Characteristics of pathological complete response in people with rectum cancer

Cancer is characterized as a heterogeneous disease consisting of many subtypes, with the common denominator of all cancers is uncontrollably body cells division.

Rectal cancer is cancer that begins in the rectum - the last several inches of the **Large Intestine**. It starts at the end of the final segment of the colon and ends when it reaches the short, narrow passage leading to the anus.

Our research question is to try to find the features that explain best in rectal cancer patients, the differentiate between 5-year survivors and those who have died from the disease.

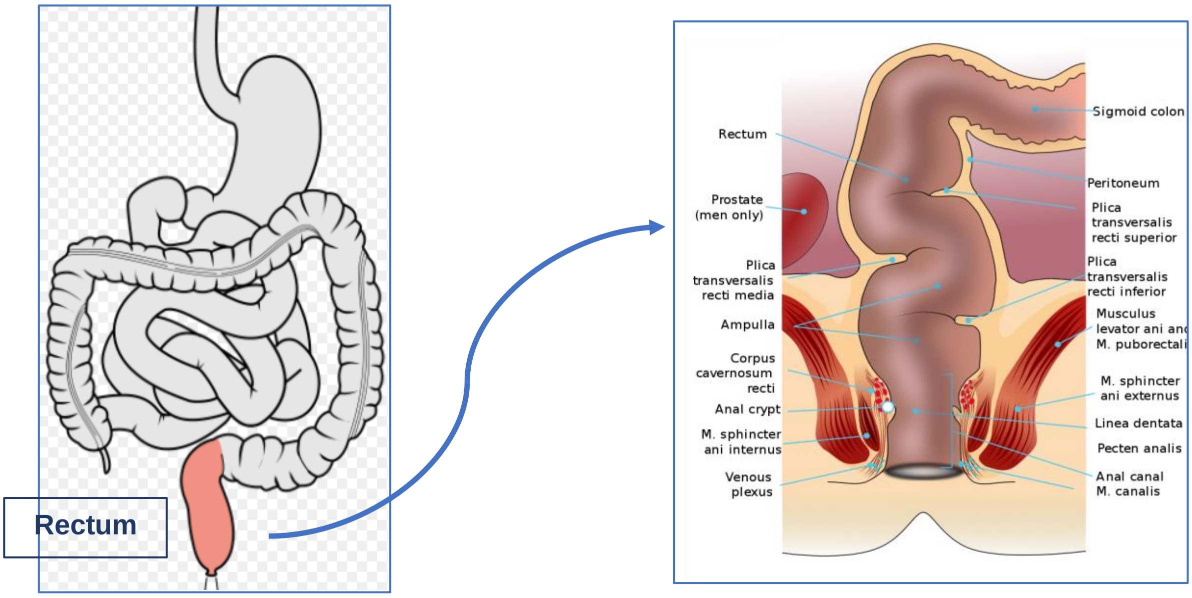
But before we approach the research question, it is important to know the characteristics of the cancer, demographic information about it as well as articles dealing with analysis of Rectal Cancer features using Machine Learning methods.

Clinical Background

The most common type of rectal cancer is **Adenocarcinoma** [1] (98%), which is a cancer arising from the mucosa - the layer wall lines the inner surface of the rectum. Cancer cells can also spread from the rectum to the lymph nodes on their way to other organs of the body

Rectal cancer shares many clinical features with colon cancer, so the medical literature treats rectal cancer in one of two ways: first as part of colon cancer, then it is called **colorectal cancer**, and secondly as stand-alone cancer and then it is just called **rectal cancer** [2].

In the following illustrations we can see the general location of the rectum in the lower part of the digestive system (left) and the detailed structure of the rectum (right).



Illustrations 1,2: taken from "Rectum" on wikipedia

Diagnosis of rectal cancer is performed using a rectal examination, followed by a biopsy. If the biopsy indicates a malignant tumor, additional imaging tests are needed to determine the size of the tumor and whether it has spread to other organs of the body.

In the following picture, taken by endoscopic camera, we can clearly see the smooth anal canal (the short tube through which stool leaves the body). At the bottom left of the picture we see the cancerous growth

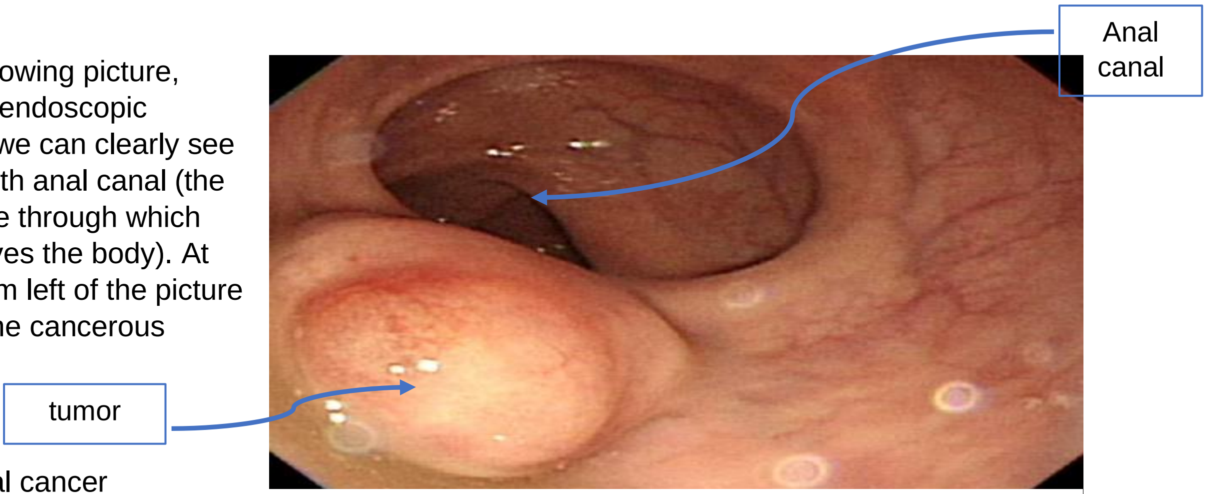
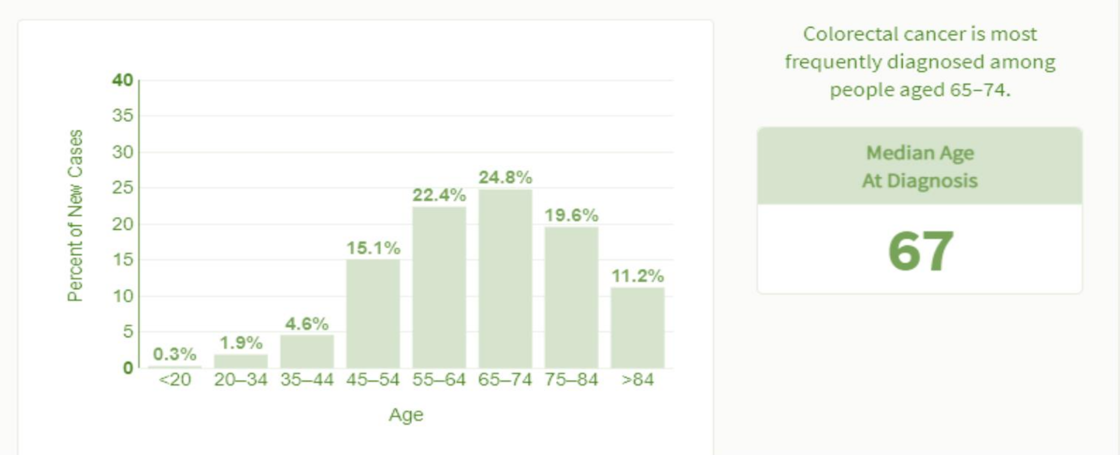


Figure 1: from scincephoto.com

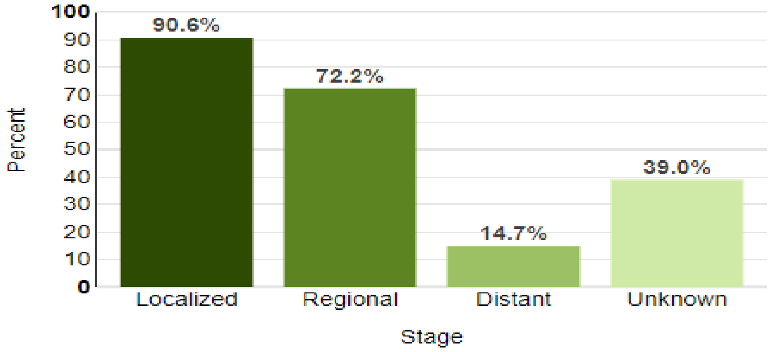
Colorectal cancer starts to be diagnosed around the age of 45 years of man and woman. The rate of diagnoses increases until the age of 70 years of man and woman, and then the rate starts do decline. The median at diagnosis is 67 as we can see in the following graph [3].



