SETUP x IMPORT Dataset

```
!pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages (0.0.6)
```

```
!pip install hvplot
```

```
Requirement already satisfied: hvplot in /usr/local/lib/python3.10/dist-packages (0.9.2)
Requirement already satisfied: bokeh>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (3.3.4)
Requirement already satisfied: colorcet>=2 in /usr/local/lib/python3.10/dist-packages (from hvplot) (3.1.0)
Requirement already satisfied: holoviews>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (1.17.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from hvplot) (2.0.3)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.10/dist-packages (from hvplot) (1.25.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from hvplot) (24.0)
Requirement already satisfied: panel>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (1.3.8)
Requirement already satisfied: param<3.0,>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (2.1.0)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (3.1.3)
Requirement already satisfied: contourpy>=1 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (1.2.1)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (9.4.0)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (6.0.1)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (6.3.3)
Requirement already satisfied: xyzservices>=2021.09.1 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (2024.4.0)
Requirement already satisfied: pyviz-comms>=0.7.4 in /usr/local/lib/python3.10/dist-packages (from holoviews>=1.11.0->hvplot) (3.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->hvplot) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->hvplot) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->hvplot) (2024.1)
Requirement already satisfied: markdown in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (3.6)
Requirement already satisfied: markdown-it-py in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (3.0.0)
Requirement already satisfied: linkify-it-py in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (2.0.3)
Requirement already satisfied: mdit-py-plugins in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (0.4.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (2.31.0)
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (4.66.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (6.1.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (4.11.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->bokeh>=1.0.0->hvplot) (2.1.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->hvplot) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->panel>=0.11.0->hvplot) (0.5.1)
Requirement already satisfied: uc-micro-py in /usr/local/lib/python3.10/dist-packages (from linkify-it-py->panel>=0.11.0->hvplot) (1.0.3)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py->panel>=0.11.0->hvplot) (0.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (2024.2.2)
```

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

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression

%matplotlib inline
```

```python
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)
```

```
13                  STDs:condylomatosis  Feature  Continuous        None
14         STDs:cervical condylomatosis  Feature  Continuous        None
15          STDs:vaginal condylomatosis  Feature  Continuous        None
16  STDs:vulvo-perineal condylomatosis  Feature  Continuous        None
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             no
1         None  None            yes
2         None  None            yes
3         None  None            yes
4         None  None            yes
5         None  None            yes
6         None  None            yes
7         None  None            yes
8         None  None            yes
9         None  None            yes
10        None  None            yes
11        None  None            yes
12        None  None            yes
13        None  None            yes
14        None  None            yes
15        None  None            yes
16        None  None            yes
17        None  None            yes
18        None  None            yes
19        None  None            yes
20        None  None            yes
21        None  None            yes
22        None  None            yes
23        None  None            yes
24        None  None            yes
25        None  None             no
26        None  None            yes
27        None  None            yes
28        None  None             no
29        None  None             no
30        None  None             no
31        None  None             no
32        None  None             no
```

DATA WRANGLIN

```python
X
```

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives	Hormonal Contraceptives (years)	IUD	...	STDs: Time since first diagnosis	STDs: Time since last diagnosis	Dx:Cancer	Dx:CIN	Dx:HPV	Dx
0	18	4.0	15.0	1.0	0.0	0.0	0.0	0.0	0.00	0.0	...	NaN	NaN	0	0	0	0
1	15	1.0	14.0	1.0	0.0	0.0	0.0	0.0	0.00	0.0	...	NaN	NaN	0	0	0	0
2	34	1.0	NaN	1.0	0.0	0.0	0.0	0.0	0.00	0.0	...	NaN	NaN	0	0	0	0
3	52	5.0	16.0	4.0	1.0	37.0	37.0	1.0	3.00	0.0	...	NaN	NaN	1	0	1	0
4	46	3.0	21.0	4.0	0.0	0.0	0.0	1.0	15.00	0.0	...	NaN	NaN	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
853	34	3.0	18.0	0.0	0.0	0.0	0.0	0.0	0.00	0.0	...	NaN	NaN	0	0	0	0
854	32	2.0	19.0	1.0	0.0	0.0	0.0	1.0	8.00	0.0	...	NaN	NaN	0	0	0	0
855	25	2.0	17.0	0.0	0.0	0.0	0.0	1.0	0.08	0.0	...	NaN	NaN	0	0	0	0
856	33	2.0	24.0	2.0	0.0	0.0	0.0	1.0	0.08	0.0	...	NaN	NaN	0	0	0	0

