



# International Journal of Research in Academic World



Received: 06/January/2025

IJRAW: 2025; 4(2):44-53

Accepted: 11/February/2025

## Boosting Disease Prediction Accuracy: Comparative Evaluation of KNN

\*<sup>1</sup>Shaheen Khatoon, <sup>2</sup>Aparna Tiwari and <sup>3</sup>Rinku Raheja

<sup>\*1,2</sup>Student, Department of Computer Science, National PG College, Lucknow, Uttar Pradesh, India.

<sup>3</sup>Assistant Professor, National PG College, Lucknow, Uttar Pradesh, India.

### Abstract

These days, machine learning is important in a variety of domains and helps businesses make wise judgments. It is important to note that (ML) is currently a crucial idea to expand artificial intelligence (AI) of smart systems by reducing the need for human interaction due to humans' limited memory and brain capacities. Artificial Intelligence (AI) is a branch of computer science study that tries to replicate human cognitive processes in machines. It has applications in several domains such as industry, commerce, energy, transportation, health, and security. Furthermore, artificial intelligence cleared the way for the development of contemporary technologies that struggle to automate processes and function without human intervention.

One of the greatest classification techniques is the kNN algo, which is a well-known pattern recognition technique. It is among the most straightforward machine learning methods for categorization. This paper explores the use of machine learning algorithms, specifically the k-nearest neighbour (KNN) algorithm, in disease risk prediction. The datasets were related to different disease contexts. For comparative study, we took into account the accuracy, precision, and recall performance metrics. These variations' average accuracy values varied from 64.22% to 83.62%. The ensemble method KNN (82.34%) and Hassanaat KNN (83.62%) both displayed the highest average accuracy. To evaluate each variant and compare the outcomes, a relative performance index is also suggested, depending on each performance indicator. Based on the accuracy-based version of this index, this study found that the Hassanaat KNN variant performed the best, followed by the ensemble approach KNN. This paper provides an overview of the kNN algorithm and its related literature, delves into the algorithm's concept, stages for implementation, and implementation code, and evaluates the benefits and drawbacks of the different improvement approaches with better accuracy score 97.3%.

**Keywords:** kNN algorithm, k nearest neighbor algorithm, machine learning, blockchain.

### Introduction

The process of giving a system the ability to forecast future events is called machine learning (ML). It makes use of a number of different algorithms to evaluate past data and draw conclusions from it in order to predict future results for data that has not yet been observed. Actually, through the training process, it looks for patterns in prior events that are concealed in related data and creates a predictive model.

There are several learning categories, such as reinforcement, unsupervised, supervised, and semisupervised learning, depending on the kinds of issues that machine learning concepts are expected to handle. The ML model in reinforcement learning depends on rewards given to the agent during interactions with the environment. Its goal is to evaluate the best practices that improve the surroundings. It is extensively utilized in the manufacturing, robotics, gaming, and advertising industries. Unsupervised learning involves the machine learning model attempting to learn from an unlabeled dataset in tasks like clustering, anomaly detection, and density estimation. Supervised learning is the other kind of machine learning algorithm that only learns from labeled data in order

to determine an appropriate mapping function. Credit scoring, sentiment analysis, email filtering, and risk assessment are among the typical uses for it <sup>[1]</sup>. Semisupervised learning combines supervised and unsupervised learning methods with both labeled and unlabeled data. Examples of applications for semisupervised learning include text categorization, speech recognition, and fraud detection. The fundamental and central technology of data mining is classification. It is widely applicable in many domains, including management, business, scientific research, and decision-making. The machine learning literature has numerous classification techniques suggested for a variety of uses. The most well-known classification methods are neural networks, support vector machines, random forests, decision trees, and naive Bayes. Furthermore, one of the most important classification methods that is also applicable to regression tasks is the k-nearest neighbors (KNN) technique. KNN, commonly referred to as "lazy learning," is an instance-based learning paradigm. This indicates that the training dataset won't be fully digested until this algorithm comes across a specific test query for prediction. In actuality, KNN is capable of carrying

out a thorough search in big dimensions. The training set for a KNN classifier is thought to be an  $m$ -dimensional space of patterns used to find the  $k$ -closest tuples to a given test question. A particular distance metric, such as the Euclidean distance, is used to characterize the closeness. Before calculating the distance, the min-max normalization approach can also be used to balance the effects of different ranges.

The  $k$  closest neighbors of a data record  $t$  are obtained in order to classify it; this creates the neighborhood around  $t$ . The classification for  $t$  is often determined by majority vote among the data records in the neighborhood, with or without the use of distance-based weighting. But in order to use  $k$ NN, we must select a suitable value for  $k$ , and the classification's outcome greatly depends on this number. The  $k$ NN approach is somewhat  $k$ -biased. While there are other methods for determining the  $k$  value, one straightforward method is to repeatedly run the algorithm with various  $k$  values and select the one that performs the best. Wang [2] suggested looking at several sets of nearest neighbors rather than just one set of  $k$ -nearest neighbors in order to make  $k$ NN less dependent on the selection of  $k$ . The suggested formalism, which is based on contextual probability, aims to provide a more dependable support value that more accurately discloses the true class of  $t$  by combining the support of several sets of nearest neighbors for different classes. Though the approach is less dependent on  $k$  and can attain classification performance similar to that for the best  $k$ , it is still very slow in its simple version, requiring  $O(n^2)$  to classify a new instance.

## Related Works

### i). Breast Cancer Dataset Studies

- **Wisconsin Breast Cancer Dataset:** This dataset is commonly used for training KNN models. Papers often analyze its features, including tumor size and cell characteristics.
- **UC Irvine Machine Learning Repository:** Many studies use datasets from this repository for developing and testing predictive models.

