CERVICAL CANCER PREDICTION MODEL

DESCRIPTION

In 2018 an estimate of 570,000 women got diagnosed with cervical cancer. Around 311,000 women lost their lives to cervical cancer in 2018. Many factors contribute to the development of cervical cancer. These factors could be high sexual activity, Human papillomavirus (HPV), presence of oral contraceptives, number of children, I.U.D, smoking etc are some of the factors that may contribute to the problem. An early diagnosis of the disease can greatly reduce the number of deaths per annum. It has been reduced by 74% between 1955 to 1992, we can apply AI and ML models to do an early detection of the disease.

In the current model, XGBoost algorithm is used to train a model using the data of 858 patients from “Hospital universitario de Caracas” in Caracas, Venezuela. The dataset was obtained from UCI Machine Learning Repository. So for this model based on XGBoost is given inputs such as age, STDs, IUD, number of pregnancies, etc and the model predicts the target variables such as biopsy.

The following project was divided into the following tasks:

1. Understand the about Cervical Cancer and study related models
2. Import the necessary Libraries and Datasets
3. Perform Analysis of the dataset
4. Data Visualization
5. Data preparation and Model Training
6. Study about XG-Boost Train
7. Evaluate XG-Boost Algorithm

Understand the about Cervical Cancer and study related models

These are the four most common test for cervical cancer diagnosis
Hinselmann : doctors examine the cervix
Schiller : Iodine test is used for cervical cancer diagnosis
Citology : cells from the body is observed under microscope
Biopsy : tissue from the body is removed and observed under the microscope

Factors that contributes to cervical cancer:
Number of sexual partners
First sexual intercourse (age)
Number of pregnancies

Smokes: yes / no
Smokes (years)
Smokes (packs/year)
Hormonal Contraceptives
Hormonal Contraceptives (years), etc.

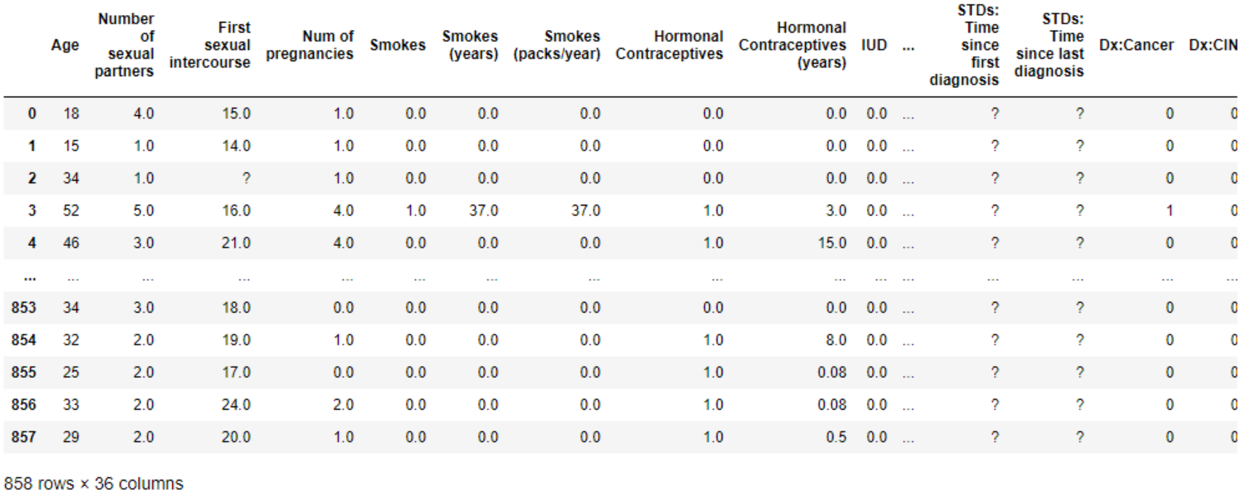
Import the necessary Libraries and Datasets

Step 1: Import the libraries

Libraries used: The various python libraries used for the project are : numpy (for multidimensional array manipulation), scikit-learn, matplotlib, pandas (for DataFrame manipulation), xgboost.

