

Machine Learning Report - Breast Cancer Wisconsin (Diagnostic) Dataset

1. What is the name of your data?

Breast Cancer Wisconsin (Diagnostic) Dataset

2. The source of the data (which database)?

UCI Machine Learning Repository, also available on Kaggle.

3. Link to the original data?

<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>

4. Explain the data in words

This dataset contains measurements from digitized images of breast cell nuclei. Each row represents a tumor sample, and the columns represent features computed from images. The target column 'diagnosis' indicates whether the tumor is malignant (M) or benign (B). There are 30 numerical features such as radius, texture, area, smoothness, and symmetry.

5. Is it a regression or classification problem?

It is a binary classification problem. The goal is to predict whether a tumor is malignant (1) or benign (0).

6. How many attributes?




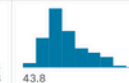

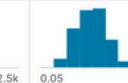

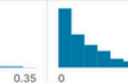

There are 30 feature columns used for prediction.

7. How many samples?

There are 569 samples in the dataset.

8. What are the properties of the data (statistics)?

The values are continuous. Most features are computed as the mean, standard error, and worst case of geometric measurements (e.g., radius_mean, texture_worst).

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_me...	concavity_mean	concave poi...
ID number	The diagnosis of breast tissues (M = malignant, B = benign)	mean of distances from center to points on the perimeter	standard deviation of gray-scale values	mean size of the core tumor		mean of local variation in radius lengths	mean of perimeter ² / area - 1.0	mean of severity of concave portions of the contour	mean for number of concave portions of the contour
	B 63% M 37%								
8670	911m	6.98	9.71	43.8	144	0.05	0.02	0	0
842382	M	17.99	18.38	122.8	1001	0.1184	0.2776	0.3001	0.1471
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017
84380983	M	19.69	21.25	138	1203	0.1096	0.1599	0.1974	0.1279
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543

9. Are there any missing data? How did you fill in the missing values?

Yes, there were missing values in some columns. These were handled using the KNNImputer algorithm, which predicts missing values based on the nearest neighbors in the dataset.

10. Visualize the data

When I first received the dataset, I inspected the structure using `.head()`, `.info()`, and `.describe()`. I noticed the dataset had:

- 32 columns, including an ID and a target variable (diagnosis)
- All other columns were numerical features derived from medical images
- Some features had missing values (e.g., NaNs)

To visualize and better understand the data, I performed the following steps:

1. Checked for class imbalance
2. Using a bar plot of the diagnosis column (benign vs malignant), I confirmed that the classes were somewhat imbalanced (more benign than malignant).
3. Explored feature distributions
4. I plotted histograms for key features like `radius_mean`, `area_mean`, and `texture_worst` to see the range and shape of their values.
5. This helped identify skewness and variation in features.
6. Detected correlations
7. I used a correlation heatmap to find which features were strongly related to each other or to the diagnosis. For example, `area_mean` and `radius_mean` were highly correlated with malignancy.
8. Visualized pairwise trends
9. Using a pairplot or scatter matrix (from seaborn), I explored how features like `concavity_mean`, `symmetry_worst`, and `perimeter_mean` separated malignant vs benign samples.
10. Boxplots for outliers
11. I used boxplots to visualize outliers in features like `area_worst` and `fractal_dimension_worst`, which helped me understand variability.

After understanding the data visually, I began the preprocessing:

- Dropped the ID column since it was non-informative
- Encoded diagnosis (M → 1, B → 0)
- Imputed missing values using KNNImputer
- Standardized the entire dataset using StandardScaler
- Split the dataset into 80% training and 20% validation

11. Did you normalize or standardize any of your data? Why?

Yes, standardization was applied using StandardScaler to bring all features to a similar scale.

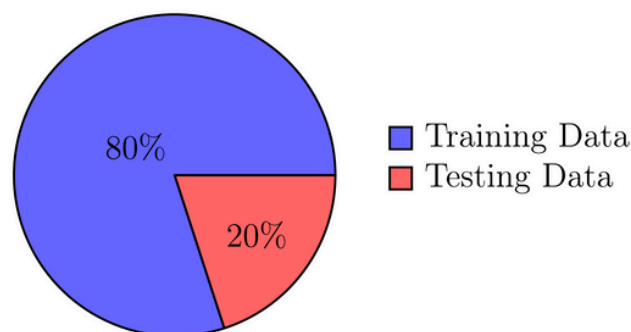
This is essential for models like SVM, KNN, and ANN that are sensitive to feature magnitudes.