### ii). KNN Algorithm in Medical Diagnosis

- **Comparison with Other Algorithms:** Research comparing KNN to other classifiers like SVM, Decision Trees, or Neural Networks in predicting breast cancer outcomes.
- **Feature Selection Techniques:** Studies focusing on how feature selection impacts KNN performance, including methods like PCA or recursive feature elimination.

### iii). Performance Metrics in KNN

- **Evaluation Metrics:** Papers discussing accuracy, sensitivity, specificity, and AUC (Area under the Curve) for assessing KNN model performance in breast cancer prediction.

### iv). Hybrid Approaches

- **KNN with Ensemble Methods:** Research that combines KNN with ensemble methods (like Random Forest or boosting) to improve prediction accuracy.

### v). Dimensionality Reduction Techniques

- **Using PCA or t-SNE:** Studies that employ dimensionality reduction techniques before applying KNN to enhance performance on high-dimensional datasets.

### vi). Preprocessing Techniques

- **Data Normalization and Standardization:** Research discussing the importance of data preprocessing steps in improving KNN accuracy for breast cancer prediction.

### vii). Real-world Applications

- **Clinical Decision Support Systems:** Papers that explore how KNN-based models can be integrated into clinical settings for breast cancer screening and diagnosis.

### viii). Case Studies

- **Specific Studies on Populations:** Research focusing on specific demographic groups or risk factors and how KNN can tailor predictions based on these.

### ix). Machine Learning Frameworks

**Software Implementations:** Studies that leverage libraries like Scikit-learn or TensorFlow for implementing KNN in breast cancer prediction.

### Comparative Study

Related Works First, this section briefly explains the most relevant works to our study. After that, it describes the various types of KNN algorithms in the literature. 2.1. A Review of Related Literature K-nearest neighbors (KNN) is a supervised machine learning method that can be utilized for both classification and regression tasks. KNN considers the similarity factor between new and available data to classify an object into predefined categories. KNN has been widely used in many fields such as industry [7–9], machine engineering [10], health [11–13], marketing [14], electrical engineering [15], security [16–18], manufacturing [19], energy [20–22], aerial [23], environment [24], geology [25,26], maritime [27,28], geographical information systems (GIS) [29], and transportation [30]. In the field of industry, the authors introduced a hybrid bag of features (HBoF) in [7] to classify the multiclass health conditions of spherical tanks. In their HBoF-based approach, they extracted vital features using the Boruta algorithm. These features were then fed into the multiclass KNN to distinguish normal and faulty conditions. This KNN-based method yielded high accuracy, surpassing other advanced techniques. In [8], the authors have reported that the KNN algorithm attained a great performance according to different evaluation metrics for predicting the compressive strength of concrete. Their model was highly recommended in the construction industry owing to the fact that it required fewer computational resources to be implemented. In another work [9], a KNN-based method was developed to efficiently detect faults by considering outliers on the basis of the elbow rule for industrial processes. Similarly, in machine engineering, the authors presented a KNN-based fault diagnosis approach for rotating machinery, which was validated through testing on the bearing datasets [10]. In the field of health, Salem *et al.* [11] investigated a KNN-based method for diabetes classification and prediction that can contribute to the early treatment of diabetes as one of the main chronic diseases. The outperformance of their method was evaluated in terms of various metrics and approved its applicability in the healthcare system for diabetes compared to other techniques. In another work [12], the researchers focused on an approach to early detect placental microcalcification by the KNN algorithm, which could lead to the improvement of maternal and fetal health monitoring during pregnancy. The results gained from real clinical

datasets revealed the efficiency of the proposed model for pregnant women. In [13], a wearable device was introduced on the basis of embedded insole sensors and the KNN machine learning model for gait analysis from the perspective of controlling prostheses. The results of applying the KNN algorithm showed the success of the device in predicting various gait phases with high accuracy. In marketing, Nguyen *et al.* [14] represented a beneficial recommendation system via KNN-based collaborative filtering, in which similar users according to their cognition parameters were effectively grouped to obtain more relevant recommendations in different e-commerce activities. In the field of electrical engineering, Corso *et al.* [15] developed a classification approach for the insulator contamination levels by the KNN algorithm with high accuracy to predict insulating features as predictive maintenance of power distribution networks. In the field of security, a KNN-based technique was proposed in [16] to classify botnet attacks in an IoT network. In addition, the forward feature selection was utilized in their technique to obtain improved accuracy and execution time in comparison to other benchmark methods. In [17], a novel security architecture was provided to effectively deal with forged and misrouted commands in an industrial IoT considering different technologies, namely, KNN, random substance learning (RSL), software-defined network Electronics 2023, 12, 3828 4 of 2(SDN), and a blockchain-based integrity checking system (BICS). In another work [18], an intrusion detection system (IDS) was implemented for wireless sensor networks by employing KNN and arithmetic optimization algorithms in that an edge intelligence framework was utilized for denial-of-service attacks in WSNs. In the field of manufacturing [19], Zheng *et al.* introduced a consensus model on the basis of KNN to classify cloud transactions regarding their priorities. In their model, different parameters, e.g., service level agreements (SLA), cloud service providers (CSP), cloud service consumers (CSC), and smart contract (SC) types, were utilized for distance calculation in the KNN algorithm. In the field of electrical energy [20], a short-term load forecasting technique was proposed by the weighted KNN algorithm to achieve high accuracy for fitting the model. Similarly, in another work [21], the authors focused on an effective KNN-based model for the failure diagnosis of wind power systems regarding particle swarm optimization.

