Mine Automation and Data Analytics

Prof. Radhakanta Koner

Department of Mining Engineering

IIT (ISM) Dhanbad

Week: 10

Lectures: 50

Naïve Bayes Classifier

Welcome to my course Mine Automation and Data Analysis. Today we will discuss the Naive classifier and the Naive Bayes classifier. It is a very popular classification technique used in machine learning applications. So, in this lesson, we will first introduce what a Naive Bayes classifier is. Then we will briefly go into the mathematics behind these NEBS-based classifiers and their assumptions. And we will elaborate with one example that will be helpful for you to understand how these classifiers basically work in a real application.

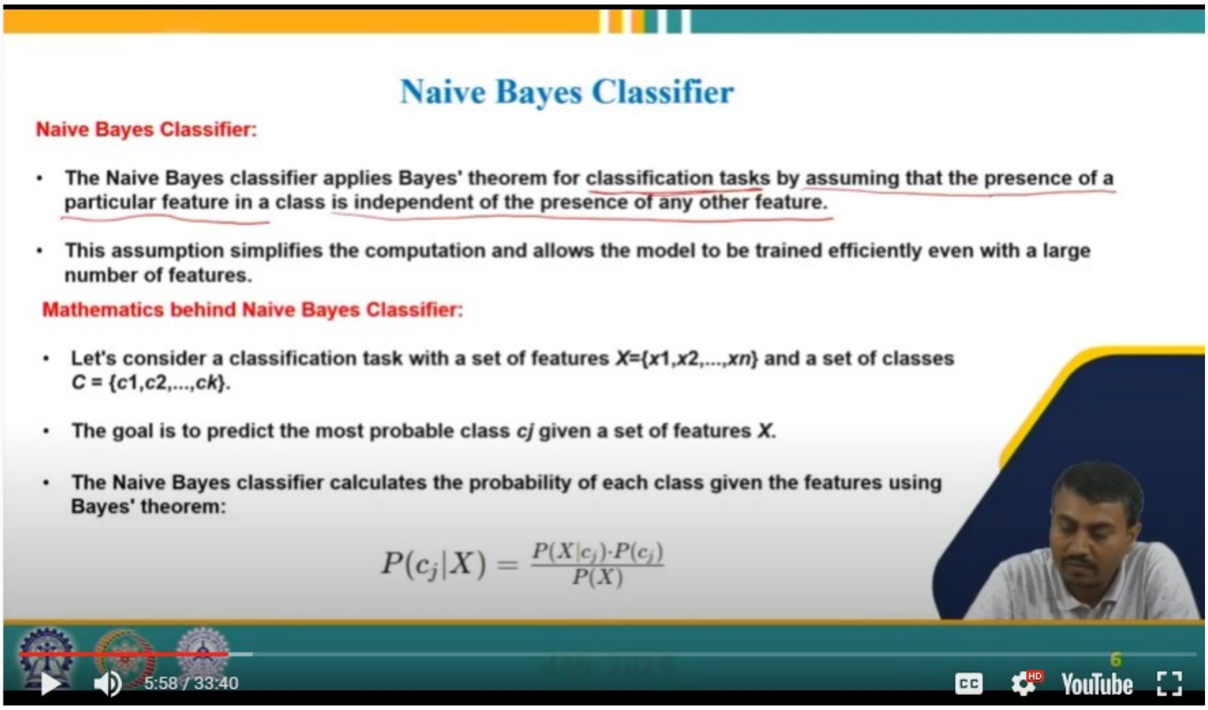
Then we will discuss the advantages and disadvantages of these classifiers and, finally, their application in the mining industry. So, what is a Naive Bayes classifier? The Naive Bayes classifier is a classifier based on the probability principle. As you have seen in the probability lesson, we have discussed the Bayes theorem for conditional probability. So, basically, some features of an image, a kind of happening, or a phenomenon might be dependent on others or might not be dependent.

