Exploring the Diversity in K-Nearest Neighbors :A Comparative Study of Different KNN Variants

Subhra Jyoti Mishra [1]

[1]National Institute of Science Education and Research

Introduction

The K-Nearest Neighbors (KNN) algorithm is a simple yet powerful tool in machine learning. It belongs to the family of supervised learning algorithms and is widely used for classification and regression tasks. Its applications span various domains, including pattern recognition, data mining, and intrusion detection.

At its core, KNN operates on the principle that similar things exist nearby. In other words, it assumes that data points with similar attributes will likely share the same label. This makes KNN particularly useful when the relationship between the data attributes and the output is complex or unknown. KNN is intuitive and easy to implement, making it a go-to algorithm for many machine learning practitioners. Its versatility allows it to adapt to different types of data and problem settings, so it remains a popular choice despite the emergence of more complex algorithms.

How KNN works ?

The K-Nearest Neighbors algorithm operates on a very straightforward principle: it classifies a new data point based on the majority class of its closest neighbors.[6] Here's a step-by-step breakdown of the process:

1. **Identify the Nearest Neighbors**: Setting the value for k
2. **Distance Calculation**: KNN calculates the distance between the new data point and each of the neighbors using a distance metric, such as Euclidean distance. The formula for Euclidean distance in a two-dimensional space is
3. **Classify the New Point**: Once the distances are calculated, the algorithm sorts the points by increasing distance and selects the top 'k' points.

