Cervical Cancer Classification
supervised learning

Sonia Petrini

[1] University of Milan
[2] Data Science and Economics

Abstract. The purpose of this work is to identify the most accurate and sensitive test for the diagnosis of cervical cancer, which is a relevant and urgent issue given the need for a prompt result, and the monetary and time costs. To address it, we will exploit supervised learning techniques for classification with four different target tests. *Schiller's test* will prove to yield the best cervical cancer prediction.

Keywords: Cervical Cancer · Supervised Learning · Classification

1 Introduction

Cervical cancer is very difficult to diagnose at an early stage, and the available tests are costly and not easily accessible. The aim of this analysis will thus be to find the test which allows to best detect the presence of cervical cancer with the information available about the patient. In this way, we will facilitate the choice of the first test to be performed when cervical cancer is suspected.
This work will be thus focused on prediction trough classification, which will target four of the most commonly used tests, namely *Hinselmann*, *Schiller*, *Citology*, and *Biopsy*, looking for the best model for each of them. Then, the four build models performances will be compared, to identify the most accurate and sensitive cervical cancer test. As a technical note, the *mlr3* framework will be used for model fitting and parameters tuning. This package was created to collect all the machine learning tools available in R in a unified setting, and it provides many convenient devices to implement, evaluate, and visualize the analysis' process.

2 Data

2.1 Presentation

Th *Cervical Cancer Risk Classification* dataset, available on *Kaggle*, is structured in the following way:

- **observations**: 858 - patients subject to the study
- **covariates**: 32 - relevant information for diagnosing cervical cancer

The variables provided concern general patient information, sexual activity, IUD (Intra-Uterine Devices) usage, possible presence of STDs (Sexually Transmitted Diseases), and past diagnoses of severe cancer-related illnesses. The peculiarity of this dataset is the presence of four possible target binary variables, all indicating the positive diagnosis of cervical cancer according to a specific test:

- *Hinselmann*
- *Schiller*
- *Citology*
- *Biopsy*

In order to perform the analysis, some pre-processing was done on the data. First, we removed both the time from first and from last STDs diagnosis, as almost all of their observations were missing. Second, the variables related to cervical condylomatosis and AIDS have also been excluded from our dataset, as every observed patient was negative to both of them, thus they were completely uninformative. Following the same reasoning, also vaginal condylomatosis, pelvic inflammatory disease, genital herpes, molluscum contagiosum, Hepatitis B, HPV, have been excluded, as they only displayed one positive case. The decrease in bias induced by the presence of these variables could not offset their increase in variance.

