Cervical Cancer Risk Classification

Student: Fereshteh Feizabadi

Student ID: 274475

Professor: Francesco Vaccarino

Index

1. Introduction

In this project, we want to analyze and implement some of the main Machine Learning algorithms using the Python language. To perform this analysis, we will use the **Cervical Cancer (Risk Factors) Data Set**. The data set was obtained from UCI's Machine Learning Repository.
We will first explore the dataset and apply some preprocessing techniques such as handling missing values and removing unneeded columns, after that we will do exploratory data analysis to understand the correlation between features and target variable, in this step we realize the dataset is imbalanced.
In the next step, we do feature engineering, we visualize the numerical features with a boxplot for detecting outliers, next we explore the importance of features. Finally, we come to choose the Machine Learning Algorithms for our problem to build a predictive model for cervical cancer based on Cytology results and potential risk factors, including demographics and patient history, in this step, we initially focused on 4 supervised learning algorithms and for further improvement, we applied oversampling technique, and hyperparameter tuning. In the end, we finish the report with the conclusion.

2. Data Description

We report here the description of the dataset provided by the creators. The dataset was collected at 'Hospital Universitario de Caracas' in Caracas, Venezuela. This dataset focuses on the prediction of indicators/diagnosis of cervical cancer.

The features cover demographic information, habits, and historic medical records of 858 patients. Several patients decided not to answer some of the questions because of privacy concerns (missing values).

We report here information about the dataset:

- Number of instances: 858
- Number of attributes: 36
- Attribute Characteristics: Integer, Real
- Data Set Characteristics: Multivariate
- Missing attribute values: Yes (Marked as ?)
- The **Biopsy** serves as the gold standard for diagnosing cervical cancer.

The following are the description of independent and the dependent attributes:

- **Age:** It indicates the age of a woman. It is expressed in terms of numerical values
- **Number of sexual partners:** It indicates the total number of sexual partners encountered. It is expressed in terms of numerical values.
- **First sexual intercourse:** It indicates the age of a woman when she had her first sexual intercourse. It is expressed in terms of the count.
- **Number of pregnancies:** It indicates the total number of times the woman got pregnant. It is expressed in terms of the total count.
- **Smokes:** It indicates whether the person smokes or not. It is expressed in terms of zeros (does not smoke) and ones(smokes).
- **Smokes (years):** It indicates the total number of years for which the woman is smoking. It is expressed in terms of total count.
- **Smokes (packs/year):** It indicates the total number of packets of cigarettes per year the woman smokes. It is expressed in terms of numbers
- **Hormonal Contraceptives:** It indicates whether the patient uses hormonal contraceptives or not.
- **Hormonal Contraceptives (years):** It indicates that for how many years the contraceptive method was used. It was expressed in terms of total number of years.
- **Intra-Uterine Device:** It indicated where the intrauterine contraceptive device was used or not. It was expressed in terms of zeros(did not use IUD) and ones(used IUD).
- **IUD (years):** It indicated that for how many years the IUD was used. It is expressed in terms of the total number of years.
- **STDs:** It indicates the presence of **Sexually Transmitted Diseases**. It is expressed in terms of zeroes and ones.
- **STDs (number):** It indicates the total number of sexually transmitted diseases present with the patient. It is expressed in terms of numbers.
- **STDs:condylomatosis:** It indicates the presence of Condylomatosis with the patient.

- **STDs:cervical condylomatosis**: It indicates the presence of Cervical condylomatosis.
- **STDs:vaginal condylomatosis:** It indicates the presence of Vaginal condylomatosis.
- **STDs:vulvo-perineal condylomatosis**: It indicates the presence of Vulvo- Perineal condylomatosis.
- **STDs:syphilis**: It indicates the presence of Syphilis.
- **STDs:pelvic inflammatory disease**: It indicates the presence of pelvic inflammatory disease.
- **STDs:genital herpes**: It indicates the presence of Genital Herpes.
- **STDs:molluscum contagiosum**: It indicates the presence of Molluscum Contagiosum.
- **STDs:AIDS**: It indicates the presence of AIDS in the patient.
- **STDs:HIV**: It indicates the presence of HIV in the patient.
- **STDs:Hepatitis B:** It indicates the presence of Hepatitis B in the patients.
- **STDs:HPV**: It indicates the presence of HPV in the patients.
- **STDs: Number of diagnosis**: It indicates the total number of times the STDs have been diagnosed.
- **STDs: Time since first diagnosis**: It indicates the total number of years since the first diagnosis.
- **STDs: Time since last diagnosis**: It indicates the total number of years elapsed since the last diagnosis.
- **Dx:Cancer**: It indicates the person had a previous cervical cancer diagnostic.
- **Dx:CIN**: It indicates the person had a previous diagnostic of Cervical intraepithelial neoplasia.
- **Dx:HPV**: It indicates the presence of Human papillomaviruses.
- **Dx**: It indicates the presence of any one among cancer, CIN and HPV.
- **Hinselmann**: also known as colposcopy, is a medical diagnostic procedure to examine an illuminated, magnified view of the cervix as well as the vagina and vulva.

- **Schiller**: Schiller Iodine test is a medical test in which iodine solution is applied to the cervix in order to diagnose cervical cancer.
- **Cytology**: also called as Pap smears test, helps detect abnormal cells in the cervix, which can develop into cancer.
- **Biopsy (Target Variable)**: A cervical biopsy is a surgical procedure in which a small amount of tissue is removed from the cervix. A cervical biopsy is usually done after an abnormality has been found during cytology.

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives	Hormonal Contraceptives (years)	IUD	IUD (years)
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	0.0
4	46	3.0	21.0	4.0	0.0	0.0	0.0	1.0	15.0	0.0	0.0
5	42	3.0	23.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	51	3.0	17.0	6.0	1.0	34.0	3.4	0.0	0.0	1.0	7.0
7	26	1.0	26.0	3.0	0.0	0.0	0.0	1.0	2.0	1.0	7.0
8	45	1.0	20.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	44	3.0	15.0	?	1.0	1.266972909	2.8	0.0	0.0	?	?

Figure 1: Dataframe head

3. Preprocessing

Missing Values

The dataset contains missing values that show with question marks (?). The total missing values for each column are according below. For handling the missing values, first, we replace the question marks with Nan. As we see in the below heatmap the missing values are the yellow color.

```
Number of missing values:
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
Dx:Cancer                             0
Dx:CIN                                0
Dx:HPV                                0
Dx                                    0
Hinselmann                            0
Schiller                              0
Citology                              0
Biopsy                                0
```

Figure 2: Count of Missing Values

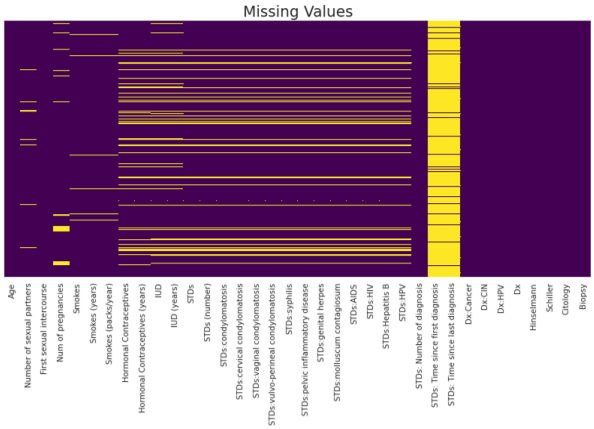


Figure 3: Missing values show in yellow colour

Dropping Unneeded Features

Two features: **STDs: Time since first diagnosis**, **STDs: Time since last diagnosis** contain mostly missing values, and these features will be dropped.
Moreover, data collected on hormonal contraceptives, STDs, and IUDs have a lot of missing values.

Features Types

We have two different data types in our dataset, **Numerical** and **Categorical**. The numerical and categorical features are separated for handling their missing values.

```
[ ]  num_cols= ['Age', 'Number of sexual partners', 'First sexual intercourse','Num of pregnancies', 'Smokes (years)',
                 'Smokes (packs/year)','Hormonal Contraceptives (years)','IUD (years)','STDs (number)']


     cat_cols=['Smokes','Hormonal Contraceptives','IUD','STDs','STDs:condylomatosis','STDs:cervical condylomatosis',
               'STDs:vaginal condylomatosis','STDs:vulvo-perineal condylomatosis', 'STDs:syphilis',
               'STDs:pelvic inflammatory disease', 'STDs:genital herpes','STDs:molluscum contagiosum', 'STDs:AIDS',
               'STDs:HIV','STDs:Hepatitis B', 'STDs:HPV', 'STDs: Number of diagnosis','Dx:Cancer', 'Dx:CIN',
               'Dx:HPV', 'Dx', 'Hinselmann', 'Schiller','Citology', 'Biopsy']
```

Figure 4: Categorical and Numerical Features

Handling Missing Values

As we see most of the binary features contain mostly values of 0. Some of the features appear to not contain any 1's. Features with 2 or fewer 1's will be dropped.

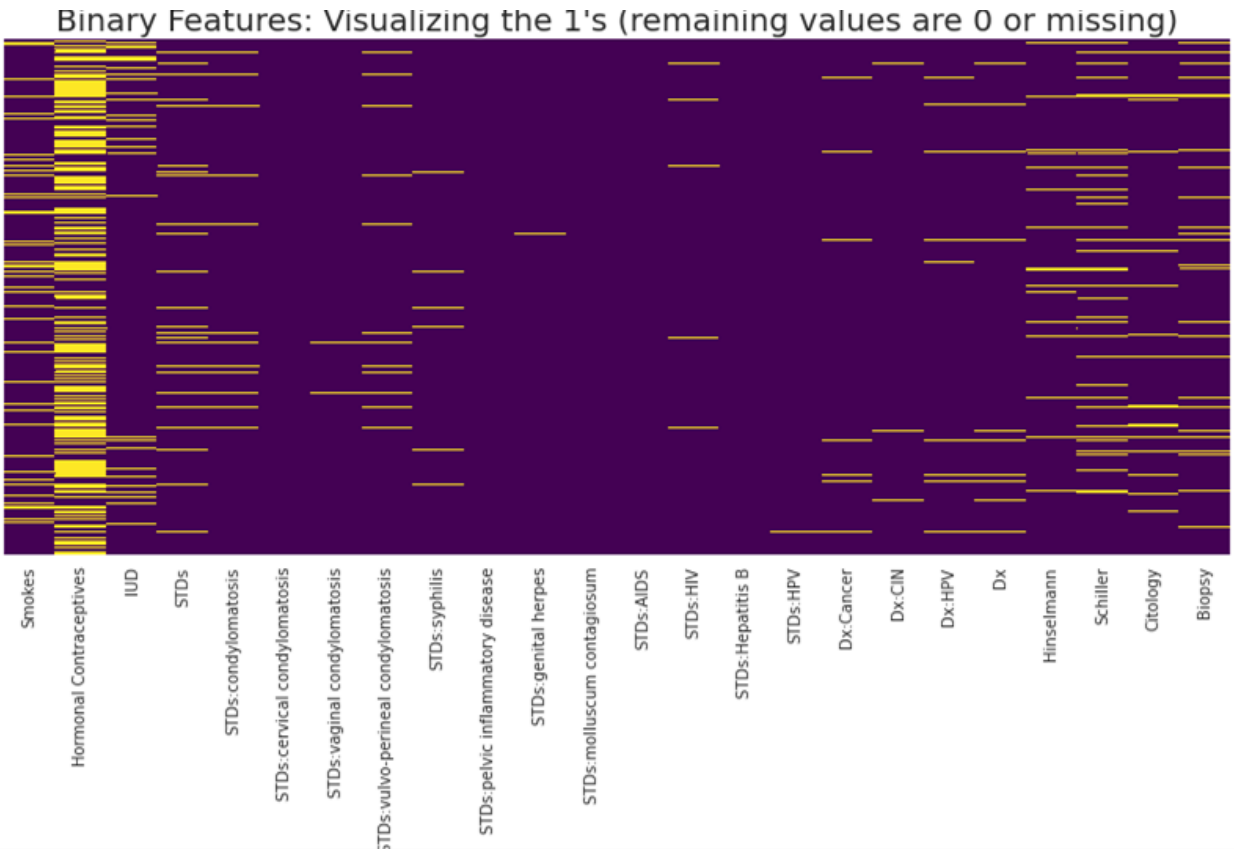


Figure 5: Heatmap for visualizing the 1's of Binary Features

The following Categorical Features are dropped because they contain 2 or fewer 1's values.