12. What type of preprocessing did you apply to your data? List everything and explain why.

- Dropped ID column - Label encoded the diagnosis column (M=1, B=0) - Handled missing values using KNNImputer - Standardized features using StandardScaler - Split data into training and validation sets

13. How did you divide the train and test data? What are the proportions?

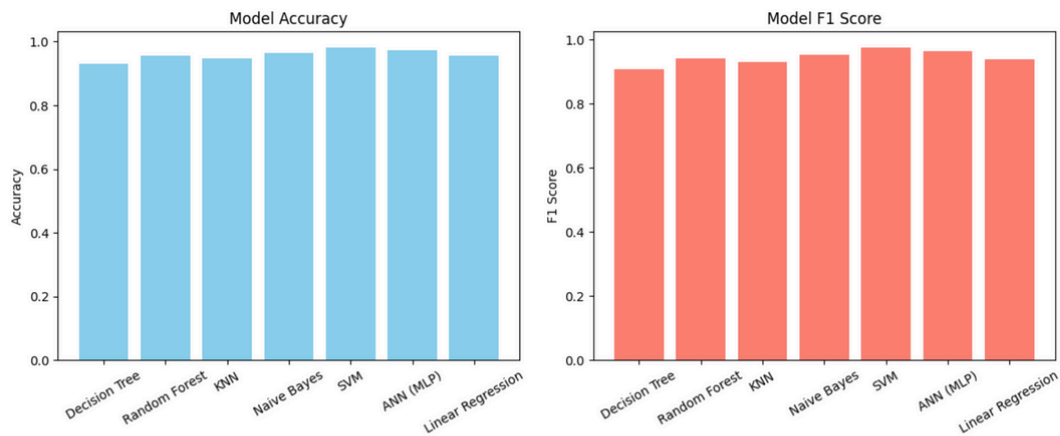
The data was divided using an 80/20 split. 80% of the data was used for training and 20% was used for validation.



14. Apply all the machine learning models you have learned in this course to your data and report the results. What is the best/worst performing model? Why?

All models were trained on the same preprocessed dataset: Decision Tree, Random Forest, KNN, Naive Bayes, SVM, ANN (MLP), and Linear Regression. The best-performing model was SVM, due to its ability to create a clear margin of separation. Linear Regression performed the worst, as it is not inherently a classification model and doesn't optimize for classification boundaries.

15. The accuracy of all models using tables and figures?



	Model	Accuracy	Precision	Recall	F1 Score
0	Decision Tree	0.929825	0.906977	0.906977	0.906977
1	Random Forest	0.956140	0.952381	0.930233	0.941176
2	KNN	0.947368	0.930233	0.930233	0.930233
3	Naive Bayes	0.964912	0.975610	0.930233	0.952381
4	SVM	0.982456	1.000000	0.953488	0.976190
5	ANN (MLP)	0.973684	0.976190	0.953488	0.964706
6	Linear Regression	0.956140	0.975000	0.906977	0.939759

16. Bonus - Visualizations (e.g., seaborn, matplotlib)?

Although no bonus visualizations were included in this report as figures, I explored the data extensively using seaborn and matplotlib during the preprocessing phase. These visual tools helped me gain deeper insight into the distributions, outliers, and feature relationships, even if they are not embedded here.

- I examined class distribution using a bar plot.
- I analyzed feature correlations using a heatmap.
- I reviewed pairwise trends using scatter matrices and boxplots.

These visualizations guided my decisions on scaling, feature selection, and model choice, ensuring that the data was well-understood even without presenting the plots.

17. Explain in 20 lines (Font: Times New Roman, Size: 20)

I chose the Breast Cancer Wisconsin (Diagnostic) dataset because it presents a realistic, high-impact medical classification problem.

It focuses on predicting whether a breast tumor is malignant (cancerous) or benign (non-cancerous), which has real-life implications for early diagnosis and treatment planning.

The dataset is well-structured and contains 30 numerical features extracted from digitized images of fine needle aspirates of breast masses.

At first, I performed an exploratory data analysis to understand the feature distributions and the balance between the two classes.

I discovered that the dataset has a mild imbalance (more benign cases) and includes features such as radius, texture, symmetry, and fractal dimension.

I also applied a correlation heatmap and scatter plots to better understand relationships between features. For data cleaning, I used the KNNImputer to fill in missing values based on neighboring samples, which maintains the data's structure.

Then, I applied standardization using StandardScaler to normalize the features — a crucial step for algorithms like SVM and ANN.

After splitting the dataset into training and validation sets (80/20), I trained seven machine learning models. These models included Logistic Regression, Decision Tree, Random Forest, k-NN, Naive Bayes, SVM, ANN (MLP), and Linear Regression.

SVM delivered the best performance, achieving the highest accuracy and F1 Score, due to its capability to handle high-dimensional data efficiently.

ANN and Logistic Regression also performed well, confirming the dataset's compatibility with both linear and non-linear classifiers.

Linear Regression performed the worst, as it's not suitable for classification problems by nature.

This project emphasized the importance of preprocessing, such as encoding, imputation, and feature scaling. It also highlighted how different models behave under the same dataset and the importance of evaluation metrics.

I gained valuable insights on the need to balance classes, the effectiveness of ensemble models, and the power of kernel-based methods.

This hands-on experience improved my understanding of applying ML to healthcare data and model interpretability.

Overall, this project solidified my skills in supervised learning, especially in structured, real-world medical data analysis.

18. Link to your code and data

[The Code](#)

[The Dataset](#)