

Mine Automation and Data Analytics

Prof. Radhakanta Koner

Department of Mining Engineering

IIT (ISM) Dhanbad

Week - 10

Lecture 48: K Nearest Neighbor

Music Welcome back to my course, automation and data analytics. Today, we will discuss the known k nearest neighbor method, another supervised machine learning method used for regression and classification. So, in this lesson, we will discuss the following. First, we will introduce the kind k nearest neighbor method and its base; that is, the distance metric based on the distance matrix classifies the data points new data points into a suitable class. Then, we will discuss the assumptions behind this kind of approach and the advantages and disadvantages of the known approach. Finally, we will discuss the potential applications of what is known in the mining industry, such as what is known as nearest neighbor, which is supervised machine learning. I already said I use it for classification and regression tasks, so let us see how it functions. So, you can see that the green data points belong to some data category, and the yellow data set represents another data category.

K-Nearest Neighbors (KNN)

k-Nearest Neighbors (k-NN) is a supervised machine learning algorithm used for both classification and regression tasks.

It is a simple, yet effective, algorithm that makes predictions based on the majority class (for classification) or the average (for regression) of the k -nearest data points in the feature space.

The diagram illustrates the K-NN process. On the left, 'Before K-NN', a 2D plot with axes x_1 and x_2 shows green points (Category A), yellow points (Category B), and a new blue data point. An arrow labeled 'K-NN' points to the right, 'After K-NN', where the blue data point is now assigned to Category A.

Okay. Now I have a new blue data point, so this blue point must belong to category B or category A. How do I reach a decision? So one of the better ways of estimating the appropriate class of this blue point is who the neighbor surrounds or around the blue points and which neighbors are closer compared to these different data points belonging to category A and category B so if it is found that this blue point is very close in terms of the distance from the

category between this two category very close to category A k, then the known approach does the thing that this blue point is now categorized as category A. okay so it's a very effective algorithm for predicting a new data point this data point belongs to each class okay or averaging purpose for new feature points in the feature space is.

So it basically relies on the distance so it calculates this and these points these points these points these points so if k is equal to 5 or k is equal to 6 6 number of points it will select nearly 6 near never 6 points 5 points 7 points 9 points so we generally avoid even number of points okay so it should be 5 7 9 11 so and so forth so basically by measuring the distance it basically tried to conclude that the distance between these feature classes category A and with a category B these new points distance is significantly less or less compared to the distance calculated of these points so blue points belongs to the category A so it has a advantage that you are basically we are basically calculating the Euclidean distance so this kind of classification is good when the data is two-dimensional data at max three-dimensional data see the dimensionality of the data or number of features are increased then these distance calculation might be a complex problem for this kind of algorithm so for a low data or low dimensional data.

It's a very efficient method for classifying, with a very high confidence level, which category this data point belongs to. Hence, this is the primary advantage of the k nearest neighbor approach, which is okay for classifying the data points based on the Euclidean distance. So, it's a mighty supervised machine learning algorithm that can be used for the classification and regression task. It is often called a lazy learning algorithm because there is no learning in that sense okay. It is non-parametric and lazy learning, making no assumptions about the underlying data distribution. It does not learn explicitly during the training phase, so the fundamental assumption is that it does not assume any data distribution.

Okay, so the KNN works based on the principle of similarity or distance similarity or affinity close affinity with the neighbor data class or the distance close distance between the points, so relying on that, the KNN approach concludes or classifies suitable classes. Hence, it classifies data points based on how their neighbors are classified, so K in KNN represents the number of nearest neighbors considered when making predictions. Therefore, it is always good if the critical value is significant because it will basically, if there is the presence of some number of outliers it will, offset okay if the number K is high, so it will calculate the distance between a high number of points an okay large number of points so that offset if some outlier is there it will accurately predict that which are the classes more closely located from these new data points. So, this particular method is always better. The case would be with an in the higher range. Okay, so what are the mathematics behind the KNN approach? It primarily involves the calculation of the distance between the data points and the method used for making predictions. So, let's start with the

K-Nearest Neighbors (KNN) Distance Metrics

The mathematics behind the K-Nearest Neighbors (KNN) algorithm involves primarily the calculation of distances between data points and the method used for making predictions. Let's delve into the mathematical details:

Distance Calculation:

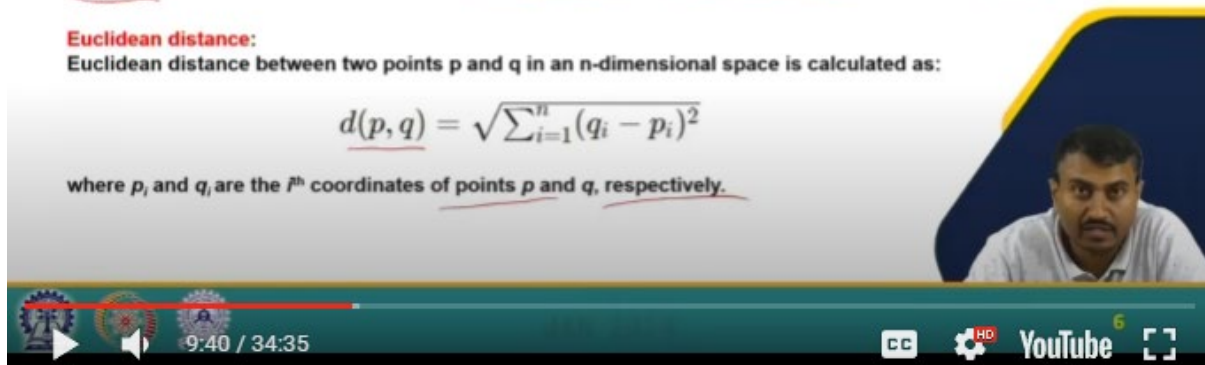
KNN uses a distance metric to measure the similarity or dissimilarity between data points. The most commonly used distance metrics are Euclidean distance, Manhattan distance, and Minkowski distance.

Euclidean distance:

Euclidean distance between two points p and q in an n -dimensional space is calculated as:

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

where p_i and q_i are the i^{th} coordinates of points p and q , respectively.



The Euclidean distance, the PQ between the two data points, P and Q in the in-dimensional space, is calculated as root over summation i is equal to 1 to n Q_i minus P_i whole square, so P_i and Q_i is the i th coordinate point of P and Q respectively okay so basically we measure the distance metric based on the similarity or dissimilarity between the data points so most commonly used metric are Euclidean distance, Manhattan distance, and Minkowski distance okay so let us see what are these distances this is Euclidean distance d_{PQ} is equal to root over summing over i is equal to 1 to n in Q_i minus P_i whole square.

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

so what is Manhattan distance? Manhattan distance, also known as city block distance or L1 distance between two points P and Q, is calculated as the sum of the absolute difference of their coordinates absolute difference of their coordinates. So d_{PQ} equals Q_i minus P_i mod summing over i equals 1 to n .

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

K-Nearest Neighbors (KNN) Distance Metrics

Manhattan distance:

Manhattan distance, also known as City Block distance or L1 distance, between two points p and q is calculated as the sum of the absolute differences of their coordinates:

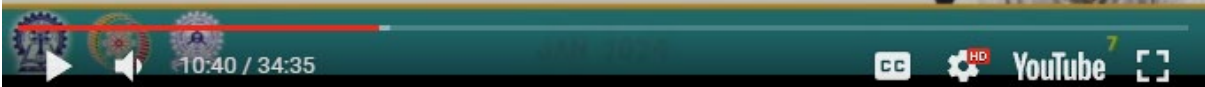
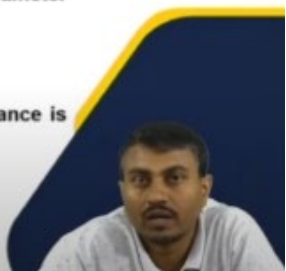
$$d(p, q) = \sum_{i=1}^n |q_i - p_i|$$

Minkowski distance:

Minkowski distance is a generalization of both Euclidean and Manhattan distances. For parameter r , the Minkowski distance between two points p and q is calculated as:

$$d(p, q) = \left(\sum_{i=1}^n |q_i - p_i|^r \right)^{\frac{1}{r}}$$

Euclidean distance is a special case of Minkowski distance when $r=2$, and Manhattan distance is the case when $r=1$.



$$d(p, q) = \left(\sum_{i=1}^n |q_i - p_i|^r \right)^{\frac{1}{r}}$$

The Minkowski distance is a generalization of both Euclidean and Manhattan distance for parameter R the Minkowski distance between two points P Q is calculated as this the P Q is equal to Q minus P I mod to the power R summing over i is equal to 1 to n and whole is to the power one minus R .so in case of R is equal to 2 it will become the Euclidean distance in case of R is equal to 1 it will become the Manhattan distance. Okay, so these are, in that case, similar approaches. Okay, a conversion can be quickly done based on the value of R , so K is the number of neighbors to consider. It is essential to choose an appropriate value of K , and this algorithm must perform efficiently so the distance metric is used to compute the distance between the data points, such as Euclidean distance, Manhattan distance, Minkowski distance, and the distance metrics.

