HW 5 Report

1. Preprocessing

For the preprocessing part, we handle the missing value(there doesn't appear to be any), normalize the dataset, and convert the categorial features to one-hot numerical features. Then we use PCA to decompose the dataset and use elbow plot to determine the number of principal components

.

1. Modeling

For the feature selection, we applied random forest to select the features of which the performance exceeded the median.

2.1. K-Nearest Neighbors (KNN)

Summary: KNN is a simple, instance-based learning algorithm where predictions for new instances are made based on the labels of the K nearest neighbors in the feature space.

Pros:

Simple and intuitive, no assumptions about data.

Effective if the training data is large.

Cons:

Computationally expensive as it requires distance calculation to all training samples.

Sensitive to the scale of the data and irrelevant features.

Hyperparameters:

k: Number of nearest neighbors to consider.

distance metric: Type of distance to use (e.g., Euclidean, Manhattan).

2.2. Naïve Bayes

Summary: A family of probabilistic algorithms based on applying Bayes’ theorem with the “naïve” assumption of conditional independence between every pair of features given the value of the class variable.

Pros:

Simple and easy to implement.

Performs well with large feature spaces and categorical data.

Cons:

Relies on an often-faulty assumption of mutually independent features.

Can be outperformed by other models on complex tasks.

Assumes certain distributions to be applied.

Hyperparameters:

alpha: Smoothing parameter.

priors: Prior probabilities of the classes (if not uniform).

2.3. C4.5 Decision Tree

Summary: C4.5 builds decision trees from a set of training data using the concept of information entropy. It iteratively splits data into subsets, which increases the predictability with each split.

Pros:

Easy to understand and interpret.

Can handle both numerical and categorical data.

Cons:

Prone to overfitting especially with noisy data.

Can become unstable because small variations in data might result in a completely different tree being generated.

Hyperparameters:

max\_depth: Maximum depth of the tree.

min\_samples\_split: Minimum number of samples required to split an internal node.

2.4. Random Forest

Summary: An ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes of the individual trees.

Pros:

Reduces overfitting in decision trees and improves accuracy.

Effective in high dimensional spaces.

Cons:

Can be computationally exhaustive, especially with large numbers of trees.

Less interpretable than decision trees.

Hyperparameters:

n\_estimators: Number of trees in the forest.

max\_features: Maximum number of features considered for splitting a node.

max\_depth: Maximum depth of each tree.

2.5. Gradient Boosting

Summary: Gradient Boosting is an ensemble technique that builds trees one at a time, where each new tree helps to correct errors made by previously trained trees.

Pros:

Often provides predictive accuracy that cannot be beaten.

Flexible, can optimize on different loss functions and provides several hyperparameter tuning options that make the function fit very flexible.

Cons:

Can be prone to overfitting if not tuned properly.

Computationally expensive and requires careful tuning of parameters.

Hyperparameters:

n\_estimators: Number of boosting stages to perform.

learning\_rate: Rate at which the contribution of each tree is shrunk.

max\_depth: Maximum depth of each individual tree.

1. Hyperparameter Tuning

We pick ​​RandomForest and DecisionTree to tune. For ​​RandomForest, we choose **max\_depth** and **min\_samples\_split**. For DecisionTree, we choose **n\_estimators** and **max\_depth.** Finally, we picked the RandomForest with max\_depth=5, n\_estimators=10 to get the best performance.

1. Results

Performance for all the models:



Importances for Features

| Feature | Importance |
| --- | --- |
| Survival Months | 0.717794 |
| Regional Node Positive | 0.141961 |
| Grade\_3 | 0.026318 |
| N Stage\_N3 | 0.020658 |
| N Stage\_N1 | 0.017556 |
| 6th Stage\_IIA | 0.014579 |
| Estrogen Status\_Negative | 0.013698 |
| Progesterone Status\_Negative | 0.013573 |
| Progesterone Status\_Positive | 0.012068 |
| 6th Stage\_IIIC | 0.010987 |
| Estrogen Status\_Positive | 0.010809 |

Then we picked 'Regional Node Positive', 'Survival Months', 'N Stage\_N1', 'N Stage\_N3', '6th Stage\_IIA', 'Grade\_3'.