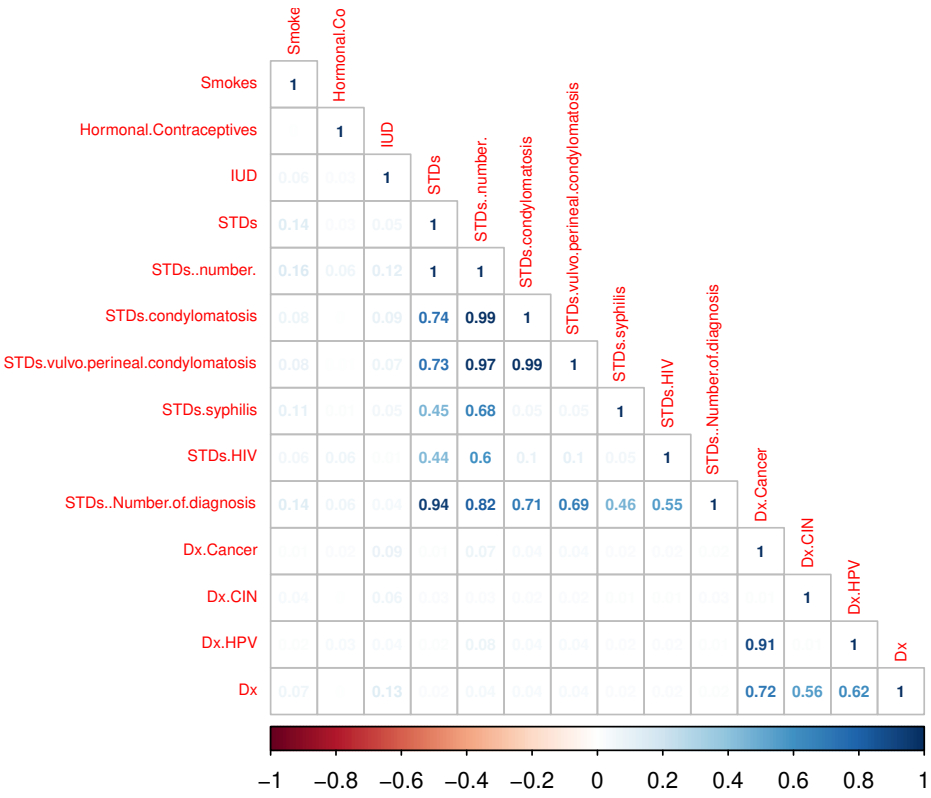

Fig. 1: Correlation matrix based on Cramer's V statistics

Since the majority of the variables are binary, we can compute the correlations among them using *Cramer's V correlation*: it is based on the Chi squared test and it takes values between 0 and 1 (the resulting correlation matrix can be seen in **Figure 1**). As it can be expected, many variables are displaying a high correlation. We will partially address this problem by exploiting the Random Forrest algorithm, which aggregates the performances of many tree base learners, optimizing the nodes' splitting to a different set of variables, thus creating uncorrelated trees. However, we also exclude the number of STDs and the number of STDs diagnoses, as they are both highly correlated among them, and with the considered STDs.

Concerning the missing values, numerical variables have been filled with their median value. This method does not have great precision, and can bias the variables' distribution, but it is widely used, as it is practical and functional. Filling the binary variables would need a deep medical knowledge, and we thus decide to remove the remaining rows containing missing values.

2.2 Exploration

Even if the focus of this work is on prediction, we can get an understanding of the features which may actually be relevant for diagnosing cancer by visualizing their relation with the targets. In particular, we would like to bring attention on the variables related to *hormonal contraceptives*: a binary variable taking value 1 if the patients is currently using hormonal contraceptives, and a numerical variable for the years of usage.

By visualizing the contingency tables for the dummy variable with respect to the four targets,which are depicted in **Figure 2**, it seems like using hormonal contraceptives does not have a relation with cervical cancer. In fact, *Schiller* and *Biopsy* don't seem to make any significant distinction between patients who use hormonal contraceptives and patients who don't, while *Citology* and *Hinselmann* provide inconsistent evidence. However, when considering the time dimension, it actually seems that a longer usage of hormonal contraceptives is indeed associated with all the four targets. The combination of these two considerations leads us to think that the risk factor for cervical cancer are not the contraceptives themselves, rather their prolonged use. This is a significant matter, which is probably not discussed enough.

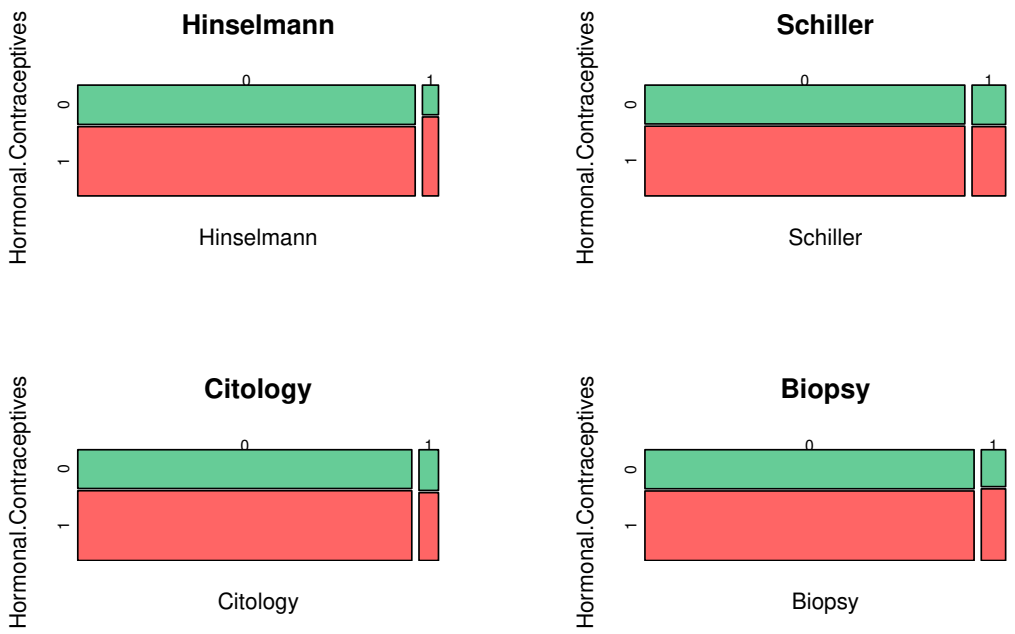

Fig. 2: Cross tabulation of binary variable for Hormonal Contraceptives usage and the four targets

