Deploy using Spydre (app.py)

from flask import Flask,request, url\_for, redirect, render\_template ## importing necessary libraries

import pickle ## pickle for loading model(Diabetes.pkl)

import pandas as pd ## to convert the input data into a dataframe for giving as a input to the model

app = Flask(\_\_name\_\_) ## setting up flask name

model = pickle.load(open("Diabetes.pkl", "rb")) ##loading model

@app.route('/') ## Defining main index route

def home():

return render\_template("index.html") ## showing index.html as homepage

@app.route('/predict',methods=['POST','GET']) ## this route will be called when predict button is called

def predict():

#int\_features=[float(x) for x in request.form.values()]

text1 = request.form['1'] ## Fetching each input field value one by one

text2 = request.form['2']

text3 = request.form['3']

text4 = request.form['4']

text5 = request.form['5']

text6 = request.form['6']

text7 = request.form['7']

text8 = request.form['8']

row\_df = pd.DataFrame([pd.Series([text1,text2,text3,text4,text5,text6,text7,text8])]) ### Creating a dataframe using all the values

print(row\_df)

prediction=model.predict\_proba(row\_df) ## Predicting the output

output='{0:.{1}f}'.format(prediction[0][1], 2) ## Formating output

if output>str(0.5):

return render\_template('index.html',pred='You have chance of having diabetes.\nProbability of having Diabetes is {}'.format(output)) ## Returning the message for use on the same index.html page

else:

return render\_template('index.html',pred='You are safe.\n Probability of having diabetes is {}'.format(output))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True) ## Running the app as debug==True

**Index.HTML**

**(Diabetes Prediction interface if you are diabetes or not )**

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1.0"/>

<title>Diabetes prediction</title>

<!-- CSS -->

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<link href="./static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>

<link href="./static/css2/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>

</head>

<body>

<nav class="gray lighten-1" role="navigation">

<div class="nav-wrapper container"><a id="logo-container" href="#" class="brand-logo">Diabetes Prediction</a>

<ul class="right hide-on-med-and-down">

<li><a href="#">Home</a></li>

</ul>

</nav>

<div class="section no-pad-bot" id="index-banner">

<div class="container">

<h5 style="text-align: center; color: brown; font-weight: 600;">{{pred}}</h5>

<h1 class="header center text">Diabetes Prediction</h1>

<div class="row center">

<h5 class="header col s12 light">Predict the probability of having Diabetes, check your doctor, or go to the hospital.

<br>

</h5>

</div>

<div class="row">

<form action='/predict' method="post" class="col s12">

<div class="row">

<div class="input-field col s4">

<label for="first\_name"><b>Pregnancies</b></label>

<br>

<input placeholder="No. of Pregnancies" name="1" id="first\_name" type="text" class="validate">

</div>

<div class="input-field col s4">

<label for="last\_name"><b>Glucose </b></label>

<br>

<input id="last\_name" name="2" placeholder="Glucose level in sugar" type="text" class="validate">

</div>

<div class="input-field col s4">

<label for="\_name"><b>BloodPressure</b></label>

<br>

<input id="\_name" name="3" placeholder="BloodPressure" type="text" class="validate">

</div>

<div class="input-field col s4">

<label for="first\_name"><b>SkinThickness</b></label>

<br>

<input placeholder="SkinThickness" name="4" id="first\_name" type="text" class="validate">

</div>

<div class="input-field col s4">

<label for="last\_name"><b>Insulin</b></label>

<br>

<input id="last\_name" name="5" placeholder="Insulin level" type="text" class="validate">

</div>

<div class="input-field col s4">

<label ><b>BMI</b></label>

<br>

<input name = '6' placeholder="Body Mass Index">

</div>

<div class="input-field col s4">

<label for="last\_name"><b>DiabetesPedigreeFunction</b></label>

<br>

<input id="last\_name" name="7" placeholder="DiabetesPedigreeFunction" type="text" class="validate">

</div>

<div class="input-field col s4">

<label for="\_name"><b>Age</b></label>

<br>

<input id="\_name" name="8" placeholder="Age" type="text" class="validate">

</div>

</div>

<div class="row center">

<button type="submit" class="btn-large waves-effect waves-light orange">Predict Probability</button>