- STDs:cervical condylomatosis
- STDs:pelvic inflammatory disease
- STDs:genital herpes
- STDs:molluscum contagiosum
- STDs:AIDS
- STDs:Hepatitis B
- STDs:HPV

The remaining categorical features that still have missing values are shown in the below plot.

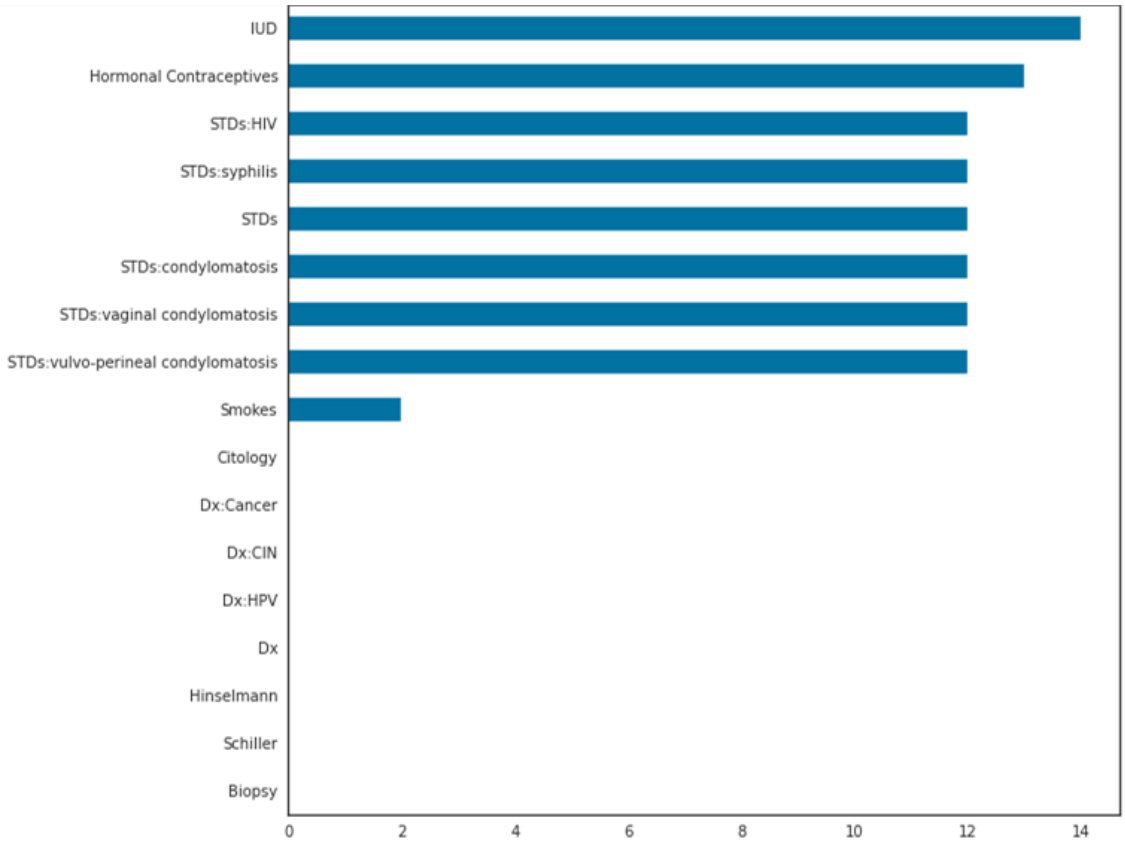


Figure 6: Missing values of Categorical Features

Fancyimput

fancyimpute is a library for missing data imputation algorithms. It uses machine learning algorithms to impute missing values. Fancyimpute uses all the columns to impute the missing values. Here we used the **KNN Imputation**.

To fill out the missing values KNN finds out the similar data points among all the features. Then it took the average of all the points to fill in the missing values.

The remaining columns after handling the missing values and dropping unneeded columns, shown in the below.

```
Number of missing values:
Age                                 0
Number of sexual partners           0
First sexual intercourse            0
Num of pregnancies                  0
Hormonal Contraceptives (years)     0
IUD (years)                         0
STDs (number)                       0
STDs: Number of diagnosis           0
Smokes (years)                      0
Smokes (packs/year)                 0
Smokes                              0
Hormonal Contraceptives             0
IUD                                 0
STDs                                0
STDs:condylomatosis                 0
STDs:vaginal condylomatosis         0
STDs:vulvo-perineal condylomatosis   0
STDs:syphilis                       0
STDs:HIV                            0
Dx:Cancer                           0
Dx:CIN                              0
Dx:HPV                              0
Dx                                  0
Hinselmann                          0
Schiller                            0
Citology                            0
Biopsy                              0
dtype: int64
```

Figure 7: Showing zero Missing Values

4. Exploratory Data Analysis (EDA)

Overview of Target

As we can see we have a total of 858 observations, divided into 803 individuals in Negative and 55 individuals are Positive Biopsy results.

There are many more individuals in the negative biopsy group, and this imbalance in categories presents some challenges for predictive modeling. On classification problems, we need to know how balanced the class values are. Since there is an imbalance in data, which needs to be taken care of in the model building section.