In [22], the researchers investigated seismic energy dissipation for rocking foundations in the seismic loading process by means of different supervised machine learning algorithms, including KNN, support vector regression (SVR), and decision tree regression (DTR). The k-fold cross-validation was applied to the mentioned algorithms and, according to the results, KNN and SVR outperformed DTR in terms of accuracy. In the aerial field, Martínez-Clark *et al.* [23] represented a KNN-based method as a flock inspiration from nature for a group of unmanned aerial vehicles (UAVs). In their method, an optimal number of UAVs was obtained with regard to heading synchronization in drone implementation. In the environment field, a predictive model was developed in [24] for the construction and demolition waste rate forecast by means of KNN and principal component analysis (PCA). In the field of geology, the authors [25] reported that the KNN algorithm was utilized for a three-dimensional envision of the stratigraphic structure of porous media related to sedimentary formations. In a similar work in geology, Bullejos *et al.* [26] implemented a technique for the evaluation of KNN prediction confidence in that the three-dimensional model for the stratigraphic structure of porous media is approved. The results of their KNN-based method contributed to improving the predictability of groundwater investigations. In the maritime field, the authors [27] introduced a novel trajectory model for offshore waters with the means of KNN and long short-term memory (LSTM) techniques for high density and low-density trajectories, respectively. Their experimental results revealed that the mean square error is significantly decreased in comparison with the previous artificial neural-network-based models. Another research [28] in this area employed a KNN-based approach. The authors utilized the PCA method for predictive ship maintenance, with a specific emphasis on propulsion engines. In the field of GIS, the authors [29] presented a predictive model with an ensemble KNN for typhoon tracks, which included supervised classification and regression applied on various samples of typhoons gaining experimental results with high accuracies and low running times. In the field of transportation [30], the effect of missed data was detected through the KNN classifier for the prediction of traffic flow on public roads. Their results show that the presented method was efficiently applicable for smart transportation in urban traffic control

**Table 1:** Advantages and Disadvantages of KNN

Advantage	Disadvantage
1. Simplicity and Ease of Use: KNN is simple to comprehend and use, making it available to researchers and practitioners.	i). Computationally intensive: KNN needs computing the length of time, especially for big datasets, between each training instance and the query instance.
2. No Assumptions About Data Distribution: Since KNN is a non-parametric approach, it makes no assumptions about the data's distribution, which might be advantageous for datasets from the real world.	ii). Sensitivity to Irrelevant characteristics: Since KNN takes into account all dimensions while computing distances, the existence of irrelevant characteristics may have a detrimental effect on the model's performance.
3. Flexibility: KNN is applicable to tasks involving both regression and classification. Its adaptability makes it useful for a wide range of issues beyond breast cancer prognosis.	iii). The Curse of Dimensionality: o KNN performance may deteriorate as feature counts rise. The distances between points become less relevant in high-dimensional spaces
4. Effective with enormous Datasets: If the dimensionality is controlled, it can handle enormous datasets fairly well.	iv). Choice of K: o The performance of KNN is greatly dependent on the choice of K (the number of neighbors). An unsuitable selection may result in either an underfit or an overfit.
5. Good Performance with short Datasets: When the class distributions are obvious and the training dataset is short, KNN can function well.	v). Storage prerequisites: Because CNN requires all training data to be stored, it may not be feasible for very big datasets or situations where memory is limited.
6. The algorithm works well in settings where data is constantly changing because it can readily adjust to new data.	vi). Inability to Interpret: Although KNN can classify data, it cannot explain the reasoning behind a given choice, which might be problematic in medical applications where interpretability is crucial.

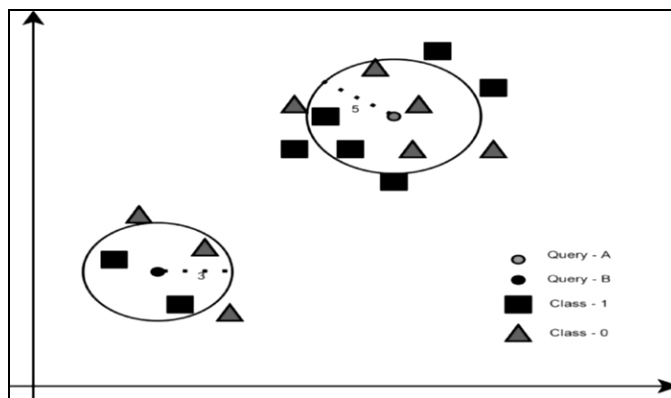


Fig 1: Visual illustration of the KNN algorithm.

Table 2: `df.drop(columns=['id', 'Unnamed: 32'],inplace=True)`  
`df.head()`

Diagnosis	Radius Mean	Texture Mean	Perimeter Mean	Area Mean	Smoothness Mean	Compactness Mean	Concavity Mean	Concave Points Mean	Symmetry Mean
M	17.99	10.38	122.80	1001.0	0.11840	0.2760	0.3001	0.14710	0.2419
M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

Table 3: `df.drop(columns=['id', 'Unnamed: 32'],inplace=True)`  
`df.head()`

Radius_worst	Texture_worst	Perimeter_worst	Area_worst	Smoothness_worst	Compactness_worst	Concavity_worst	concave points_worst	Symmetry_worst	Fractal_dimension_worst
25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

**Implementation**

**K-nearest Neighbour Algorithm and its Different Variants**

**The Classic KNN Algorithm:** The technique uses a variable parameter called k, which stands for "nearest neighbour" in English. Finding the closest data point or neighbours for a query from a training dataset is how the KNN algorithm operates. The closest distances from the query point are used to determine which data points are the closest. It finds the class that appears the most by using a majority voting mechanism after discovering the k nearest data points. The final categorization for the query is determined by looking at the class that showed up the most. An illustration is shown in Figure 1. Given that k for Query B is three, it looks for the three closest neighbours and discovers that two of them belong to class 1 and one to class 0. It then assigns its class a number of one using the majority voting rule. Similar to this, Query A classifies its class as 0 because k is 5 and there are more neighbours that are classified as Class 0.