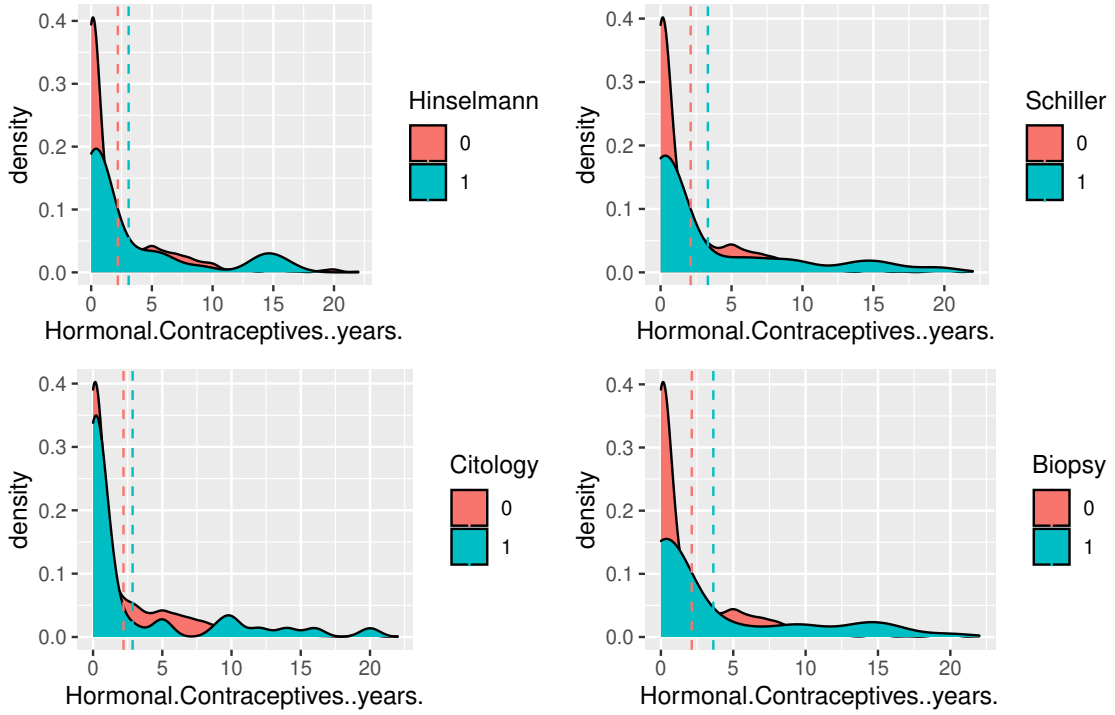

Fig. 3: Density plots of years of Hormonal Contraceptives usage and the four targets

3 Methodology

In this section we will present the method followed for our analysis. The same approach has been used for all the four targets, adapting it to the peculiarities of each of them. The procedure is based on the following steps:

- Oversample the minority class in the Train set
- Fit a Logistic Regression as a *Benchmark*
- Fit a Random Forrest to exploit the advantages of *Ensembling Methods*
- *Optimize* the Random Forrest's parameters
- *Compare* the performances of the three models

Once the best model has been chosen for each test, they are compared with one another. This procedure allows us to find the model with the highest predictive performance, for the best test for cervical cancer prediction, given the known information.

3.1 Data Balancing

To begin with, it is important to understand the balance of the dataset, i.e. the ratio between positive and negative cases. In the medical field this ratio is often very low, especially when dealing with rare illnesses.

Table 1: Dataset Balance by Target

	Value	**Freqs (% of Valid)**
Hinselmann	0	693 (95.5%)
	1	33 (4.5%)
Schiller	0	657 (90.5%)
	1	69 (9.5%)
Citology	0	686 (94.5%)
	1	40 (5.5%)
Biopsy	0	676 (93.1%)
	1	50 (6.9%)

As we can see in **Table 1**, the dataset is strongly unbalanced with respect to all four targets. Noticeably, the four tests provide different results for the same patients. In particular, *Hinselmann* is the most "optimistic" test, with a share of positives corresponding to 4.5%, while *Schiller* is the most "pessimistic" one, labelling 9.5% of patients as positives.