Graph 1: Percent of New Cases by Age Group. taken from SEER

After the diagnosis, the stage of the disease is determined. The stage of the disease describes the size of the tumor and whether it has spread to other organs. The numbering method divides the disease stage into four stages: stage 1 - a small tumor that has not spread, and up to stage 4 - a tumor that has spread to other organs. There is also an internal divisions of staging, addressing penetration of the tumor to the lymph nodes and spreading to other organs in the body.

The following graph shows that the less advanced the cancer is, the higher the chances of survival. when the cancer is diagnosed in stage 1 the chances of survival over 5 years are over 90%. On the other hand, if the cancer is diagnosed in stage 4 and has already metastasized to distant organs in the body, the chances of recovery drop to only 15%.



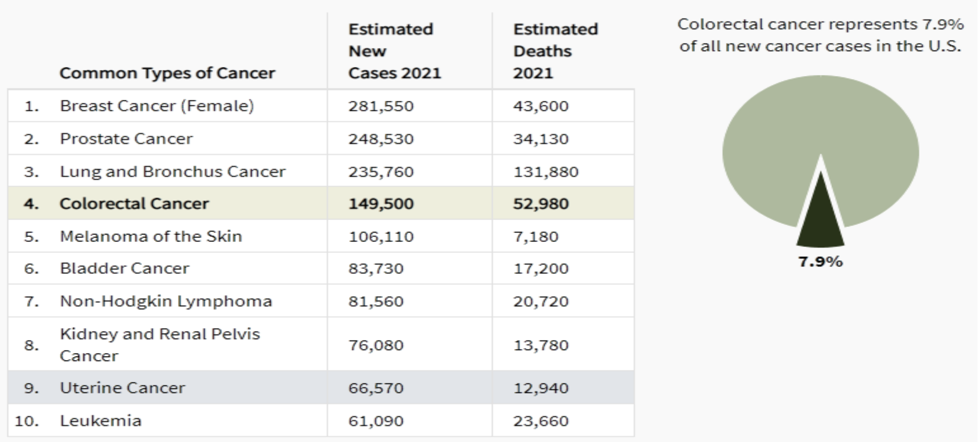
Graph 2: five-Year Relative Survival according to diagnostic stage. taken from SEER

Treatment for colorectal cancer usually involves surgery to remove the cancer. Other treatments, such as radiation therapy and chemotherapy, might also be recommended, and given in different doses and in different order, depending on the patient's age, his health status, stage of the disease and general chances of recovery.

demographic data

Colorectal cancer is the fourth most common cancer in men and women in the United States, accounting for 8% of all cancers. Rectal cancer accounts for 25% -30% of colorectal cancers [4].

As shown in the following graph, about 150,000 people In the United States are diagnosed with colorectal cancer each year. 65% of them survive over 5 years since diagnosis. Colorectal cancer is the second leading cause of cancer death in the United States. More than 50,000 people die in the United States each year and rates of mortality increase with age.



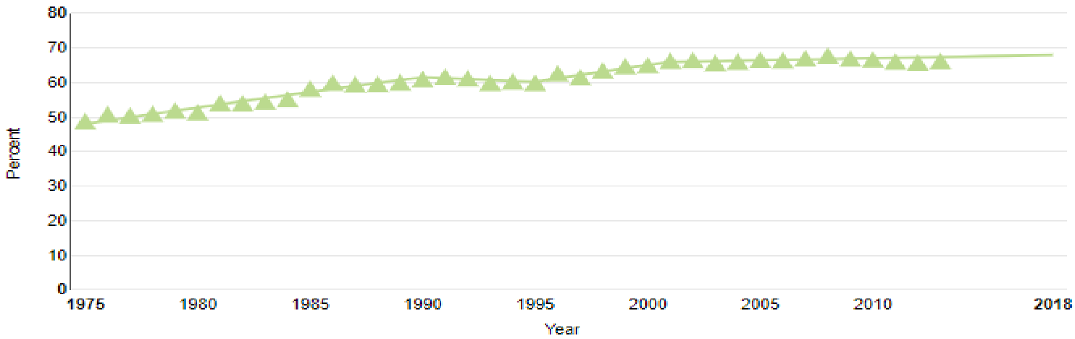
	Common Types of Cancer	Estimated New Cases 2021	Estimated Deaths 2021
1.	Breast Cancer (Female)	281,550	43,600
2.	Prostate Cancer	248,530	34,130
3.	Lung and Bronchus Cancer	235,760	131,880
4.	**Colorectal Cancer**	**149,500**	**52,980**
5.	Melanoma of the Skin	106,110	7,180
6.	Bladder Cancer	83,730	17,200
7.	Non-Hodgkin Lymphoma	81,560	20,720
8.	Kidney and Renal Pelvis Cancer	76,080	13,780
9.	Uterine Cancer	66,570	12,940
10.	Leukemia	61,090	23,660

Colorectal cancer represents 7.9% of all new cancer cases in the U.S.

Graph 3: Most common cancers in the US. taken from SEER

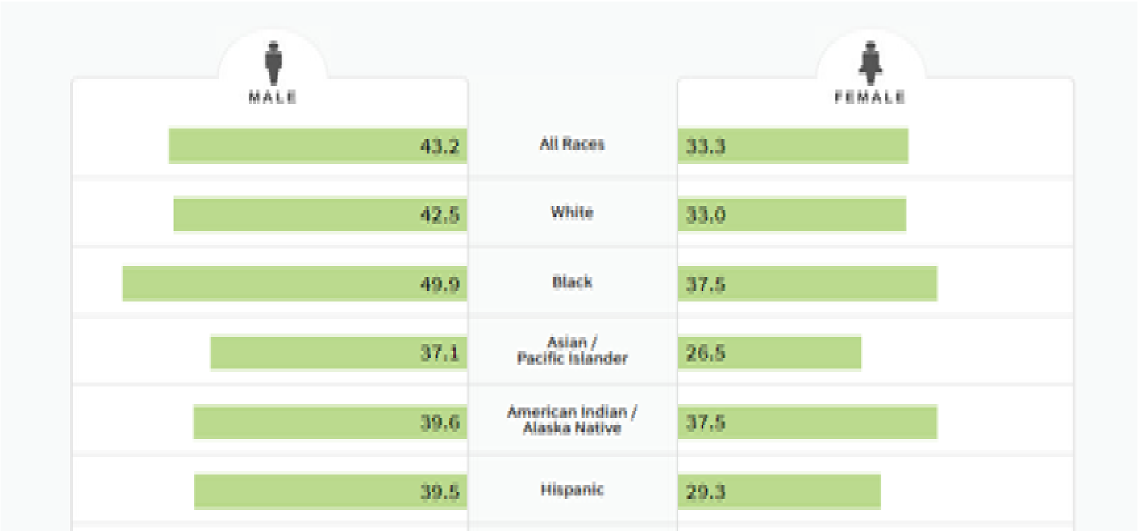
However, we are seeing an increasing trend in life expectancy in colorectal cancer patients. Over approximately 45 years of SEER data collection, the survival rate over 5 years from the date of diagnosis has increased from approximately 48% in 1975 to 65% in 2018.

.



Graph 4: five-Year Relative Survival. taken from SEER

Colorectal cancer is not evenly distributed in the population. People of different races have different chances of getting this cancer. We see that Colorectal cancer is about 1/3 more common in men than in women, it is more common in African Americans, and less common in Americans of Asian origin, Native American, and Hispanic origin.



