Brain tumor prediction model - HarvardX Data science Capstone assignment #2

edX learner DBel_17

April 25nd, 2023

INTRODUCTION

Glioma is the most frequent brain tumor and consists of cells that most closely resemble those brain cells which normally fulfill supportive and structural tasks, i.e. glial and astrocyte cells, as opposed to the signal transmitting neurons [1]. Clinical diagnosis and prognosis of glioma depend on magnetic resonance tomography (MRI) imaging and evaluation of the scans by experienced radiologists. In the past decade, machine-learning methods have been devised to identify and demarcate different regions of brain tumors, which is referred to as brain tumor segmentation. The goal of these endeavors is to facilitate and objectify tumor diagnosis. Precise diagnosis is at the core of appropriate patient care and prognostic projections [2].

In the present project, a pre-processed dataset was retrieved from the Kaggle repository [3], containing a series of MRI scan images from patients diagnosed with glioma. The scans show either completely normal or tumor-infiltrated areas of tissue. Each image was assigned a ground truth allowing for evaluation of prediction algorithms. Although the purpose of the original dataset was to classify and segment the MRI sections into areas of either normal tissue or four different types of tumorous tissue [4], the creator of the Kaggle dataset had simplified the task. Now, the outcome measure differentiates between images strictly positive or negative for the appearance of a tumor. Thus, it becomes a binary classification challenge.

Furthermore, the texture features of the images were extracted using mathematical methods applying the gray-level co-occurrence matrix (GLCM) [6] yielding first-order and second-order image features. First-order features, namely variance, standard deviation, kurtosis, skewness and mean, provide a measure of how grey pixels are distributed across the image overall, while second-order features such as angular second moment (ASM), entropy, contrast or dissimilarity quantify the relationship between pixels within the image and can be used to extrapolate to the coarseness or smoothness of an image.

METHODS

Data partitioning

To independently validate our model, we partitioned the initial dataset containing 3762 images into a development and a final holdout dataset, using a 90/10 partition, leaving enough data for training. The development data were further split into a training and test set, again using a 90/10 partition. Models were then trained and performance assessed. Finally, the models were trained on the complete development set to feed as much data as possible and the prediction algorithm evaluated on the final holdout set.

Feature selection

There were 13 predictors present in the Kaggle dataset: **Mean, Variance, Standard.Deviation, Entropy, Skewness, Kurtosis, Contrast, Energy, ASM, Homogeneity, Dissimilarity, Correlation,**

Coarseness. Predictors with low variance were identified with nearZeroVar() from the *caret* package and thus **Coarseness** was removed.

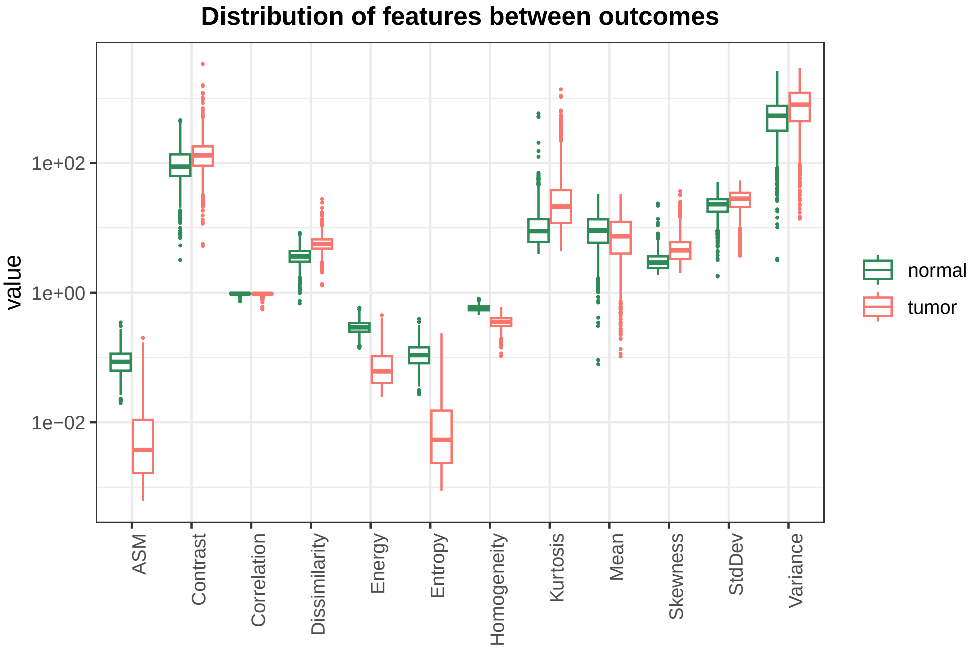


Figure 1: Outcome discrimination by features

As can be seen in Fig. 1, some features have more discriminative power than others and some like ASM and Entropy display almost the same distribution.

To exclude redundant features and therefore to decrease the possibility of overfitting, to reduce dimensionality, as well as to save computational resources, a filter method was applied examining the correlation and relationships between predictors among themselves and between predictors and the outcome.

To assume high correlation, the Pearson correlation coefficient had to be greater than 0.7. Deterministic relationships between features spoke in favor of retaining only the one with the highest correlation to the outcome, while stochastic relationships were weighed in favor of retaining a feature in a correlation cluster. To not lose important information, the decision to exclude features was therefore conservative. A pairwise scatter plot grid (Fig. 2) and a pairwise correlation matrix (Fig. 3) guided the selection.

