- **What did you do to prepare the data? (1 point)**

  In preparing the data for classification, several preprocessing steps were done. This included checking and dropping missing values, null values, non-numerical and string values. In addition, encoding categorical string variables, drop breast variable as some degree of duplication to the breast-quarter variable, and scaling numerical features. Additionally, analyze each variable with data plots visualization to visualize each variable and the relationships between one and others.

- **Which techniques did you use to train the model? (1 point)**

  For model training, various classification supervised learning techniques were used to train the data which included logistic regression, decision trees, random forests, and k-nearest neighbor classifiers. Hyperparameter tuning was performed using techniques like grid search or random search to optimize each model's performance.

- **How does each model perform to predict the dependent variable? (1point)**

  After training the data, the K Nearest Neighbor (KNN) classifier shows an accuracy of 0.87, on the training data. This means that KNN accurately predicts the class label for 87% of instances in the training set. Comparing this accuracy with classifiers we find that both Logistic Regression and KNN have the accuracy score on the training data indicating similar performance in terms of overall correctness in predictions. However, the Random Forest Classifier and Decision Tree Classifier surpass both KNN and Logistic Regression achieving accuracy scores of 0.98 on the training data. This suggests that these ensemble methods, Random Forest excel at capturing underlying patterns in the training data leading to precise predictions.

  When it comes to precision assessments the KNN classifier demonstrates varying precision scores for classes. It achieves a precision of 0.88 for the class indicating a level of correctness in identifying true negatives. However, its precision drops significantly to 0.33 for the class suggesting accuracy in identifying true positives. In contrast the Logistic Regression classifier attains a precision of 0.86 for the class and a perfect precision score of 1.00 for the class, on the training dataset.
  Logistic Regression seems to excel in recognizing instances when compared to KNN. On the hand Random Forest and Decision Tree classifiers show precision scores, for both negative and positive case as opposed to KNN and Logistic Regression.
  Move on to the recall. The KNN model shows a recall of 0.95, for the negative instances indicating its ability to catch most true negatives. However, its recall drops to 0.17 for the instances showing that it misses a lot of positives. On the hand the Logistic Regression model nails it with a 1.00 recall for negative instances but

completely misses out on capturing any true positives resulting in a 0.00 recall for positive instances. This highlights a weakness of Logistic Regression, in detecting instances. In contrast both Random Forest and Decision Tree models have recall scores compared to KNN and Logistic Regression for both positive classes.
Lastly, the F1 score now which balances precision and recall. The KNN model achieves an F1 score of 0.91 for negatives and 0.22 for positives. In comparison the Logistic Regression model gets F1 scores of 0.93 for negatives and 0.00 for positives. Both Random Forest and Decision Tree models have F1 scores compared to KNN and Logistic Regression across both positive classes.
In short, when comparing KNN and Logistic Regression classifiers they show performance levels in terms of precision, recall and F1 score, for the class. However, their performance differs significantly for the class variable. On the hand ensemble methods like Random Forest and Decision Tree classifiers surpass KNN and Logistic Regression in accuracy on the training data. Nonetheless each classifier has its advantages and limitations, across evaluation metrics underscoring the need to choose the most suitable model based on specific task requirements.

- **Which model would you recommend to be used for this dataset (1 point)**

  After reviewing the results, it seems that the Random Forest model is the choice for this dataset as it shows performance across various measures. By taking average of all trees, it performs slightly better results performance compare to the decision tree model when it comes to both models have same high accuracy compare to the other methods. Yet we can still enhance its performance of each method by using cross validation and improve recall for KNN model if needed.

- **How does the model perform with respect to false positives and false negatives? Which standard model performance metric is most important to optimize? Explain why. (1 point)**

  Regarding false positives and false negatives, optimizing recall (minimizing false negatives) is crucial for this problem. In applications such as medical diagnosis or fraud detection, missing a positive instance (false negative) can have severe consequences. Therefore, minimizing false negatives is paramount, making recall the most important metric to optimize. However, it's essential to strike a balance between precision and recall, as increasing one may lead to a decrease in the other. When it comes to dealing with results and missed detections ensuring accuracy (reducing missed detections) is vital, in this scenario. In fields like healthcare diagnosis or identifying fraud, overlooking a case (missed detection) can have implications. Hence, reducing missed detections is crucial to making accuracy the key metric to focus on. However, it's important to find a balance between accuracy and thoroughness since improving one might result in a decline in the other. Thus, achieving an F1 score, which evaluates both accuracy and thoroughness, is ideal for this situation.