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

Abstract

Our data mining project is to evaluate publicly available cancer diagnosis and treatment records to determine if there are useful correlations between demographic information and likely treatment outcomes. We additionally evaluated the effects of tumor characteristics from the same dataset to see if there were additional linkages to the demographic information or disease outcomes, since it was available from the same source.

Our model found the strongest useful correlations between blood pressure, genetic risk, and age with survival rates. These correlations are relatively intuitive, but none of the attributes scored noticeably high when performing a correlation heatmap. It was only by comparing them against the other similarly low-scoring correlations that we were able to select these as the most important.

While initial correlation heatmaps did not highlight any single attribute as highly predictive, deeper analysis revealed that blood pressure, genetic risk, and age consistently showed the strongest associations with survival outcomes. These findings, though intuitive, were only evident when comparing their relative strength against other low-scoring variables.

After training the Random Forest Classifier model, plotting the Feature Importance shows a more useful comparison illustrating the relatively high importance of the patient's blood pressure, genetic risk, and age on the survival outcomes.

Introduction

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.

Related Work

There is an extensive list of models involving diverse types of cancer and cancer detection methods:

1. *Breast Cancer Survival Prediction Model*

This model focuses on predicting survival rates for breast cancer patients by analyzing clinical and genetic data. Its goal was to assess the likelihood of survival based on factors such as tumor size, genetic mutations, and overall health history of the patient and is aimed at helping medical professionals make informed decisions about treatment plans, diagnosis, and patient education on early detection. It could be a helpful tool towards patients because it offers a clear picture of their prognosis and helps create a bridge between genetic and clinical data.

2. *Lung Cancer Survival Prediction Model*

This model was developed in 2021 by Y Wu, and it emphasizes the importance of early detection for improving survival rates in lung cancer patients. Factors in this data include Age, smoking history and tumor characteristics are used to predict survival rate for these patients. The main goal of this model is to emphasize early detection and screening for patients and is aims for those patients who are at elevated risk for this – those with a smoking history or family history and could be a great resource for patient education.

3. *Socio-Economic Factors and Cancer Survival*

This model considers socio–economic factors and their impact on cancer survival rates such as: patient age, income, education, and healthcare access. This model is used to show the correlation between the patient socio-economic factors and their chance of survival and provides a different viewpoint of how these factors impact chance of survival compared to the usual factors different studies and models show – clinical and genetic data. This model could be used when discussing public health and policy decisions to improve health care and treatment options and accessibility with cancer patients of diverse backgrounds

Dataset

For this project we are using the Brain Tumor Prediction Dataset, containing approximately 250,000 data points and 22 attributes. These attributes include clinical data such as tumor size, genetic risks, patient symptoms, and some socio-economic details such as patient lifestyle and health habits, allowing for the explorations of both biological and environmental influences on survival outcomes.

URL:
https://www.kaggle.com/datasets/ankushpanday1/brain-tumor-prediction-dataset

This dataset provides a realistic representation of medical data that can be used for predictive modeling and survival analysis. It includes key information about tumor location, growth rate, and survival rate, offering comprehensive insights into the factors that influence brain tumor progression and patient outcomes. Additionally, it allows for diverse and inclusive analysis of how different patient demographics and environmental factors impact tumor survival. The data's scale and scope make it particularly suitable for classification tasks and

detailed survival analysis models that require large balanced samples for training and testing. This versatility enhances its values for a wide range of datamining technologies

Main Techniques Applied

We chose the Random 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 Risk 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.

We did not elect to change our model to a Cox Proportional Hazard version, as our model showed good reliability in our cross-validation and performance evaluation scores.

Common issues in medical datasets are class imbalances, where to 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 in model interpretability but also helps simplify the model by allowing practitioners to focus on the most significant attributes.

