

Submitted by: Angelo Luis C. Cu

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

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from sklearn.preprocessing import binarize
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression

%matplotlib inline

!pip install ucimlrepo

Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
cervical_cancer_risk_factors = fetch_ucirepo(id=383)

# data (as pandas dataframes)
X = cervical_cancer_risk_factors.data.features
y = cervical_cancer_risk_factors.data.targets

# metadata
print(cervical_cancer_risk_factors.metadata)

# variable information
print(cervical_cancer_risk_factors.variables)
```

17	STDs:syphilis	Feature	Continuous	None
18	STDs:pelvic inflammatory disease	Feature	Continuous	None
19	STDs:genital herpes	Feature	Continuous	None
20	STDs:molluscum contagiosum	Feature	Continuous	None
21	STDs:AIDS	Feature	Continuous	None
22	STDs:HIV	Feature	Continuous	None
23	STDs:Hepatitis B	Feature	Continuous	None
24	STDs:HPV	Feature	Continuous	None
25	STDs: Number of diagnosis	Feature	Integer	None
26	STDs: Time since first diagnosis	Feature	Continuous	None
27	STDs: Time since last diagnosis	Feature	Continuous	None
28	Dx:Cancer	Feature	Integer	None
29	Dx:CIN	Feature	Integer	None
30	Dx:HPV	Feature	Integer	None
31	Dx	Feature	Integer	None
32	Hinselmann	Feature	Integer	None
33	Schiller	Feature	Integer	None
34	Citology	Feature	Integer	None
35	Biopsy	Feature	Integer	None

	description	units	missing_values
0		None	None
1		None	None
2		None	None
3		None	None
4		None	None
5		None	None
6		None	None
7		None	None
8		None	None
9		None	None
10		None	None
11		None	None
12		None	None
13		None	None
14		None	None
15		None	None
16		None	None
17		None	None
18		None	None
19		None	None
20		None	None
21		None	None
22		None	None
23		None	None
24		None	None
25		None	None
26		None	None
27		None	None
28		None	None
29		None	None
30		None	None
31		None	None
32		None	None
33		None	None
34		None	None
35		None	None

```
cervical_cancer = pd.concat([X, y], axis = 1)
cervical_cancer
```

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormo Contracepti
0	18	4.0	15.0	1.0	0.0	0.0	0.0	
1	15	1.0	14.0	1.0	0.0	0.0	0.0	
2	34	1.0	NaN	1.0	0.0	0.0	0.0	
3	52	5.0	16.0	4.0	1.0	37.0	37.0	
4	46	3.0	21.0	4.0	0.0	0.0	0.0	
...	
853	34	3.0	18.0	0.0	0.0	0.0	0.0	
854	32	2.0	19.0	1.0	0.0	0.0	0.0	
855	25	2.0	17.0	0.0	0.0	0.0	0.0	
856	33	2.0	24.0	2.0	0.0	0.0	0.0	
857	29	2.0	20.0	1.0	0.0	0.0	0.0	

Data Wrangling

```
# checks for duplicate values
cervical_cancer[cervical_cancer.duplicated()].shape[0]

23

# removes duplicate rows
cervical_cancer.drop_duplicates(inplace=True)

# checking for missing values
cervical_cancer.info()
# no. of sexual partners- STDs:HPV,
# STD time since first and last diagnosis have missing values

<class 'pandas.core.frame.DataFrame'>
Index: 835 entries, 0 to 857
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         835 non-null    int64
1   Number of sexual partners                 810 non-null    float64
2   First sexual intercourse                  828 non-null    float64
3   Num of pregnancies                        779 non-null    float64
4   Smokes                                    822 non-null    float64
5   Smokes (years)                           822 non-null    float64
6   Smokes (packs/year)                      822 non-null    float64
7   Hormonal Contraceptives                  732 non-null    float64
8   Hormonal Contraceptives (years)          732 non-null    float64
9   IUD                                       723 non-null    float64
10  IUD (years)                              723 non-null    float64
11  STDs                                     735 non-null    float64
12  STDs (number)                           735 non-null    float64
13  STDs:condylomatosis                     735 non-null    float64
14  STDs:cervical condylomatosis             735 non-null    float64
15  STDs:vaginal condylomatosis             735 non-null    float64
16  STDs:vulvo-perineal condylomatosis      735 non-null    float64
17  STDs:syphilis                           735 non-null    float64
18  STDs:pelvic inflammatory disease        735 non-null    float64
19  STDs:genital herpes                     735 non-null    float64
20  STDs:molluscum contagiosum              735 non-null    float64
21  STDs:AIDS                               735 non-null    float64
22  STDs:HIV                                735 non-null    float64
23  STDs:Hepatitis B                       735 non-null    float64
24  STDs:HPV                                735 non-null    float64
25  STDs: Number of diagnosis               835 non-null    int64
26  STDs: Time since first diagnosis         71 non-null     float64
27  STDs: Time since last diagnosis         71 non-null     float64
28  Dx:Cancer                               835 non-null    int64
29  Dx:CIN                                  835 non-null    int64
30  Dx:HPV                                  835 non-null    int64
31  Dx                                       835 non-null    int64
32  Hinselmann                              835 non-null    int64
33  Schiller                                835 non-null    int64
34  Citology                                835 non-null    int64
35  Biopsy                                  835 non-null    int64
dtypes: float64(26), int64(10)
memory usage: 241.4 KB

