University of Granada

Business Intelligence

Practice 1: Resolution of classification problems and experimental analysis

Author:
Mikhail Raudin
Practice Group 1
mraudin@correo.ugr.es

Lecturer:
Dr. Daniel Molina
dmolinac@ugr.es

November 3, 2020

Contents

List of Figures

1 Introduction

The goal of this practice is to analyze the performance of five different classification models on the Mammographic Mass dataset[1]. The classification models are

- K-Nearest Neighbour
- Decision Tree
- Support Vector Machine
- Logistic Regression
- Random Forrest

The feature that is being predicted is the type of tumor (benign or malignant). The models are trained on the rest of the features from the dataset, which are

- BI-RADS
- Age of patient
- Shape of the abnormal mass detected
- Mass margin
- Tumor mass density

The results show a superiority of the Random Forrest Classifier, with an accuracy of 84,53% and more significantly, a recall of 83,41%.

2 Data Processing

The data of the dataset will be described, as well as the techniques being used for handling corrupt data.

2.1 The Data

The datasets contains 5 features from which we try to obtain knowledge about the severity of the tumor. It has a total of 961 rows. As it can be seen in fig. 1, the dataset contains missing values, marked with a "?". Fig. 1 also shows the amount of missing values for each feature. We see, that the margin and density clearly dominate this weak point of our dataset, and therefore techniques to handle this problems will be analyzed, in order to enhance the predictions.

[1] https://bigml.com/user/TotyB/gallery/dataset/509a98c6035d0706dd0001dd

	BI-RADS	Age	Shape	Margin	Density	Severity
0	5	67	L	5	3	maligno
1	4	43	R	1	?	maligno
2	5	58	I	5	3	maligno
3	4	28	R	1	3	benigno
4	5	74	R	5	?	maligno

```
BI-RADS      2
Age          5
Shape        0
Margin      48
Density     76
Severity     0
```

Figure 1: the head of the dataframe (left) and the amount of missing values for each feature (right)

2.2 Handling missing values

The easiest and most straightforward way is just deleting all the rows that contain a missing value. When we apply it on the dataset, it leaves us with 847 rows, so we lose 114 rows, which is not inherently a bad number. This is the dataset we will train our classification models on, in order to see if restoring the missing values later leads to a significant improvement of the evaluation results, wich is part of the analysis.

Due to the fact, that we don't have a lot of data (961 samples is not considered as a large number in data science), it is useful to apply some data preprocessing techniques to handle the missing values. A popular technique is data imputing, which exchanges all the missing values with a specific numerical value. The numerical value that replaces the missing values is estimated with statistical methods, using the present values of the column in the dataset. We apply each of the following methods on the whole dataset to replace missing values with

- the mean of the column,
- the median of the column,
- and the most frequent value of the column.

Fig.2 shows this values for our features.

Feature	mean	median	most_frequent
BI-RAIDS	3.294484911550468	3.0	3
Age	38.276795005202914	40.0	42
Density	2.6805411030176898	3.0	3
Margin	2.6566077003121746	3.0	1

Figure 2: Shows the statistical values for each feature column.

3 Algorithm Configuration

This section describes the configuration of each classification algorithm, including the hyperparameter tuning. Each time a hyperparameter is tuned, a grid search is used to find the optimal combination of parameters, evaluating each combinations performance with a cross validation over 5 folds. The optimal combination is ranked by mean test accuracy and by mean test recall.

3.1 K Nearest Neighbor

The K-Nearest Neighbor algorithm is a simple yet effective machine learning algorithm that exists since 1979. It puts new data into the category that is most similar to the new data, by observing the categories of the k most similar data points in the dataset. So the most important parameter to analyze is K, the number of neighbors the algorithm takes into account making a decision.

We run the algorithm on 5 different partitions of the dataset using the cross-validation techniques for K=1,..,31. Maximizing the accuracy the best value for K is 5 or 15 (the difference occurs in the fifth digit), because our dataset is not very large it does not matter which of the two values we choose, but if computational time is relevant, a large value of K leads to slower classification. The disadvantage of a small value for K is noise due to outliers, but for K=5 it works well on our data. The graph in fig. 3 shows the changes of the accuracy when K rises from 1 to 31. We can see that the results don't change a lot and

only when K=1 the standard deviation of the accuracy over the 5 folds is very high, probably due to partitions with outliers.

Table 1: This table shows the top 5 mean accuracy's

k	mean test accuracy
5	80,17%
15	80,17%
10	79,93%
11	79,93%
16	79,92%

Additionally, another run is made by comparing the mean recall achieved with different values of K. In the end, and as further described in (Section 5) this is the most important score to improve. The graph in fig. 4 shows that the recall improves for higher values of K. Not only the recall score, but also the standard deviation over the 5 folds gets better for higher K as well.

model with K=27 is selected as the best model, in order to compare it with the other algorithms in the following sections.

change a lot and only when K=1 the standard deviation of the accuracy over the 5 folds is very high, probably due to partitions with outliers.

Table 2: This table shows the top 5 mean recall values

k	mean test recall
27	85,85%
29	85,37%
28	85,12%
30	84,88%
25	84,88%

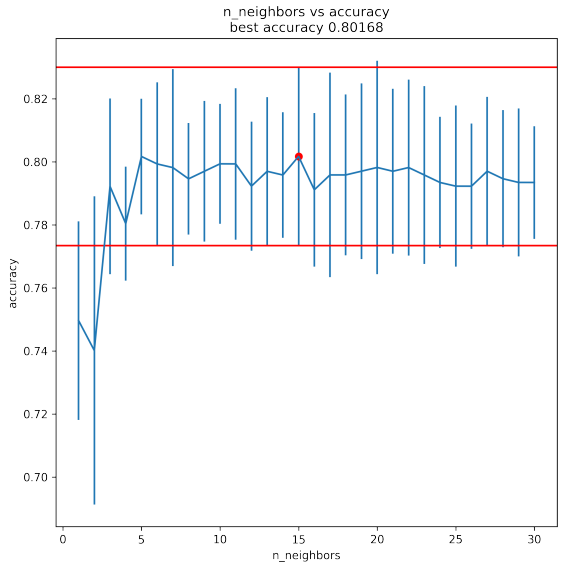

Figure 3: Shows the best accuracy score for every k. The blue vertical lines show the standard deviation over the 5 folds, and the red horizontal lines indicate the best standard deviation achieved.

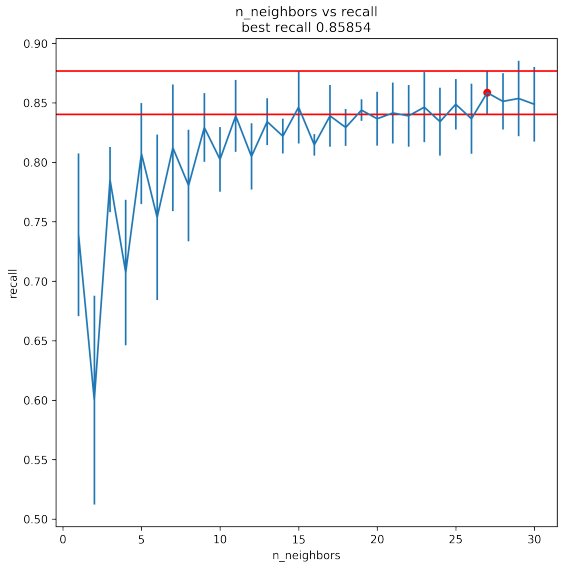

Figure 4: Shows the best recall value for every k. the blue vertical lines show the standard deviation over the 5 folds, and the red horizontal lines indicate the best standard deviation achieved.