Step 2: Import the dataset and explore it

The dataset hs 858 rows and 36 columns

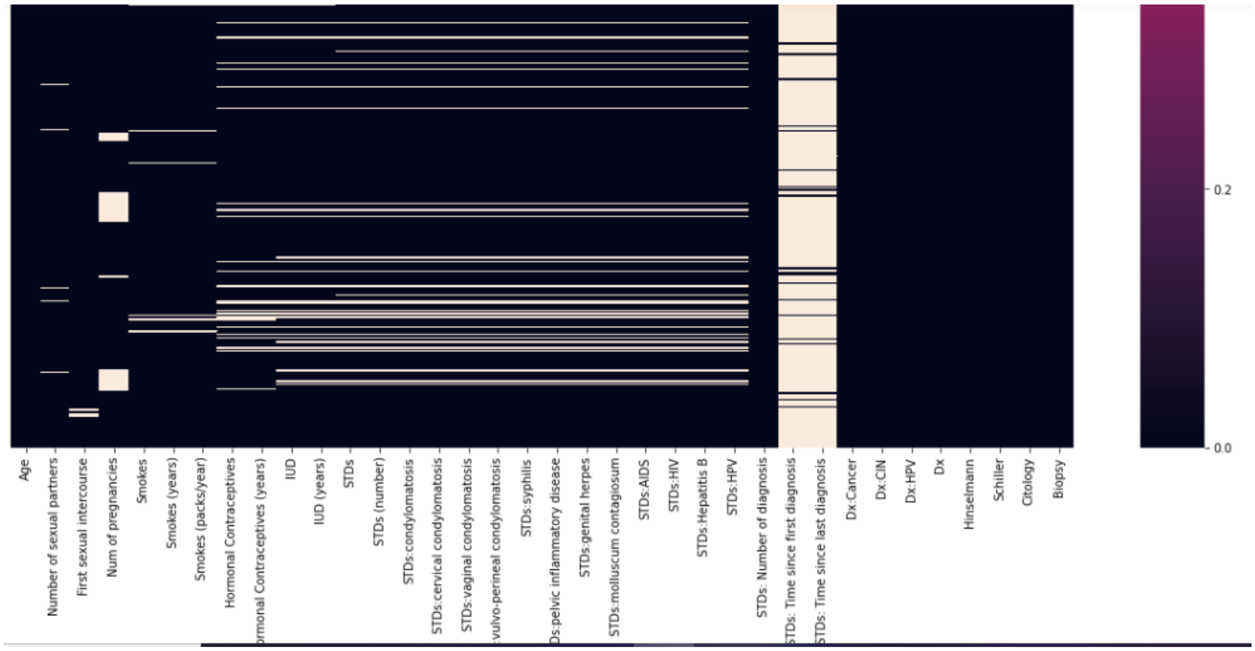


	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives	Hormonal Contraceptives (years)	IUD	...	STDs: Time since first diagnosis	STDs: Time since last diagnosis	Dx:Cancer	Dx:CIN
0	18	4.0	15.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	?	?	0	0
1	15	1.0	14.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	?	?	0	0
2	34	1.0	?	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	?	?	0	0
3	52	5.0	16.0	4.0	1.0	37.0	37.0	1.0	3.0	0.0	...	?	?	1	0
4	46	3.0	21.0	4.0	0.0	0.0	0.0	1.0	15.0	0.0	...	?	?	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
853	34	3.0	18.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	?	?	0	0
854	32	2.0	19.0	1.0	0.0	0.0	0.0	1.0	8.0	0.0	...	?	?	0	0
855	25	2.0	17.0	0.0	0.0	0.0	0.0	1.0	0.08	0.0	...	?	?	0	0
856	33	2.0	24.0	2.0	0.0	0.0	0.0	1.0	0.08	0.0	...	?	?	0	0
857	29	2.0	20.0	1.0	0.0	0.0	0.0	1.0	0.5	0.0	...	?	?	0	0

