**Machine Learning Engineer Nanodegree**

**Capstone Project**

Kareem Wahid   
June 10th, 2017

**I. Definition**

**Project Overview**

Radiomics is a budding new field of medical informatics that seeks to extract mathematically defined quantitative features such as statistics, shape, and texture from medical images 1. Medical imaging up until recently could only provide visual qualitative information to a clinician. Radiomics allows hidden quantitative information in medical images to be deciphered and analyzed, with the potential to aid in prognosis of disease.

Cancer imaging in particular is a heavily researched area of radiomics. Tumors are often spatially and temporally heterogeneous. This frequently requires multiple tissue biopsies to be performed in order to capture the molecular heterogeneity of the tumor, which can be dangerous for the patient. Radiomics provides a non-invasive window into probing the heterogeneity of a tumor 2. Gliomas are the most common variety of primary brain malignancies and have a high degree of intrinsic heterogeneity. This heterogeneity is apparent in their appearance and shape upon imaging, making prognosis difficult 3. Radiomic analysis of glioma medical imaging can provide additional information about a patient’s prognosis and likely survival outcomes 4-6.

Though significant research has been conducted on the application of machine learning algorithms to radiomic features for prognostic prediction 7-8, there is still much that is unknown about which models are best due to lack of standardization in the field. Herein, supervised machine learning algorithms are trained on radiomic features extracted from glioblastoma magnetic resonance images obtained from the 2017 BraTS Challenge to predict patient overall survival outcomes. The popular Python machine learning library scikitlearn was used for all calculations. Classifier performance was evaluated using the area under the receiver operator curve metric and compared to previously published data.

**Problem Statement**

The Multimodal Brain Tumor Segmentation (BraTS) Challenge 9 is an annual competition that seeks to employ the brightest minds in computational radiology to develop the following:

- Accurate segmentation algorithms of gliomas in magnetic resonance images.

- *Prediction of patient overall survival through radiomics and machine learning algorithms.*

It is the goal of this project to address the second listed objective of the BraTS Challenge. In this study we will determine which supervised machine learning classification models are the most suitable for predicting prognostic information from radiomic features of glioblastoma magnetic resonance imaging (MRI) scans acquired from the 2017 BraTS Challenge. A variety of supervised classification methods trained on radiomic feature data and known prognostic outcomes will be compared through quantifiable metrics to determine which method most accurately predicts the prognostic class for a set of new patients unseen radiomic feature inputs. In addition, different feature selection methods and the number of features used for training classifiers will be varied to examine their effect on classifier predictive performance.

**Metrics**

The Receiver Operating Characteristics (ROC) curve is a commonly utilized metric to evaluate binary classifier output performance 12. ROC curves typically display true positive rate on the Y-axis and false positive rate on the X-axis as discrimination threshold is varied. The area under the ROC curve, AUC, can be used to quantify the degree to which a model is able to accurately classify data. AUC values close to 0.5 are worse, signifying random guessing, while AUC values close to 1.0 are better, signifying perfectly accurate classification. Herein, we will use the AUC values to measure the predictive performance of our classification method/feature selection method combinations, and compare them to the results in Parmar et al.

