# Early Detection and Diagnostic Insights

CSBP 4502 – Group 7

Carolina Perez
University of Colorado Boulder
Boulder CO USA
cape5274@colorado.edu

Cody Folgmann
University of Colorado Boulder
Boulder CO USA
cody.folgmann@colorado.edu

Dain Kim
University of Colorado Boulder
Boulder CO USA
dain.kim@colorado.edu

Cancer survival outcomes have been a major focus of medical research, particularly in early detection and diagnostic technologies that aim to improve patient prognosis. Clinical and genetic data are often used to identify patterns in cancer diagnosis, but predicting survival rates based on early detection remains a significant challenge. This project aims to bridge this gap by investigating how early detection factors such as tumor size, biopsy results, and patient demographics can be linked to improved survival rates. By identifying and analyzing these key variables, we aim to create a predictive model that assesses cancer survival probabilities, enabling healthcare professionals to make better-informed decisions. This has the potential to lead to earlier interventions, better patient outcomes, and enhanced patient education.

The knowledge we apply in this research comes from various fields, including data analysis, machine learning, and healthcare studies. We plan to utilize machine learning techniques to process and analyze clinical data, such as tumor size and biopsy results, along with patient demographics like age, gender, and overall health. These factors can have a profound impact on survival and integrating them into a robust model can help predict survival probabilities and offer insights into the relationship between early detection and patient outcomes.

By understanding how early detection factors correlate with survival, we hope to guide better treatment strategies and inform healthcare practices. Through this work, we aim to contribute to improving decision-making in cancer treatment, enabling healthcare professionals to personalize care for their patients. Ultimately, this research can provide critical insights that support the development of more accurate and efficient diagnostic methods and treatment plans for cancer patients, potentially leading to better long-term outcomes and a higher quality of life.Furthermore, the development of a reliable predictive model can assist healthcare professionals in personalizing treatment plans based on individual risk factors. This will not only benefit patients but also streamline the diagnostic process, making it more efficient and accurate.

## Dataset:

We are using the **Brain Tumor Prediction Dataset** from Kaggle (URL:https://www.kaggle.com/datasets/ankushpanday1/brain-tumor-prediction-dataset), which contains approximately 250,000 data points across 22 attributes. The dataset includes clinical and demographic data relevant to our analysis.

We chose the Radom Forest Classifier algorithm for this project due to its ability to handle non-linear complex relationships in this medical dataset. Our data set has different attributes such as Tumor Size, Genetic Rish and MRI Findings which may have nonlinear correlations with attributes related to the outcome such as Survival Rate or Tumor Presence. The random Forest model can also handle categorical and continuous variables which is helpful for our dataset where both variable types are present.

Although the Random Forest model is a strong choice, we have to consider others model for comparisons such as k-Nearest Neighbors. KNN involves classification and imputation of missing values. This model works by finding the closest "neighbor" data points to the missing values and depending on the variable types will replace it using the average or the most frequent value in that attribute. When comparing this to Random Forest, KNN should be more sensitive to noisy features, whereas Random Forest is less affected by irreverent features due to its ability to perform feature selection during its model training. KNN could also be computationally expensive, especially with large datasets since its process involves calculating the distance between all pairs of data points, the distance and the number of neighbors chosen.

In our original proposal feedback it was suggested that we investigate Cox Proportional Hazards models to assist with our survival analysis. This model accounts for tie-to-event data in order to predict the time until an outcome occurs such as in our case survival rate Some variables that it may be useful to use this model for include:

- Tumor Size: This is a continuous variable as the size of a tumor may indicate the risk of poor or good survival. It is directly related to the severity of the condition.
- Tumor location: this is a categorical variable and may influence the rate of survival due to their location to crucial areas.

Key variables that we would want to be included in the Cox Proportional Hazards model would be those that are well known to influence survival outcomes such as the two listed above, however, arguments can be made for additional variables as well.

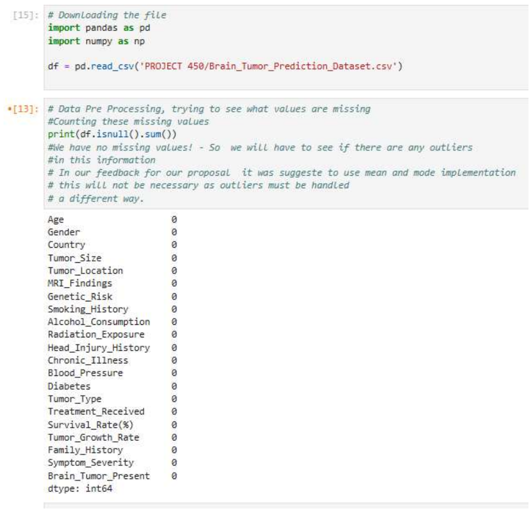
Common issues in medical datasets are class imbalances, where the number of survivors will outweigh the non-survivors. This can lead to inaccuracy for the minority class of predictions. The Random Forest model can handle these imbalances by aggregating multiple decision trees. Evaluation methods can also be implemented such as F1-scor to better asses the model's precision in predicting the minority class of non-survivors.