858 rows × 36 columns

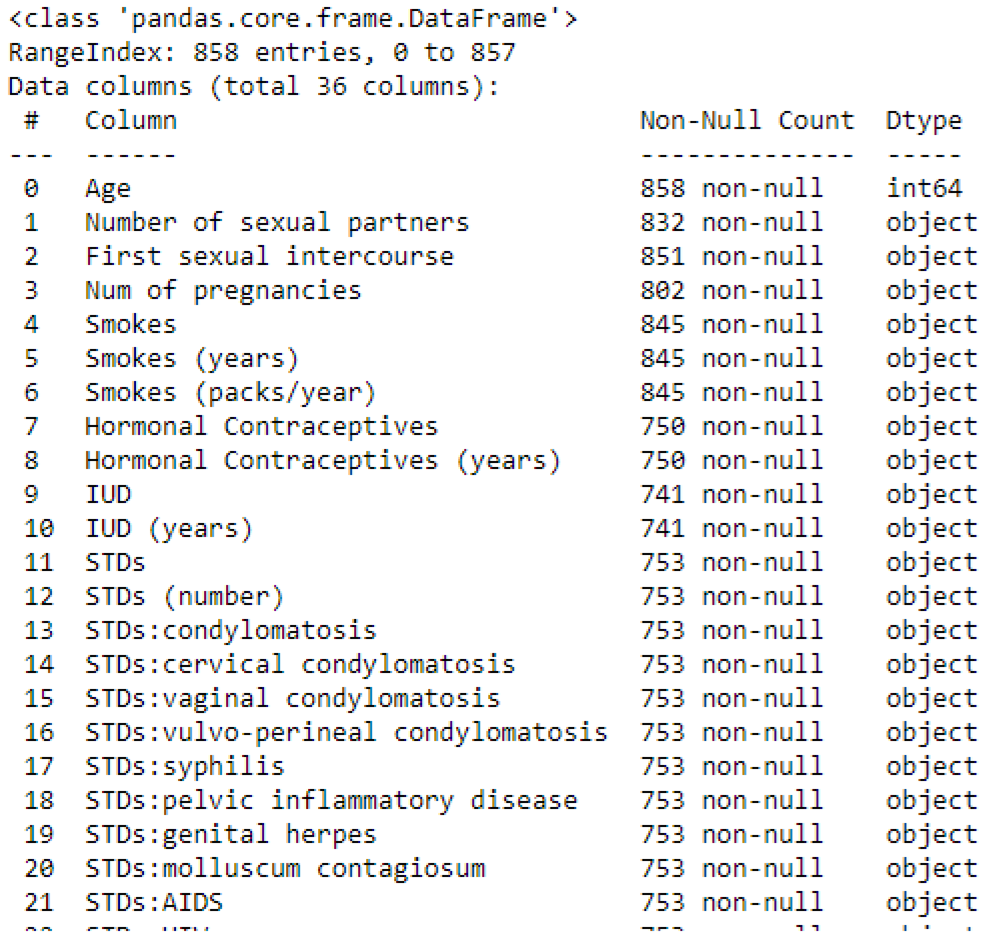
Perform Analysis of the dataset

We explore the data and observe that there are many missing data shown as ‘?’. The ‘?’ was replaced with NaN and we then got the heatmap.



