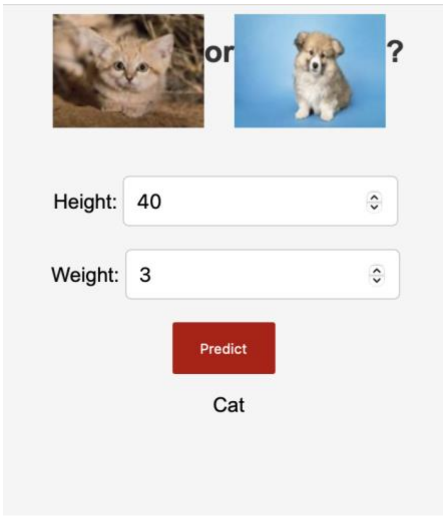
Zongdao Wen

LISUM21

Submission Date: May 27 2023

Submitted to: Data Glacier

Web App name: Cat or Dog?



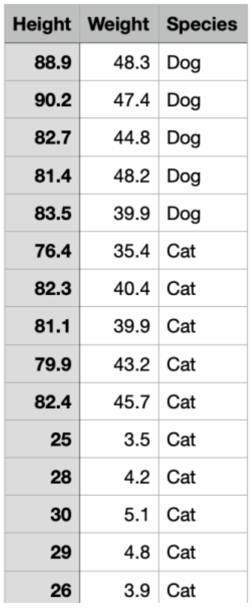
Height: 40

Weight: 3

Cat

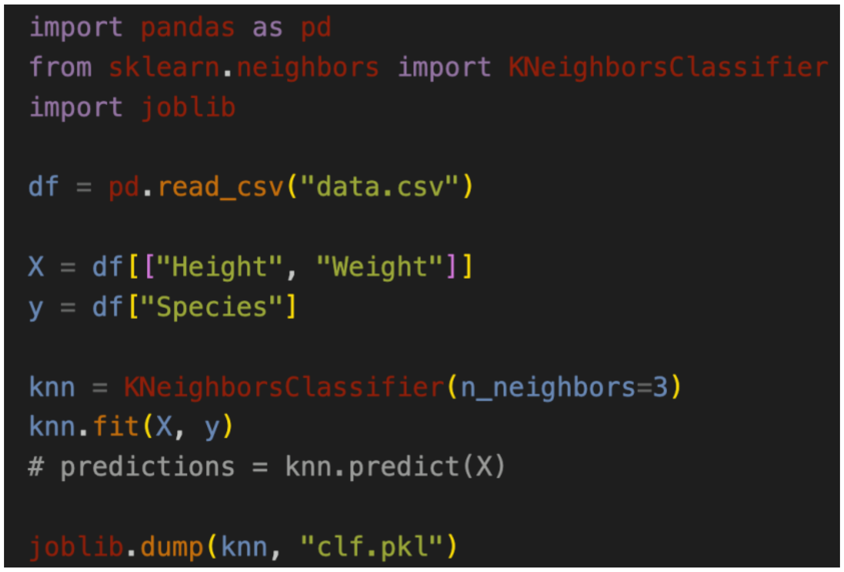
The app can tell if the species is either a cat or dog base on height(cm) and weight(kg).

Data: 200 entries of height and weight with labeled either cat or dog.



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
25	3.5	Cat
28	4.2	Cat
30	5.1	Cat
29	4.8	Cat
26	3.9	Cat

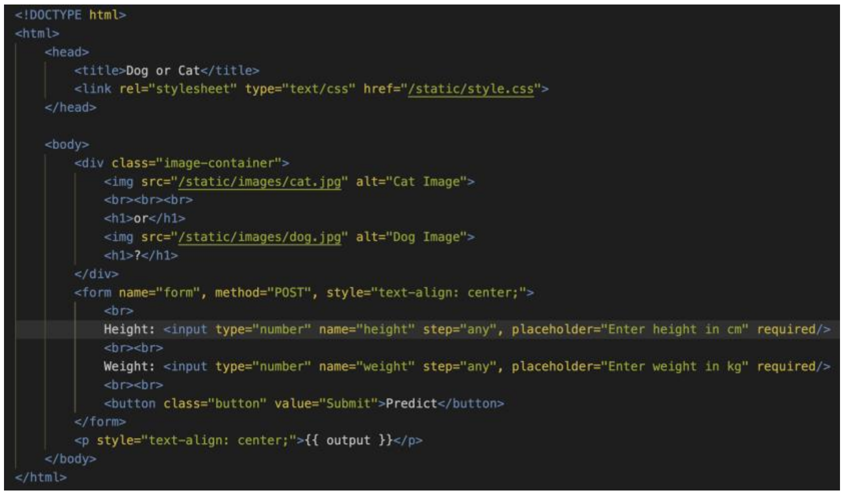
model.py



```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
import joblib

df = pd.read_csv("data.csv")

X = df[["Height", "Weight"]]
y = df["Species"]

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X, y)
# predictions = knn.predict(X)

joblib.dump(knn, "clf.pkl")
```

KNN is trained and used to predict the species base on height and weight. The model is then saved as a pkl file.