**Methods**

**Datasets for Research:** To prevent bias, the research is based on a single primary domain, medical domains, and other secondary domains that are completely random. The datasets included in this investigation are shown in Table 1 along with their corresponding characteristics, including the quantity of features, size of the data, and domain. These were extracted

from OpenML21, Kaggle19, and the UCI Machine Learning Repository (20). The features, properties, and volumes of the datasets vary, and the majority pertain to the medical field in terms of the significance of illness risk prediction.

**Metrics for Comparing Performance:** Confusional grid. Renowned academic performance measures that center on the application of the confusion matrix<sup>28</sup> were employed to analyze the data. Figure 2 shows the matrix's visual representation. The results of classifications are used to create the matrix, which includes four main characteristics that display the outcome data. In the event that the true value is 1 and the classification is predicted to be 1, the outcome is categorized as true positive (TP) data. The same principle, known as true negative (TN), is centered around the number 0. False positives (FP) are results where the prediction is 1 and the true value is 0. False negatives (FN) are the opposite of FPs.

Three performance metrics are created in this study using the confusion matrix: accuracy

By taking all of the true predictions and dividing them among all of the predicted values, including the true forecasts, the accuracy measure is computed.

Accuracy = (TP + TN)/TP + TN + FP + FN where TP, TN, FP and FN are the true positive, true negative, false positive and false negative cases of the result data, respectively.

```
[9]: #knn algo
    from sklearn.neighbors import KNeighborsClassifier
    knn=KNeighborsClassifier(n_neighbors=5)
    knn.fit(x_train,y_train)

[9]: KNeighborsClassifier
     KNeighborsClassifier()

[10]: from sklearn.metrics import accuracy_score
      y_pred=knn.predict(x_test)
      accuracy_score(y_test,y_pred)

[10]: 0.9736842105263158
```

Fig 2: Model Training

## Frequently Asked Questions

### i). In illness diagnosis, what is the K-Nearest Neighbors (KNN) algorithm?

A supervised learning method called KNN is employed for categorization tasks like illness diagnosis. In order to classify new data points (like a patient's symptoms), it compares them to the training set's closest known data points (like previously diagnosed patients), where "K" denotes the number of nearest neighbors taken into account. KNN aids in the identification of patterns from medical data to forecast situations or results in the diagnosis of disease.

### ii). How does KNN function in relation to diagnosing medical conditions?

- A multidimensional space is plotted using the patient's medical data.
- The KNN method determines the distance—typically the Euclidean distance—between the data of the new patient and the data of previous patients.
- Patients with established diagnoses are identified as the "K" closest neighbors by the algorithm.
- The algorithm predicts the diagnosis of the new patient based on the majority label (disease or no disease) of these neighbors.

### iii). Which distance measures are frequently employed in KNN for medical diagnosis?

- Measures the distance in a straight line between two points in space, or Euclidean Distance. Often in data that is continuous.
- **Manhattan Distance:** This method adds the absolute differences in coordinates between two points to find their distance.  
Minkowski Distance: When you wish to alter the distance metric, this generalized version of the Manhattan and Euclidean distances is helpful.
- **Hamming Distance:** applied to binary or categorical data (e.g., the existence or non-existence of an illness).

### iv). What are a few of KNN's shortcomings in terms of medical diagnosis?

- **Computational Complexity:** KNN uses a lot of computation since it must compare every new patient to every piece of data in the training set, which uses a lot of resources when dealing with big datasets.
- **Feature Scaling Sensitivity:** Appropriate data normalization is necessary because the algorithm is sensitive to the feature scale.
- **Curse of Dimensionality:** Accuracy may decrease as

the number of characteristics (dimensions) rises and the significance of the distances between points decreases.

### v). What are KNN variants used in illness diagnosis?

- **Weighted KNN:** In medical datasets where some traits are more crucial, weighted KNN gives closer neighbors a greater influence on classification than faraway ones.
- **KNN with Dimensionality Reduction:** The curse of dimensionality can be lessened by reducing the amount of features through methods like PCA (Principal Component Analysis).
- **Fuzzy KNN:** This variation allows for more nuanced decision-making in ambiguous situations by providing a likelihood of belonging to various classes rather than a clear classification.

### vi). Why is the "K" value important in KNN for medical data?

It is crucial to select the appropriate value for "K." A small "K" (for example, K=1) can cause the model to be excessively responsive to noise, while a large "K" could potentially obscure important details in the data. Cross-validation is frequently employed to identify the best "K" value for medical diagnosis purposes.

### vii). How might other machine learning approaches be incorporated with KNN algorithms for disease diagnosis?

#### KNN is frequently combined with other methods:

- **Ensemble Methods:** KNN can be used in ensemble techniques such as boosting or bagging, which integrate predictions from several models to increase the accuracy of diagnoses.
- **Hybrid Models:** By combining the advantages of several methods, KNN can improve disease prediction when paired with neural networks or decision trees.

### viii). What performance measures does KNN offer for diagnosing diseases?

- **Accuracy:** Indicates how accurate a prediction is overall.
- **Precision:** The percentage of accurately anticipated positive diagnoses among all positive diagnoses.
- **Recall (Sensitivity):** The percentage of real positive cases that the model properly detected.
- **F1 Score:** A balanced performance indicator derived from the harmonic mean of recall and precision.
- **ROC-AUC:** A curve that illustrates how true positive rate and false positive rate are traded off.

