Significant Feature Selection and Optimized Tumor Prediction

Shareek & Mohammad

2024-11-20

Project Contextualization

In this project we will be analyzing the Wisconsin Diagnostic Breast Cancer (WDBC) data to determine which cellular features strongly correlate with malignancy and benignity, then applying statistical and machine learning techniques to optimize predictive classification of malignant and benign tumors.

This data set created by Dr. William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian was used for various research, including medical literature.

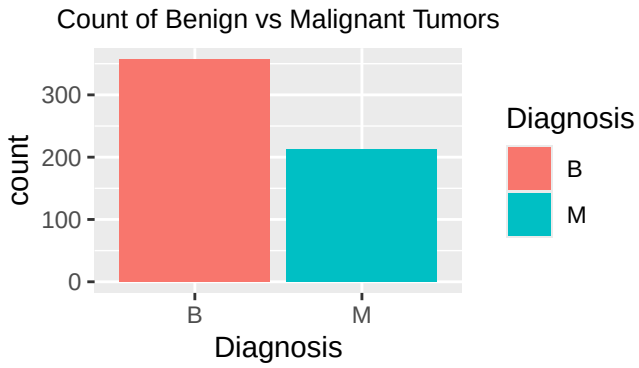
All feature variables are numerical except for `Diagnosis`*. We will convert it from a character variable to a factor with two levels:* `B` *and* `M`*, representing benignity and malignancy respectively. Also, no NA values were found so we will not be omitting any points of data.*

The calculations below are for cleaned and factored data

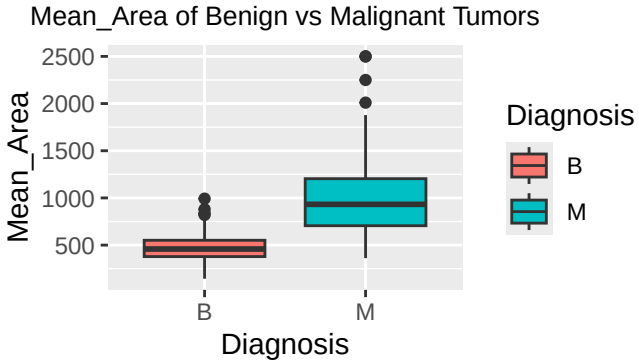
Preliminary Exploratory Data Analysis

The data set contains 569 observations and 32 variables. The first column is the `ID` of the patient, the second column is the diagnosis of the patient (M = malignant, B = benign), and the remaining 30 columns are the features of the cell nuclei. The features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

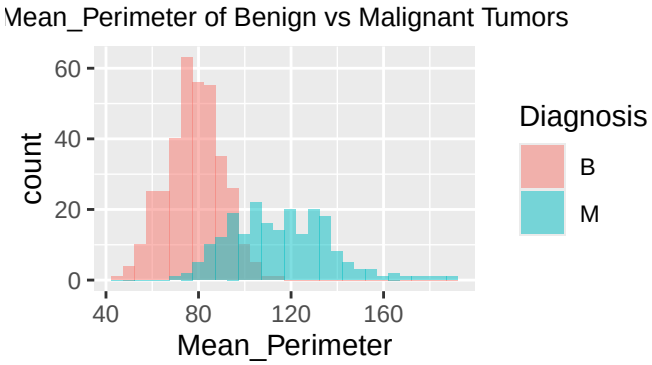
Radius *Distances from center to points on the perimeter*
Texture *Standard deviation of gray scale values*
Perimeter *The total distance between the "snake" points constitutes the nuclear perimeter*
Area *Counting the number of pixels on the interior of a cell and one-half the pixels in the perimeter*
Smoothness *Local variation in radius lengths*
Compactness *A measure of the compactness of a cell using the formula perimeter^2/ area - 1.0*
Concavity *Severity of concave portions of the contour*
Concave points *Number of concave portions of the contour*
Symmetry *A measure of symmetry of a cell*
FractalDimmension *"coastline approximation" – 1, "coastline approximation" is described in Mandelbrot [2]*



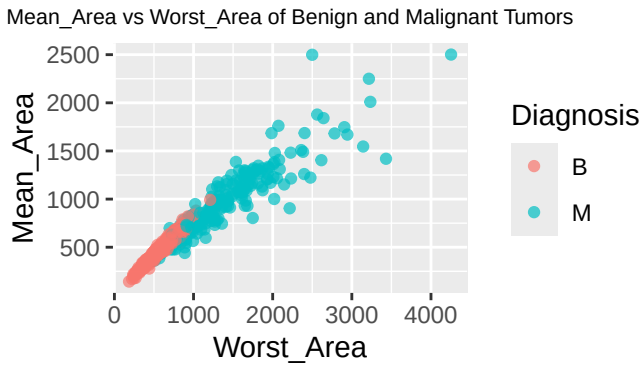
The bar chart shows the distribution of `Diagnosis` in the data set, which we can see is around 60% (357) of the tumors are benign and 40% (212) are malignant.



