Data Intake Report

Name: Ana Lilliam Recio Garcia

Report date: 10/24/2022 Internship Batch: LISUM14

Version:<1.0>

Data intake by: Ana Lilliam Recio Garcia

Data intake reviewer:

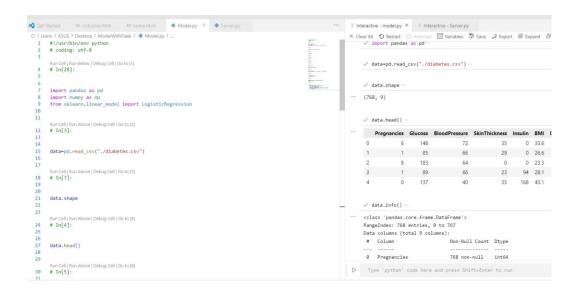
Data storage location: https://data-glacier-ana.herokuapp.com/

Tabular data details:

Total number of observations	768
Total number of files	1
Total number of features	9
Base format of the file	.csv
Size of the data	24KB

Proposed Approach:

I have created from the file diabetes.csv a model using Logistic Regression, to evaluate the dependent variable which is Outcome (last column) in the dataset, from the independent variables which are the other 7 variables.



```
Outcome_mappings = {0: "Not Diabetic", 1: "Diabetic"}
    Run Cell | Run Above | Debug Cell | Go to [9]
    # In[9]:
    X = data.iloc[:, 0:-1]
    y = data.iloc[:, -1]
    Run Cell | Run Above | Debug Cell | Go to [10]
    # In[10]:
8
    logreg = LogisticRegression(max iter=1000)
    logreg.fit(X, y)
    Run Cell | Run Above | Debug Cell | Go to [11]
    # In[12]:
    def outcome(a, b, c, d, e, f, g, h):
8
       arr = np.array([a, b, c, d, e, f, g, h])
9
        arr = arr.astype(np.float)
       query = arr.reshape(1, -1)
        prediction = Outcome_mappings[logreg.predict(query)[0]]
        return prediction
```

I've created a simple form in HTLM so the user can input on the variables and have an outcome

```
Luit Selection view do Nuit leminia Help
Get Started ◇ outcome.html ◇ home.html × ♣ Model.py ♣ Server.py
                                                                                                                                                                                                                                                                            □ 📴 🔲 ··· | ≡
mplates > • home.html > • body > • div#login-form-container > • form > • div.card > • div.card-content > • div.content > • div
                      <div id="login-form-container">
                                 <form action="classify" method="GET">
                                            <div class="card" style="width: 400px">
                                             <div class="card-content">
                                                        <div class="media">
                                                        <div class="is-size-4 has-text-centered">Diabetes</div>
                                                        </div>
                                                        <div class="content">
                                                        <div class="field">
                                                                  Pregnancies: <input class="input" type="number" value='0.00' step='0.01'
                                                        </div>
                                                        <div class="field">
                                                                 Glucose: <input class="input" type="number" value='0.00' step='0.01' nam
                                                        </div>
                                                        <div class="field">
                                                                 Blood Pressure: <input class="input" type="number" value='0.00' step='0.
                                                         </div>
                                                         <div class="field">
                                                                  Skin thickness: <input class="input" type="number" value='0.00' step='0.
                                                                   </div>
                                                        <div class="field">
```

```
★ Get Started

               Server.py
C: > Users > ASUS > Desktop > ModelWithFlask > templates > ↔ outcome.html > �� body > �� div#login-form-container > �� div.card >
       <body>
           <div id="login-form-container">
  2
               <div class="card" style="width: 400px">
                  <div class="card-content">
  5
                      <div class="media">
  6
                          <div class="is-size-4 has-text-centered">
                          {{ output }}
  8
                          </div>
  9
                      </div>
  10
                      <form action="home">
                          <div class="field">
  11
  12
                            <button class="button is-fullwidth is-rounded is-success">Retry</but</pre>
  13
  14
                      </form>
  15
                   </div>
  16
               </div>
           </div>
  17
  18
       </body>
```