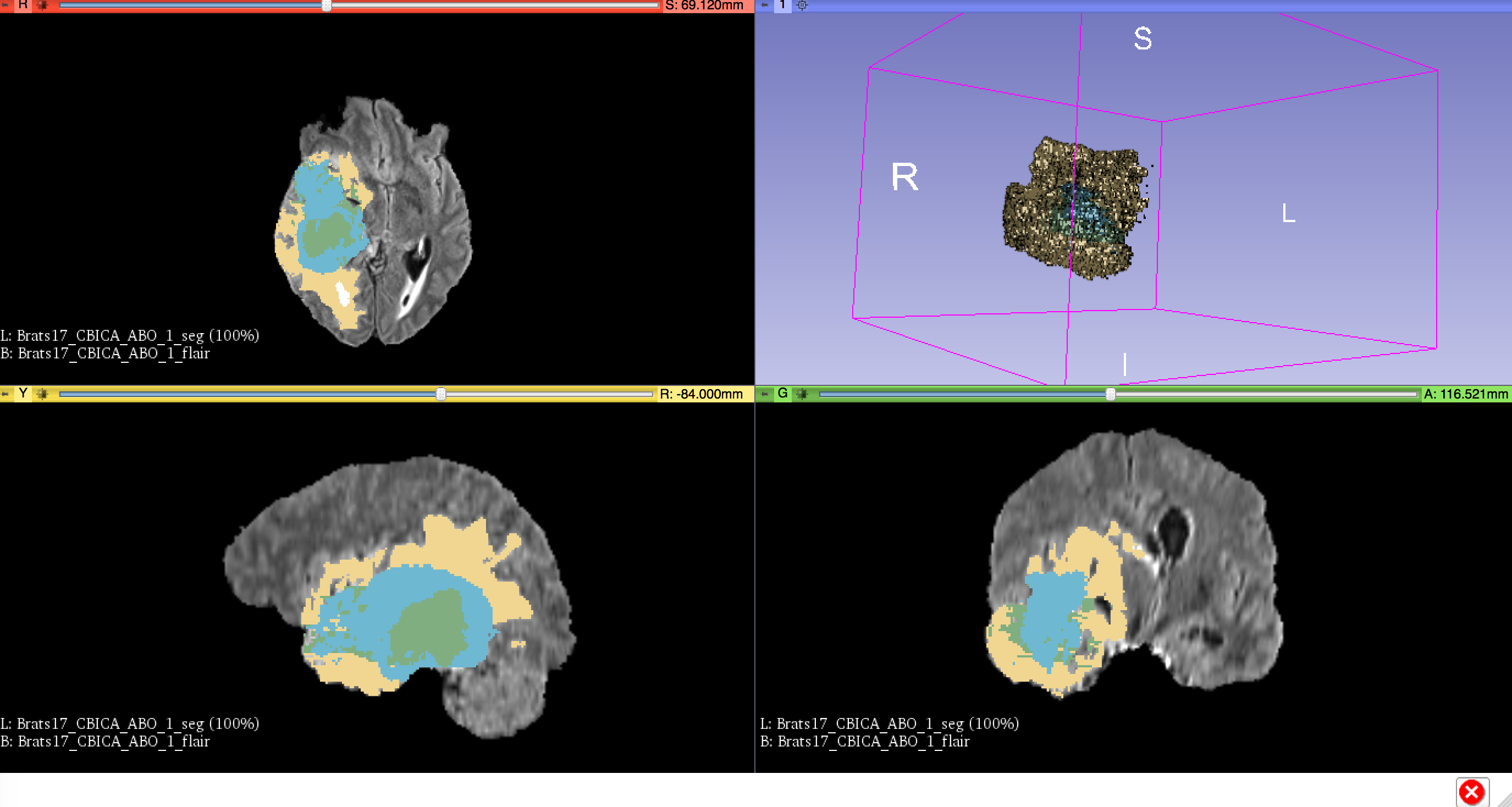
Desirable models should be stable in response to small perturbations to the input space. Classifier stability will be measured via the relative standard deviation (RSD), a metric described in Parmar et al. RSD can be defined as:

where and are the standard deviation and mean of the AUC values generated from repeated subsamples of the training data using a bootstrap approach. In this study, train test split will be iterated 10 times with different random states to generate and .

**II. Analysis**

**Data Exploration**

The BraTS Challenge provides a standardized MRI dataset of patients diagnosed with gliomas from the publically available Cancer Imaging Archive; tumor volumes have been segmented by expert radiologists and are included in the dataset. An example of a high-grade glioblastoma (GBM) MRI scan and corresponding tumor segmentation from the BraTS dataset is visualized in Figure 1. In addition the BraTS Challenge provides overall survival data for patients corresponding to these MRI scans. The MRI dataset and patient survival data from the 2017 BraTS Challenge was obtained for this project by entering the competition.



**Figure 1:** Visualization of GBM FLAIR MRI scan with corresponding tumor segmentation obtained from BraTS dataset. Tumor segmentation represented by colored areas. Color-coding represents subregions of tumor but will not be considered separately in this study. Tumor segmentation 3D volume is visualized in top right corner. Visualization performed in 3D Slicer 10.