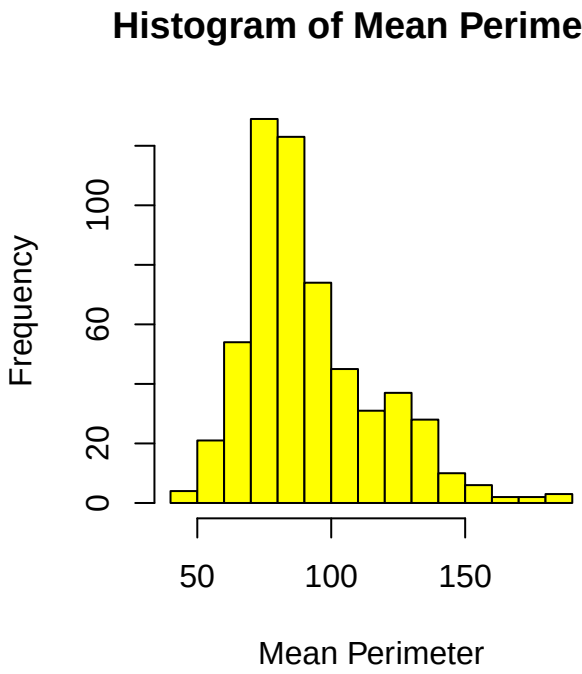
The boxplot shows the Malignant tumors have a higher `Mean_Area` compared to benign tumors. This indicates that `Mean_Area` could possibly be a significant feature in distinguishing between malignant and benign tumors.

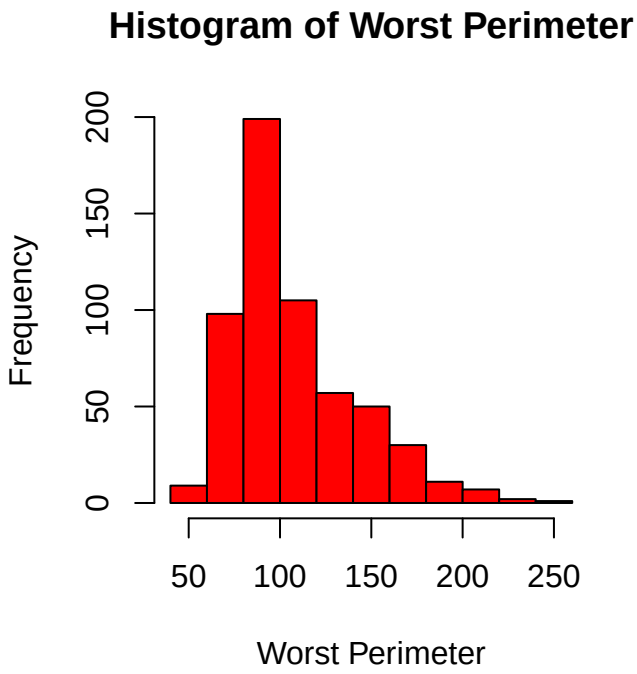


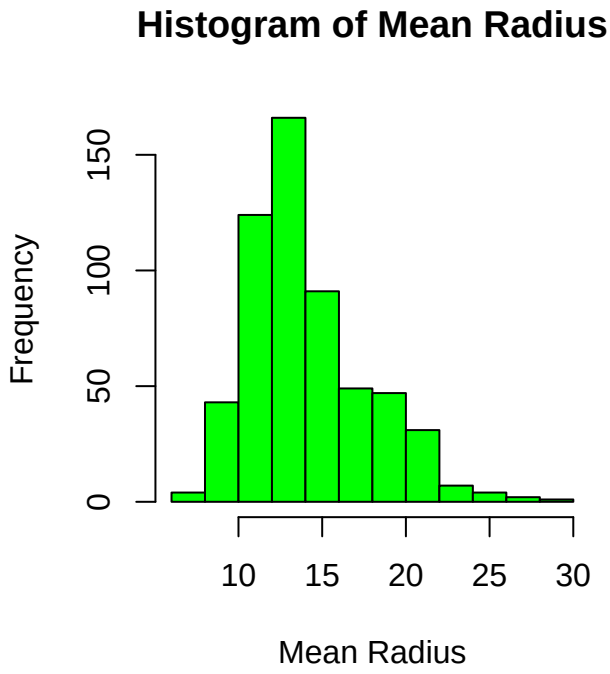
The histogram shows the distribution of `Mean_Perimeter` in the data set. The malignant tumors have a wider range of `Mean_Perimeter` values, whereas the benign data can be seen clustering at a higher level, with values centered around 80, compared to the malignant tumors.

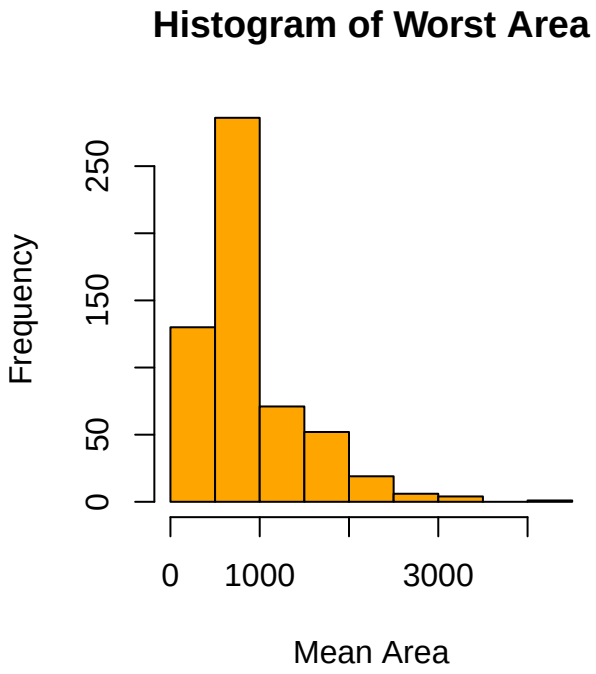


The Scatterplot shows a strong positive correlation between `Mean_Area` and `Worst_Area`, regardless of diagnosis. The malignant tumors are in the upper right corner suggesting that it has larger values for both metrics.









Performing the t-test for the `Mean Area` indicates a highly significant difference in the `Mean Area` between benign and malignant tumors. The mean `Mean_Area` for benign tumors is 462.79, while for malignant tumors it is 978.38, therefore, because of the large difference in means this suggests that `Mean_Area` is a key feature we can use.