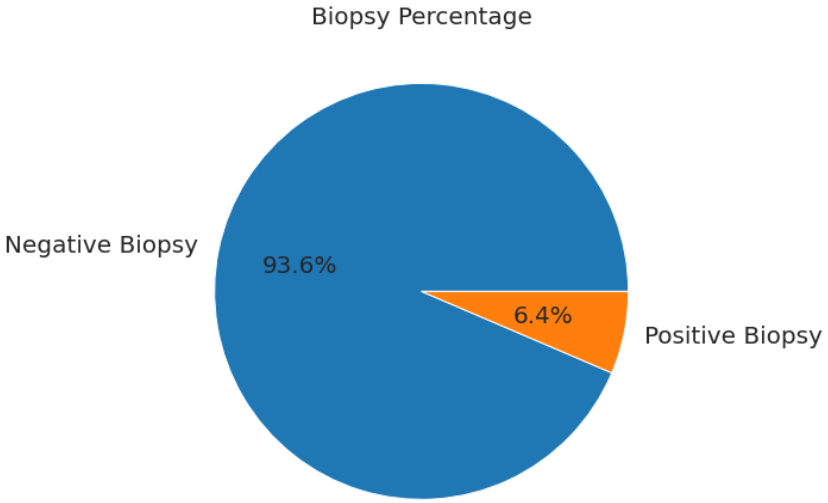


Figure 8: Percentage of Positive and Negative Biopsy Result

Uni Variate Analysis

Count plots of Categorical Columns

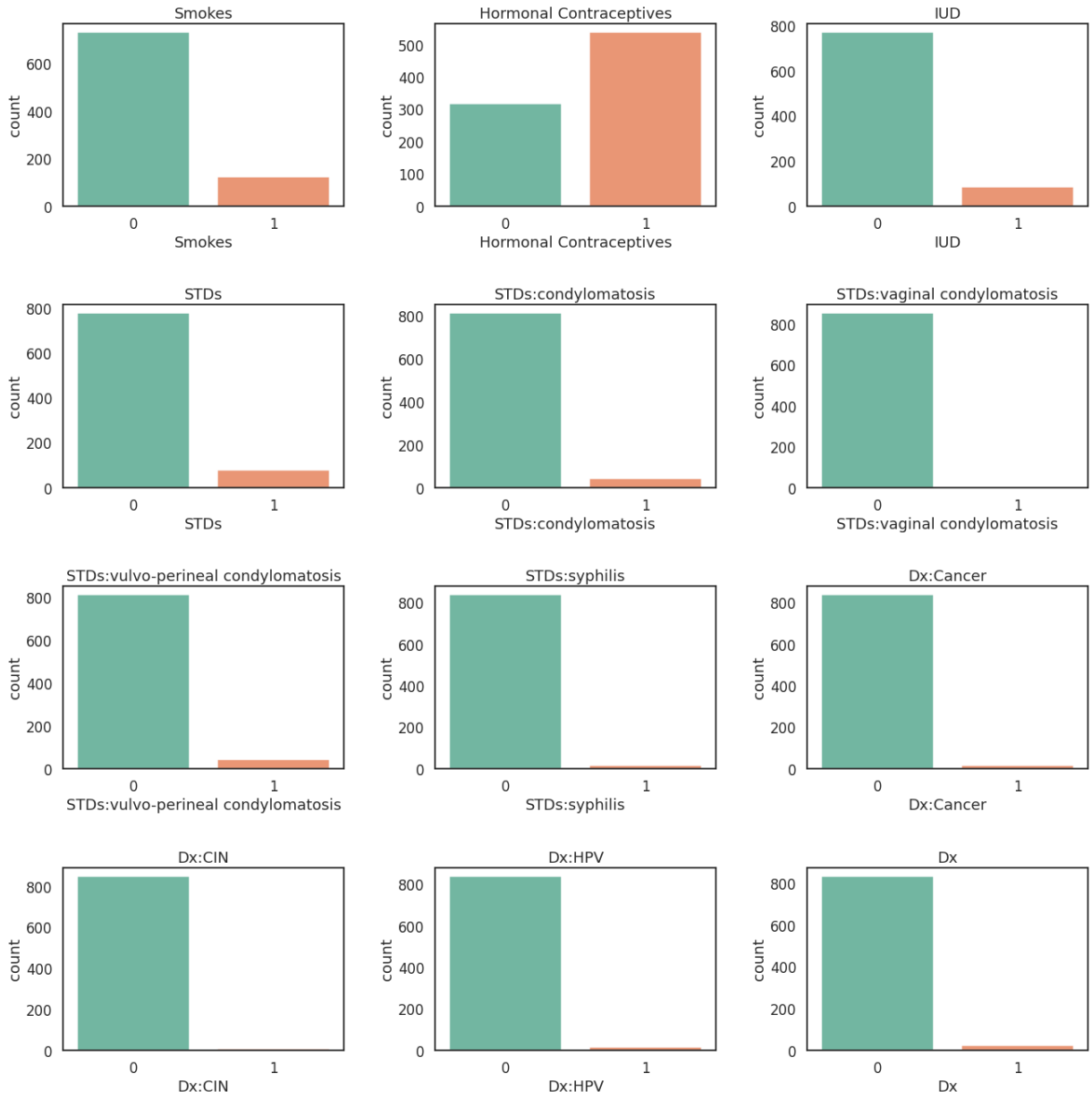


Figure 9: Count plots of Categorical Columns

Density plots of Numerical Columns

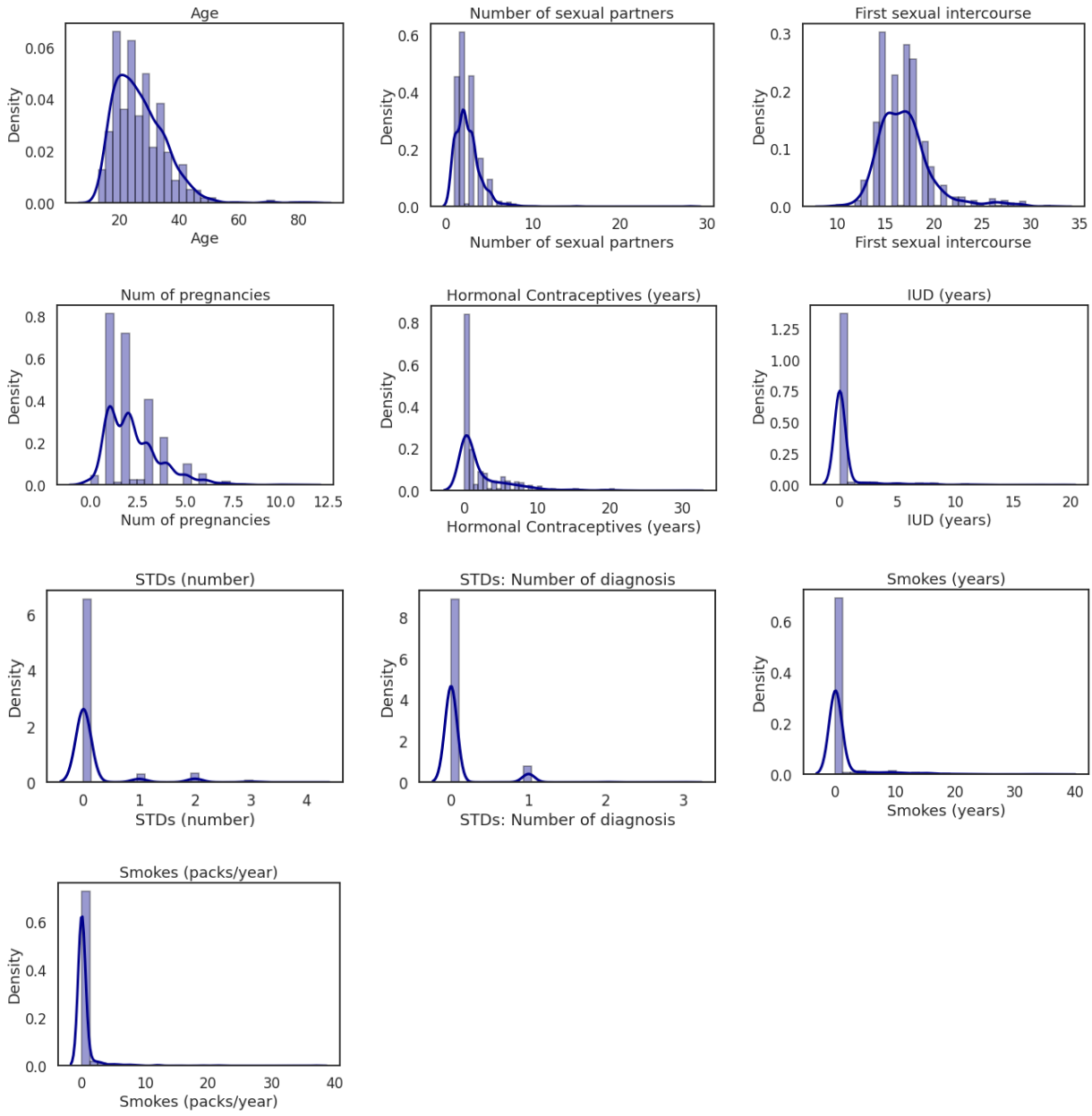


Figure 10: Density plot of Numerical Columns

Multivariate Analysis

Age and Sexual Habits vs Biopsy

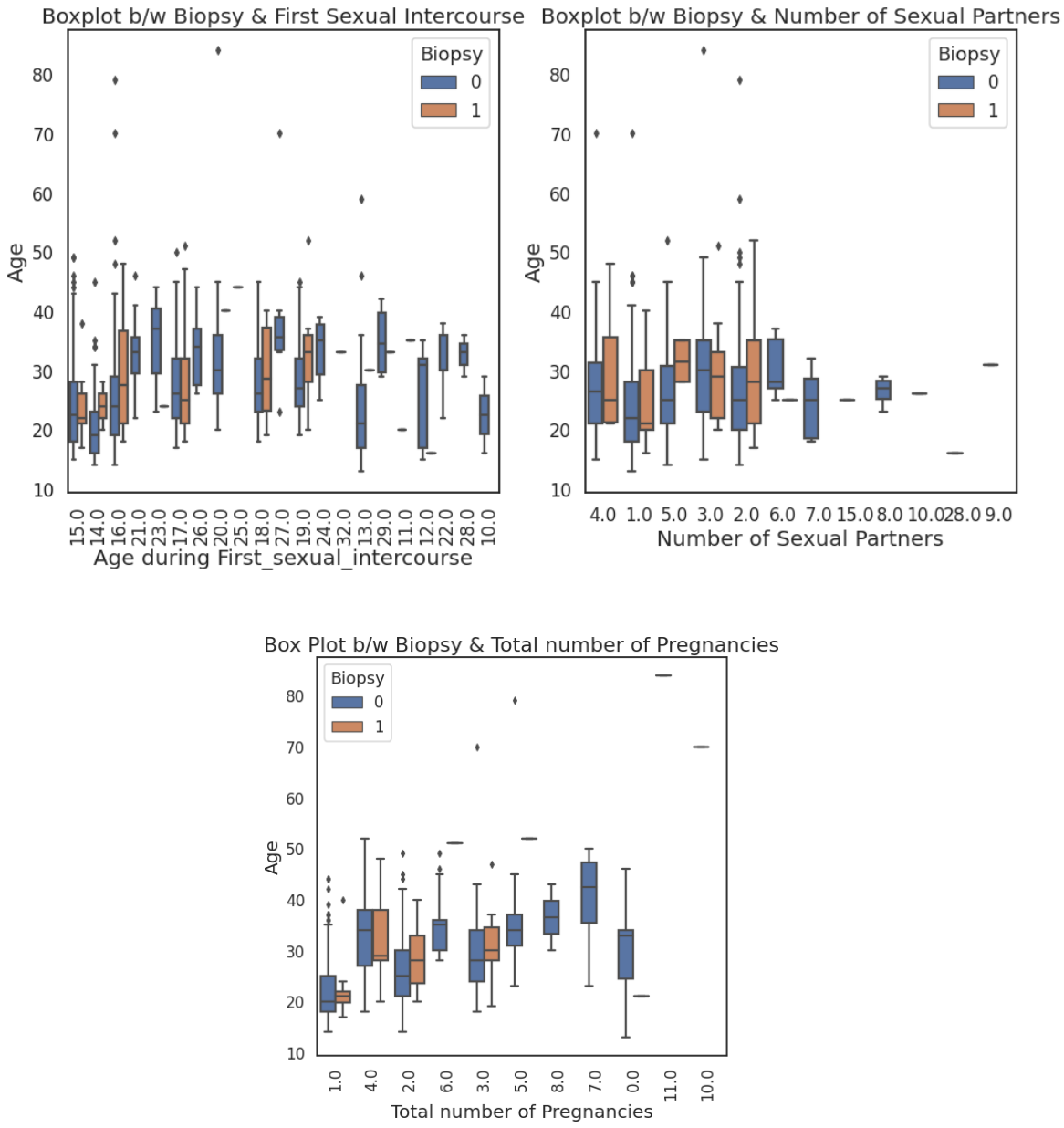


Figure 11: Age and Sexual Habits vs Biopsy

Smoke and Sexual Habits vs Biopsy

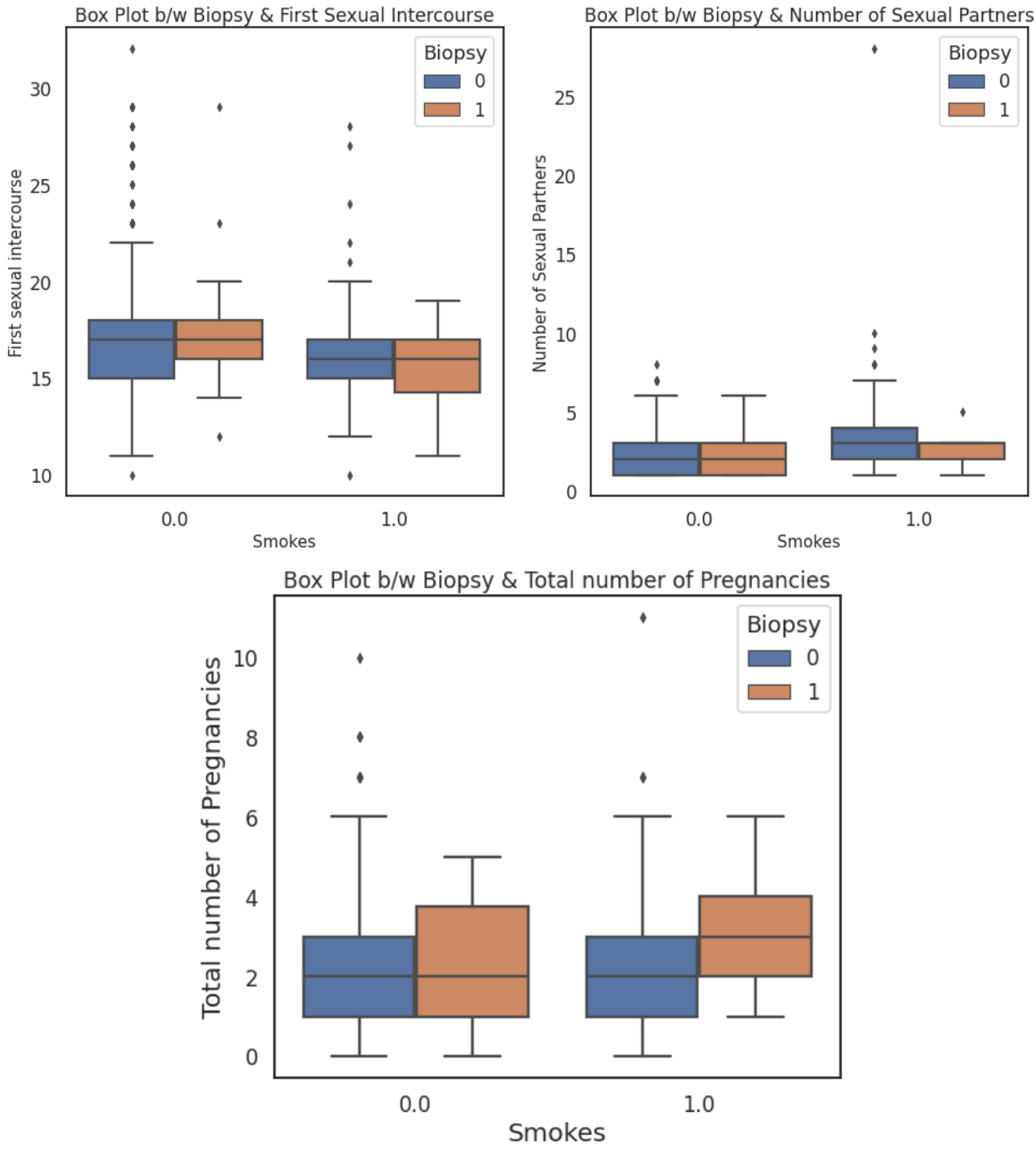


Figure 12: Smoke and Sexual Habits vs Biopsy

we plot the most correlated features with the labels with violin plots. **Violin plots** associate the two different data distributions given from the two different labels. They are similar to box plots but they show the probability density.