So, correlating all these features finally and predicting some specific outcome out of it for this kind of situation and applying these probability principles is very helpful for reaching a fruitful conclusion. And that is why this kind of technique is very much in use for text classification and spam filtering. So, let us subdivide this problem into a few components. Firstly, the Bayes theorem. So, what is the Bayes theorem? The Bayes theorem is the fundamental theorem in probability theory that describes the probability of an event based on prior knowledge of conditions that might be related to the event.

Means the probability of happening A when the probability is that the event B is happening, for example. So, this is basically represented by the Bayes theorem representation. So, that same thing is imported here, and we are applying for the large data set, and it is really showing a good result. So, it is represented by this mathematical representation that probability of event A occurring given that the event B has occurred. Probability of event A occurring given that event B has occurred.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

So, this is represented by the P of BA, which is the probability of B happening when event A has already happened. So, this is the probability BA, which is the probability of event B occurring given that event A has occurred. And probability A and probability B are the probability of occurring that event independently. So, this expression is basically used, and this has been extended for the large data set and the large number of features and classes in the Naive Bayes classifier. So, the Naive Bayes classifier is basically applying the Bayes theorem to the classification task by assuming that the presence of a particular feature in a class is independent of the presence of any other feature. In the classification task, it is assumed that the presence of a particular feature in a class is independent of the presence of any other feature.



So, this feature is basically assumed to be basically obeying some distribution of some features. And these assumptions basically simplify the computation and allow the model to be trained efficiently, even with a large number of features. So, this is the advantage. So, let us see the mathematics. So, these are the features in classes X1, X2, and up to Xn, and the number of classes is C1, C2, up to Ck. So, our goal is to predict the most probable class, Cj, given a set of features. Our task is to predict the most probable class Cj when a set of features X is given. So, the Naive Bayes classifier calculates the probability of each class given the feature using the Bayes theorem. So, probability Cj when X is basically the feature, and that is represented by the probability of X given that the class for that is Cj, it is happening multiplied by the Pcj priority probability of Cj divided by probability. of X.

$$P(c_j|X) = \frac{P(X|c_j) \cdot P(c_j)}{P(X)}$$

So, here is an interesting reduction. The reduction is that the Px is constant for all classes. So, we can simplify the above equation by saying that Pcj of the feature X is happening, or given that particular feature X is proportional to the probability or likelihood of observing feature X given the class Cj multiplied by the priority probability of class J. So, let me summarize again: the probability of class Cj given the feature X is proportional to the likelihood of observing the feature X given the class Cj multiplied by the prior probability of class Cj. So, the Naive Bayes classifier assumes that the features are conditionally independent, and here the theorem is applied because of this assumption given the class level Cj.