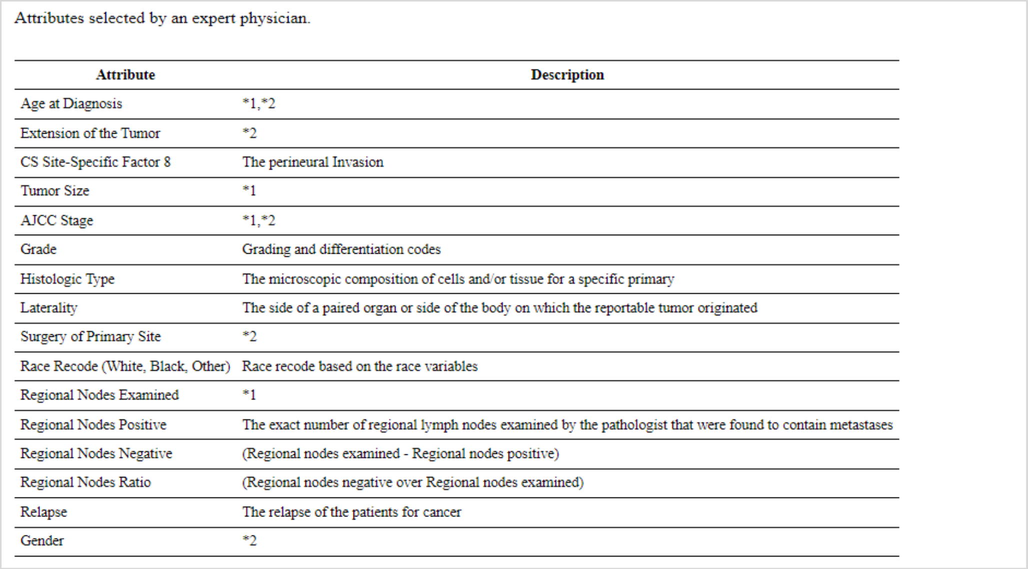
Graph 5: Death rate per 100,000 persons by Race/Ethnicity & Sex. Taken from SEER

Articles

The first article “Survivability Prediction of Colorectal Cancer Patients: A System with Evolving Features for Continuous Improvement” from 2018 [5], was published in a scientific-technological journal called sensors. The article deals with predicting the survival of colorectal cancer patients based on SEER data, and tries to detect if there is a difference in predicting survival over a period of 1 to 5 years, once using 18 features selected by specialized physician, and a second, using 6 feature selected by an algorithm.

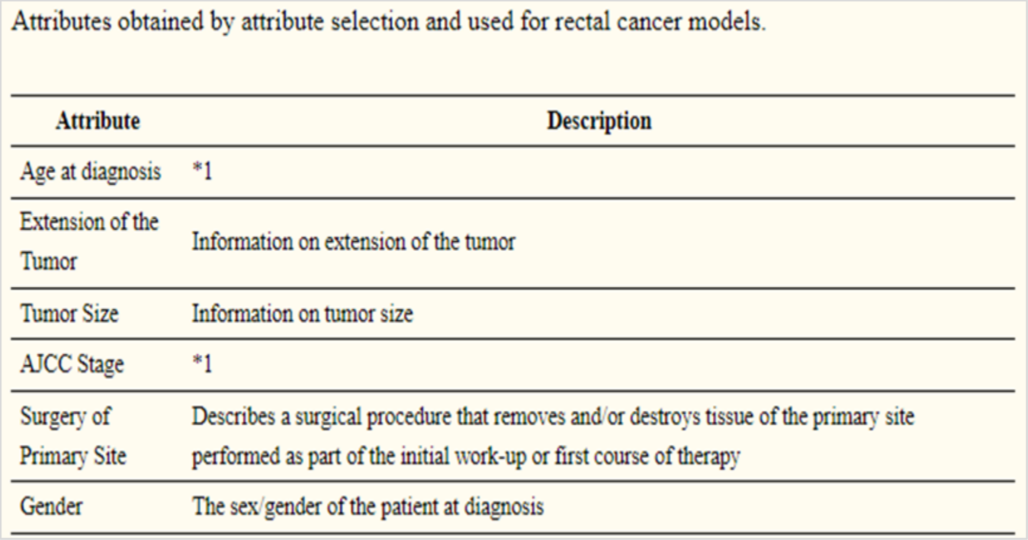
To select the features by the algorithm the researchers used the forward selection method - each time they added one feature and checked for the minimal error. then they add another feature - until they reached 6 features.

In the next two tables we can see the 18 features selected by the physician and the 6 features selected by the algorithm. Needless to say, the 6 features selected by the algorithm are included in the 18 features selected by the physician.



Attributes selected by an expert physician.

Attribute	Description
Age at Diagnosis	*1,*2
Extension of the Tumor	*2
CS Site-Specific Factor 8	The perineural Invasion
Tumor Size	*1
AJCC Stage	*1,*2
Grade	Grading and differentiation codes
Histologic Type	The microscopic composition of cells and/or tissue for a specific primary
Laterality	The side of a paired organ or side of the body on which the reportable tumor originated
Surgery of Primary Site	*2
Race Recode (White, Black, Other)	Race recode based on the race variables
Regional Nodes Examined	*1
Regional Nodes Positive	The exact number of regional lymph nodes examined by the pathologist that were found to contain metastases
Regional Nodes Negative	(Regional nodes examined - Regional nodes positive)
Regional Nodes Ratio	(Regional nodes negative over Regional nodes examined)
Relapse	The relapse of the patients for cancer
Gender	*2



Attributes obtained by attribute selection and used for rectal cancer models.

Attribute	Description
Age at diagnosis	*1
Extension of the Tumor	Information on extension of the tumor
Tumor Size	Information on tumor size
AJCC Stage	*1
Surgery of Primary Site	Describes a surgical procedure that removes and/or destroys tissue of the primary site performed as part of the initial work-up or first course of therapy
Gender	The sex/gender of the patient at diagnosis

Table 1, 2: In the upper table - the 18 features selected by physician and in the lower table - the 6 features selected by the algorithm. The features selected by the algorithm turn up in the features selected by the physician almost in the same order.

The researchers then trained several models in an attempt to predict survival. The models they examined were: KNN, Naive Bayes, Decision Tree and Random Forest

Eventually they built a hybrid model that combines these four methods. And then, they compared the hybrid model with six features to the same model with 18 features. The measurement for comparison were: Accuracy, AUC and F-measure

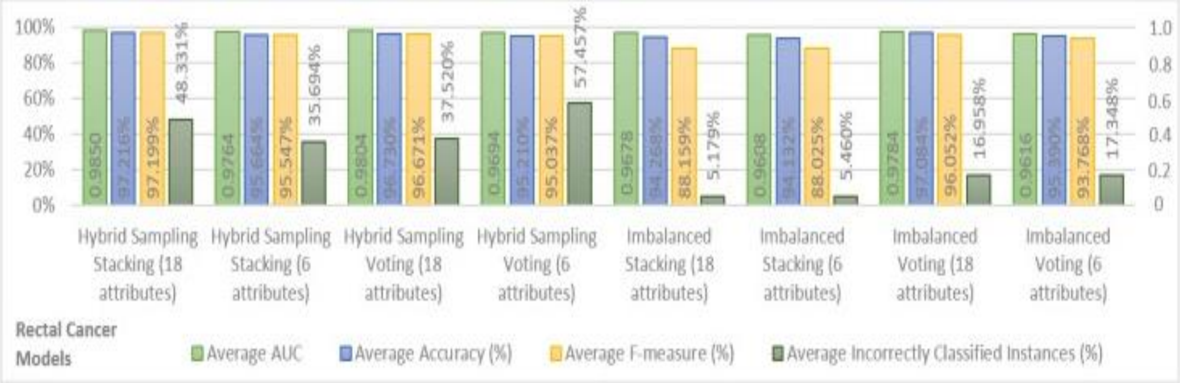
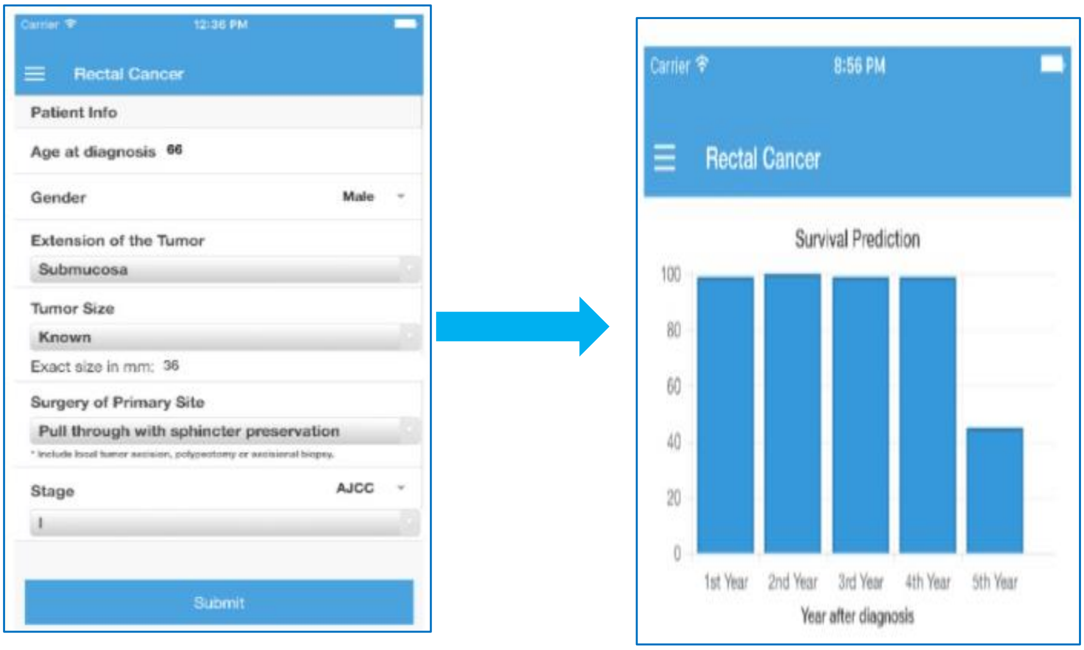


Table 3 :Comparative measures of the same models,
with 18 physician-selected features, versus 6 algorithm-selected features

As shown in the comparison in the table above, the results were very similar, with a very slight advantage for the model with the 18 features over the model with the 6 features. For example, in the AUC measure we see 98% accuracy in the model that uses 18 features and 97% accuracy in the model of the 6 features, while in the average F-measure measure we see 97% accuracy in the model that uses 18 Features and 95% accuracy in the model of the 6 features.