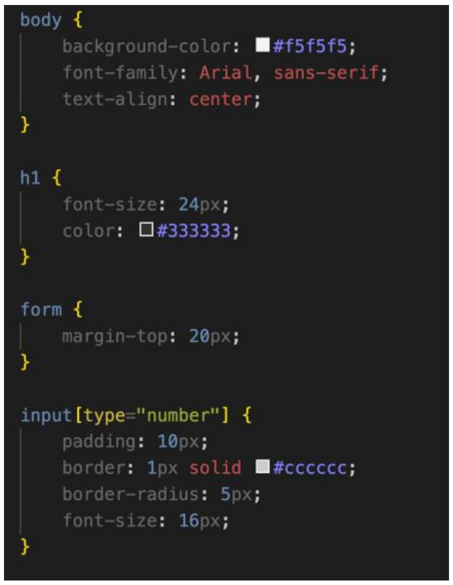
website.html



```
<!DOCTYPE html>
<html>
    <head>
        <title>Dog or Cat</title>
        <link rel="stylesheet" type="text/css" href="/static/style.css">
    </head>

    <body>
        <div class="image-container">
            <img src="/static/images/cat.jpg" alt="Cat Image">
            <br><br><br>
            <h1>or</h1>
            <img src="/static/images/dog.jpg" alt="Dog Image">
            <h1>?</h1>
        </div>
        <form name="form", method="POST", style="text-align: center;">
            <br>
            Height: <input type="number" name="height" step="any", placeholder="Enter height in cm" required/>
            <br><br>
            Weight: <input type="number" name="weight" step="any", placeholder="Enter weight in kg" required/>
            <br><br>
            <button class="button" value="Submit">Predict</button>
        </form>
        <p style="text-align: center;">{{ output }}</p>
    </body>
</html>
```

Contains the html code for the web app.

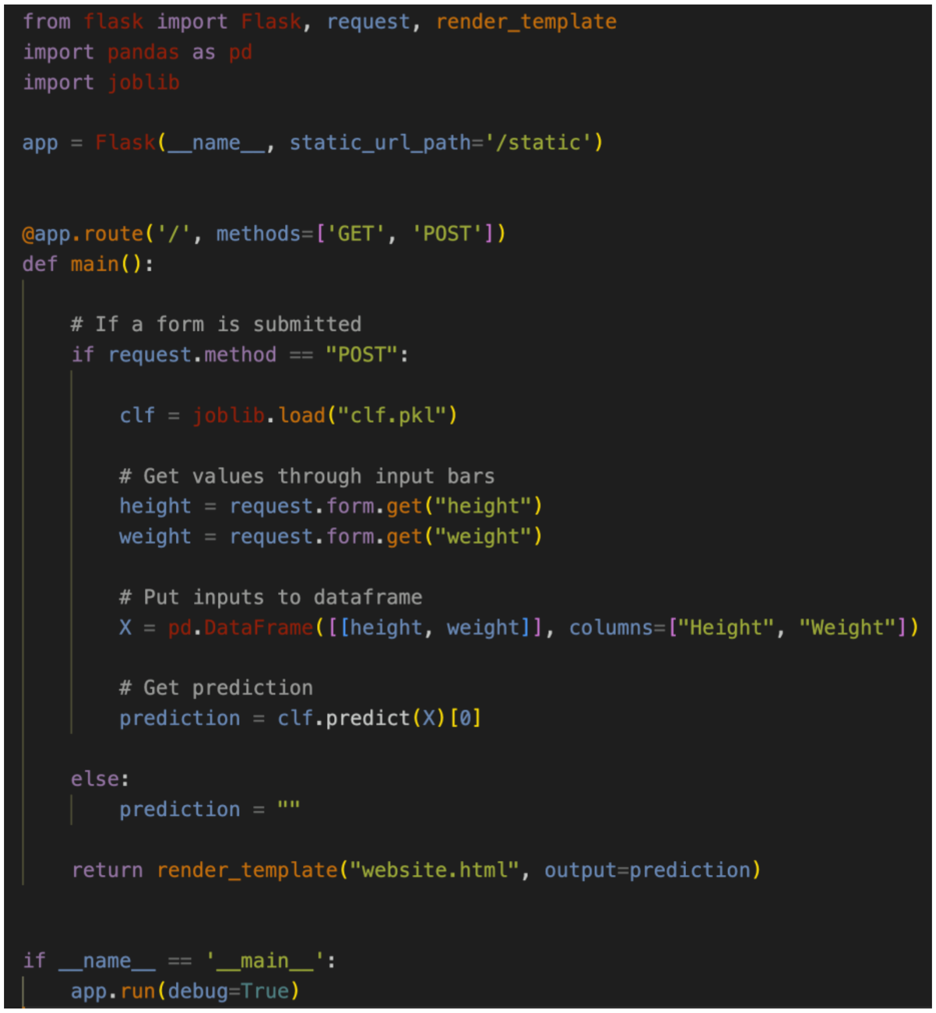
style.css



```css
body {
    background-color: #f5f5f5;
    font-family: Arial, sans-serif;
    text-align: center;
}

h1 {
    font-size: 24px;
    color: #333333;
}

form {
    margin-top: 20px;
}

input[type="number"] {
    padding: 10px;
    border: 1px solid #cccccc;
    border-radius: 5px;
    font-size: 16px;
}
```

Contains the style of the UI.

app.py.



```
from flask import Flask, request, render_template
import pandas as pd
import joblib

app = Flask(__name__, static_url_path='/static')


@app.route('/', methods=['GET', 'POST'])
def main():

    # If a form is submitted
    if request.method == "POST":

        clf = joblib.load("clf.pkl")

        # Get values through input bars
        height = request.form.get("height")
        weight = request.form.get("weight")

        # Put inputs to dataframe
        X = pd.DataFrame([[height, weight]], columns=["Height", "Weight"])

        # Get prediction
        prediction = clf.predict(X)[0]

    else:
        prediction = ""

    return render_template("website.html", output=prediction)


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

The inputs are fetched and fed into the trained model in mode.py and the result is saved as prediction. The prediction is then passed to the website.html using render_template from flask.