</div>

</form>

</div>

<br>

<br><br>

</div>

</div>

<footer class="page-footer center orange" style="padding-top: 0px;">

<div class="footer-copyright">

<div class="container" style="padding-bottom: 1rem; font-size: 1.2rem;">

<p>&copy;project created by Reste</p>

<p>rste23@gmail.com</p>

</div>

</div>

</footer>

<!-- Scripts-->

<script src="https://code.jquery.com/jquery-2.1.1.min.js"></script>

<script src=".js/materialize.js"></script>

<script src="js/init.js"></script>

</body>

</html>

**Result for diabetic’s to show**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>**

**<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1.0"/>**

**<title>Diabetes prediction</title>**

**<!-- CSS -->**

**<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">**

**<link href="./static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>**

**<link href="./static/css2/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>**

**</head>**

**<body>**

**<nav class="orange lighten-1" role="navigation">**

**<div class="nav-wrapper container"><a id="logo-container" href="/" class="brand-logo">Diabetes Prediction</a>**

**<ul class="right hide-on-med-and-down">**

**<li><a href="/">Home</a></li>**

**</ul>**

**</nav>**

**<div class="row" style="margin:15% 0% 0% 10%">**

**<h3 >{{pred}}</h3>**

**</div>**

**<br>**

**<br><br>**

**</div>**

**</div>**

**<footer class="page-footer orange">**

**<div class="footer-copyright">**

**<div class="container" style="padding-bottom: 1rem; font-size: 1.2rem;">**

**<p>&copy;project created by Reste</p>**

**<p>rste23@gmail.com</p>**

**</div>**

**</div>**

**</footer>**

**<!-- Scripts-->**

**<script src="https://code.jquery.com/jquery-2.1.1.min.js"></script>**

**<script src=".js/materialize.js"></script>**

**<script src="js/init.js"></script>**

**</body>**

**</html>**

**diabetesPrediction.ipynp**

**(Data Set: Diabetes)**

**Use Case: predict whether a person will have diabetes or not?**

* **Importing important libraries & reading the Dataset**

**# Importing important liabraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

**# KNN**

from sklearn.neighbors import KNeighborsClassifier

from sklearn.neighbors import KNeighborsRegressor

**# Other models for comparision**

from xgboost import XGBClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.ensemble import AdaBoostClassifier

from keras.models import Sequential

from keras.layers import Dense

**#Evaluation**

from sklearn.model\_selection import train\_test\_split,cross\_val\_score,cross\_validate,cross\_val\_predict

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import confusion\_matrix, classification\_report

from sklearn.metrics import precision\_score, recall\_score, f1\_score

from sklearn.metrics import roc\_curve,auc

from sklearn.metrics import accuracy\_score

from mlxtend.plotting import plot\_decision\_regions

**#for warning**

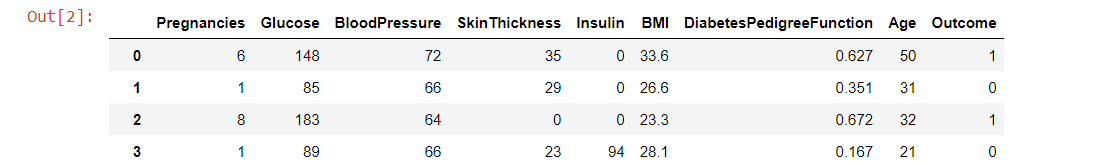
from warnings import filterwarnings

filterwarnings("ignore")

* **Using csv file**

data = pd.read\_csv("diabetes.csv")

data

****

**Explaining the data set**

print("dimension of diabetes data: {}".format(data.shape))

print("n dimentions of diabetes data: {}".format(data.ndim))

print("size of diabetes data: {}".format(data.size))

I have a data set of 768 people who were or were not diagnosed with diabets.

* There are eight input variables and one output variable (the last column). I will be learning a model to map rows of input variables (X) to an output variable (y), which I often summarize as y = f(X).

