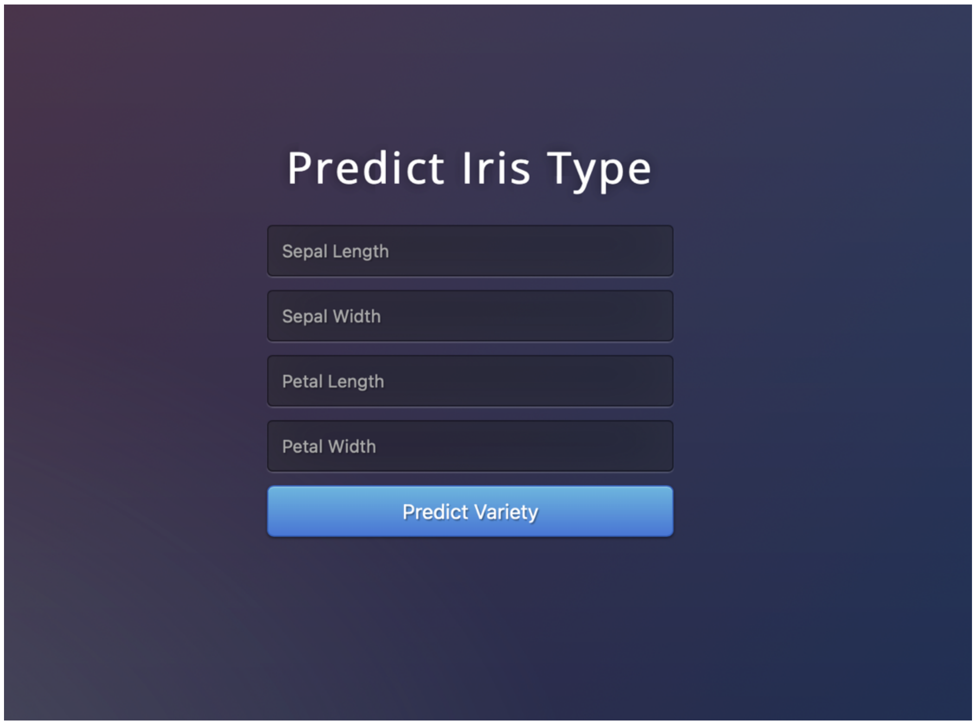
```python
import numpy as np
import pickle
from flask import Flask, request, render_template

app = Flask(__name__, template_folder='template')
model = pickle.load(open('model.pickle', 'rb'))

@app.route('/', methods=['GET'])
def home():
    return render_template('iris.html')

@app.route('/predict', methods=['POST'])
def predict():
    """
    For rendering results on HTML GUI
    """
    int_features = np.array([float(x) for x in request.form.values()])
    final_features = [np.array(int_features)]
    prediction = round(model.predict(final_features)[0])

    print(prediction)

    if prediction < 0.5:
        return render_template("iris.html", prediction_text='Iris Variety should be Setosa'.format(prediction))
    elif (prediction >= 0.5) and (prediction < 1.5):
        return render_template("iris.html", prediction_text='Iris Variety should be Versicolor'.format(prediction))
    else:
        return render_template("iris.html", prediction_text='Iris Variety should be Virginica'.format(prediction))