Age & Smokes vs Biopsy

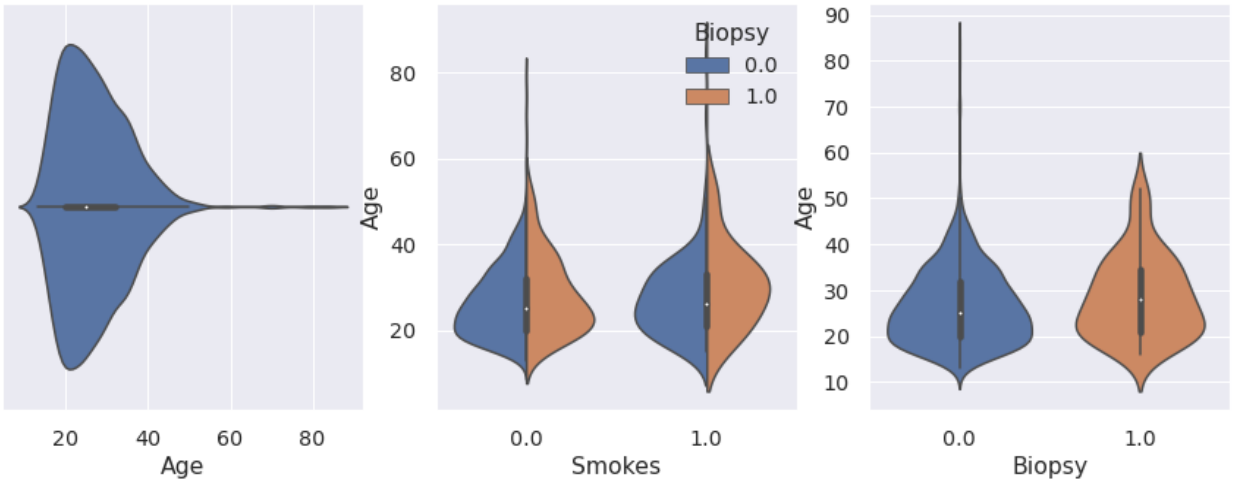


Figure 13: Age & Smokes vs Biopsy

Age & Number of sexual partners vs Biopsy

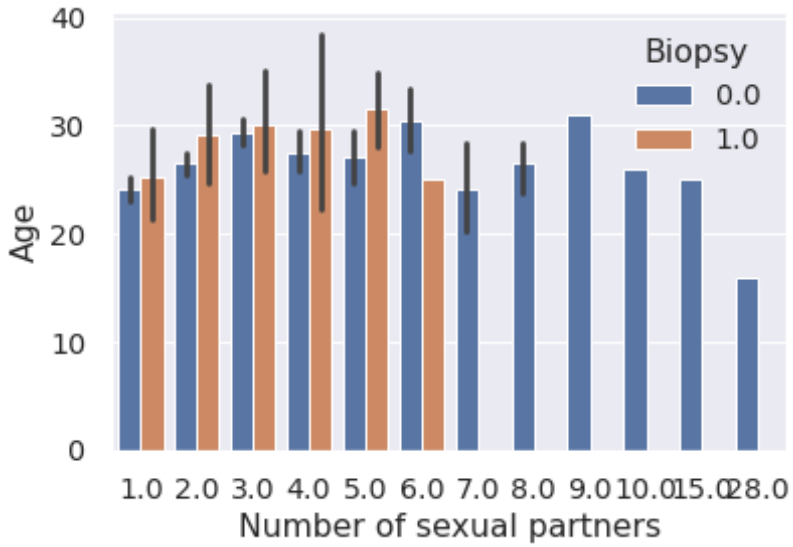


Figure 14: Age & Number of sexual partners vs Biopsy

Age & First sexual intercourse vs Biopsy

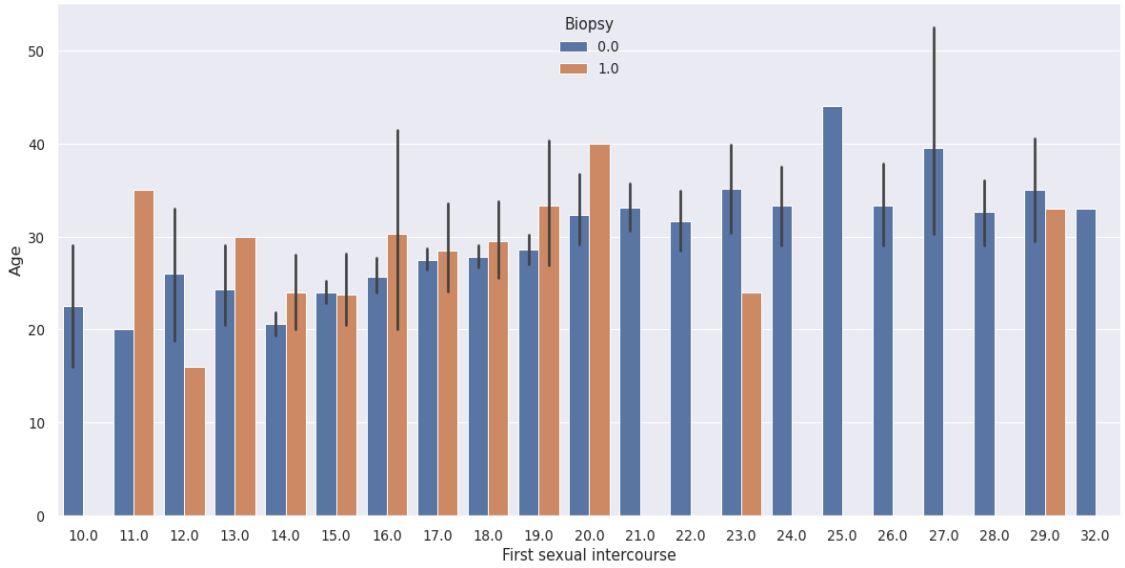


Figure 15: Age & First sexual intercourse vs Biopsy

Age & Smokes(years)

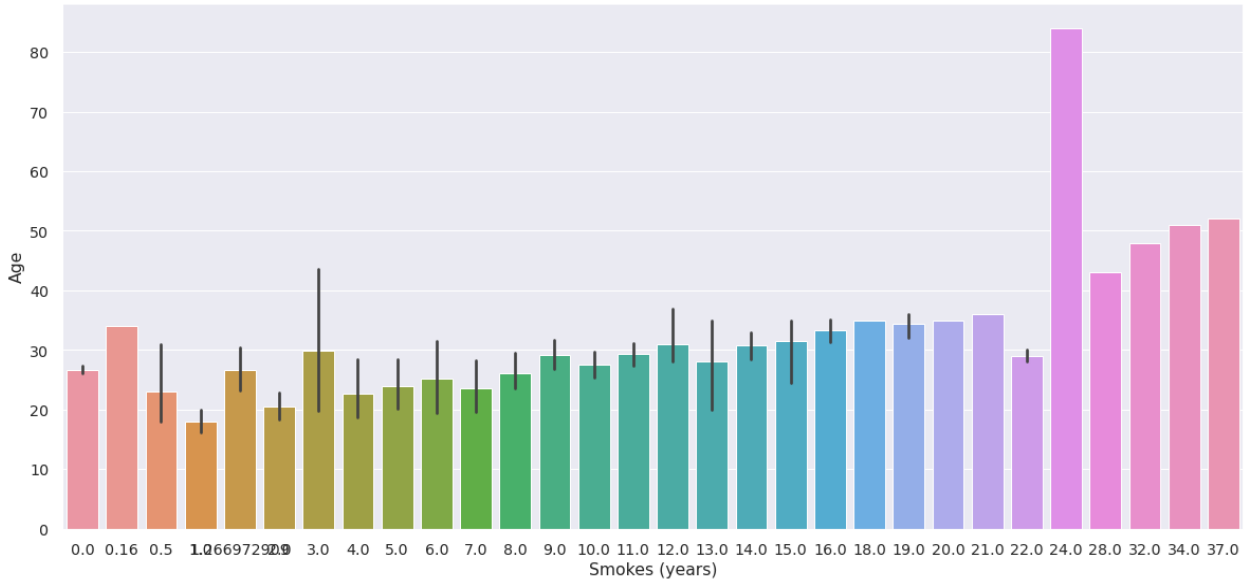


Figure 16: Age & Smokes(years)