The next step of the researchers was to develop a mobile application based on the 6 features chosen by the model. In the app, the physician enters a total of 6 parameters and receive the patient's survival chances over a period of 5 years in Accuracy, AUC and F-measure all over 95%.


Rectal Cancer

Patient Info

Age at diagnosis 66

Gender: Male

Extension of the Tumor: Submucosa

Tumor Size: Known

Exact size in mm: 36

Surgery of Primary Site: Pull through with sphincter preservation

* Include local tumor excision, polypectomy or excisional biopsy.

Stage: AJCC

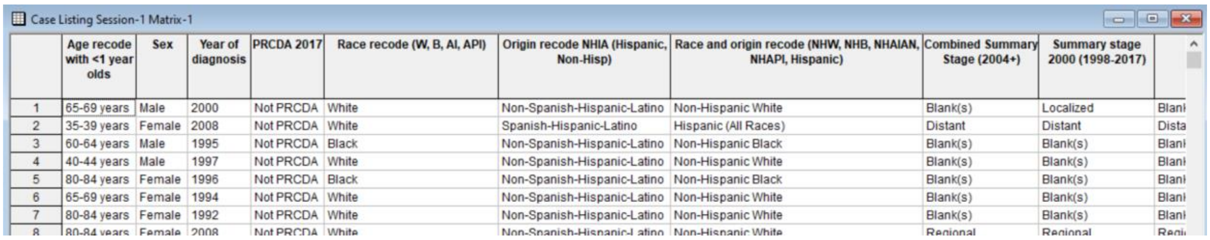
I

figure 2: The doctor enters the parameter in the left screen,
and receive 5-year survival prediction in the right screen

Pre-Processing

Data Pre-processing, or dropping of data before it is used in order to ensure and enhance performance, is an important step in machine learning projects.

Our initial data downloaded from the SEER database contained over 500,000 records and 213 features.



Case Listing Session-1 Matrix-1

	Age recode with <1 year olds	Sex	Year of diagnosis	PRCDA 2017	Race recode (W, B, AI, API)	Origin recode NHIA (Hispanic, Non-Hisp)	Race and origin recode (NHW, NHB, NHAIAN, NHAPI, Hispanic)	Combined Summary Stage (2004+)	Summary stage 2000 (1998-2017)	
1	65-69 years	Male	2000	Not PRCDA	White	Non-Spanish-Hispanic-Latino	Non-Hispanic White	Blank(s)	Localized	Blank
2	35-39 years	Female	2008	Not PRCDA	White	Spanish-Hispanic-Latino	Hispanic (All Races)	Distant	Distant	Dista
3	60-64 years	Male	1995	Not PRCDA	Black	Non-Spanish-Hispanic-Latino	Non-Hispanic Black	Blank(s)	Blank(s)	Blank
4	40-44 years	Male	1997	Not PRCDA	White	Non-Spanish-Hispanic-Latino	Non-Hispanic White	Blank(s)	Blank(s)	Blank
5	80-84 years	Female	1996	Not PRCDA	Black	Non-Spanish-Hispanic-Latino	Non-Hispanic Black	Blank(s)	Blank(s)	Blank
6	65-69 years	Female	1994	Not PRCDA	White	Non-Spanish-Hispanic-Latino	Non-Hispanic White	Blank(s)	Blank(s)	Blank
7	80-84 years	Female	1992	Not PRCDA	White	Non-Spanish-Hispanic-Latino	Non-Hispanic White	Blank(s)	Blank(s)	Blank
8	80-84 years	Female	2008	Not PRCDA	White	Non-Spanish-Hispanic-Latino	Non-Hispanic White	Regional	Regional	Regi

Figure 3 :Example of data as it appears in the SEER application

SEER - stands for Surveillance, Epidemiology, and End Results. It is a program of the US National Cancer Institute. The data includes patient information, such as age, sex, race, etc., as well as information about the cancer, such as the patient's organ in the body, stage at which the cancer was diagnosed, types of treatments, and survival data. The data in the SEER program have been collected since 1973, initially in very few states in the United States, when later additional regions were added. As of 2018, the Database contains data on cancer disease of about 35% of US population.

The main preprocessing stages were:

I. Dropping records with no target value – about 5,000 records

II. Dropping columns that contained single value –75 columns

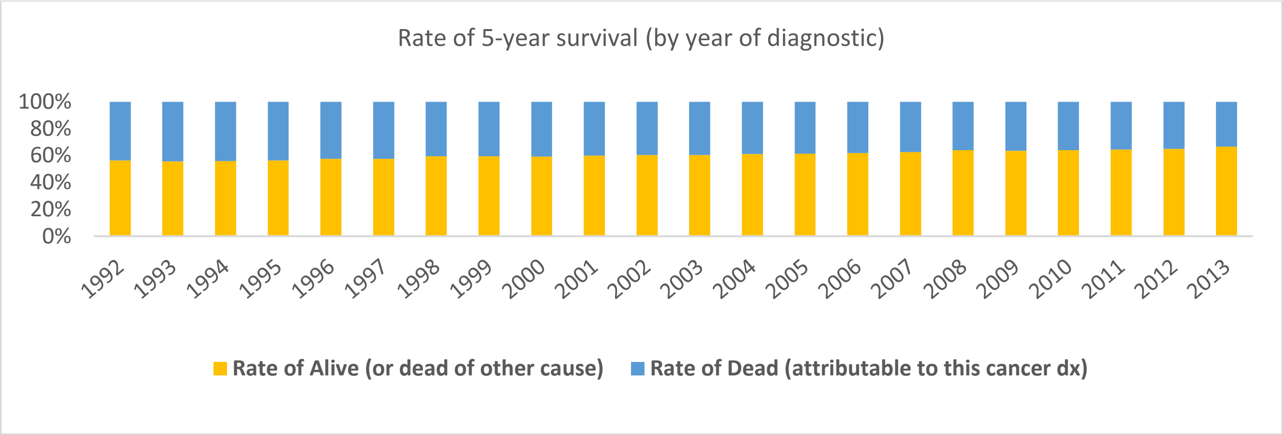
III. Dropping columns that contained the same values in all records – 2 columns

IV. Dropping columns that data was contained in other columns – 2 columns

V. Focusing on the years 2010-2013 and deleting all previous and late records.
We have taken this step for a few reasons. First, our computers failed to properly handle so many records and the processes got stuck or crashed.

Second, we chose 2010 because there were many fields where it began collecting data this year, and we stopped in 2013 because data from later years did not include full information on the 5-year survival patients.

And third, It is more important for us to give tools, for example to doctors, with reference to up-to-date information and not according to what was 30 - 40 years ago. For this purpose, we examined 5-year survival rate over the years and saw a trend of improvement in the percentage of survival as years progress, which makes data from previous years less influential for our purpose.
Activation of this step resulted in a dropping of 425 thousand records.



Graph 6: 5-year survival rate between 1992-2013. trend of improvement as years progress

VI. We took all the columns that at least 30% of the data was missing and performed the hypothesis test, in order not to lose information that could be valuable. In the test:
H_0 - There is no difference between the two populations deleting the column
H_1 - There is a difference between the populations keeping the column
We have performed the test at a confidence level of 99% and through it we dropped 5 columns

VII. Conversion categorical data to ordinal categories

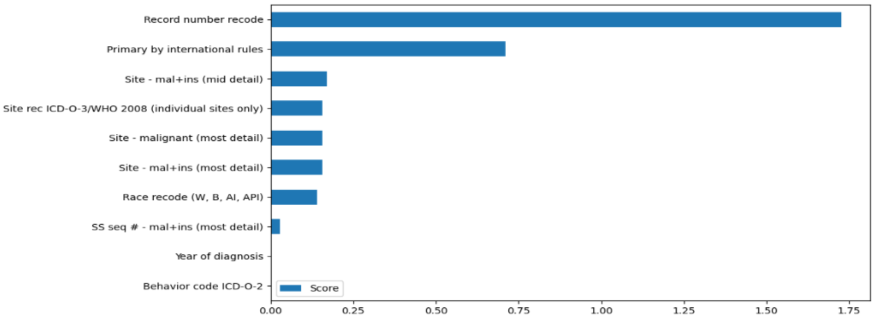
At the end of the pre-processing phase we were left with 60 features and about 70,000 records, which we randomly divided into train and test in a ratio of 80/20.