K-Nearest Neighbors (KNN)

1. Training Phase:

1. The algorithm stores the entire training dataset in memory.

2. Prediction Phase:

1. Given a new, unseen data point, the algorithm identifies the k-nearest neighbors to that point in the feature space.
2. For classification, the majority class among the k-neighbors is assigned to the new data point.
3. For regression, the average (or weighted average) of the target values of the k-neighbors is assigned to the new data point.

3. Distance Metric:

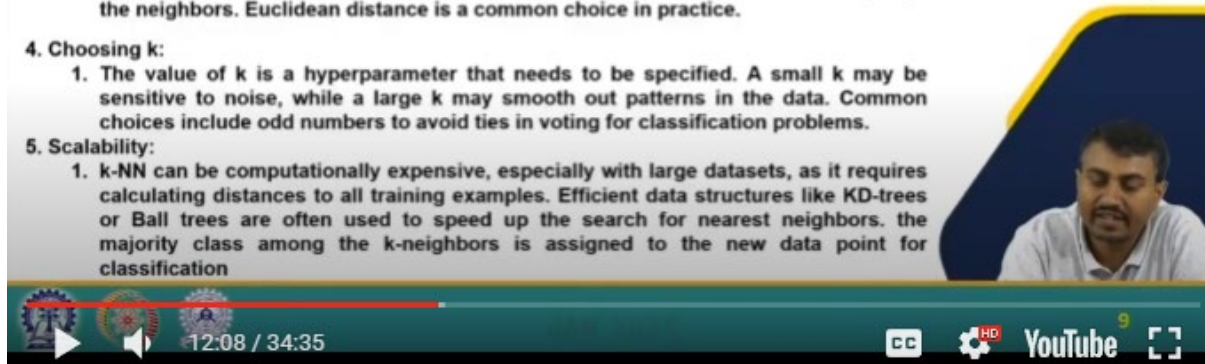
1. The choice of distance metric (such as Euclidean distance, Manhattan distance, etc.) is crucial in determining the neighbors. Euclidean distance is a common choice in practice.

4. Choosing k:

1. The value of k is a hyperparameter that needs to be specified. A small k may be sensitive to noise, while a large k may smooth out patterns in the data. Common choices include odd numbers to avoid ties in voting for classification problems.

5. Scalability:

1. k-NN can be computationally expensive, especially with large datasets, as it requires calculating distances to all training examples. Efficient data structures like KD-trees or Ball trees are often used to speed up the search for nearest neighbors. The majority class among the k-neighbors is assigned to the new data point for classification.



So, how does it work? First is the training phase, where the algorithm stores the entire training data set in memory, so it's a memory-intensive method. Okay, the second is the prediction phase, so let's take new data points. Okay, Using this algorithm, the K is the nearest neighbor to that point in the feature space. Okay, so for classification, the majority class among the K neighbors is assigned to the new data points for regression. The average or weight average of the target value of the K near neighbor is transferred to the latest data points. The distance metric, the choice of distance metrics such as Euclidean distance, Manhattan distance, or Minkowski distance, is crucial in determining whether the neighbors are okay, and most commonly, we use the Euclidean distance in two-dimensional space.

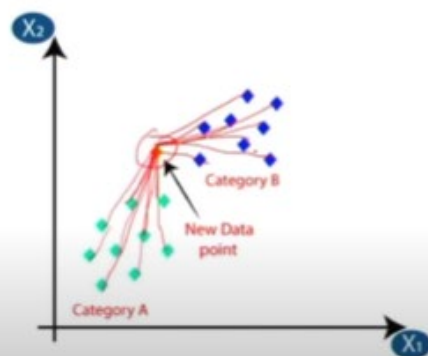
Choosing K, the value K is a hyperparameter that needs to be specified before starting this process so that a small K may be sensitive to noise. At the same time, large K may smooth out patterns in the data, so common choices include odd numbers to avoid ties in voting for classification problems. Hence, it is always better to go for odd numbers 7, 9, 11, 15, and 17. That scalability of the KNN can be computationally expensive, especially with large data sets, because it is stored in the data in the memory as it requires calculating the distance to all training and samples, so efficient data structures like kd3 or ball trees are often used to speed up the search for nearest neighbor. The majority class among the K neighbors is assigned to the new data points for classification.