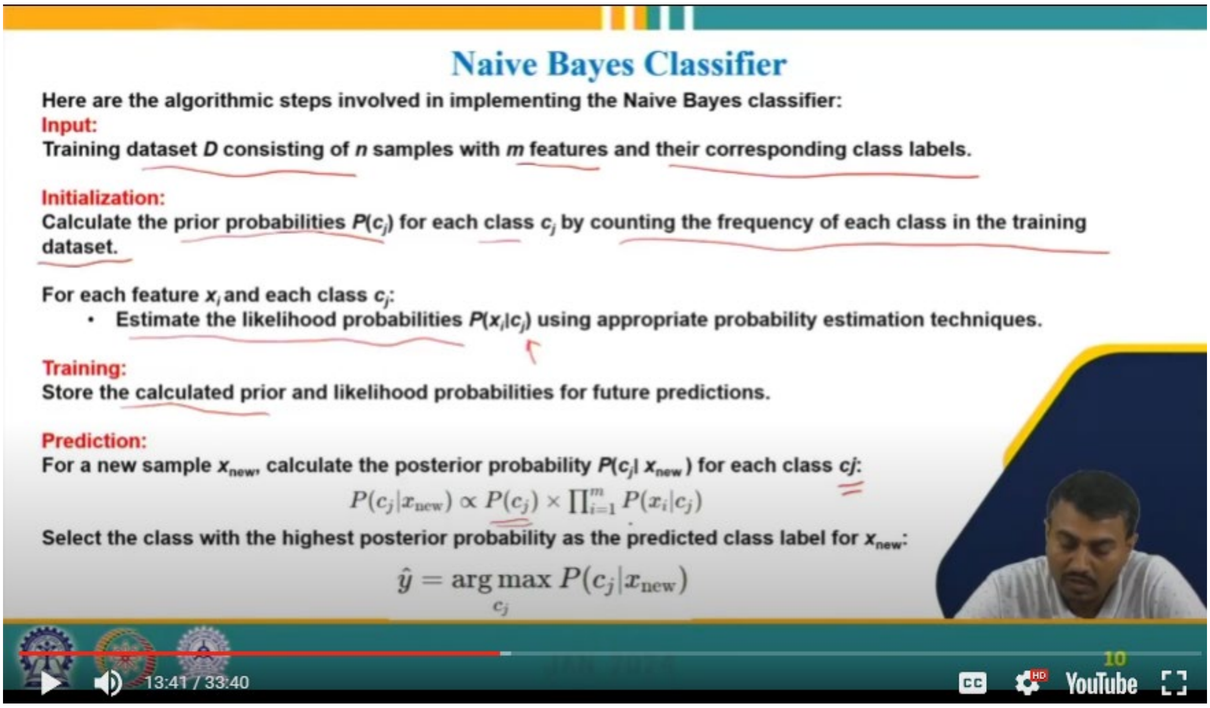
$$P(c_j|X) \propto P(X|c_j) \cdot P(c_j)$$

So, the Px the given class is Cj is equal to Px1 Cj into Px2 Cj multiplied up to the Pxn Cj. So, here, we have to estimate the parameters. So, to classify new instances, the Naive Bayes classifier needs to estimate two types of probabilities. One is the prior probability probability Cj, which is the probability of each class occurring in the data set, usually estimated by the frequency of each class in the training set. This data is known, and based on the data, we will prepare a table, and from the table, by counting, we can easily estimate the frequency of each class in the training set, which is the data given. Second is the likelihood probability Xi given the class Cj, the probability of observing each feature given the class of an estimated using maximum likelihood estimation or depending upon other types of features as well. Now the classification begins once the prior and likelihood probability are estimated for P of Xi Cj and the prior probability of Cj is estimated.

$$P(X|c_j) = P(x_1|c_j) \cdot P(x_2|c_j) \cdot ... \cdot P(x_n|c_j)$$

Now the Naive Bayes classifier predicts the class Cj with the highest posterior probability P Cj given the feature X, that is, Y cap predicted class level, which is equal to the argmax of Cj probability Cj given the feature X. So, it will give the maximum probability or highest posterior probability that the class is here with the features X maybe. So, let us say we have a binary classification problem with features X1 and X2 and two classes C1 and C2. So, we can calculate the posterior probability using a Naive Bayes classifier and make the prediction based on the highest probability.

So, here are the algorithmic steps for implementing the Naive Bayes classifier: input the training data set D consisting of N samples with M number of features and their corresponding class level. For initialization, we have to calculate the prior probability Cj for each class of Cj by counting the frequency of each class in the training data set. A data set is given to us, and from that, we can estimate that. So, for each feature Xi and each class Cj, we have to estimate the likelihood probability of P Xi given class Cj using the probability estimation technique. Then we have to store the calculated prior and likelihood probabilities of the future prediction.



So, the prediction is done for a new sample X new. Calculating the posterior probability P Cj given the feature X new for each class Cj is nothing, but P Cj new feature X new is proportional to the prior probability Cj into the probability of Xi and Cj given the class Cj summing over i is equal to 1 to m. So, here, select the class with the highest posterior probability as the predicted class level X new is Y cap is equal to the argmax of P Cj given the class feature X new. So, similar to the other machine learning methods, the Naive Bayes classifier also has several assumptions in order to simplify the calculation of probabilities and make predictions. And these assumptions are very important for all of us to understand and apply to specific machine learning applications. The first is feature independence, and it is one of the most significant assumptions of the naive Bayes classifier that the features are conditionally independent given the class level.