Feature Selection

In the pre-processing phase we have dropped columns mainly from "technical reasons". In the Feature Selection phase we reduce the number of input variables to both: reduce the computational cost of modelling, and to improve the performance of the model.

Chi-Squared test

At this point we examined three Feature Selection methods. The first, Chi-Squared test, belong to the Filter methods, considering the data type of the input and target variable to evaluate the relationship between them.

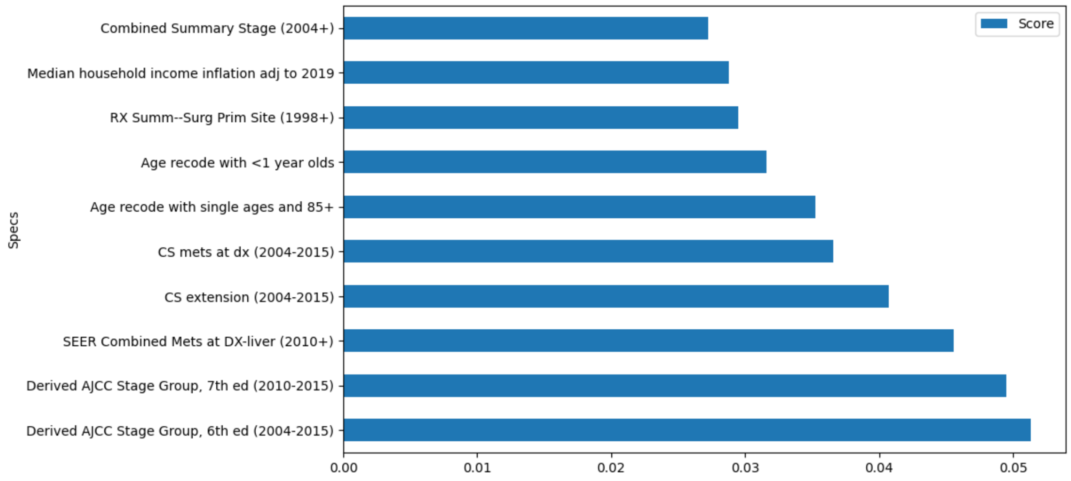


Graph 7: Chi squared test, shows the correlation between the Expending variable and the explained variable

We have ranked the features by the correlation to the target variable and looked at the first 10 features. The first feature to come up is recode-number recode which counts the number of tumors diagnosed.

Decision Tree

After that we have investigated another feature selection method – Decision Tree. DT belong to the wrapper methods - which measure the usefulness of a subset of feature by actually training a model on it. A decision tree ranks the importance of features according to their Entropy - From low to high (The upper node in the tree has the lowest entropy and so on). We have ranked the features by their importance.



Graph 8: Ten most important features according to Decision Tree ranking

The first two most important features deal with the staging of cancer, and there is high correlation between them.

Boruta

The other wrapper feature selection method me performed on the data named Boruta [6]. Boruta is an algorithm that is based on Random Forest but is more robust compare to RF. Random Forest ranks the features and then you decide using a threshold value which features go into the model. But whoever decides on the threshold value - this is an arbitrary decision.

In Boruta, the first step is to build a Random Forest. Then, we build a **shadow column** for each column by randomly shuffling the data of the column. Then, when we will compare the features with the shadow feature, they will not compete with each other for who has the greatest impact, but rather they compete against the shadow columns.

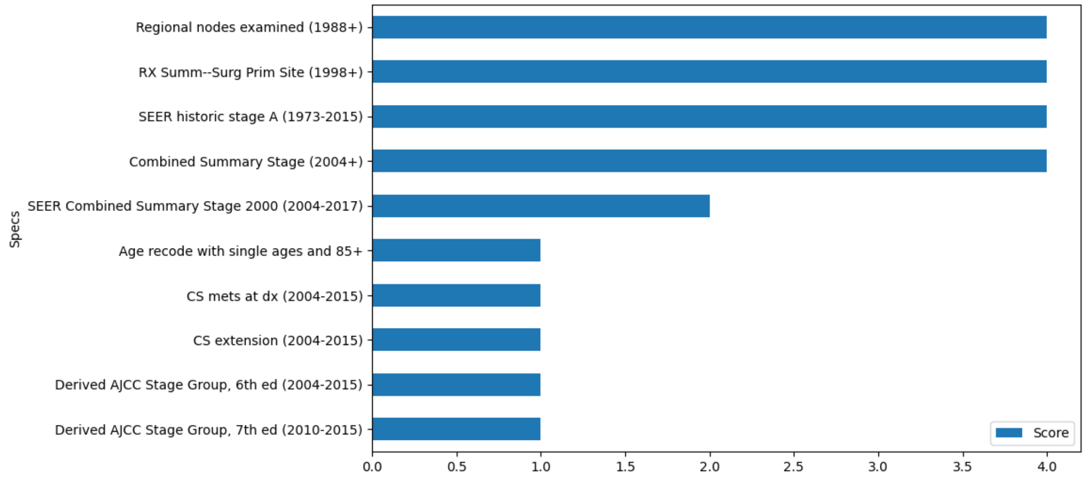
Next we calculate feature importance for each feature using Gini Impurity (measurement of the likelihood of an incorrect classification of a new instance of a random variable), and compare all original columns against a threshold value that we do not arbitrarily set, but the threshold value is the shadow column that gives the best prediction - has the highest Gini

Impurity index. Then compare the feature importance of each original feature against the threshold value. If the feature importance of the original column is higher than the threshold of the shadow column, we will keep the feature, and if the threshold is lower, we will drop it.

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

Formula 1: Pi denotes the probability of an element being classified for a distinct class.

As happens many times in ML, to get more reliable results this process need to repeated, so we ran it for 100 iterations. Then how do we get a decision criterion? Let's take a feature that we have no idea if it is useful to us or not. What is the probability that we will keep it? The maximum uncertainty whether we leave it or not is 50% (like tossing a coin) because each iteration, each experiment is independent, i.e. gives a binary result, a series of N attempts is a binomial probability, then the result looks like a bell graph (when calculating Probability Mass Function).



Graph 9: Ten most important features according to Boruta algorithm

In Boruta there is no strict criterion of rejection and acceptance zone. Instead, the bell divided in to **slots** and the algorithm sets the feature in to these slots. The features with the lowest probability are located in the left tail of the bell graph. They defined as noise and dropped from the model, the features with the highest probability are located in the right tail of the bell graph and sure to be kept. The rest of the features in the middle area - Boruta method are undecided about them, but for conservative reasons we decided to keep them.

After running the 3 methods on the data, we were uncertain how to implement feature selection phase. Whether to vote between the methods, or perhaps run them again consecutively, and at each stage (after each method) drop number of features. We decided to put the results of each method of the feature selection in a table, and examine the results.

Feature	Boruta	Chi2	DT
Combined Summary Stage (2004+)	1	35	4
Derived AJCC Stage Group, 7th ed (2010-2015)	1	48	2
Derived AJCC Stage Group, 6th ed (2004-2015)	1	47	1
CS extension (2004-2015)	1	50	3
Age recode with single ages and 85+	1	39	6
SEER historic stage A (1973-2015)	2	36	8
RX Summ--Surg Prim Site (1998+)	2	46	11
CS mets at dx (2004-2015)	2	52	5
Regional nodes examined (1988+)	4	28	12
Age recode with <1 year olds	5	30	7
Derived AJCC T, 6th ed (2004-2015)	5	40	15

Looking at the results it can be seen that the chi squared test results are very different from the other two methods. This is because chi squared examines the correlation between each attribute and the target variable and does not address the correlation between the features themselves. In choosing between a decision tree and a Boruta algorithm, we chose to use only Boruta. The Boruta is a more robust algorithm, That his running includes many iterations on the data (while a Decision Tree only run once) and its results are more reliable. After running the Boruta we have left with 30 features.

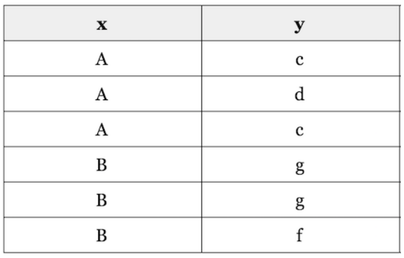
Theil's U

At the next phase of the pipeline, we were looking for better way to measure correlation between categorical features, so we can reduce some of them.

The first method we thought of was Cramer's - Cramer's is a two-way method - Pearson's chi-squared statistic. In Kramer there is symmetry in the correlation between two variables, i.e. the intensity of the relationship between A and B is equal to the intensity of the relationship between B and A.