So, how does KNN work? So, let us discuss the steps that are followed in KNN. Step one determines the value of K, representing the number of neighbors to consider. The second stage is to compute the distance metric, preferably Euclidean distance, for the K neighbors. Then, the third identifies the K nearest neighbor based on the calculated Euclidean distance. The fourth is to tally the occurrences of data points in each category among these K neighbors and allocate the new data points to the category with the highest neighbor count. Okay, the model is now prepared for stage 6. so let us start with a problem. So we have taken new data points

that are yellow color points here, okay? We have two categories of data, which are green data points. Category B is another set of data, so if the number is high, for example, I take the number high, and number K is very high, so I calculate all these distances. Okay, so as I said, tell you the occurrences of the data points in each category among these K neighbors. Align new data points to the category with the highest neighbor count. Okay, so once this distance calculation is done, we must suitably assign a new category to the latest data points. So here we have selected K as equal to 5, so 3 points from category A and 2 points from category B are okay. We will compute the Euclidean distance between the data points, which is the distance between these data points, and the Euclidean distance represents the distance between two points. We know $\sum_{i=1}^K (Q_i - P_i)^2 = 1$ to 5 okay because K is five, so upon calculating the Euclidean distance, we identify the nearest neighbor with three closest neighbors belonging to category A and two closets near belonging to category B which count is high? It is category A.

K-Nearest Neighbors (KNN)

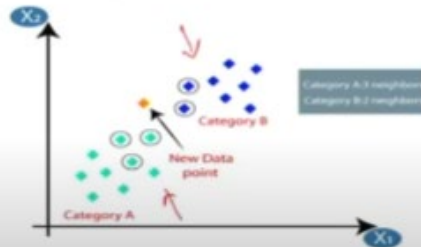
Imagine we have a new data point and we need to assign it to the appropriate category. Please refer to the image below:



K-Nearest Neighbors (KNN)

Initially, we'll select the number of neighbors, opting for $k=5$.

Following that, we'll compute the Euclidean distance between the data points. The Euclidean distance represents the distance between two points. Upon calculating the Euclidean distance, we identify the nearest neighbors, with three closest neighbors belonging to category A and two closest neighbors belonging to category B. Please refer to the image below for further clarification:



Observing that the three closest neighbors belong to category A, it is evident that this new data point should be categorized as belonging to category A.

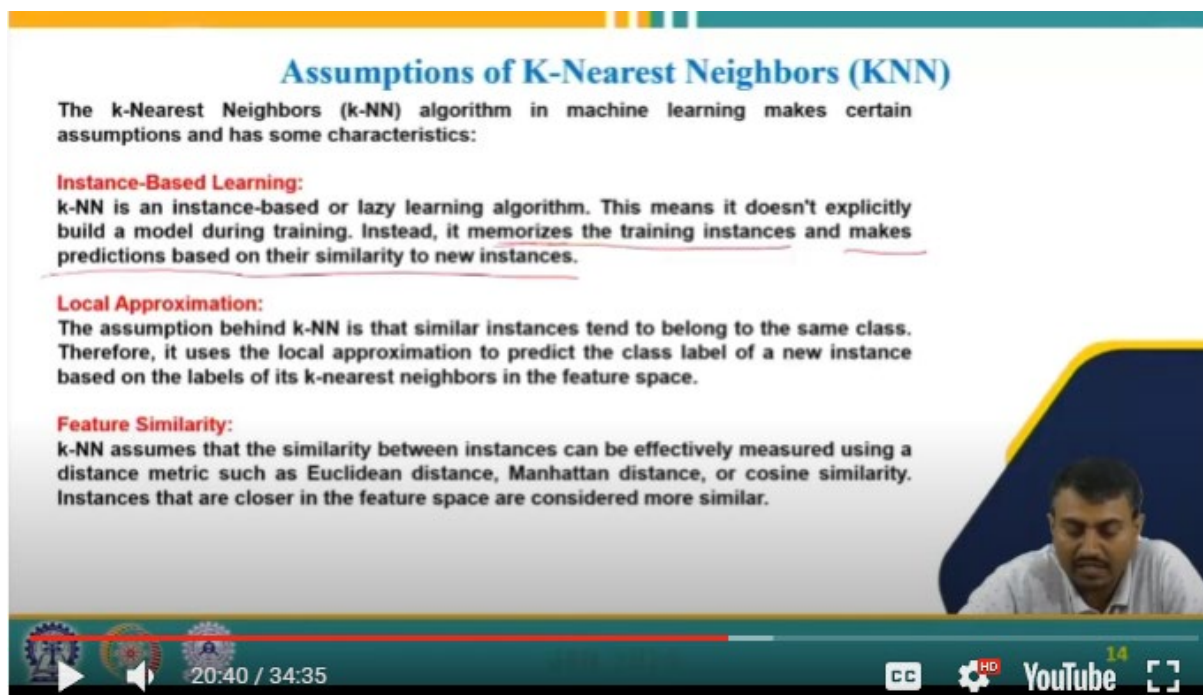
So please see these particular examples. So these yellow by observing these three neighbors are closer to several neighbors closer from category B observing the three nearest neighbors belonging to category A; these new data points should be categories belonging to category A, so that is why we are looking at odd numbers it is four this is three this is seven this is 4 11 this is seven it is always because in case of even numbers so 3 3 both are close. So, it would be challenging to assign which class it belongs to.