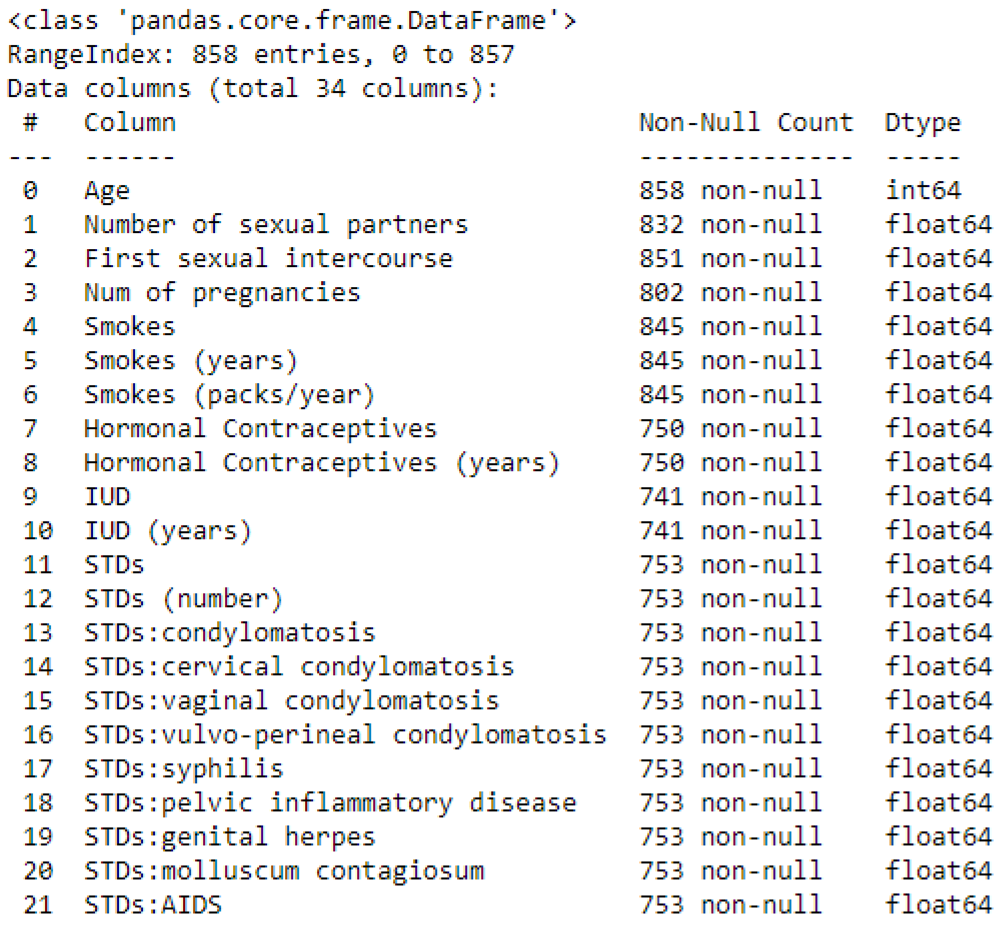
Observing the heatmap allows us to identify the columns with maximum missing data. There were two such columns : STDs: Time since first diagnosis and STDs: Time since last diagnosis. These two columns were dropped.

Getting information about the dataset, we observe the column types as objects.



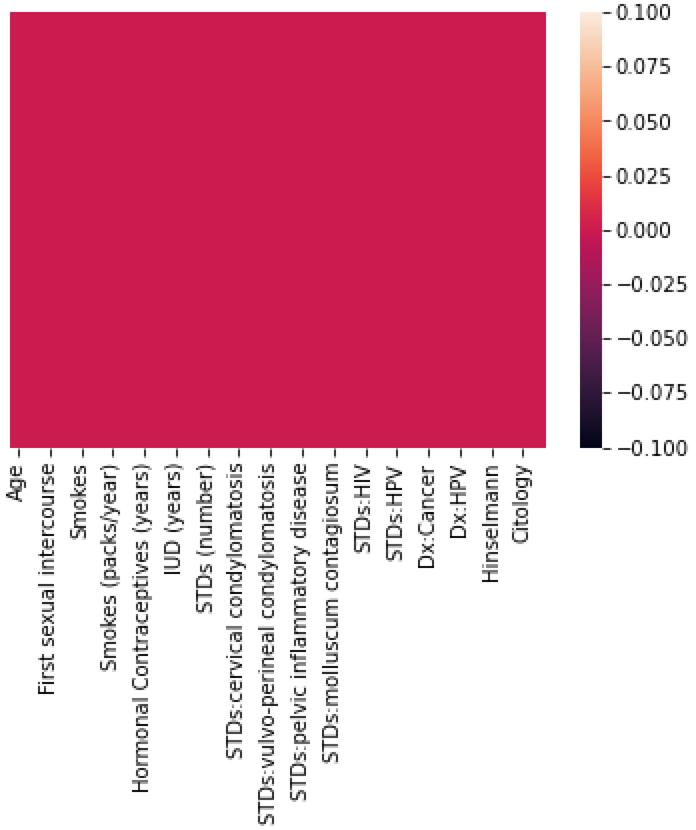
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 858 entries, 0 to 857
Data columns (total 36 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Age                                 858 non-null    int64
 1   Number of sexual partners           832 non-null    object
 2   First sexual intercourse            851 non-null    object
 3   Num of pregnancies                  802 non-null    object
 4   Smokes                              845 non-null    object
 5   Smokes (years)                      845 non-null    object
 6   Smokes (packs/year)                 845 non-null    object
 7   Hormonal Contraceptives             750 non-null    object
 8   Hormonal Contraceptives (years)     750 non-null    object
 9   IUD                                 741 non-null    object
 10  IUD (years)                         741 non-null    object
 11  STDs                                753 non-null    object
 12  STDs (number)                       753 non-null    object
 13  STDs:condylomatosis                 753 non-null    object
 14  STDs:cervical condylomatosis        753 non-null    object
 15  STDs:vaginal condylomatosis         753 non-null    object
 16  STDs:vulvo-perineal condylomatosis  753 non-null    object
 17  STDs:syphilis                       753 non-null    object
 18  STDs:pelvic inflammatory disease    753 non-null    object
 19  STDs:genital herpes                 753 non-null    object
 20  STDs:molluscum contagiosum          753 non-null    object
 21  STDs:AIDS                           753 non-null    object
 22  STDs:HIV                            753 non-null    object
```

These were then converted to numeric types.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 858 entries, 0 to 857
Data columns (total 34 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
```