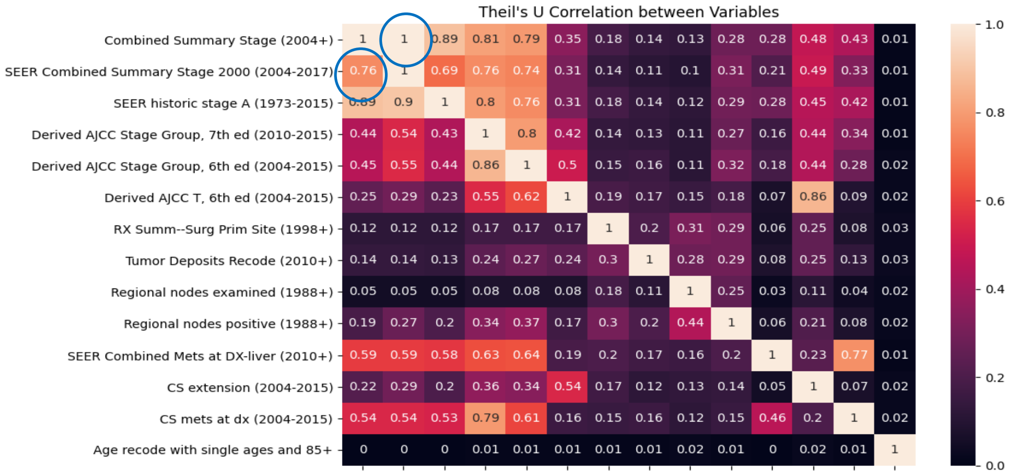
But in reality, sometimes a first variable explains a second variable, but the second variable does not explain the first variable, and if we use a symmetric algorithm we may lose information. So we chose to use a method named Theil's U [7].

Let's examine Theil's U algorithm using the following simple dataset. We can see that if the value of X is known, we do not know for sure the value of Y is. But if we know the value of Y, then surely we know the value of X. in Cramer's ,This valuable information is lost due to the symmetry of it. This is where the Theil's U method comes to our aid. The advantage of Theil's U is that it is not a symmetrical method and it takes into account separately the effect of each variable on the other variable - the method is based on the calculation of **Conditional Entropy** between two variables. In other words, given the value of variable X, how many possible states does variable y have and how often do these situations occur. Similar to Cramer's, the result of the algorithm is between 0 and 1



x	y
A	c
A	d
A	c
B	g
B	g
B	f

where 0 means there is no correlation and 1 means there is complete correlation, however its advantage over Cramer's is that it provides more information about the correlation between two variables.



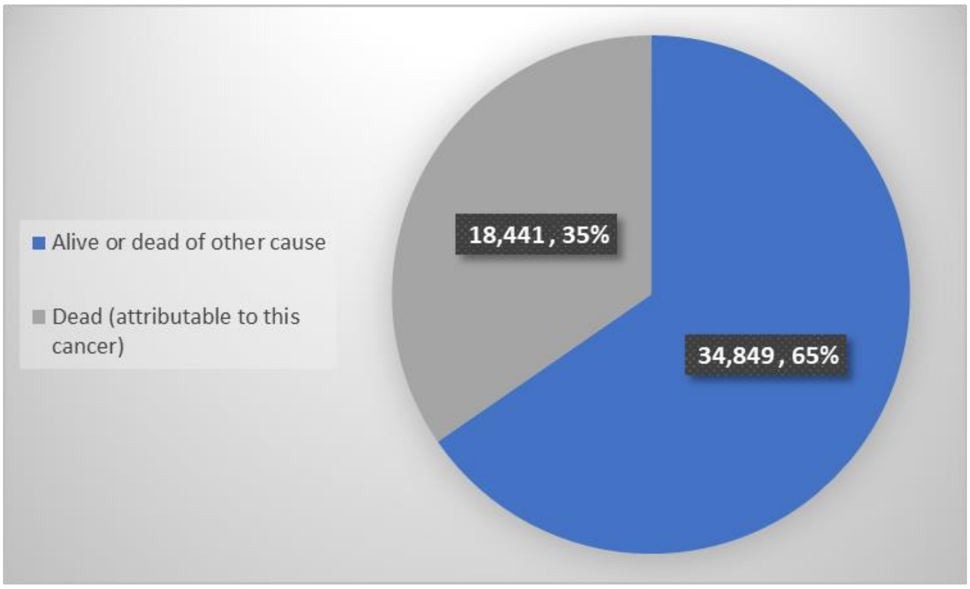
Graph 10: when running Theil's U algorithm on the dataset we can see the a-symmetry: in the top left there are 2 variables with correlation of 1.00, while on opposite side the correlation is 0.76. we do not see the symmetry of the correlation - which shows correlation is mostly in one direction

We used a threshold of 0.8, and with the help of Tiles-U we dropped 3 more features. We reached the models running stage with the amount of 15 features.

Evaluation metrics

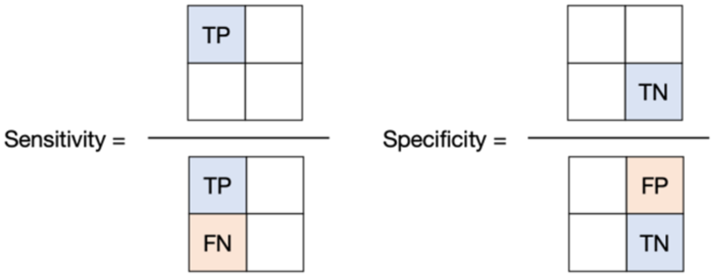
The right choice of critical evaluation indices for the project, has a direct impact on the results of the study.

To examine which metrics are most appropriate requires first, understanding the data and how our target values are distributed: 65% of patients survive. So, without any model, just by systematically predict "alive" (class 1) we have an accuracy of 65%.



For this reason, we decide to combine two metrics:

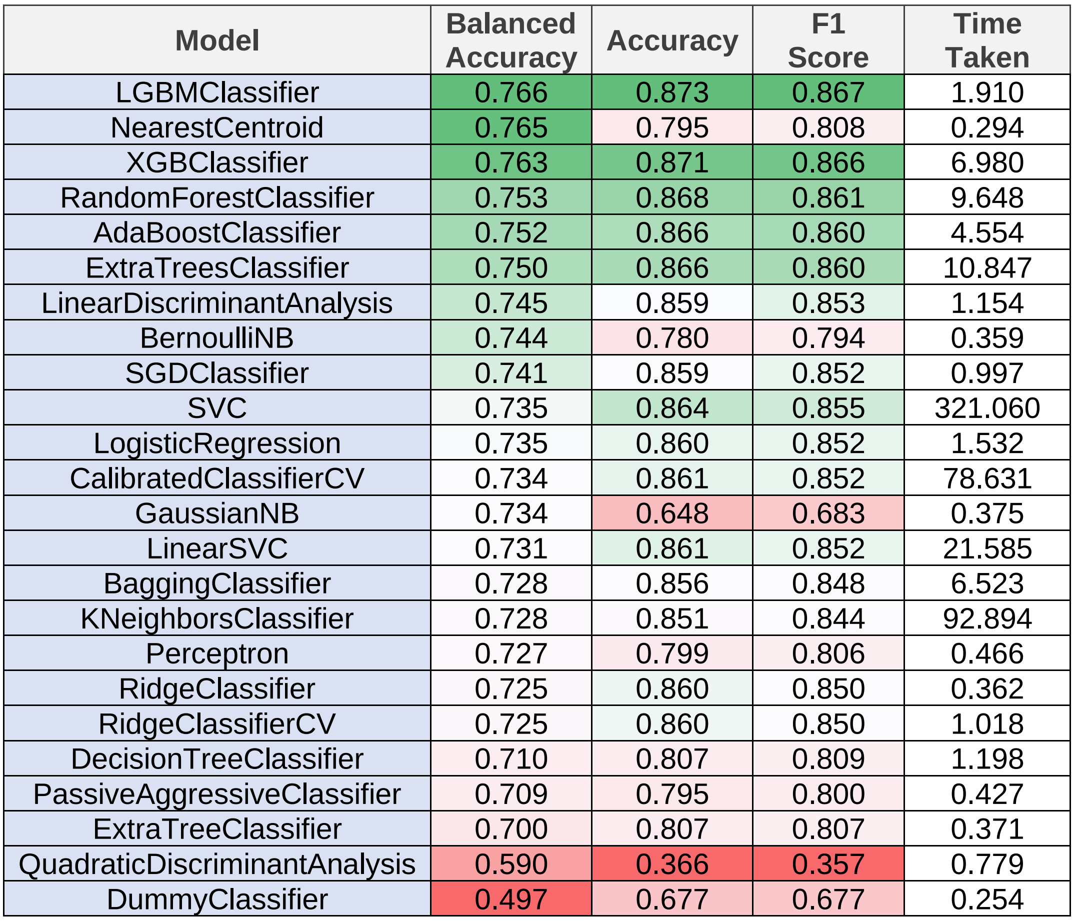
1. Sensitivity: the ability of a model to correctly predict which patients will survive.
2. Specificity: the ability of a model to correctly predict which patients will not survive



For our study there is not preference between False Positive and False Negative, so we decide to use balanced accuracy that is average between specificity and sensitivity.
It's important to understand that maximization of balanced accuracy can lead to classic accuracy decrease.