So, that is why one crucial assumption we have made is that it is always better that you assume K as an odd value. Hence, the KNN is a nonparametric algorithm, meaning it does not make strong assumptions about the underlying data distribution. One advantage is that it does not assume data distribution to adapt to complex decision boundaries. It is beneficial when a simple mathematical model needs to quickly characterize the relationship between the input features and the target variable. So, KNN is straightforward to understand, but it might need to perform better in high-dimensional spaces with data sets or irrelevant features present. I have already told you that when the dimension is very high, KNN will perform, as it is always better to use KNN with low dimensional data points or data sets.

Additionally, the choice of the distance metric and the value K can significantly impact the algorithm's performance. In practice, we have the scikit-learn package in Python and some libraries in other suitable software that we can use to perform those codes.

So, what are the assumptions in KNN machine learning? First is instant base learning, so KNN is an instant or lazy learning algorithm. It does not explicitly build a model during training; instead, it memorizes the training instances and makes predictions based on the similarity to

new distances. Second local approximation, the assumption behind KNN is that similar cases tend to belong to the same class. Therefore, it uses a local approximation to predict the class level of a new instance based on the level of its K nearest neighbor in the feature space. Feature similarity KNN assumes that the similarity between the cases can be effectively measured using a distance metric such as Euclidean distance, Manhattan distance, means a distance, or cosine similarity instances closer in the feature space are considered more similar. K parameter selection: the performance of KNN depends on the, as I said already, the distance metric as well as the K , the choice of the parameter K so it represents the number of nearest neighbors to consider a smaller value K may result in more complex decision boundaries potentially leading to overfitting so a more significant value of K may result in smoother decision boundaries but might miss local pattern.



Assumptions of K-Nearest Neighbors (KNN)

The k-Nearest Neighbors (k-NN) algorithm in machine learning makes certain assumptions and has some characteristics:

Instance-Based Learning:
k-NN is an instance-based or lazy learning algorithm. This means it doesn't explicitly build a model during training. Instead, it memorizes the training instances and makes predictions based on their similarity to new instances.

Local Approximation:
The assumption behind k-NN is that similar instances tend to belong to the same class. Therefore, it uses the local approximation to predict the class label of a new instance based on the labels of its k-nearest neighbors in the feature space.

Feature Similarity:
k-NN assumes that the similarity between instances can be effectively measured using a distance metric such as Euclidean distance, Manhattan distance, or cosine similarity. Instances that are closer in the feature space are considered more similar.

Noisy data handling: KNN is sensitive to noisy data and outliers because it considers all training instances equally when making predictions, so noisy data can significantly affect the classification results and may require pre-processing or noise reduction techniques. Computational efficiency While KNN is conceptually simple, it can be computationally expensive, especially for extensive data sets for large data sets, because it requires computing distance between the new instances and all training instances. Efficient data structures such as kd or ball trees are often used to speed up the search for nearest neighbors. So, understanding these assumptions and characteristics is crucial for effectively applying the KNN approach and interpreting its result in various machine learning tasks. So, what are the advantages of KNN? The primary benefit is the simplicity of the KNN, which is easy to understand and implement, making it a great starting point for beginners in machine learning. Second is no training phase. KNN is a lazy learning algorithm, meaning it does not require a training phase, and the model directly uses the training data for prediction, making it efficient for incremental learning scenarios or new data points that can be added without retaining retraining the model.