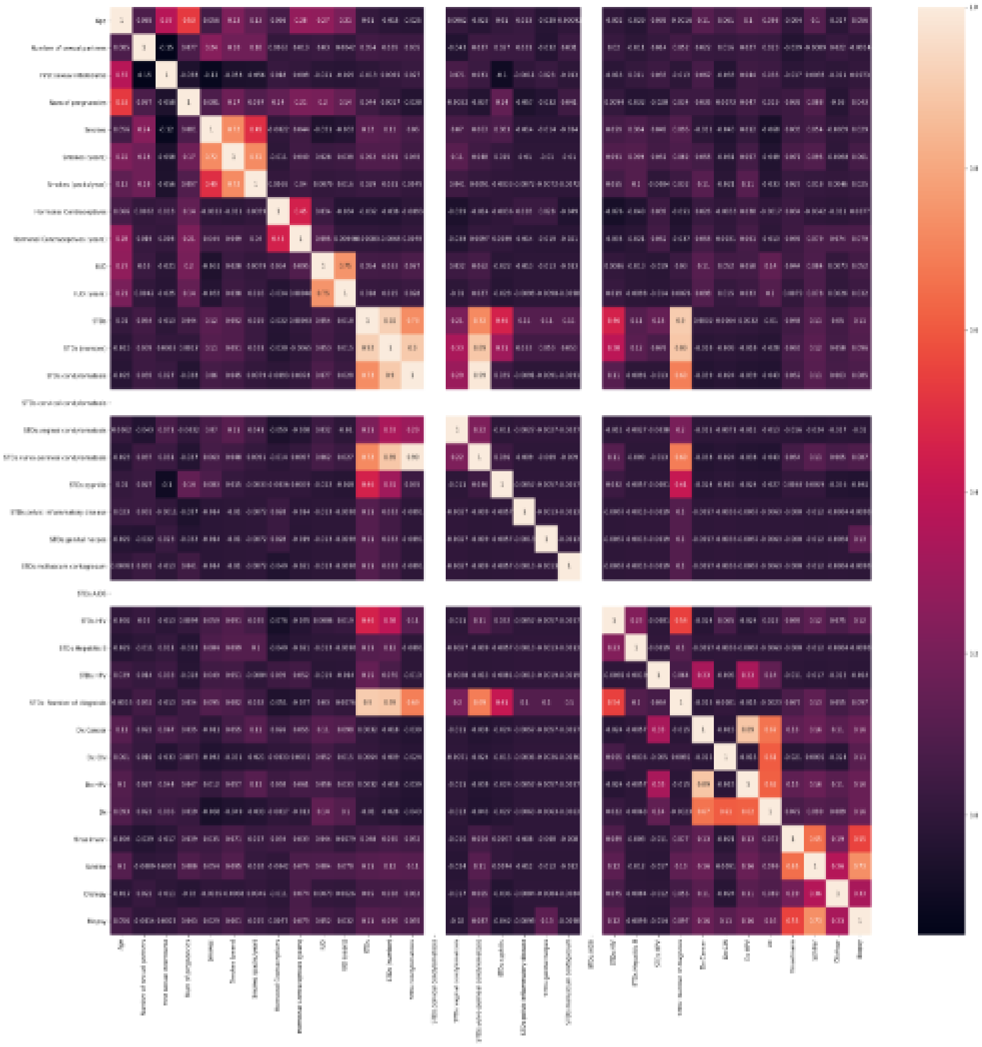
Then the NaN values were replaced with the mean and the heatmap was plotted.



