From given data predict that animal is dog or cat by weight and height.

Data.csv

```
Height,Weight,Species
88.9,48.3,Dog
90.2,47.4,Dog
82.7,44.8,Dog
81.4,48.2,Dog
83.5,39.9,Dog
76.4,35.4,Cat
82.3,40.4,Cat
81.1,39.9,Cat
79.9,43.2,Cat
82.4,45.7,Cat
```

App.py

```
from flask import Flask, render_template, request, jsonify
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import joblib

app = Flask(__name__)

# Load your CSV data
data = pd.read_csv('data.csv')

# Features (height, weight)
X = data[['Height', 'Weight']]

# Target variable (species)
y = data['Species']

# Split the data 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)

# Train a decision tree classifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Model Accuracy: {accuracy * 100:.2f}%')

# Save the trained model to a file
model_filename = 'model.joblib'
joblib.dump(model, model_filename)

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

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get user input from the form
        height = float(request.form['height'])
        weight = float(request.form['weight'])

        # Use the trained model to make predictions
        user_input = [[height, weight]]
        prediction = model.predict(user_input)[0]

        return render_template('result.html', prediction=prediction)
    except Exception as e:
        error_message = f"An error occurred: {str(e)}"
        return render_template('error.html',
error_message=error_message)

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

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Cat or Dog Predictor</title>
  </head>
  <body>
    <h1>Cat or Dog Predictor</h1>

    <form action="/predict" method="post">
      <label for="height">Enter Height:</label>
      <input type="text" id="height" name="height" required />
      <br />
      <label for="weight">Enter Weight:</label>
      <input type="text" id="weight" name="weight" required />
      <br />
      <button type="submit">Predict</button>
    </form>
  </body>
</html>
```

Result.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Result</title>
  </head>
  <body>
    <h1>Result</h1>

    <p>The predicted species is: <strong>{{ prediction }}</strong></p>
  </body>
</html>
```

Error.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Error</title>
  </head>
  <body>
    <h1>Error</h1>

    <p>{{ error_message }}</p>
  </body>
</html>
```

Output