3.2 Decision Tree

The main parameters we focus on is the function of the split, which can be "gini" for the Gini impurity and "entropy" for the information gain. The other parameter is the depth of the tree, to see if a more complex tree provides better results. But the opposite is the case, as it can be seen in fig. 5

Table 3: This table shows the top 5 function and depth combination

parameters	mean test accuracy
criterion: entropy, max depth: 3	84,06%
criterion: gini, max depth: 3	83,94%
criterion: gini, max depth: 4	82,17%
criterion: entropy, max depth: 5	81,70%
criterion: entropy, max depth: 4	81,58%

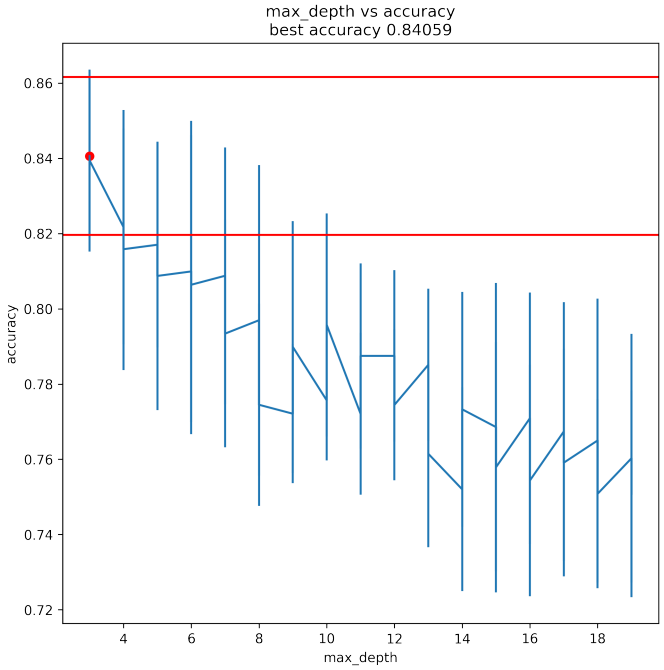

Figure 5: The graph shows the best accuracy value for every tree depth. The blue vertical lines show the standard deviation over the 5 folds, and the red horizontal lines indicate the standard deviation of the best value.

3.3 Support Vector Machine

A Support Vector Machine (SVM) with a linear kernel and the default settings will be used, for example, the regularization parameter C is 1. This configuration leads to a good model, especially in terms of predicting the benign tumor (class 0), with the only weak point being the recall of the malignant tumor (class 1). To see if this weak point can be improved, another kernel is tested, the non-linear rbf kernel. The non-linear rbf kernel has a very good recall value, but the price of this are worse scores in nearly all the other evaluation measurements (Section 5). A short comparison of the two configurations provides the following table:

Table 4: This table compares two different SVM kernels

kernel	mean accuracy	mean recall
linear	85,88%	72,5%
rbf	80,05%	85,85%

3.4 Logistic Regression

The default configuration in sklearn is used and no hyperparameters are studied. The penalty is l2 and the inverse regularization strength C is equal to 1.

3.5 Random Forrest

The Random Forrest Algorithm has a lot of different hyperparameters that can be tuned. The focus was set on three important parameters,

- maximum depth: the longest path between the root node and the leaf node in every tree of the forest. No tree will grow larger then this value
- number of estimators: the number of trees in the forest
- maximum features: the number of features to consider every time a tree tries to find the best split for a node
- criterion function: this parameter is also about finding the best split, it defines the function with which we measure the split quality. For classification problems Gini and entropy are the relevant functions. As we saw for the decision tree, a different function can lead to a slightly improvement of the performance (0,12%) so we will tune this parameter again.

The first round of hyperparameter tuning has the following parameter values

- maximum depth: 3,4,5,6,7

- number of estimators: 10,20,30,...,90,100
- maximum features: auto=sqrt(number features) and log2=log2(number features)
- criterion function: gini and entropy

The recall achieved is 82,68% with a maximum depth of 4, 40 estimators, the auto function for the maximum features and gini as a split criterion.

The next round narrows down the parameter values, taking into account that we can likely perform better or equally good with a less complex random forest algorithm, which means less trees in the forest and keeping the depth of each tree small. Therefore the range will be as following:

- maximum depth: 3,4,5
- number of estimators: 1,2,3,4,...,18,19,20
- maximum features: auto=sqrt(number features) and log2=log2(number features)
- criterion function: gini and entropy

This lead to an improvement in the recall performance: 83,41%. The best combinations can be seen in table. The complexity of the model could be decreased significantly from 40 to 3 estimators.

Table 5: This table shows the top 5 parameter combinations for the random forest algorithm

parameters	mean test recall
criterion: gini, max depth: 4, estimators: 3, features: auto	83,41%
criterion: gini, max depth: 4, estimators: 3, features: log2	83,41%
criterion: entropy, max depth: 5, estimators: 2, features: auto	82,68%
criterion: entropy, max depth: 5, estimators: 2, features: log2	82,68%
criterion: entropy, max depth: 4, estimators: 5, features: log2	82,44%

4 Results

In this section each models results (with its best combination of hyper-parameters obtained in the previous section) will be presented using the confusion matrix and ROC Curve. In addition to this, each data preprocessing technique applied on the dataset will be evaluated on every model.

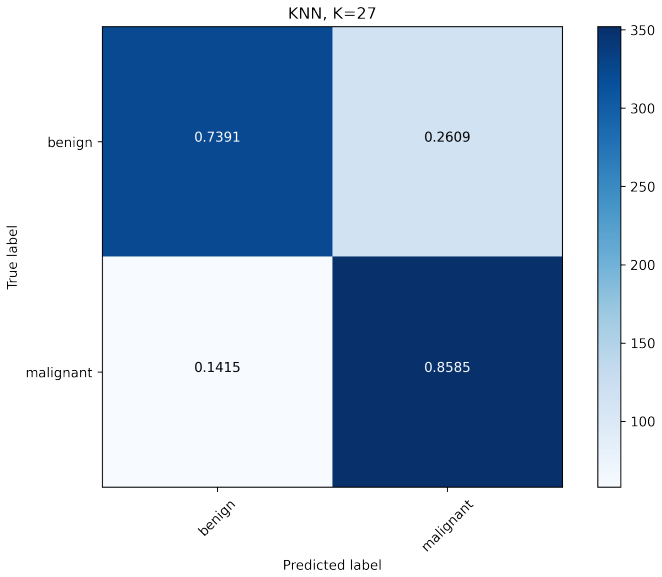

Figure 6: Confusion Matrix

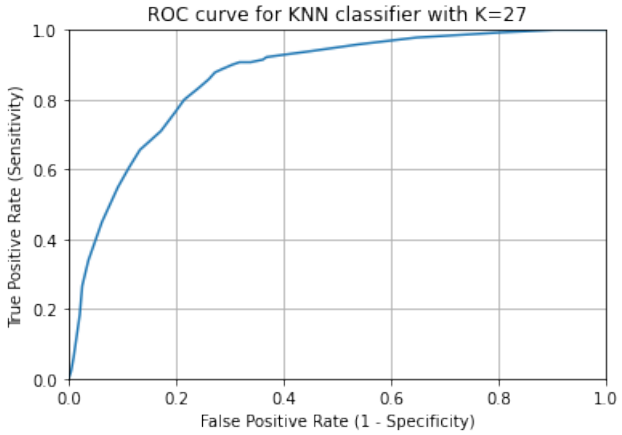

Figure 7: ROC curve

4.1 K-Nearest Neighbor

The Confusion Matrix and ROC curve for KNN can be seen in fig. 6 and fig. 7.

