SDS 2020 - Module 1: Group Assignment
Group members: Lukas Bielmeier, Marta Fedeli, Mikolaj Józef Grzesik, Youjing Zhu

# STAKEHOLDER REPORT: MACHINE LEARNING OPTIONS FOR CERVICAL CANCER RISK FACTOR ANALYSIS

## 1. Introduction

Although cervical cancer is easily prevented, many in developing countries are still greatly at risk, mostly because of issues regarding the screening process. While multiple steps indicate discernable characteristics for cervical cancer, the ultimate step to confirm cervical cancer is a biopsy. Our goal with this paper is to predictively classify positive and negative cancer diagnosis based on the independent features in the dataset so that a patient could immediately be pointed out to a biopsy only when very strictly necessary (when Machine Learning predicts biopsy to be positive).

## 2. Choice of data

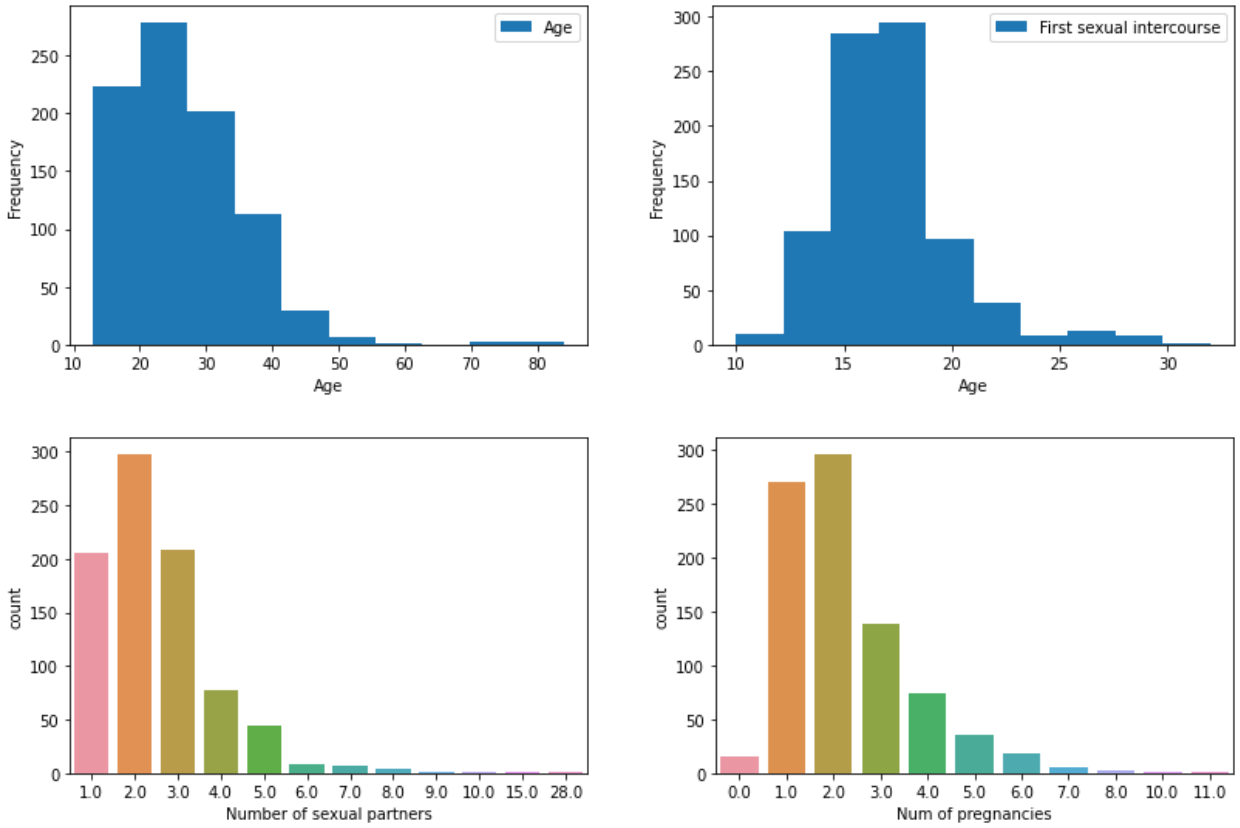
The dataset at hand was accessed and downloaded from Kaggle. Its data originates from a random sampling of patients attending the gynecology service at Hospital Universitario de Caracas in Caracas (Venezuela) between 2012-2013. It was collected in a previous study by Kelwin Fernandes K., Chicco D., Cardoso J. and Fernandes J. in 2017 where they applied transfer learning techniques for cervical cancer screening. With the same dataset, we tried to automatically classify patients at risk of cervical cancer by using Machine Learning (ML) methods instead.