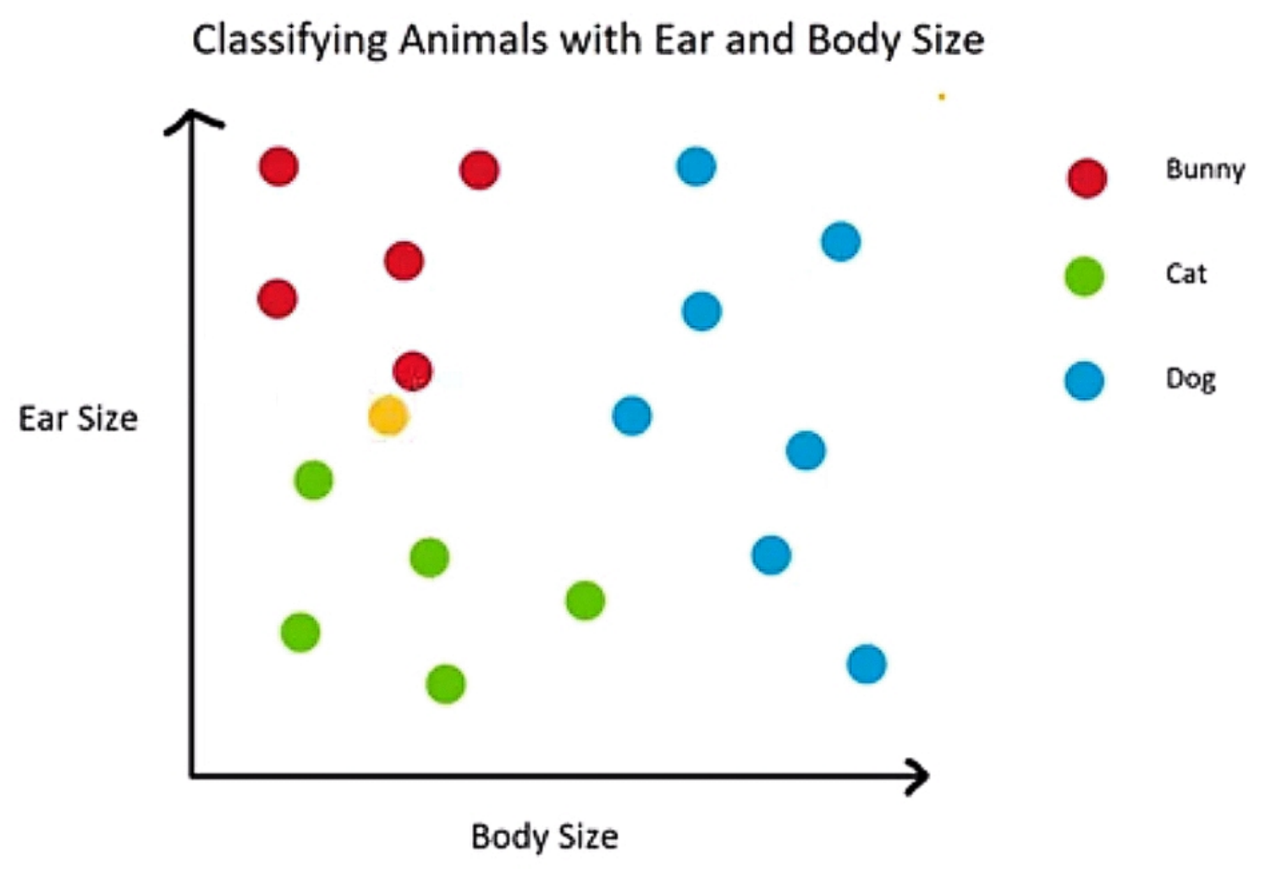

Figure 1: Identifying an unknown organism

Types of KNN

Classic KNN:

This is the standard version of KNN that relies on distance metrics like