Smokes(years) & Number of sexual partners vs Biopsy

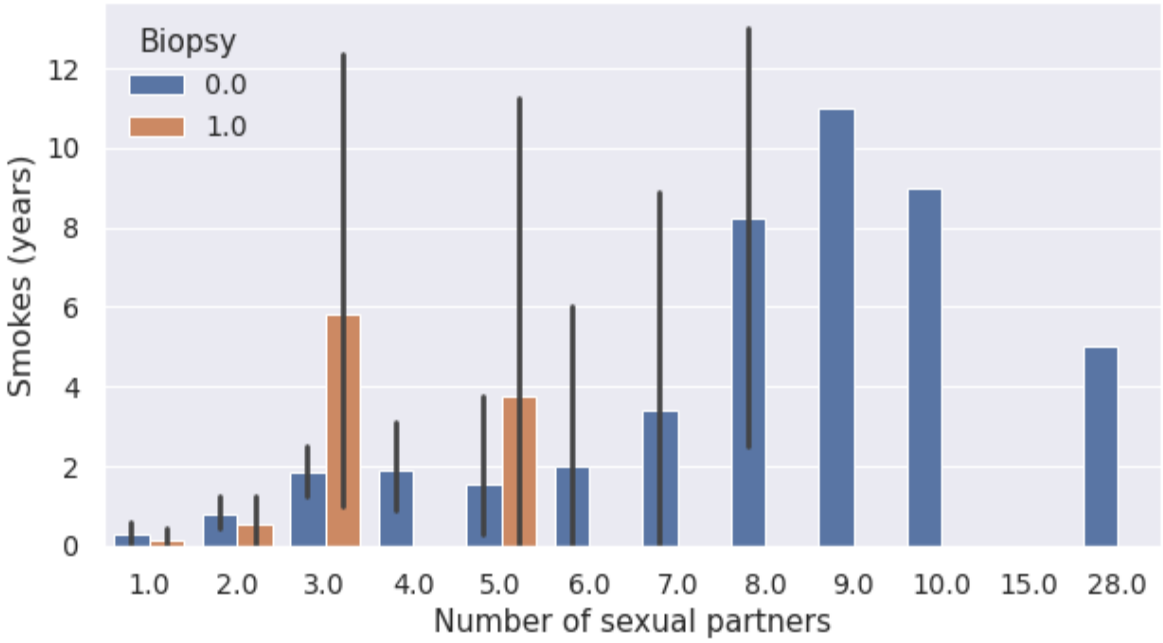


Figure 17: Smokes(years) & Number of sexual partners vs Biopsy

Age & Num of pregnancies vs Biopsy

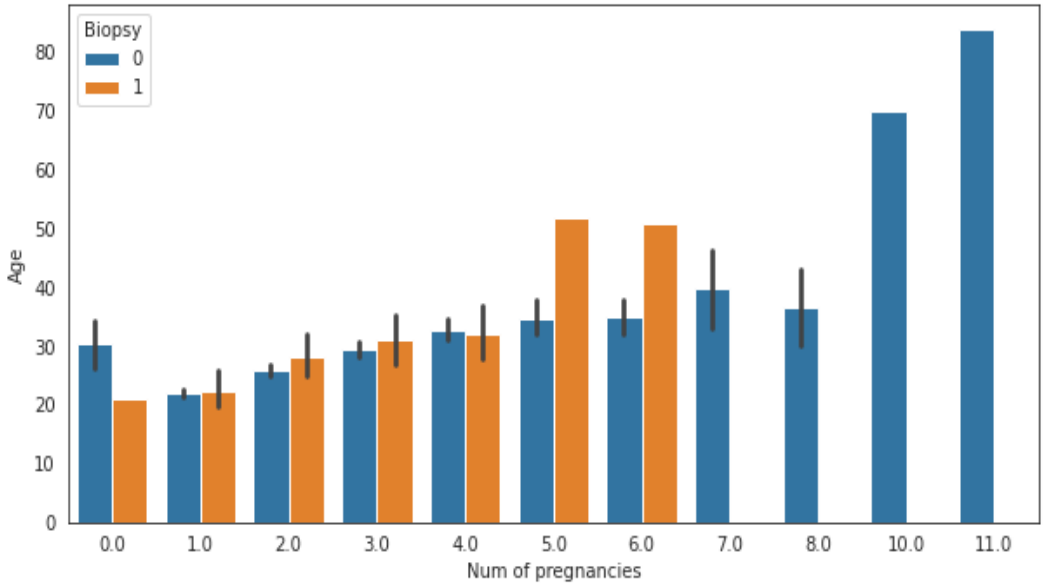


Figure 18: Age & Num of pregnancies vs Biopsy

Number of sexual partners vs Biopsy

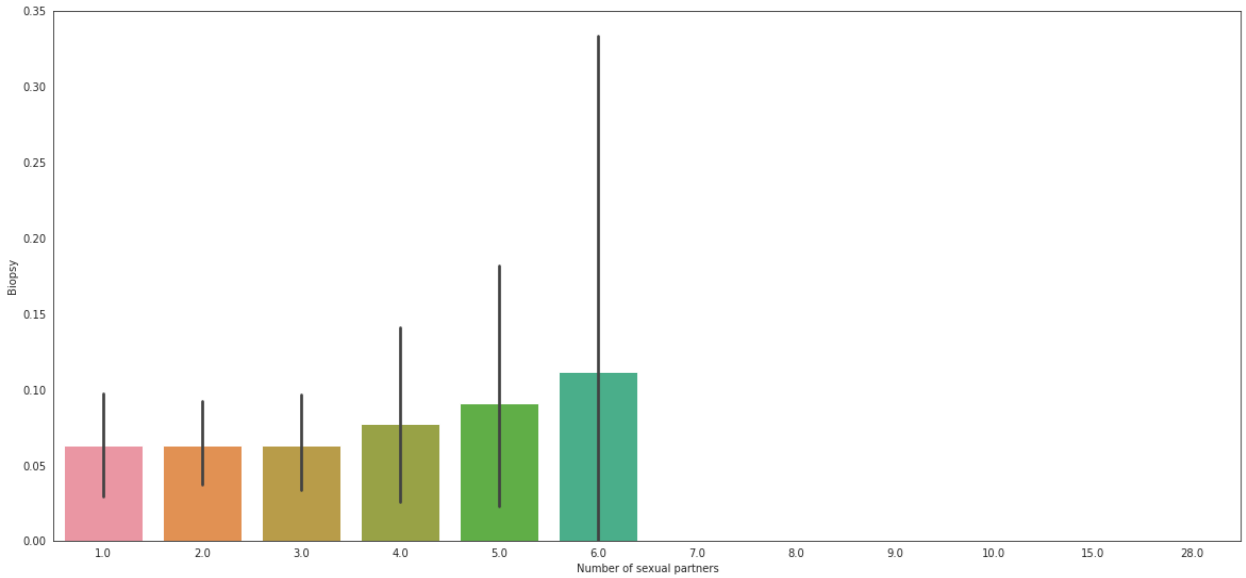


Figure 19: Number of sexual partners vs Biopsy

Conclusion:

- Most of the patients are in the age group 20-40
- Predominant of the patients had 0-5 sexual partners
- Most of the patients their first sexual intercourse between 15-20 years
- The larger group of patients had 1-3 pregnancies overall in their life
- Relatively larger proportion of the patients are non-smokers (around 700) and only a very few (around 100) are smokers.
- Most of the patients have used Hormonal contraceptives methods like pills and medications for birth controls whereas only a few of them have opted for intrauterine devices (IUDs).
- Only a very few people are affected by any one of the STDs.

Hormonal Contraceptives (years) and Age vs Biopsy Result

We want to check the relationship between **Hormonal Contraceptives (years)** and **Age** features with **Biopsy results**.

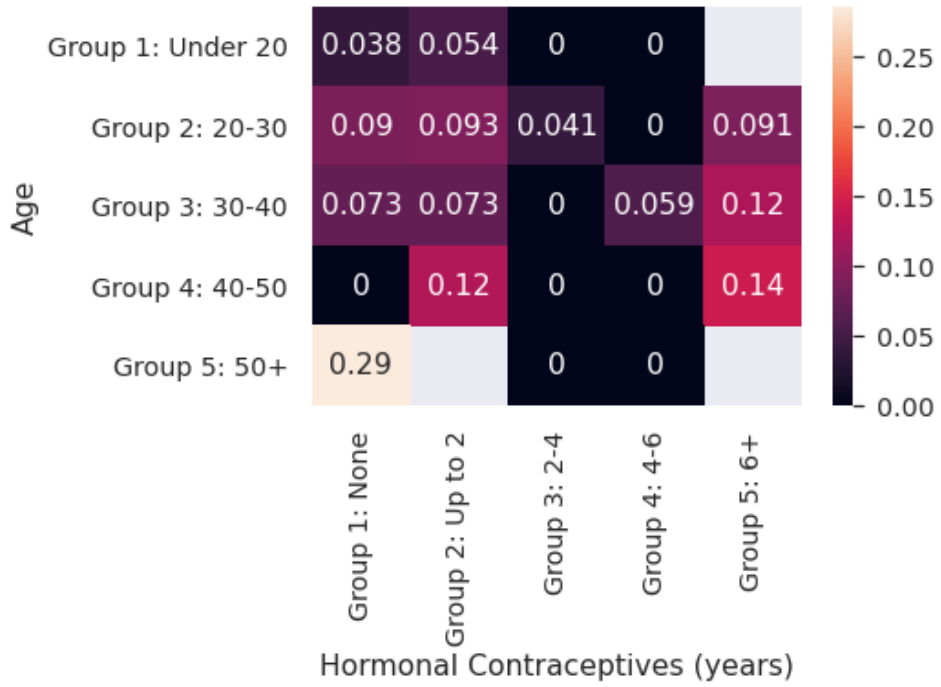


Figure 20: Heatmap Showing Percent with Positive Biopsy Results by Age & Hormonal Contraceptive Grouping

The matrix plot shows that individuals who took hormonal contraceptives for 6 or more years and who were in the 30 to 50 age groups had some of the highest percentages of positive biopsy results. Although the percentage of people with positive biopsy results in the 50+ age group who did not take hormonal contraceptives was high, note that there were only 7 individuals in this grouping (see counts below).

Counts within each grouping for age and hormonal contraceptives

Age	Hormonal Contraceptives (years)	Count
Group 1: Under 20	Group 1: None	78
	Group 2: Up to 2	56
	Group 3: 2-4	42
	Group 4: 4-6	3
Group 2: 20-30	Group 1: None	111
	Group 2: Up to 2	108
	Group 3: 2-4	98
	Group 4: 4-6	33
	Group 5: 6+	44
Group 3: 30-40	Group 1: None	55
	Group 2: Up to 2	55
	Group 3: 2-4	44
	Group 4: 4-6	17
	Group 5: 6+	49
Group 4: 40-50	Group 1: None	18
	Group 2: Up to 2	8
	Group 3: 2-4	6
	Group 4: 4-6	3
	Group 5: 6+	21
Group 5: 50+	Group 1: None	7
	Group 3: 2-4	1
	Group 4: 4-6	1

Figure 21: Count of Group Age within each Group of years for Hormonal using

Age vs Biopsy Result

Median age was slightly higher for those with positive test results.

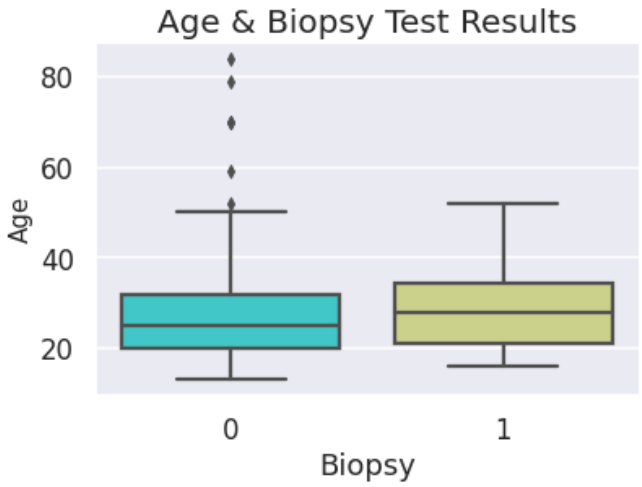


Figure 22: Age and Biopsy Result

Binary Features

The following table shows:

- % individuals with feature and cancer divided by the total number of individuals with cancer
- % individuals with feature divided by the total number of individuals

	Feature	% w/ Cancer who have Feature	% of All Indiv. w/ Feature	Number with Feature
0	Schiller	87.27	8.62	74
1	Hormonal Contraceptives	65.45	66.55	571
2	Hinselmann	45.45	4.08	35
3	Citology	32.73	5.13	44
4	STDs	21.82	9.21	79
5	Smokes	18.18	14.34	123
6	IUD	16.36	10.02	86
7	STDs:condylomatosis	12.73	5.13	44
8	STDs:vulvo-perineal condylomatosis	12.73	5.01	43
9	Dx	12.73	2.80	24
10	Dx:Cancer	10.91	2.10	18
11	Dx:HPV	10.91	2.10	18
12	STDs:HIV	9.09	2.10	18
13	Dx:CIN	5.45	1.05	9
14	STDs:vaginal condylomatosis	0.00	0.47	4
15	STDs:syphilis	0.00	2.10	18

Figure 23: Count of Binary Features

Conclusion:

- In the list above, large gaps between the two percentages shown may suggest that a given feature is related to the outcome of the biopsy test results.
- The **Schiller test** and **Hormonal contraceptives** showed the strongest relationship with biopsy results; 87**%** of the individuals with positive biopsy results also had positive results on the Schiller test.
- The **Hinselmann test** and **cytology** showed a weaker relationship with biopsy results, 45% of those who had positive biopsy results also had positive Hinselmann results, and 33% of those with positive biopsy results also had positive cytology results.
- The list above also suggests that **STDs may be related to biopsy results**. Approximately 22% of individuals with positive biopsy results also had an STD in the past, whereas only 9% of the total people surveyed had an STD in the past.

Correlation Matrix with Heatmap

Correlation states how the features are related to each other or the target variable. Correlation can be positive or negative. Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the **seaborn library**.

