towards
data science

Follow 524K Followers

Diabetes Prediction Application using Streamlit

Using PIMA Indian Diabetes Dataset to create Machine Learning Application

Himanshu Sharma Oct 14, 2020 · 4 min read ★

Streamlit is an open-source Python library which is blazingly fast that makes it easy to build beautiful custom web-apps for machine learning and data science. It is an awesome tool that allows you to create highly interactive dashboards just with some knowledge of python.

Creating applications using streamlit creates an impact on the end-user as it has a good user interface and supports a lot of widgets that are user-friendly. Also creating apps in streamlit is easy. We will create an application using streamlit which will predict whether a user has diabetes or not. The dataset we will be using is PIMA Indian Diabetes Dataset which contains 8 prediction variables and 1 target variable.

Let us look at what are the different attributes in the dataset. The predictor variable is named Outcome which is encoded as 0 and 1 where 0 represents Non-Diabetic and 1 represents Diabetic. Other attributes information is given below.

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Blood Pressure: Diastolic blood pressure (mm Hg)
- Skin Thickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- Diabetes Pedigree Function: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 means non-diabetic or 1 means diabetic)

Exploring the Dataset

Let's start by exploring the dataset we will be using. In order to explore the dataset, we will use a jupyter notebook to load the dataset using pandas and perform exploratory data analysis.

```
import pandas as pd
df = pd.read_csv('Diabetes.csv')
df.head()
```

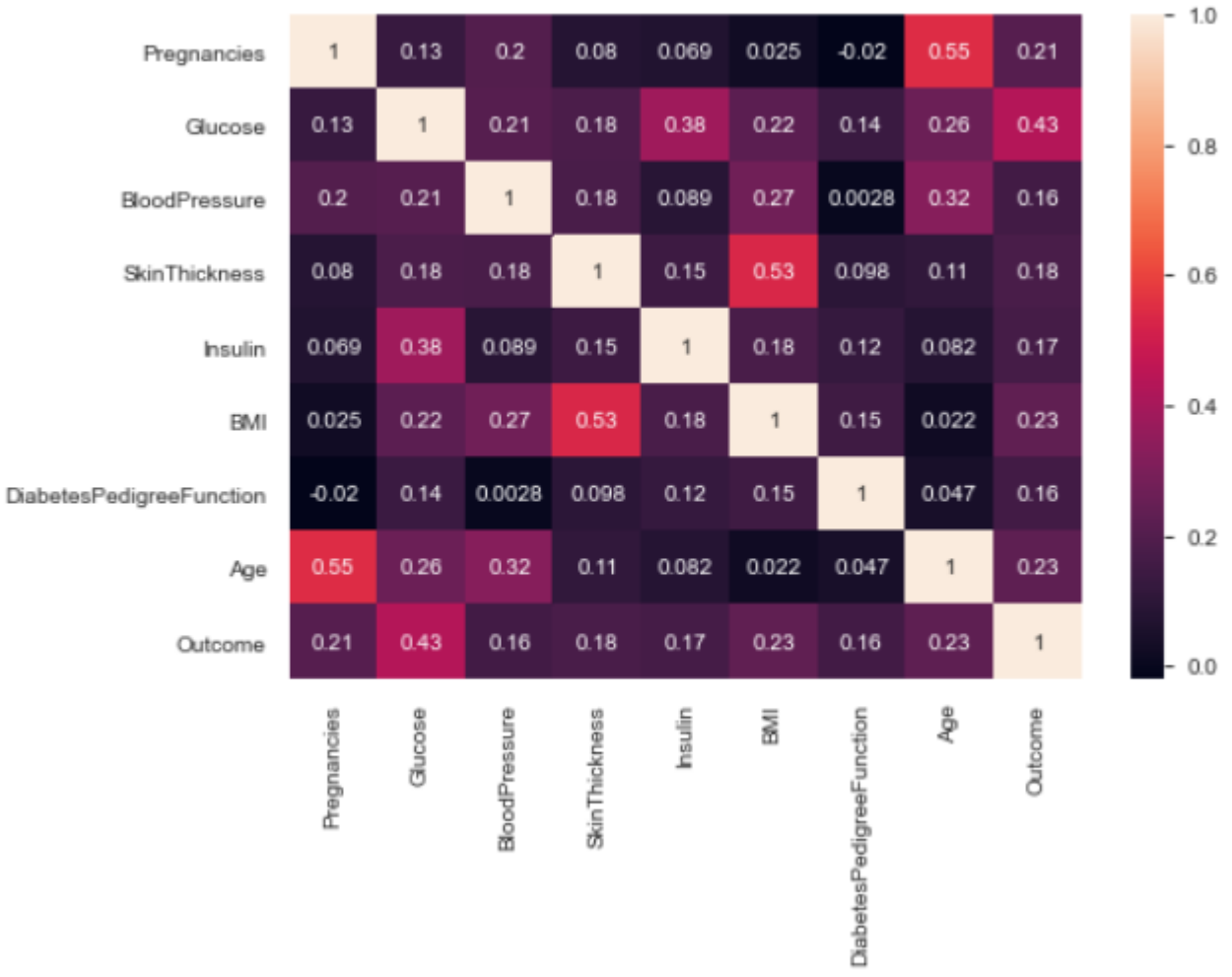
0	6	148	72	35	0	33.6	0.627	50	0
1	1	85	66	29	0	26.6	0.351	31	1
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	1
4	0	137	40	35	168	43.1	2.288	33	1