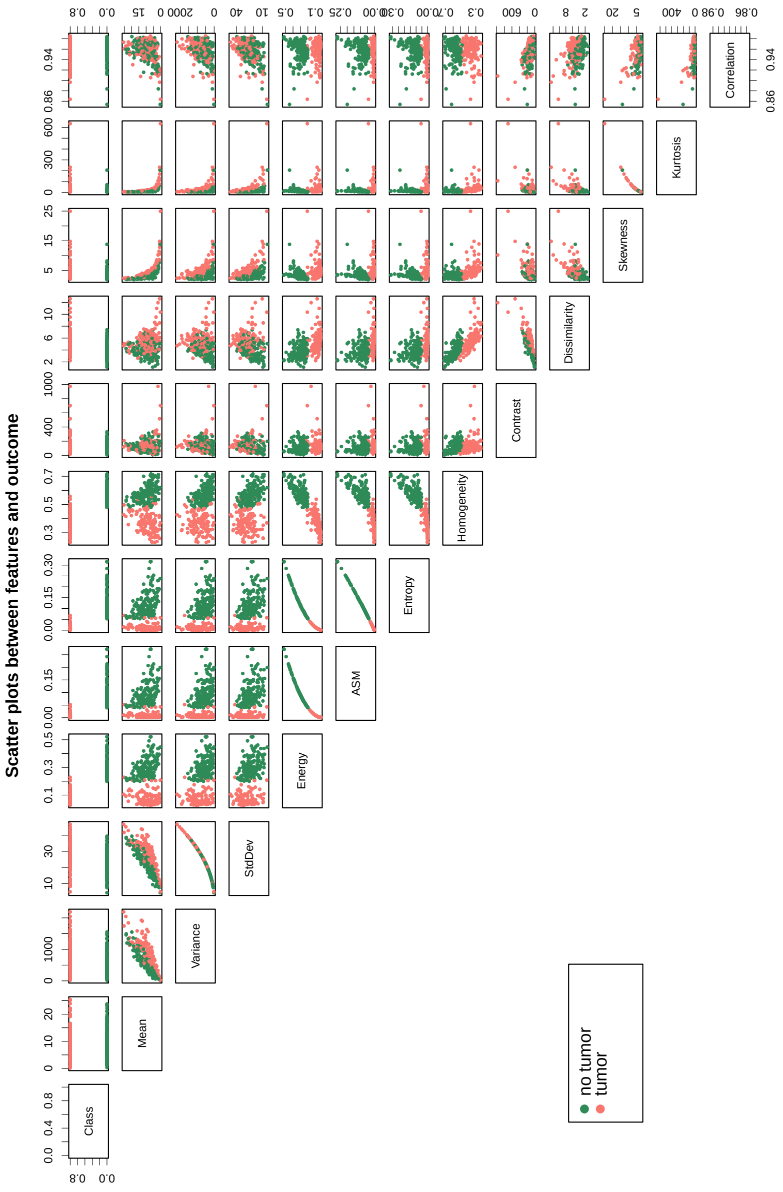
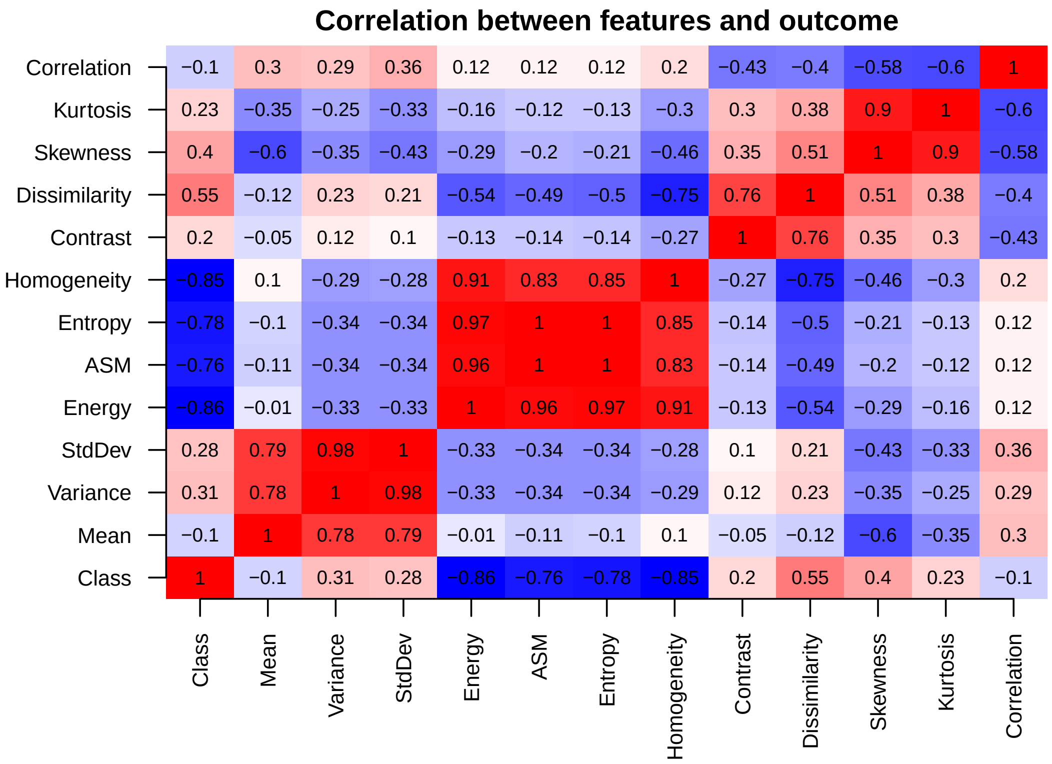


Figure 2: Scatter plots between features and outcome.