# changes spaces to underscores for easier column access
cervical_cancer.columns = [column.replace(' ', '_') for column in cervical_cancer.columns]
cervical_cancer.columns = [column.strip('_') for column in cervical_cancer.columns]
cervical_cancer
```

	Age	Number_of_sexual_partners	First_sexual_intercourse	Num_of_pregnancies	Smokes
0	18	4.0	15.0	1.0	0
1	15	1.0	14.0	1.0	0
2	34	1.0	NaN	1.0	0
3	52	5.0	16.0	4.0	1
4	46	3.0	21.0	4.0	0
...	
853	34	3.0	18.0	0.0	0
854	32	2.0	19.0	1.0	0
855	25	2.0	17.0	0.0	0
856	33	2.0	24.0	2.0	0
857	29	2.0	20.0	1.0	0

835 rows × 36 columns

```
# I decided to get the mean by age to fill NaN values in column 1 to 3
cervical_cancer['Number_of_sexual_partners'] = cervical_cancer.groupby('Age')['Number_of_sexual_partners'].transform(lambda x: x.fillna(x.mean()))
cervical_cancer['First_sexual_intercourse'] = cervical_cancer.groupby('Age')['First_sexual_intercourse'].transform(lambda x: x.fillna(x.mean()))
cervical_cancer['Num_of_pregnancies'] = cervical_cancer.groupby('Age')['Num_of_pregnancies'].transform(lambda x: x.fillna(x.mean()))
cervical_cancer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 835 entries, 0 to 857
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         835 non-null    int64
1   Number_of_sexual_partners                 835 non-null    float64
2   First_sexual_intercourse                 835 non-null    float64
3   Num_of_pregnancies                      835 non-null    float64
4   Smokes                                    822 non-null    float64
5   Smokes_(years)                          822 non-null    float64
6   Smokes_(packs/year)                     822 non-null    float64
7   Hormonal_Contraceptives                 732 non-null    float64
8   Hormonal_Contraceptives_(years)         732 non-null    float64
9   IUD                                       723 non-null    float64
10  IUD_(years)                             723 non-null    float64
11  STDs                                     735 non-null    float64
12  STDs_(number)                           735 non-null    float64
13  STDs:condylomatosis                     735 non-null    float64
14  STDs:cervical_condylomatosis             735 non-null    float64
15  STDs:vaginal_condylomatosis              735 non-null    float64
16  STDs:vulvo-perineal_condylomatosis       735 non-null    float64
17  STDs:syphilis                           735 non-null    float64
18  STDs:pelvic_inflammatory_disease         735 non-null    float64
19  STDs:genital_herpes                     735 non-null    float64
20  STDs:molluscum_contagiosum               735 non-null    float64
21  STDs:AIDS                               735 non-null    float64
22  STDs:HIV                                 735 non-null    float64
23  STDs:Hepatitis_B                       735 non-null    float64
24  STDs:HPV                                 735 non-null    float64
25  STDs:_Number_of_diagnosis                835 non-null    int64
26  STDs:_Time_since_first_diagnosis         71 non-null     float64
27  STDs:_Time_since_last_diagnosis          71 non-null     float64
28  Dx:Cancer                               835 non-null    int64
29  Dx:CIN                                   835 non-null    int64
30  Dx:HPV                                   835 non-null    int64
31  Dx                                       835 non-null    int64
32  Hinselmann                             835 non-null    int64
33  Schiller                                835 non-null    int64
34  Citology                                835 non-null    int64
35  Biopsy                                  835 non-null    int64
dtypes: float64(26), int64(10)
memory usage: 241.4 KB
```

```
# for smokes (columns 4-6)
cervical_cancer.Smokes.value_counts()
```

```
Smokes
0.0    699
1.0    123
Name: count, dtype: int64
```

```
# since most of the data is at 0, I decided to fill the NaN values with 0 (false / do not smoke)
cervical_cancer.iloc[:, 4:7] = cervical_cancer.iloc[:, 4:7].fillna(0)
cervical_cancer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 835 entries, 0 to 857
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         835 non-null    int64
1   Number_of_sexual_partners                 835 non-null    float64
2   First_sexual_intercourse                 835 non-null    float64
3   Num_of_pregnancies                      835 non-null    float64
4   Smokes                                    835 non-null    float64
5   Smokes_(years)                          835 non-null    float64
6   Smokes_(packs/year)                     835 non-null    float64
7   Hormonal_Contraceptives                 732 non-null    float64
8   Hormonal_Contraceptives_(years)         732 non-null    float64
9   IUD                                       723 non-null    float64
10  IUD_(years)                             723 non-null    float64
11  STDs                                     735 non-null    float64
12  STDs_(number)                           735 non-null    float64
```