The variables can be summarized as follows:

\*\*Input Variables (X):\*\*

\*\*Pregnancies\*\*=Number of times pregnant

\*\*Glucose\*\*=Plasma glucose concentration in plasma a 2 hours in an oral glucose tolerance test

\*\*BloodPressure\*\*=Diastolic blood pressure (mm Hg)

\*\*SkinThickness\*\*=Triceps skin fold thickness (mm)

\*\*Insulin\*\*=2-Hour serum insulin (mu U/ml)

\*\*BMI\*\*=Body mass index (weight in kg/(height in $(m)^2$)

\*\*DiabetesPedigreeFunction\*\*=a function which scores likelihood of diabetes based on family history

\*\*Age\*\*=Age (years)

\*\*Output Variables (y):\*\*

\*\*Outcome\*\*=Class variable (0 or 1)

Once the CSV file is loaded into memory, I can split the columns of data into input and output variables.

The data will be stored in a 2D array where the first dimension is rows and the second dimension is columns, e.g. [rows, columns].

**# Understanding the Data and statistical analysis**

data.info()

**Output**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 768 entries, 0 to 767

Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Pregnancies 768 non-null int64

1 Glucose 768 non-null int64

2 BloodPressure 768 non-null int64

3 SkinThickness 768 non-null int64

4 Insulin 768 non-null int64

5 BMI 768 non-null float64

6 DiabetesPedigreeFunction 768 non-null float64

7 Age 768 non-null int64

8 Outcome 768 non-null int64

dtypes: float64(2), int64(7)

memory usage: 54.1 KB

**data.Outcome.value\_counts()**

0 500

1 268

Name: Outcome, dtype: int64

**\*\*“Outcome” is the feature I am going to predict, 0 means No diabetes, 1 means diabetes. Of these 768 data points, 500 are labeled as 0 and 268 as 1.\*\***

**data.isnull().sum()**

Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

DiabetesPedigreeFunction 0

Age 0

Outcome 0

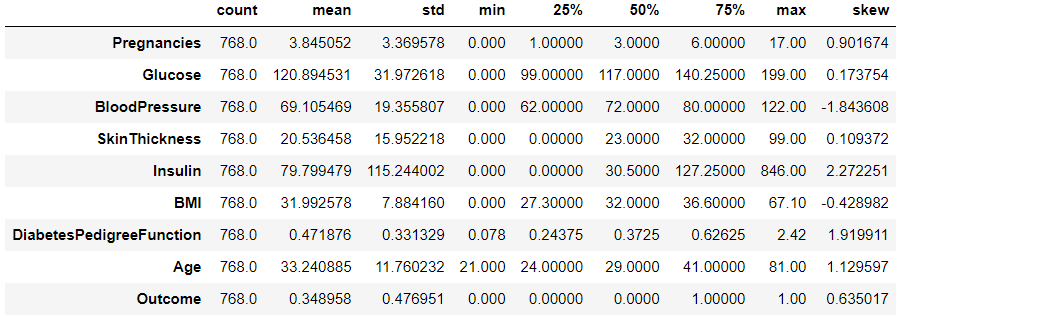
dtype: int64

**\*\*The dataset shows that it has no null values, so I have to use descriptive statistics to check if it's right or not.\*\***

def describe(df):

return pd.concat([df.describe().T, df.skew().rename('skew'),], axis=1)

describe(data)



**Data Visualization**

# comparing glucose with age

diabetics = data[data.Outcome == 1]

healthly = data[data.Outcome == 0]

plt.figure(figsize=(14,7))

# Make an example drawing by looking at gloucose:

# At the end of program, machine learning model will make a prediction by looking at all the other data, not just glucose.

plt.scatter(healthly.Age, healthly.Glucose, color="green", label="healthly", alpha = 0.4)

