

Data Intake Report

Name: Healthcare – Persistency of a Drug

Report date: 8/21/2023

Internship Batch: LISUM22

Version: 1.0

Data intake by: Farzana Chowdhury

Tabular data details: Healthcare_data

Total number of observations (rows)	3424
Total number of files	1
Total number of features (columns)	69
Base format of the file	.xlsx
Size of the data	887 KB (908,966 bytes)

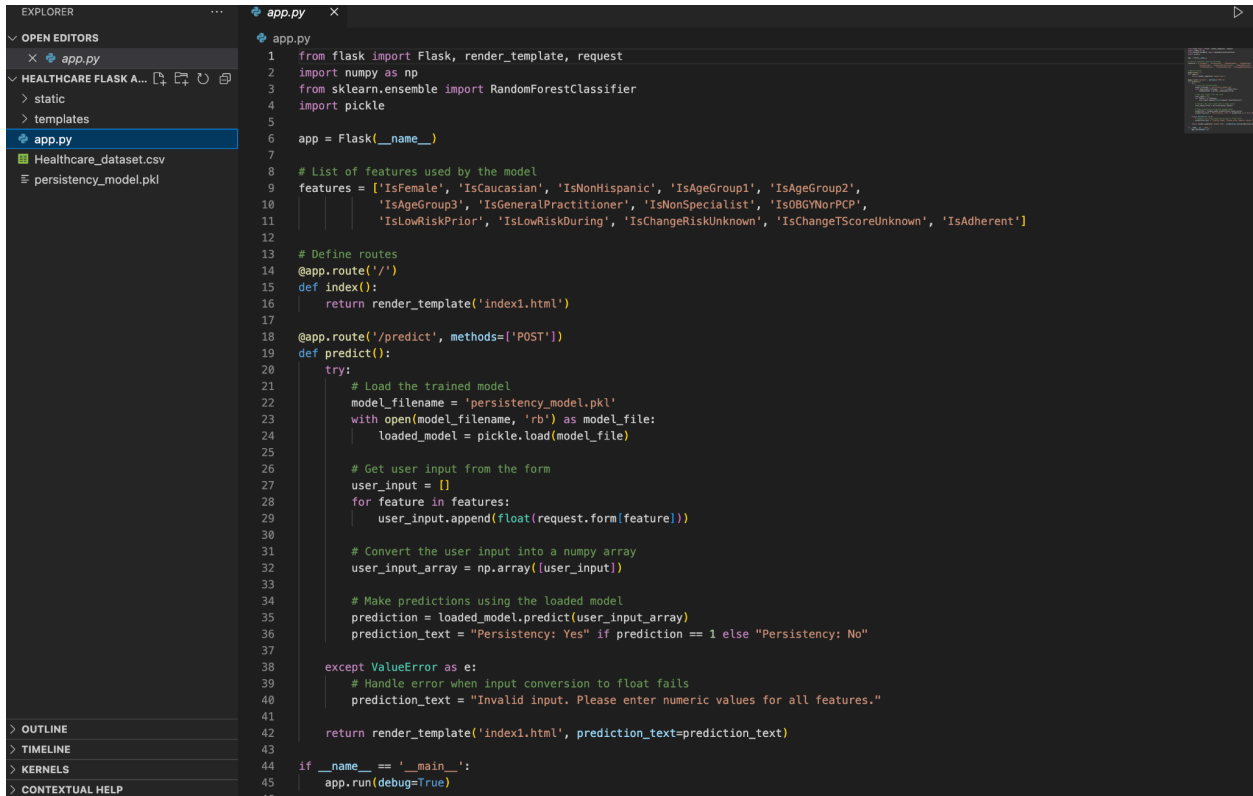
Using isnull(): Are there any null/missing values in this dataset? No

Using duplicated(): Are there any duplicates in this dataset? No

Using dtypes(): What are the datatypes in this dataset?

2 integer (int64) columns, 67 object columns

Flask deployment on the persistency model:

A screenshot of a code editor with a dark theme. The left sidebar shows the 'EXPLORER' view with a file tree containing 'app.py', 'Healthcare_dataset.csv', and 'persistency_model.pkl'. The main editor area shows the code for 'app.py'. The code imports Flask, numpy, RandomForestClassifier, and pickle. It defines a Flask app and a list of features. It has two routes: a root route for an index page and a '/predict' route for a POST request. The '/predict' route loads a model, gets user input, converts it to a numpy array, makes a prediction, and returns a response. It also includes an exception handler for ValueError.

```
1 from flask import Flask, render_template, request
2 import numpy as np
3 from sklearn.ensemble import RandomForestClassifier
4 import pickle
5
6 app = Flask(__name__)
7
8 # List of features used by the model
9 features = ['IsFemale', 'IsCaucasian', 'IsNonHispanic', 'IsAgeGroup1', 'IsAgeGroup2',
10            'IsAgeGroup3', 'IsGeneralPractitioner', 'IsNonSpecialist', 'IsOBGYNorPCP',
11            'IsLowRiskPrior', 'IsLowRiskDuring', 'IsChangeRiskUnknown', 'IsChangeTScoreUnknown', 'IsAdherent']
12
13 # Define routes
14 @app.route('/')
15 def index():
16     return render_template('index1.html')
17
18 @app.route('/predict', methods=['POST'])
19 def predict():
20     try:
21         # Load the trained model
22         model_filename = 'persistency_model.pkl'
23         with open(model_filename, 'rb') as model_file:
24             loaded_model = pickle.load(model_file)
25
26         # Get user input from the form
27         user_input = []
28         for feature in features:
29             user_input.append(float(request.form[feature]))
30
31         # Convert the user input into a numpy array
32         user_input_array = np.array(user_input)
33
34         # Make predictions using the loaded model
35         prediction = loaded_model.predict(user_input_array)
36         prediction_text = "Persistency: Yes" if prediction == 1 else "Persistency: No"
37
38     except ValueError as e:
39         # Handle error when input conversion to float fails
40         prediction_text = "Invalid input. Please enter numeric values for all features."
41
42     return render_template('index1.html', prediction_text=prediction_text)
43
44 if __name__ == '__main__':
45     app.run(debug=True)
```