Fig. 8 shows the effect of each preprocessing technique on the evaluation metrics. None of the techniques improves the prediction results.

Evaluation Measure	mean	median	most_frequent	constant
accuracy	0.7378	0.7378	0.7378	0.7940
sensitivity	0.7528	0.7528	0.7528	0.8045
specificity	0.7248	0.7248	0.7248	0.7849
FPR	0.2752	0.2752	0.2752	0.2151
preciscion	0.7023	0.7023	0.7023	0.7633
AUC	0.7733	0.7733	0.7733	0.8568

Figure 8: Performance of KNN after imputing missing values

4.2 Decision Tree

The Confusion Matrix and ROC curve for the decision tree can be seen in fig. 9 and fig. 10.

Fig. 11 shows the effect of each preprocessing technique on the evaluation metrics. None of the techniques improves the prediction results.

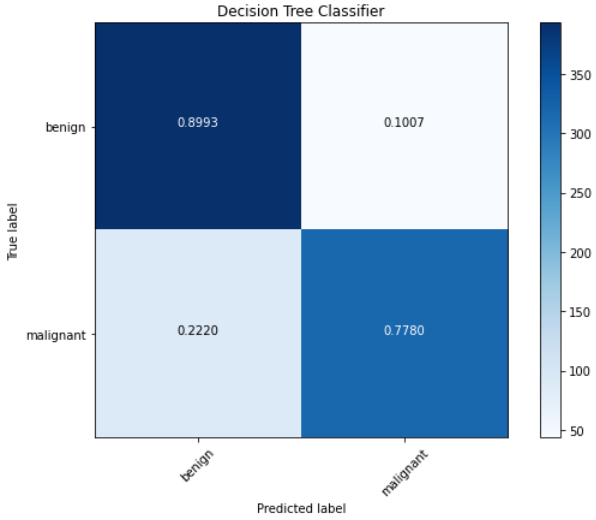

Figure 9: Confusion Matrix

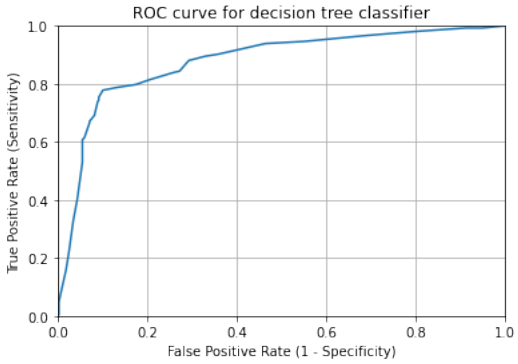

Figure 10: ROC curve

Evaluation Measure	mean	median	most_frequent	constant
accuracy	0.7825	0.7825	0.7825	0.8137
sensitivity	0.8090	0.8090	0.8090	0.7775
specificity	0.7597	0.7597	0.7597	0.8450
FPR	0.2403	0.2403	0.2403	0.1550
preciscion	0.7438	0.7438	0.7438	0.8122
AUC	0.7901	0.7901	0.7901	0.8772

Figure 11: Performance of decision tree after imputing missing values

4.3 Kernel SVM

The Confusion Matrix and ROC curve for the kernel SVM (rbf) can be seen in fig. 12 and fig. 13.

Fig. 14 shows the effect of each preprocessing technique on the evaluation metrics. None of the techniques improves the prediction results.

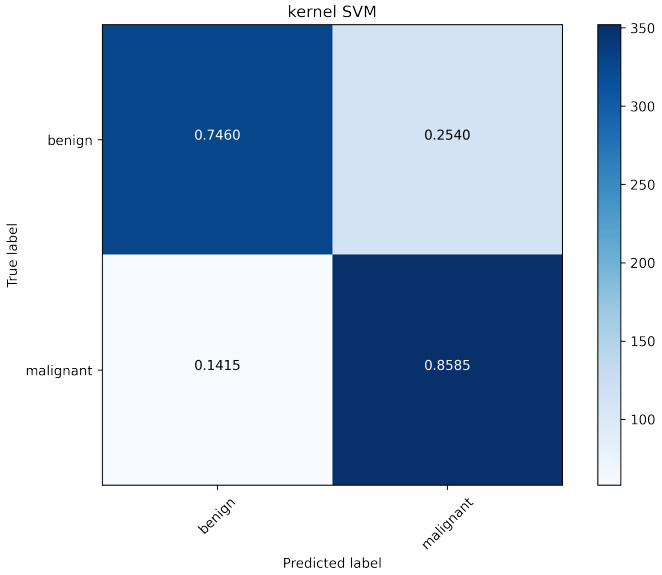

Figure 12: Confusion Matrix

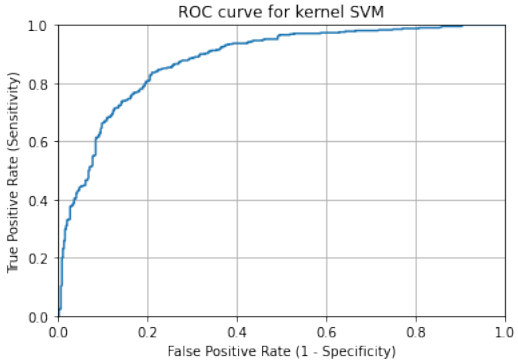

Figure 13: ROC curve

Evaluation Measure	mean	median	most_frequent	constant
accuracy	0.7586	0.7586	0.7586	0.7981
sensitivity	0.7708	0.7708	0.7708	0.8337
specificity	0.7481	0.7481	0.7481	0.7674
FPR	0.2519	0.2519	0.2519	0.2326
preciscion	0.7252	0.7252	0.7252	0.7556
AUC	0.8014	0.8009	0.8012	0.8715

Figure 14: Performance of kernel SVM after imputing missing values

4.4 Logistic Regression

The Confusion Matrix and ROC curve for the Logistic Regression model can be seen in fig. 15 and fig. 16.

Fig. 17 shows the effect of each preprocessing technique on the evaluation metrics. None of the techniques improves the prediction results.