## 3. Implementation

As it will be shown belows, we explored the unsupervised machine learning (UML) way to find the most significant pattern to automatically cluster patients in categories more/less at risk of cervical cancer based on their sexual behaviour. Then, through the supervised machine learning (SML), we also proceed with a binary classification type of problem, whether a patient at risk of cervical cancer would in fact have a positive or negative biopsy.

## 4. Data Visualization

First of all, we try some visualizations to get more visual and intuitive information about the dataset. The first four features present the general information about this random patient sample in terms of their sexual behaviour and age distribution.




The population dataset appears to be fairly young, mainly between 20 years old and 30, and the vast majority has been sexually active since their mid-teenage years and most of them have also already experienced two pregnancies.

In this age group the risk of cervical cancer is low in general. However, because the dataset refers to a population who mostly belongs to "the lowest socioeconomic status [...] with low income and educational level, being the population with the highest risk" (Fernandes et al, 2018, p. 7). Moreover, in developing countries, whether for lack of resources or because of poor adherence to screening routines because of low perception of the problem, this young population is very exposed to cervical cancer risk (Fernandes et al, 2018).

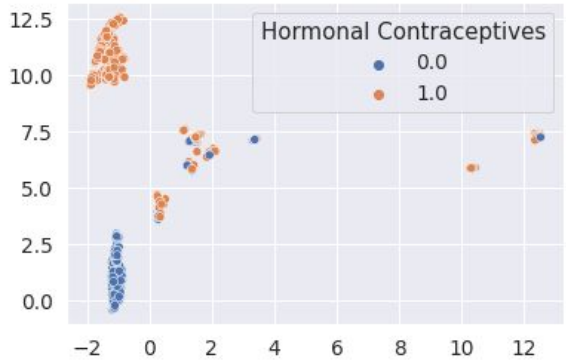
As it will be further explained below, we decided to draw attention to the sexual behaviour patterns of the population in the dataset because of the known link between STDs and cervical cancer.

## 5. Unsupervised Machine Learning (UML)

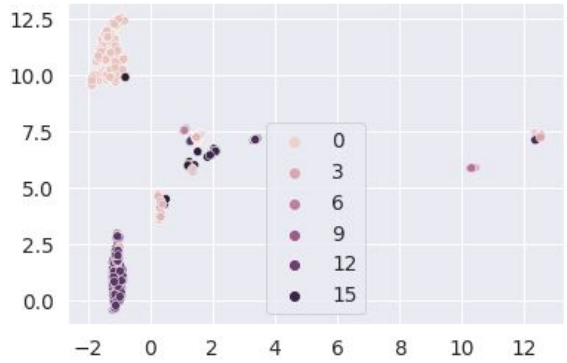
As far as finding any underlying pattern within this data without a target feature, out of the different features collected for the random sample, we firsty guessed that what could set the dataset's population apart the most, was the individuals' sexual behaviour (especially considering the connection between cervical cancer and HPV).

Rather than a higher number of sexual partners or early sexual activity, our first guess was a safe approach towards sexual conduct: we thus considered the use of Hormonal Contraceptives (HD) as a feature that could significatively set apart the different patients.