So, it means that the presence of one feature is assumed to be unrelated to the presence of any other feature, given the class level. So, despite this being a simplification and often not strictly true in real-world data, naive Bayes can still perform well in practice. This is also an advantage of this classifier. Class conditional feature distribution: the naive Bayes classifier assumes that each class has its own distribution of feature values. And in other words, the probability distribution of each feature given the class is assumed to be independent of the distribution of other features given the same class.

Predictive features: the classifier assumes that the features used for prediction are relevant and informative for the classification task. And features that are irrelevant or redundant may still be included in the model, but they should ideally have minimal impact on the classification decision. Data quality names Bayes assume classifiers assume that the training data is representative of the population and is of sufficient quality. Poor quality or bad data may lead to inaccurate predictions. Class prior probability: the classifier assumes that the prior probability of each class is known or can be accurately estimated from the training data.

If the class distribution is highly skewed or imbalanced, these assumptions may not hold true, or this might pose a problem. So, in that case, we may have to assign class weight or resample the data based on the available conditions. So, these assumptions are very important to understand. Another assumption is that the continuous feature is assumed to follow a specific distribution. Particularly in the Gaussian Bayes theorem, it is assumed that continuous features follow a Gaussian normal distribution within each class.

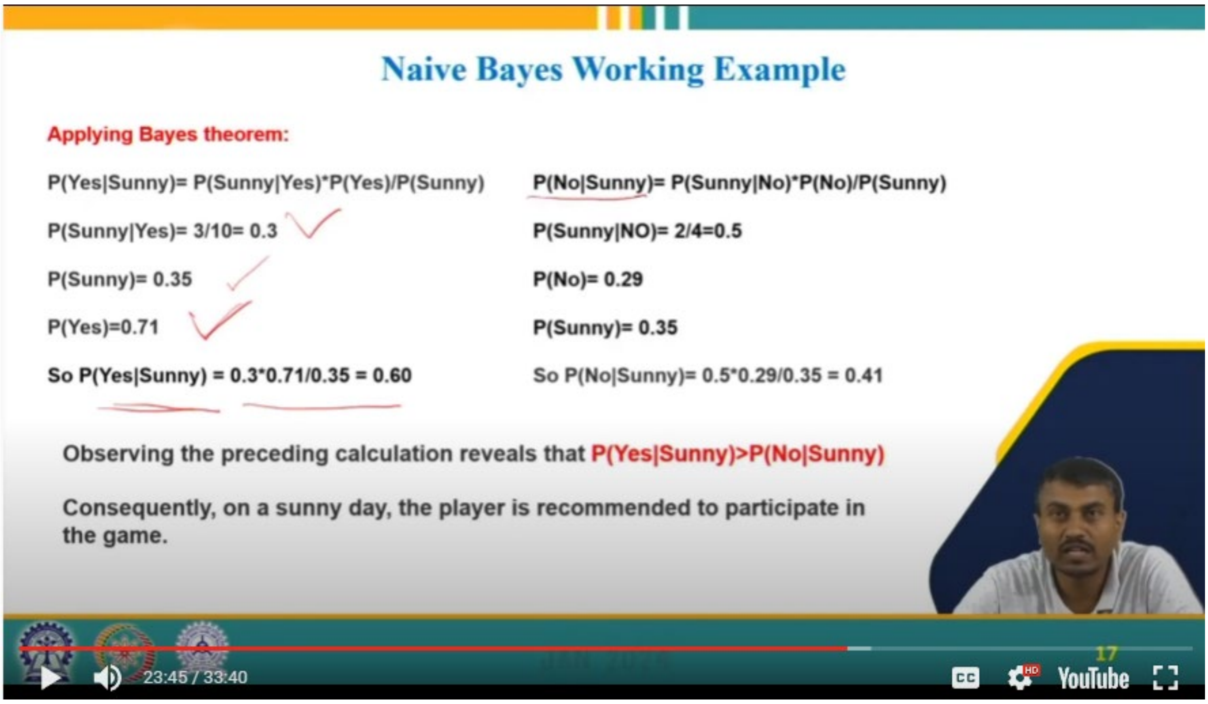
So, these assumptions may not always hold true in practice, especially for features with complex distributions. Zero-conditional probability handling. Naive Bayes classifiers assume that no conditional probability is zero. In practice, this may lead to an issue where a particular feature value does not occur in the training set for a given class. So, techniques like Laplace smoothing are often used to address this issue by adding a small constant to all counts.