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
```

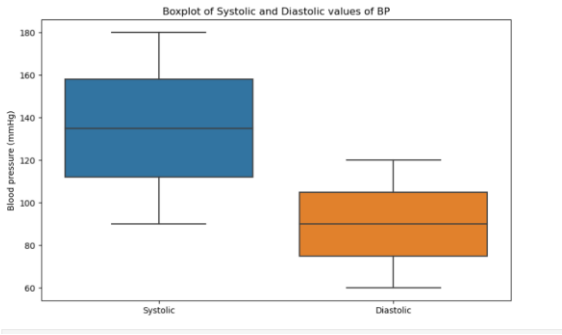
Handled missing values: In previous feedback, it was suggested to use mode and mean imputation for handling missing values in our dataset. This method involves replacing missing numerical values with the average value of the data collected for that attribute. Mode imputation involves replacing missing 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 there 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 diastolic 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 exist 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.

Key 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. These findings suggest that easily accessible health indicators may be more predictive than some of the more complex or traditionally emphasized clinical variables.

Applications

Applications in early detections : This model identifications of key features such as blood pressure,

genetic risk and age as significant predictors of survival have direct implications for early cancer diagnosis. There are all attributes that can be obtained through routine screening that are non-invasive to the patient. This enables healthcare providers to identify high risk individuals even before tumor detection. If this mode is integrated into routine checkups or electronic health records system, these high-risk patients could be flagged for earlier diagnostic imaging or genetic counselling, potentially improving early detection rates and survival outcomes.

Survival percentages and tumor size often require extensive diagnosis or are only confirmed post treatment. Being able to estimate survival likelihood from demographic and baseline health data could help decision making in resource limited settings. Hospitals or clinics with constrained imaging or biopsy capacity could prioritize patients flagged by the model as high risk, ensuring faster intervention for those patients.

Using this project and creating an understanding in the predictive role of blood pressure, age, genetic risk creates an opportunity to improve public and patient education. Awareness could focus on the importance of managing blood pressure and knowing one's family history as part of cancer risk assessments. Additionally, personalizing preventive strategies based on these predictors could empower individuals to seek earlier screenings or lifestyle adjustments, fostering a more proactive approach to cancer prevention and early detection.

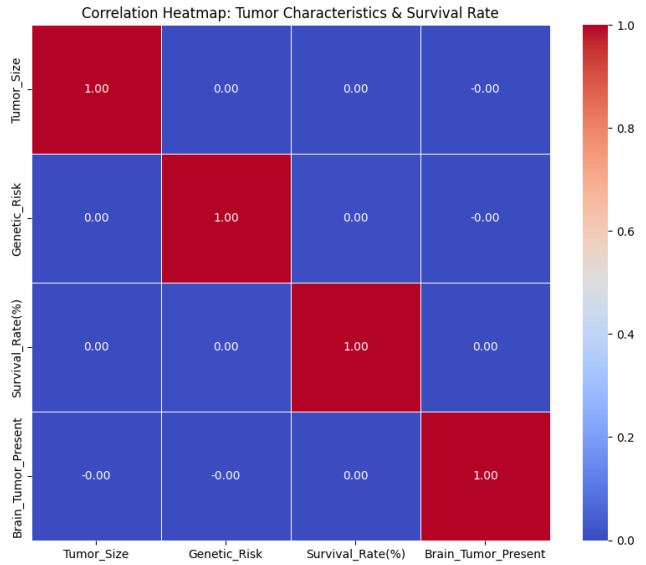
Visualization

Data visualization is a fundamental step in any data-driven project. It enables a deeper understanding by graphically representing data distributions, relationships, and trends, revealing insights that simple numerical values alone cannot provide. In machine learning and data analysis, visual exploration is crucial for detecting outliers, understanding interactions between features, identifying data imbalances, and verifying whether the basic assumptions for modelling approaches are met. Effective visualizations not only support analysts and data scientists during the model development process but also enhance communication with non-technical stakeholders by making patterns, correlations, and results accessible and intuitive. In medical applications, where datasets are often complex and sensitive, visualization plays an additional role in validating domain knowledge and increasing the interpretability of machine learning results, thereby building greater trust in the models.

In the context of our project, visualization was fundamental to exploring key features such as Tumor Size, Genetic Risk, and Systolic Blood Pressure. These visual tools not only facilitated more informed feature selection but also suggested potential areas for data preprocessing refinement. Ultimately, visualization guided the development of a robust, clinically meaningful predictive model, aligning with our main objective of building a reliable and interpretable brain tumor detection system.

The first investigation was performed using the seaborn heatmap function. This generated a correlation matrix, which unfortunately showed 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. In the context of our project, visualization was essential for exploring key features such as Tumor Size, Genetic Risk, and Systolic Blood Pressure.
low correlation for all factors, a small section of which is shown in Figure 1, with all values less than +/- 0.01.

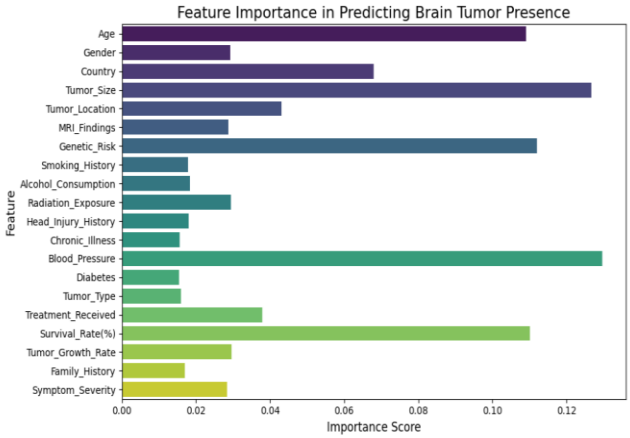
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Load dataset
df = pd.read_csv("Brain_Tumor_Prediction_Dataset.csv")

# Convert categorical variables to numerical where necessary
df["Brain_Tumor_Present"] = df["Brain_Tumor_Present"].map({"Yes": 1, "No": 0})

# --- 1. Heatmap: Correlation between tumor characteristics and survival rates ---
plt.figure(figsize=(10, 8))
corr_matrix = df[["Tumor_Size", "Genetic_Risk", "Survival_Rate(%)", "Brain_Tumor_Present"]].corr()
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap: Tumor Characteristics & Survival Rate")
plt.show()
```