Euclidean or Manhattan distance to find the closest neighbors.

Adaptive KNN :

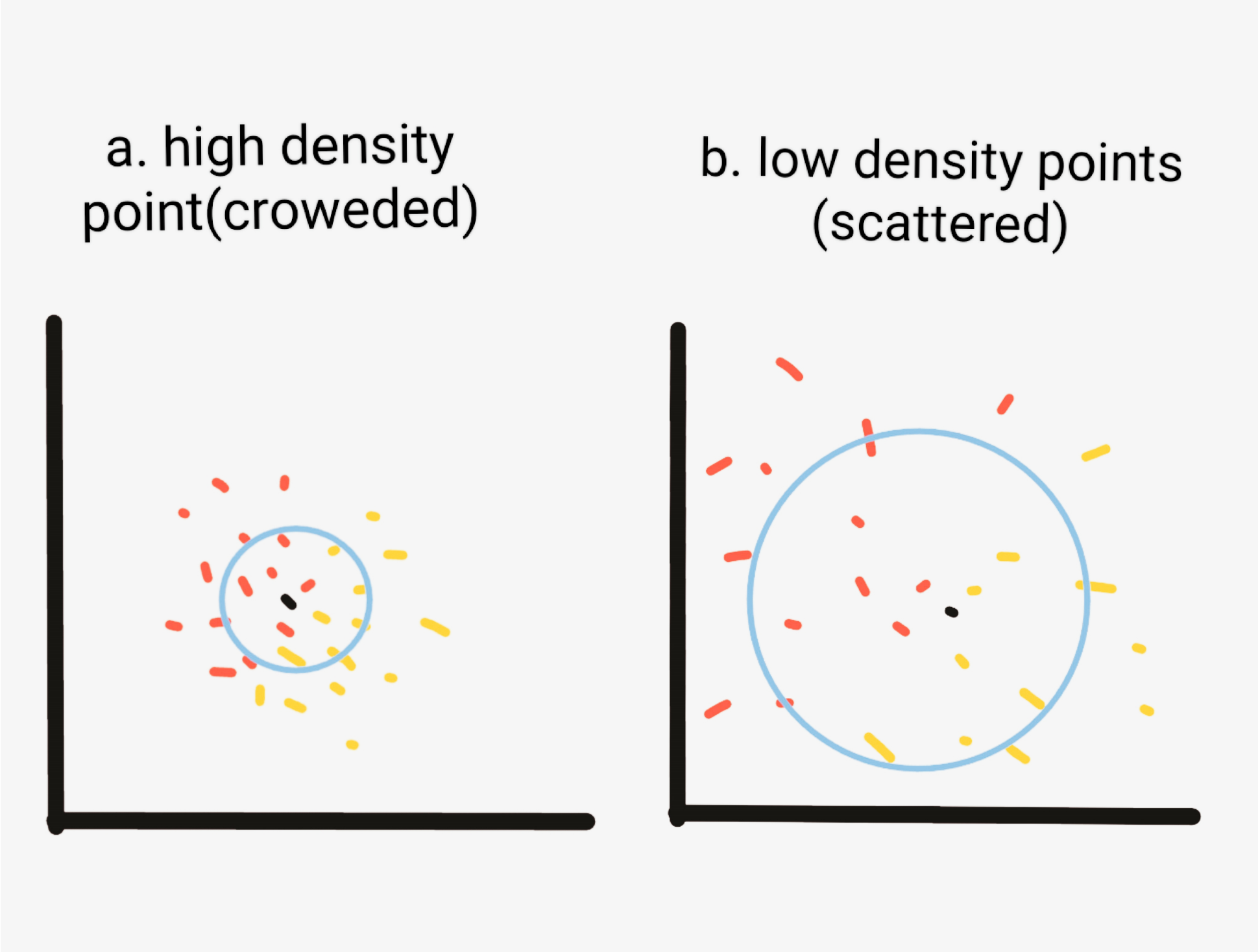
a. high density point(croweded)
b. low density points (scattered)

