CLASSIFYMEISTER

- CANCER PREDICTOR

Project Description

About Data-Set Used

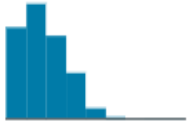
The data is comprised of 36 columns and has a size of 852.

It has columns related to:

- Age
- Smoking habits
- Details related the sexual life
- Hormonal Contraceptives
- Time since their first and last diagnosis
- Target data – related to their cancer status

Dataset

Cervical Cancer Risk Factors for Biopsy

Age	Number of sexual...	First sexual interc...	Num of pregnanci...	Smokes	Smokes (years)	Smokes (packs/y...	Hormonal Contra...	Hormonal Contra...	IUD
 13 84	2.0 32%	15.0 19%	1.0 31%	0.0 84%	0.0 84%	0.0 84%	1.0 56%	0.0 31%	0.0 77%
	3.0 24%	17.0 18%	2.0 28%	1.0 14%	1.266972909 2%	0.5132021277 2%	0.0 31%	? 13%	? 14%
	Other (378) 44%	Other (544) 63%	Other (348) 41%	Other (13) 2%	Other (121) 14%	Other (118) 14%	Other (108) 13%	Other (481) 56%	Other (83) 10%
18	4.0	15.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
15	1.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
34	1.0	?	1.0	0.0	0.0	0.0	0.0	0.0	0.0
52	5.0	16.0	4.0	1.0	37.0	37.0	1.0	3.0	0.0
46	3.0	21.0	4.0	0.0	0.0	0.0	1.0	15.0	0.0
42	3.0	23.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
51	3.0	17.0	6.0	1.0	34.0	3.4	0.0	0.0	1.0
26	1.0	26.0	3.0	0.0	0.0	0.0	1.0	2.0	1.0
45	1.0	20.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0
44	3.0	15.0	?	1.0	1.266972909	2.8	0.0	0.0	?
44	3.0	26.0	4.0	0.0	0.0	0.0	1.0	2.0	0.0
27	1.0	17.0	3.0	0.0	0.0	0.0	1.0	8.0	0.0
45	4.0	14.0	6.0	0.0	0.0	0.0	1.0	10.0	1.0
44	2.0	25.0	2.0	0.0	0.0	0.0	1.0	5.0	0.0

Modifications in the data set

Checking the number of NULL values present in each column:

```
null_counts = df.isnull().sum()
null_counts
```

Age	0
Number of sexual partners	26
First sexual intercourse	7
Num of pregnancies	56
Smokes	13
Smokes (years)	13
Smokes (packs/year)	13
Hormonal Contraceptives	108
Hormonal Contraceptives (years)	108
IUD	117
IUD (years)	117
STDs	105
STDs (number)	105
STDs:condylomatosis	105
STDs:cervical condylomatosis	105
STDs:vaginal condylomatosis	105
STDs:vulvo-perineal condylomatosis	105
STDs:syphilis	105
STDs:pelvic inflammatory disease	105
STDs:genital herpes	105
STDs:molluscum contagiosum	105
STDs:AIDS	105
STDs:HIV	105
STDs:Hepatitis B	105
STDs:HPV	105
STDs: Number of diagnosis	0
STDs: Time since first diagnosis	787
STDs: Time since last diagnosis	787
Dx:Cancer	0
Dx:CIN	0
Dx:HPV	0
Dx	0
Hinselmann	0
Schiller	0
Citology	0
Biopsy	0

For columns having a very large no. of null values:
We have removed the columns having a large number of null inputs greater than 750 (87%).

For columns having a low no. of null values:
If the number of null values is less than 100 we have replaced it with the mode value.

For the rest of the columns having NULL values:
We used the KNN model to fill the value with help of the three nearest neighbours.

```
// To delete columns having large no. of null values around 750
df = df.drop(['STDs: Time since first diagnosis', 'STDs: Time since last diagnosis'], axis=1)
df.head()

// To convert the null values to mode of  3 nearest neighbour if no. of null inputs is above 100
from sklearn.impute import KNNImputer

# create an imputer object with the chosen imputation strategy
imputer = KNNImputer(n_neighbors=3)

# impute missing values in the DataFrame
df = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)

/* To convert the null inputs the mode if no. of null inuts is less than 100
columns_to_replace = ['Number of sexual partners' , 'First sexual intercourse', 'Num of pregnancies',   'Smokes',   'Smokes (years)',   'Smokes (packs/year)']
# replace null values in a particular column with mode
for cols in columns_to_replace:
  df[cols].fillna(df[cols].mode()[0], inplace=True)
```

Compression of data between 0 & 1

Here we have compressed the data by using the following code between 0 &1

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
for col in df.columns:
  df[col] = scaler.fit_transform(df[[col]])

null_counts = df.isnull().sum()
null_counts

```

Assigning values to some column

Here we have filled missing cells of columns –number of sexual partner, first sexual intercourse, num of pregnancies, smokes, smokes(years) & smoke(pack/year) their mode value.

```
columns_to_replace = ['Number of sexual partners' , 'First sexual intercourse', 'Num of pregnancies',  'Smokes',  'Smokes (years)',  'Smokes (packs/year)']
# replace null values in a particular column with mode
for cols in columns_to_replace:
  df[cols].fillna(df[cols].mode()[0], inplace=True)
```

Removing the outliners

With the following code we have removed the exceptional values(outliers) from the data set of all the columns

```
outlier_columns = ['Age','Number of sexual partners','First sexual intercourse','Num of pregnancies','Smokes','Smokes (years)','Smokes (packs/year)','Hormonal Contraceptives','Hormonal Contraceptives (years)'
from scipy import stats
z_scores = df.apply(lambda x: (x - x.mean()) / x.std())
outlier_threshold = 3
df = df[(z_scores <= outlier_threshold) | (z_scores >= -outlier_threshold)]

num_outliers = ((z_scores > outlier_threshold) | (z_scores < -outlier_threshold)).sum()
print('Number of outliers in each column:')
for col_name, num in num_outliers.iteritems():
    print(f"{col_name}: {num}")
```