```python
print(y) # y doesnt have anything in it, ignored
```

```
None
```

```python
cc = X
```

```python
print(f'there are {cc.shape[0]} rows and {cc.shape[1]} columns \n')
cc.info()
```

there are 858 rows and 36 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 858 entries, 0 to 857
Data columns (total 36 columns):
 #   Column                              Non-Null Count  Dtype  
---  ------                              --------------  -----  
 0   Age                                 858 non-null    int64  
 1   Number of sexual partners           832 non-null    float64
 2   First sexual intercourse            851 non-null    float64
 3   Num of pregnancies                  802 non-null    float64
 4   Smokes                              845 non-null    float64
 5   Smokes (years)                      845 non-null    float64
 6   Smokes (packs/year)                 845 non-null    float64
 7   Hormonal Contraceptives             750 non-null    float64
 8   Hormonal Contraceptives (years)     750 non-null    float64
 9   IUD                                 741 non-null    float64
 10  IUD (years)                         741 non-null    float64
 11  STDs                                753 non-null    float64
 12  STDs (number)                       753 non-null    float64
 13  STDs:condylomatosis                 753 non-null    float64
 14  STDs:cervical condylomatosis        753 non-null    float64
 15  STDs:vaginal condylomatosis         753 non-null    float64
 16  STDs:vulvo-perineal condylomatosis  753 non-null    float64
 17  STDs:syphilis                       753 non-null    float64
 18  STDs:pelvic inflammatory disease    753 non-null    float64
 19  STDs:genital herpes                 753 non-null    float64
 20  STDs:molluscum contagiosum          753 non-null    float64
 21  STDs:AIDS                           753 non-null    float64
 22  STDs:HIV                            753 non-null    float64
 23  STDs:Hepatitis B                    753 non-null    float64
 24  STDs:HPV                            753 non-null    float64
 25  STDs: Number of diagnosis           858 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                           858 non-null    int64  
 29  Dx:CIN                              858 non-null    int64  
 30  Dx:HPV                              858 non-null    int64  
 31  Dx                                  858 non-null    int64  
 32  Hinselmann                          858 non-null    int64  
 33  Schiller                            858 non-null    int64  
 34  Citology                            858 non-null    int64  
 35  Biopsy                              858 non-null    int64  
dtypes: float64(26), int64(10)
memory usage: 241.4 KB
```

```
for col in cc.columns:
  print(f"Unique values in '{col}':\n {cc[col].unique()}")
```

```
 [ 0.  1. nan]
Unique values in 'Hormonal Contraceptives (years)':
 [ 0.         3.        15.         2.         8.        10.
  5.         0.25       7.        22.        19.         0.5
  1.         0.58       9.        13.        11.         4.
 12.        16.         0.33            nan  0.16       14.
  0.08       2.28220052  0.66       6.         1.5        0.42
  0.67       0.75       2.5        4.5        6.5        0.17
 20.         3.5        0.41      30.        17.        ]
Unique values in 'IUD':
 [ 0.  1. nan]
Unique values in 'IUD (years)':
 [ 0.    7.      nan  5.    8.    6.    1.    0.58  2.   19.    0.5  17.
  0.08  0.25 10.   11.    3.   15.   12.    9.    1.5  0.91  4.    0.33
  0.41  0.16  0.17]
Unique values in 'STDs':
 [ 0.  1. nan]
Unique values in 'STDs (number)':
 [ 0.  2.  1. nan  3.  4.]
Unique values in 'STDs:condylomatosis':
 [ 0.  1. nan]
Unique values in 'STDs:cervical condylomatosis':
 [ 0. nan]
Unique values in 'STDs:vaginal condylomatosis':
 [ 0. nan  1.]
Unique values in 'STDs:vulvo-perineal condylomatosis':
 [ 0.  1. nan]
Unique values in 'STDs:syphilis':
 [ 0.  1. nan]
Unique values in 'STDs:pelvic inflammatory disease':
 [ 0. nan  1.]
Unique values in 'STDs:genital herpes':
 [ 0. nan  1.]
Unique values in 'STDs:molluscum contagiosum':
 [ 0. nan  1.]
Unique values in 'STDs:AIDS':
 [ 0. nan]
Unique values in 'STDs:HIV':
 [ 0.  1. nan]
Unique values in 'STDs:Hepatitis B':
 [ 0. nan  1.]
Unique values in 'STDs:HPV':
 [ 0. nan  1.]
Unique values in 'STDs: Number of diagnosis':
 [0 1 3 2]
Unique values in 'STDs: Time since first diagnosis':
 [nan 21.  2. 15. 19.  3. 12.  1. 11.  9.  7.  8. 16.  6.  5. 10.  4. 22.
 18.]
Unique values in 'STDs: Time since last diagnosis':
 [nan 21.  2. 15. 19.  3. 12.  1. 11.  9.  7.  8. 16.  6.  5. 10.  4. 22.
 18.]
Unique values in 'Dx:Cancer':
 [0 1]
Unique values in 'Dx:CIN':
 [0 1]
Unique values in 'Dx:HPV':
 [0 1]
Unique values in 'Dx':
```

```
cc = cc.drop(columns = ['First sexual intercourse', 'Dx', 'Smokes', 'Hormonal Contraceptives', 'IUD',
                        'STDs:condylomatosis', 'STDs:vaginal condylomatosis',
                        'STDs:vulvo-perineal condylomatosis', 'STDs:syphilis', 'STDs:pelvic inflammatory disease', 'STDs:genital herpes', 'STDs:molluscum contagiosum',
                        'STDs:Hepatitis B', 'STDs:HPV','STDs: Time since first diagnosis', 'STDs: Time since last diagnosis', 'Hinselmann', 'Schiller', 'Citology', 'Biopsy'])
```

```
print(f'there are now {cc.shape[0]} rows and {cc.shape[1]} columns \n')
cc.info()  # dropped the columns I won't be needing
```

