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

app = Flask(__name__)

model = pickle.load(open('breast_cancer_detector.pickle', 'rb'))


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


@app.route('/predict1', methods=['POST'])
def predict1():
    input_features = [float(x) for x in request.form.values()]
    # features_value=np.array(input_features)
    # print(features_value)
    # print(input_features)
    features_name = {'mean radius': input_features[0], 'mean texture': input_f
eatures[1],'mean perimeter': input_features[2], 'mean area': input_features[3]
,'mean smoothness': input_features[4], 'mean compactness': input_features[5],
'mean concavity': input_features[6],'mean concave points': input_features[7],
'mean symmetry': input_features[8], 'mean fractal dimension': input_features[9
],'radius error': input_features[10], 'texture error': input_features[11],'per
imeter error': input_features[12], 'area error': input_features[13],'smoothnes
s error': input_features[14], 'compactness error': input_features[15],'concavi
ty error': input_features[16],'concave points error': input_features[17], 'sym
metry error': input_features[18], 'fractal dimension error': input_features[19
],'worst radius': input_features[20], 'worst texture': input_features[21], 'wo
rst perimeter': input_features[22], 'worst area': input_features[23],'worst sm
oothness': input_features[24], 'worst compactness': input_features[25], 'worst
 concavity': input_features[26],'worst concave points': input_features[27], 'w
orst symmetry': input_features[28], 'worst fractal dimension': input_features[
29]}

    df = pd.DataFrame(features_name,index=[0])
    print(df)
    output = model.predict(df)

    if output == 0:
        res_val = "** breast cancer **"
    else:
        res_val = "** no breast cancer **"

    return render_template('index.html', prediction_text='Patient has{}'.forma
t(res_val))
```

```python
if __name__ == "__main__":
    app.run()
```