So, understanding these assumptions is crucial when we are applying the Naive Bayes classifier in a real-world scenario. And if these conditions and assumptions are violated, then it will affect the performance of this classifier. And it is a very effective method for handling text classification or higher-dimensional data, and it is very computationally efficient to handle large feature spaces. Let us understand the Naive Bayes classifier in a real-world example. So, we have the data set of some target variable, play, indicating whether one should engage in outdoor activity on a given day based on these conditions.

So, that data is also available. So, we have to take these steps. Translate the data set into frequency tables, and then construct a likelihood table by computing the probabilities associated with the given features. Features include whether to utilize the Bayes theorem to ascertain the posterior probability. So, for instance, let us tackle the question: given the sunny weather, should the player engage in playing games or not? That is a very important question.

To resolve this, let us examine the data set provided here. The data set is here: the 13 number of observations total 14 number of observations we have, starting from 0 to 13, and these indicate some weather: sunny, overcast, rainy. These are the three situations. And we have the target variable, which is playing yes or no. So, here are the two responses: yes or no. So, we have prepared the frequency table.

So, these are the weather conditions: overcast, rainy, and sunny. Overcast yes is 5, rainy yes is 2, and sunny yes is 3. Total 10 and no overcast is 0 rainy is 2 sunny is 2 5. So, whether the weather is now overcast, no and yes.



So, this is basically a yes. So, likelihood of weather condition is 5 by 14 total 14 number of observation. Rainy is 4 by 14; sunny is 5 by 14. So, the weather no. for all conditions is 0. 29 or, yes, 0.71. So, apply the Bayes theorem here now to the to the probability of playing yes given that it is sunny weather. So, it is equal to the probability of sunny weather given that it is playing into the probability of playing yes and sunny weather divided by the probability of sunny weather. So, probability sunny yes is 0.3 probability sunny is 0. 35 probability yes is 0.71. So, probability yes sunny is 0.6, and probability no sunny is basically 0.41, similarly calculated. So, observing this probability, yes, sunny is greater than probability no sunny. So, based on these, on a sunny day, it is recommended that players participate in the game.

So, let us see the advantages of these technologies or methods. It is a very simple and fast algorithm, and Naive Bayes is a very simple and fast algorithm, making it easy to implement and computation efficient. And it scale well with large data set and high dimensional feature space efficient with large feature space. The Naive Bayes classifier performs well even with a large number of features. And it can handle a data set with thousands of features without significant computational overhead.

Robust to irrelevant features: The Naive Bayes classifier is robust to irrelevant features, and it can still produce good results even if some features are not informative for the classification task. It is handled well with the missing data, basically by simply ignoring the missing value during the trading and prediction. Effective for text classification because of these advantages, it is very much used for text classification, spam filtering, and sentiment analysis. And it is also applicable to a relatively small amount of training data as well. In a probabilistic framework, the Naive Bayes classifier provides probabilistic prediction, allowing for easy interpretation of results and uncertainty estimation.