	Class	Mean	Variance	StdDev	Energy	ASM	Entropy	Homogeneity	Contrast	Dissimilarity	Skewness	Kurtosis	Correlation
Correlation	−0.1	0.3	0.29	0.36	0.12	0.12	0.12	0.2	−0.43	−0.4	−0.58	−0.6	1
Kurtosis	0.23	−0.35	−0.25	−0.33	−0.16	−0.12	−0.13	−0.3	0.3	0.38	0.9	1	−0.6
Skewness	0.4	−0.6	−0.35	−0.43	−0.29	−0.2	−0.21	−0.46	0.35	0.51	1	0.9	−0.58
Dissimilarity	0.55	−0.12	0.23	0.21	−0.54	−0.49	−0.5	−0.75	0.76	1	0.51	0.38	−0.4
Contrast	0.2	−0.05	0.12	0.1	−0.13	−0.14	−0.14	−0.27	1	0.76	0.35	0.3	−0.43
Homogeneity	−0.85	0.1	−0.29	−0.28	0.91	0.83	0.85	1	−0.27	−0.75	−0.46	−0.3	0.2
Entropy	−0.78	−0.1	−0.34	−0.34	0.97	1	1	0.85	−0.14	−0.5	−0.21	−0.13	0.12
ASM	−0.76	−0.11	−0.34	−0.34	0.96	1	1	0.83	−0.14	−0.49	−0.2	−0.12	0.12
Energy	−0.86	−0.01	−0.33	−0.33	1	0.96	0.97	0.91	−0.13	−0.54	−0.29	−0.16	0.12
StdDev	0.28	0.79	0.98	1	−0.33	−0.34	−0.34	−0.28	0.1	0.21	−0.43	−0.33	0.36
Variance	0.31	0.78	1	0.98	−0.33	−0.34	−0.34	−0.29	0.12	0.23	−0.35	−0.25	0.29
Mean	−0.1	1	0.78	0.79	−0.01	−0.11	−0.1	0.1	−0.05	−0.12	−0.6	−0.35	0.3
Class	1	−0.1	0.31	0.28	−0.86	−0.76	−0.78	−0.85	0.2	0.55	0.4	0.23	−0.1

Figure 3: Correlation between features and outcome. Deep red represents stronlgy positive and deep blue strongly negative correlation.

Mean, Variance and Std.Dev. are highly correlated between each other. Variance has the highest positive correlation to the outcome (“Class”), but Mean is anti-correlated and shows a stochastic relationship to Variance. Therefore, Mean and Variance were kept.

Entropy, Energy, ASM and Homogeneity are highly correlated. Energy has the highest anti-correlation to class. Homogeneity is the only one to show a stochastic relationship to the other three features in the cluster. We therefore kept Energy and Homogeneity.

Dissimilarity and Contrast are highly correlated. Dissimilarity has higher correlation to Class and was retained.

Kurtosis and Skewness are highly correlated and have a deterministic relationship. Skewness has the stronger correlation to Class and was retained.

In summary, we kept the following 7 of the 12 non-zero-variance features for training: **Variance**, **Mean**, **Energy**, **Homogeneity**, **Skewness**, **Dissimilarity** and **Correlation**.

Scan visualization

MRI scans were read with the readJPEG() function of the *jpeg* package and visualized using the base *R* plot(as.raster()) function. An example of randomly selected images of normal and tumorous tissues is provided in Fig. 4. It can be seen that all sections are in the transverse plane.

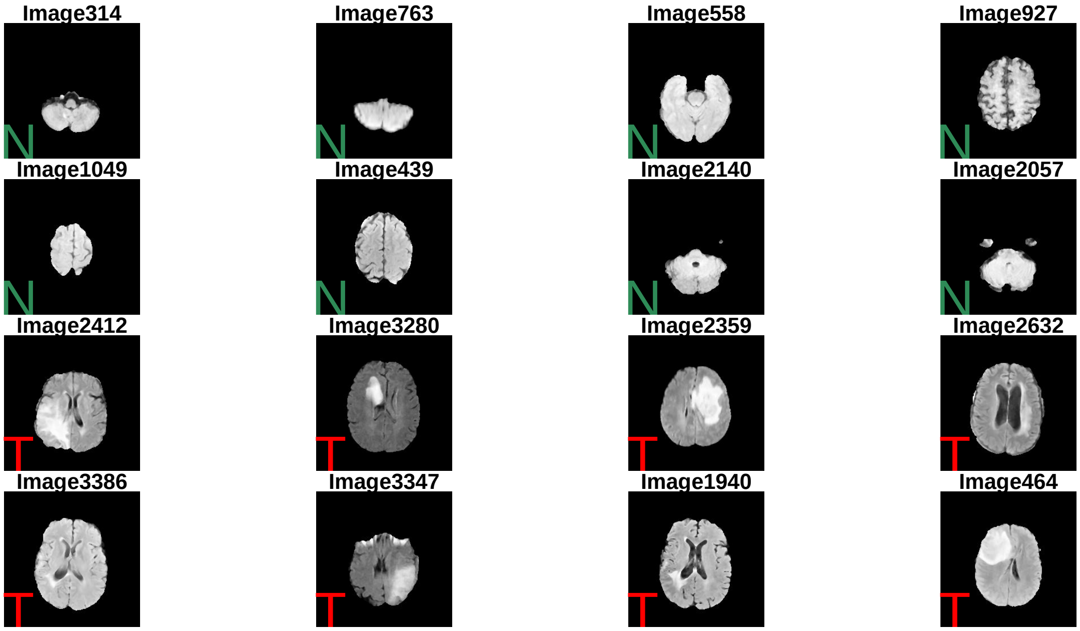


Figure 4: Examples of MRI images. N stands for normal, T for tumor.

Model training

Six different machine learning algorithms supported by the *caret* [7] package were selected, then tuned and trained on the training data set. The commonly used, 10-fold cross-validation was used in order to prevent overfitting, while keeping the validation set for training similarly sized as the eventual test and final holdout evaluation sets.