Cross-validation is a technique used in machine learning to assess the performance of a model by splitting the data into multiple subsets or folds. In k-fold cross-validation, the dataset is divided into 'k' equal parts, where the model is trained on 'k-1' folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the test set once. Cross-validation helps ensure that the model performs well on unseen data, providing a more reliable estimate of its generalization ability and preventing overfitting. It also gives us a better understanding of how the model will perform in real-world scenarios.

In the context of the cancer survival prediction model, handling feature importance is essential to interpret the results and understand which variables contribute most to survival prediction. Random Forests, which are well-suited for this type of analysis, can automatically rank the importance of each feature based on how much they reduce the impurity during the model's decision-making process. By identifying the most influential features, such as tumor size, biopsy results, and patient demographics, we can better understand which factors are critical for survival outcomes. This information is valuable for clinicians, as it helps prioritize the most important risk factors and informs decisions regarding treatment options and patient care. Feature importance not only aids model interpretability but also helps simplify the model by allowing practitioners to focus on the most significant attributes.

## Milestones Completed – Data Processing

It is likely that with this dataset coming from Kaggle, the majority of it has already been cleaned and potentially missing attributes have already been accounted for. However, in model building and processing it is crucial to not assume that the data is prepared without several verification steps. Assumption could lead to errors during model processing, especially when using algorithms that may be sensitive to missing values or inconsistencies. such as Random Forest. Verification assures the reliability of our model and its predictions, and accuracy.



```
[15]: # Downloading the file
      import pandas as pd
      import numpy as np

      df = pd.read_csv('PROJECT 450/Brain_Tumor_Prediction_Dataset.csv')
```

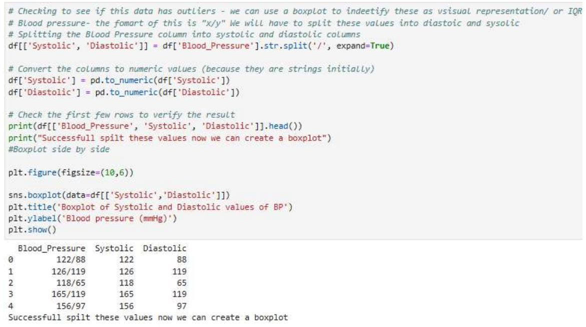
```
•[13]: # Data Pre Processing, trying to see what values are missing
       #Counting these missing values
       print(df.isnull().sum())
       #We have no missing values! - So  we will have to see if there are any outliers
       #in this information
       # In our feedback for our proposal  it was suggeste to use mean and mode implementation
       # this will not be necessary as outliers must be handled
       # a different way.
```

```
Age                    0
Gender                 0
Country                0
Tumor_Size             0
Tumor_Location         0
MRI_Findings           0
Genetic_Risk           0
Smoking_History        0
Alcohol_Consumption    0
Radiation_Exposure     0
Head_Injury_History    0
Chronic_Illness        0
Blood_Pressure         0
Diabetes               0
Tumor_Type             0
Treatment_Received     0
Survival_Rate(%)       0
Tumor_Growth_Rate      0
Family_History         0
Symptom_Severity       0
Brain_Tumor_Present    0
dtype: int64
```

Handled missing values: In previous feedback, it was suggested to use mode and mean imputation for handling missing values in our dataset. This method involves replacing missing numerical values with the average value of the data collected for that attribute. Mode imputation involves replacing missing categorical values with the most frequent category. For this data set, once importing it into python as a csv file, the check for missing values was done using the ”isnull()” function on each attribute to get an overall sum of the missing values. The result of this shows that in our data preprocessing these is no need for mode or mean implementation that needs to be added into our preprocessing process.

