In preparing the data for analysis and modeling to predict breast cancer recurrence, I followed several essential steps to ensure the datasets readiness. Initially, I conducted an overview of the dataset, consisting of 286 entries with 10 attributes, to understand its structure and size comprehensively. This examination revealed 286 rows and 10 columns, comprising a total of 2860 data points. None of the columns had less than 286 non-null values meaning that there was no need to address the issue of having missing data entries.

Upon inspecting the dataset using the info() method, I observed that most columns were in the object datatype, which wasnt optimal for machine learning algorithms. Particularly, columns representing categories, such as age and tumor size, were initially in object format with ranges as values. To address this, I converted categorical columns (class, age, menopause, tumor-size, inv-nodes, node-caps, breast, breast-quad, and irradiat) to the category datatype. This conversion streamlined the dataset, ensuring proper treatment of features by algorithms and reducing memory usage. Further analysis revealed the presence of ? placeholders for missing or unknown values in the node-caps and breast-quad columns. To handle these invalid entries, I replaced them with the mode of their respective columns, a method chosen for its simplicity and effectiveness in maintaining dataset integrity without introducing significant bias.

To gain deeper insights into the dataset, I created various visualizations, including countplots and histograms, to explore class distribution, age ranges, menopause status, tumor sizes, involved nodes, node caps, degrees of malignancy, breast cancer locations, and irradiation statuses. These visualizations highlighted imbalances in class distribution, predominant age ranges and tumor sizes, and distributions of other categorical variables, providing a comprehensive understanding of the datasets characteristics. To prepare the dataset for machine learning algorithms, I applied one-hot encoding to categorical variables. This process

transformed categorical variables into a series of binary columns, one for each category, making the dataset suitable for the algorithms requirements. This step was crucial for converting nominal and ordinal data into a format efficiently processed by the models.

Moving on to model training and hyperparameter tuning, Ive opted for K-Nearest Neighbors (KNN), Random Forest, and Decision Trees as classification algorithms. Firstly, K-Nearest Neighbors (KNN) is an instance-based learning algorithm where a samples class is determined by the majority class among its k-nearest neighbors. Next, the Random Forest Classifier, a robust ensemble learning technique, constructs multiple decision trees during training and outputs the class mode of the individual trees. This method corrects for decision trees tendency to overfit their training set. Additionally, Decision Trees serve as a straightforward decision support tool, employing a tree-like model of decisions and their consequences. They are particularly useful for preliminary analysis due to their simplicity. I trained a decision tree classifier and visualized it to gain insights into the decision-making process.

For the KNN model, there was an accuracy of 0.779 on the test data, 0.74 on the train data, with a recall of 0.346, precision of 0.818, and an F1 score of 0.486. Since the training data had an accuracy that was slightly higher at 74%, it indicates reasonable consistency in predictions. However, it's notable that the model performs slightly better on the data it was trained on than on unseen data. In the context of breast cancer recurrence prediction, several key considerations emerge from these results. Firstly, while accuracy is relatively high, indicating the model's overall capability to identify both recurrence and non-recurrence events correctly, it's crucial to recognize that accuracy alone might not be the most critical metric, particularly in medical diagnostics, where the consequences of false negatives and false positives are uneven. Secondly, the recall metric stands out as notably lower, which is concerning for a medical

diagnostic tool, as with a recall of 34.6%, the model identifies just over a third of all true recurrence events in the test data, falling short of the clinical standard where maximizing the identification of all positive cases is imperative. Although precision is high at 81.8%, indicating that the model is correct about 82% of the time when predicting a recurrence event, the trade-off between precision and recall becomes crucial.

For random forest model, there is an assessment of both the test and training data. On the test data, accuracy stands at 67%, with precision for class 0 (No recurrence) at 0.74 and for class 1 (Recurrence) at 0.45, coupled with recalls of 0.82 for class 0 and 0.35 for class 1. Conversely, the training data showcases a significantly higher accuracy of 90%, with precision values for class 0 and class 1 at 0.92 and 0.84, respectively, and corresponding recalls at 0.94 and 0.81. Theres a decline in all performance metrics when applied to the test data so that means that there was overfitting. Something concerning is the model's low recall of 0.35 for class 1 on the test data, a crucial metric in a medical diagnostic context where missed positive cases can carry severe consequences. This model shows promise despite this because the other recall values are much higher.

For the Decision trees model, the model has an accuracy of 64%, with precision for class 0 at 0.72 and for class 1 at 0.37, alongside recalls of 0.80 for class 0 and a concerning 0.27 for class 1. Conversely, the training data showcases exceptional performance, with a high accuracy of 97% and precision scores of 0.96 for class 0 and a perfect 1.00 for class 1, coupled with recalls of 1.00 and 0.90, respectively. Despite the outstanding training metrics, the model's substantial performance drop on the test data, especially evident in the low recall rate of 0.27 for class 1, indicates overfitting, suggesting the model has learned the training data too intricately, thereby compromising its ability to generalize to unseen instances. the decision tree model shows

the lowest recall and precision for the positive class among the three models. This indicates its lesser effectiveness in correctly identifying recurrence cases without generating excessive false positives.

When the model's purpose is to predict the recurrence of breast cancer, the consequence of false negatives failing to predict a recurrence when it actually will occur is significantly higher than the cost of false positives. This is because missing a cancer recurrence could delay treatment for a patient, potentially leading to worse health outcomes. Therefore, the recall metric becomes especially important in this context, as it measures the model's ability to correctly identify all positive cases. When comparing all the model performances, the Random Forest Classifier emerges as the most balanced model. Despite not boasting the highest individual recall, precision, or accuracy scores, it provides a reasonable trade-off between recall and precision. This balance ensures a lower likelihood of missing true recurrence cases while keeping false positives manageable. While the KNN model exhibits high precision, its lower recall for the positive class raises concerns, especially in medical diagnosis contexts where missing true positive cases can have severe consequences. The Decision Tree model was very overfitted, making the Random forest classifier the best option of the three.

ChatGPT use:

what does this warning mean:

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

This warning typically occurs in the context of classification tasks when calculating precision and F-score metrics. It means that for some labels in your classification problem, there were no predicted samples. As a result, the precision and F-score for these labels are undefined mathematically and are being set to 0.0.

To address this warning, you have a couple of options:

1. **Ignore the warning**: If you are okay with the precision and F-score being set to 0.0 for labels with no predicted samples, you can ignore the warning.

2. **Set the `zero_division` parameter**: You can control this behavior explicitly by setting the `zero_division` parameter to either `"warn"`, `"warn"` or a numeric value in the function call where you're calculating these metrics. This parameter determines what happens when the precision and/or recall denominators are 0. You can choose to raise a warning, ignore, or set a specific value.

Here's how you can set `zero_division` parameter to `"warn"`:

```python
from sklearn.metrics import precision_score, recall_score, f1_score

precision = precision_score(y_true, y_pred, zero_division='warn')
recall = recall_score(y_true, y_pred, zero_division='warn')
f1 = f1_score(y_true, y_pred, zero_division='warn')
```

Or, if you prefer to set a specific value when there are no predicted samples:

```python
precision = precision_score(y_true, y_pred, zero_division=1.0)  # Set to 1.0 when no predicted samples
recall = recall_score(y_true, y_pred, zero_division=1.0)
f1 = f1_score(y_true, y_pred, zero_division=1.0)
```

Choose the approach that best suits your needs and the nature of your data.