The BraTS Challenge offers the number of days survived after diagnoses (overall survival) for 163 GBM patients. The patient overall survival data are continuous numerical values. The distribution of the patient survival data is plotted in Figure 2a. It can be observed that the survival data has a leftward skew with a mean survival centered around 400 days. Since the patient survival is our output, it is necessary to transform the continuous values to categorical values for use in classification. The data has been partitioned into two categorical values using a cutoff time of one year, which is based on median survival rates for glioblastoma 5; patients who live less than one year are assigned a 0, patients who live longer than one year are assigned a 1. These patients will be referred to as short-term and long-term survivors throughout this manuscript. These transformed overall survival categories will serve as the output for our models. It can also be seen that partitioning the survival data using a cutoff time of one year yields approximately equal instances of short-term and long-term classes, which makes subsequent analysis more manageable (Fig 2b).

Macintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:Survival_distribution.pdfMacintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:Survival_0_1_distribution.pdf

**Figure 2**: Survival distribution before (a) and after partitioning (b).

94 radiomic features corresponding to statistics (19), shape (16), and texture (59) were extracted from each of the 163 GBM MRI images/segmentations by implementing a standardized open-source radiomics python library, PyRadiomics 11, with default parameter configuration. The inputs to the radiomics algorithm were FLAIR MRI scans as “Images” and corresponding tumor segmentations as “Masks”. 13 of the image/mask combinations suffered from geometry mismatch, so these samples were discarded. Statistical information for five of the radiomic features extracted is shown in Table 1. A detailed list of the features extracted can be found in the PyRadiomics documentation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Textural Features** | | |
|  | **Variance (statistics)** | **Volume (shape)** | **Autocorrelation (GLCM)** | **Small Area Emphasis (GLSZM)** | **Short Run Emphasis (GLRLM)** |
| **count** | 150 | 150 | 150 | 150 | 150 |
| **mean** | 8643.94 | 11856.81 | 190.71 | 0.55 | 0.81 |
| **std** | 14328.38 | 14355.17 | 215.69 | 0.08 | 0.10 |
| **min** | 20.58 | 47.00 | 5.11 | 0.33 | 0.31 |
| **25%** | 2034.99 | 3274.75 | 67.45 | 0.50 | 0.75 |
| **50%** | 4273.07 | 7820.50 | 118.01 | 0.56 | 0.84 |
| **75%** | 8120.41 | 15919.75 | 240.79 | 0.61 | 0.88 |
| **max** | 98968.98 | 91299.00 | 1329.48 | 0.74 | 0.96 |

**Table 1**: Sample set of radiomic features derived via PyRadiomics for the 5 feature categories. GLCM = Gray Level Co-occurrence Matrix, Gray Level Size Zone Matrix = GLSZM, Gray Level Run Length Matrix = GLRLM are textural features.

The radiomic feature data and categorical survival data have been collated to yield the final dataset that was used for this project. The dataset contains 150 rows corresponding to the number of patients, and 95 columns corresponding to the radiomic features and the overall survival category (short-term: 0, or long-term: 1). The features are the input to our models while the overall survival category is the output of our models. A shortened version of the dataset with the first 5 samples showing the survival data and first 2 and last 2 features is shown in Table 2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Survival** | **Maximum 3D Diameter** | **Compactness 2** | **...** | **Zone Entropy** | **Small Area Low Gray Level Emphasis** |
| **0** | 0 | 72.36712 | 0.022687 | ... | 5.630367 | 0.017143 |
| **1** | 0 | 51.584882 | 0.032489 | ... | 5.748432 | 0.03035 |
| **2** | 0 | 63 | 0.038263 | ... | 6.093996 | 0.014043 |
| **3** | 1 | 34.655447 | 0.397477 | ... | 6.182728 | 0.00722 |
| **4** | 1 | 21.470911 | 0.076978 | ... | 4.051376 | 0.120535 |

**Table 2**: Shortened list of final dataset used for project.

**Exploratory Visualization**

One could intuit that larger tumor size may correlate to a worse prognostic outcome for GBM. The reality is not as clear-cut as our intuition would lead us to believe. To examine this concept, 2 radiomic shape features corresponding to tumor size (Volume and Maximum 3D Diameter) are plotted for each sample with corresponding survival data in different colors in figure 3. On average short-term survivors tend to have slightly larger volume and diameter when compared to long-term survivors. However, no obvious separation boundary exists between short-term and long-term survivors; survival classes are intermixed between high and low values for tumor volume and diameter. This highlights the fact that GBM is spatially heterogeneous and survival outcome would be difficult to predict from visual inspection of tumor size alone. The combination of statistic, shape, and texture based features should lead to the generation of a multi-dimensional decision boundary that can differentiate short term survivors from long term survivors.

Macintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:Volume_vs_3DDiamter.pdf

**Figure 3**: Relationship between shape features and survival classes.

**Algorithms and Techniques**

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
* *Are the techniques to be used thoroughly discussed and justified?*
* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

10 supervised machine learning classification algorithms were selected for examination. These algorithms were chosen in accordance with those studied in Paramer et al and are listed in column 1 of table X.

The feature selection method SelectKBest with 3 different scoring functions was utilized to select the top 10, 20, 30, and 40 features in a systematic fashion for each classifier. SelectKBest is a univariate feature selection method that uses a scoring function to assign values to each feature and removes all but the K highest scoring features from the feature set. It selects the top features that have the most relevance to the target variable, in this case survival outcome. The selection process is classifier independent so can be considered a filter method. The 3 scoring functions chosen are f\_classif which represents the ANOVA F-score (A), mutual\_info\_classif which represents mutual information (M), and chi2 which represents a chi-squared statistic (C). Each classifier was then trained using the selected feature subsets.

Hyperparameter tuning for each classifier/feature selection combination was accomplished through a stratified 5-fold cross validation grid search via GridSearchCV. GridSearchCV systematically performs an exhaustive search over a constructed hyperparameter grid to select the best hyperparameters for each classifier/feature selection method. Stratification creates folds that preserve the percentage of samples for each class and leads to models with lower bias and variance when compared to regular cross validation. The values of hyperparameters for the tuning process were chosen in accordance with Parmer et al and are shown in column 2 of table X. Classifiers without hyperparameters such as Gaussian Naïve Bayes and Quadratic Discriminant Analysis skipped the tuning step. It should be noted that in order to expedite computational time, some hyperparameter tuning values were shortened from Parmer et al. For example, the n\_splits parameter for Random Forrest in Parmer et al was set to values of 2 to 30 with step size of 3, but in our study we only utilized values of 2, 20, and 30.

|  |  |
| --- | --- |
| **Classifiers** | **Hyperparameters** |
| Random Forrest | n\_estimators: 500  min\_samples\_split: [2,20,30] |
| Gaussian Naïve Bayes | None |
| Decision Tree | min\_samples\_split: [2,20,30] |
| Multi Layer Perceptron | hidden\_layer\_sizes:[(100,),(1,),(9,)]  beta\_1: [0.9, 0.1, 0.0001] |
| Bagging | n\_estimators: [5,10,100] |
| Gradient Boosting | n\_estimators: [1,10,20,100] |
| Support Vector Machines | C: [0.1,1.0,10.0]  kernel: ['poly', 'rbf'] |
| Logistic Regression | C: [0.1,1.0,10.0],  penalty: ['l1','l2'] |
| K Nearest Neighbors | n\_neighbors: [5,10,15,20] |
| Quadratic Discriminant Analysis | None |

**Table 2**: List of classifiers used in this study (column 1) along with their hyperparameter tuning values (column 2).

**Benchmark**

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* *Has some result or value been provided that acts as a benchmark for measuring performance?*
* *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

Parmar et al. published a landmark study comparing fourteen feature selection methods and twelve classification methods in terms of their performance and stability for predicting overall lung cancer patient survival 7. This study contains AUC values for each of the feature selection and classification combinations, which can be directly compared with our projects AUC values. In addition, they developed a method for evaluating classifier stability utilizing the relative standard deviation, which can also be directly compared with our classifiers.

Though the models in Parmar et al. were trained using lung cancer computed tomography imaging and a different survival classification threshold (2 years instead of 1), it can be inferred that underlying radiomic principles are similar regardless of the tumor type studied or imaging methodology. Therefore, it is logical to predict our classification methods will follow comparable predictive performance trends as observed in this study. However, this may not necessarily be the case, as it has been shown in the past that different classifiers will work better for different cancer types and imaging modalities. It should be noted that Parmer et al also utilizes a larger training set than what is available to us (310 vs. 120), so our models may not be as accurate or generalizable as theirs.