```
13  STDs:condylomatosis          735 non-null    float64
14  STDs:cervical_condylomatosis 735 non-null    float64
15  STDs:vaginal_condylomatosis  735 non-null    float64
16  STDs:vulvo-perineal_condylomatosis 735 non-null    float64
17  STDs:syphilis                735 non-null    float64
18  STDs:pelvic_inflammatory_disease 735 non-null    float64
19  STDs:genital_herpes           735 non-null    float64
20  STDs:molluscum_contagiosum    735 non-null    float64
21  STDs:AIDS                     735 non-null    float64
22  STDs:HIV                      735 non-null    float64
23  STDs:Hepatitis_B              735 non-null    float64
24  STDs:HPV                     735 non-null    float64
25  STDs:_Number_of_diagnosis      835 non-null    int64
26  STDs:_Time_since_first_diagnosis 71 non-null     float64
27  STDs:_Time_since_last_diagnosis 71 non-null     float64
28  Dx:Cancer                     835 non-null    int64
29  Dx:CIN                        835 non-null    int64
30  Dx:HPV                       835 non-null    int64
31  Dx                            835 non-null    int64
32  Hinselmann                    835 non-null    int64
33  Schiller                      835 non-null    int64
34  Citology                      835 non-null    int64
35  Biopsy                       835 non-null    int64
dtypes: float64(26), int64(10)
memory usage: 241.4 KB
```

```
# for hormonal contraceptives (columns 7-8)
cervical_cancer['Hormonal_Contraceptives'].value_counts()
```

```
Hormonal_Contraceptives
1.0      477
0.0      255
Name: count, dtype: int64
```

```
# since there is still a significant amount of data in 1,
# I decided to fill the NaN values for column 7 with 1 (true)
cervical_cancer['Hormonal_Contraceptives'].fillna(1, inplace=True)
cervical_cancer['Hormonal_Contraceptives'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 835 entries, 0 to 857
Series name: Hormonal_Contraceptives
Non-Null Count  Dtype
-----
835 non-null    float64
dtypes: float64(1)
memory usage: 13.0 KB
```

```
# for column 8, the mean per age would be filled since I know that the hormonal contraceptives row are linked
cervical_cancer['Hormonal_Contraceptives_(years)'] = cervical_cancer.groupby('Age')['Hormonal_Contraceptives_(years)'].transform(lambda x: x)
cervical_cancer['Hormonal_Contraceptives_(years)'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 835 entries, 0 to 857
Series name: Hormonal_Contraceptives_(years)
Non-Null Count  Dtype
-----
835 non-null    float64
dtypes: float64(1)
memory usage: 13.0 KB
```

```
# for IUDs (columns 9-10)
cervical_cancer.IUD.value_counts()
```

```
IUD
0.0      640
1.0       83
Name: count, dtype: int64
```

```
# since most of the data is 0, I decided to fill the NaN values with 0 (false)
cervical_cancer.IUD.fillna(0, inplace=True)
cervical_cancer.IUD.info()
```

```
<class 'pandas.core.series.Series'>
Index: 835 entries, 0 to 857
Series name: IUD
Non-Null Count  Dtype
-----
835 non-null    float64
dtypes: float64(1)
memory usage: 13.0 KB
```

```
# for column 10, I decided to fill them with 0 already since I know that the STD rows are linked to column 11
cervical_cancer.iloc[:, 10] = cervical_cancer.iloc[:, 10].fillna(0)
cervical_cancer['IUD_(years)'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 835 entries, 0 to 857
Series name: IUD_(years)
Non-Null Count  Dtype
-----
835 non-null    float64
dtypes: float64(1)
memory usage: 13.0 KB
```

```
# for STDs (columns 11-24)
cervical_cancer.STDs.value_counts()
```

```
STDs
0.0      656
```

```
1.0      79
Name: count, dtype: int64

# since most of the data is 0, I decided to fill the NaN values with 0 (false)
cervical_cancer.STDs.fillna(0, inplace=True)
cervical_cancer.STDs.info()

<class 'pandas.core.series.Series'>
Index: 835 entries, 0 to 857
Series name: STDs
Non-Null Count  Dtype
-----
835 non-null    float64
dtypes: float64(1)
memory usage: 13.0 KB

# for columns 12-24, I decided to fill them with 0 already since I know that the STD rows are linked to column 11
cervical_cancer.iloc[:, 12:25] = cervical_cancer.iloc[:, 12:25].fillna(0)
cervical_cancer.info()