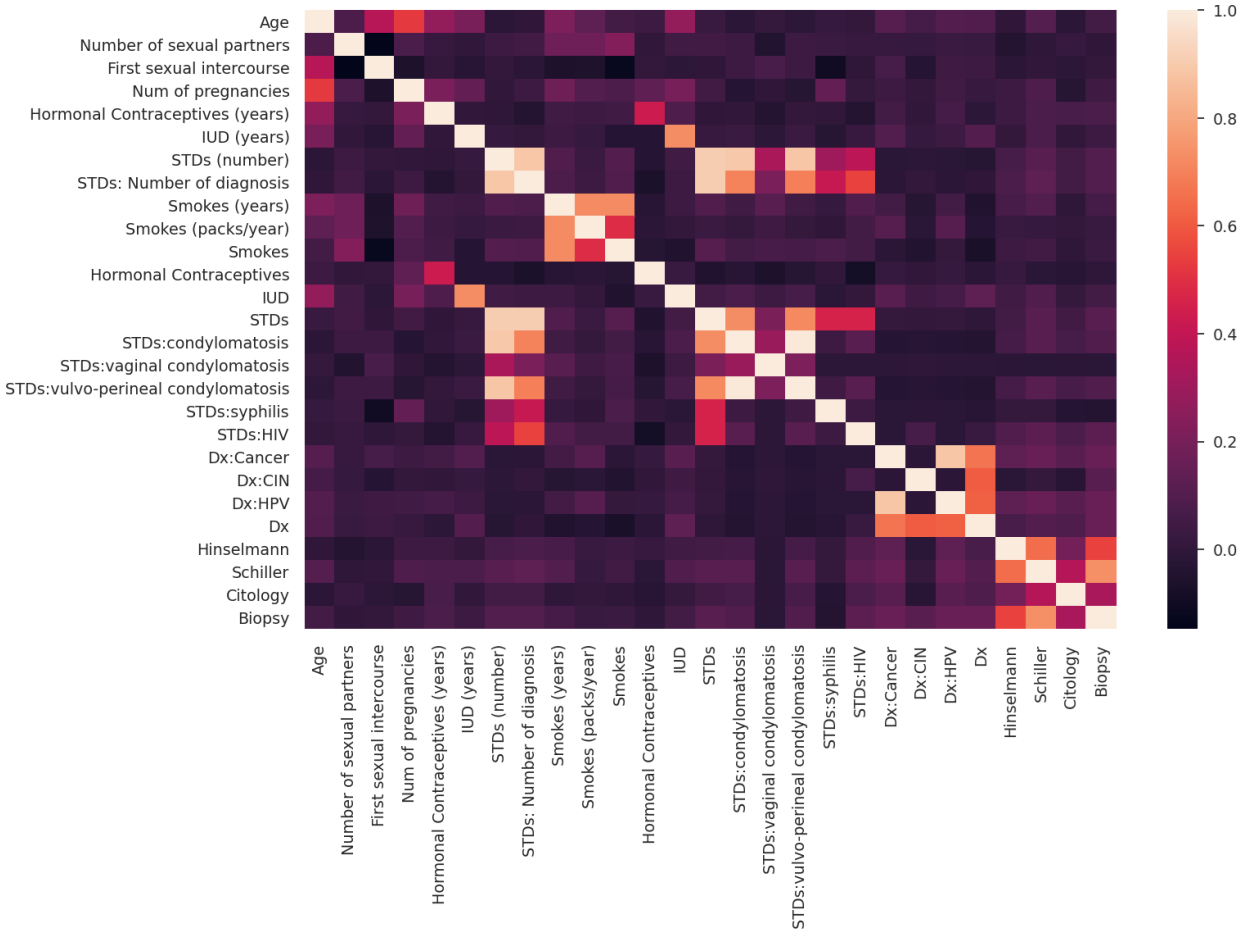


Figure 24: Correlation Matrix with Heatmap

5. Feature Engineering

To improve the performance of the model, prepare the proper input dataset and be compatible with the machine learning algorithm requirements used in feature engineering in this study. There are many techniques that are used for feature engineering. In this study used handling outliers.

Outliers

We also visualize all the features with box plots. A **box plot** is a graphical representation to visualize quartiles and outliers of a data distribution. Outliers are depicted as single points. The below graph implies that the data contains outliers.

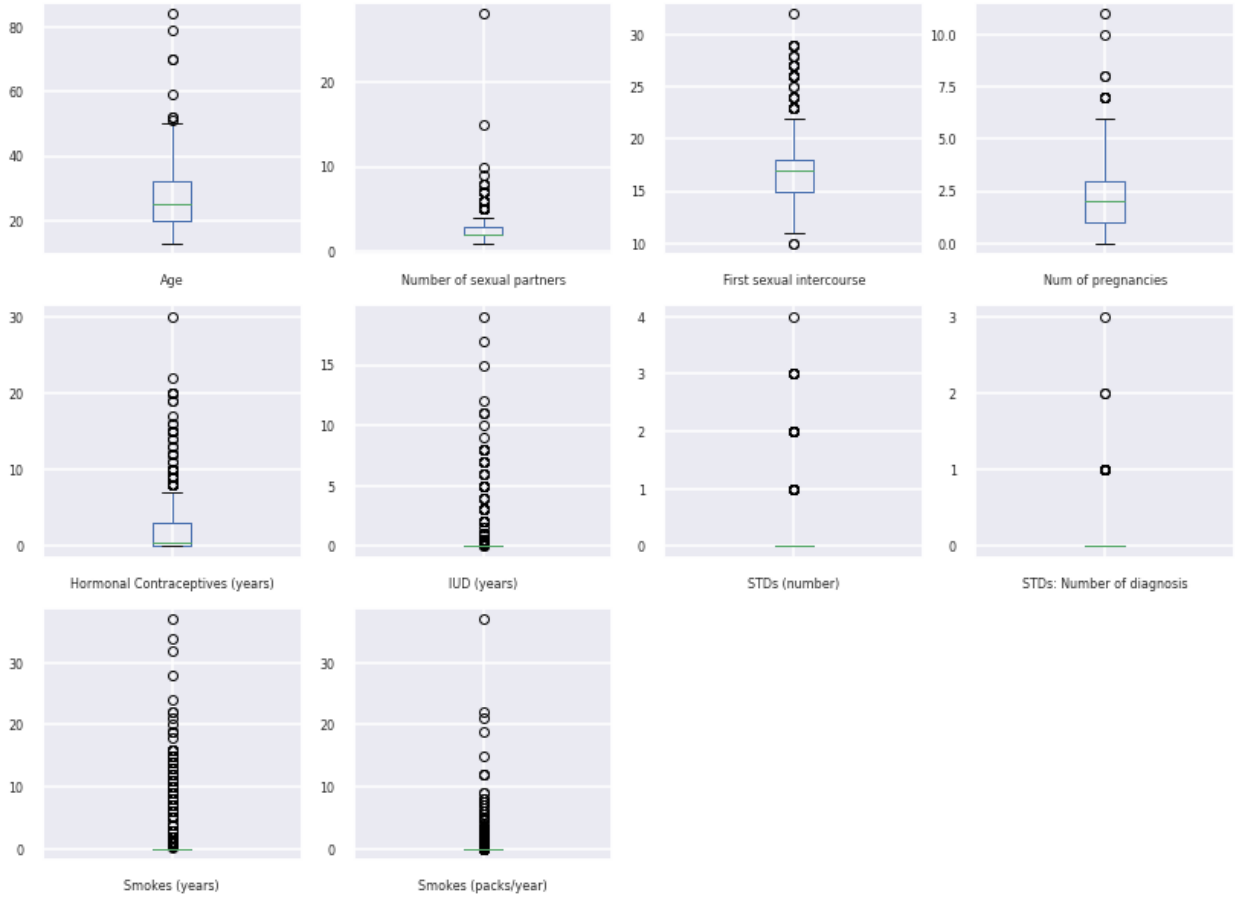


Figure 25: Outlier detection with Box Plot

Here using **IQR technique**, we have capped the extreme values above the upper whisker value to the value of upper whisker and similarly capped the extreme lower values to the value of lower whisker value.

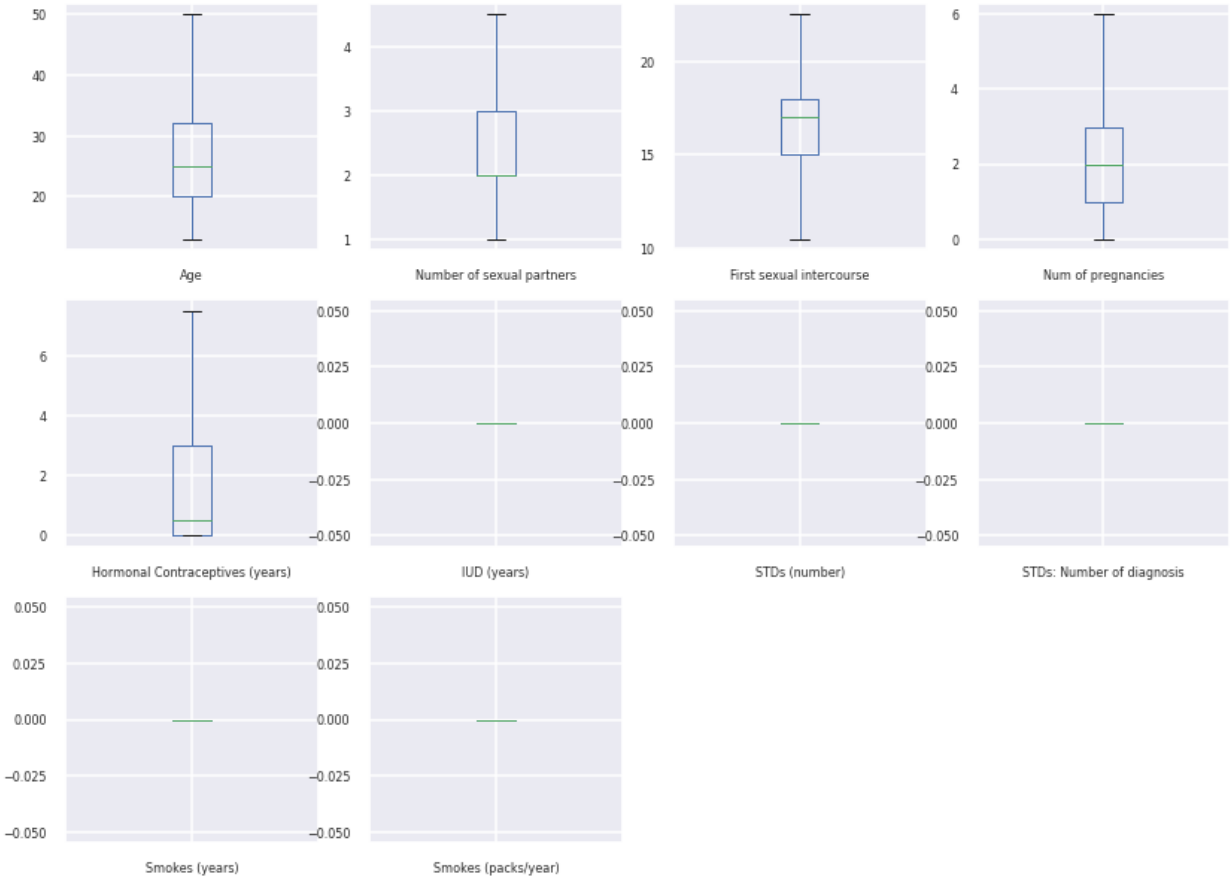


Figure 26: Outliers are removed

Now the outliers have been removed/capped. While building models, though outliers removal may have positive impact like getting higher accuracy and other metrics. Since it's an medical dataset, it's not recommended to just cap or remove outliers. for example: there are some females who are aged 70+ which comes out as extreme values, generally we should not be capping them to the upper whisker value (around 50) as it would alter the information provided by the data. Hence we are here building models with the original values.

6. Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model.

It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

Feature selection is a technique where we choose those features in our data that contribute most to the target variable. In other words we choose the best predictors for the target variable.

The classes in the **sklearn.feature_selection** module can be used for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets.

Benefits of performing feature selection before modeling:

- **Reduces Overfitting:** Less redundant data means less possibility of making decisions based on redundant data/noise.
- **Improves Accuracy:** Less misleading data means modeling accuracy improves.
- **Reduces Training Time:** Less data means that algorithms train faster.