Advantages of K-Nearest Neighbors (KNN)

Flexibility:

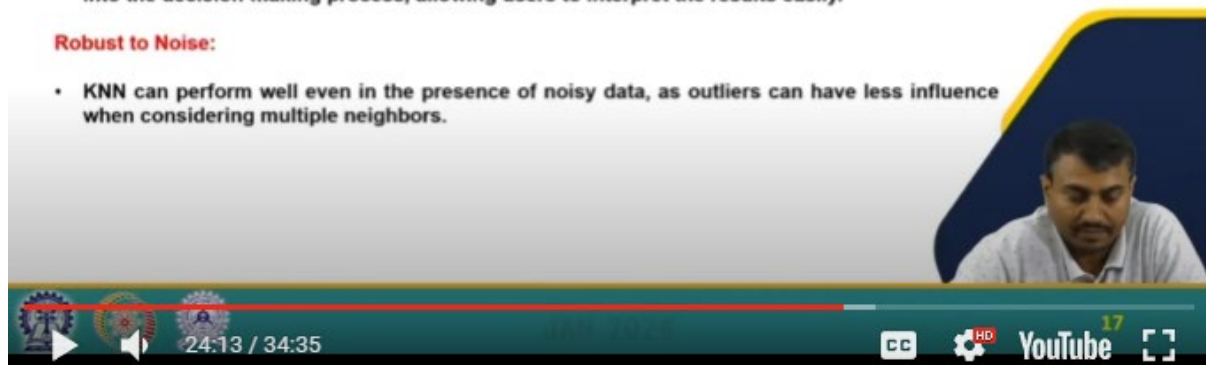
- KNN can be used for both classification and regression tasks, making it versatile.

Interpretability:

- Since predictions are based on nearby points in the feature space, KNN can provide insights into the decision-making process, allowing users to interpret the results easily.

Robust to Noise:

- KNN can perform well even in the presence of noisy data, as outliers can have less influence when considering multiple neighbors.



Non-parametric KNN does not assume the underlying data distribution because earlier, you saw that the residual should be generally distributed in the regression. Here, we are not assuming anything. making it suitable for a wide range of applications, including nonlinear data. Flexibility: KNN can be used for classification and regression tasks, making it versatile and interpretable since predictions are based on nearby points in the feature space. KNN can provide insight into the decision-making process, allowing users to interpret the result quickly and robust to noise. KNN can perform well even in noisy data, as outliers can have less influence when considering multiple neighbors.

Disadvantages of K-Nearest Neighbors (KNN)

Computational Complexity:

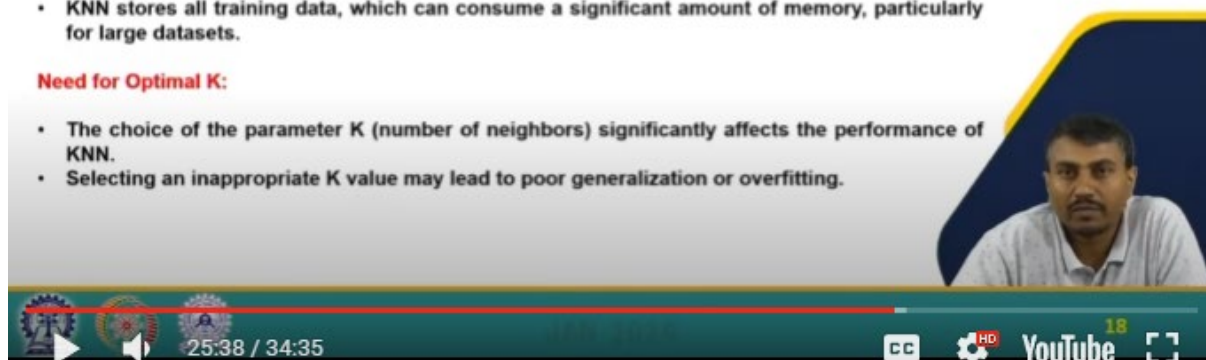
- During prediction, KNN needs to calculate the distances between the query point and all training points, which can be computationally expensive for large datasets, especially in high-dimensional feature spaces.