Model selection

In order to have the best results and to do a first sorting we decided to test all the models using the **Lazypredict package** which allows to activate all the SKLearn models and return a data frame with the models and their scores including the metric that interests us: Balanced accuracy.



Model	Balanced Accuracy	Accuracy	F1 Score	Time Taken
LGBMClassifier	0.766	0.873	0.867	1.910
NearestCentroid	0.765	0.795	0.808	0.294
XGBClassifier	0.763	0.871	0.866	6.980
RandomForestClassifier	0.753	0.868	0.861	9.648
AdaBoostClassifier	0.752	0.866	0.860	4.554
ExtraTreesClassifier	0.750	0.866	0.860	10.847
LinearDiscriminantAnalysis	0.745	0.859	0.853	1.154
BernoulliNB	0.744	0.780	0.794	0.359
SGDClassifier	0.741	0.859	0.852	0.997
SVC	0.735	0.864	0.855	321.060
LogisticRegression	0.735	0.860	0.852	1.532
CalibratedClassifierCV	0.734	0.861	0.852	78.631
GaussianNB	0.734	0.648	0.683	0.375
LinearSVC	0.731	0.861	0.852	21.585
BaggingClassifier	0.728	0.856	0.848	6.523
KNeighborsClassifier	0.728	0.851	0.844	92.894
Perceptron	0.727	0.799	0.806	0.466
RidgeClassifier	0.725	0.860	0.850	0.362
RidgeClassifierCV	0.725	0.860	0.850	1.018
DecisionTreeClassifier	0.710	0.807	0.809	1.198
PassiveAggressiveClassifier	0.709	0.795	0.800	0.427
ExtraTreeClassifier	0.700	0.807	0.807	0.371
QuadraticDiscriminantAnalysis	0.590	0.366	0.357	0.779
DummyClassifier	0.497	0.677	0.677	0.254

Figure 11: Metric results of 24 Lazypredict algorithm

The first results show us a fairly large advantage for tree-based models especially boosting type models.

From the results we also understand that the interaction of features is more significant than their direct influence on patient survival.

This also explains the significant differences between the features selected with the filter method (Chi square) and the wrapper method (Boruta) from the feature selection stage.

Finally, we decide to use the Light Gradient Boosted model. First of all, LGB model gives the best-balanced accuracy and in addition it's very fast.

The specialty of Light GBM is that it grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithms grow level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm [8].

Below diagrams explain the implementation of LightGBM and other boosting algorithms [9].

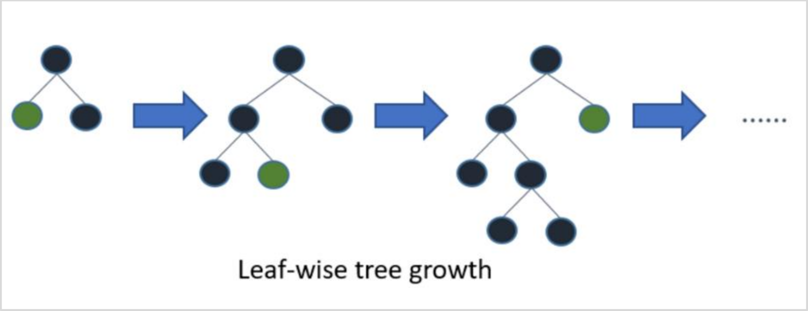


Figure 12: Explains how LightGBM works

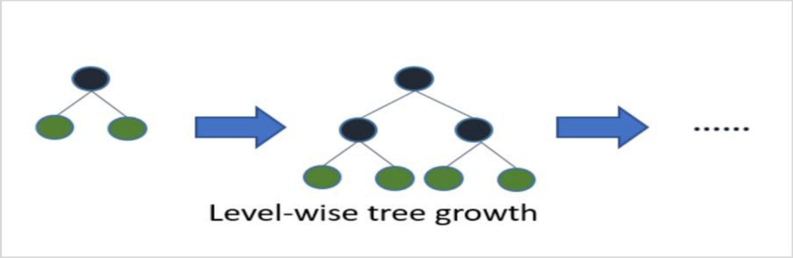


Figure 13: Explain how other boosting algorithm works

Hyperparameters Tuning

It's important to know that Light GBM is sensitive to overfitting and can easily overfit small data.

Light GBM covers more than 100 parameters [10].

The Most influential parameters:

1. max_depth: It describes the maximum depth of tree. This parameter is used to handle model overfitting. Any time you feel that your model is overfitted, my first advice will be to lower max_depth.
2. min_data_in_leaf: It is the minimum number of the records a leaf may have. The default value is 20, optimum value. It is also used to deal over fitting
3. feature_fraction: Used when your boosting is random forest. 0.8 feature fraction means LightGBM will select 80% of parameters randomly in each iteration for building trees.
4. bagging_fraction: specifies the fraction of data to be used for each iteration and is generally used to speed up the training and avoid overfitting.
5. early_stopping_round: This parameter can help you speed up your analysis. Model will stop training if one metric of one validation data doesn’t improve in last early_stopping_round rounds. This will reduce excessive iterations.
6. lambda: lambda specifies regularization. Typical value ranges from 0 to 1.
7. min_gain_to_split: This parameter will describe the minimum gain to make a split. It can used to control number of useful splits in tree.

8. max_cat_group: When the number of category is large, finding the split point on it is easily over-fitting. So LightGBM merges them into ‘max_cat_group’ groups, and finds the split points on the group boundaries, default:64
9. num_boost_round: Number of boosting iterations, typically 100+
10. learning_rate: This determines the impact of each tree on the final outcome. GBM works by starting with an initial estimate which is updated using the output of each tree. The learning parameter controls the magnitude of this change in the estimates. Typical values: 0.1, 0.001, 0.003…
11. num_leaves: number of leaves in full tree, default: 31
12. max_bin: it denotes the maximum number of bins that feature value will bucket in

Bayesian optimization

For Hyperparameters tuning [11] we ran a Bayesian optimization.

There were two principal alternatives to Bayesian optimization: search grid or random grid.

Search grid is a "brute force" algorithm that check all the possibilities, the problem is the running time, especially for LGBM that has a lot of parameters, with just 7 parameters and just 7 different values we need to check 7^10 states (in our case we have much more) it can take years to run.

Random grid solves the runtime problem since it tests randomly only part of the possibilities but on the other hand it does not give good results, far from to the optimum.

Bayesian optimization is particularly advantageous for complex problems that are difficult to evaluate like black box models with some unknown structure.

Since the objective function is unknown, the Bayesian strategy is to treat it as a random function and place a prior over it unlike random grid. The prior captures beliefs about the behaviour of the function. After gathering the function evaluations, which are treated as data, the prior is updated to form the posterior distribution over the objective function. The posterior distribution, in turn, is used to construct an acquisition function (often also referred to as infill sampling criteria) that determines the next query point.

There are several methods used to define the prior/posterior distribution over the objective function. The most common two methods use Gaussian Processes in a method called Kriging. Another less expensive method uses the Parzen-Tree Estimator to construct two distributions for 'high' and 'low' points, and then finds the location that maximizes the expected improvement, like in our model.

For the optimization we used the Optuna package, with a total of 200 trials. Finally, we got:

Trial 113 finished with value: 0.78 and parameters:

```
'lambda_l1': 2.72e-05
'lambda_l2': 3.81
'num_leaves': 19
'feature_fraction': 0.66
'bagging_fraction': 0.57
'bagging_freq': 7
'min_child_samples': 24
```

Backward stepwise selection

After all the parameters optimization, we wanted to check what happens when we take off features from the model, if supplementary dimension reduction will affect the results.

It is important to remember that the main objective of this study is to better understand what influences the survival of patients, and we are ready to take off a feature if this has little influence on the chances of survival.

In order to reduce dimension, instead of lowering one feature at a time, we preferred to lower the threshold of the Boruta, which means that each time we lower the bar on the Gaussian bell.

We then sought a local maximum that would meet the required accuracy requirements while reducing the dimensions (accuracy / interpretability tradeoff) [12].