**III. Methodology**

*(approx. 3-5 pages)*

**Data Preprocessing**

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
* *Based on the****Data Exploration****section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
* *If no preprocessing is needed, has it been made clear why?*

As previously discussed, 13 samples displayed geometry mismatches, leading to the generation of NaN values in the resulting dataset in place of feature values. These samples were removed from the dataset leading to a remaining total of 150 samples in the dataset. Furthermore, survival data was mapped from continuous values to categorical values of 0 and 1 for short and long-term survivors, respectively. Additionally, feature values were normalized, described in more detail in the implementation section. No outliers were removed from this dataset since the BraTS Challenge has ensured the correct segmentation and prognostic outcome of each patient, making every data point valuable for the machine learning process.

**Implementation**

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

The dataset of 150 samples, each with 94 features and a corresponding output class, was split into training and testing sets with a test size = 0.2, yielding a training set with 120 samples and a testing set with 30 samples. MinMaxScaler was used to normalize features with respect to the training set features and subsequently applied to the test set features. For each classifier, hyperparameters were tuned using GridSearchCV. 12 different feature selection combinations were applied to each classifier corresponding to the 3 scoring functions (A, M, C) with 10, 20, 30 or 40 top features selected. AUC scores were then computed for each of the classifier/feature selection combinations. Since 10 classifiers were investigated in this study, a total of 120 AUC scores corresponding to each classifier/feature selection combination were generated.

The entire process outlined above was iterated 10 times with different train test splits to determine the final AUC values for each classifier/feature selection combination by taking the mean over all iterations. The reason repeat iterations of train test split were necessary was because our models were observed to be somewhat unstable upon perturbation. By taking the mean over 10 iterations we are able to ensure a more representative value for each classifier/feature selection combination. Similarly, we obtain a final standard deviation value for each classifier/feature selection combination by taking the mean of the standard deviation over all iterations.

To obtain a final representative AUC value for a given classifier, the final AUC values for each classifier/feature selection combination were averaged for their respective classifier. Similarly, to obtain a final representative standard deviation value for a given classifier, the final standard deviation values for each classifier/feature selection combination were averaged for their respective classifier.

Relative standard deviation for each classifier was calculated by dividing the final representative standard deviation of the AUC by the final representative mean AUC and multiplying by 100 as described in METHODS.

**Refinement**

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

Initially only 4 classification methods described in Parmer et al were utilized. The number of classifiers used was increased to 10 in order to more closely model Parmer et al. Additionally, only untuned models with default hyperparemeters were initially implemented. Shortend versions of the tuning parameters from Parmer et al were implemented via k fold cross validation. Hyperparameter tuning increased classification accuracy of all applicable models.