When performing the t-test for `Worst_Area`, it shows a highly significant difference in `Worst_Area` between benign and malignant tumors. This is another significant feature we can use to distinguish the difference.

To further identify significant features, we will now perform a correlation analysis to identify highly correlated features.

To support the correlation, we have a heatmap where we can visually identify the features' relationships.

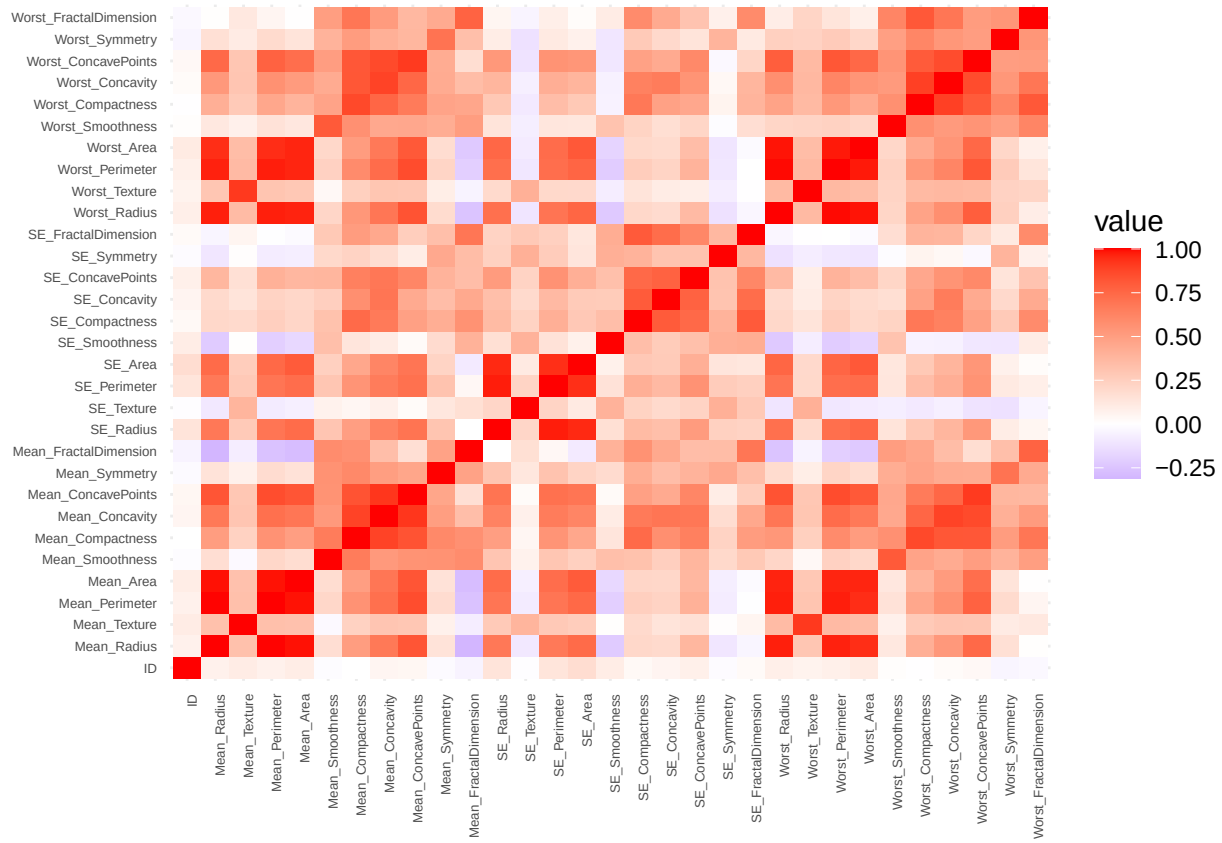


Table 1: Highly Correlated Features

Feature1	Feature2	Correlation
Mean_Perimeter	Mean_Radius	0.9978553
Mean_Area	Mean_Radius	0.9873572
Mean_ConcavePoints	Mean_Radius	0.8225285
Worst_Radius	Mean_Radius	0.9695390
Worst_Perimeter	Mean_Radius	0.9651365
Worst_Area	Mean_Radius	0.9410825

Logistic Regression Model

As we are trying to find the probability of a tumor being malignant or benign based on different predictors, we must first identify the predictors with the strongest correlation. Therefore, we perform backward selection to remove insignificant predictors. Then, after running the backward selection which removed less significant variables, we must confirm the absence of multicollinearity (the correlation of several independent variables in a model).

Multicollinearity Check

Initial Model: formula = Mean_Radius ~ Mean_Perimeter + Mean_Area + Mean_Smoothness + Mean_Compactness + Mean_Concavity + Mean_FractalDimension + SE_Radius + SE_Texture + SE_Perimeter + SE_Concavity + SE_ConcavePoints + SE_FractalDimension + Worst_Radius + Worst_Texture + Worst_Perimeter + Worst_Area + Worst_Smoothness + Worst_Compactness

We will now perform a VIF test to check for multicollinearity.

`Worst_Radius` has a VIF of 457.55 which is very high, so we will remove it from the model.

`Mean_Perimeter` has a VIF of 294.16 which is very high, so we remove it from the model.

`Worst_Area` has a VIF of 50.03 which is very high, so we remove it from the model.

`Worst_Perimeter` has a VIF of 26.36 which is very high, so we remove it from the model.

`SE_Perimeter` has a VIF of 25.53 which is very high, so we remove it from the model.

`Mean_Compactness` has a VIF of 19.78 which is high, so we remove it from the model.

`Mean_Concavity` has a VIF of 13.2 which is high, so we remove it from the model.