The way to deal with this kind of situation are *resampling methods*, which allow to increase the numerosity of the minority class by introducing new synthetic data, or to undersample the majority class. Since our data already contains a limited number of observations, we will proceed with the first method. In particular, we will exploit a SMOTE function for both categorical and numerical variables.

Despite not being reported here, the first step of the analysis had been to fit both logistic regression and Random Forrest on the raw data; the built learners were however not able to predict any label as positive, no matter how low the threshold. As a matter of fact, when dealing with this kind of unbalanced classification, the task can be very problematic to solve without resampling.

An important point concerns *when* to perform resampling. It is important to apply the SMOTE after train-test splitting, otherwise the learner will be facilitated in classification, as it could have to predict a synthetic test example which is identical to a training one.

3.2 Modelling

The first model to be fit for each target is **Logistic Regression** which, despite not yielding the best accuracy, usually provides a stable generalizable model. Logistic Regression will be used as

a benchmark for more sophisticated models. An issue with Logistic Regression is the presence of correlated variables, which can alter the coefficients' standard errors. This issue can be addressed through the employment of **Random Forrests**; these algorithms exploit both resampling of training data, of features, and ensembling.

Of course, since we are aggregating the results of many different base learners, we expect to obtain a better performance. Moreover, to try improving the performance even further, we will exploit the *mlr3 library* to perform hyper-parameters tuning on the Random Forrest algorithm. The parameters we are going to tune through a grid search are:

- *splitrule*: splitting rule, one among "Gini" and "Extratrees".
- *importance*: one among 'none', 'impurity', 'impurity_corrected', 'permutation'. 'Impurity' corresponds to the Gini index.
- *min node size*: Minimal node size (default for classification is 1).
- *num trees*: Number of trees to grow.

To allow the comparison among all the different models, we will exploit a custom-made function which plots the values of sensitivity, specificity, and accuracy for a grid of thresholds. In this way, we can identify the threshold granting a sufficient balance among the different types of error: even if we are mainly interested in identifying true positives, we also want to have a good rate of true negatives identified. In fact, given the monetary and time cost necessary to perform the test, we want to make sure that the available resources are efficiently and wisely managed.

4 Empirical Analysis

4.1 Hinselmann's Test

We will begin by exploring the first, more optimistic test for cervical cancer, namely *Hinselmann's*. To implement the SMOTE function we need to specify two parameters: the number of neighbours and the proportion we want to obtain among positives and negatives. Given that this target is the most unbalanced, we choose a *balancing* equal to 40 (100 corresponds to same number of negatives and positives); notice that we don't want this value to be too high, otherwise we will incur in overfitting, and we will completely undermine the original balance.

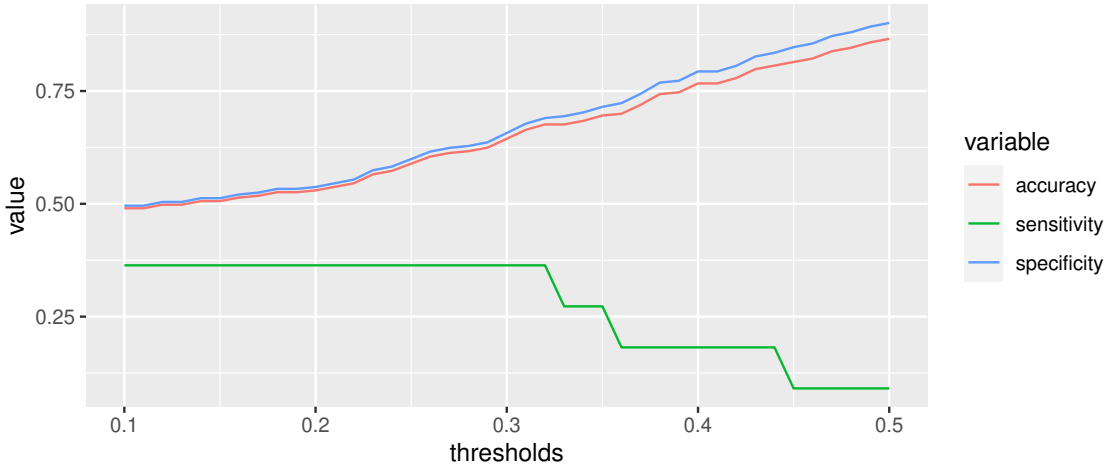

Fig. 4: Accuracy, sensitivity, and specificity values for the different threshold values.
Test performance of logistic regression - Hinselmann