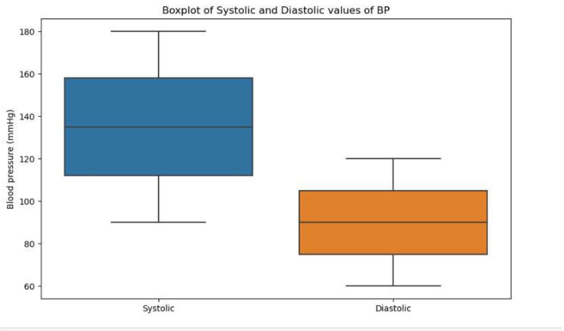
Identifying any Outliers Present: Identifying outliers is a crucial step in data preprocessing, as these extreme values can significantly impact the performance of machine learning models, including Random Forests. Outliers are data points that deviate significantly from other observations, and if not handled properly, they can skew the results, leading to inaccurate predictions. In the context of Random Forests, outliers can affect the splitting criteria of decision trees, potentially causing them to overfit or underfit the data. By identifying and addressing outliers, we can ensure that the model focuses on the overall patterns in the data, rather than being overly influenced by extreme or irrelevant values. This helps to improve the generalization ability of the model, leading to better predictive accuracy and more reliable insights from the dataset. Thus, detecting and managing outliers is an essential preprocessing step that enhances the performance of the Random Forest algorithm.

Example - Blood Pressure:

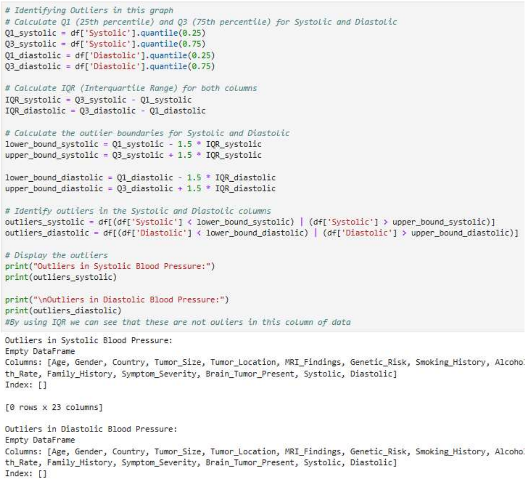


```
# Checking to see if this data has outliers - we can use a boxplot to indeetify these as vsisual representation/ or IQR
# Blood pressure- the fomart of this is "x/y" We will have to split these values into diastoic and sysolic
# Splitting the Blood Pressure column into systolic and diastolic columns
df[['Systolic', 'Diastolic']] = df['Blood_Pressure'].str.split('/', expand=True)

# Convert the columns to numeric values (because they are strings initially)
df['Systolic'] = pd.to_numeric(df['Systolic'])
df['Diastolic'] = pd.to_numeric(df['Diastolic'])

# Check the first few rows to verify the result
print(df[['Blood_Pressure', 'Systolic', 'Diastolic']].head())
print("Successfull spilt these values now we can create a boxplot")
#Boxplot side by side

plt.figure(figsize=(10,6))

sns.boxplot(data=df[['Systolic','Diastolic']])
plt.title('Boxplot of Systolic and Diastolic values of BP')
plt.ylabel('Blood pressure (mmHg)')
plt.show()
```

```
  Blood_Pressure  Systolic  Diastolic
0         122/88       122         88
1        126/119       126        119
2         118/65       118         65
3        165/119       165        119
4         156/97       156         97
Successfull spilt these values now we can create a boxplot
```



Now that we have a boxplot of this attribute, we can use IQR to identify any outliers that exists in this data. The Interquartile Range (IQR) measures statistical dispersion, representing the range between the first quartile (Q1) and the third quartile (Q3) of a dataset. Data points outside the range of Q1 - 1.5 * IQR and Q3 + 1.5 * IQR are considered outliers. Identifying and handling outliers is important to ensure they don't distort model performance. We can easily do this in Python:



```
# Identifying Outliers in this graph
# Calculate Q1 (25th percentile) and Q3 (75th percentile) for Systolic and Diastolic
Q1_systolic = df['Systolic'].quantile(0.25)
Q3_systolic = df['Systolic'].quantile(0.75)
Q1_diastolic = df['Diastolic'].quantile(0.25)
Q3_diastolic = df['Diastolic'].quantile(0.75)

# Calculate IQR (Interquartile Range) for both columns
IQR_systolic = Q3_systolic - Q1_systolic
IQR_diastolic = Q3_diastolic - Q1_diastolic

# Calculate the outlier boundaries for Systolic and Diastolic
lower_bound_systolic = Q1_systolic - 1.5 * IQR_systolic
upper_bound_systolic = Q3_systolic + 1.5 * IQR_systolic

lower_bound_diastolic = Q1_diastolic - 1.5 * IQR_diastolic
upper_bound_diastolic = Q3_diastolic + 1.5 * IQR_diastolic

# Identify outliers in the Systolic and Diastolic columns
outliers_systolic = df[(df['Systolic'] < lower_bound_systolic) | (df['Systolic'] > upper_bound_systolic)]
outliers_diastolic = df[(df['Diastolic'] < lower_bound_diastolic) | (df['Diastolic'] > upper_bound_diastolic)]

# Display the outliers
print("Outliers in Systolic Blood Pressure:")
print(outliers_systolic)

print("\nOutliers in Diastolic Blood Pressure:")
print(outliers_diastolic)
#By using IQR we can see that these are not ouliers in this column of data
```

```
Outliers in Systolic Blood Pressure:
Empty DataFrame
Columns: [Age, Gender, Country, Tumor_Size, Tumor_Location, MRI_Findings, Genetic_Risk, Smoking_History, Alcoho
th_Rate, Family_History, Symptom_Severity, Brain_Tumor_Present, Systolic, Diastolic]
Index: []

[0 rows x 23 columns]

Outliers in Diastolic Blood Pressure:
Empty DataFrame
Columns: [Age, Gender, Country, Tumor_Size, Tumor_Location, MRI_Findings, Genetic_Risk, Smoking_History, Alcoho
th_Rate, Family_History, Symptom_Severity, Brain_Tumor_Present, Systolic, Diastolic]
Index: []
```