We selected algorithms suitable for classification tasks: the k-nearest-neighbors (caret method: “knn”), generalized linear model (“glm”, suitable to model logistic regression), locally estimated scatterplot smoothing (“gamLoess”, only *span* was tuned keeping *degree* constant at 1), classification trees (“rpart”), random forests (“rf”) and gradient boosting machines (“gbm”). Gradient boosting machines are similar to random forests, but they construct trees sequentially and add weights to observations based on the errors of the previous tree.

Based on its comparatively poor performance, we excluded the knn model from further consideration. The remaining five models were used to generate a majority ensemble vote to predict the outcome variable:

$$E(Y|\mathbf{X}) = \begin{cases} 1 & \text{if} \frac{1}{B}\sum_{b=1}^{B}\hat{Y}_b > 0.5 \\ 0 & \text{otherwise} \end{cases}, \text{ with } B \text{ being the number of models providing the prediction } \hat{Y}_b$$

Assessing model performance

Since we have a discrete, binary outcome, we used accuracy A as a measure of model performance which is defined as

$$A = \frac{TP+TN}{TP+TN+FP+FN}, \text{ with T being "true", F "false", P "positive" and N "negative".}$$

Any model performance has to be measured against a random prediction which in the case of a binary outcome would have an accuracy of 50%.

Accuracy was assessed on the test data set to gauge robustness of the trained models. Eventually it was evaluated on the final holdout test.

RESULTS

k-nearest neighbors (kNN)

The kNN algorithm was found to yield the highest accuracy of 83.2% at $k = 5$.

Logistic regression

Logistic regression via a generalized linear model (glm) provided an accuracy of 98.5%, a much better performance than the kNN model.

Locally estimated scatterplot smoothing (loess)

A loess smooth function yielded the highest accuracy of 99.1% at a *span* of 0.16.

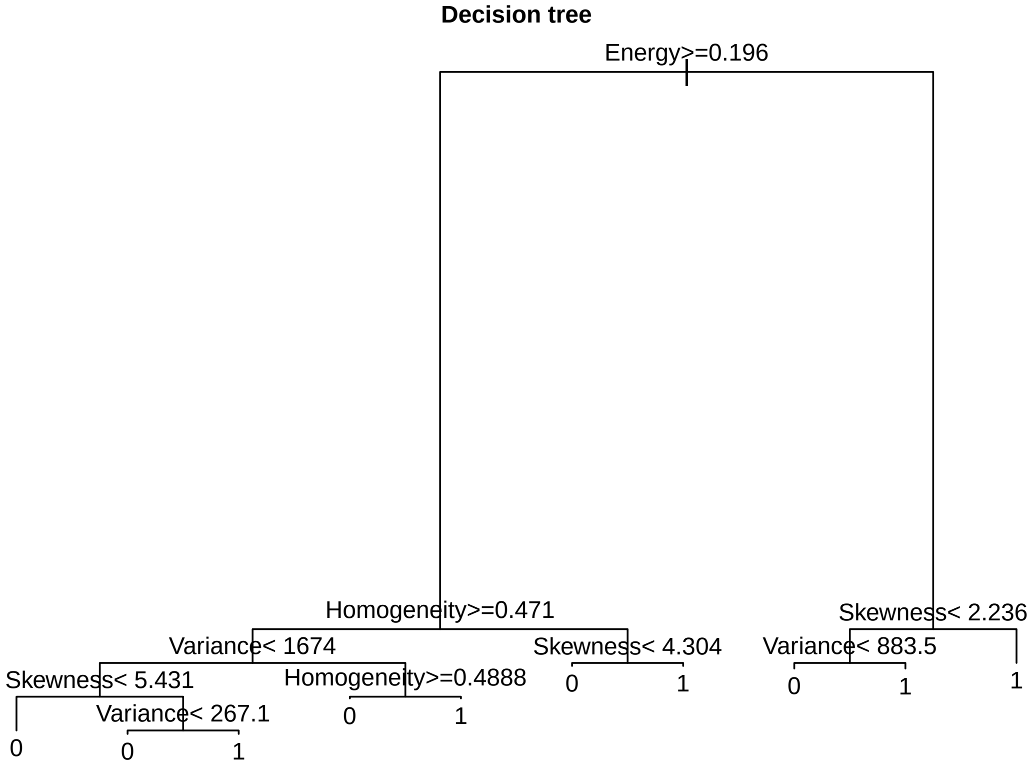


Figure 5: Decision tree obtained by "rpart" algorithm with cp = 0.001, minsplit = 50 and minbucket = 21. 0 is "normal", 1 is "tumor".

Classification trees

We tuned a classification tree model and obtained optimal parameters for complexity: $cp = 0.001$ (minimum required improvement of the residual sum of squares to add a partition); the minimum number of observations to allow a further partition: $minsplit = 50$; and minimum amount of observations in each final node: $minbucket = 21$, resulting in an accuracy of 98.8%. The obtained tree is displayed in Fig. 5.

Random forest

For the training of a random forest model we obtained an optimal number of predictors (*mtry* parameter) of 4 and a minimum amount of observations in each final node (*nodesize* parameter) of 6. This resulted in an accuracy of 99.1%.

Table 1: *Variable importance in random forest model*

	Var.Import
Energy	100.0000000
Homogeneity	46.5770720
Dissimilarity	5.9693053
Skewness	2.7836935
Variance	1.1266177
Correlation	0.0333372
Mean	0.0000000

The obtained forest assigned the highest variable importance to **Energy**, followed by the highly correlated **Homogeneity** (*cf.* Fig. 3) and then **Dissimilarity** (Table 1).