<class 'pandas.core.frame.DataFrame'>
Index: 835 entries, 0 to 857
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         835 non-null    int64
1   Number_of_sexual_partners                 835 non-null    float64
2   First_sexual_intercourse                 835 non-null    float64
3   Num_of_pregnancies                       835 non-null    float64
4   Smokes                                     835 non-null    float64
5   Smokes_(years)                          835 non-null    float64
6   Smokes_(packs/year)                     835 non-null    float64
7   Hormonal_Contraceptives                 835 non-null    float64
8   Hormonal_Contraceptives_(years)         835 non-null    float64
9   IUD                                       835 non-null    float64
10  IUD_(years)                             835 non-null    float64
11  STDs                                     835 non-null    float64
12  STDs_(number)                          835 non-null    float64
13  STDs:condylomatosis                    835 non-null    float64
14  STDs:cervical_condylomatosis           835 non-null    float64
15  STDs:vaginal_condylomatosis            835 non-null    float64
16  STDs:vulvo-perineal_condylomatosis     835 non-null    float64
17  STDs:syphilis                         835 non-null    float64
18  STDs:pelvic_inflammatory_disease       835 non-null    float64
19  STDs:genital_herpes                   835 non-null    float64
20  STDs:molluscum_contagiosum             835 non-null    float64
21  STDs:AIDS                             835 non-null    float64
22  STDs:HIV                             835 non-null    float64
23  STDs:Hepatitis_B                     835 non-null    float64
24  STDs:HPV                             835 non-null    float64
25  STDs:_Number_of_diagnosis              835 non-null    int64
26  STDs:_Time_since_first_diagnosis        71 non-null    float64
27  STDs:_Time_since_last_diagnosis         71 non-null    float64
28  Dx:Cancer                             835 non-null    int64
29  Dx:CIN                                835 non-null    int64
30  Dx:HPV                                835 non-null    int64
31  Dx                                     835 non-null    int64
32  Hinselmann                            835 non-null    int64
33  Schiller                              835 non-null    int64
34  Citology                              835 non-null    int64
35  Biopsy                                835 non-null    int64
dtypes: float64(26), int64(10)
memory usage: 241.4 KB

# since the data is already filled, I decided to move on to columns 26-27
# for STD time since first and last diagnosis, it would still base on the STD column (column 11)
# if the data from column 11 is 0, the data would be filled with 0
cervical_cancer['STDs:_Time_since_first_diagnosis'] = cervical_cancer.apply(lambda row: 0 if pd.isnull(row['STDs:_Time_since_first_diagnosis']) else row['STDs:_Time_since_first_diagnosis'], axis=1)
cervical_cancer['STDs:_Time_since_last_diagnosis'] = cervical_cancer.apply(lambda row: 0 if pd.isnull(row['STDs:_Time_since_last_diagnosis']) else row['STDs:_Time_since_last_diagnosis'], axis=1)
cervical_cancer.info()

<class 'pandas.core.frame.DataFrame'>
Index: 835 entries, 0 to 857
Data columns (total 36 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         835 non-null    int64
1   Number_of_sexual_partners                 835 non-null    float64
2   First_sexual_intercourse                 835 non-null    float64
3   Num_of_pregnancies                       835 non-null    float64
4   Smokes                                     835 non-null    float64
5   Smokes_(years)                          835 non-null    float64
6   Smokes_(packs/year)                     835 non-null    float64
7   Hormonal_Contraceptives                 835 non-null    float64
8   Hormonal_Contraceptives_(years)         835 non-null    float64
9   IUD                                       835 non-null    float64
10  IUD_(years)                             835 non-null    float64
11  STDs                                     835 non-null    float64
12  STDs_(number)                          835 non-null    float64
13  STDs:condylomatosis                    835 non-null    float64
14  STDs:cervical_condylomatosis           835 non-null    float64
15  STDs:vaginal_condylomatosis            835 non-null    float64
16  STDs:vulvo-perineal_condylomatosis     835 non-null    float64
17  STDs:syphilis                         835 non-null    float64
18  STDs:pelvic_inflammatory_disease       835 non-null    float64
19  STDs:genital_herpes                   835 non-null    float64
20  STDs:molluscum_contagiosum             835 non-null    float64
21  STDs:AIDS                             835 non-null    float64
22  STDs:HIV                             835 non-null    float64
23  STDs:Hepatitis_B                     835 non-null    float64
24  STDs:HPV                             835 non-null    float64
25  STDs:_Number_of_diagnosis              835 non-null    int64
26  STDs:_Time_since_first_diagnosis        827 non-null    float64
27  STDs:_Time_since_last_diagnosis         827 non-null    float64
28  Dx:Cancer                             835 non-null    int64
```

```
29 Dx:CIN 835 non-null int64
30 Dx:HPV 835 non-null int64
31 Dx 835 non-null int64
32 Hinselmann 835 non-null int64
33 Schiller 835 non-null int64
34 Citology 835 non-null int64
35 Biopsy 835 non-null int64
dtypes: float64(26), int64(10)
memory usage: 241.4 KB
```