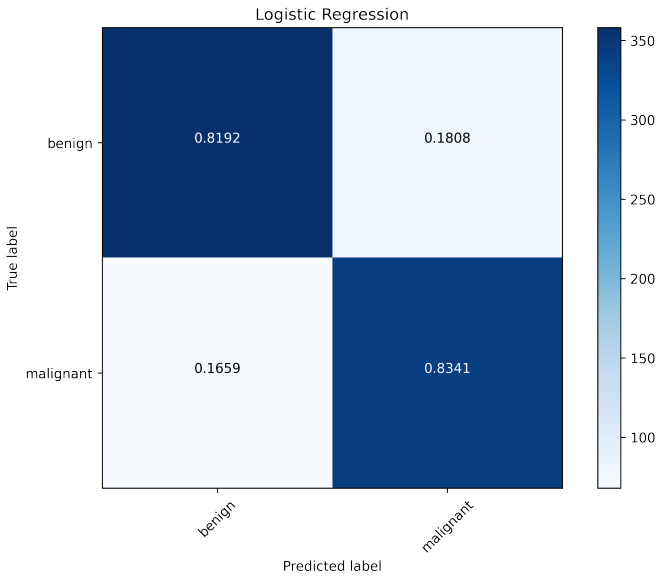

Figure 15: Confusion Matrix

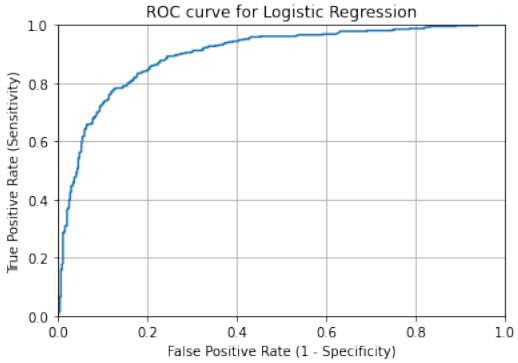

Figure 16: ROC curve

Evaluation Measure	mean	median	most_frequent	constant
accuracy	0.7742	0.7742	0.7742	0.8169
sensitivity	0.7978	0.7978	0.7978	0.8135
specificity	0.7539	0.7539	0.7539	0.8198
FPR	0.2461	0.2461	0.2461	0.1802
preciscion	0.7365	0.7365	0.7365	0.7956
AUC	0.8098	0.8098	0.8098	0.8912

Figure 17: Performance of Logistic Regression after imputing missing values

4.5 Random Forest

The Confusion Matrix and ROC curve for the Random Forest can be seen in fig. 18 and fig. 19.

Fig. 20 shows the effect of each preprocessing technique on the evaluation metrics. None of the techniques improves the prediction results.

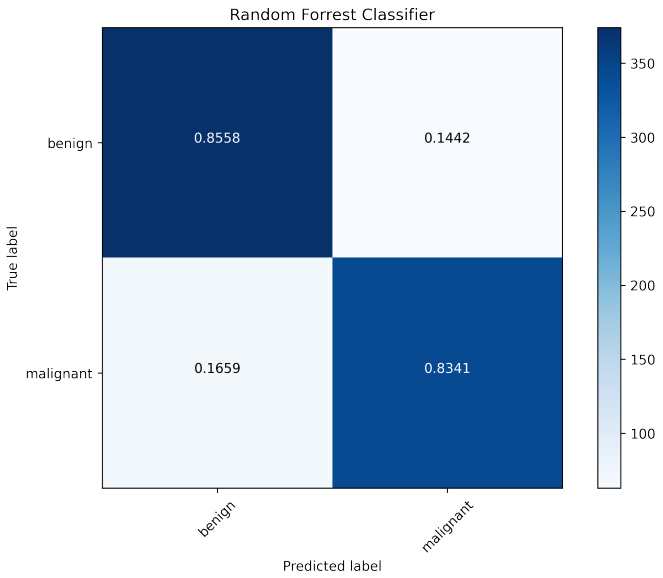

Figure 18: Confusion Matrix

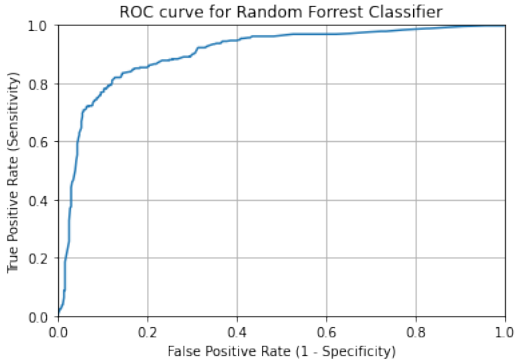

Figure 19: ROC curve

Evaluation Measure	mean	median	most_frequent	constant
accuracy	0.7617	0.7617	0.7617	0.8252
sensitivity	0.7596	0.7596	0.7596	0.8000
specificity	0.7636	0.7636	0.7636	0.8469
FPR	0.2364	0.2364	0.2364	0.1531
preciscion	0.7348	0.7348	0.7348	0.8184
AUC	0.7876	0.7876	0.7876	0.8967

Figure 20: Performance of Random Forest after imputing missing values

5 Analysis of the results

This section compares the results (using the metrics: accuracy, sensitivity, specificity, false positive rate, precision, ROC curve and AUC score) of all models. All evaluations are obtained using scikit-learns "cross val predict" method, which returns the prediction for every data point when it was in a test split. This method is good to compare the predictions between different models. Moreover we discuss the advantages and disadvantages for certain models and give an interpretation of the results.

5.1 Comparing results by metrics

The table with all classifiers and all metrics and a ROC curve with all classifiers can be seen in fig. 21 and fig. 22.

Sensitivity

Due to the fact that we study a medical dataset, the error cost of a false negative classification is much higher then of a false positive classification. A false negative is an instance of a positive class classified as negative, namely, a malignant tumor that remains undetected. On the opposite, a false positive would be a benign tumor classified as a malignant tumor, a negative instance wrongly

Evaluation Measure	KNN, K=27	Decision Tree	Kernel SVM	Logistic Regression	Random Forrest
accuracy	0.7969	0.8406	0.8005	0.8264	0.8453
sensitivity	0.8585	0.7780	0.8585	0.8341	0.8341
specificity	0.7391	0.8993	0.7460	0.8192	0.8558
FPR	0.2609	0.1007	0.2540	0.1808	0.1442
preciscion	0.7554	0.8788	0.7603	0.8124	0.8444
F1 Score	0.8037	0.8254	0.8064	0.8231	0.8393
AUC	0.8649	0.8779	0.8783	0.8982	0.9015