Finally we are left with `Mean_Area`, `Mean_Smoothness`, `Mean_FractalDimension`, `SE_Radius`, `SE_Texture`, `SE_Concavity`, `SE_ConcavePoints`, `SE_FractalDimension`, `Worst_Texture`, `Worst_Smoothness`, and `Worst_Compactness` which have VIF values below 5. This indicates that there is no multicollinearity between the features and we can use them in the model.

We will now test the accuracy of the logistic regression model. We will use a confusion matrix to see how well the model predicts compared to the actual diagnosis.

Accuracy = 0.97
Precision for Malignant = 0.96
Recall for Malignant = 0.98

Looking at the results above, we can confirm that the model correctly classified 97.5% of the cases. It's a balanced performance with a high precision and recall rates with a very few misclassifications.

Model Implementations

Before we begin to run any predictive Machine Learning techniques on the model, we need to split the data into training and testing sets.

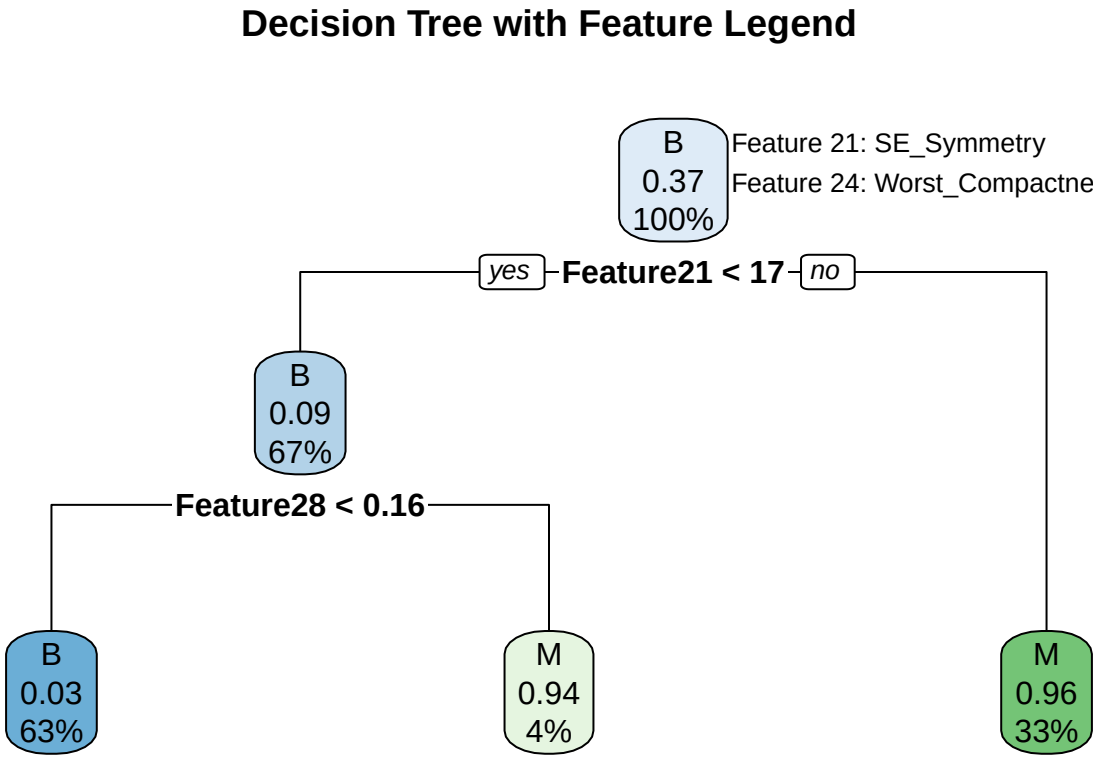
Data Splitting

We split the data into training and testing sets using a 70/30 split, and now will use the training set to train the models, and the testing set to evaluate the models.

We will be modeling the data using four machine learning techniques:
1) Decision Tree Model
2) Random Forest Model
3) K-Nearest Neighbors (KNN) Model with Optimized `k` using Cross-Validation
4) Support Vector Machine with Best Kernel Identification and Hyperparameter Fine Tuning

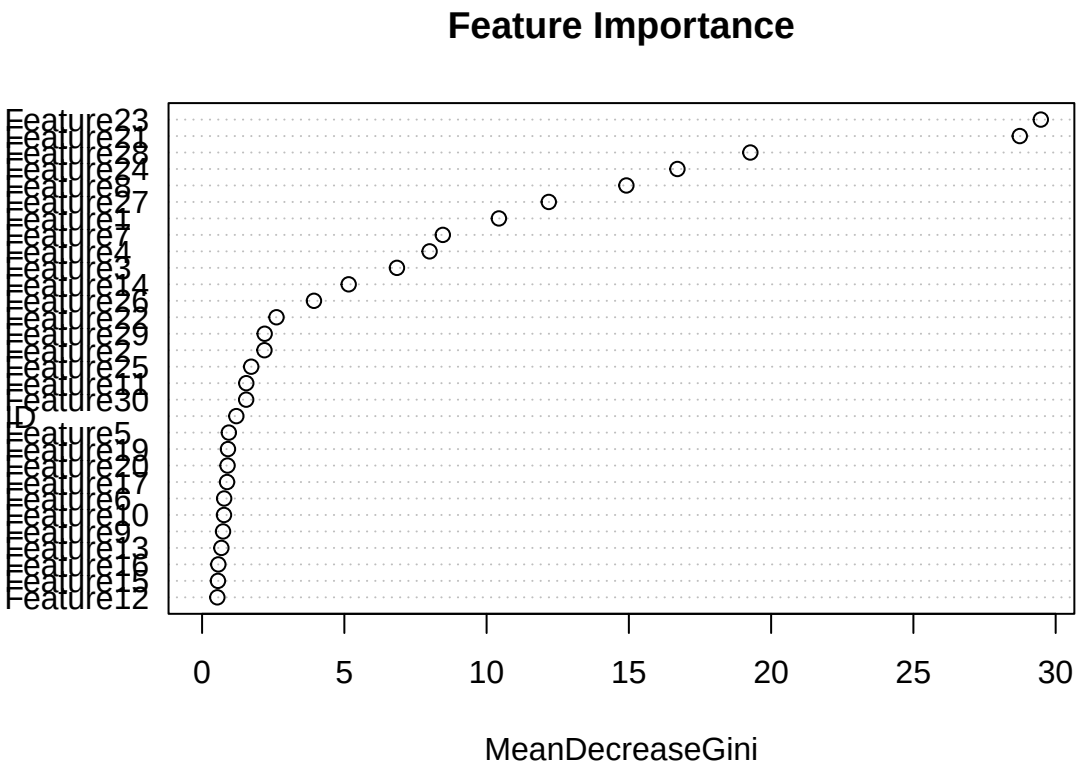
Based off of our initial exploratory data analysis and feature relationship introduction, we decided to select these models because of the complexity of the relationship between the features in the data.