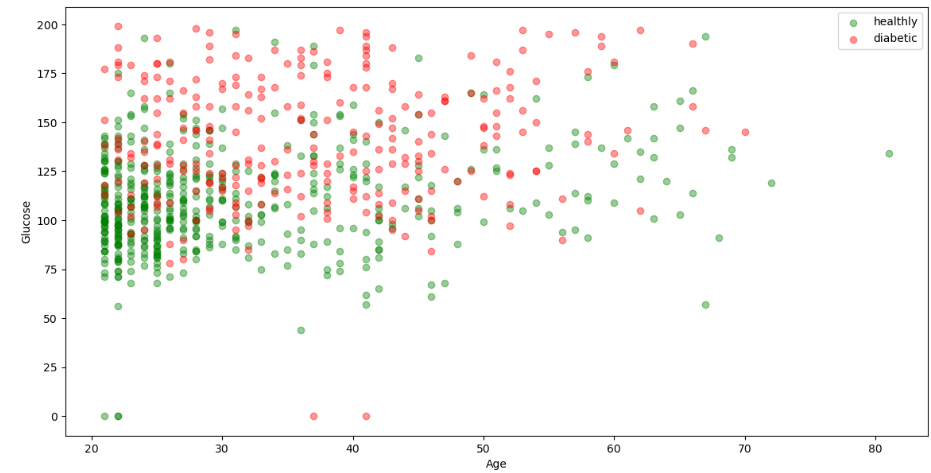
plt.scatter(diabetics.Age, diabetics.Glucose, color="red", label="diabetic", alpha = 0.4)

plt.xlabel("Age")

plt.ylabel("Glucose")

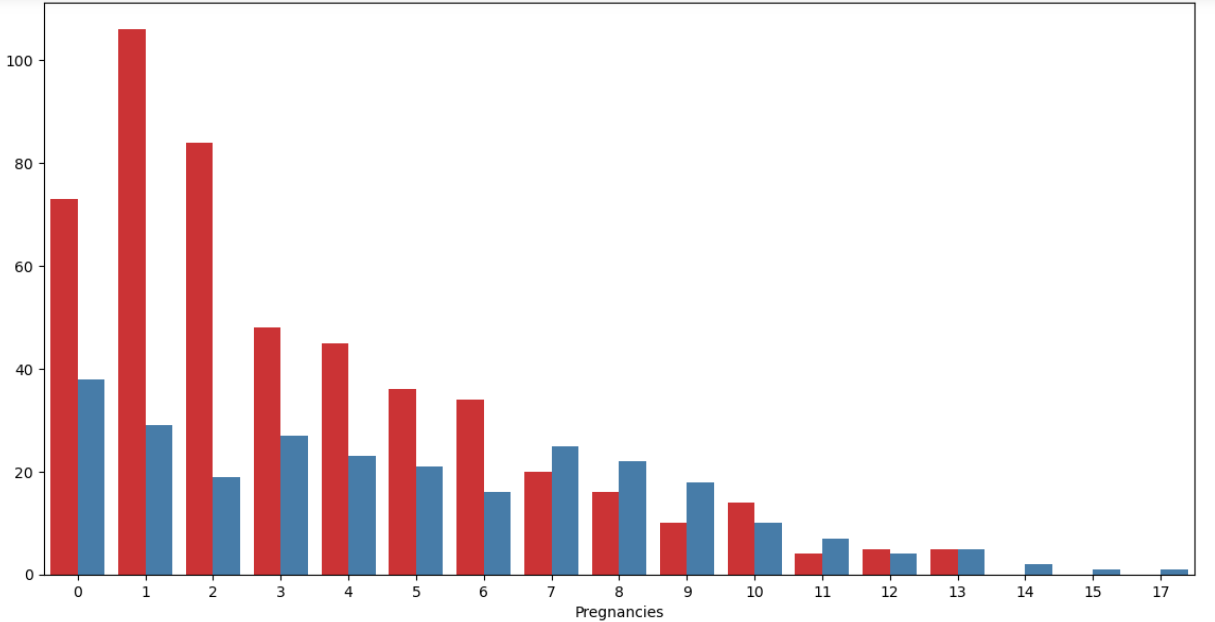
plt.legend()

plt.show()



plt.figure(figsize=(14,7))

sns.countplot(x='Pregnancies', hue='Outcome', data=data, palette='Set1')



**#find out Blood Pressure and age of entries who have diabetes**

plt.figure(figsize=(14,7))

# Scatter with positive example

plt.scatter(data.Age[data.Outcome==1],data.BloodPressure[data.Outcome==1],c="Red");

# Scatter with negative example

plt.scatter(data.Age[data.Outcome==0],data.BloodPressure[data.Outcome==0],c="lightblue");

# Add some helpful info

plt.title("Diabetes in function of Age and Blood pressure")

plt.xlabel("Age")

plt.ylabel("Blood Pressure")

plt.legend(["Diabetes","No Diabetes"]);