Figure 2: working of A-KNN

Adaptive KNN is a variant that dynamically adjusts the number of neighbors (k) based on the local density of points within the dataset. This adaptive approach enables the algorithm to handle better situations where data points are not uniformly distributed across the feature space. If it's crowded, it might look at fewer neighbors; if it's sparse, it might look at more.[1]

K-Means Clustering KNN:

K-Means Clustering KNN is an innovative technique that merges the K-Nearest Neighbors (KNN) algorithm with K-means clustering to enhance the classification of data points. K-means clustering groups similar data points into clusters based on their features, creating a more structured dataset representation.

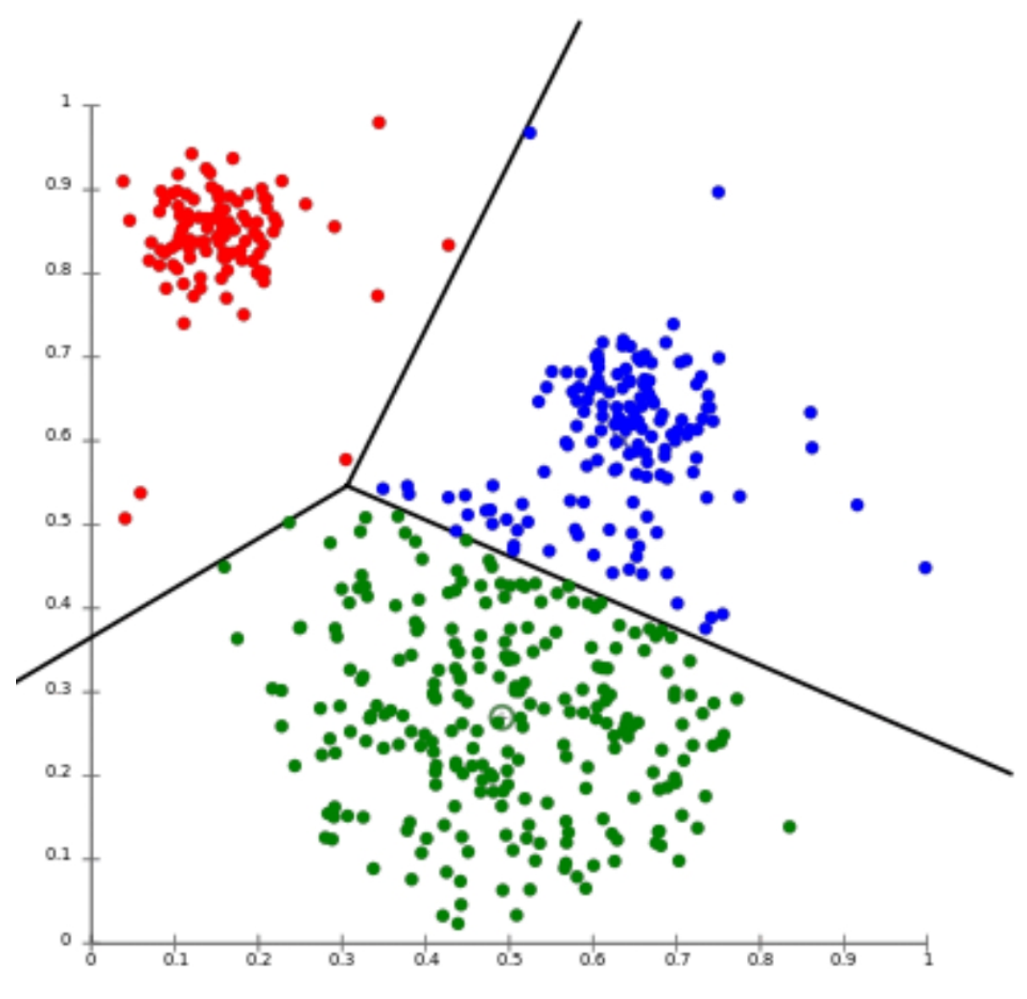

Figure 3: Division in cluster for application of KNN

Fuzzy KNN:

Fuzzy KNN is like KNN with a twist. Instead of making strict decisions based on the nearest neighbors, it considers how much each neighbor belongs to each class. It assigns a "fuzzy" membership value to each neighbor for each class, indicating the degree to which that neighbor belongs to that class.So, when you want to classify a new data point, instead of just counting which class has the most neighbors, Fuzzy KNN looks at the membership values of all neighbors for each class. Then, it takes a weighted average to decide the class of the new point.

Mutual KNN:

Mutual KNN is a method that improves the performance of the K-Nearest Neighbors (KNN) algorithm by incorporating feature selection based on mutual information. It eliminates the noisy instances by a concept called mutual nearest neighbor.

Ensemble KNN:

Multiple KNN models are combined to make predictions, which can lead to improved accuracy and robustness. Ensemble KNN trains multiple KNN models on the dataset, each with different settings or subsets of data. [5]