### ix). What are some actual instances of KNN in the diagnosis of illness?

- **Cancer Diagnosis:** By comparing genetic markers or tumor characteristics to known cases, KNN has been utilized to identify various cancer kinds.
- **Heart Disease Prediction:** KNN has been used to forecast the probability of heart disease based on risk factors like age, blood pressure, and cholesterol levels.
- **Diabetes Prediction:** KNN is used to forecast a patient's likelihood of developing diabetes by analyzing BMI, insulin, glucose levels, and other variables.

### x). In medical diagnostics, how can KNN manage incomplete or missing data?

By forecasting the missing data points depending on their closest neighbors, KNN can be modified to impute missing values. Nevertheless, KNN's efficacy could decline if an excessive amount of data is lacking. Healthcare practitioners may use KNN and its variations' prediction powers to enhance patient outcomes and diagnostic precision by investigating these algorithms' potential.

### Algorithm Flow for KNN in Breast Cancer Prediction

#### i). Data Gathering

- **Input:** A breast cancer dataset including features such as mean radius, texture, perimeter, area, and more (e.g., Wisconsin Breast Cancer Dataset).
- **Goal:** Obtain a labeled dataset that comprises features relating to breast cancer characteristics coupled with diagnosis labels (malignant or benign).

#### ii). Preparing the Data

##### Step 2.1: Managing Missing Values

- Look for missing data and use imputation techniques like mean or median imputation to manage it.

##### Step 2.2: Feature Scaling:

- Standardize or normalize the dataset to ensure that every feature has a uniform scale. Since that KNN depends on distance measurements, this is essential.
- Normalization Formula:

$$[X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}]$$

- **The Formula for Standardization is as Follows:** ( $X_{\text{min}}$ ) and ( $X_{\text{max}}$ ) are the minimum and maximum values of the feature, ( $\mu$ ) is the mean, and ( $\sigma$ ) is the standard deviation. This formula may be used to calculate  $X'$ .

Step 2.3: Divide Data into Sets for Testing and Training

- Using an 80:20 ratio, divide the dataset into training and testing subsets.
- Make sure the distribution of benign and malignant cases in the training and testing sets is equal by using \*stratified sampling\*

#### iii). Establish the Ideal K Value

##### Step 3.1: K Value Range

- For ( $K$ ), begin with a range of values (e.g.,  $K = 1, 3, 5, 7, 9$ , etc.).

##### Step 3.2: Cross-validation

- To assess the model performance for every value of ( $K$ ), use \*k-fold cross-validation\* (often 5 or 10 folds). The chance of overfitting is decreased with the aid of cross-validation.

##### Step 3.3: Select the Ideal K

- Choose the ( $K$ ) that produces the least variance and maximum average accuracy.

#### iv). KNN Classification Algorithm

##### Step 4.1: Compute Distance:

Using a specified distance metric (also known as the Euclidean distance), calculate the distance for each test sample between the test point and each training sample as follows:

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Where the feature vectors of a training point and a test point are denoted, respectively, by ( $q_i$ ) and ( $p_i$ ).

##### Step 4.2: Determine the K Nearest Neighbors:

Choose the ( $K$ ) closest training samples (nearest neighbors) by sorting the calculated distances.

Classify the Test Point (Step 4.4): Based on the majority vote, give the test sample a class label.

#### v). Model Assessment

##### Step 5.1: Precision

- Apply the following formula to get the total classification accuracy on the test dataset:

$$[\text{Accuracy}] = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

##### Step 5.2: Confusion Matrix:

- To examine True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), create a confusion matrix:
- **Precision:**  $[\text{Precision}] = \frac{TP}{TP + FP}$
- **Recall (Sensitivity):**  $[\text{Recall}] = \frac{TP}{TP + FN}$
- **Score for F1:**  $[\text{F1}] = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

##### Step 5.3: ROC Curve and AUC

- To see the trade-off between True Positive Rate (Sensitivity) and False Positive Rate (1-Specificity), plot the Receiver Operating Characteristic (ROC) Curve.
- To assess the discriminatory capability of the model, compute the \*Area Under the Curve (AUC)\*.

#### vi). Hyperparameter Tuning (Optional)

##### Step 6.1: Hyperparameter Tuning with Grid Search

- To investigate other ( $K$ ) values and distance metrics (such as Manhattan and Euclidean), use Grid Search.

##### Step 6.2: Performance Optimization

- Modify settings to optimize recall, accuracy, precision, or other pertinent metrics.

#### vii). Optional Deployment

##### Step 7.1: Real-Time Prediction

After selecting the best model, include it into a system for predicting breast cancer in real time while processing patient data.

**Step 7.2: User Interface**

Create a straightforward user interface that allows medical professionals to enter patient information and get an estimated diagnosis

- i). Compile and prepare the dataset for breast cancer.
- ii). Normalize and scale the data.
- iii). Divide the data into sets for testing and training.
- iv). Apply cross-validation to (K) optimization.
- v). Classify test data using KNN.
- vi). Use ROC-AUC, accuracy, precision, recall, and evaluation metrics.
- vii). Adjust the hyperparameters as desired and run the model.

**Data Preprocessing**

The data pre-processing procedures are essential to ensuring clean and structured data for the best model performance while developing a machine learning model for breast cancer prediction. This is a detailed description of the data pre-processing procedure that includes pertinent Python, NumPy, and Pandas code snippets.

**1. Bringing Libraries in:**

We must first load all necessary libraries, including pickle to save our model, sklearn for machine learning tasks, pandas for data processing, and NumPy for numerical operations.