```
# else the data would be filled with the mean per age
cervical_cancer['STDs:_Time_since_first_diagnosis'] = cervical_cancer.groupby('Age')['STDs:_Time_since_first_diagnosis'].transform(lambda x:
cervical_cancer['STDs:_Time_since_last_diagnosis'] = cervical_cancer.groupby('Age')['STDs:_Time_since_last_diagnosis'].transform(lambda x: x
cervical_cancer.info()
# all missing data are handled
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 835 entries, 0 to 857
Data columns (total 36 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age                                    835 non-null    int64
1   Number_of_sexual_partners             835 non-null    float64
2   First_sexual_intercourse              835 non-null    float64
3   Num_of_pregnancies                    835 non-null    float64
4   Smokes                                835 non-null    float64
5   Smokes_(years)                        835 non-null    float64
6   Smokes_(packs/year)                   835 non-null    float64
7   Hormonal_Contraceptives                835 non-null    float64
8   Hormonal_Contraceptives_(years)        835 non-null    float64
9   IUD                                    835 non-null    float64
10  IUD_(years)                           835 non-null    float64
11  STDs                                    835 non-null    float64
12  STDs_(number)                         835 non-null    float64
13  STDs:condylomatosis                   835 non-null    float64
14  STDs:cervical_condylomatosis          835 non-null    float64
15  STDs:vaginal_condylomatosis           835 non-null    float64
16  STDs:vulvo-perineal_condylomatosis    835 non-null    float64
17  STDs:syphilis                         835 non-null    float64
18  STDs:pelvic_inflammatory_disease      835 non-null    float64
19  STDs:genital_herpes                   835 non-null    float64
20  STDs:molluscum_contagiosum            835 non-null    float64
21  STDs:AIDS                             835 non-null    float64
22  STDs:HIV                              835 non-null    float64
23  STDs:Hepatitis_B                      835 non-null    float64
24  STDs:HPV                              835 non-null    float64
25  STDs:_Number_of_diagnosis              835 non-null    int64
26  STDs:_Time_since_first_diagnosis       835 non-null    float64
27  STDs:_Time_since_last_diagnosis        835 non-null    float64
28  Dx:Cancer                             835 non-null    int64
29  Dx:CIN                                835 non-null    int64
30  Dx:HPV                                835 non-null    int64
31  Dx                                     835 non-null    int64
32  Hinselmann                            835 non-null    int64
33  Schiller                              835 non-null    int64
34  Citology                              835 non-null    int64
35  Biopsy                                835 non-null    int64
dtypes: float64(26), int64(10)
memory usage: 241.4 KB
```

Exploratory Data Analysis

```
cervical_cancer.describe()
"""
According to this data, the average age is 27
On average, a person has 3 sexual partners,
They experienced their first sexual intercourse at 17
And females have an average of 2 pregancies
"""
```

	Age	Number_of_sexual_partners	First_sexual_intercourse	Num_of_pregnanci
count	835.000000	835.000000	835.000000	835.0000
mean	27.023952	2.554967	17.022322	2.2942
std	8.482986	1.653366	2.806011	1.4223
min	13.000000	1.000000	10.000000	0.0000
25%	21.000000	2.000000	15.000000	1.0000
50%	26.000000	2.000000	17.000000	2.0000
75%	32.000000	3.000000	18.000000	3.0000
max	84.000000	28.000000	32.000000	11.0000

8 rows × 36 columns

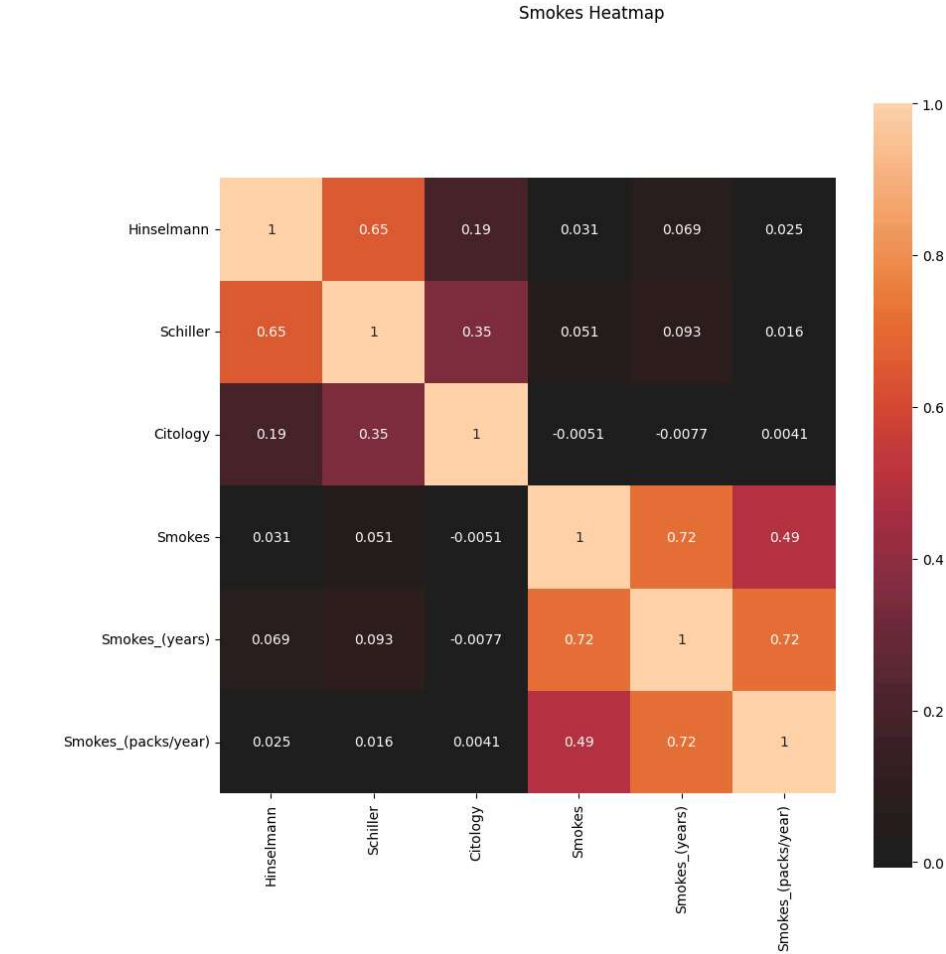
```
# separating by categories (smokes, hormonal contraceptives, IUDs, STDs, and Dx)
# as per the documentation, the target variables are the last 4 columns