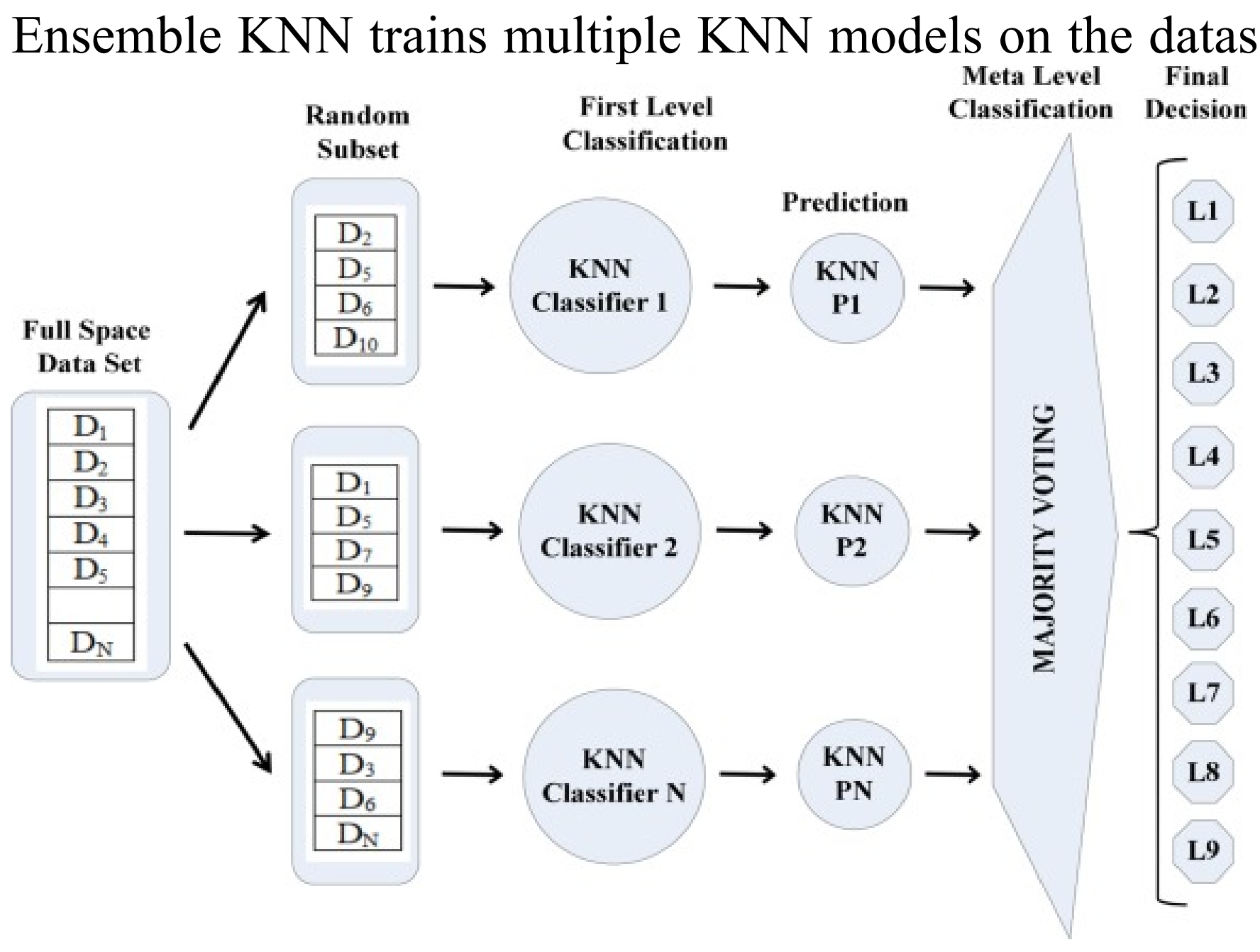

Figure 4: KNN based random ensemble classifier

Hassanat KNN:

A modified version of KNN designed to improve performance by altering the distance metric calculation.The Hassanat variant modifies the distance calculation between data points. It aims to improve the performance of the KNN algorithm, particularly in terms of accuracy. This variant was analyzed in a comparative study alongside other KNN variants. It showed promising results, especially for disease prediction tasks

Generalized Mean Distance KNN:

This variant uses the generalized mean for distance calculation, allowing for a more flexible definition of 'closeness' between points. The formula for the generalized mean is:

$d(x, y) = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i|^{p})^{\frac{1}{p}}$

(p) is the parameter that determines the degree of "averageness." Selecting this p value allows for customized distance calculations based on the specific characteristics and relationships within the dataset allowing more flexibility

Comparative Performance Analysis

The study's findings revealed that the **Hassanat KNN** variant achieved the highest average accuracy **(83.62%)**, followed closely by the **Ensemble KNN (82.34%)**. The **classic KNN** comes with an average accuracy of **64.22%**. Other KNNs are data-specific and range from 64.22 to 83.34%. These results were based on performance measures such as accuracy, precision, and recall. The study also proposed a relative performance index to assess each variant comprehensively[4].

Specific use cases of KNN in disease prediction include heart disease prediction. KNN's ability to classify data based on the closest training data points makes it a valuable tool for identifying potential health risks. Other chronic diseases like diabetes, breast cancer, and kidney ailments have also been predicted using KNN variants, demonstrating the algorithm's versatility and effectiveness in the healthcare domain. These comparative analyses underscore the importance of selecting the appropriate KNN variant for predictive analytics in healthcare, ensuring that stakeholders can leverage the most accurate and reliable methods for disease risk prediction.

References

[1] GeeksforGeeks.
K-nearest neighbor(knn) algorithm.
`https://www.geeksforgeeks.org/k-nearest-neighbours/`, 2021.
Accessed: 2024-04-04.

[2] Sarang Anil Gokte.
Most popular distance metrics used in knn and when to use them.
`https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html`, 2020.
Accessed: 2024-04-04.

[3] Varun Jain.
Introduction to knn algorithms.
`https://www.analyticsvidhya.com/blog/2022/01/introduction-to-knn-algorithms/`, 2022.
Accessed: 2024-04-04.

[4] Haque I. Lu H. et al. Uddin, S.
Comparative performance analysis of k-nearest neighbour (knn)algorithm and its different variants for disease prediction.
Sci Rep 12, 6256, (2022).

[5] Hua Wang, Hua-can He, and Sheng Li.
An adaptive k-nearest neighbor algorithm.
In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, pages 305–309. IEEE, 2010.

[6] Zhongheng Zhang.
Introduction to machine learning: K-nearest neighbors.
Annals of Translational Medicine, 4(11):218, 2016.

subhrajyoti.mishra@niser.ac.in

CS460 Machine Learning

+91 8327762910