The empty Data Frames confirm that there are no outliers in this attribute. This process was repeated with the other numerical variables in our dataset to check for outliers that may mess with our model.

## Initial Results

Initial analysis showed that blood pressure, tumor size, age, and genetic risk factors had the greatest impact on survival, and correlation insights highlighted the importance of early detection and the need for timely intervention to increase the chances of survival, under the assumption that the earlier a tumor is detected, the better chance there is of it being small. By leveraging data modelling and processing, we were able to better understand the relationships between variables and their predictive potential.

Preliminary results were used with various python tools/functions to create visualizations of the relationships between predictive factors and tumor presence or patient survival. Survival percentage shows a large correlation with tumor presence but the two are not the most useful predictors of each other. Survival percentage is not available in advance of treatment or postmortem, and while tumor presence can be determined before beginning treatment to use in survival estimates, it generally requires deliberate medical diagnosis. One of the key purposes of this model is rather to predict either of those features using demographic data or patient history to save time and resources.

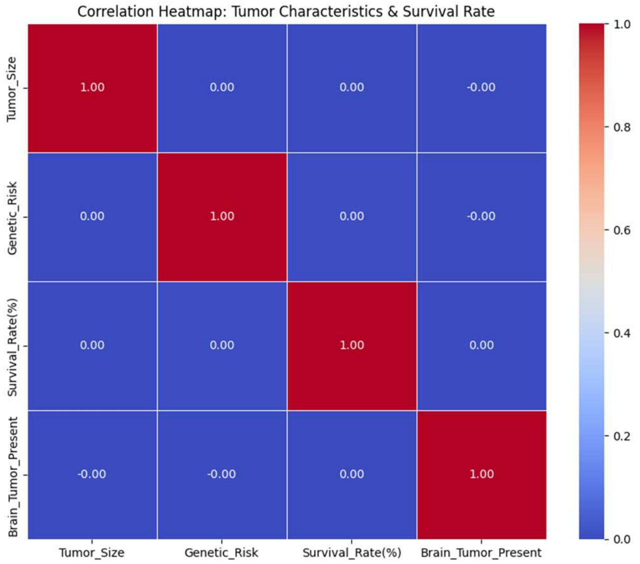
Ignoring the survival percentage factor in predicting tumor presence (and vice versa), as well as tumor size for similar reasons, we see the greatest importance values assigned to blood pressure, genetic risk, and age. If we're focusing on survival percentage and tumor diagnostic information is available, tumor size becomes a highly important factor. Growth rate and symptom severity show a surprisingly low importance value in our initial runs.

Class imbalances remain a significant risk for our dataset and will be the next focus of investigation to determine if further preprocessing is needed.

## Visualization

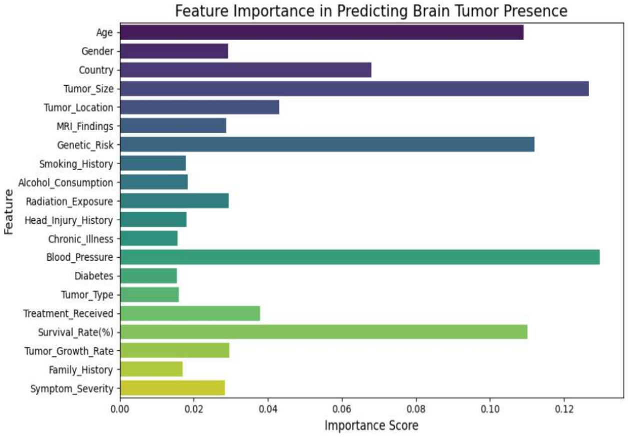
Data visualization techniques were used to more easily inspect factor correlation and suggest where we might need to improve or alter our model or data preprocessing.