Figure 21: final results for each classifier

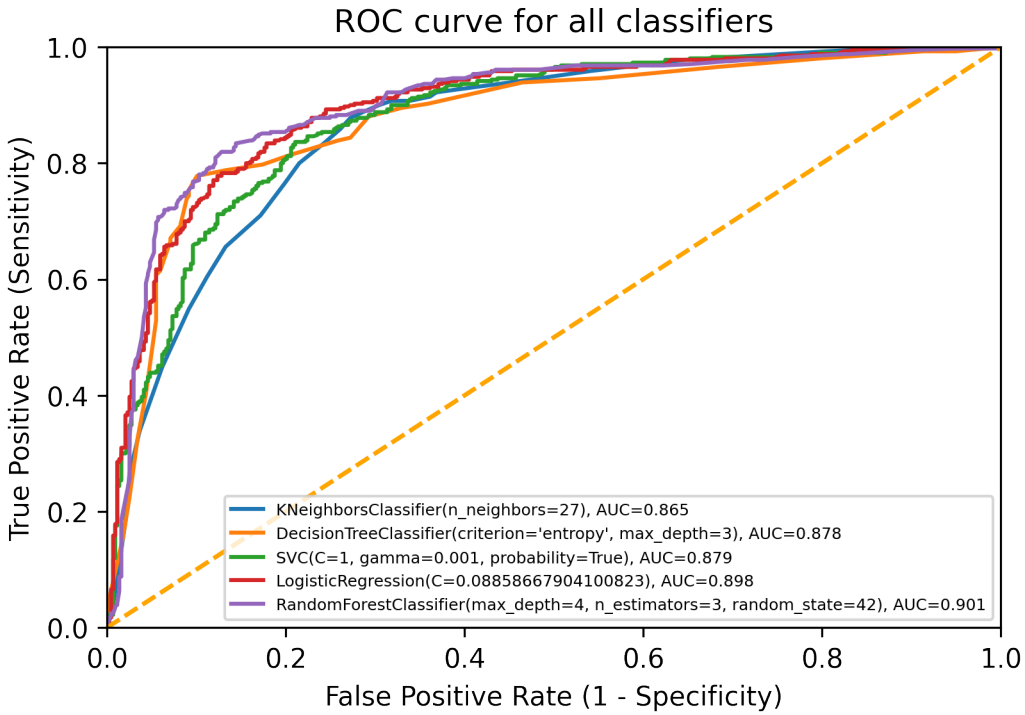

Figure 22: ROC curve with all classifiers

classified as positive. To observe the occurrence of false negatives, we can calculate the sensitivity of each model, that is, how sensitive it is towards detecting positive instances of a class. It is also called True Positive Rate (TPR) or Recall and thus the False Positive Rate is 1-TPR.

We can see in fig. 21 that the KNN and SVM model have both the best sensitivity overall (85,85%), which means that those models detect 85,85% of the malignant tumor patients in the dataset correctly, a good result. We could assume that the malignant class has not many outliers, because the sensitivity went up by just 2,1% when taking into account 27 instead of 5 nearest neighbors in KNN.

We can not conclude that those are the best models yet, because it could be that they predict the other class, benign tumor, poorly. Taking a closer look, those two models indeed have the worst False Positive Rates (FPR), KNN with 26,09% and SVM with 25,4%. This means that the model roughly classifies every fourth patient in the dataset that has a benign tumor as a malignant tumor, which is a high error rate for the benign class.

F1-Score

In order to see how balanced the model performs in terms of false positives and false negatives, the F1-Score is a great metric. It is a weighted average of the models precision and sensitivity. The precision tells how often a positive prediction is correct, compared to all positive predictions. If it is low, it is indicating a high false positive rate. We can see that the KNN and SVM have the lowest precision and F1-Score of all models. We can conclude that the great performance on positive instances of those two models comes with the cost of the worst performance on the negative instances, causing a high False Positive Rate.

ROC Curve

The ROC Curve plots the True Positive Rate (y-axis) against the False Positive Rate (x-axis). The Area Under The Curve (AUC) shows the separability of the classes, how well a model can separate between benign and malignant tumor instances. The Random Forrest has the highest AUC with 90,14%, so it achieves a high sensitivity with a small false negative rate, meanwhile keeping the false positive rate small as well, leading to a great overall performance on predicting both classes correctly. The high false positive rate of KNN leads to the smallest AUC values of all models.

Conclusion

The best model overall by metrics is the Random Forrest Classifier, having also the highest F1 Score 83,93%. By considering only a subset of the features at each step creating a wide variety of trees, it classifies more effectively than any individual decision tree (and in our case more effective then any other model).

The variety of trees in the Random Forest leads to a robust and more accurate classifier, which also indicates the best AUC value out of all models. Because Random Forest work well on categorical and numerical data, it might be the reason for outperforming models like Logistic Regression or SVM which have a great performance if all of the input data is numerical rather then categorical.

Since we focus on keeping the false negative rate as small as possible, the kernel SVM can be seen as the best model, having the highest sensitivity with a slightly better performance on the benign class indicated through a slightly (0,28%) higher F1-Score, a 1,36% higher AUC and a smaller FPR (by 0,69%) then the KNN. Since Random Forest and kernel SVM are both 'black-box' models, they have the ability to capture high non-linearity and interactions between features and generate more accurate classification results.

We can also conclude that the two classes we are predicting might be linearly separable, because Logistic Regression, which can not handle non-linearity well, did not perform worse then the other non-linear models, it actually outperforms most of them.

5.2 Interpretable models

K-Nearest Neighbor, Decision Tree and Logistic Regression are all 'white-box' models, meaning that we can explain how the model predicts data and why. This transparency is a huge advantage of the models, because we can explain why the model thinks a patient has a malignant tumor.

K-Nearest Neighbor

This model is interpretable because no parameters are learned, neither global weights nor structures. We can always get the k nearest neighbors that were used for the prediction of an instance. In this way, we could explain to a patient that the model predicted a malignant tumor because e.g. 23 out of 27 patients who have been previously diagnosed and had similar values for BI-RADS, Shape, Margin and Density, had a malignant tumor. If the number of features would grow, the interpretation would suffer, but in our scenario we can always take a look on the values of the k nearest neighbors to understand the decision.

Decision Tree

The most interpretable of all the algorithms analyzed is the decision tree. Each node presents a simple two partition of a feature. When we arrive at a leaf node we can read the predicted class. Fig. 27 shows our best decision tree model, having 3 levels and using entropy as a split quality measure. The root node tells us to look at the BI-RADS first, if they are smaller then 3.5, we take the left edge and look at the shape value in our patients data. If the shape is bigger then 0.5, we continue to the right child node and so on. This way we can explain every decision the model made when predicting a cancer severity to a patient.

The problem with our tree model is, that it learns to classify the benign instances really well, with a specificity of 89,93%, but performs worst of all models on the malignant tumor predictions, with a sensitivity of just 77,80%. Therefore, it is not very useful in our problem domain, because the sensitivity is much more important and the cost of a wrong malignant tumor prediction higher. Another downside is the feature age, dominating 3 out of 7 decision nodes, which is not the best criteria to rely the prediction of the the severity on. This downside explores more section 6.

Logistic Regression

Depending on the decision tree size the Logistic Regression could be more comprehensible, as it is a list containing a coefficient for each feature in the input data. An advantage of that is, that if we analyze our input data and the features more (Section 6), we can see how changes of a feature value affect the prediction. In general, the predictions of a Logistic Regression model are more precise in the prediction confidence, because it gets calculated for each single input, whereas in a decision tree every data subset representing an edge leaving a node gets the same prediction confidence assigned. With this information we can see for each input if we are unsure about the severity depending on the probability.