Diabetes Dataset

Visualizing different attributes:

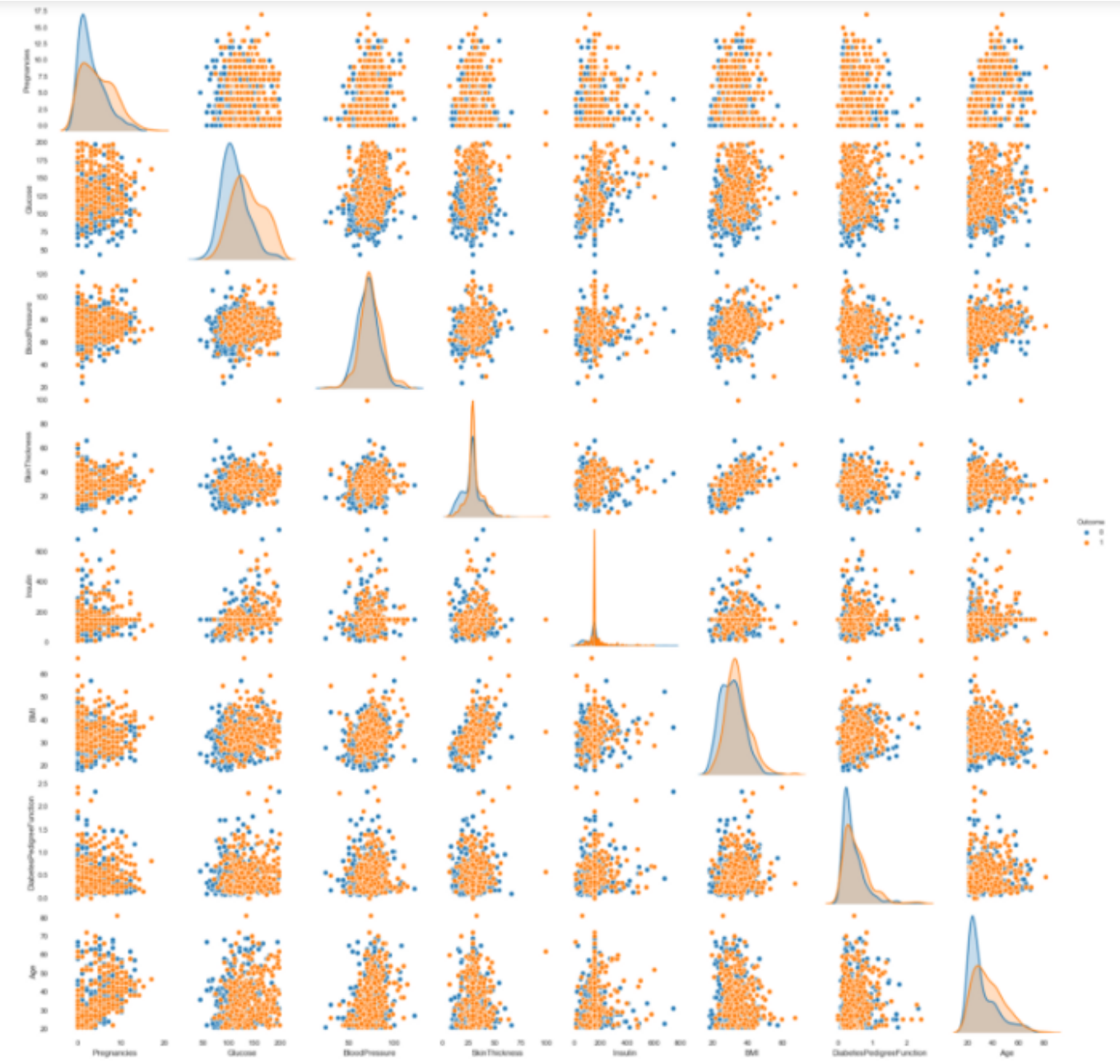
1. Heatmap

```
sns.heatmap(df.corr(),annot=True)
```



