Name : Abhishek Jani

Department : Information Technology

Experiment No.2
Perform Exploratory data analysis of Healthcare Data.
Date of Performance: 11/8/2023
Date of Submission: 19/8/2023

Aim: Perform Exploratory data analysis of Healthcare Data.

Objective: The objective of this experiment is to perform Exploratory data analytics on healthcare data using python numpy functions

Theory:
EDA is applied to investigate the data and summarize the key insights.

It will give you the basic understanding of your data, it's distribution, null values and much more. You can either explore data using graphs or through some python functions.

There will be two types of analysis:

Univariate and Bivariate.

In the univariate, you will be analysing a single attribute. But in the bivariate, you will be analysing an attribute with the target attribute. In the non-graphical approach, you will be using functions such as shape, summary, describe, isnull, info, datatypes and more.

In the graphical approach, you will be using plots such as scatter, box, bar, density and correlation plots.

Code: -

```
[1] # importing libraries
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
```

```
#Load the csv file data
df = pd.read_csv("diabetes_prediction_dataset.csv")
df
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0
...	...	...	...	...	...	...	...	...	...
99995	Female	80.0	0	0	No Info	27.32	6.2	90	0
99996	Female	2.0	0	0	No Info	17.37	6.5	100	0
99997	Male	66.0	0	0	former	27.83	5.7	155	0
99998	Female	24.0	0	0	never	35.42	4.0	100	0
99999	Female	57.0	0	0	current	22.43	6.6	90	0

100000 rows × 9 columns

0s completed at 5:17 PM

```
[3] #print all variables
    df.columns
```

```
Index(['gender', 'age', 'hypertension', 'heart_disease', 'smoking_history',
       'bmi', 'HbA1c_level', 'blood_glucose_level', 'diabetes'],
      dtype='object')
```

```
# The data set contaion 1 lakh rows and we are working on the first 1000 data only.
sample = df.head(1000)
sample
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0
...	...	...	...	...	...	...	...	...	...
995	Male	62.0	0	0	never	29.26	5.0	200	0
996	Female	44.0	0	0	No Info	46.07	5.0	145	0
997	Male	21.0	0	0	never	31.44	6.2	85	0
998	Male	45.0	0	1	current	38.25	6.1	140	0
999	Female	43.0	0	0	never	27.32	6.6	130	1

0s completed at 5:17 PM

+ Code + Text

```
sample.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   gender               1000 non-null   object
 1   age                  1000 non-null   float64
 2   hypertension         1000 non-null   int64
 3   heart_disease        1000 non-null   int64
 4   smoking_history      1000 non-null   object
 5   bmi                  1000 non-null   float64
 6   HbA1c_level          1000 non-null   float64
 7   blood_glucose_level  1000 non-null   int64
 8   diabetes             1000 non-null   int64
dtypes: float64(3), int64(4), object(2)
memory usage: 70.4+ KB
```

[8]
```
#  0 says No diabetes
#  1 says YES to diabetes
sample['diabetes'].value_counts()
```

```
0    918
1     82
Name: diabetes, dtype: int64
```

[9]
```
sample['smoking_history'].value_counts()
```

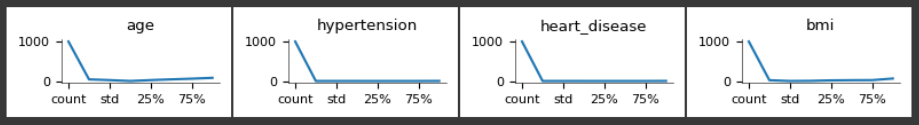
```
never          372
No Info        351
former          89
current         82
not current     62
ever            44
```

0s completed at 5:17PM

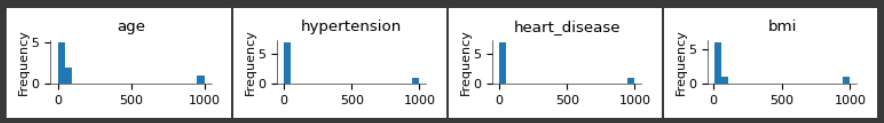
[10]
```
#Describe the data
sample.describe()
```

	age	hypertension	heart_disease	bmi	HbA1c_level	blood_glucose_level	diabetes
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	41.273680	0.068000	0.032000	26.959350	5.559500	138.419000	0.082000
std	22.909619	0.251872	0.176088	6.786204	1.098479	38.703636	0.274502
min	0.080000	0.000000	0.000000	10.300000	3.500000	80.000000	0.000000
25%	22.750000	0.000000	0.000000	22.947500	4.800000	100.000000	0.000000
50%	42.000000	0.000000	0.000000	27.320000	5.800000	145.000000	0.000000
75%	59.000000	0.000000	0.000000	29.065000	6.200000	159.000000	0.000000
max	80.000000	1.000000	1.000000	69.370000	9.000000	300.000000	1.000000