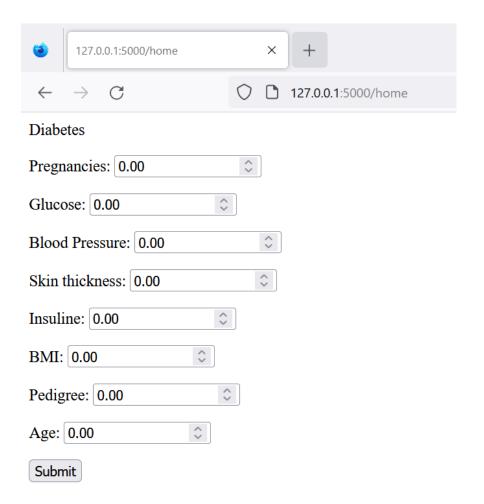
Finally, I've imported Flask to use as server of the model and made two routes, one to home.html and the other to outcome.html

```
4
      # յոլյայ։
 5
 6
 7
     import Model
 8
     from flask import Flask, render_template, request
 9
10
     Run Cell | Run Above | Debug Cell | Go to [2]
      # In[19]:
11
12
13
14
      app = Flask(__name__,template_folder="templates")
15
16
     @app.route('/home')
17
      def home():
          return render_template('home.html')
18
19
20
      Run Cell | Run Above | Debug Cell | Go to [4]
21
      # In[21]:
22
```

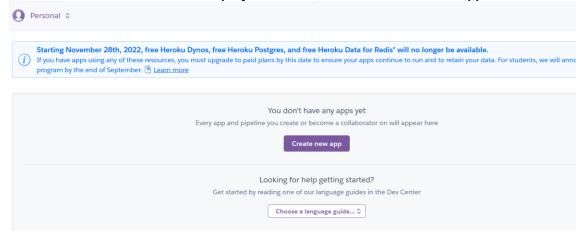
```
3
    @app.route('/outcome',methods=['GET'])
    def classify_type():
           a = request.args.get('Pregnancies')
3
           b = request.args.get('Glucose')
           c = request.args.get('BPressure')
)
          d = request.args.get('SkinThick')
)
          e = request.args.get('Insuline')
L
)
          f = request.args.get('BMI')
           g = request.args.get('Pedigree')
3
Ļ
          h = request.args.get('Age')
          output = Model.outcome(a, b, c, d, e, f, g, h)
3
           return render_template('outcome.html', output=output)
)
       except:
)
       return 'Error'
   if(__name__=='__main__'):
    app.run(debug=True)
```

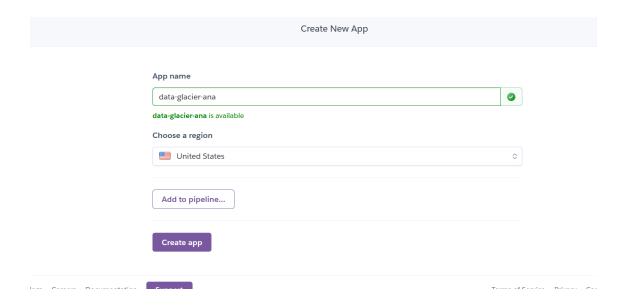
I ran the server with the console and could see the port where it will run.

```
Microsoft Windows [Version 10.0.22621.674]
(c) Microsoft Corporation. All rights reserved.
C:\Users\ASUS>cd Desktop
C:\Users\ASUS\Desktop>cd ModelWithFlask
C:\Users\ASUS\Desktop\ModelWithFlask>python Server.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#
    Column
                              Non-Null Count Dtype
                               768 non-null
                                               int64
0
    Pregnancies
1
    Glucose
                               768 non-null
                                               int64
2
    BloodPressure
                               768 non-null
                                               int64
    SkinThickness
                               768 non-null
                                               int64
4
    Insulin
                               768 non-null
                                               int64
    BMI
                               768 non-null
                                               float64
6
    DiabetesPedigreeFunction 768 non-null
                                               float64
                               768 non-null
                                               int64
8
    Outcome
                               768 non-null
                                               int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
* Serving Flask app 'Server'
* Debug mode: on
JARNING: This is a development server. Do not use it in a production deployment
* Running on http://127.0.0.1:5000
ress CTRL+C to quit
* Restarting with stat
```

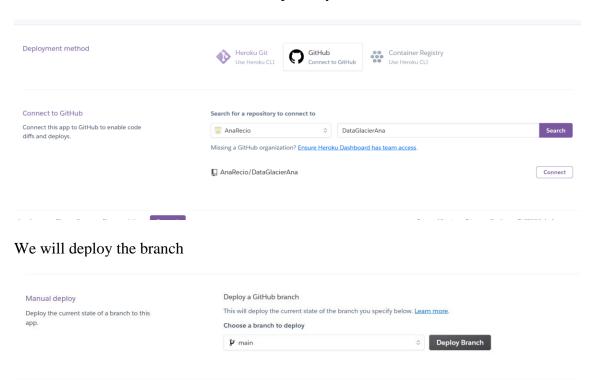


Now in we will use Heroku to deploy the model, we will create a new app





Then, we will connect to GitHub to the repository where the model is



And once is deployed, we will receive confirmation

Deploy the current state of a branch to this app.

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

P main

Deploy Branch

Receive code from GitHub

Build main ad7afc2f

Release phase

Your app was successfully deployed.

②

Deploy to Heroku

✓ View