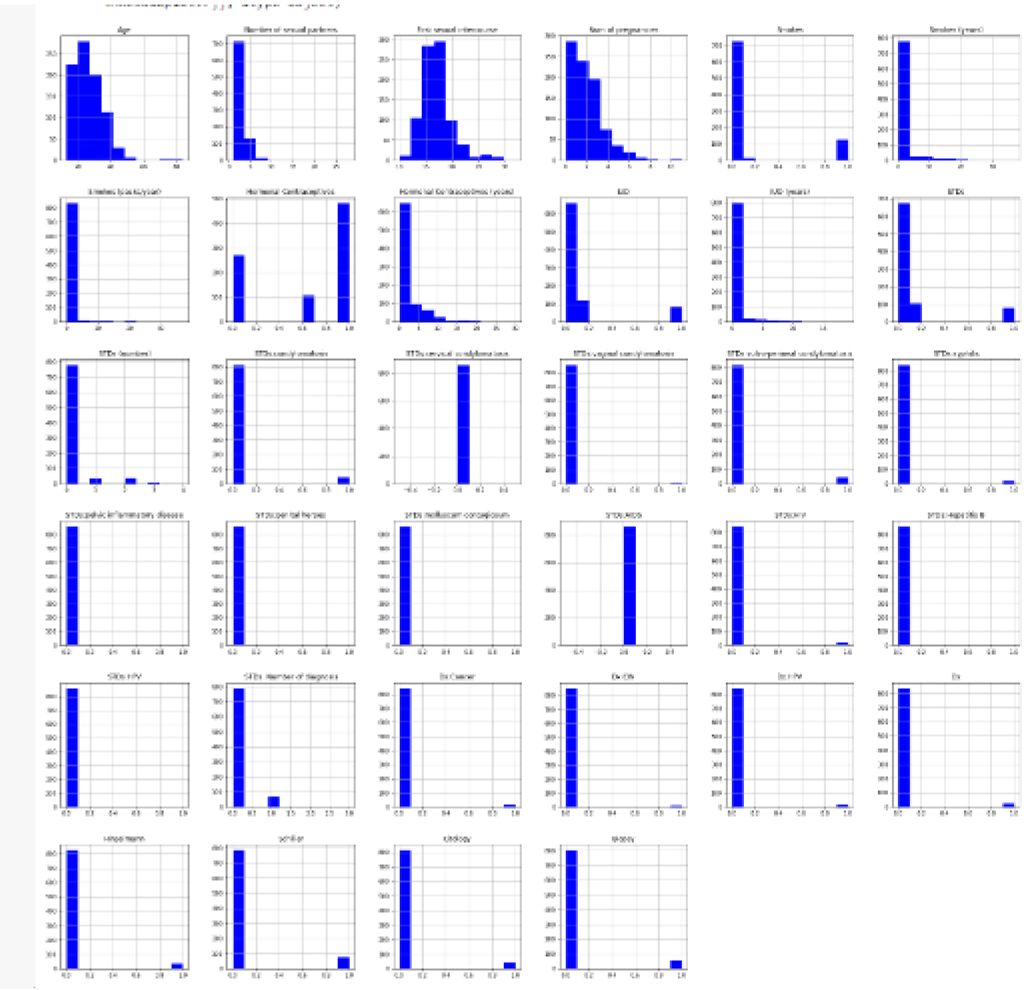
The above heatmap shows that there are no null values (one homogeneous colour is seen) which is exactly what we are looking for.

Data Visualization

We got the correlation matrix for the dataset and plotted it. We observe 1 for perfect correlation and -1 for inverse correlation.