If it is true that under a medical point of view hormonal contraception does in fact slightly raise the risk of cervical cancer, however it was here approached in terms of contraception/protection in a patient sex life, as it seemed it could be more sigificant under this point of view.
However, there seemed not to be a correlation between hormonal contraceptives and STDs judging from our data. We hypothesize that this is due to the fact that the feature in itself, doesn't provide an exhaustive description of the patient sexual behaviour: if on the one hand hormonal contraceptives do not ensure protection from STDs, patients who have chosen this method over others could have done so because having a committed relationship with one single partner, thus not truly having a dangerous sexual behaviour in terms of STDs.
We identified this ambiguity as the reason for a potential correlation between STDs and hormonal contraceptives.
As in fact we proceeded by trying out UML, with PCA and UMAP dimensionality reduction first and KMeans clustering later, our hypothesis appears to be confirmed: Hormonal Contraceptive was a strong feature to set apart the different patients in the dataset. Although k in KMeans (and therefore, the most stable number for clusters) needed to be quite a high number and thus not so clearly split in two single, opposite clusters, the plotted graph shows that indeed clusters were split in a way that matches the Hormonal contraceptive affirmative-negative. As shown below, although there is a third cluster not very clearly defined and some background noise, out of the two significant clusters, clusters 0-3 match almost perfectly the 1.0 hormonal contraceptive value and clusters 12-15 the 0.0 value for hormonal contraceptives.




*HC variable after UMAP dimensionality reduction*




*KMeans clustering*

On the one hand this confirmed our initial hypothesis with regards to the latent relationship between hormonal contraceptives and cervical cancer. The underlying pattern in the dataset was captured quite well by applying UMAP as an UML method. On the other hand, because of the ambiguity of the feature itself, it would not prove such a valuable resource for detecting those more prone to be cancer positive with a positive biopsy and those who are not, and therefore, it would not be the preferred way to serve our purpose.

## 6. Supervised Machine Learning (SML)

First, we need to consider the nature of the problem to tackle and which feature we want to predict. In the present case, we are trying to distinguish whether or not a person has cervical cancer based on some risk factors. Our target feature for SML is hence the feature *Cancer* (the renamed *Biopsy*), which indicates whether or not a person has cancer based on the medical procedure of a biopsy whose final result is unambiguous.

If the feature *Cancer* shows the score **1**, it means that the person was found to have cervical cancer, while **0** indicates the opposite. As we are trying to differentiate between two possible values for our target variable *Cancer*, we are dealing with a **binary classification problem**. This is important to know before starting to interpret SML results.

The main idea behind our SML application to the cervical cancer risk factors dataset is to be able to predict and classify cervical cancer based on various risk factors. To do so, we first define our target variable **y** and our explaining variables **X**. As we are trying to predict and classify whether or not a person has cervical cancer, we define our **y** value as just one variable, which is the target feature *Cancer*. In turn, we define **X** as all features **except** for the target feature *Cancer*. A train and test set for **y** and **X** are prepared, where the train set of **X** and **y** is used to train our model for classification, while the test set is then used to test the model based on the train set.

We decided for a test size of 0.25 for the *train_test_split* function, meaning that 75% of data is used to train the model and 25% is used for the testing. For our model to be as accurate as possible, it is important to have a large enough train set for the model to be able to properly learn for classification.

We proceeded by applying three different classifier algorithms and comparing them based on their performance in classifying the target feature. The first, the RandomForestClassifier, creates decision trees on randomly selected data samples and gets predictions from each tree, selecting the best solution by means of voting. The second, KNeighborsClassifier, checks the distance from test samples to the known values of training samples. The group of data points that would give the smallest distance between the training points and the testing point is the class chosen. Lastly, we used XGBClassifier, a boosting algorithm based on gradient boosted decision trees algorithm instead of neighbors.