Gradient boosting machines (GBM)

Using an optimal set of parameters (*n.trees* = 150, *interaction.depth*=3, *shrinkage*=0.1, *n.minobsinnode* = 10) the GBM model reached an accuracy of 99.1%, equivalent to the random forest, albeit the importance of **Energy** greatly outweighed all the other features (Table 2).

Table 2: *Variable importance in gradient boosting machine model*

	Var.Import
Energy	100.0000000
Homogeneity	6.2179084
Skewness	2.8501365
Variance	1.3461381
Correlation	0.7609690
Mean	0.4794917
Dissimilarity	0.0000000

Ensemble

Table 3 shows the accuracies of all six models obtained on the test set, as well as the accuracy obtained with the ensemble majority vote from the five more accurate models.

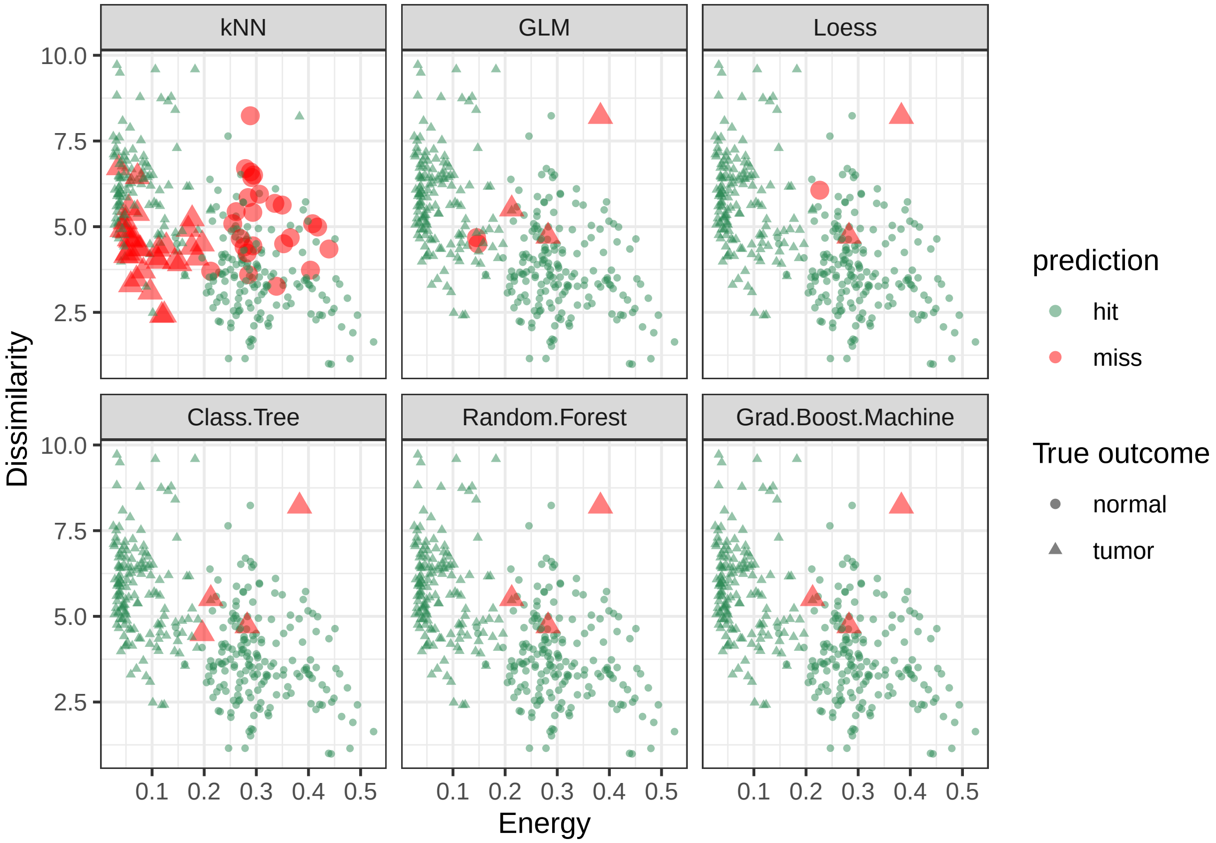


Figure 6: Prediction accuracy of the six examined models projected onto a Dissimilarity vs. Energy plot. Failed predictions in red ("miss").

Table 3: *Comparison of the accuracies of different models during training*

Model	Accuracy
kNN	0.8318584
GLM	0.9852507
Loess	0.9911504
Class. tree	0.9882006
Random forest	0.9911504
GBM	0.9911504
Ensemble	0.9911504

Curiously, the missed predictions by the more accurate models were mostly (85.7%) correctly predicted by the kNN model (Fig. 6).

The ensemble method (excluding kNN) finally yielded an accuracy of 99.1%.

Evaluation on the final holdout test

We trained the final model on the combined development (train and test) set and assessed its performance on the final holdout validation set.

Table 4: *Comparison of the accuracies of different models during validation*

Model	Accuracy
kNN	0.8435013
GLM	0.9761273
Loess	0.9840849
Class. tree	0.9734748
Random forest	0.9840849
GBM	0.9840849
Ensemble	0.9840849

As can be seen in Table 4, the ensemble method yielded an accuracy of 98.4%, performing equivalently to the Loess, random forest and GBM methods, as observed during training (*cf.* Table 3). Fig. 7 shows the mapping of the model predictions and Fig. 8 shows the same for the ensemble method. The accuracy was approximately within the stochastic range from the initial training accuracy, suggesting that we did not overtrain the model. The kNN model correctly predicted 64.3% of the missed calls from the other models, a lower ratio than during training.