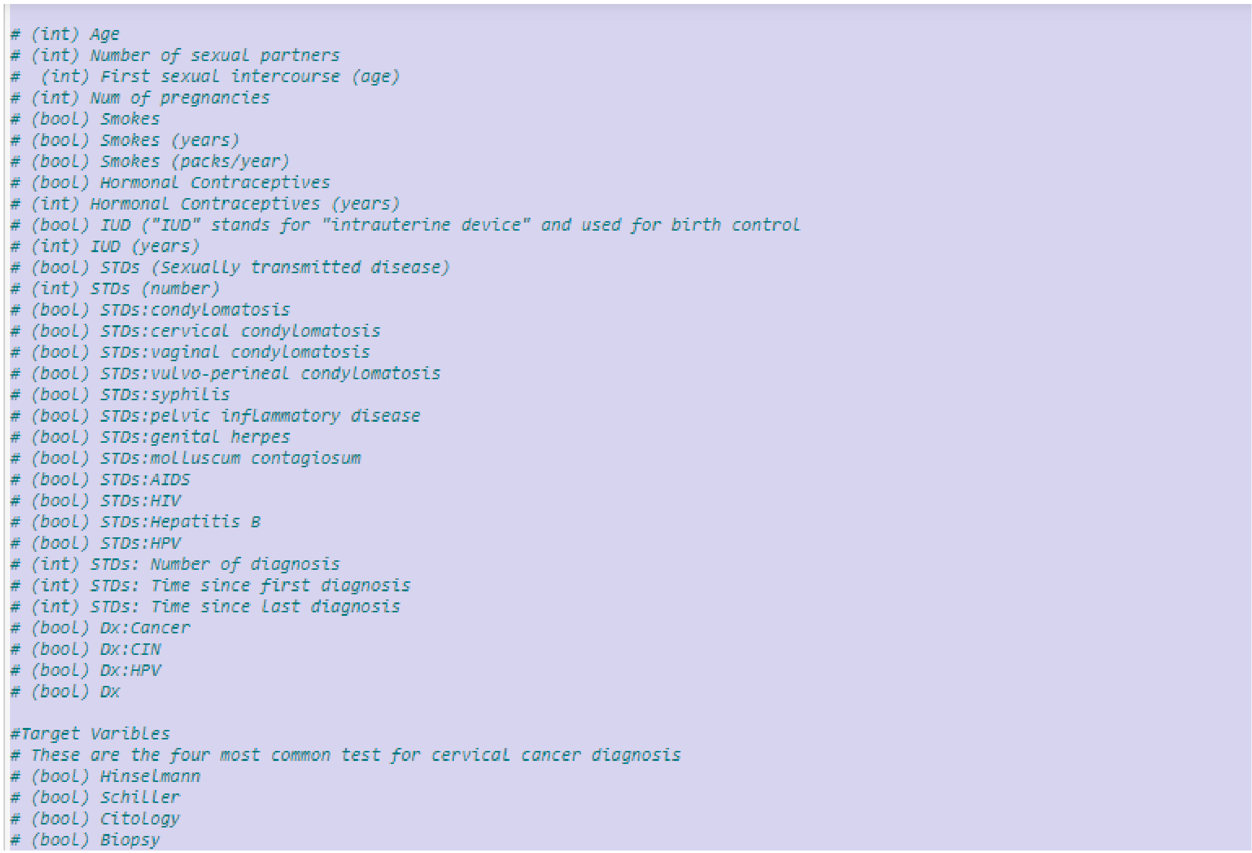


We then plotted the histogram



Data preparation and Model Training

Next we set column Biopsy as target variable and rest as input variables.



```
# (int) Age
# (int) Number of sexual partners
#  (int) First sexual intercourse (age)
# (int) Num of pregnancies
# (bool) Smokes
# (bool) Smokes (years)
# (bool) Smokes (packs/year)
# (bool) Hormonal Contraceptives
# (int) Hormonal Contraceptives (years)
# (bool) IUD ("IUD" stands for "intrauterine device" and used for birth control
# (int) IUD (years)
# (bool) STDs (Sexually transmitted disease)
# (int) STDs (number)
# (bool) STDs:condylomatosis
# (bool) STDs:cervical condylomatosis
# (bool) STDs:vaginal condylomatosis
# (bool) STDs:vulvo-perineal condylomatosis
# (bool) STDs:syphilis
# (bool) STDs:pelvic inflammatory disease
# (bool) STDs:genital herpes
# (bool) STDs:molluscum contagiosum
# (bool) STDs:AIDS
# (bool) STDs:HIV
# (bool) STDs:Hepatitis B
# (bool) STDs:HPV
# (int) STDs: Number of diagnosis
# (int) STDs: Time since first diagnosis
# (int) STDs: Time since last diagnosis
# (bool) Dx:Cancer
# (bool) Dx:CIN
# (bool) Dx:HPV
# (bool) Dx

#Target Varibles
# These are the four most common test for cervical cancer diagnosis
# (bool) Hinselmann
# (bool) Schiller
# (bool) Citology
# (bool) Biopsy
```