A first investigation was performed using the seaborn heatmap function. This generated a correlation matrix, which unfortunately showed low correlation for all factors, a small section of which is shown in Figure 1, with all values less than +/- 0.01.



[ Figure 1: Feature Importance Score Heatmap ]

Using a Random Forest Classifier, we can extract features of importance that indicate how much each variable contributes to predicting the presence of a brain tumor. It visualizes plots with color palette to highlight features based on their importance for clarity. Despite the low correlation factors shown in our heatmap, this shows that individual factors have significantly different importances within that low correlation.

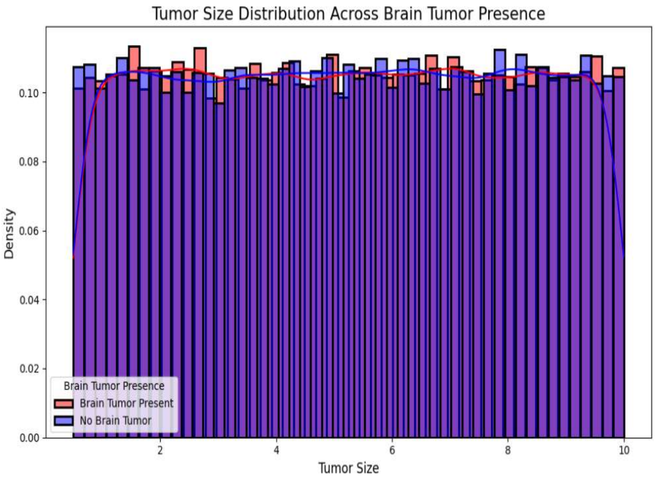


[ Figure 2: Feature Importance Scores ]

Finally, distribution plots provide insight into the prevalence of specific factors given another factor. Figure 3 shows the distribution of tumor size based on the presence or absence of brain tumors. These factors seem counterintuitive (how could you have a size for tumors which aren't present?) which needs further investigation as we continue. This may be categorizing tumors in a general sense vs "brain tumors" in a way that is not explained in the dataset itself, or it may be whether brain tumors existed at the initial diagnosis. A lack of medical knowledge here limits our ability to apply human knowledge to the dataset.

The analysis leading to Figure 3 focuses on comparing the distribution of tumor sizes for two classes: those with and without brain tumors. Using the histogram function in Seaborne library, we plot two separate distributions: red for samples marked 'with brain tumor' and blue for samples marked 'without brain tumor'. Each distribution includes a kernel density estimation (KDE) overlay to smooth and highlight the underlying patterns in the data. KDE allows for a more continuous and intuitive representation of the data, making it easier to compare two classes. This is especially useful for identifying subtle differences in the distribution.

This visualization shows potential differences in the tumor size distribution between the two groups, such as changes in central tendency, spread, or peak density. These differences can help identify whether tumor size plays a role in diagnosing the presence of a brain tumor. The plots are enhanced with descriptive labels, titles, and legends to ensure clarity and interpretability. Using this method, you can gain a deeper understanding of how tumor size differs between classes, providing a foundation for generating additional hypotheses or incorporating them into a predictive model.



[ Figure 3: Tumor Size Distribution ]

## Milestones Remaining

In the remaining weeks, we will continue to refine our methodology and optimize model performance, hopefully uncovering strong correlations or insights into cancer outcomes and laying the groundwork to finalize and submit our study.

1. Feature selection (21 March): Finalize the selection of key features for modelling.
    a. Our preliminary analysis suggests 3-4 key factors that will be our primary candidates going into final selection: blood pressure, tumor size, genetic risk, and age
    b. For the initial intent of predicting patient survival, tumor size remains a key factor. However, the visualizations above suggest that there is an interesting second function we can predict from this dataset which is the presence or absence of tumors based on demographic data that does not require medical resources to gather
2. Model implementation (4 April): Train and validate the random forest classifier for the selected features.
    a. The random forest classifier used in our preliminary analysis will be refined and tuned to focus on the features identified in the previous milestone, and outcomes re-evaluated at each iteration

3. Model evaluation (18 April): Evaluate model performance through cross-validation and refine hyperparameters.

4. Final model selection and report (25 April): Make a final selection of the best performing models and document the results.