# getting the target variables
target_variables = cervical_cancer.iloc[:, 32:35]

# getting the categories
c_smokes = cervical_cancer.iloc[:, 4:7]
c_hormonal_contraceptives = cervical_cancer.iloc[:, 7:9]
c_iuds = cervical_cancer.iloc[:, 9:11]
c_stds = cervical_cancer.iloc[:, 11:28]
c_dx = cervical_cancer.iloc[:, 28:32]
```

```
# getting the heatmap per category
plt.figure(figsize=(10,10))
sns.heatmap(
    pd.concat([target_variables, c_smokes], axis = 1).corr(),
    annot=True, center=0, square=True
)
plt.suptitle('Smokes Heatmap')

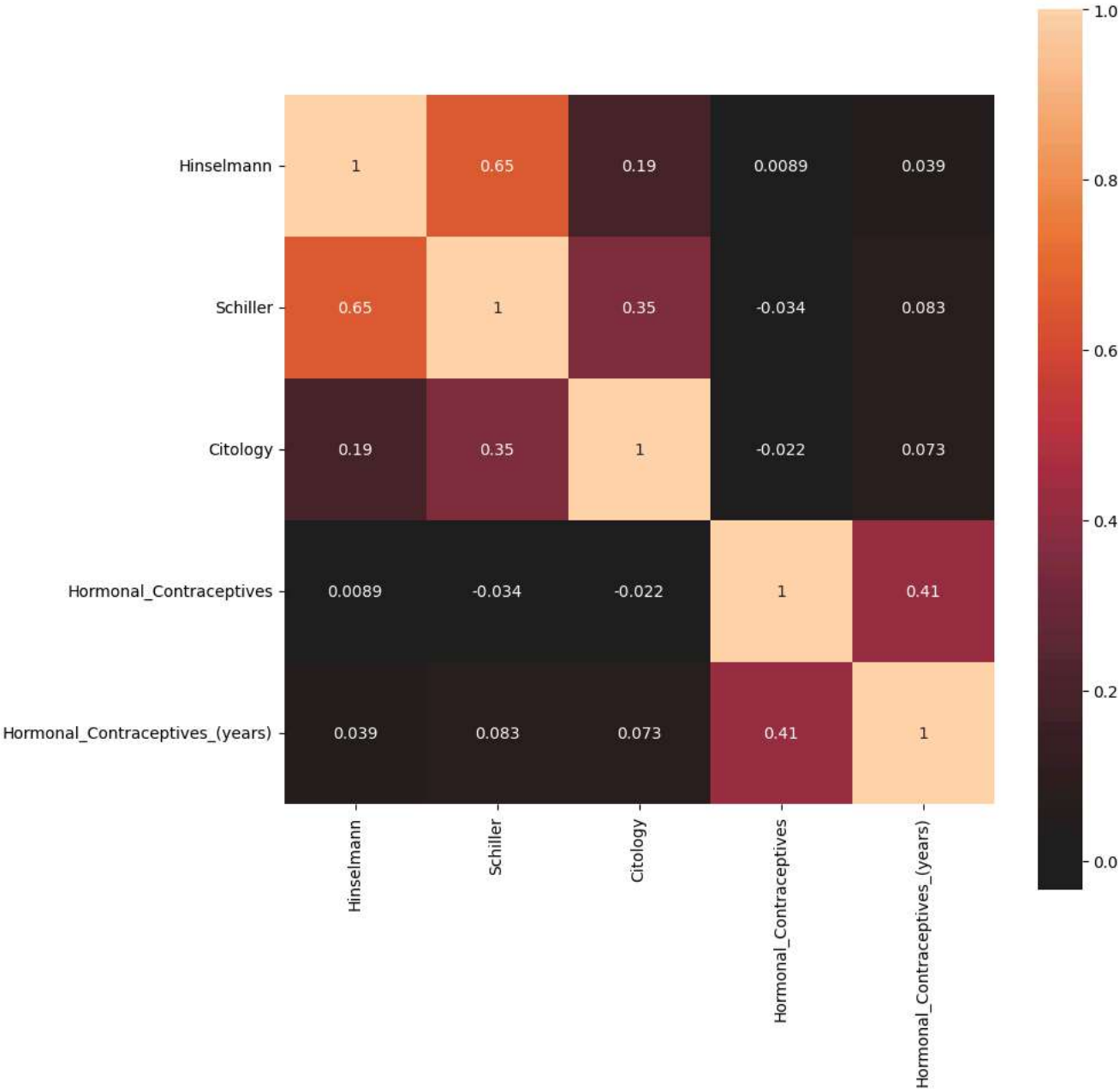
Text(0.5, 0.98, 'Smokes Heatmap')
```



```
plt.figure(figsize=(10,10))
sns.heatmap(
    pd.concat([target_variables, c_hormonal_contraceptives], axis = 1).corr(),
    annot=True, center=0, square=True
)
plt.suptitle('Hormonal Contraceptives Heatmap')
```

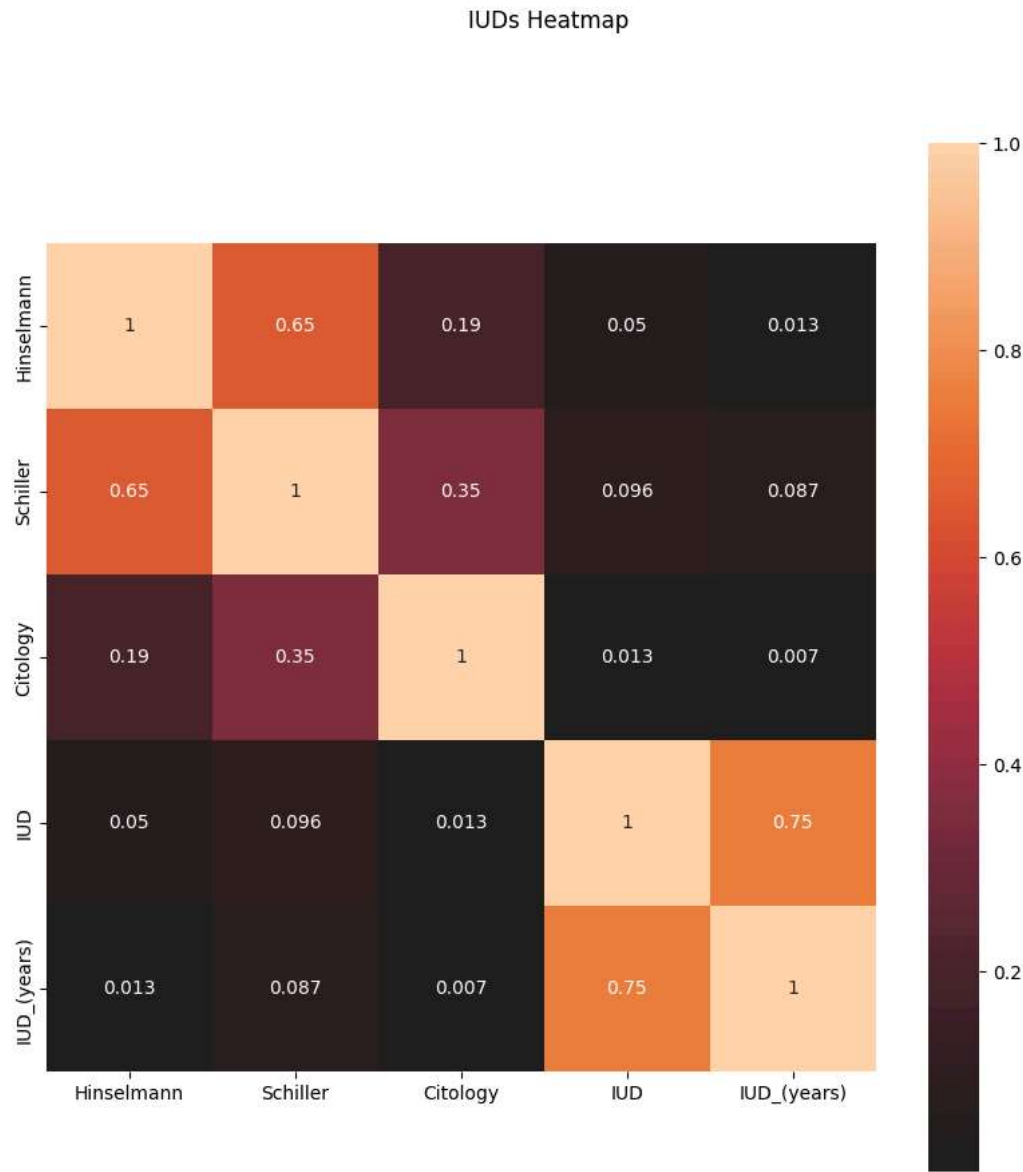
Text(0.5, 0.98, 'Hormonal Contraceptives Heatmap')

Hormonal Contraceptives Heatmap