**IV. Results**

*(approx. 2-3 pages)*

**Model Evaluation and Validation**

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

A shortened AUC value mean, median, and standard deviation list of the 10 iterations for each classifier/feature selection combination is shown in TABLE X. A full table for each classifier/feature selection combination can be found in the accompanying jupyter notebook.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Classifier** | **Feature Selection Method** | **Feature number** | **Mean** | **Median** | **Standard Deviation** |
| 0 | RandomForestClassifier | ANOVA F-score | 10 | 0.58 | 0.58 | 0.08 |
| 1 | GaussianNB | ANOVA F-score | 10 | 0.56 | 0.59 | 0.09 |
| 2 | DecisionTreeClassifier | ANOVA F-score | 10 | 0.59 | 0.59 | 0.07 |
| 3 | MLPClassifier | ANOVA F-score | 10 | 0.58 | 0.58 | 0.06 |
| 4 | BaggingClassifier | ANOVA F-score | 10 | 0.58 | 0.53 | 0.10 |
| 5 | GradientBoostingClassifier | ANOVA F-score | 10 | 0.58 | 0.59 | 0.11 |
| 6 | SVC | ANOVA F-score | 10 | 0.58 | 0.55 | 0.08 |
| 7 | LogisticRegression | ANOVA F-score | 10 | 0.57 | 0.56 | 0.06 |
| 8 | KNeighborsClassifier | ANOVA F-score | 10 | 0.57 | 0.58 | 0.05 |
| 9 | QuadraticDiscriminantAnalysis | ANOVA F-score | 10 | 0.50 | 0.51 | 0.08 |
| 10 | RandomForestClassifier | ANOVA F-score | 20 | 0.60 | 0.62 | 0.06 |
| … | … | … | … | … | … | … |
| 110 | RandomForestClassifier | Chi-sqr | 40 | 0.62 | 0.62 | 0.07 |
| 111 | GaussianNB | Chi-sqr | 40 | 0.51 | 0.50 | 0.06 |
| 112 | DecisionTreeClassifier | Chi-sqr | 40 | 0.56 | 0.55 | 0.07 |
| 113 | MLPClassifier | Chi-sqr | 40 | 0.57 | 0.57 | 0.06 |
| 114 | BaggingClassifier | Chi-sqr | 40 | 0.62 | 0.60 | 0.09 |
| 115 | GradientBoostingClassifier | Chi-sqr | 40 | 0.64 | 0.65 | 0.08 |
| 116 | SVC | Chi-sqr | 40 | 0.56 | 0.56 | 0.09 |
| 117 | LogisticRegression | Chi-sqr | 40 | 0.55 | 0.55 | 0.06 |
| 118 | KNeighborsClassifier | Chi-sqr | 40 | 0.54 | 0.57 | 0.08 |
| 119 | QuadraticDiscriminantAnalysis | Chi-sqr | 40 | 0.52 | 0.50 | 0.07 |

**Table 3**: AUC statistics for each classifier/feature selection method combination. Full list can be found in accompanying jupyter notebook.

FIG XA depicts a heatmap of the performance of feature selection in rows and classification methods in columns. The AUC value for each combination was obtained by averaging over the feature number variations, i.e. mean of 10, 20, 30, 40 top features. Similarly, FIG XB depicts a heatmap of the performance of feature number in rows and classification methods in columns where the AUC value for each combination was obtained by averaging over the feature method variations, i.e. mean of ANOVA, MI, CHI. The heatmaps demonstrate noticeable difference between classifiers but minimal difference upon altering the feature number or feature selection method.

Macintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:FSheatmap_6-6-17.pdfMacintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:FNheatmap_6-6-17.pdf

**Figure X**: (A) Feature Selection Method vs. Classifier, (B) Feature Number vs. Classifier.

By averaging over all feature selection combinations we obtain the final representative AUC value for each classifier. In addition RSD values for each classifier can be calculated as described in previous sections. A full table of these values can be found in the accompanying jupyter notebook. To help visualize predictive performance and stability of classifiers a scatterplot is shown in FIGX.

Macintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:Scatterplot_performance_RSD.pdf

**Figure X**: Scatterplot of AUC score vs. RSD score. Red area in upper right corner represents classifiers with high predicative performance and high stability.

It can be seen that Random Forest (AUC = 0.61, RSD = 11.6), Gradient Boosting (AUC = 0.62, RSD = 11.2), and Decision Tree (AUC = 0.59, RSD = 12.0) have predictive performance higher than median values (AUC = 0.56) and RSD lower than median values (RSD = 12.9). Therefore, these classifier methods should be considered to have relatively high predictive performance and stability when compared to the other classifiers studied. Gaussian Naïve Bayes (AUC = 0.50, RSD = 15.3) and Quadratic Discriminant Analysis (AUC = 0.51, RSD = 15.6) showed the worst predictive performance and stability with AUC values lower than the median and RSD values higher than the median. Some classifiers such as Bagging (AUC = 0.61, RSD = 14.6) showed predictive performance above the median AUC value but stability above the median RSD value, indicating high predictive performance but low stability. Oppositely, some classifiers such as Logistic Regression (AUC = 0.55, RSD = 11.7) and Multi Layer Perceptron (AUC = 0.55, RSD = 12.6) demonstrate predictive performance below the median AUC value and stability above the median RSD value, indicating low predicative performance but high stability.