there are now 858 rows and 16 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 858 entries, 0 to 857
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Age                             858 non-null    int64  
 1   Number of sexual partners       832 non-null    float64
 2   Num of pregnancies              802 non-null    float64
 3   Smokes (years)                  845 non-null    float64
 4   Smokes (packs/year)             845 non-null    float64
 5   Hormonal Contraceptives (years) 750 non-null    float64
 6   IUD (years)                     741 non-null    float64
 7   STDs                            753 non-null    float64
 8   STDs (number)                   753 non-null    float64
 9   STDs:cervical condylomatosis    753 non-null    float64
 10  STDs:AIDS                       753 non-null    float64
 11  STDs:HIV                        753 non-null    float64
 12  STDs: Number of diagnosis       858 non-null    int64  
 13  Dx:Cancer                       858 non-null    int64  
 14  Dx:CIN                          858 non-null    int64  
 15  Dx:HPV                          858 non-null    int64  
dtypes: float64(11), int64(5)
memory usage: 107.4 KB
```

```
cc.isnull().sum() # checking for null values
```

```
Age                                 0
Number of sexual partners          26
Num of pregnancies                 56
Smokes (years)                     13
Smokes (packs/year)                13
Hormonal Contraceptives (years)   108
IUD (years)                       117
STDs                              105
STDs (number)                     105
STDs:cervical condylomatosis      105
STDs:AIDS                         105
STDs:HIV                          105
STDs: Number of diagnosis           0
Dx:Cancer                           0
Dx:CIN                              0
Dx:HPV                              0
dtype: int64
```

```
cc.mode()
```

	Age	Number of sexual partners	Num of pregnancies	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives (years)	IUD (years)	STDs	STDs (number)	STDs:cervical condylomatosis	STDs:AIDS	STDs:HIV	STDs: Number of diagnosis	Dx:Cancer	Dx:CIN	Dx:HPV
0	18	2.0	2.0	1.2	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0

```
# assumes and change the nulls to 0 since majority (mode) of the columns are 0
cc.fillna(0, inplace=True)
```

```
cc.isnull().sum()
```

```
Age                               0
Number of sexual partners         0
Num of pregnancies                0
Smokes (years)                    0
Smokes (packs/year)               0
Hormonal Contraceptives (years)   0
IUD (years)                       0
STDs                              0
STDs (number)                     0
STDs:cervical condylomatosis      0
STDs:AIDS                         0
STDs:HIV                          0
STDs: Number of diagnosis         0
```

```
Dx:Cancer                           0
Dx:CIN                              0
Dx:HPV                              0
dtype: int64
```

```
for col in cc.columns:
  print(f"Unique values in '{col}':\n {cc[col].unique()}")
```