The values of test and train sensitivity, specificity, and accuracy obtained with each model after threshold optimization are displayed in **Table 2**. We immediately notice the very low value recorded for test sensitivity in the *Logistic Regression*. From **Figure 4**, we can observe how, when

fitting this baseline model, the learner is never able to correctly identify more than 36.4% of the positive patients. To improve this situation, we proceed with the implementation of the *Random Forrest*.

Noticeably, despite having improved, the test performance of the model is still not satisfactory, as it is only slightly better than a random guess. On the other hand, training performance has considerably improved, suggesting an *overfitting* issue. The model we have build has too much variability, and is not able to generalize the information it is learning to the new data.

Table 2: Hinselmann's Models Comparison

	Train Sens.	**Test Sens.**	**Train Spec.**	**Test Spec.**	**Train Acc.**	**Test Acc.**
Logistic	0.778	0.364	0.787	0.698	0.784	0.684
Random Forrest	0.989	0.545	0.976	0.537	0.979	0.538
Random Forrest Opt	0.978	0.545	0.98	0.574	0.979	0.573

To try and overcome this issue, we perform threshold tuning, focusing on those parameters which allow to control the depth of the base trees. Indeed, when the trees are not constraint in their growth, the Random Forrest algorithm can yield this kind of result, which reflects the high variability issue of very big and deep trees upon which it is based. Moreover, by taking a larger number of trees, we are able to aggregate more results, ideally reducing variance. The parameters related to the size and numerosity of the trees are the *minimum node size* and the *number of trees*.

The optimization is performed through a grid search, and the stop criterion specified is any between stagnation (no more improvement) and the reaching of 40 evaluations. Indeed, the distance between test and train performance is slightly reduced, even if the improvement is lower than expected. Still, we are able to identify the best model to predict cervical cancer through **Hinselmann's test**, namely the ***Optimized Random Forrest***, whose optimized parameters are:

- *splitrule*: "Gini"
- *importance*: "none"
- *min node size*: 12
- *num trees*: 5000

As we can see, the minimum node size has been set to 12 which, compared to the default of 1, constraints the trees to be smaller.

4.2 Schiller's Test

Next, we are going to evaluate the most pessimistic target, with labels as positive 9.5% of the patients. The results are shown in **Table 3**. Given the lower imbalance compared to the other considered tests, we choose a *majority percentage* parameter equal to 30 for the SMOTE function (we used 40 in the previous section), while the k parameters is still 10. Compared to the results obtained with Hinselmann's, *Logistic Regression* is better able to classify test data, but the results are still unsatisfactory. However, there is more balance among test and train performance, meaning that the training error is now a better approximation of the real risk of the learner.

This improvement with respect to the overfitting issue emerges also from the *Random Forrest* model: training performance is slightly worse comparing to Hinselmann's, but the performance on test has notably improved, reaching suitable levels.
The optimized *Random Forrest* has the following set of parameters:

- *splitrule*: "Gini"
- *importance*: "none"
- *min node size*: 10
- *num trees*: 2000

Table 3: Schiller's Models Comparison

	Train Sens.	Test Sens.	Train Spec.	Test Spec.	Train Acc.	Test Acc.
Logistic	0.68	0.417	0.654	0.437	0.66	0.435
Random Forrest	0.961	0.625	0.979	0.729	0.975	0.719
Random Forrest Opt	0.969	0.625	0.974	0.707	0.973	0.7

The chosen splitrule and importance are the same as in Hinselmann's, while the minimum node size is not as strict as in the previous optimized model. Moreover, the number of trees is noticeably smaller.

Overall, the best model for **Schiller's test** is the ***Random Forrest*** without optimization, which yields a test sensitivity of 62.5%.

4.3 Citology Test

We can now discuss the **Citology Test**, which has a positive rate of 5.5%. For this target, the selected balance percentage parameter after different trials is 50, while k is still set to 10. The results are shown in **Table 4**. Both test and train performance obtained through the *Logistic Regression* are higher compared to the first two targets. On the other hand, the built *Random Forrest* is performing worse than the Schiller's test, while having the same overfitting problem.