**NumPy:** For numerical calculations, use NumPy.

**Pandas:** Used to manage and work with data in DataFrames.

**Sklearn:** Offers resources for model training, evaluation, and data splitting.

**Pickle:** To save the learned model for later use, pickle it.

```
[1]: import numpy as np
import pandas as pd
```

Fig 3: Importing libraries

**2. Information Gathering and File Reading**

Open the dataset on breast cancer.

Pandas is a tool that may be used to read CSV files.

Table 4: Information Gathering and File Reading

id	diagnosis	Radius_mean	Texture_mean	Perimeter_mean	Area_mean	Smoothness_mean	Compactness_mean	Concavity_mean	Concave points_mean	Texture_worst
842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	1
842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	2
843009	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	3
843483	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	2
843584	M	20.29	14.34	135.10	1297.0	0.10300	0.13280	0.19800	0.10430	3

**3. Total of Columns That Are Not Null and Null**

The quality of the data is vital. It's critical to look for any missing values. It is possible to either drop or impute

missing data.

You may determine how many null or missing values each feature contains by following this procedure.

```
[3]: df.isnull().sum()
[3]: id 0
diagnosis 0
radius_mean 0
texture_mean 0
perimeter_mean 0
area_mean 0
smoothness_mean 0
compactness_mean 0
concavity_mean 0
concave points_mean 0
symmetry_mean 0
fractal_dimension_mean 0
radius_se 0
texture_se 0
perimeter_se 0
area_se 0
smoothness_se 0
compactness_se 0
concavity_se 0
concave points_se 0
symmetry_se 0
fractal_dimension_se 0
radius_worst 0
texture_worst 0
perimeter_worst 0
area_worst 0
smoothness_worst 0
```

Fig 4: Checking of null values

**4. Eliminating Superfluous Information**

Certain characteristics might not be important for forecasting. For example, features that are wholly unnecessary or identifiers like 'ID' columns should be

removed. Eliminating superfluous data streamlines the dataset and directs the model's attention to relevant aspects.

**Table 5:** df.drop(columns=['id', 'Unnamed: 32'],inplace=True)  
df. head()

Diagnosis	Radius_mean	Texture_mean	Perimeter_mean	Area_mean	Smoothness_mean	Compactness_mean	Concavity_mean	concave points_mean	Symmetry_mean
M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419
M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812
M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069
M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597
M	20.29	14.34	135.10	1297.0	0.10300	0.13280	0.19800	0.10430	0.1809

## 5. Determining the Data's Shape

Knowing the data's form helps us make sure we have the appropriate number of training samples and features.

The number of rows (samples) and columns (features) will be displayed in the output.



```
[5]: df.shape
[5]: (569, 31)
```

**Fig 5:** Checking no of rows and columns

## 6. Dividing Data into Target and Features

The characteristics (independent variables) and the target label (dependent variable), which signals whether the tumor is benign or malignant, must be separated before the model is trained. Here, 'y' is the goal variable, or whether the tumor is benign or malignant, and 'X' contains all of the input features.

## Data Training and Testing

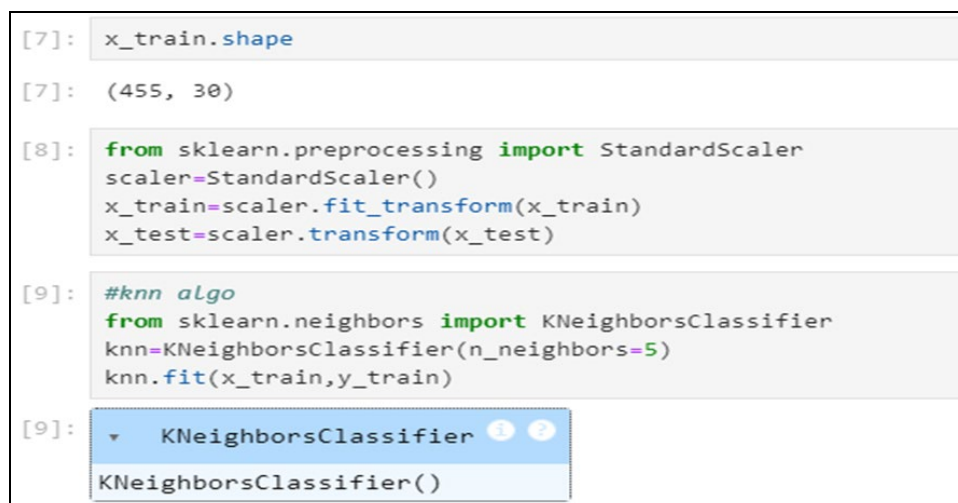
After that, the data is divided into testing and training sets. Usually, 20–30% of the data is used for testing and 70–80% is used for training. This division guarantees that we test the model on unseen data to assess its performance after training it on a portion of the data.