```
plt.figure(figsize=(10,10))
sns.heatmap(
    pd.concat([target_variables, c_iuds], axis = 1).corr(),
    annot=True, center=0, square=True
)
plt.suptitle('IUDs Heatmap')
```

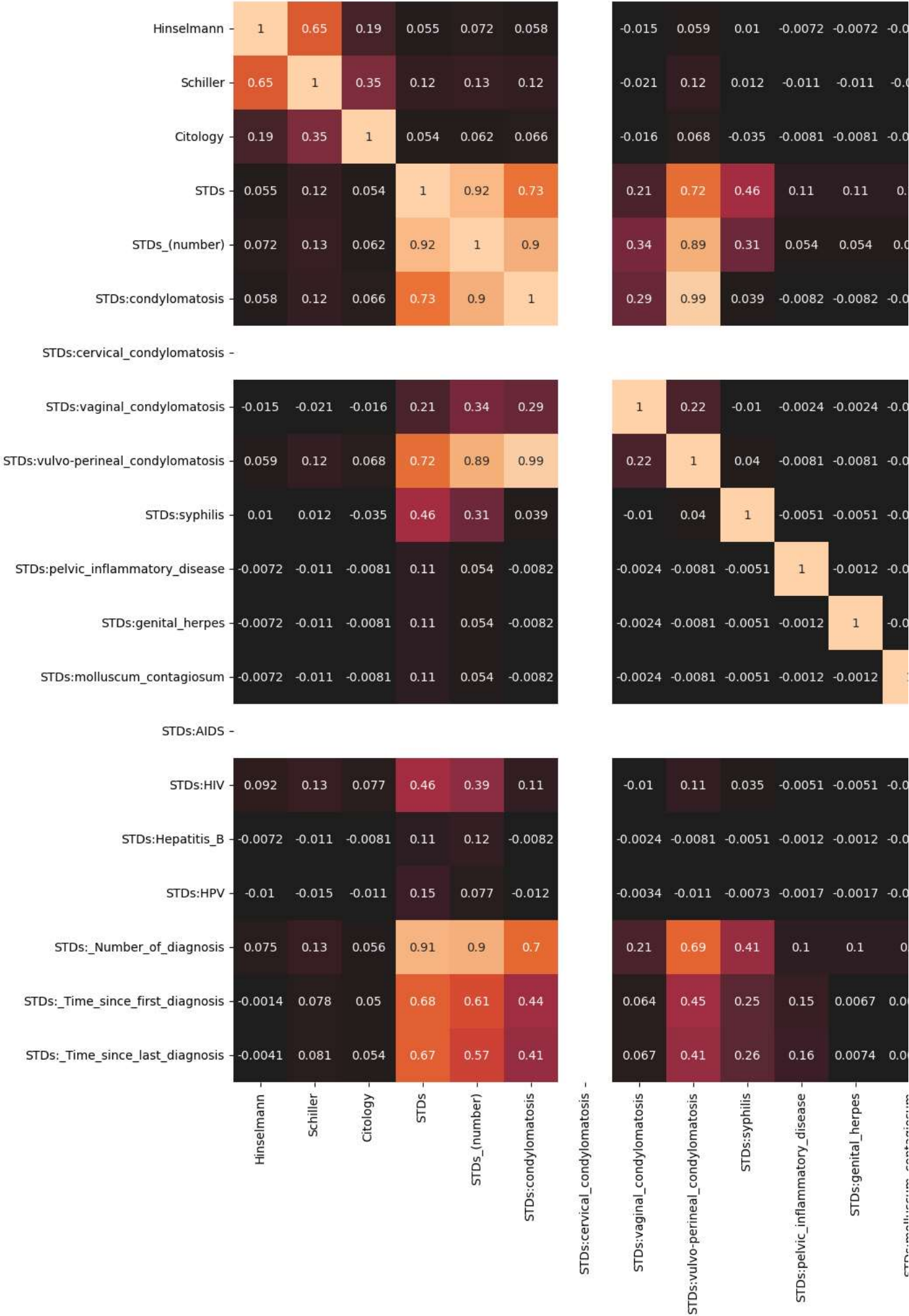

Text(0.5, 0.98, 'IUDs Heatmap')



```
plt.figure(figsize=(20,20))
sns.heatmap(
    pd.concat([target_variables, c_stds], axis = 1).corr(),
    annot=True, center=0, square=True
)
plt.suptitle('STDs Heatmap')
# AIDS and Cervical Condylomatosis are white since all values are 0
```

Text(0.5, 0.98, 'STDs Heatmap')

STDs Heat



```
plt.figure(figsize=(10,10))
sns.heatmap(
    pd.concat([target_variables, c_dx], axis = 1).corr(),
    annot=True, center=0, square=True
)
plt.suptitle('Dx Heatmap')

Text(0.5, 0.98, 'Dx Heatmap')
```