[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.

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder


#Load dataset
df = pd.read_csv("Brain_Tumor_Prediction_Dataset.csv")

df["Brain_Tumor_Present"] = df["Brain_Tumor_Present"].map({"Yes": 1, "No": 0})

df_encoded = df.copy()

# Encode categorical columns
categorical_cols = df_encoded.select_dtypes(include=['object']).columns
for col in categorical_cols:
    df_encoded[col] = LabelEncoder().fit_transform(df_encoded[col])

X = df_encoded.drop(columns=["Brain_Tumor_Present"])
y = df_encoded["Brain_Tumor_Present"]

# Train RandomForest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X, y)

# Extract feature importances
importances = model.feature_importances_
feature_names = X.columns

# Plot feature importance
plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=feature_names, palette="viridis")
plt.title("Feature Importance in Predicting Brain Tumor Presence", fontsize=16)
plt.xlabel("Importance Score", fontsize=12)
plt.ylabel("Feature", fontsize=12)
plt.tight_layout()
plt.show()


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

# Plot tumor size distributions for both classes (Tumor Present and Tumor Absent)
sns.histplot(df[df["Brain_Tumor_Present"] == 1]["Tumor_Size"], kde=True, color="red", label="Brain Tumor Present", stat="
sns.histplot(df[df["Brain_Tumor_Present"] == 0]["Tumor_Size"], kde=True, color="blue", label="No Brain Tumor", stat="dens