Univariate Selection

Statistical tests can be used to select those features that have the strongest relationship with the output variable. The scikit-learn library provides the **SelectKBest** class that can be used with a suite of different statistical tests to select a specific number of features. We used the **chi-squared (chi²) statistical test** for non-negative features to select 5 of the best features from our dataset.

	Feature	Score
4	Hormonal Contraceptives (years)	28.977849
0	Age	7.224831
3	Num of pregnancies	1.410119
2	First sexual intercourse	0.020709
1	Number of sexual partners	0.001730

Figure 27: 5 of Best Features

Feature Importance

You can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable. Feature importance is an inbuilt class that comes with Tree Based Classifiers, we will be using **Extra Tree Classifier** for extracting the top 10 features for the dataset.

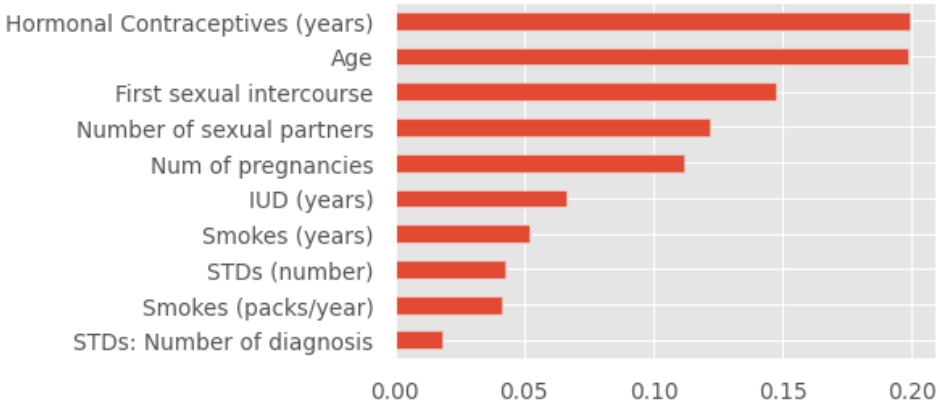


Figure 28: Top 10 Important Features

7. Overview on Machine Learning Algorithms

Machine learning algorithms can be divided into four categories: Supervised learning, Unsupervised learning, Semi-supervised learning, and Reinforcement learning.

In this project, we will focus on supervised learning. In supervised learning, we are feeding the algorithm with the training set and their relative labels. The supervised learning algorithms we will analyze are:

- Logistic Regression
- Decision Trees
- Random Forests
- Naive Bayes

Once we define our model, we have to train the model on the training set, validate it on the validation set and then test it on the test set. To evaluate our models we introduce some metrics and some definitions.

At the end of our training and validation phase, we will have a vector of predicted labels $Ypred$ to compare to the true labels $Ytrue$

Doing such comparisons we can come across four different scenarios:

- **TP:** correct predictions of positive class
- **TN:** correct predictions of negative class
- **FP:** wrong predictions of positive class
- **FN:** wrong predictions of negative class

The metrics we will refer to are the following:

- **Precision:** the accuracy of the positive predictions

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** also called sensitivity or true positive rate, it is the ratio of positive instances that are correctly detected by the classifier

$$recall = \frac{TP}{TP + FN}$$

- **Accuracy:** it is the measure of all the correctly identified samples, mostly used where classes are balanced

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

- **F1 score:** which combines precision and recall

$$F_1 = 2 \text{ x } \frac{precision \text{ x } recall}{precision + recall}$$

- The **ROC** curve is a plot that represents the variance of true positive rate and false positive rate considering different thresholds values.Important is the area under the graph, the **AUC** metrics that tells how the model is able to distinguish between the classes.
  - $True\ Positive\ Rate = \frac{TP}{TP+FP}$
  - $False\ Positive\ Rate = \frac{FP}{FP+TN}$

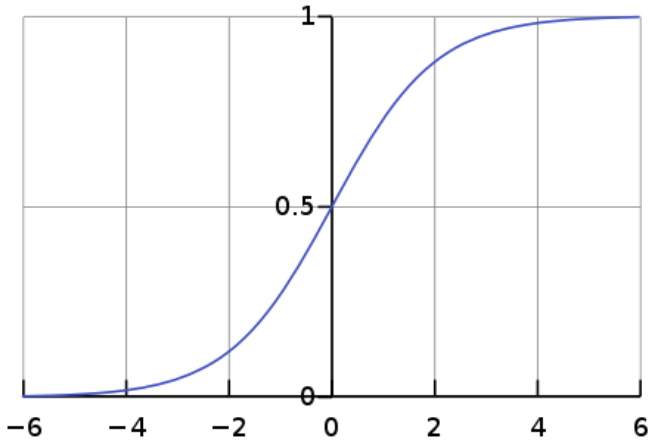
Logistic Regression

Logistic Regression is commonly used to estimate the probability that an instance belongs to a particular class. If the estimated probability is greater than 50%, then the model predicts that the instance belongs to that class (called the positive class, labeled "1"), or else it predicts that it does not (i.e., it belongs to the negative class, labeled "0"). This makes it a binary classifier. Like a Linear Regression model, a Logistic Regression model computes a weighted sum of the input features (plus a bias term), but instead of outputting the result directly like the Linear Regression model does, it outputs the logistic of this result

$$\hat{p} = \sigma(x^T w)$$

Where σ is the sigmoid function that outputs a number between 0 and 1, x are our training data and w the parameter of the model we want to find.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



29: The standard logistic function

If $\hat{P}$< 0.5, the predicted class is equal to 0
If $\hat{P}$>= 0.5, the predicted class is equal to 1

The way the model is trained can be derived by the cost function. The objective of training is to set the parameter vector w so that the model estimates high probabilities for positive instances (y = 1) and low probabilities for negative instances (y = 0)

$$c(w) = \begin{cases} -log(\hat{p}) & \text{if y} = 1 \\ -log(1-\hat{p}) & \text{if y} = 0 \end{cases}$$

PROS:

- Feature scaling not needed;
- Hyper-parameters tuning not needed.

CONS:

- Poor performance on non-linear data (image data);
- Poor performance with irrelevant and highly correlated features;
- Not a very powerful algorithm in respect to others.

Decision Trees

Decision trees can perform both classification and regression tasks, in this project we only consider the classification case.
A decision tree consists of split nodes and leaf nodes. Each split node performs a split decision and routes a data sample x to the left child node or to the right child node. Starting at the root node, the training data is recursively split into subsets. In each step the best split is determined based on a criterion. Commonly used criteria are **Gini** index and **Entropy**:

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$$

$$H_i = -\sum_{k=1}^{n} p_{i,k} log p_{i,k}$$

Where Pi, k is the ratio of class k instances among the training instances in the ith node. Another common hyperparameter the user can set in the model is the **max depth** of the tree.

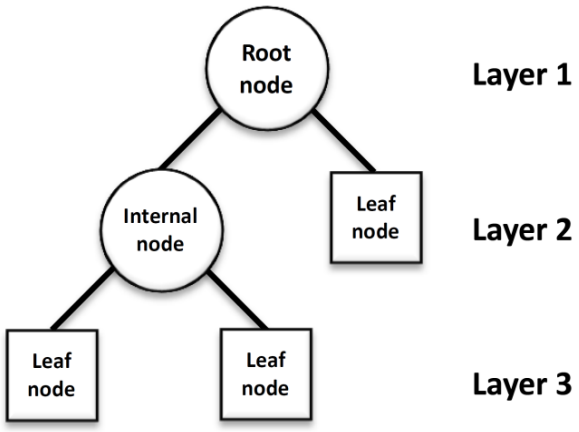


Figure 30: Graphical Representation of Decision Tree

PROS:

- Normalization or scaling of data not needed
- No considerable impact of missing values
- Easy visualization
- Automatic Feature selection

CONS:

- Prone to overfitting when using too much layers
- Sensitive to data. If data changes slightly, the outcomes can change to a very large extent
- Higher time required for training

Random Forest

A Random Forest is an ensemble of Decision Trees, trained via the **bagging method**. Ensemble combines multiple algorithms in order to improve accuracy and stability and also to avoid overfitting.
Starting from the original dataset D, B random samples are taken with replacement, note that the size of the samples is equal to the size of training data. Then one decision tree is trained for each sample B, this sampling technique is called **bootstrap**. Once the training phase is done, for each candidate split, a random subset of features ($\sqrt{n}$)is selected: This process is called **feature bagging**. At the end the class is assigned with a **majority vote** coming from all the decision trees. All this process can be observed in the figure below.

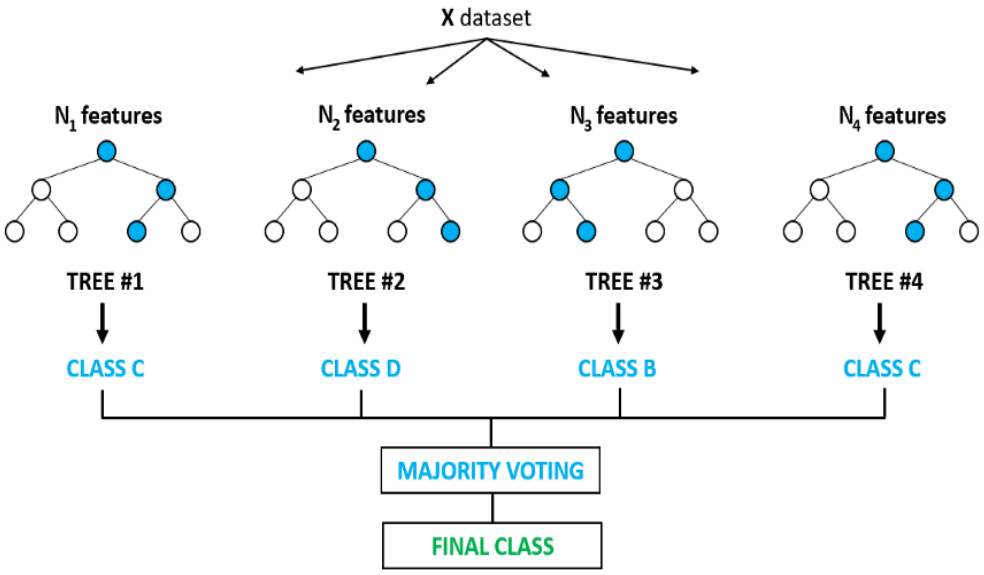


Figure 31: Graphical Representation of Random Forest

PROS:

- Good Performance on imbalanced datasets
- Handling of huge amount of data
- Good handling of missing data like decision trees
- Little impact of outliers
- Prevents overfitting
- Useful to extract feature importance

CONS:

- Predictions of the trees need to be uncorrelated
- Black Box

Some important Parameters in Random Forest:

- **max_depth: int, default=None** This is used to select how deep you want to make each tree in the forest. The deeper the tree, the more splits it has, and it captures more information about the data.
- **criterion**: {“Gini,” “entropy”}, default=” Gini”: Measures the quality of each split. “Gini” uses the Gini impurity while “entropy” makes the split based on the information gain.
- **max_features: {“auto,” “sqrt,” “log2”}, int or float, default=” auto”:** This represents the number of features that are considered on a pre-split level when finding the best split. This improves the model's performance as each tree node is now considering a higher number of options.
- **min_samples_leaf: int or float, default=1:** This parameter helps determine the minimum required number of observations at the end of each decision tree node in the random forest to split it.
- **min_samples_split: int or float, default=2:** This specifies the minimum number of samples that must be present from your data for a split to occur.
- **n_estimators: int, default=100:** This is the most important parameter. This represents the number of trees you want to build within a random forest before calculating the predictions. Usually, the higher the number, the better, but this is more computationally expensive.