Memory Intensive:

- KNN stores all training data, which can consume a significant amount of memory, particularly for large datasets.

Need for Optimal K:

- The choice of the parameter K (number of neighbors) significantly affects the performance of KNN.
- Selecting an inappropriate K value may lead to poor generalization or overfitting.



Let us discuss the disadvantages of KNN's computational complexity during prediction. KNN needs to calculate the distances between the query and all training points, which can be

computationally expensive for extensive data sets, especially in high-dimensional feature space. Memory intensive: KNN stores all training data, which can consume a significant amount of memory, particularly for large data sets. Need for optimal K The choice of the parameter K number of neighbors significantly affects the performance of KNN. Selecting an appropriate K value may lead to better generalization or overfitting. It is sensitive to feature scaling since KNN relies on distance metric features with larger scales that may dominate the distance calculation, leading to biased results. Therefore, feature scaling is often necessary for imbalanced data in classification tasks with imbalanced data or imbalance class distribution. KNN may bias predictions towards the majority class, especially when K is small. Curse of dimensionality KNN performance can degrade rapidly as the dimensionality of the feature space increases. I have already interpreted that we prefer using low-dimensional data space in KNN. This phenomenon is known as curse dimensionality, where the volume of the feature space grows exponentially, causing the nearest neighbor to become less meaningful. Understanding these advantages and disadvantages is very important for the learner to apply the KNN to different machine-learning tasks and scenarios. Hence, we must consider this factor and perform proper experimentation and tuning to achieve optimum results.

So, let us discuss some potential applications of the KNN algorithm in the mining industry exploration targeting because exploration is a crucial step in the mining process. The exploration gives us an idea of the reserve's quantity and quality. It was so promising that locations for mineral exploration should be identified based on geological or geochemical data. It uses the KNN to analyze historical exploration data and geological features of known deposits to predict potential new mineralization areas by considering similar geological contexts and proximity to existing deposits. So, here, KNN can help predict whether exploration should be done based on the preliminary level of predictions based on some of the data found in the nearby area.

Rock classification classifies different rock or mineral deposits based on their physical and chemical properties. It is essential to classify the rock sometimes for the underground support design and the explosive amount used for blasting drilling patterns for the rock types. So, there are valid applications in mining to estimate the rock type. So, the KNN approach classified rock samples collected from mining sites based on their spectral signatures, mineral composition, and other characteristics. KNN learned from level training data to accurately classify new rock samples into predefined categories heading in geological mapping and resource estimation.

Applications of K-Nearest Neighbors (KNN) in Mining

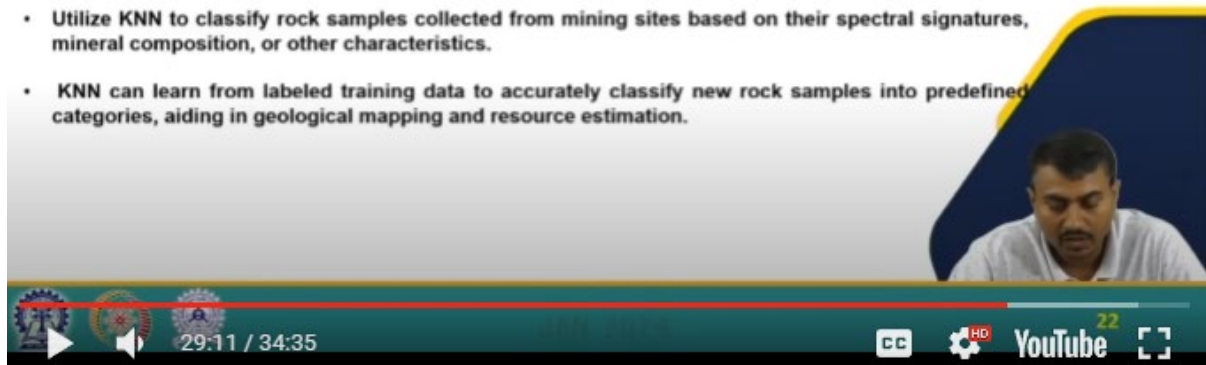
Rock Classification:

Problem:

- Classifying different types of rocks or mineral deposits based on their physical and chemical properties.

Solution:

- Utilize KNN to classify rock samples collected from mining sites based on their spectral signatures, mineral composition, or other characteristics.
- KNN can learn from labeled training data to accurately classify new rock samples into predefined categories, aiding in geological mapping and resource estimation.