Values



Distributions



2-d distributions

0s completed at 5:17PM

+ Code + Text

```
[11] #Find the duplicates
     sample.duplicated().sum()
```

```
0
```

```
     #Find null values
     sample.isnull().sum()
```

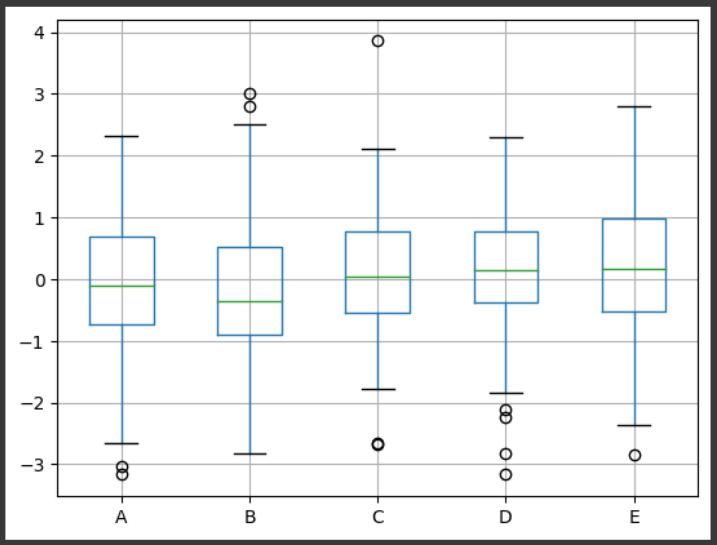
```
gender                0
age                   0
hypertension          0
heart_disease         0
smoking_history       0
bmi                   0
HbA1c_level           0
blood_glucose_level   0
diabetes              0
dtype: int64
```

```
[13] #Datatypes
     sample.dtypes
```

```
gender                 object
age                   float64
hypertension            int64
heart_disease           int64
smoking_history        object
bmi                   float64
HbA1c_level           float64
blood_glucose_level     int64
diabetes                int64
dtype: object
```

0s completed at 5:17 PM

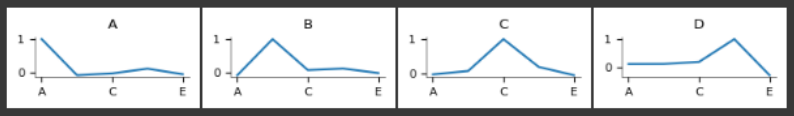
```
[22] sample = pd.DataFrame(np.random.randn(100, 5), columns=list('ABCDE'))
     sample.boxplot(return_type='axes');
```



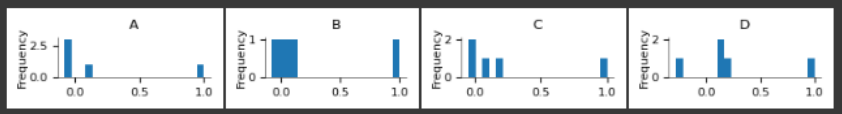
```
[36] #Correlation: find the correlation among the variables, we can make use of the correlation function.
     # This will give you a fair idea of the correlation strength between different variables.
     sample.corr()
```

	A	B	C	D	E
A	1.000000	-0.076712	-0.023471	0.118574	-0.047758
B	-0.076712	1.000000	0.076370	0.122405	-0.012277
C	-0.023471	0.076370	1.000000	0.193908	-0.046426
D	0.118574	0.122405	0.193908	1.000000	-0.277433
E	-0.047758	-0.012277	-0.046426	-0.277433	1.000000

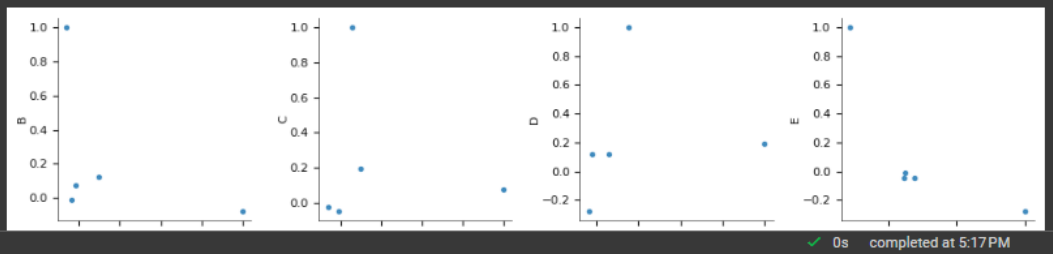
Values



Distributions



2-d distributions



0s completed at 5:17 PM

Google Collaboratory Link: -

AIML EXP 2

Conclusion: - EDA is the most important part of any analysis. A lot of information will be obtained about the dataset in consideration. Most data related answers are obtained using EDA.