```
 [18 15 34 52 46 42 31 28 43 44 27 45 40 41 35 37 36 30 33 39 31 32 50 23
 28 29 20 25 21 24 22 48 19 17 16 14 59 79 84 47 13 70 50 49]
Unique values in 'Number of sexual partners':
 [ 4.  1.  5.  3.  2.  6.  0.  7. 15.  8. 10. 28.  9.]
Unique values in 'Num of pregnancies':
 [ 1.  4.  2.  6.  3.  5.  0.  8.  7. 11. 10.]
Unique values in 'Smokes (years)':
 [ 0.         37.         34.          1.26697291  3.         12.
 18.          7.         19.         21.         15.         13.
 16.          8.          4.         10.         22.         14.
  0.5        11.          9.          2.          5.          6.
  1.         32.         24.         28.         20.          0.16      ]
Unique values in 'Smokes (packs/year)':
 [0.00000000e+00 3.70000000e+01 3.40000000e+00 2.80000000e+00
 4.00000000e-02 5.13202128e-01 2.40000000e+00 6.00000000e+00
 9.00000000e+00 1.60000000e+00 1.90000000e+01 2.10000000e+01
 3.20000000e-01 2.60000000e+00 8.00000000e-01 1.50000000e+01
 2.00000000e+00 5.70000000e+00 1.00000000e+00 3.30000000e+00
 3.50000000e+00 1.20000000e+01 2.50000000e-02 2.75000000e+00
 2.00000000e-01 1.40000000e+00 5.00000000e+00 2.10000000e+00
 7.00000000e-01 1.20000000e+00 7.50000000e+00 1.25000000e+00
 3.00000000e+00 7.50000000e-01 1.00000000e-01 8.00000000e+00
 2.25000000e+00 3.00000000e-03 7.00000000e+00 4.50000000e-01
 1.50000000e-01 5.00000000e-02 2.50000000e-01 4.80000000e+00
 4.50000000e+00 4.00000000e-01 3.70000000e-01 2.20000000e+00
 1.60000000e-01 9.00000000e-01 2.20000000e+01 1.35000000e+00
 5.00000000e-01 2.50000000e+00 4.00000000e+00 1.30000000e+00
 1.65000000e+00 2.70000000e+00 1.00000000e-03 7.60000000e+00
 5.50000000e+00 3.00000000e-01]
Unique values in 'Hormonal Contraceptives (years)':
 [ 0.         3.         15.         2.         8.         10.
  5.         0.25       7.         22.         19.         0.5
  1.         0.58       9.         13.         11.         4.
 12.        16.         0.33       0.16       14.         0.08
  2.28220052  0.66       6.         1.5        0.42       0.67
  0.75       2.5        4.5        6.5        0.17       20.
  3.5        0.41      30.         17.        ]
Unique values in 'IUD (years)':
 [ 0.   7.   5.   8.   6.   1.   0.58  2.   19.   0.5  17.   0.08
  0.25 10.   11.   3.   15.   12.   9.   1.5  0.91  4.   0.33  0.41
  0.16  0.17]
Unique values in 'STDs':
 [0. 1.]
Unique values in 'STDs (number)':
 [0. 2. 1. 3. 4.]
Unique values in 'STDs:cervical condylomatosis':
 [0.]
Unique values in 'STDs:AIDS':
 [0.]
Unique values in 'STDs:HIV':
 [0. 1.]
Unique values in 'STDs: Number of diagnosis':
 [0 1 3 2]
Unique values in 'Dx:Cancer':
 [0 1]
Unique values in 'Dx:CIN':
 [0 1]
Unique values in 'Dx:HPV':
 [0 1]
```

```
cc
```

	Age	Number of sexual partners	Num of pregnancies	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives (years)	IUD (years)	STDs	STDs (number)	STDs:cervical condylomatosis	STDs:AIDS	STDs:HIV	STDs: Number of diagnosis	Dx:Cancer	Dx:CIN	Dx:HPV
0	18	4.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
1	15	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
2	34	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
3	52	5.0	4.0	37.0	37.0	3.00	0.0	0.0	0.0	0.0	0.0	0.0	0	1	0	1
4	46	3.0	4.0	0.0	0.0	15.00	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
853	34	3.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
854	32	2.0	1.0	0.0	0.0	8.00	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
855	25	2.0	0.0	0.0	0.0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0
856	33	2.0	2.0	0.0	0.0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0

Next steps: View recommended plots

TRAINING

for Cancer Diagnosis (Dx:Cancer)