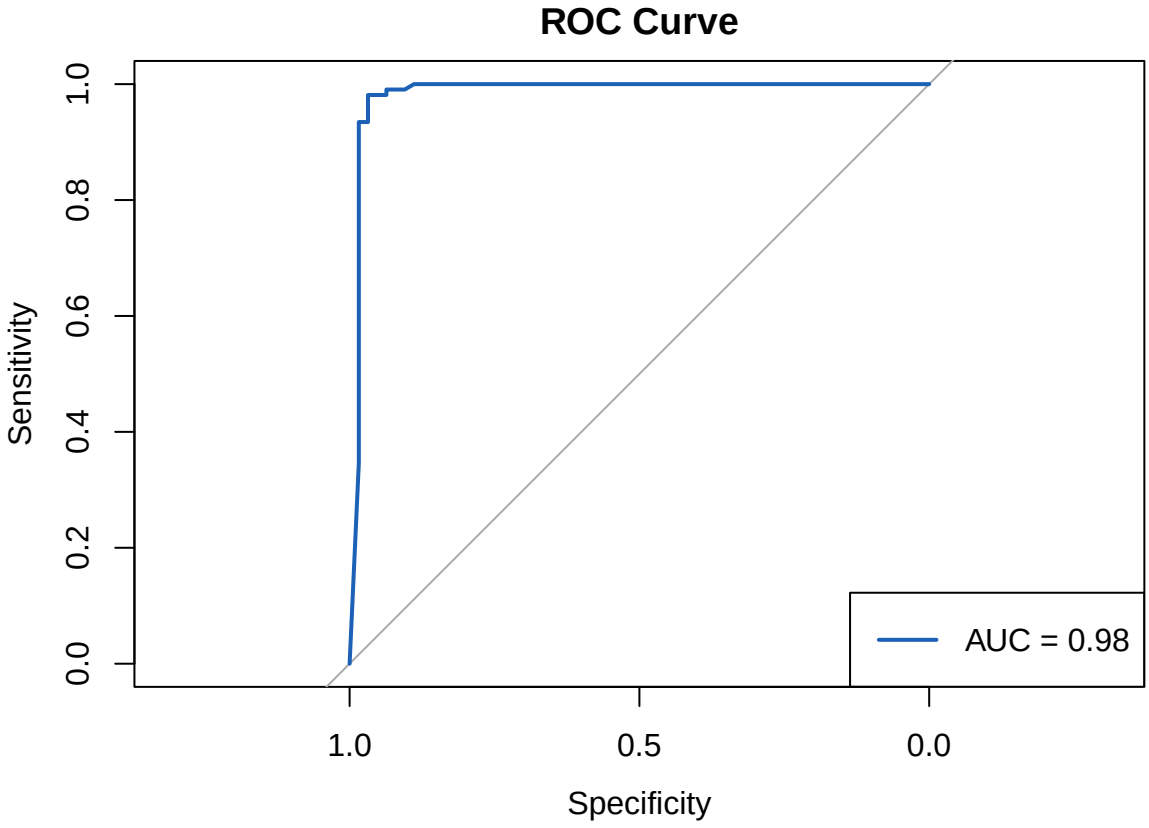
Decision Tree Model

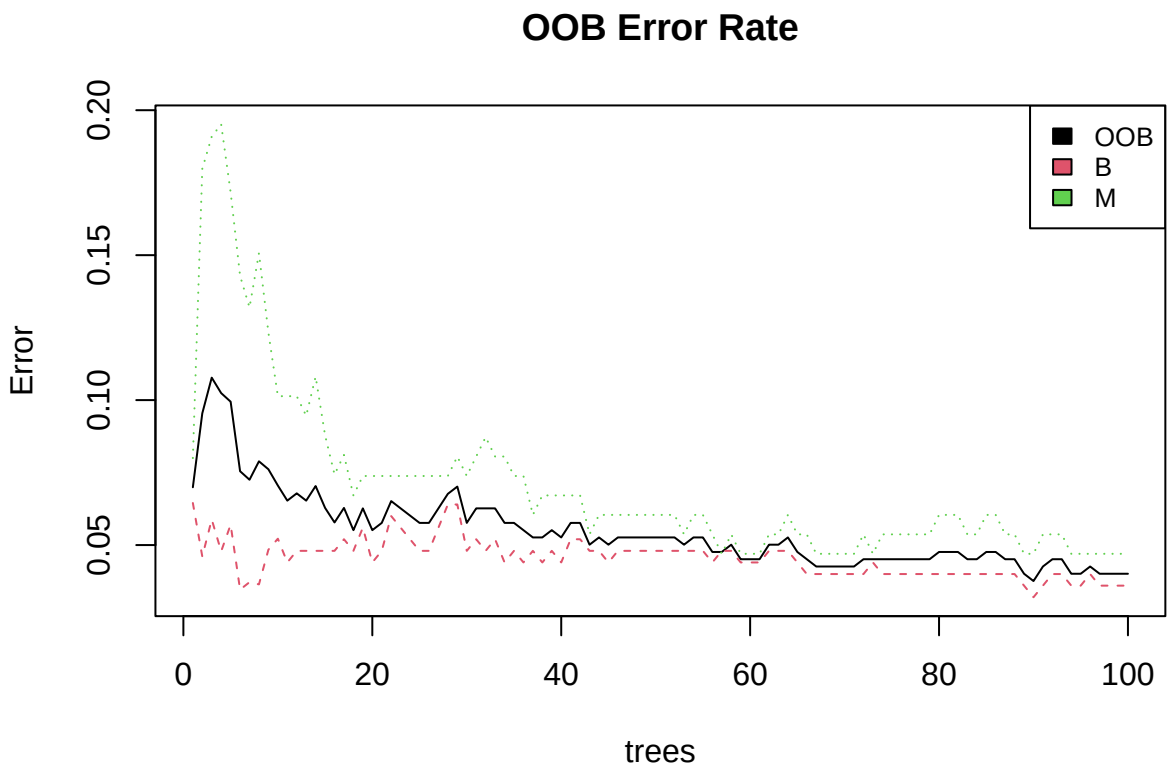


The decision tree is simple, easy to read, and achieves a strong classification performance with high accuracy, sensitivity, and specificity. Misclassifications are balanced between False Positives and False Negatives, with 7 instances each, which indicates good performance in handling both classes. Also, the strong Kappa score of 0.8235 reflects that the model performs significantly better than random guessing.

Random Forest Model