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

iris.html:



```
<!DOCTYPE html>
<html >
<!-- From https://codepen.io/frytyler/pen/EGdtg-->
<head>
    <meta charset="UTF-8">
    <title>ML API</title>
    <!-- <link rel="stylesheet" type="text/css" href="../static/css/styles.css"> -->
    <link href = 'https://fonts.googleapis.com/css?family=Pacifico' rel = 'stylesheet' type ='text/css'>
    <link href = 'https://fonts.googleapis.com/css?family=Arimo' rel = 'stylesheet' type ='text/css'>
    <link href = 'https://fonts.googleapis.com/css?family=Hind:300' rel = 'stylesheet' type ='text/css'>
    <link rel = "stylesheet" type ="text/css" href ="{{ url_for('static', filename ='css/scratch.css') }}">
</head>
<body>
<div class = "login">
    <h1> Predict Iris Type </h1>
    <form action="{{ url_for('predict')}}" method="post">
        <input type="text" name="sepal.length" placeholder="Sepal Length" required="required"/>
        <input type="text" name="sepal.width" placeholder="Sepal Width" required="required"/>
        <input type="text" name="petal.length" placeholder="Petal Length" required="required"/>
        <input type="text" name="petal.width" placeholder="Petal Width" required="required"/>
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict Variety</button>
    </form>
    <br>
    <br>

    {{ prediction_text }}

</div>

</body>
</html>
```

Deployment:

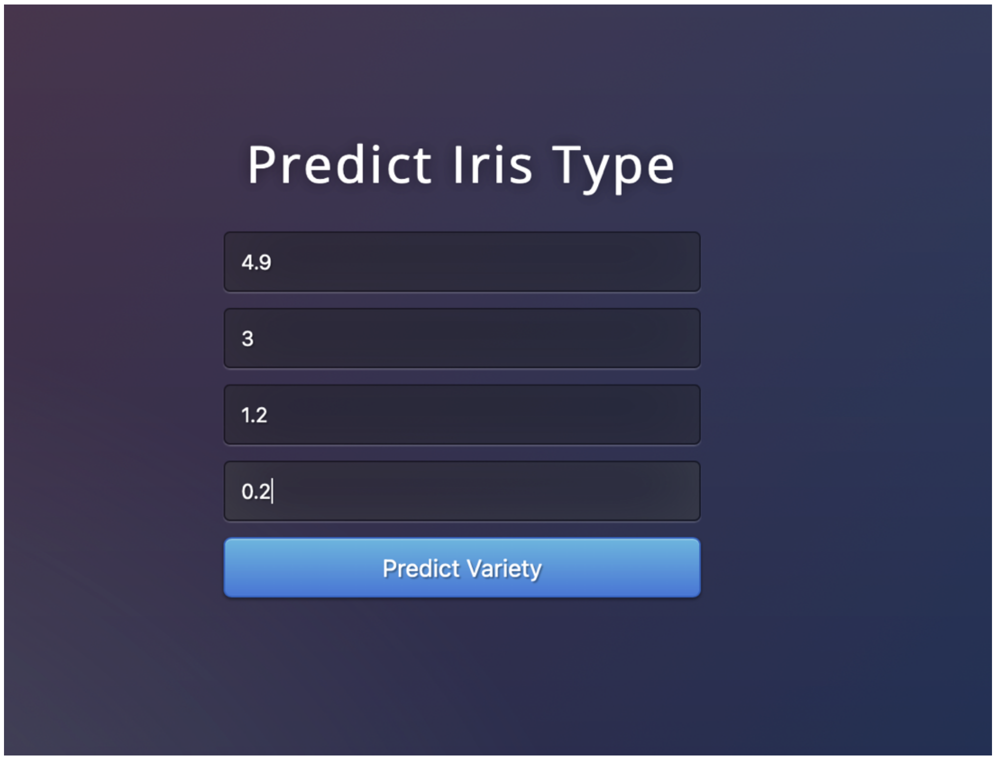

Predict Iris Type

Sepal Length

Sepal Width

Petal Length

Petal Width

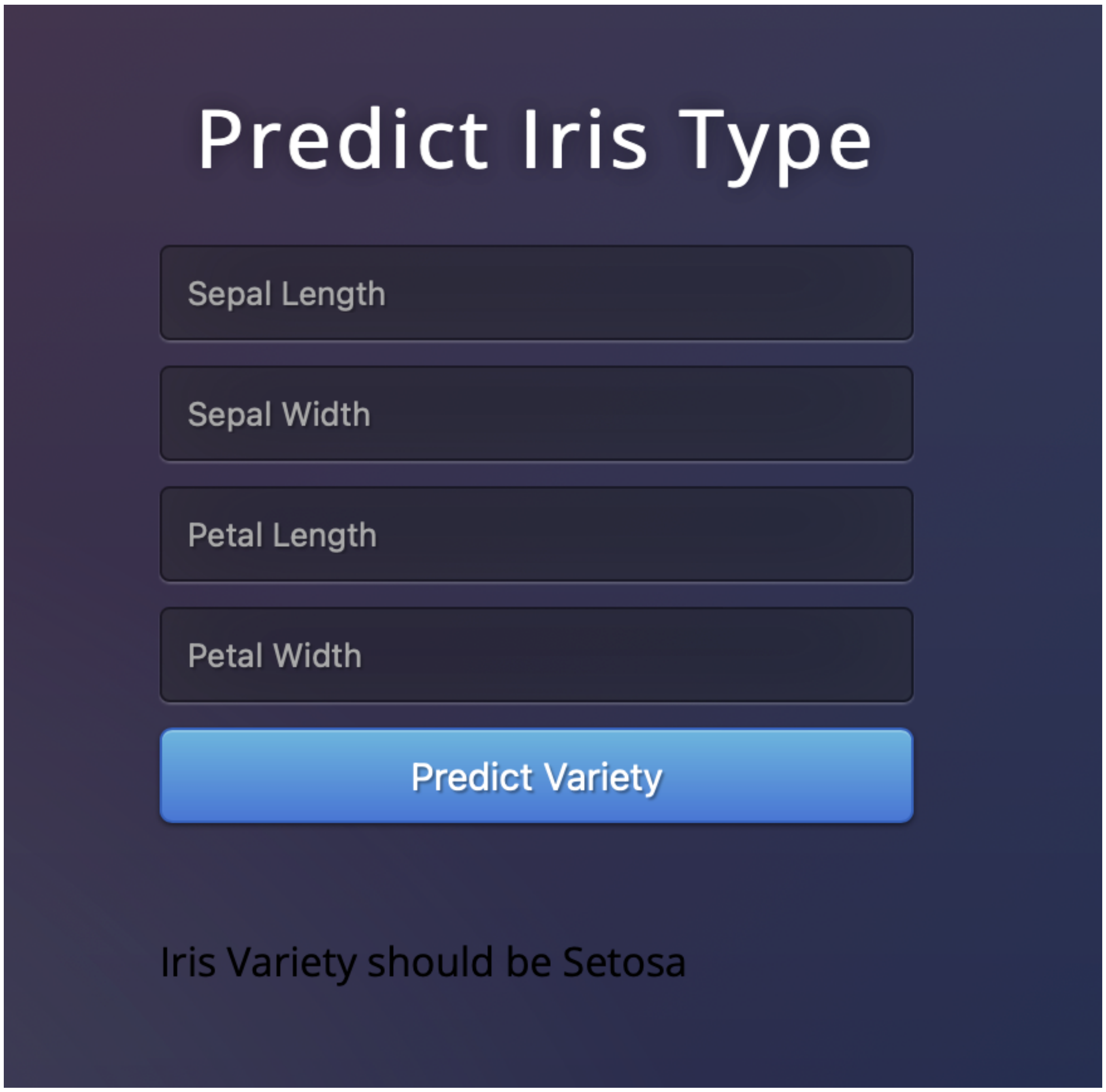

Predict Iris Type

4.9

3

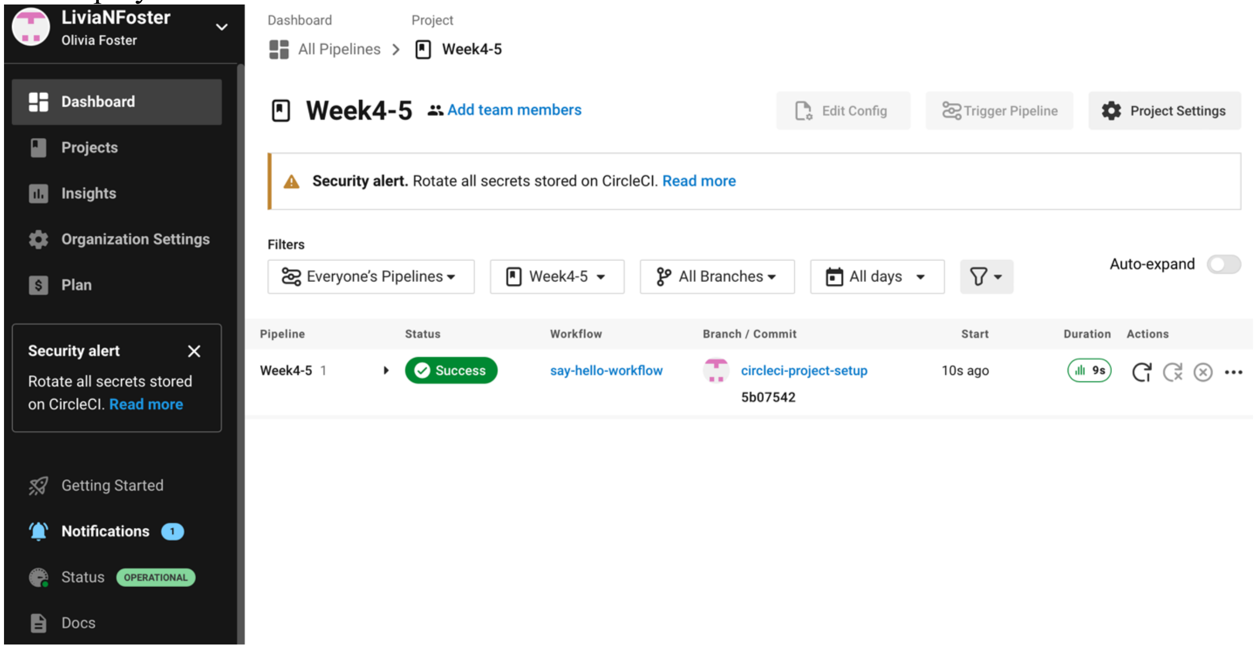
1.2

0.2



Iris Variety should be Setosa

API deployment on CircleCi:



Security alert
Rotate all secrets stored on CircleCI. Read more

Week4-5

Security alert. Rotate all secrets stored on CircleCI. Read more

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
Week4-5 1	Success	say-hello-workflow	circleci-project-setup 5b07542	10s ago	9s	