# Feature Selection and Data Classification Report

(a)Data Preprocessing:

1. Data Loading and Exploration

This process began by loading the dataset "Breast_Cancer_dataset.csv" into a pandas DataFrame. Then the `isnull().sum()` function is applied to ensure there are no missing values.

```
: # Check for missing values
  print("Missing values:", data.isnull().sum())
```

```
Missing values: Age                       0
Race                      0
Marital Status            0
T Stage                   0
N Stage                   0
6th Stage                 0
differentiate             0
Grade                     0
A Stage                   0
Tumor Size                0
Estrogen Status           0
Progesterone Status       0
Regional Node Examined    0
Reginol Node Positive     0
Survival Months           0
Status                    0
dtype: int64
```

2.Encoding Categorical Variables

To prepare this dataset for machine learning models, categorical variables should be transformed into numerical ones. This was done through label encoding by the `LabelEncoder` from scikit-learn.

3.Handling Outliers

I used the z-score method to detect any outliers within each numerical feature and got rid of them by removing any data points with a z-score greater than 3 from the dataset.

4.Dimensionality Reduction

In this project, Principal Component Analysis (PCA) was used for dimensionality reduction with a variance threshold of 0.95. After PCA, there are 9 Principal Components kept.

5. Feature Importance Ranking:

Feature Importance:

| | Principal Component | Importance |
|---|---|---|
| 0 | PC1 | 0.244434 |
| 1 | PC2 | 0.134738 |
| 2 | PC3 | 0.105435 |
| 3 | PC4 | 0.097993 |
| 4 | PC5 | 0.091158 |
| 5 | PC6 | 0.085601 |
| 6 | PC7 | 0.072527 |
| 7 | PC8 | 0.067278 |
| 8 | PC9 | 0.053734 |

(b)Five different models were trained and evaluated:
1. KNN
2. Naive Bayes
3. Decision Tree
4. Random Forest
5. Gradient Boosting

The models were evaluated using accuracy, precision, recall, and F1 score. The results were displayed in a table format.

Original Models:

| | Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 0 | KNN | 0.878472 | 0.819505 | 0.972366 | 0.889415 |
| 1 | Naive Bayes | 0.752604 | 0.756098 | 0.749568 | 0.752819 |
| 2 | Decision Tree | 0.950521 | 0.910377 | 1.000000 | 0.953086 |
| 3 | Random Forest | 0.985243 | 0.971477 | 1.000000 | 0.985532 |
| 4 | Gradient Boosting | 0.848090 | 0.840067 | 0.861831 | 0.850810 |

(C)Hyperparameter Tuning

Hyperparameter tuning was performed for two selected algorithms: KNN and Gradient Boosting.

For KNN, a grid search was conducted with the following hyperparameters:
- `n_neighbors`: [3, 5, 7, 9, 11, 13, 15]
- `weights`: ['uniform', 'distance']
- `p`: [1, 2]

The best parameters found for KNN were:
- `n_neighbors`: 3

- `weights`: 'distance'
- `p`: 1

For Gradient Boosting, a grid search was conducted with the following hyperparameters:
- `n_estimators`: [50, 100, 200, 500]
- `learning_rate`: [0.01, 0.1, 0.15, 0.2]
- `max_depth`: [1, 2, 3, 4, 5]

The best parameters found for Gradient Boosting were:
- `n_estimators`: 500
- `learning_rate`: 0.15
- `max_depth`: 5

Evaluation of the best models, with regard to hyperparameters was done on the test set and displayed as table. Both models perform better after tunning, indicating the effectiveness of the grid search tunning technique.

The feature importance for that GradienBoosting has given a list which shows that PC6 is first, followed by PC5 and finally PC1

```
Tuned Models:
               Model  Accuracy  Precision  Recall  F1 Score
0                KNN  0.922743   0.866766     1.0  0.928629
1  Gradient Boosting  0.977431   0.957025     1.0  0.978041
Feature Importances (Gradient Boosting):
  Feature  Importance
5     PC6    0.269867
4     PC5    0.170146
0     PC1    0.164686
2     PC3    0.090111
7     PC8    0.077364
6     PC7    0.064476
8     PC9    0.057666
3     PC4    0.056816
1     PC2    0.048869
```

(d)Conclusion

Various data mining techniques are used in this task to predict how long a patient with breast cancer is likely to survive. Preprocessing of the dataset such as categorical variables encoding, outlier handling and dimension reduction using PCA were applied.

Five different models were trained and evaluated. The results show that the tuning did significantly improve the model performance of KNN and Gradient Boosting. The findings show that the used methods are effective in correctly distinguishing patients based on survivability outcomes.