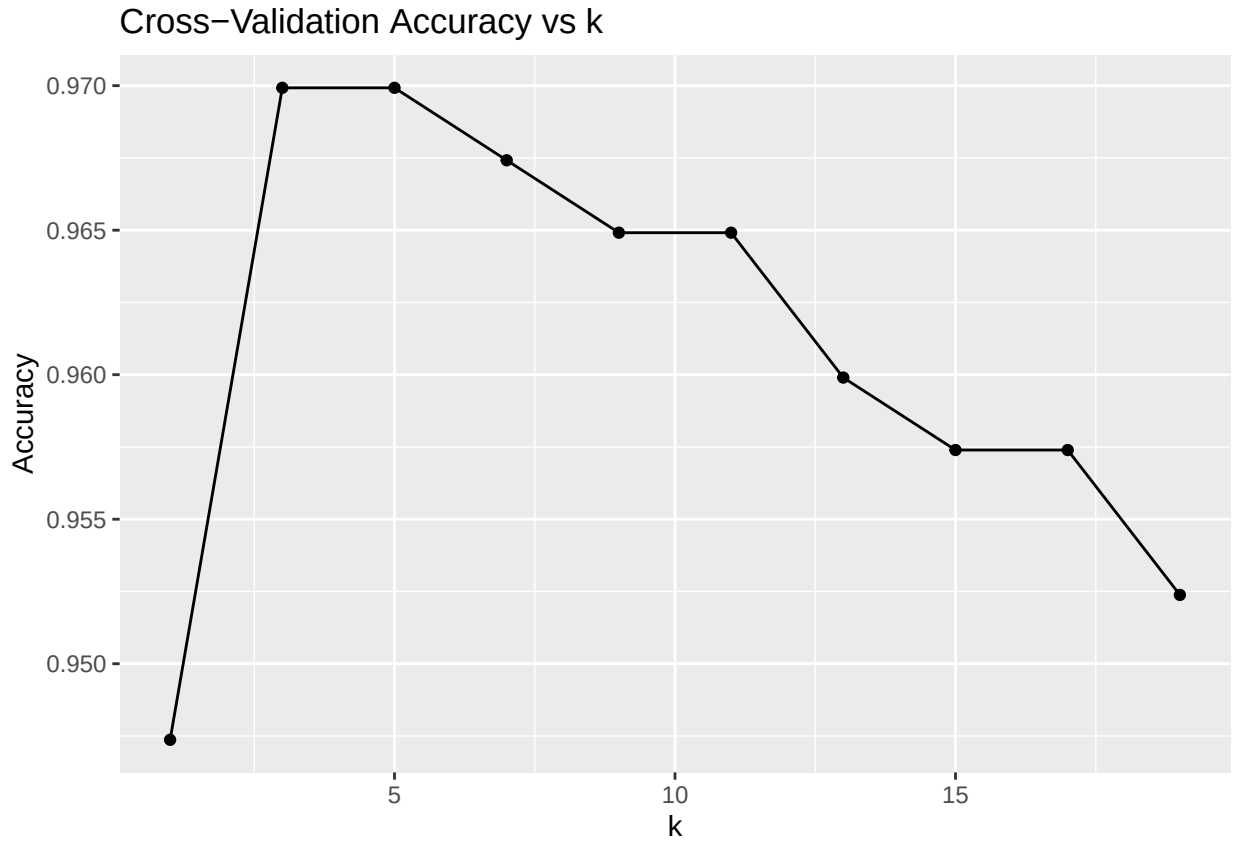




The Random Forest model shows strong classification performance, with a high AUC (0.98) and low OOB error rates. Increasing the number of trees beyond 40 does not significantly improve the model, suggesting that 40-50 trees may be sufficient for this classification problem. Another point to take note of is that the model performs slightly better at classifying the benign class compared to the malignant class, which is common in imbalanced datasets because of the difference in amount of data classified for each label.