**Table 6:** Data Training and Testing

Radius_mean	Texture_mean	Perimeter_mean	Area_mean	Smoothness_mean	Compactness_mean	Concavity_mean	concave points_mean	Symmetry_mean	Fractal_dimension
14.05	27.15	91.38	600.4	0.09929	0.11260	0.04462	0.04304	0.1537	0
11.13	16.62	70.47	381.1	0.08151	0.03834	0.01369	0.01370	0.1511	0
19.18	22.49	127.50	1270.0	0.08523	0.14280	0.11140	0.06772	0.1767	0
13.81	23.75	95.16	597.8	0.13230	0.17680	0.15580	0.09176	0.2251	0
10.95	21.35	71.90	371.1	0.12270	0.12180	0.10440	0.05669	0.1895	0

## 7. Model Training

Let's use the training data to train a machine learning model, like KNN (k nearest neighbor).



```
[7]: x_train.shape
[7]: (455, 30)

[8]: from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train=scaler.fit_transform(x_train)
x_test=scaler.transform(x_test)

[9]: #knn algo
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train,y_train)

[9]: KNeighborsClassifier
KNeighborsClassifier()
```

**Fig 6:** Training and testing the data



**8. Model Assessment (Accuracy Rating)**

After the model has been trained, it is crucial to compute the accuracy score in order to assess how well it performed using the test data. The percentage of accurate predictions the model made on the test set is represented by the accuracy score.

```
[10]: from sklearn.metrics import accuracy_score
      y_pred=knn.predict(x_test)
      accuracy_score(y_test,y_pred)
[10]: 0.9736842105263158
```

Fig 7: Accuracy Score

**9. Using Pickle to Save the Model**

The 'pickle' module can be used to store the trained model for later use. By doing this, we can prevent having to retrain the model each time we need to make predictions. The binary format in which the model is saved allows it to be loaded and utilized at a later time without requiring retraining.

- ii). **Data Collection:** Use 'pd.read\_csv()' to read the dataset.
- iii). **Total of Columns That Are Null or Not Null:** Determine which values are missing.
- iv). **Eliminating Superfluous Information:** Eliminate superfluous elements such as 'ID'.
- v). **Shape Extraction:** Recognize the quantity of features and samples.
- vi). **Train-Test Split:** Separate the data into sets for testing and training.
- vii). **Model Training:** To train the model, use a Random Forest classifier.
- viii). **Accuracy Score:** Assess how well the model works with unknown data.
- ix). Save the model for later use with the Pickle Module.

```
[11]: import pickle
[14]: pickle.dump(knn,open('bcp.pkl','wb'))
[15]: loaded_model=pickle.load(open('bcp.pkl','rb'))
```

Fig 8: Loading the data in the model

**An Overview of the Steps**

- i). **Importing Libraries:** Important libraries are imported, including sklearn, pandas, and numpy.

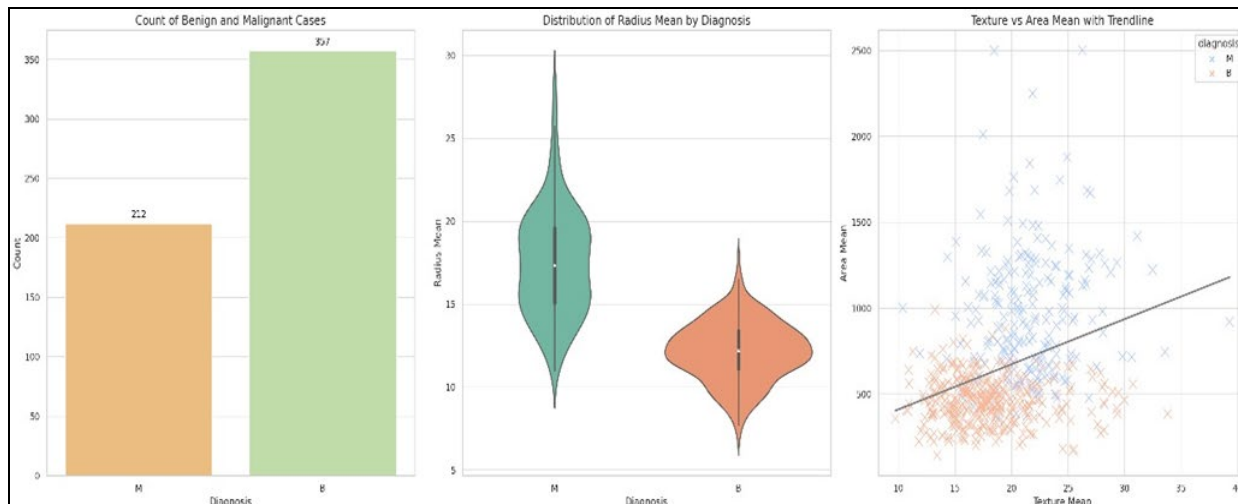


Fig 9: Graphical representation of data

**Graph:**

- i). Representation of Count of Benign and Malignant Cases
- ii). Representation of Distribution of Radius Mean by Diagnosis
- iii). Texture Vs Area Mean with Trendline

**Conclusion**

This study demonstrates the usefulness of the KNN algorithm as a dependable machine learning classification technique and emphasizes its important role in illness risk prediction. By comparing multiple KNN variations across diverse datasets, we built a comparative framework based on performance parameters such as accuracy, precision, and recall.

Our findings indicate that the Hassanat KNN variant achieved the highest average accuracy of 83.62%, closely followed by the ensemble method KNN at 82.34%.

In our improvement work focused specifically on breast cancer prediction, we achieved an impressive precision rate of 97%, showcasing the potential of KNN to deliver highly accurate and reliable predictions in critical healthcare applications. This study underscores the ability of KNN to

enhance predictive analytics in healthcare, providing a foundational basis for future research that may further refine its application in disease prediction.

The implementation of a relative performance index also offers a novel approach for evaluating machine learning models, ensuring a more comprehensive understanding of their strengths and weaknesses. As we continue to explore the intersection of artificial intelligence and healthcare, KNN remains a promising tool for improving decision-making processes in disease management, paving the way for smarter, more efficient healthcare solutions.

**Future Scope**

Based on the findings and insights presented in this paper, several avenues for future research can be explored:

- i). **Enhanced Algorithm Variants:** Building upon the success of the Hassanat KNN and ensemble methods, researchers could develop new kNN variants that incorporate additional features, such as weighted distances or adaptive neighbors based on local data characteristics, to further improve accuracy and

robustness in disease risk prediction.