Geotechnical stability assessment assessing the stability of the mining structures or mining structures, for example, slope tunnels and underground excavation, to prevent collapse and ensure worker safety. so here, we can employ the KNN to analyze the geotechnical data, for instance, the ucs shear strength of the rock, the intensity or distribution of the joints. Okay, ground conditions. Based on that, we will predict the risk associated with this kind of condition by identifying similarities between these geological conditions and historical data on the stability records. Uh, KNN can help to evaluate the stability condition under newer situations. Whether this condition is susceptible to failure or risky scenarios, we can use KNN convincingly.

Predictive maintenance is a continuous practice in the mining automation industry, mainly because we use many machines. Machine behavior is noted down, and based on that, it is required that we should take care of the maintenance problem at frequent times or regular intervals so that we can utilize the machine for a more significant period and efficiently without much downtime. So, for maintenance management, scheduling maintenance, reducing downtime, and reducing operational costs for this kind of case, KNN might be helpful because these machines are fitted with many sensors. Sensors collect data about the behavior of the machines under different conditions, so considering the historical data and past failure of the machines under certain conditions of the observed parameter, we can use the KNN approach to avoid breakdowns. Proactively, we can suggest maintenance times, frequencies, or intervals. That will save a good amount of money for the mining companies.

Applications of K-Nearest Neighbors (KNN) in Mining

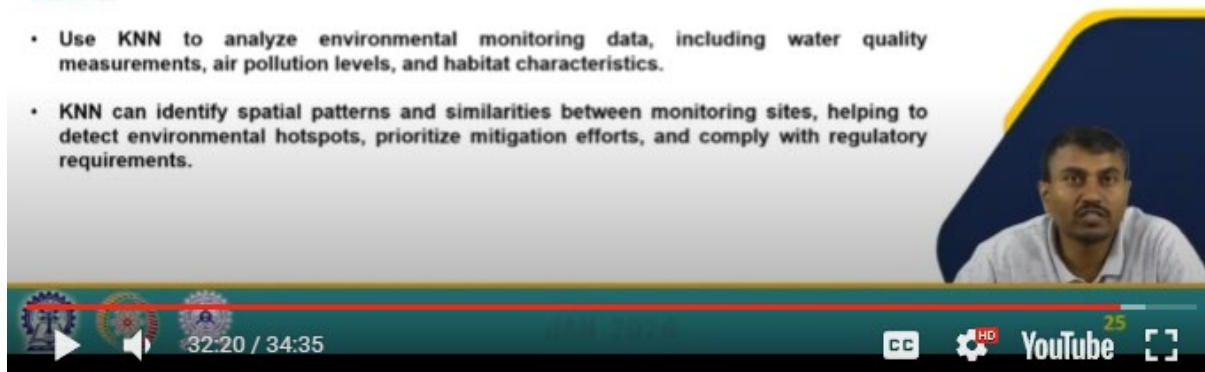
Environmental Monitoring:

Problem:

- Monitoring and managing environmental impacts of mining activities, such as water pollution and habitat destruction.

Solution:

- Use KNN to analyze environmental monitoring data, including water quality measurements, air pollution levels, and habitat characteristics.
- KNN can identify spatial patterns and similarities between monitoring sites, helping to detect environmental hotspots, prioritize mitigation efforts, and comply with regulatory requirements.



Environmental monitoring has a significant impact, and most companies are now closely working with government agencies and regulatory agencies to reduce the impact on the environment by the mining process. Okay, so based on the different data uh collected from the surrounding uh data about the dust data about the water quality data about some of the noxious gases and how these habitation has been disturbed so based on that uh we can predict in a more extended long-term basis for this kind of mining what type of possible damage to be done in the mine in the environment so that we can pre-plant beforehand for the reclamation and some uh actions that be that to be taken care.

This effect is reduced or mitigated in due time so that these applications can leverage the benefits of these algorithms to be used in the mining industry, which will help us to reach a decision quickly and efficiently, saving much money. It will reduce running costs as well. So, there is a vast potential for applying the KNN approach in the mining industry. These are the references, so let me summarize in a few sentences what we have covered in this lecture. We have discussed the known and its distance metric principle and the uh parameters used in this method, and we also have discussed the assumptions. Advantages and disadvantages We have also discussed the potential application in the mining industry approach. Thank you.