Heatmap to Visualize the Correlation between different attributes.

```
sns.pairplot(df,hue='Outcome')
```



A pair plot is used to visualize the similarities as well as the difference between diabetic and non-diabetic patients.

Similarly, we can create many more plots for the EDA process and explore different properties of all the attributes.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pickle

# Split dataset into training set and test set
X_train, X_test, y_train, y_test =
train_test_split(df[['Pregnancies',
'Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPe
digreeFunction','Age']], df['Outcome'], test_size=0.3,
random_state=109)

#Creating the model
logisticRegr = LogisticRegression(C=1)
logisticRegr.fit(X_train, y_train)
y_pred = logisticRegr.predict(X_test)

#Saving the Model
pickle_out = open("logisticRegr.pkl", "wb")
pickle.dump(logisticRegr, pickle_out)
pickle_out.close()
```

Now let's start creating the application. In order to create the application, we need to create a script in python and for that, we need to have a code editor installed in our system. You can use any code editor but I personally use Atom because of its features. We will be creating the prediction model using Logistic Regression. Let's start by importing the required libraries.

Loading the required libraries

```
import streamlit as st
import pandas as pd
import numpy as np
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

```
pickle_in = open('logisticRegr.pkl', 'rb')
classifier = pickle.load(pickle_in)
```

Creating the UI for the application:

Streamlit has predefined UI for its interface. We will use different widgets to display information and take user inputs for prediction.