Disadvantage: assumption of feature independence. These assumptions, that is, feature independence, may not hold true in many real-world data sets. So, in cases where features are highly correlated, a Naive Bayes classifier may produce a suboptimal result. Sensitivity to feature distribution: Naive Bayes assumes features follow a specific distribution within each class, for example, Gaussian for continuous features. If this assumption is violated, the classifier performance metric rate. Due to its simple probabilistic model, the Naive Bayes classifier may not capture complex relationships between features and class levels as effectively as more sophisticated algorithms like decision trees or neural networks.

Zero-frequency Naive Bayes classifiers may encounter issues when a categorical feature value appears in the testing data set but not in the training data set. This can lead to zero frequency counts and affect the classifier's performance. Can not handle numeric data well, while the Gaussian Naive Bayes classifier can handle continuous numeric data. Other variants, like multinomial and Bernoulli Naive Bayes classifiers, are designed for categorical data or discrete features. So, handling numeric data directly with these variants may require binding or other pre-processing techniques. Require well-balanced classes. Naive Bayes classifiers tend to perform better when the class distribution is balanced, and in a data set with highly imbalanced classes, they may favor the majority class and produce biased predictions.

So, it is important for us to understand these limitations, especially under the conditions where we need to classify the context. So, in what context may we use these, or in what context may we not use them for accurate predictions? Let us discuss some of the potential applications of this Naive Bayes classifier in mining engineering. First is the mineral prospectivity mapping. Mineral prospectivity mapping aims to identify areas with high potential for mineral deposits based on geological, geophysical, and geochemical data.

So, here, the Naive Bayes classifier analyzes a spatial data set containing geological features, mineral occurrences, and exploration data to predict the likelihood of finding an economically viral mineral deposit in an unexplored or underexplored region. So, by considering factors such as geological formation, structural control, and mineralization indicators, the Naive Bayes classifier can assist in prioritizing exploration targets and optimizing resource allocation in mineral exploration campaigns. Underground mine safety monitoring involves detecting hazardous conditions and ensuring the safety of workers in the underground mining environment. So, here we can analyze the real-time sensor data from equipment, ventilation system data, gas detector data, and personnel tracking device data to identify potential safety hazards such as gas leaks, equipment malfunctions, and personnel emergencies. So, by classifying these sensor data patterns associated with safety-critical events, the Naive Bayes classifier can trigger alarms, initiate safety protocols, and provide early warning to miners and supervisors, and by doing so, we can avoid accidents in the underground mines.

Ore grade estimation is essential for optimizing mineral processing operators, resource planning, and mine economics by accurately quantifying the mineral content and quality of ore. So, here we can analyze the multinomial data set comprising geological, geochemical, and mineralogical data to predict ore grade and mineral recoveries in mining operations. Water management and tailings prediction involves assessing water resources, minimizing water consumption, and predicting the behavior of tailings facilities to mitigate environmental risk. So, here we can analyze hydrological data, meteorological data, and geological data to predict water inflows, groundwater interactions, and their stability in different mining operations. So, by considering factors such as precipitation pattern, surface water runoff, and geological characteristics, these classifiers can forecast water-related risks, optimize water management strategies, and support decision-making regarding tailings disposal, dam construction, and environmental remediation efforts. Another application is energy-efficient efficiency optimization.

Energy efficiency optimization aims to reduce energy consumption and greenhouse gas emissions in mining operations by improving energy efficiency, implementing renewable energy sources, and optimizing energy management practices. So, we can analyze the energy consumption data, process the parameters of the of the operational variables to identify opportunities for energy savings, optimize equipment utilization, and prioritize energy efficiency measures in mining operations. So, by classifying the energy usage pattern, equipment performance, metrics, and energy efficiency indicators, these classifiers can support decision-making regarding energy audits, equipment upgrades, and renewable energy integration. So, contribute to sustainable mining practices and cost reduction initiatives.

These are the references. Let me conclude in a few sentences what we have covered. So, we have introduced the Naive Bayes classifier mathematically, discussed its assumptions, shown a worked-out example, discussed the advantages and disadvantages of these classifying methods, and finally, discussed the potential applications in the mining industry. Thank you.