```
x = cc.drop(columns = ['Dx:Cancer','Dx:CIN','Dx:HPV'])
x
```

	Age	Number of sexual partners	Num of pregnancies	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives (years)	IUD (years)	STDs	STDs (number)	STDs:cervical condylomatosis	STDs:AIDS	STDs:HIV	STDs: Number of diagnosis
0	18	4.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
1	15	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
2	34	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
3	52	5.0	4.0	37.0	37.0	3.00	0.0	0.0	0.0	0.0	0.0	0.0	0
4	46	3.0	4.0	0.0	0.0	15.00	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
853	34	3.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
854	32	2.0	1.0	0.0	0.0	8.00	0.0	0.0	0.0	0.0	0.0	0.0	0
855	25	2.0	0.0	0.0	0.0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0
856	33	2.0	2.0	0.0	0.0	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0
857	29	2.0	1.0	0.0	0.0	0.50	0.0	0.0	0.0	0.0	0.0	0.0	0

Next steps: View recommended plots

```
y = cc['Dx:Cancer']
y
```

```
0      0
1      0
2      0
3      1
4      0
      ..
853    0
854    0
855    0
856    0
857    0
Name: Dx:Cancer, Length: 858, dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 57)
```

```
X_train
```

	Age	Number of sexual partners	Num of pregnancies	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives (years)	IUD (years)	STDs	STDs (number)	STDs:cervical condylomatosis	STDs:AIDS	STDs:HIV	STDs: Number of diagnosis
369	18	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
43	36	3.0	3.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
1	15	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
397	19	2.0	2.0	0.0	0.0	3.00	0.0	1.0	2.0	0.0	0.0	0.0	1
678	32	1.0	1.0	0.0	0.0	0.25	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
632	21	2.0	2.0	0.0	0.0	3.00	0.0	0.0	0.0	0.0	0.0	0.0	0
79	35	2.0	3.0	0.0	0.0	0.50	0.0	0.0	0.0	0.0	0.0	0.0	0
406	17	1.0	1.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
5	42	3.0	2.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0
726	29	3.0	3.0	0.0	0.0	10.00	0.0	0.0	0.0	0.0	0.0	0.0	0

Next steps: View recommended plots

```
X_test
```

	Age	Number of sexual partners	Num of pregnancies	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives (years)	IUD (years)	STDs	STDs (number)	STDs:cervical condylomatosis	STDs:AIDS	STDs:HIV	STDs: Number of diagnosis
446	18	3.0	1.0	6.0	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
341	19	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
465	17	1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
120	33	1.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
791	15	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
737	20	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
178	27	4.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
268	23	2.0	3.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0
824	36	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
627	23	1.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

Next steps: View recommended plots

```
log_reg = LogisticRegression(random_state = 0).fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
log_reg.predict(X_train)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0])
```

```
log_reg.score(X_train, y_train)
```

```
0.9783333333333334
```

```
log_reg.score(X_test, y_test)
```

```
0.9806201550387597
```

↑↑↑ it's accurate

for Cervical intraepithelial neoplasia diagnosis (Dx:CIN)

```
y1 = cc['Dx:CIN']
y1
```

```
0      0
1      0
2      0
3      0
4      0
      ..
853    0
854    0
855    0
856    0
857    0
Name: Dx:CIN, Length: 858, dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y1, test_size = 0.3, random_state = 57)
```

```
log_reg = LogisticRegression(random_state = 0).fit(X_train, y_train)
```

```
log_reg.predict(X_train)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0])
```

```
log_reg.score(X_train, y_train)
```

```
0.9866666666666667
```

```
log_reg.score(X_test, y_test)
```

```
0.9961240310077519
```

↑↑↑ it's still accurate

for

```
y2 = cc['Dx:HPV']
y2
```

```
0      0
1      0
2      0
3      1
4      0
      ..
853    0
854    0
855    0
856    0
857    0
Name: Dx:HPV, Length: 858, dtype: int64
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y2, test_size = 0.3, random_state = 57)
```

```
log_reg = LogisticRegression(random_state = 0).fit(X_train, y_train)
```

```
log_reg.predict(X_train)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0])
```

```python
log_reg.score(X_train, y_train)
```

```
0.9783333333333334
```

```python
log_reg.score(X_test, y_test)
```

```
0.9806201550387597
```

↑↑↑ it's still really accurate