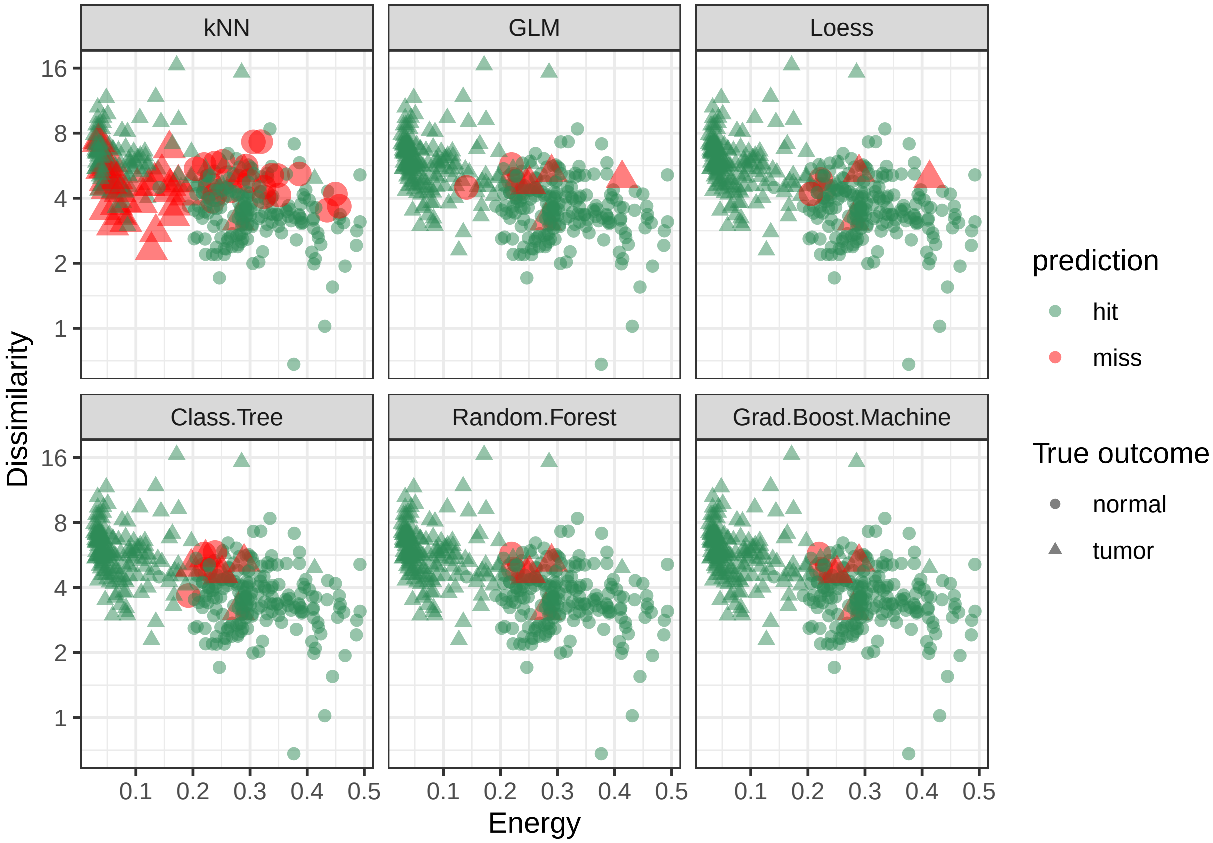


Figure 7: Prediction accuracy of the six examined models in the validation set, projected onto a Dissimilarity vs. Energy plot. Failed predictions in red ("miss").

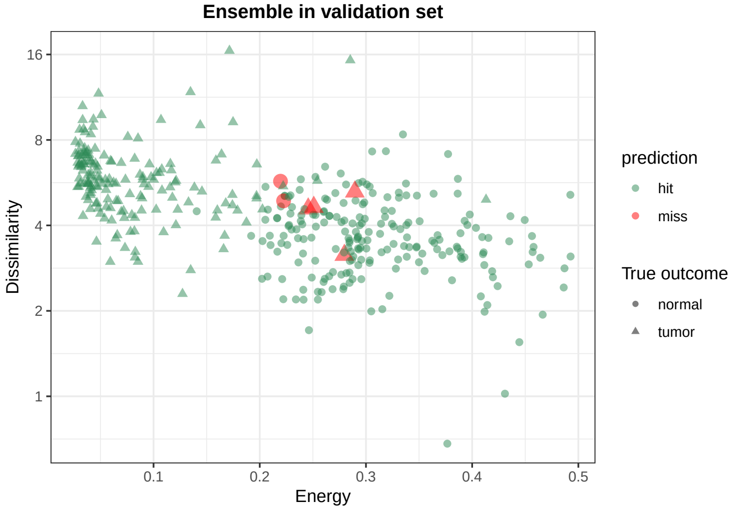


Figure 8: Prediction accuracy of the Ensemble model in the validation set, projected onto a Dissimilarity vs. Energy plot. Failed predictions in red ("miss").

Finally, we visualized those nine MRI scans that were misclassified by the ensemble method during both training and validation (Fig. 9) and a selection of random MRI scans there were correctly classified (Fig. 10). It seems evident, that misclassified images contain sections from more anatomically superior and inferior slices of the brain that are potentially harder to classify correctly as they cover less area.