The disadvantage of the Logistic Regression model is that the decision line is a single hyperplane dividing the two classes. A decision tree can capture the decision boundaries more precise, it creates multiple small regions of one class membership with it's leaf nodes. Of course, this could lead to overfitting on the data, and we can see in fig. 9 and fig. 21 that the tree model does by far the best job in predicting instances of benign class (89,93% Specificity), but predicted the malignant class (77,8% Sensitivity) far worse, which is a sign that the tree learns to detect one class better then the other.

5.3 Non-Interpretable models

The kernel SVM model as well as the Random Forrest are both 'black-box' models, which means that their results can't be interpreted like of the previous models. It is difficult to understand how the 'black-box' model makes a decision, due to a high complexity.

In our domain, a 'black-box' model has certain disadvantages because we can't explain why the model thinks a patient has a malignant tumor. An explanation is important for such a serious and critical decision, with heavy consequences. Generally we want to provide the diagnose with the highest confidence, similarly to the authority of a well known doctor (combined with clearly interpretable laboratory results). Using machine learning algorithms to classify the severity of cancer could provide new and useful knowledge about the factors that contribute to a malignant tumor diagnosis and how the factors influence one another. However, implementing a black-box model for our classification problem limits the knowledge extraction about the features and the ability to

learn more about the cancer diagnosis apart from class affiliation, because they don't easily provide information on how the features interact with each other nor the significance of each feature for the model on a global level.

Moreover, the models need to be refit several times in the future to take into account new cancer diagnosis data with different correlations between the features instead of only predicting based on the correlations discovered in the training dataset.

On the other hand, we want to provide the highest diagnose quality possible, meaning that false diagnoses should occur as little as possible. This is the main advantage for choosing one of the 'black-box' models, because it could provide a confidence with a diagnose, by communicating its superb past performance measures, that is how accurate and precise the detection of malignant tumor was on the data the model was trained on (and on a study of a lot of unseen labeled data examples).

Kernel SVM

Normally, creating a good SVM model takes time and the training process is very slow. Our dataset is not large, thus we are not facing this problem. It delivers better results on the malignant class then the decision tree, logistic regression and random forest model and an equal result as the KNN. This could mean that the data points of the malignant class are easily separable from the benign class points by a non-linear hyperplane (that creates the rbf kernel SVM).

Random Forest

Although it is a black-box model, Random Forest delivers information about the feature importance, which could be used to extract new information about the disease and understand it better (Section 6.3). Further to leave out unimportant features in order to reduce noise and improve the prediction results. Nevertheless, the prediction decision can not be easily visualized compared to the decision tree model with a small depth.

5.4 Further Improvements

When we transformed the categorical feature shape, which does not have a natural order, we included a new bias in which some shape categories are closer to each other. This effects the results of distance based models like the KNN. It can be solved by using One-hot Encoding to create a vectorial space, so that the categories have the same distance to each other. In general, further inspection of the dataset and advanced preprocessing techniques could improve the performance of the models.

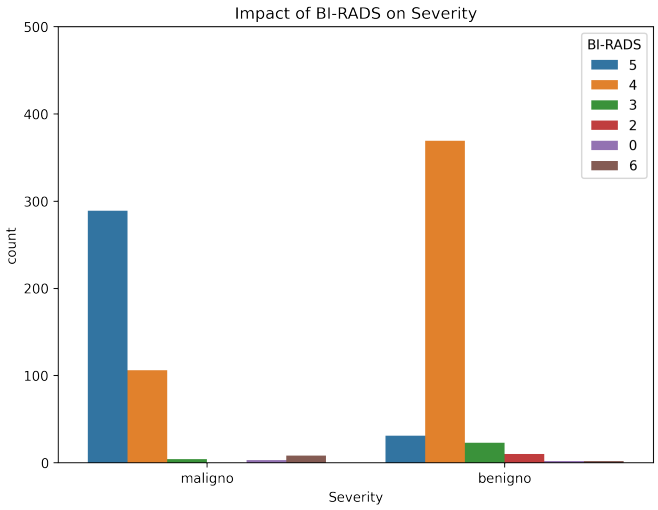

Figure 23: Impact of BI-RADS on Severity

6 Interpretation of data

"More data beats clever algorithms, but better and cleaner data beats more data" - Peter Norvik. We want to find out if some features are not significant for the prediction and remove them from the training set. For instance, the models based on Decision Trees as well as the dataset will be visualized in order to identify the features that determine each cancer severity. Afterwards we evaluate if we can further improve the results by eliminating features.

6.1 Visualization of feature impact on severity

Fig. 23 shows the impact of the feature BI-RADS on the severity of the cancer. We notice two very important observations. First, patients with a BI-RADS value of 5 are extremely more likely to have a malignant tumor. Second, a lot of patients in the data who have BI-RADS of 4, have a benign tumor. It is also more likely to have a malignant tumor with BI-RADS 4 then with any other (apart from 5), so the value 4 is not as predictive as 5.

Next, fig. 24 shows the impact of Margin on Severity. Here we can observe that if the mass margin is circumscribed (1), then the tumor is more likely to be benign. On the other hand, a malignant tumor occurs more often if the mass margin is speculated or ill-defined.

In addition, the impact of the shape of the abnormal mass detected on the severity is plotted in fig. 25. It shows that most patients with malignant tumor have a irregular (I) shape. On the contrary, patients with a round and oval shape have a higher probability of a benign tumor. For patients with not defined or

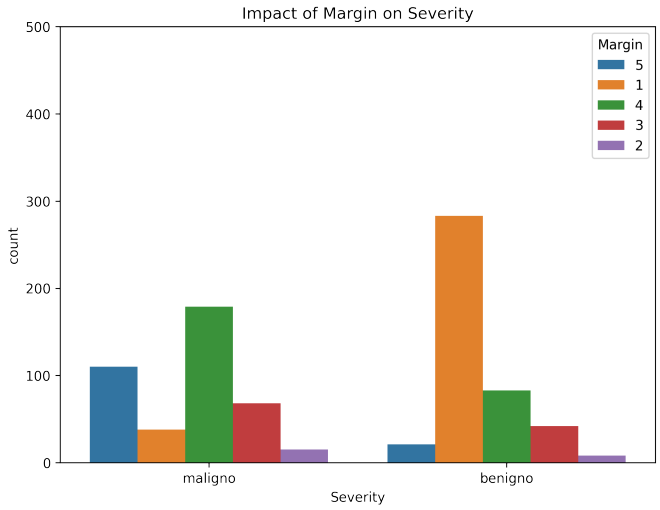

Figure 24: Impact of mass margin on severity

lobular shapes the chances for both tumors are equal.

Fig. 26 shows the impact of density on severity. The chances of both tumor types are more or less equal no matter what values density takes on.