5	10	15	20	25	30	35	40	rank
80%	81%	80%	81%	81%	81%	81%	82%	A
76%	77%	77%	78%	77%	77%	78%	78%	BA
8	12	15	17	20	26	30	37	K

Rank = threshold (there are 50 parts of the Gaussian bell)

A = accuracy

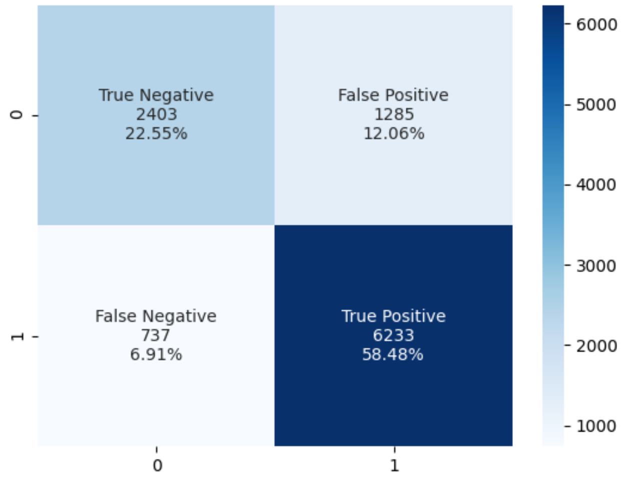
BA = Balanced accuracy

K = number of features

Finally, we got that the optimum regarding the dilemma between accuracy and interpretability is 12 features.

Results:

Due to the inequality in the probability of surviving, in order to evaluate the results, it was important to examine the confusion matrix:



From the confusion matrix we can observe that there is a large proportion, almost two thirds of errors that are false positive (predict that they will survive but die).

Sensitivity = 0.894

Specificity = 0.651

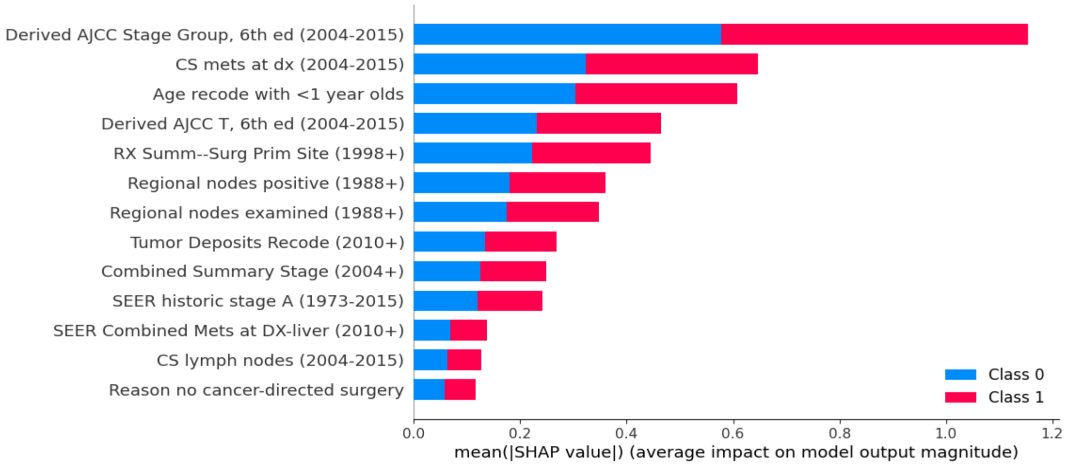
Also we can see from sensitivity and specificity that it's more difficult for the model to predict people that will die cause of the cancer.

Interpretation

To increase model interpretability, we used SHAP (SHapley Additive exPlanations) value.

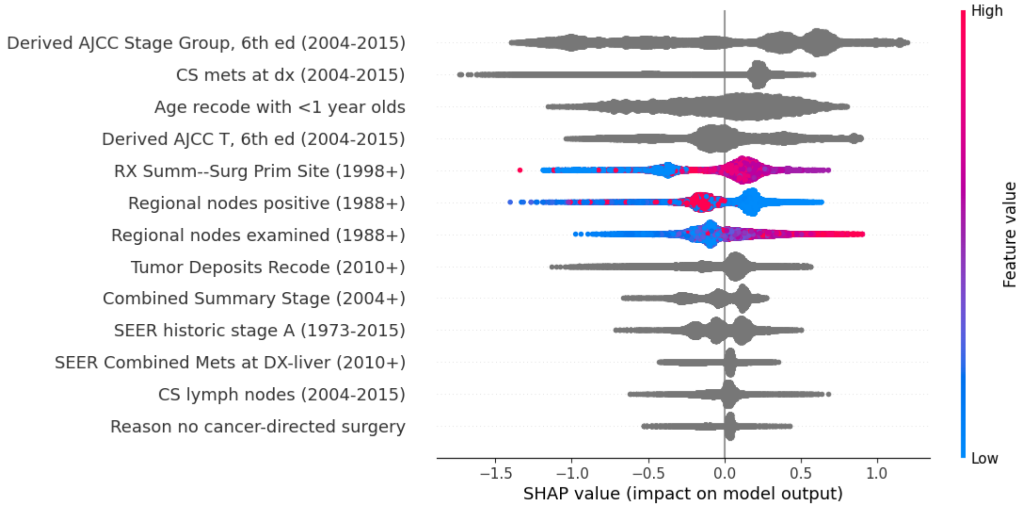
SHAP values are based on Shapley values, a concept coming from game theory.
SHAP has the particularity of reverse-engineering the output of any predictive algorithm by removing single feature and check the average of the marginal contributions across all permutations [13].

We first check the global influence of each feature on all the predictions by the summary plot when class 1 is survivor, 0 is dead.



From the plot we can observe that all features influence almost symmetrically on both classes.

We can also look at The SHAP Variable Importance Plot, but the majority of the features are categorical, so this graph is less informative:



In contrast to neural networks, it can be seen with the help of force plot why an individual observation was classified in one way or another according to its data.

Discussion

Initially the results did not really surprise us, for example it is obvious that the size and the dispersion have an influence on the severity of the disease and therefore on the mortality.

An interesting fact is that unlike previous studies we see that gender and race do not influence mortality, and from the beginning, the inequality of patient types relative to the population we thought these data were problematic.

The program, unlike doctors from the study, has noticed automatically that asking for the number of negative regional nodes was useless knowing that we already had the number examined and the number of positive.

Another interesting supplement of our model is that the presence of metastases (by expansion) to the liver has an additional on mortality, we discovered this perhaps thanks to the approach we took that saved initial data columns despite a large number of missing values.

Conclusion and directions for future research

In this study we have tried different approaches in the hope of finding out new insights on the disease mortality, with the use of innovative and relatively new methods.

we can say that our objective has been partially achieved: on the one hand we have found new factors that influence on the mortality of the disease, but on the other hand all are based on calculation and algorithm without any possibility of combining with experience and real knowledge in the field, the support and advice of an expert would have been very beneficial to the research.

The biggest improvement point of the research is feature engineering, with an in-depth look at each feature.

References

[1] Inés Mármol 1,Cristina Sánchez-de-Diego 1,2,Alberto Pradilla Dieste 1,Elena Cerrada. Colorectal Carcinoma: A General Overview and Future Perspectives in Colorectal Cancer. International Journal of Molecular Sciences January 2017

[2] Fazeli MS, Keramati MR. Rectal cancer: a review. Med J Islam Repub Iran. 2015;29:171. Published 2015 Jan 31.

[3] SEER Cancer Stat Facts: Colorectal Cancer

[4] Lee Y-C, Lee Y-L, Chuang J-P, Lee J-C (2013) Differences in Survival between Colon and Rectal Cancer from SEER Data. PLoS ONE 8(11): e78709.

[5] Oliveira T, Silva A, Satoh K, Julian V, Leão P, Novais P. Survivability Prediction of Colorectal Cancer Patients: A System with Evolving Features for Continuous Improvement. Sensors (Basel). 2018;18(9):2983. Published 2018 Sep 6. doi:10.3390/s18092983

[6] Kursa and Rudnicki, 2010 M.B. Kursa, W.R. Rudnicki Feature Selection with the Boruta Package**.** J. Stat. Softw., 36 (2010), pp. 1-13

[7] The search for Categorical Correlation

[8] G Ke, Q Meng, T Finley, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, 2017

[9] Microsoft documentation, https://github.com/Microsoft/LightGBM

[10] J. Bergstra, Algorithms for Hyper-Parameter Optimization, 2011

[11] Gang Luo, A review of automatic selection methods for machine learning algorithms and hyper-parameter values, 2016

[12] Samuele Mazzanti, Boruta Explained Exactly How You Wished Someone Explained to You, 2020

[13] S. Lundberg, S Lee, A Unified Approach to Interpreting Model Predictions, 2017