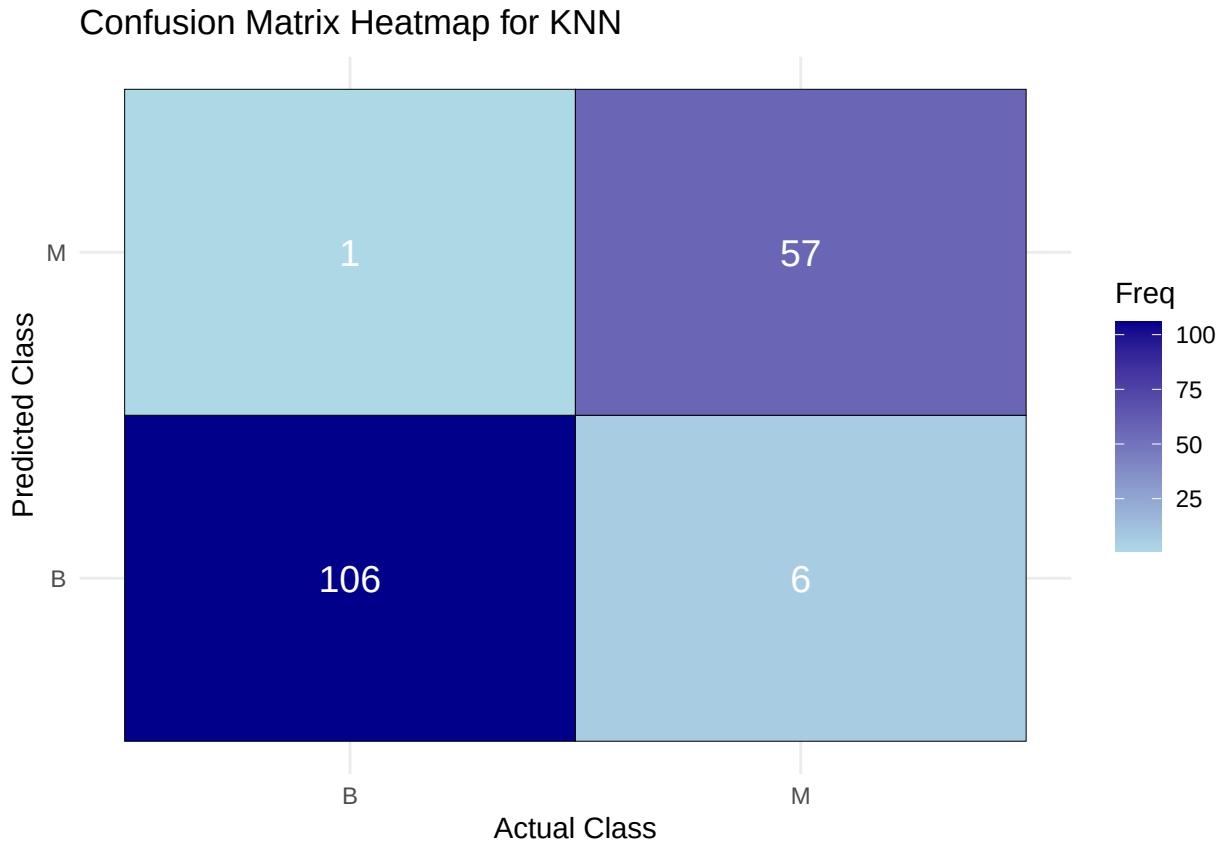
As for feature importance, see here that the Random Forest model relies heavily on a few key features (`SE_Symmetry`, `Worst_Radius`, `Worst_Texture`) for classification. This suggests that these are the most significant variables. The model also exhibits diminishing returns in predictive power for less important features, which indicates that feature selection or dimensionality reduction can further optimize the model.

