Healthcare DS capstone P2

by Kishan

The National Institute of Diabetes and Digestive and Kidney Diseases research creates knowledge about and treatments for the most chronic, costly, and consequential diseases. The dataset used in this project is originally from NIDDK. The objective is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset.

Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Datset Description The dataset consists of several medical predictor variables and one target variable (Outcome). Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age and more.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
data = pd.read_csv("health_care_diabetes.csv")
```

```
data.shape
```

```
(2768, 9)
```

```
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	1	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
data.tail()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

```
data.columns
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2768 entries, 0 to 2767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               2768 non-null   int64
 1   Glucose                   2768 non-null   int64
 2   BloodPressure             2768 non-null   int64
 3   SkinThickness             2768 non-null   int64
 4   Insulin                   2768 non-null   int64
 5   BMI                       2768 non-null   float64
 6   DiabetesPedigreeFunction  2768 non-null   float64
 7   Age                       2768 non-null   int64
 8   Outcome                   2768 non-null   int64
dtype: float64(2), int64(7)
memory usage: 194.7 KB
```

```
data.isnull()
```

```
Pregnancies  0
Glucose      0
BloodPressure  0
SkinThickness  0
Insulin      0
BMI          0
DiabetesPedigreeFunction  0
Age          0
Outcome      0
dtype: int64
```

There is no missing value in the dataset

```
data.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	2768.000000	2768.000000	2768.000000	2768.000000	2768.000000	2768.000000	2768.000000	2768.000000	2768.000000
mean	3.745318	120.927457	69.145954	20.537572	79.779624	31.992578	0.471876	33.240888	0.342847
std	3.306030	32.000449	19.351561	15.952218	115.749002	7.884160	0.331329	11.690954	0.474944
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

```
import seaborn as sns
```

```
plt.figure(figsize=(6,4))
sns.countplot(data['Outcome'], palette='Set2')
plt.xlabel('Outcome')
plt.ylabel('Count')
```

```
Text(0, 3.7111111111111117, 'Count')
```

```
data['Outcome'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0    1
1    0
2    1
3    0
4    1
     ..
763  0
764  0
765  0
766  1
767  0
Name: Outcome, Length, dtype: int64>
```

```
data['Outcome'].value_counts()
```

```
0    1768
1    1000
Name: Outcome, dtype: int64
```

```
data['Pregnancies'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0    1
1    1
2    8
3    1
4    0
     ..
763  10
764   2
765   5
766   1
767   1
Name: Pregnancies, Length, dtype: int64>
```

```
data['Pregnancies'].value_counts()
```

```
1    291
0    257
2    232
3    196
4    171
     ...
14    14
15    12
16    11
17     8
13     7
Name: Pregnancies, dtype: int64
```

```
data['Glucose'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0    148
1     85
2    183
3     89
4    137
     ...
763  101
764  122
765  121
766  126
767   93
Name: Glucose, Length, dtype: int64>
```

```
data['Glucose'].value_counts()
```

```
99    17
100   17
111   14
129   14
125   14
      ...
191    1
177    1
44     1
62     1
190    1
Name: Glucose, dtype: int64
```

The Insulin and DiaTriches columns has more number of missing values

```
data['DiaTriches'].describe()
```

```
count    2768.000000
mean        0.471876
std         0.331329
min         0.078000
25%         0.243750
50%         0.372500
75%         0.626250
max         2.420000
Name: DiaTriches, dtype: float64
```

```
data.groupby('Outcome').mean()
```

Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	3.298643	109.980769	68.184389	19.664027	68.792421	30.859674	0.429734	31.190000
1	4.865672	141.257463	70.824627	22.164179	100.335821	35.381809	0.550500	37.067164

```
data.groupby('Outcome').count()
```

Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	1768	1768	1768	1768	1768	1768	1768	1768
1	1000	1000	1000	1000	1000	1000	1000	1000

```
data.groupby('Outcome').sum()
```

Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	5833	194471	120611	34761	121630	54571	759.7	55151
1	4865	141257	70825	22164	100336	35382	550.5	37067

```
import seaborn as sns
```

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

```
<seaborn.axisgrid.PairGrid at 0x7f8d0a0a0a90>
```

```
data.hist(figsize=(12,10),bins=30,)
```

```
array([[
```