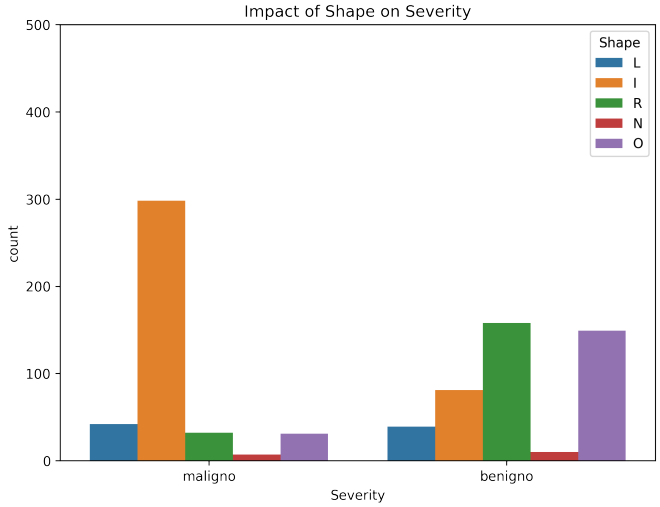

Figure 25: Impact of shape on severity

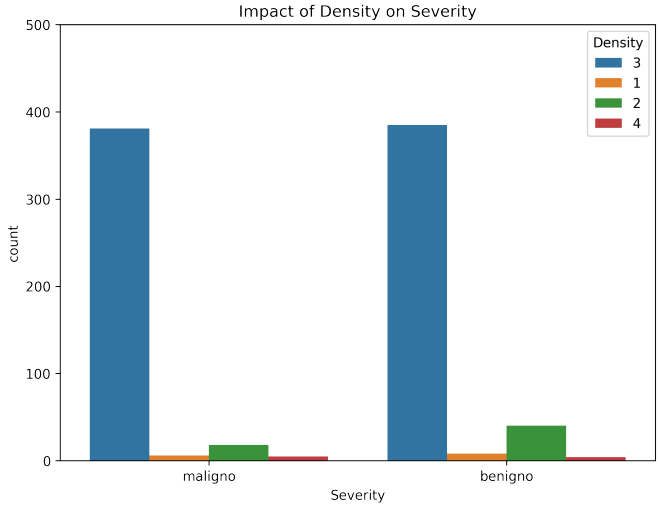

Figure 26: Impact of density on severity

Given the above, BI-RADS is the most important feature to predict the tumor type. On the contrary, Density gives us the least information on the severity. Yet, since there are 5 different features in the dataset, a plot showing the frequency of one feature and it's impact on the severity is not sufficient. We need to experiment with different combinations of features to see how they affect together the severity type.

6.2 Visualization of the Decision Tree

The best decision tree is shown in fig. 27. Th feature age, dominating 3 out of 7 decision nodes, is not the best criteria to rely the prediction of the the severity on. We remove age from the dataset and see how the tree is build in fig. 28.

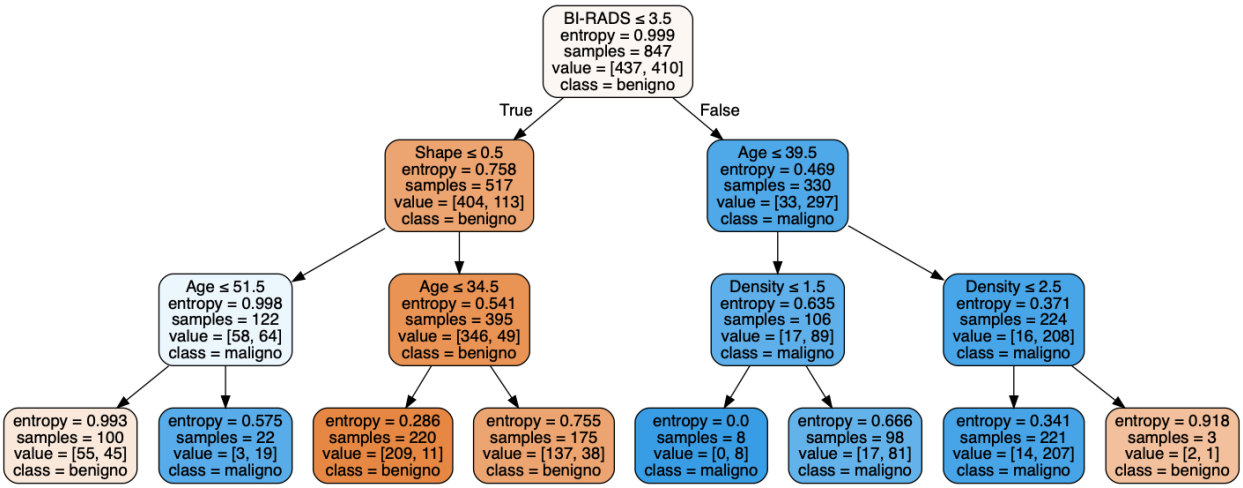

Figure 27: The graphic visualizes the best decision tree model.

Removing BI-RADS (fig. 29) leaves us a classification model that does not depend on the opinion of a radiologist, which assigns the BI-RADS. This makes the application of the model more powerful and meaningful.

Another tree model can be seen in fig. 30, which neither relies on BI-RADS nor on the age of the patient.

6.3 Feature importance from Random Forest

The Random Forest algorithm learns which features are important to predict the severity of the cancer, and the normalized feature importance scores can be seen in fig. 31. Correspondingly to the previous analysis, the Density has no importance and the BI-RADS score is the most important feature for the prediction. We will drop the Density feature to remove noise.

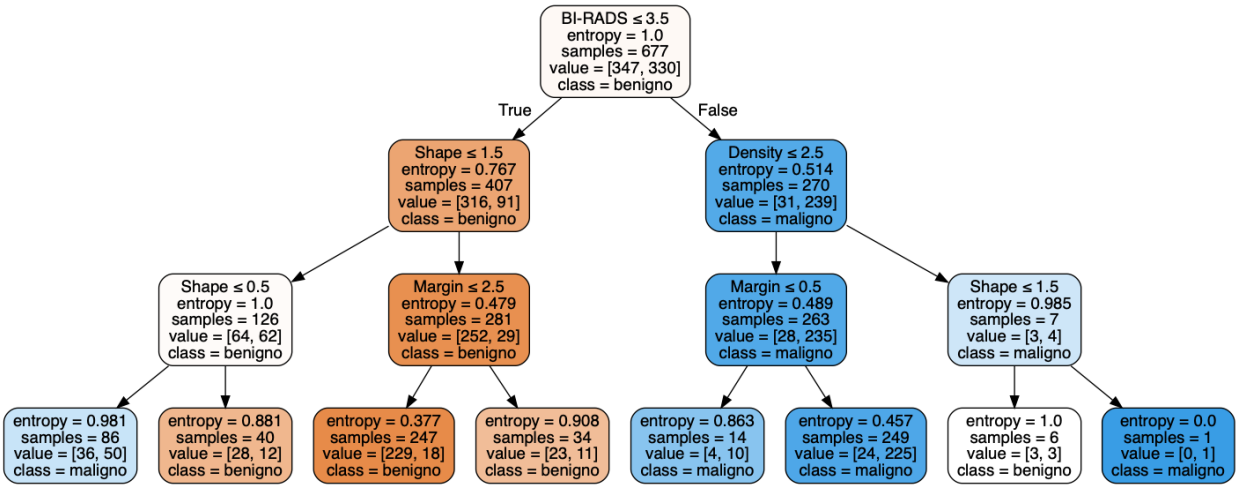

Figure 28: The graphic visualizes decision tree on the dataset without the age feature.