Optimize `k` for KNN using Cross-Validation



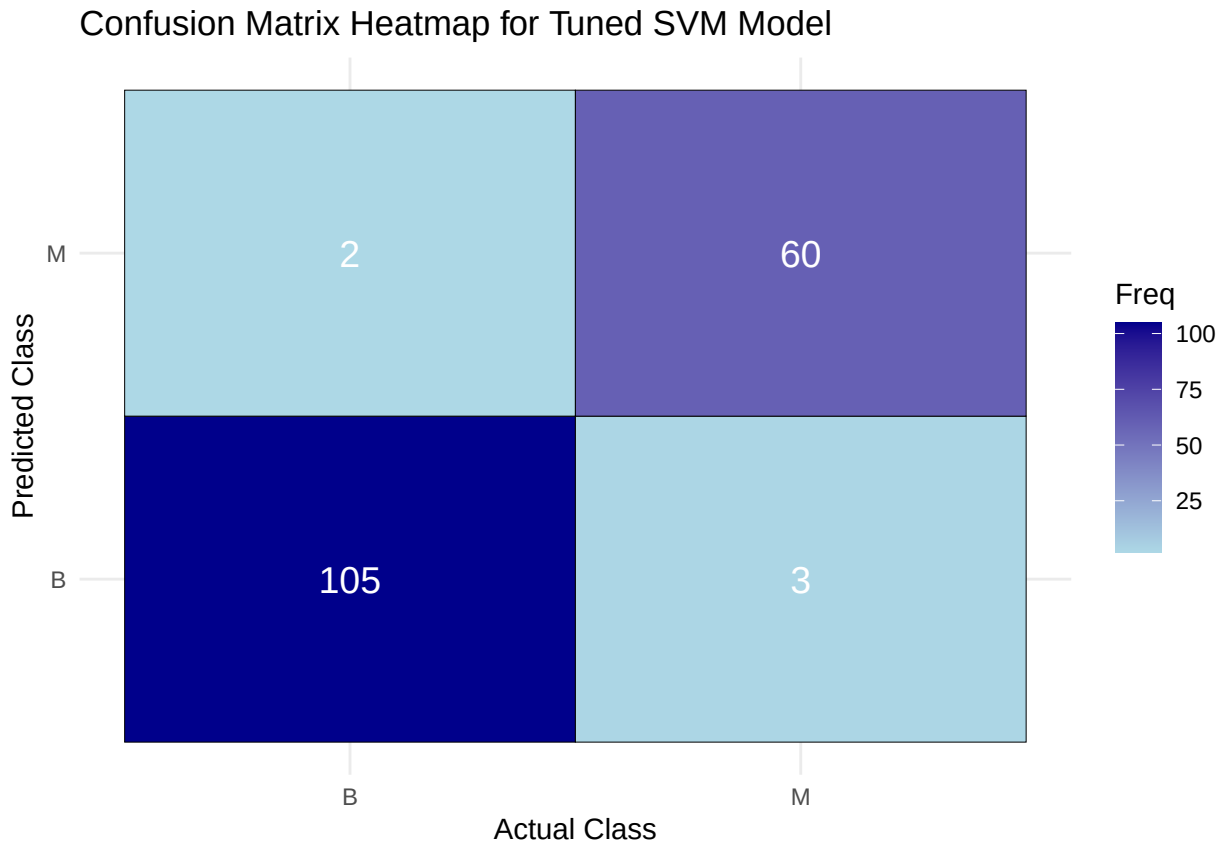
Based on the visualization of the accuracy of `k` values 0 - 20, we can conclude that the optimal value for `k` is 5. Now, we implement the KNN Model using this optimized `k` value.

K-Nearest Neighbors (KNN) Model



Looking at the above heatmap presenting the results of the KNN classification, we see the model performs well overall, with a high number of correct classifications for both benign and malignant cases (only 1 benign case was misclassified as malignant). In terms of model weakness however, we see it struggles slightly with sensitivity, as 6 malignant cases were misclassified as benign. In summary, The model has strong precision for both classes, but a slight imbalance in sensitivity for malignant cases.

Support Vector Machine with Best Kernel Identification and Hyperparameter Fine Tuning (only on best kernel for optimized time complexity)



Best Kernel: linear
Best Parameters for the linear Kernel: cost=0.5 and gamma=0.25
Tuned Model Accuracy: 0.9705882

The SVM model correctly identifies a majority of both malignant and benign cases, as seen by the high true positives (60) and true negatives (105). The number of false positives (2) is very low, indicating the model rarely misclassifies benign cases as malignant. In terms of model weakness, we see it misclassifies 3 malignant cases as benign (false negatives), but overall, the model is highly accurate with particularly strong true positive and true negative rates.

Model Comparison and Conclusion

Decision Tree Accuracy: 0.9176471
Random Forest Accuracy: 0.9705882
KNN Accuracy: 0.9588235
Tuned SVM Linear Kernel Accuracy: 0.9705882

In conclusion, we applied and evaluated several machine learning models, including **Decision Tree**, **Random Forest**, **K-Nearest Neighbors (KNN)**, and **Support Vector Machine (SVM)**, to classify breast tumors as malignant or benign using the Wisconsin Diagnostic Breast Cancer dataset, using only the features that have are statistically significant predictors. Each model was assessed for its performance based on accuracy, precision, recall, and overall classification quality.

The **Decision Tree** model achieved an accuracy of **91.76%**, making it the simplest and most interpretable model among the four options. While it performed generally well, it had balanced but slightly higher misclassification rates (false positives and false negatives), which reduced its overall reliability relative to the more advanced techniques implemented.

The **Random Forest** model demonstrated very strong performance with an accuracy of **97.06%**, making it the best-performing model along with KNN. By using ensemble learning, it handled both malignant and benign cases accurately, achieving high sensitivity and specificity. Furthermore, the variable importance plot identified key features, such as SE_Symmetry, Worst_Radius, Worst_Texture, as key contributors to classification. However, the Random Forest model, while accurate, is less interpretable and viewer-friendly due to its complexity and reliance on multiple trees.

The **K-Nearest Neighbors (KNN)** model, optimized with k=5 through cross-validation, achieved an accuracy of **95.88%**. The KNN model performed well overall but struggled slightly with sensitivity, misclassifying six malignant cases as benign. Its performance shows us very clearly the importance of scaling and parameter tuning for distance-based models, like KNN. The K-Nearest Neighbors Model generally provides simplicity in implementation but lacks interpretability and robustness compared to other models.

Finally, the **Support Vector Machine (SVM)** model with a linear kernel was hyperparameter-tuned and achieved an accuracy of **97.06%**, tying with Random Forest as the top performer. The SVM model displayed strong precision and sensitivity, with minimal false positives (only two) and false negatives (only three). Its reliance on support vectors allows it to handle decision boundaries very efficiently. However, SVM can be time consuming and computationally expensive, especially when kernel functions require a lot of parameter tuning.

In conclusion, both **Random Forest** and **SVM** emerged as the best-performing models, achieving the highest accuracy and balanced classification performance. While Random Forest is ideal for feature importance analysis and ensemble robustness, SVM is a strong contender for clean, linear separable datasets. Decision Tree and KNN are valuable for interpretability and simplicity, respectively, but they fall short in terms of overall accuracy and sensitivity. Future analysis could be fruitful if we explore deeper feature engineering, balancing techniques, and hybrid models to enhance classification performance even further.