When optimizing the Random Forrest algorithm, we find that once again the non optimized model is performing better on test. However, training performance went down, meaning that through optimization we were partially able to address overfitting. For this reason, we will choose the ***Optimized Random Forrest*** as the best model for **Citology Test**, as we prefer a more stable model, which is able to generalize on new unseen data.

The optimized parameters are the following:

- *splitrule*: "Gini"
- *importance*: "impurity"
- *min node size*: 24
- *num trees*: 1180

The minimum node size is quite large, meaning that the base trees for the Random Forrest are small, which explains the improved stability of the model.

Table 4: Citology's Models Comparison

	Train Sens.	Test Sens.	Train Spec.	Test Spec.	Train Acc.	Test Acc.
Logistic	0.735	0.429	0.693	0.446	0.707	0.445
Random Forrest	0.987	0.563	0.975	0.567	0.979	0.567
Random Forrest Opt	0.969	0.5	0.964	0.546	0.966	0.543

4.4 Biopsy Test

For the last test, namely **Biopsy**, we choose a majority percentage parameter equal to 50, while k is still equal to 10. As we can see in **Table 5**, there has been an improvement in the test performance of *Logistic Regression*, while training performance has slightly decreased comparing to the previous target.

The *Random Forrest* algorithm is now performing better compared to the Citology test, both in terms of test performance and in terms of overfitting. Noticeably, the *Optimized Random Forrest* yields the same sensitivity as the base Random Forrest on test, but higher accuracy and specificity. In the meanwhile, training performance has become slightly worse, meaning that the optimization is indeed partially addressing the overfitting issue.

Once again, the ***Optimized Random Forrest*** is the best model, also for **Biopsy**, and it has the following parameters:

- *splitrule*: "Gini"
- *importance*: "permutation"
- *min node size*: 13
- *num trees*: 4819

The number of trees is very large, and the chosen importance measure is *permutation*, which measures the change in classification error after a feature's value is modified.

Table 5: Biopsy's Models Comparison

	Train Sens.	**Test Sens.**	**Train Spec.**	**Test Spec.**	**Train Acc.**	**Test Acc.**
Logistic	0.7	0.529	0.689	0.487	0.692	0.49
Random Forrest	0.977	0.647	0.98	0.581	0.979	0.585
Random Forrest Opt	0.973	0.647	0.977	0.602	0.976	0.605

4.5 Best Model

Now that we have chosen the best model for each of the available targets, we can make a final comparison to identify which is the most reliable test given the information provided about the patient. The results are shown in **Table 6**.

Table 6: Best Models Comparison

	Train Sens.	**Test Sens.**	**Train Spec.**	**Test Spec.**	**Train Acc.**	**Test Acc.**
Hinselmann	0.978	0.545	0.98	0.574	0.979	0.573
Schiller	0.961	0.625	0.979	0.729	0.975	0.719
Citology	0.969	0.5	0.964	0.546	0.966	0.543
Biopsy	0.973	0.647	0.977	0.602	0.976	0.605

Even if *Biopsy* has a slightly higher percentage of true positives identified comparing to *Schiller*, the latter has considerably higher values of specificity and accuracy. As we said before, we are not only interested in correctly identifying patients with cervical cancer, but also in not wasting limited resources which could benefit someone who really needs them. Moreover, *Schiller's test* is also performing better in terms of overfitting, thus yielding a more stable and reliable model.

It is important to notice that Schiller's test is also the one with the highest number of original positive occurrences, and this could play a role in its higher performance. The greater number of positives identified by this test could be due to a sort of "pessimistic criterion", according to which it is better to classify a patient as having cancer when this is not true rather than the opposite.

5 Conclusions

In this work we exploited supervised learning techniques to contribute to the addressing of a relevant issue, which is the prompt detection of cervical cancer in women. Given the limited resources for conducting tests, it is important to know which are the most effective and efficient instruments to diagnose cancer timely. Thee models have been built for each of the four available cancer tests: *Logistic Regression*, *Random Forrest*, and an optimized version of the *Random Forrest*. After analysing each of these configurations, the best model for each target has been chosen, and we managed to identify the most useful and accurate test with **Schiller's**. While the overfitting problem has been addressed, it has not been completely solved. Further work on this topic could certainly focus on this issue, trying different models or tuning configurations.

6 Appendix

The R code used to perform the analysis is available on Github.