EEach classifier is tested for their accuracy score in correctly classifying people with and without cervical cancer. This core can be optimized by so-called hyperparameter tuning. This method will loop through predefined hyperparameters (such as the number of estimators of a classifier) allowing to find the best possible value for our chosen hyperparameter. Below, the accuracy scores before and after hyperparameter tuning for each of the three classifiers are reported.

| | Accuracy score **before** hyperparameter tuning | Accuracy score **after** hyperparameter tuning |
|---|---|---|
| **RandomForestClassifier** | 0.9488372093023256 (n_estimators = 100) | 0.958139534883721 (n_estimators = 50) |
| **KNeighborsClassifier** | 0.9255813953488372 (n_neighbors = 5) | 0.958139534883721 (n_neighbors = 1) |
| **XGBClassifier** | 0.9627906976744186 (n_estimators = 100) | 0.9534883720930233 (n_estimators = 150) |

Whilst having improved accuracy for the first two, hyperparameter tuning resulted in a reduced accuracy for the XGBClassifier. This is mostly due to an interaction between the different hyperparameters that changes as the number of estimators of the XGBClassifier changes. As the scope of this assignment does not allow for a deeper dive into this matter, we will proceed with this classifier with the model specifications as before hyperparameter tuning.

It is not enough to look at the accuracy score to fully grasp the effectiveness of the different classifiers: other metrics such as the recall and f1-score need to be also considered. As we are dealing with a binary classification case, recall can also be understood as sensitivity. The higher this score, the better. The f1-score on the other hand is the weighted average of precision and recall. Again, the higher the f1-score, the better. Both of these measures can be observed for each classifier in the below classification reports and their respective confusion matrix.

```
[[196   2]
 [  7  10]]
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       198
           1       0.83      0.59      0.69        17

    accuracy                           0.96       215
   macro avg       0.90      0.79      0.83       215
weighted avg       0.96      0.96      0.95       215
```

*Matrix 1: : RandomForestClassifier*

```
[[197   1]
 [  8   9]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       198
           1       0.90      0.53      0.67        17

    accuracy                           0.96       215
   macro avg       0.93      0.76      0.82       215
weighted avg       0.96      0.96      0.95       215
```

*Matrix 2: KNeighborsClassifier*

```
[[195   3]
 [  5  12]]
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       198
           1       0.80      0.71      0.75        17

    accuracy                           0.96       215
   macro avg       0.89      0.85      0.86       215
weighted avg       0.96      0.96      0.96       215
```

*Matrix 3: XGBClassifier*

As shown above, the confusion matrix can be seen in the top left corner of each output. This number shows how many true positives/negatives as well as the amount of false positives/negatives. Therefore, in comparison, the RandomForestClassifier and the KNeighborsClassifier provide a similar performance in terms of correct classification. The XGBClassifier on the other hand appears to be more slightly more successful in its classification, (195 true negatives and 12 true positives). In terms of precision all three have quite similar performance. However, in terms of recall and f1-score the XGBClassifier outperforms both the RandomForestClassifier and the KNeighborsClassifier, especially where the target feature has the value 1 a clear difference in recall and f1-score can be observed. Consistent with the earlier provided accuracy scores, each classifier achieves roughly a 96% accuracy.

Therefore, although all classifiers perform quite good, the XGBClassifier appears to perform best, especially when other metrics of the classification report, such as the recall and f1-score, are taken into consideration. These results are especially intriguing when considering the base of each of the algorithms. Both the RandomForestClassifier and the XGBClassifier use decision tree classification, while objects using the KNeighborsClassifier are classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. Interestingly, both the RandomForestClassifier and the KNeighborsClassifier exhibit the same accuracy score after parameter tuning even though using a different calculation base in their algorithm.

In conclusion, the XGBClassifier is the best option to carry out SML for our binary cervical cancer classification problem. It has the highest accuracy score while also outperforming the other classifiers in relevant sensitivity metrics such as the recall and f1-score, thus ensuring the highest level of performance.

**Bibliography & Sitography**

"Cervical Cancer Risk Classification". Retrieved at https://www.kaggle.com/loveall/cervical-cancer-risk-classification

Fernandes K., Chicco D., Cardoso J.S., Fernandes J. (2018) *Supervised deep learning embeddings for the prediction of cervical cancer diagnosis*. Retrieved at https://peerj.com/articles/cs-154.pdf