Naive Bayes

The Naive Bayes algorithm is a supervised machine learning algorithm used for classification derived from the Bayes theorem. Given two events A and B, the bayes theorem can be written as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

This rule can be reformulated as it follows:

$$P(\text{hypothesis} \mid \text{evidence}) = \frac{P(\text{evidence} \mid \text{hypothesis})P(\text{hypothesis})}{P(\text{evidence})}$$

One assumption for applying the Naive Bayes classifier is that all samples have to i.i.d (independent identically distributed)

PROS:

- Real time predictions;
- Scalable with Large datasets;
- Multi class prediction is effectively done;
- Good performance with high dimensional data.

CONS:

- Independence of features does not hold: the fundamental Naive Bayes assumption is that each feature makes an independent and equal contribution to the outcome. However this condition is not met most of the times;
- Bad estimator: Probability outputs from predict_proba are not to be taken too seriously;
- Training data should represent population well: If you have no occurrences of a class label and a certain attribute value together (e.g. class=”No”, shape=”Overcast “) then the posterior probability will be zero. So if the training data is not representative of the population, Naive bayes does not work well.

Initial Models

The goal with the predictive model is to improve our chances of accurately predicting biopsy results. A model will be built to predict biopsy results using cytology results and other potential risk factors. Cytology results were included in the predictive models due to the fairly routine nature of this test. In the below image the result of the models with their different metrics is shown.

	Model	Train_Score	Test_accuracy	f1score	recall	precision	roc_auc
0	LogisticRegression	0.966667	0.957364	0.702703	0.684211	0.722222	0.831645
1	Decision Tree	1.000000	0.926357	0.558140	0.631579	0.500000	0.790685
2	Random Forest	1.000000	0.937984	0.555556	0.526316	0.588235	0.748514
3	GaussianNB	0.818333	0.844961	0.487179	1.000000	0.322034	0.916318

Figure 32: Initial Models and their results for different metrics

Specifically, as this is sensitive medical data, recall score needs to be given higher importance. We are choosing both "Decision Tree" and "Random Forest" models as our base model for doing optimization, although other models have higher recall and roc_auc scores.

8. Final Model and Optimization

Using Sampling Strategies

An approach to the problem of imbalance is to use some form of sampling, in order to balance the classes before giving them to the model. This allows for greater control of the data and domain appropriate strategy selection.

Oversampling

In oversampling we create additional data for the minority class either by making duplicates from the minority class or by some method to make additional synthetic data that is representative of the minority class.

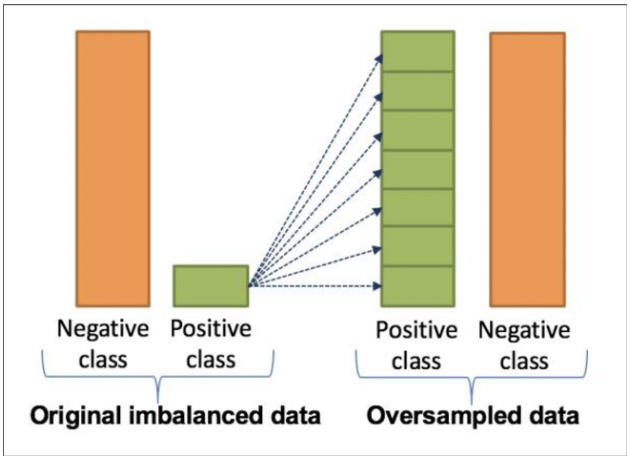


Figure 33: Oversampling Technique

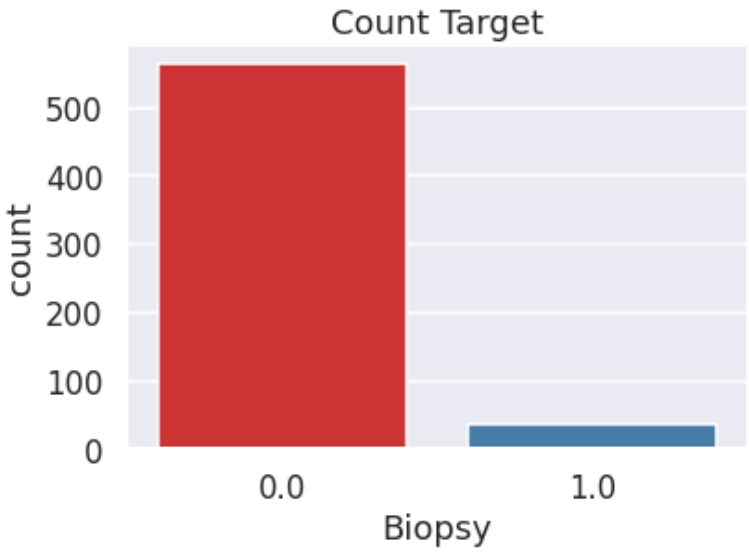


Figure 34: Imbalanced class

SMOTE

The Synthetic Minority Over-sampling TEchnique (SMOTE) generates new points for the minority class by fully connecting all points in the minority class with straight lines. Then for each existing data point SMOTE then determines a point on these interconnections to make a new point based on how many of the closest neighbours are considered for synthesis (k_neighbors). Now 0 and 1 classes have the same proportions. The recall and roc_auc score has improved for Random Forest and Decision Trees after SMOTE.

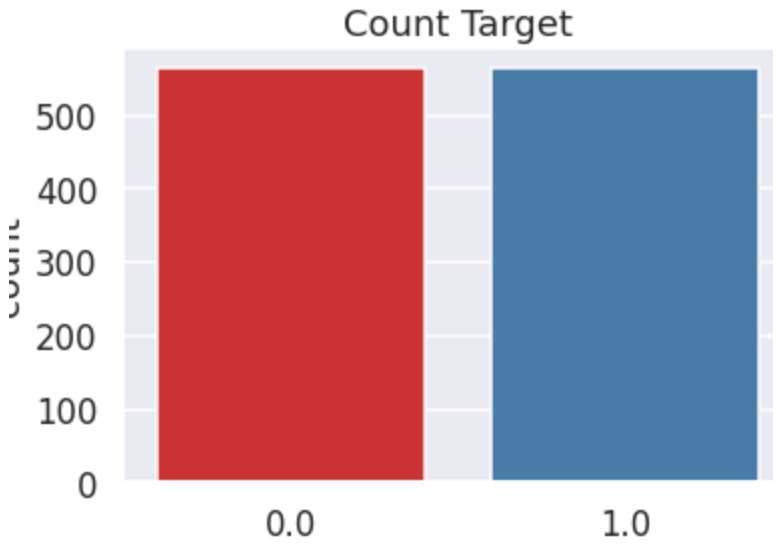


Figure 35: Balanced class after SMOTE

	Model	Train_Score	Test_accuracy	f1score	recall	precision	roc_auc
0	Decision Tree After Sampling	1.0	0.953488	0.739130	0.894737	0.629630	0.926448
1	Random Forest After Sampling	1.0	0.957364	0.744186	0.842105	0.666667	0.904316

36: Models results after SMOTE

Hyperparameter Tuning

Here we used **Grid Search Cross Validation** for Decision Trees and **Randomized Search Cross Validation** for Random Forest for choosing the best parameter values. The recall and roc_auc score has improved for Random Forest.

	Model	Train_Score	Test_accuracy	f1score	recall	precision	roc_auc
0	Decision Tree After Sampling	1.000000	0.953488	0.739130	0.894737	0.629630	0.926448
1	Random Forest After Sampling	1.000000	0.957364	0.744186	0.842105	0.666667	0.904316
0	Decision Tree after Hyperparameter Tuning	0.964539	0.937984	0.652174	0.789474	0.555556	0.869632
1	Random Forest After Hyperparameter Tuning	0.978723	0.965116	0.800000	0.947368	0.692308	0.956948

Figure 37: Models after Hyperparameter Tuning

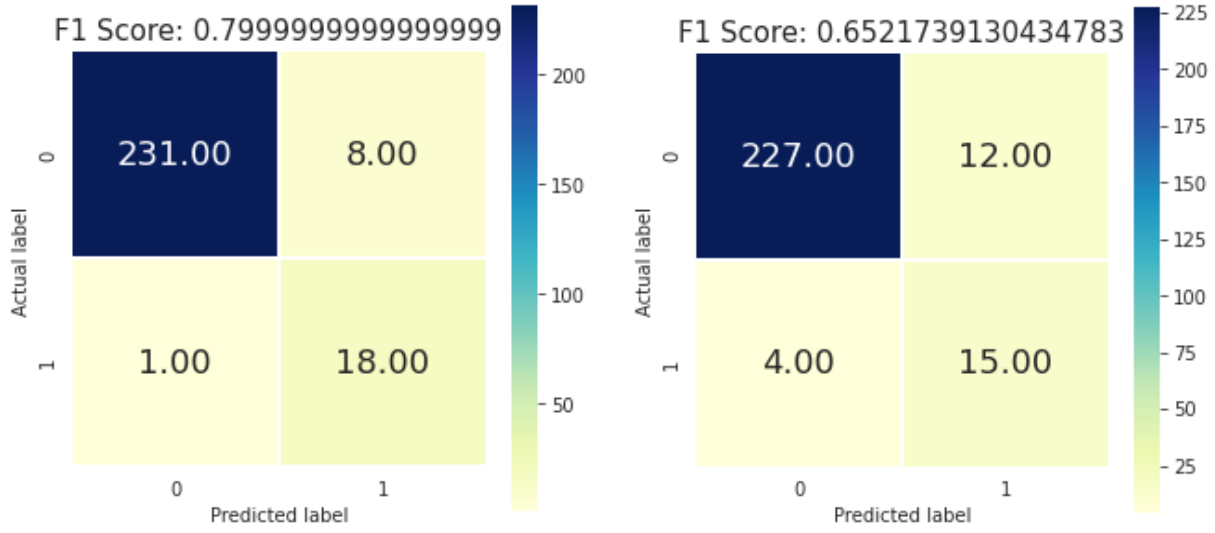


Figure 38: Confusion Matrix for Random Forest and Decision Trees

9. Conclusion

- Exploratory data analysis suggests that having positive cytology results, Number of Pregnancies, age, and hormonal contraceptives (years) may increase someone's likelihood to have positive biopsy results.
- This report presents the comparison between different machine learning classifiers for Cervical Cancer dataset. Since the dataset was imbalanced we used oversampling (SMOTE) to balance the dataset, then we applied hyperparameter tuning to improve the models.

10. Software

We used Google Colab to run the analysis. As far as the python libraries are concerned, we used:

- pandas
- scikit-learn
- matplotlib
- seaborn

11. References

- **Dataset: UCI's Machine Learning Repository:** https://archive.ics.uci.edu/ml/datasets/Cervical+cancer+%28Risk+Factors%29

- **Cancer.gov Website**: https://www.cancer.gov/types/cervical/pap-hpv-testing-fact-sheet#q3

- **towards data science: Practical tips for class imbalance in binary classification**: https://towardsdatascience.com/practical-tips-for-class-imbalance-in-binary-classification-6ee29bcdb8a7

- **Medium: Fundamental Techniques Of Feature Engineering For Machine Learning:** https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114
- **Missing data imputation with fancyimpute:** https://www.geeksforgeeks.org/missing-data-imputation-with-fancyimpute/

- **Medium: How to handle missing values :** https://medium.com/analytics-vidhya/how-to-handle-missing-values-byaryan-cb76b9dbaae2

- **Medium: Dealing with Imbalanced Data:** https://medium.com/digital-catapult/dealing-with-imbalanced-data-8b21e6deb6cd