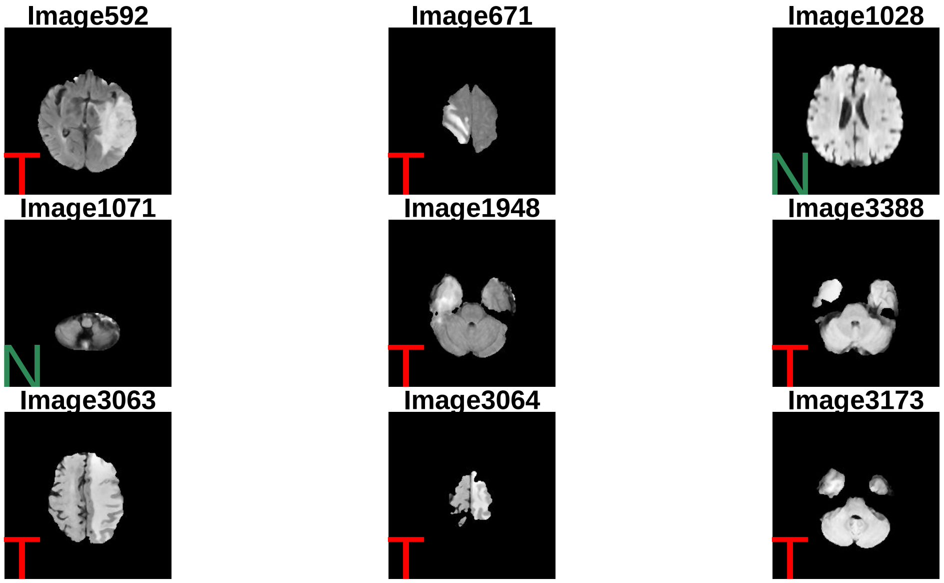


Figure 9: Combination of all MRI scans that were misclassified by the ensemble-model during training plus during validation. Correct class is displayed.

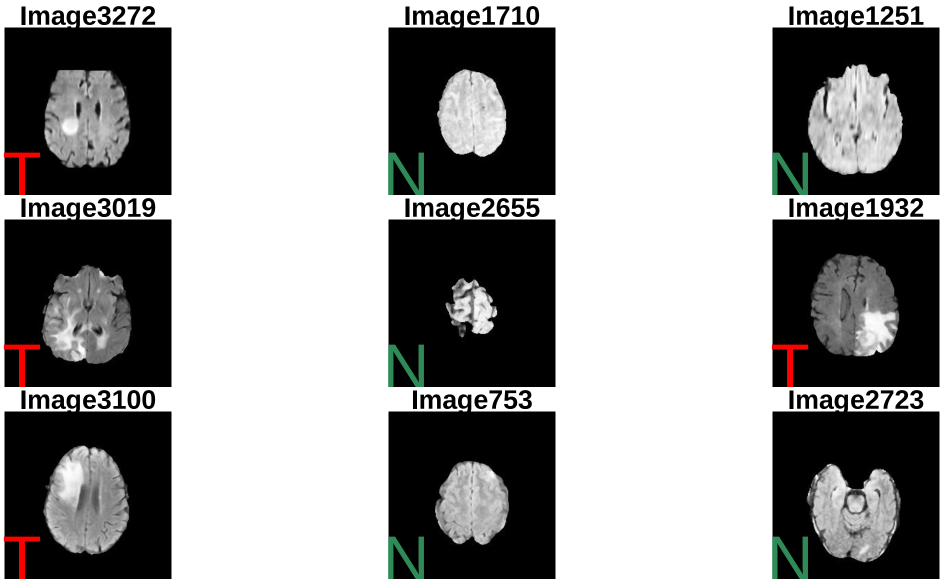


Figure 10: Randomly selected MRI scans that were correctly classified by the ensemble-model in validation.

When we look at the scan images more closely, it becomes apparent that they are organized in sequences of transverse sections from the same patients' MRI results. We examined images that seem to be the matching scans to Images 3063 and 3064 (misclassified, Fig. 9), from the more inferior areas of the same brain (Fig. 11). All but the most inferior were assigned to be tumor-infiltrated as ground truth.

There is obviously additional information about the spatial relationship between the sections that can help improve the prediction algorithm by assigning weighted probabilities depending on how adjacent sections have been classified and how close they are to the section in question. If, for instance, multiple sections in a row were classified as tumor-infiltrated, then the probability that the immediately adjacent section will also contain tumorous tissue would increase. The appropriate annotation to implement such an approach was not available for this dataset, however.

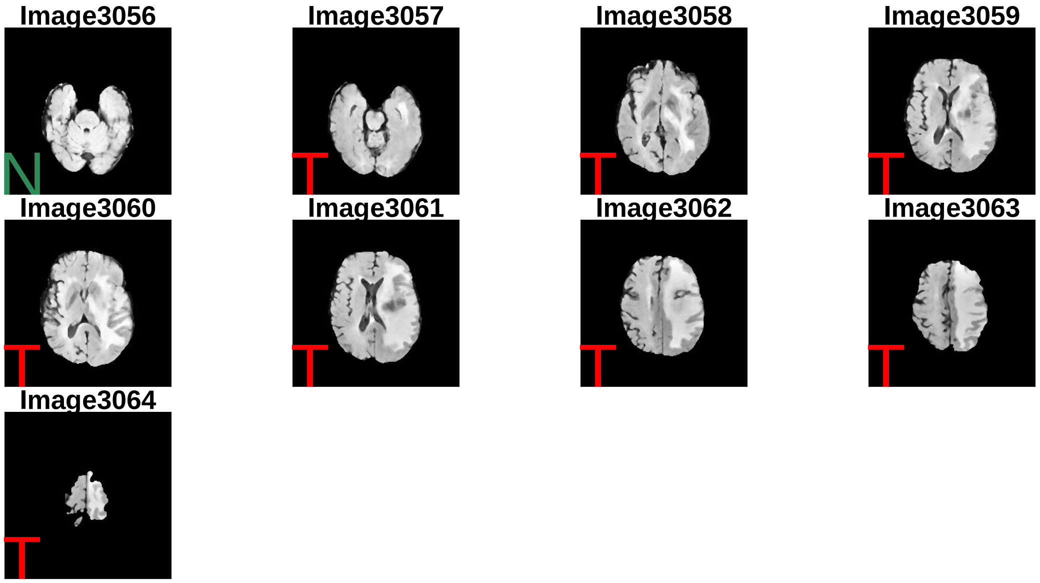


Figure 11: Images from an MRI slices stack

DISCUSSION AND CONCLUSION

Harnessing the power of image feature extraction by GLCM enabled us to train a model that was highly accurate in distinguishing between normal and tumorous brain tissues in MRI scans. A drawback is that all scanned brains were from tumor patients. An additional cohort of MRI scans from healthy individuals would provide a background against which the diagnosis of a tumor by machine learning could be ascertained.