We then Normalisation of the data(scaling the data before feeding the model)
Next we split the data into test (20%) and train (80%) sets. We further split test data into validation and testing data

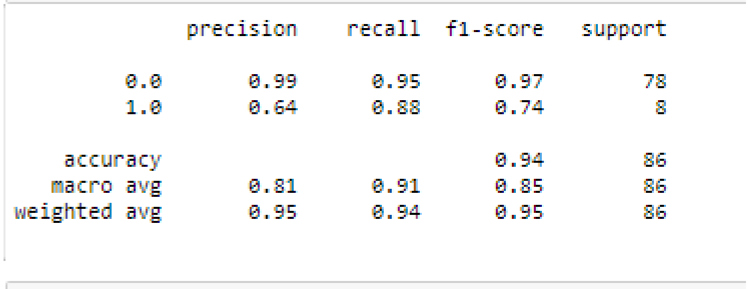
Study about XG-Boost Train

- XGBoost is a supervised machine learning algorithm
- It implements gradient boosted tree algorithm
- It makes better prediction by combines the predicts of the previous weak models
- It works by learning from the mistakes made in the previous models
- It works by training the model in a sequential manner
- It first makes a model based on training data and then train the second model based on the mistakes of the first.

Evaluate XG-Boost Algorithm

Step 1: Install XGBOOST
Step 2: Train an XGBoost classifier model
Step 3: Evaluate the models performance
we see that we have achieved 97% accuracy with our training data
Step 4: We predict the score of the trained model using the testing dataset

we see that we have achieved 94% accuracy with our testing data
Step 5: Next we predicted the score of the trained model using the testing dataset
Step 6: Next we print the classification report and confusion matrix

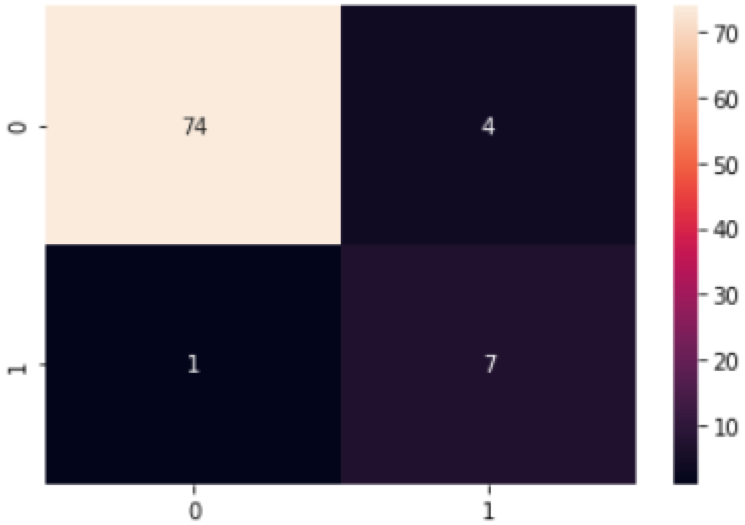


```
              precision    recall  f1-score   support

         0.0       0.99      0.95      0.97        78
         1.0       0.64      0.88      0.74         8

    accuracy                           0.94        86
   macro avg       0.81      0.91      0.85        86
weighted avg       0.95      0.94      0.95        86
```

we observe precision of 99% on class zero which is pretty good, however the precision and recall for class1 is not that good



The model corretly classify 74(top left) and 7(bottom right) samples and misclassify 4(top right) and 1(bottom left) samples as seen above in the heatmap

Reference:

1. https://www.coursera.org/learn/machine-learning-with-python
2. Chen, T., & Guestrin, C. (2016). XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16. doi:10.1145/2939672.2939785
3. https://www.youtube.com/watch?v=GrJP9FLV3FE