# Add titles and labels
plt.title("Tumor Size Distribution Across Brain Tumor Presence", fontsize=16)
plt.xlabel("Tumor Size", fontsize=12)
plt.ylabel("Density", fontsize=12)
plt.legend(title="Brain Tumor Presence")
plt.tight_layout()
plt.show()
```



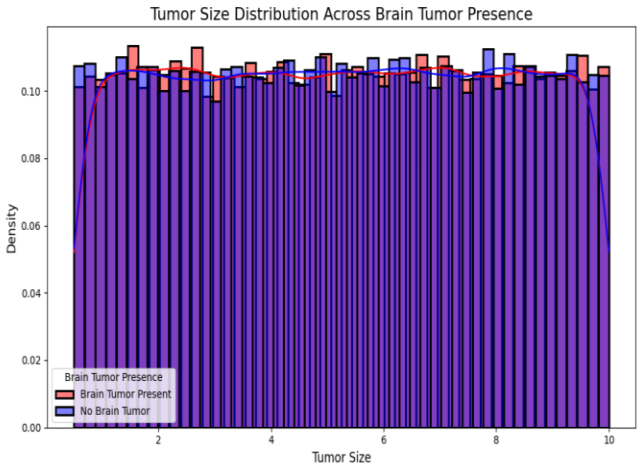
[Figure 2 Feature Importance Scores]

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]

Data Modeling

The Random Forest Classifier was selected as the primary model for this project based on its robustness, ability to handle structured tabular data, and strong interpretability — all critical factors in medical decision-making contexts.

Random Forest is an ensemble method that constructs multiple decision trees during training and outputs the majority class prediction of the individual trees. This ensemble approach significantly reduces variance compared to a single decision tree, improves generalization, and minimizes overfitting — making it well-suited for datasets of moderate size and complexity, such as ours.

Given the structured nature of the data — with features like Age, Tumor Size, Genetic Risk, and Systolic Blood Pressure — Random Forest can capture nonlinear interactions between features without requiring heavy preprocessing or feature engineering.

Moreover, Random Forest models provide feature importance scores, allowing clinicians and researchers to understand which variables most influence tumor prediction, enhancing the model's explainability

To evaluate the model's generalization ability on unseen data, we split the dataset into training (70%) and testing (30%) subsets. The train-test split allows us to train the model on one part of the data and evaluate it on a separate set that simulates new, unseen patient records. This is crucial for avoiding overfitting and evaluating real-world model performance. Additionally the random_state parameter has been modified during partitioning and model training. By setting a random seed (random_state=42), the split and subsequent model behavior can be reproduced. This is very important in

scientific research, where transparency and reproducibility of results are required for peer review and further validation.

```
from sklearn.model_selection import train_test_split

# Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

# Fit the model to the training data
rf_model.fit(X_train, y_train)
```

In our analysis, variables such as Tumor Size and Genetic Risk emerged as highly influential in predicting the presence of a brain tumor. This finding aligns with clinical expectations, as tumor size is a direct indicator of disease severity, and genetic predispositions significantly increase cancer risk.

Meanwhile, features like Age and Systolic Blood Pressure also contributed, though to a lesser extent, providing a comprehensive view of the patient's health status. Feature importance analysis not only validates known medical knowledge but also provides an opportunity to uncover new associations or risk factors that may merit further investigation.

Thus, integrating explainable machine learning approaches like Random Forest supports both model performance and clinical accountability, offering a bridge between data science and real-world healthcare outcomes.

Model Evaluation

To ensure the reliability and generalization of our model, we began the evaluation process by implementing K-fold cross-validation. In predictive modeling, relying solely on a single train-test split can be misleading, as model performance may vary significantly depending on how the data is divided. Cross-validation mitigates this risk by systematically splitting the dataset into multiple folds, training the model on different subsets, and testing it on the unseen fold across multiple iterations. This approach provides a more robust and unbiased estimate of the model's true performance, helping to detect overfitting and ensuring that the model performs consistently across various subsets of the data[5].

Beyond cross-validation, we employed several key evaluation metrics to comprehensively assess model performance. The confusion matrix allowed us to visualize the distribution of true positives, true negatives, false positives, and false negatives — particularly critical in medical applications where misclassification can have serious consequences. We also generated a classification report, summarizing crucial indicators such as precision (the proportion of correct positive predictions), recall (the ability to identify all true positives), and the F1 score (the harmonic mean of precision and recall). Each metric provides valuable insights: while high precision reduces false positives, high recall ensures critical cases like brain tumors are not missed, which is essential in healthcare risk assessment [6].

In this project, we specifically implemented 5-fold cross-validation, splitting the data into five equal parts. During each iteration, the model was trained on four folds and tested on the remaining one. We repeated this procedure five times to ensure that every fold was used as a test set exactly once. By averaging the evaluation scores across the five iterations, we were able to obtain a more reliable and generalized measure of model performance, mitigating the risk of both overfitting and underfitting. Python code can be as below:

```
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import cross_val_score

# Predictions
y_pred = rf_model.predict(X_test)

# Confusion Matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Perform cross-validation (5-fold)
cv_scores = cross_val_score(rf_model, X, y, cv=5)

# Display average cross-validation score
print("\nAverage Cross-Validation Score:", np.mean(cv_scores))
```

After training the Random Forest model, predictions were made on the held-out test set (X_test) using rf_model.predict(X_test). To quantify model performance, a confusion matrix was generated using confusion_matrix(y_test, y_pred) to visualize correct and incorrect predictions for the tumor and non-tumor classes.Furthermore detailed report (classification_report(y_test, y_pred)) was produced, which summarizes key metrics such as precision, recall, F1-score, and support for each class. Beyond the simple train-test split, 5-fold cross-validation was performed by calling cross_val_score, and the resulting scores were averaged to obtain a more reliable final cross-validation score, providing deeper insights into the model's generalization capability.

Looking ahead, there are several opportunities for future improvement. Techniques such as hyperparameter tuning (e.g., optimizing the number of trees, maximum depth, or minimum samples per split) could further enhance model performance [7]. Addressing potential class imbalances using methods like SMOTE or adjusting class weights could improve sensitivity to rare but critical cases. Additionally, exploring alternative modeling techniques like XGBoost or LightGBM could offer performance gains, especially if the dataset grows larger or more complex. By continuously refining the evaluation approach and leveraging advanced modeling strategies, we aim to build predictive models that are not only accurate but also transparent, reliable, and directly valuable in clinical decision-making.

Conclusion

This project set out to explore predictive potential of clinical and demographic data in the assessment of cancer survival outcomes using machine learning techniques. Using a publicly available brain tumor dataset and applying datamining methods we were able to identify meaningful attributes and patterns that may improve early detection efforts and support personalized treatment strategies

Thoughout the project, we focused on understanding how early stages indicators (tumor size and genetic predisposition) interact with patient specific variables like age and blood pressure to influence survival rates. Our results consistently highlighted blood pressure, genetic risk and age as the most influential variables in predicting survival outcomes, even when traditional correlation analysis did not initially reveal strong relationships. Using a Random Forest Classifier, we were able to overcome these limitations and interpret feature importance more effectively.

The model’s ability to handle both categorical and continuous data along with K-fold cross validations(K=5) helped to ensure reliability and exposed complex, nonlinear relationships on the data.

The applications of this model span clinical practice, healthcare resource management and public health education. It can assist with early detection leading to earlier screening and treatment. Additionally, public health campaigns could use these findings to promote awareness of modifiable risk factors like blood pressure and the importance of genetic counseling. This gives patient clear insights into their own health through personalized education tools.

Even with promising outcomes, areas of improvement for this project include class imbalance. Particularity the underrepresentation of non-survivors that may affect the model and its outcomes. Addressing this involves applying techniques such as SMOTE or adjusting classes weights to ensure more accurate predictions. We can also aim to explore survival analysis models to incorporate time to event data which may enhance our predictions as well.

In conclusion, this research lays the groundwork for a predictive tool that bridges the gap between accessible patient data and informed clinical dicisions. Continueted development and validation across diverse data sets can expand its impact and improve

early cancer detection, guiding treatment strategies and ultimately leading to better patient outcomes with the incorporation of data mining.

References:

[1] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145-1159.

[2] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432.

[3] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432.

[4] Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. BMC Bioinformatics, 7(1), 91.

[5] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 2(12), 1137–1143.

[6] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437.

[7] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281–305