The kNN model performed much worse than other commonly used models. This is due to the curse of dimensionality [8] and accordingly, reducing the number of features to just two (Energy and Dissimilarity) markedly improves accuracy to 96.3%. The Loess, random forest and GBM models performed with the same accuracy as the ensemble during both training and validation. Training and tuning Loess was much more computationally expensive and therefore future endeavors should prioritize random forests or the even faster GBM model. The ensemble method did not seem to have drawbacks and should be used to increase robustness against outliers and possibly include a kNN model trained with an appropriate amount of predictor variables.

Accuracy could likely be improved even further if scan images were assigned to individuals. This would allow to use the spatial relationship between images of the same brain and to add a subsequent re-evaluation step using conditional probabilities based on the classification of adjacent slices. The consideration of the four different MRI modalities that were used to generate the images might also increase the confidence of classification.

Lastly, this project is just a step stone to developing a brain tumor segmentation algorithm that identifies the precise areas of a tumor and differentiates between tumor tissue types.

SESSION INFO

```
## R version 4.2.2 (2022-10-31)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] pandoc_0.1.0    float_0.3-1     tinytex_0.44    ggrepel_0.9.3
##  [5] scales_1.2.1    gridExtra_2.3   dslabs_0.7.4    jpeg_0.1-10
##  [9] gam_1.22-2      foreach_1.5.2   gbm_2.1.8.1     caret_6.0-94
## [13] lattice_0.21-8  lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0
## [17] dplyr_1.1.1     purrr_1.0.1     readr_2.1.4     tidyr_1.3.0
## [21] tibble_3.2.1    ggplot2_3.4.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] prodlim_2023.03.31   stats4_4.2.2         yaml_2.3.7
##  [4] globals_0.16.2       ipred_0.9-14         pillar_1.9.0
##  [7] glue_1.6.2           pROC_1.18.0          digest_0.6.31
## [10] randomForest_4.7-1.1 hardhat_1.3.0        colorspace_2.1-0
## [13] recipes_1.0.5        htmltools_0.5.5      Matrix_1.5-4
## [16] plyr_1.8.8           timeDate_4022.108    pkgconfig_2.0.3
## [19] listenv_0.9.0        gower_1.0.1          lava_1.7.2.1
## [22] tzdb_0.3.0           proxy_0.4-27         timechange_0.2.0
## [25] farver_2.1.1         generics_0.1.3       withr_2.5.0
## [28] nnet_7.3-18          cli_3.6.1            survival_3.5-5
## [31] magrittr_2.0.3       evaluate_0.20        fs_1.6.1
## [34] future_1.32.0        fansi_1.0.4          parallelly_1.35.0
## [37] nlme_3.1-162         MASS_7.3-58.3        class_7.3-21
## [40] tools_4.2.2          data.table_1.14.8    hms_1.1.3
## [43] lifecycle_1.0.3      munsell_0.5.0        e1071_1.7-13
## [46] compiler_4.2.2       rlang_1.1.0          grid_4.2.2
## [49] iterators_1.0.14     rstudioapi_0.14      rappdirs_0.3.3
## [52] labeling_0.4.2       rmarkdown_2.21       gtable_0.3.3
## [55] ModelMetrics_1.2.2.2 codetools_0.2-19     reshape2_1.4.4
## [58] R6_2.5.1             knitr_1.42           fastmap_1.1.1
## [61] future.apply_1.10.0  utf8_1.2.3           stringi_1.7.12
## [64] parallel_4.2.2       Rcpp_1.0.10          vctrs_0.6.1
## [67] rpart_4.1.19         tidyselect_1.2.0     xfun_0.38
```

REFERENCES

1 Zong, H. *et al.* The cellular origin for malignant glioma and prospects for clinical advancements. *Expert Review of Molecular Diagnostics*, 2012, 12 (4), pp. 383–394. DOI: 10.1586/erm.12.30

2 Menze, B. *et al.* The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 2014, p. 33. DOI: 10.1109/TMI.2014.2377694

3 Bohaju, J. Brain tumor. 2020. DOI: 10.34740/KAGGLE/DSV/1370629

4 *BRATS 2015: Brain tumor image segmentation challenge.* 2015. Available from: https://www.smir.ch/BRATS/Start2015 [Accessed 22 April 2023]

5 Haralick, R.M. *et al.* Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 1973, SMC-3 (6), pp. 610–621. DOI: 10.1109/TSMC.1973.4309314

6 Aggarwal, N. *et al.* First and second order statistics features for classification of magnetic resonance brain images. *Journal of Signal and Information Processing*, 2012, 3 (2), pp. 146–153. DOI: 10.4236/jsip.2012.32019.

7 Kuhn, M. Building predictive models in r using the caret package. *Journal of Statistical Software*, 2008, 28 (5), pp. 1–26. DOI: 10.18637/jss.v028.i05

8 Irizarry, R.A. Introduction to data science. 2022. Available from: http://rafalab.dfci.harvard.edu/dsbook [Accessed 22 April 2023]