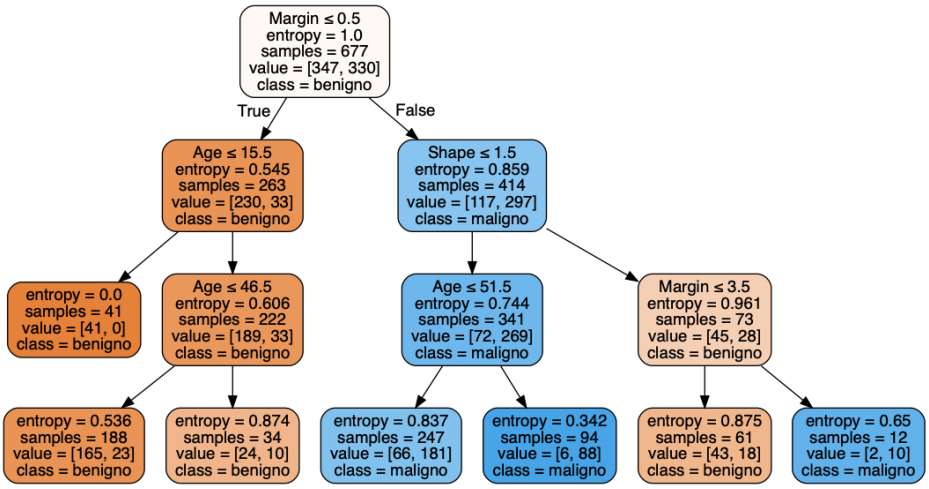

Figure 29: The graphic visualizes decision tree on the dataset without the BI-RADS feature.

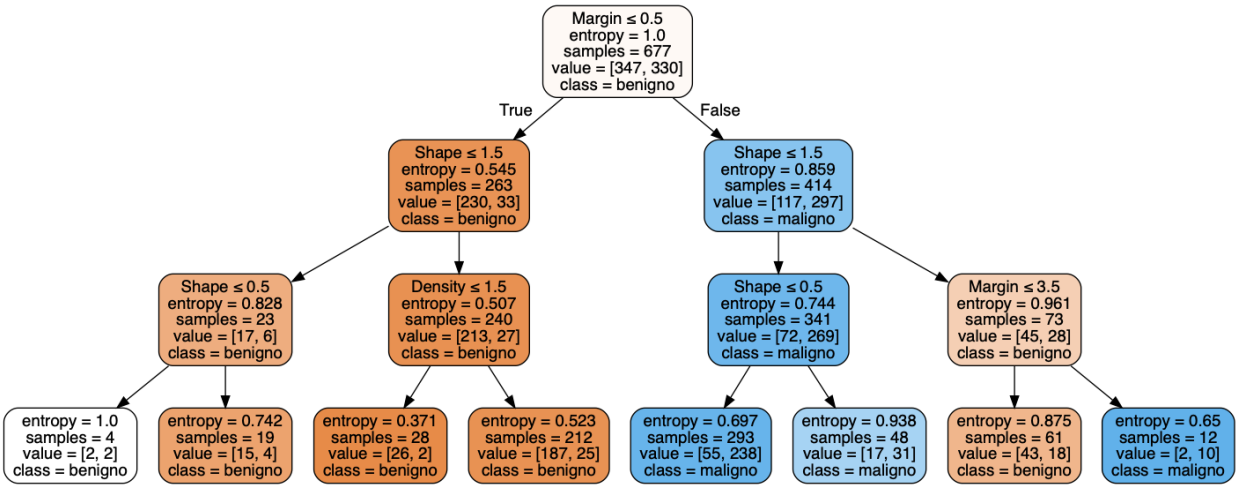

Figure 30: The graphic visualizes decision tree on the dataset without the age and B-RADS feature.

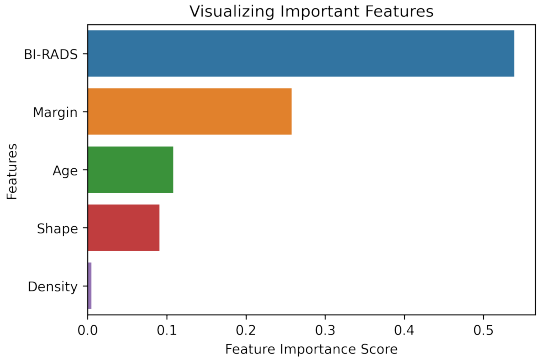

Figure 31: The graph visualizes the importance of each feature in the random forest model.

6.4 Results on modified datasets

Based on the disadvantages of the decision tree model, the following 3 datasets are created:

- data model 1: BI-RADS, Shape, Margin, Density (no Age)
- data model 2: Age, Shape, Margin, Density (no BI-RADS)
- data model 3: Shape, Margin, Density (no BI-RADS nor Age)

In addition, based on the feature importance of the random forest, the following data models are created:

- data model 4: BI-RADS, Age, Shape, Margin (no Density)
- data model 5: BI-RADS, Shape, Margin (no Density and no Age)

Decision Tree

Fig. 32 shows that we can improve the sensitivity by 2,2% when removing the age feature. The AUC decreases by 1,35% and the specificity decreases a lot. When we remove the BI-RADS, the sensitivity increases by 8,05%, but the specificity along with other measures like AUC to describe the balance of the performance on both classes decrease significantly. It turns around the result we had on the full dataset: having a great performance on detecting malignant tumor increases the error rate (false negatives) of the benign tumor class.

Evaluation Measure	full data	data model 1	data model 2	data model 3
accuracy	0.8406	0.8135	0.7957	0.7922
sensitivity	0.7780	0.8000	0.8585	0.8585
specificity	0.8993	0.8261	0.7368	0.7300
FPR	0.1007	0.1739	0.2632	0.2700
preciscion	0.8788	0.8119	0.7537	0.7489
F1 Score	0.8254	0.8059	0.8027	0.8000
AUC	0.8779	0.8644	0.8325	0.8125

Figure 32: The results obtained on the modified input data for the decision tree model.

Random Forrest

Fig. 33 shows that we can improve the sensitivity by 1,47% when removing the density feature, which we believe is creating noise and has no importance for the predictions. The AUC increases by 0,09% and the specificity decreases by 2,74%. When we remove the Age as well, the sensitivity increases by 3,42% and the AUC decreases only by 1,71%. Trying to predict without the BI-RADS (data model 2) leads astonishingly to high improvement in the sensitivity by 3,66% to 87,07%, at the cost of a false positive rate of 28,15%, double as on the full dataset.

Evaluation Measure	full data	data model 4	data model 5	data model 2
accuracy	0.8453	0.8383	0.8312	0.7922
sensitivity	0.8341	0.8488	0.8683	0.8707
specificity	0.8558	0.8284	0.7963	0.7185
FPR	0.1442	0.1716	0.2037	0.2815
preciscion	0.8444	0.8227	0.8000	0.7438
F1 Score	0.8393	0.8355	0.8327	0.8022
AUC	0.9015	0.9024	0.8844	0.8526

Figure 33: The results obtained on the modified input data for the random forest model.

References

[1] https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/ *Logistic Regression versus Decision Trees*, last access 03-11-2020.

[2] Christoph Molnar *Interpretable Machine Learning A Guide for Making Black Box Models Explainable*, 11-02-2020.