Project 1: Report

Name: Hannah Marques

Date: 03/04/2025

1. Data Preparation

To prepare the data for the analysis and modeling parts, these steps were followed:

1. **Data Cleaning:** Data was identified, and duplicate rows were removed to ensure unique observations.
2. **Handling Missing Values:** Categorical columns were replaced with the mode, and numerical columns were filled with the median to avoid skewing the data.
3. **One-Hot Encoding:** Categorical variables were converted into numerical format using one-hot encoding, ensuring that the models could interpret them properly.
4. **Class Balancing:** The dataset was imbalanced, with more non-recurrence cases than recurrence cases. To address this issue, **Synthetic Minority Over-sampling Technique (SMOTE)** was applied to generate synthetic data for the minority class after double checking results on ChatGPT.
5. **Feature Scaling:** The numerical features were standardized using **StandardScaler** to ensure fair weight distribution across all variables.

2. Insights from Data Preparation

Through exploratory data analysis (EDA), I gained the following insights:

- **Class Imbalance:** The dataset had significantly more non-recurrence cases than recurrence cases, making it necessary to apply SMOTE to avoid biased model predictions.
- **Feature Importance:** The degree of malignancy showed a strong distribution across three levels, suggesting it plays a crucial role in recurrence prediction.
- **Data Distribution:** Visualizations indicated that some features had distinct clusters, indicating that a non-linear model (such as Decision Trees, Random Forest) might be more effective than a purely linear approach for this analysis.

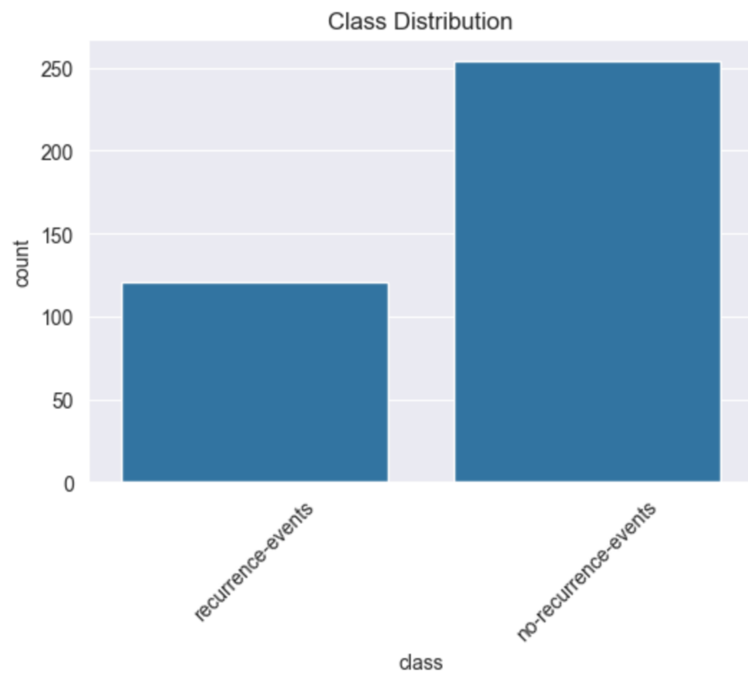

Figure 1: Class Distribution Histogram

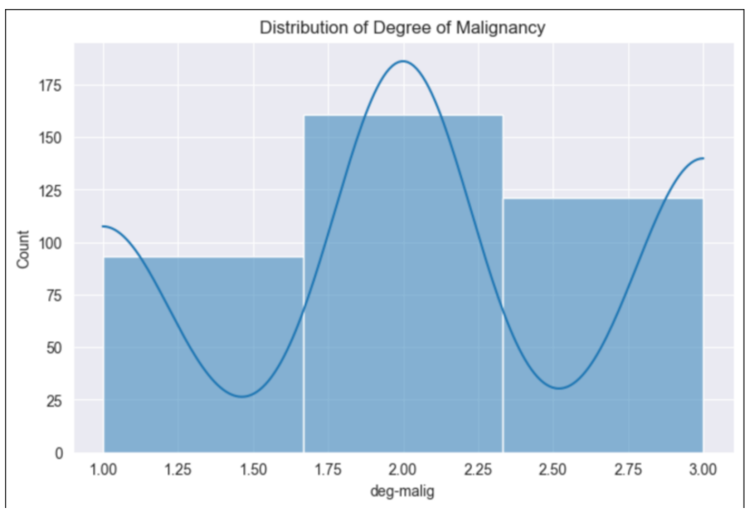

Figure 2: Degree of Malignancy Graph

3. Model Training Procedure

Three classification models were trained to predict cancer recurrence:

1. **K-Nearest Neighbors (KNN):** A distance-based model that classifies new instances based on the majority class among the k-nearest neighbors.
2. **KNN with GridSearchCV:** A fine-tuned version of KNN where it optimized hyperparameters (e.g., number of neighbors) using grid search.

3. **Logistic Regression:** A linear classification model that predicts the probability of recurrence based on feature relationships.

For training:

- I decided to split the dataset into **80% training and 20% testing**, ensuring class proportions remained similar (stratified split).
- I applied **SMOTE** only to the training set to prevent data leakage.
- Lastly, **scaled features** were used to standardize numerical values before training.

4. Model Performance and Explanation

Each model was evaluated using **accuracy, precision, recall, and F1-score**. Key observations:

- **KNN (k=3):**
  - KNN relies on feature similarity, so it struggles when data points from different classes overlap.
  - It performed reasonably well but struggled with false positives and false negatives.
  - KNN is commonly used when data has clear clusters and is well-distributed in feature space, which was not the case here.
- **KNN with GridSearchCV:**
  - GridSearchCV was applied to optimize hyperparameters, but results were similar to basic KNN.
  - This suggests that the model's performance was more dependent on data distribution than the number of neighbors.
  - KNN with tuning is often used for pattern recognition and recommendation systems, where class separation is clearer.
- **Logistic Regression:**
  - Since logistic regression assumes linear relationships, it struggled with non-linearly separable data.
  - However, it showed slightly better recall for recurrence cases, making it more useful in medical applications where missing a positive case is critical.
  - Logistic regression is widely used in **medical diagnoses, credit scoring, and fraud detection** due to its interpretability and ability to model probabilities.

5. Model Confidence and Future Improvements

While the models show some predictive capability, their performance is **not highly reliable for real-world medical use** due to:

- **Low recall for recurrence cases (Class 1),** meaning the model still misclassifies some patients who experience recurrence.
- **Potential overfitting in KNN**, as performance was similar before and after hyperparameter tuning.
- **Class imbalance challenges,** despite SMOTE improving recall slightly.

To improve confidence in the model, future work should explore:

- **Testing Decision Trees and Random Forest models**, which can handle non-linear feature interactions better.
- **Applying feature selection techniques** to remove less relevant variables and reduce noise.
- **Using ensemble methods** (such as boosting) to improve predictive performance.

Overall, while this model provides a baseline understanding of recurrence prediction, additional refinement is required before it can be confidently applied in a medical setting.

6. References and Acknowledgment

- ChatGPT was used to assist with a few parts of this project, including:
- Polishing the code to make it more efficient and well structured.
- Adding SMOTE for class balancing Suggested to improve the model's ability to detect recurrence cases.
- Providing tips for better analysis to check class imbalance, evaluating recall, and suggesting alternative models.

Polishing comments, writing, and formatting in order to keep the code and documents in a organized manner.

The core work—coding, testing, and analysis—was done independently, but ChatGPT gave useful suggestions to refine the final version.