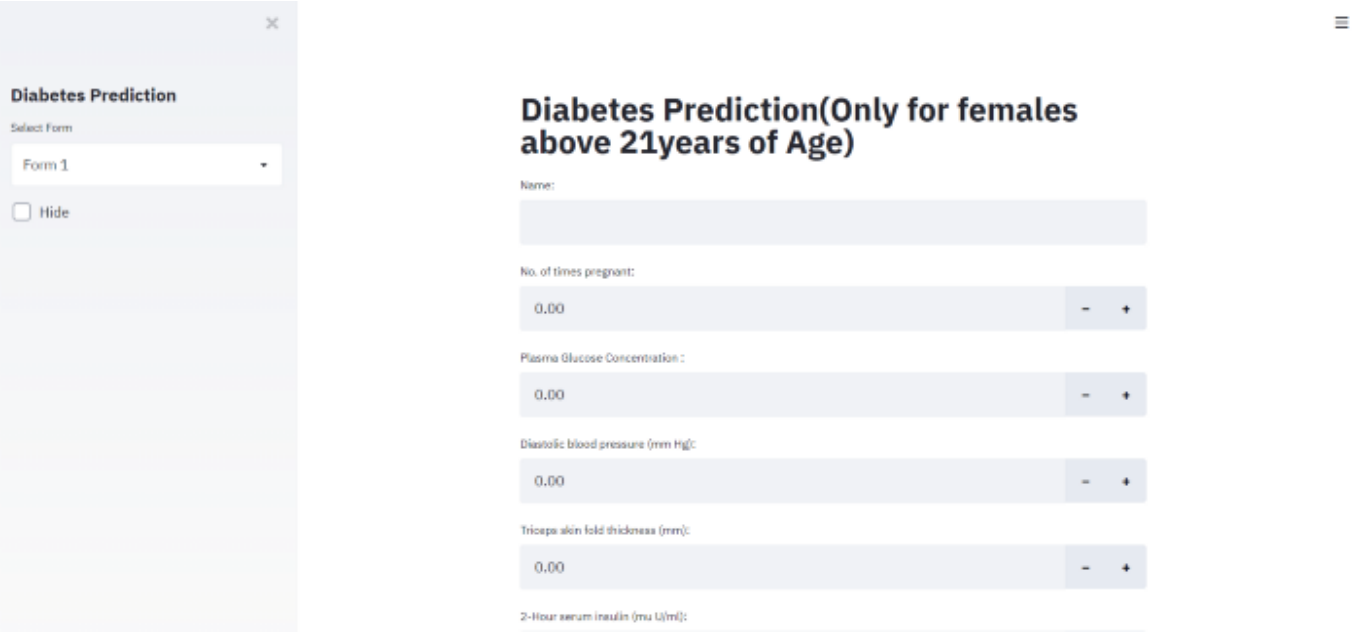
```
st.sidebar.header('Diabetes Prediction')
select = st.sidebar.selectbox('Select Form', ['Form 1'], key='1')
if not st.sidebar.checkbox("Hide", True, key='1'):
    st.title('Diabetes Prediction(Only for females above 21years of
Age)')
    name = st.text_input("Name:")
    pregnancy = st.number_input("No. of times pregnant:")
    glucose = st.number_input("Plasma Glucose Concentration :")
    bp =  st.number_input("Diastolic blood pressure (mm Hg):")
    skin = st.number_input("Triceps skin fold thickness (mm):")
    insulin = st.number_input("2-Hour serum insulin (mu U/ml):")
    bmi = st.number_input("Body mass index (weight in kg/(height in
m)^2):")
    dpf = st.number_input("Diabetes Pedigree Function:")
    age = st.number_input("Age:")

submit = st.button('Predict')

if submit:
        prediction = classifier.predict([[pregnancy, glucose, bp,
skin, insulin, bmi, dpf, age]])
        if prediction == 0:
            st.write('Congratulation',name,'You are not diabetic')
        else:
            st.write(name," we are really sorry to say but it seems
like you are Diabetic.")
```

This will create the application and now we need to save it using the ".py" extension. For example if you need to save your application as "predict" then you need to save it as "predict.py". In order to launch the application, we will open the Anaconda Command Prompt and type the command given below.

This will launch our application and now we can use different user inputs to create whether a person has diabetes or not.



Home Page of the Application

By entering different users' information we can generate different outcomes and see how good and accurate our model is.

This is just an example of what streamlit can do. You can explore more to learn about infinite features that streamlit provide to create web-apps and dashboards. Create your application and share your experiences in the responses of this article.

Before You Go

*Thanks for reading! If you want to get in touch with me, feel free to reach me on hmix13@gmail.com or my **LinkedIn Profile**. You can also view the code and data set I have used here in my **Github**. Also, feel free to explore **my profile** and read different articles I have written related to Data Science.*

Machine Learning | Logistic Regression | Streamlit | Modeling | Data Science