- ii). **Integration with Deep Learning:** Investigating hybrid approaches that combine kNN with deep learning techniques could enhance performance. For instance, utilizing deep feature extraction methods to preprocess data before applying kNN may yield superior results in complex datasets.
- iii). **Real-Time Implementation:** Future studies could focus on the real-time application of kNN algorithms in clinical settings, evaluating their effectiveness and efficiency in live environments, which would provide insights into practical challenges and solutions in disease risk prediction.
- iv). **Cross-Domain Applications:** Expanding the application of kNN beyond health-related datasets to other critical domains (e.g., finance, agriculture, and environmental monitoring) could validate its versatility and adaptability in various contexts, providing a broader understanding of its capabilities.
- v). **Data Privacy and Ethics:** As ML applications in healthcare raise concerns about data privacy, future research could explore secure and ethical data-sharing frameworks that facilitate the use of kNN while protecting patient information.
- vi). **Automated Hyperparameter Tuning:** Investigating automated methods for hyperparameter tuning specific to kNN variants could optimize model performance and reduce the time and expertise required for effective implementation.
- vii). **Longitudinal Studies:** Conducting longitudinal studies that track the predictive performance of kNN models over time in dynamic healthcare environments could provide insights into model stability and reliability.
- viii). **Comparative Studies with Other Algorithms:** Future research could perform extensive comparative analyses between kNN and other state-of-the-art machine learning algorithms, such as support vector machines or random forests, to better understand their strengths and weaknesses in disease risk prediction tasks.
- ix). **User-Friendly Tools and Frameworks:** Developing user-friendly tools or frameworks that enable healthcare professionals to apply kNN and its variants without extensive ML expertise could enhance the practical utility of the findings.
- x). **Incorporating Multi-Modal Data:** Exploring the integration of multi-modal data sources (e.g., genetic, clinical, and lifestyle data) into kNN models could lead to more comprehensive disease risk assessments and personalized healthcare strategies.

By pursuing these research directions, future studies can significantly enhance the understanding and application of the kNN algorithm in disease risk prediction and other fields, fostering innovation and improved outcomes.

## References

1. Sarker IH. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Comput. Sci. 2021, 2, 160.
2. H. Wang. Nearest Neighbours without k: A Classification Formalism based on
3. Probability, technical report, Faculty of Informatics, University of Ulster, N.Ireland, UK (2002).
4. Uddin S, Khan A, Hossain ME & Moni MA. Comparing different supervised machine learning algorithms for

- disease prediction. *BMC Med. Inform. Decis. Mak.* 2019; 19:1–16.
5. Bzdok D, Krzywinski M & Altman N. Machine learning: supervised methods. *Nat. Methods.* 2018; 15:5–6.
6. Mahesh, B. Machine learning algorithms—a review. *Int. J. Sci. Res.* 2020; 9:381–386.
7. Zhang S, Li X, Zong M, Zhu X & Cheng D. Learning k for kNN classification. *ACM Trans. Intell. Syst. Technol.* 2017; 8:1–19.
8. Bhatia N & Vandana. Survey of nearest neighbor techniques. *Int. J. Comput. Sci. Inf. Secur.* 2010; 8:1–4.
9. Jiang Y, Li X, Luo H, Yin S, Kaynak O. Quo Vadis Artificial Intelligence? *Discov. Artif. Intell.* 2022; 2:629–869. [CrossRef]
10. Janiesch C, Zschech P, Heinrich K. Machine Learning and Deep Learning. *Electron. Mark.* 2021; 31:685–695. [CrossRef]
11. Sarker IH. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Comput. Sci. 2021, 2, 160.[CrossRef] [PubMed]
12. Han J. Pei, J, Tong, H. Data Mining: Concepts and Techniques, 4th ed, Morgan Kaufmann: Burlington, MA, USA, 2022.
13. Ahmad SR, Bakar AA, Yaakub MR, Yusop NMM. Statistical validation of ACO-KNN algorithm for sentiment analysis. *J.Telecommun. Electron. Comput. Eng.* 2017; 9:165–170.
14. Cover T, Hart TP. Nearest neighbor pattern classification [J]. *IEEE.* 1967; (1):21-27.
15. Cover T. Rates of convergence for nearest neighbor procedures [J]. *Systems Sciences*, 1968.
16. Stone CJ. Consistent Nonparametric Regression [J]. *Institute of Mathematical Statistics*, 1977; (7)5, (4):595-620.
17. Cleveland W S. Robust locally weighted regression and smoothing scatterplots [J]. *Journal of the American Statistical Association*, 1979, 74:829-836.
18. Brown T, Kopolowitz, Jack. The weighted nearest neighbor rule for class dependent sample sizes (Corresp.) [J]. *IEEE*, 1979; (9).IT-25: 617-619.
19. D Hand, H Mannila, P. Smyth.: Principles of Data Mining. The MIT Press, 2001.
20. H Wang. Nearest Neighbours without k: A Classification Formalism based on Probability, technical report, Faculty of Informatics, University of Ulster, N. Ireland, UK, 2002.
21. F Sebastiani. Machine Learning in Automated Text Categorization. In *ACM Computing Surveys.* 2002; 34(1):1-47.
22. H Wang, I Dutsch, D Bell. Data Reduction Based on Hyper Relations. In *proceedings of KDD98*, New York, 1998, 349-353.
23. P Hart. The Condensed Nearest Neighbour Rule, *IEEE Transactions on Information Theory.* 1968; 14:515-516.