**Justification**

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

Overall, classifiers were shown to have lower predictive performance and higher RSD values when compared to their counterparts in Parmer et al. For example, Random Forrest was shown to have an AUC and RSD of 0.66 and 3.5 respectively in Parmer et al, which is higher in predictive performance and stability when compared to our Random Forrest implementation. This is possibly due to a smaller training size utilized in our study when compared to Parmer et al, i.e. 120 samples in our study vs. 310 samples in Parmer et al. Random Forrest was shown to be one of the best classifiers in Parmer et al and this held true for our data as well. Interestingly, some of the classifiers demonstrate the opposite behavior that was observed in Parmer et al. For example, Naïve Bayes was shown to be one of the best classifiers in terms of both predictive performance and stability in Parmer et al but was shown to be the worst for both these values in our study. Additionally, Decision Tree and Boosting were shown to have relatively poor predictive performance and stability in Parmer et al, but were among the best models in our study. This potentially highlights the idea that the best classification methods for radiomics studies are highly dependent on the imaging modality and the type of tumor being studied.

**V. Conclusion**

*(approx. 1-2 pages)*

**Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

In this study there are three experimental factors that may affect the prediction of radiomics based survival outcomes: classification method, feature selection method, and number of features used. In addition, the interactions between these factors also have a role in predictive performance. Multivariate analysis of variance (ANOVA) was performed to determine the variability in AUC scores contributed by each of the experimental factors and their interactions. The AUC scores used for each classifier/feature selection combination were the mean values after 10 iterations of the train test split procedure TABLE X. In order to compare the variability contributed to predictive performance by each factor, the variance (sum of squares) calculated for each factor was divided by total variance and multiplied by 100 to yield the percent variance for each factor (FIG X).

Macintosh HD:Users:kareemwahid:Desktop:Brats17TrainingData:Variation_graph_6-8-17.pdf

**Figure X**: Variation of AUC explained by the experimental factors and their interactions.

It can be clearly observed that classification method had the largest contribution to variability accounting for 75.5 % of the total variance in AUC scores with a corresponding p-value of 2.5e-25 indicating it was a statistically significant factor contributing to variability (p < 0.05). Feature selection method accounted for 4.9 % of the variability with a corresponding p-value of 9.2e-7 indicating it was a statistically significant factor contributing to variability (p < 0.05). Finally, the number of features used in a model accounted for the least of the total variance at 0.2 % with a p-value of 7.0e-1, indicating it was not a statistically significant factor contributing to variability (p > 0.05). Interactions between factors followed similar trends. Classifier: feature number, feature selection: feature number, and classifier: feature selection accounted for 4.2, 0.7, and 7.3 % of the variability respectively with corresponding p-values of 3.2e-1, 5.2 e-1, and 9.6e-4 respectively. Since we used many more classifier methods than any other experimental factor these results could be anticipated but were still interesting to see quantified. The results are in accordance with Parmer et al where a similar trend was observed.

**Reflection**

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Several classifier methods/feature selection combinations were evaluated for predictive performance and stability in GBM survival outcome prediction using AUC and RSD values. Heatmaps show that the feature selection method and number of features used did not contribute to predictive performance as significantly as the classifier used; ANOVA tests demonstrate classifier identiy as accounting for 75.5 % of the total variance in AUC scores. The best classifiers for predicting GBM survival outcome were Random Forest, Gradient Boosting, and Decision Trees, while the worst classifiers were Gaussian Naïve Bayes and Quadratic Discriminant Analysis. Upon comparison with the results of Parmer et al., classifier performance is suggested to be highly dependent on the tumor type and imaging modality used in radiomic analysis. This is an interesting result that highlights the need for independent measurements on classifier performance for different types of tumors and imaging modalities.

**Improvement**

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Hyperparameters can be further tuned to improve classifier performance, though it should be noted that for some classifiers such as Random Forest, parameter tuning can lead to computational slow downs. In addition, additional classifiers can be studied to find if any have higher predictive performance for this set of data then Random Forest and Boosting. Finally, more feature selection methods, or alternatives to feature selection, such as dimensionality reduction, could be implemented to see how it compares with our results. It is apparent that only a few features contribute significantly to the variability of the data, so PCA could be an attractive alternative to SelectKBest feature selection. These methods should be considered for future studies.