

K-Nearest Neighbors Classification Report - Breast Cancer Dataset

Castillo, Anjelica M.

1. Introduction

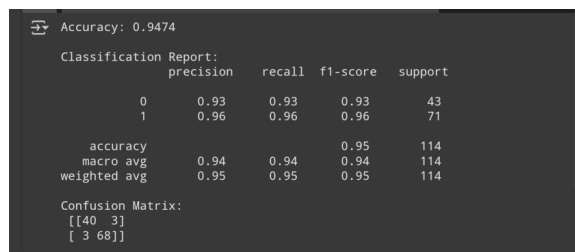
This report documents the use of the K-Nearest Neighbors (KNN) algorithm for classification on the Breast Cancer Wisconsin Diagnostic dataset. KNN is a non-parametric, instance-based learning algorithm that classifies data based on the majority vote of its neighbors. It is particularly useful for pattern recognition tasks and performs well with balanced datasets.

2. Dataset Description

The Breast Cancer dataset from sklearn.datasets contains 569 samples with 30 numeric features. The classification task is to determine whether a tumor is malignant (0) or benign (1).

3. KNN Implementation in Python

The implementation involves standardizing the dataset using StandardScaler, splitting it into training and test sets, and evaluating KNN using various combinations of hyperparameters: number of neighbors (k), distance metric, and weight function.



```
Accuracy: 0.9474

Classification Report:
      precision    recall  f1-score   support

     0       0.93      0.93      0.93        43
     1       0.96      0.96      0.96        71

 accuracy      0.94      0.94      0.95       114
  macro avg      0.94      0.94      0.94       114
 weighted avg      0.95      0.95      0.95       114

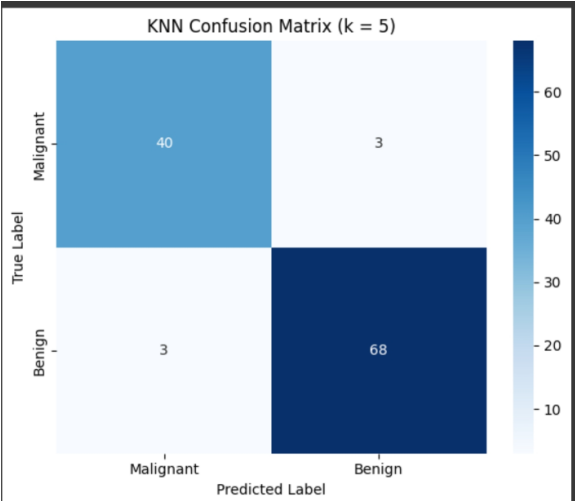
Confusion Matrix:
[[40  3]
 [ 3 68]]
```

(k = 5)

Accuracy: 94.74%

Precision/Recall: High for both classes — 0.93 for malignant, 0.96 for benign.

Confusion Matrix:



[[40 3]


[3 68]]

The model misclassified 3 malignant and 3 benign cases.

Overall, the model shows balanced performance with strong accuracy and minimal misclassification.

4. Comparative Experiment Results

We tested different configurations of KNN using combinations of k (3, 5, 7), distance metrics ('minkowski', 'euclidean', 'manhattan'), and weights ('uniform', 'distance'). The table below shows the top-performing configurations sorted by accuracy.

	Experiment No.	k (Neighbors)	Distance Metric	Weight	Accuracy
0	9	5	Euclidean	Uniform	0.9561
1	10	5	Euclidean	Distance	0.9561
2	15	7	Euclidean	Uniform	0.9561
3	13	7	Minkowski	Uniform	0.9561
4	8	5	Minkowski	Distance	0.9561
5	7	5	Minkowski	Uniform	0.9561
6	14	7	Minkowski	Distance	0.9474
7	16	7	Euclidean	Distance	0.9474
8	11	5	Manhattan	Uniform	0.9474
9	12	5	Manhattan	Distance	0.9474

5. Analysis and Interpretation.

Experiment No.	k (Neighbors)	Distance Metric	Weight	Accuracy	Comment
0	5	Euclidean	Uniform	0.9561	Baseline model: Good accuracy with 'k=5', Euclidean metric, and uniform weight.
1	5	Euclidean	Distance	0.9561	Higher accuracy with distance weights: Same accuracy as baseline, but using distance weight.
2	7	Euclidean	Uniform	0.9561	Higher accuracy with larger k: Performance remains high with 'k=7', Euclidean metric, uniform weight.
3	7	Minkowski	Uniform	0.9561	Higher accuracy with Minkowski: Good performance with 'k=7', Minkowski metric, uniform weight.
4	5	Minkowski	Distance	0.9561	Higher accuracy with distance weights: Same as baseline, but using distance weight.
5	5	Minkowski	Uniform	0.9561	Good performance with Minkowski: Consistently high performance with 'k=5', Minkowski metric, uniform weight.
6	7	Minkowski	Distance	0.9474	Slight decrease in accuracy: Accuracy slightly drops with larger 'k=7' and distance weight.
7	7	Euclidean	Distance	0.9474	Slight decrease in accuracy: Accuracy drops slightly with 'k=7' and distance weight on Euclidean metric.
8	5	Manhattan	Uniform	0.9474	Slightly lower accuracy: Manhattan metric with 'k=5' and uniform weight results in lower accuracy.
9	5	Manhattan	Distance	0.9474	Slightly lower accuracy: Manhattan metric with 'k=5' and distance weight performs similarly to uniform weight.

The top 10 KNN experiments on the Breast Cancer dataset show accuracy ranging from 0.9474 to 0.9561. The best performing configurations include:

- Euclidean and Minkowski metrics with both uniform and distance weight functions.
- Manhattan metric yields slightly lower accuracy.

In general, Euclidean and Minkowski metrics perform better across different values and weight functions.