Information Theory

▼ Probability in information theory (conditional probability, joint probability), accuracy and robustness of machine learning classifiers, confusion matrix, ROC, Information, Entropy.

1. **Probability in Information Theory**:

    - **Conditional Probability**: Conditional probability is the probability of an event occurring given that another event has already occurred. Mathematically, it is denoted as P(A|B), the probability of event A given event B.
      Example: In a deck of cards, suppose we draw two cards without replacement. What is the probability that the second card drawn is a King given that the first card drawn was a King? Here, the conditional probability would be calculated as the probability of drawing a King as the second card given that the first card drawn was a King.

      A = sun will rise
      B = it is raining

      P(A|B) = sun will rise given that it is raining

      P(B|A) = It is raining given that sun will rise

    - **Joint Probability**: Joint probability is the probability of two (or more) events occurring simultaneously. Mathematically, it is denoted as P(A ∩ B), the probability of event A and event B occurring.
      Example: Consider tossing two fair coins. What is the probability of getting heads on both coins? This is an example of joint probability, as it involves the probability of both events (getting heads on the first coin and getting heads on the second coin) occurring simultaneously.

      It is raining and sun will rise

2. **Accuracy and Robustness of Machine Learning Classifiers**:

    - **Accuracy**: Accuracy is a measure of how often a classifier correctly predicts the class label. It is calculated as the ratio of the number of correct predictions to the total number of predictions made.

Example: In a binary classification problem where we predict whether emails are spam or not, if the classifier correctly classifies 90 out of 100 emails, its accuracy would be 90%.

- **Robustness**: Robustness refers to the ability of a classifier to maintain its performance under different conditions, such as noisy data or changes in the distribution of the data.
  Example: A sentiment analysis classifier trained on movie reviews should ideally perform well even if the reviews contain misspellings or grammatical errors.

3. **Confusion Matrix**:

   - A confusion matrix is a table that is often used to evaluate the performance of a classification algorithm. It presents a summary of the classifier's predictions compared to the true labels of the data.

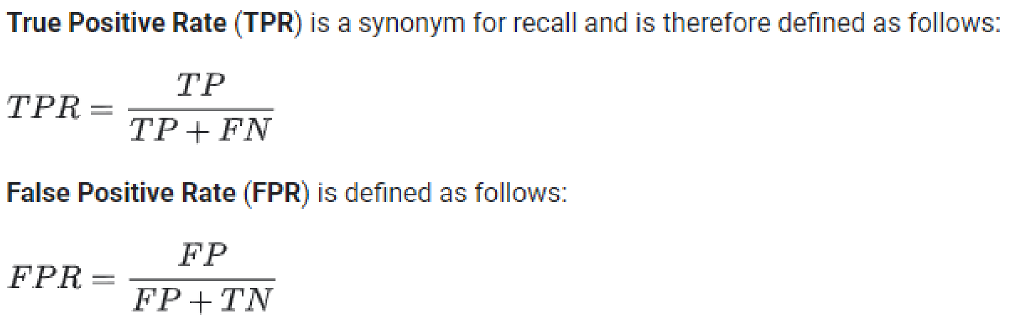
   - Example:

```
             Predicted Class
          |  Positive  |  Negative  |
-------------------------------------
Actual    |            |            |
Class     |------------|------------|
          |            |            |
 Positive |    TP      |    FN      |
          |            |            |
-------------------------------------
          |            |            |
 Negative |    FP      |    TN      |
          |            |            |
-------------------------------------
```

     - TP: True Positive (correctly predicted positive instances)
     - TN: True Negative (correctly predicted negative instances)
     - FP: False Positive (incorrectly predicted as positive when it's actually negative, also known as Type I error)

- FN: False Negative (incorrectly predicted as negative when it's actually positive, also known as Type II error)

4. **ROC (Receiver Operating Characteristic) Curve**:
    - ROC curve :An ROC curve (receiver operating characteristic curve) is **a graph showing the performance of a classification model at all classification thresholds.** It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

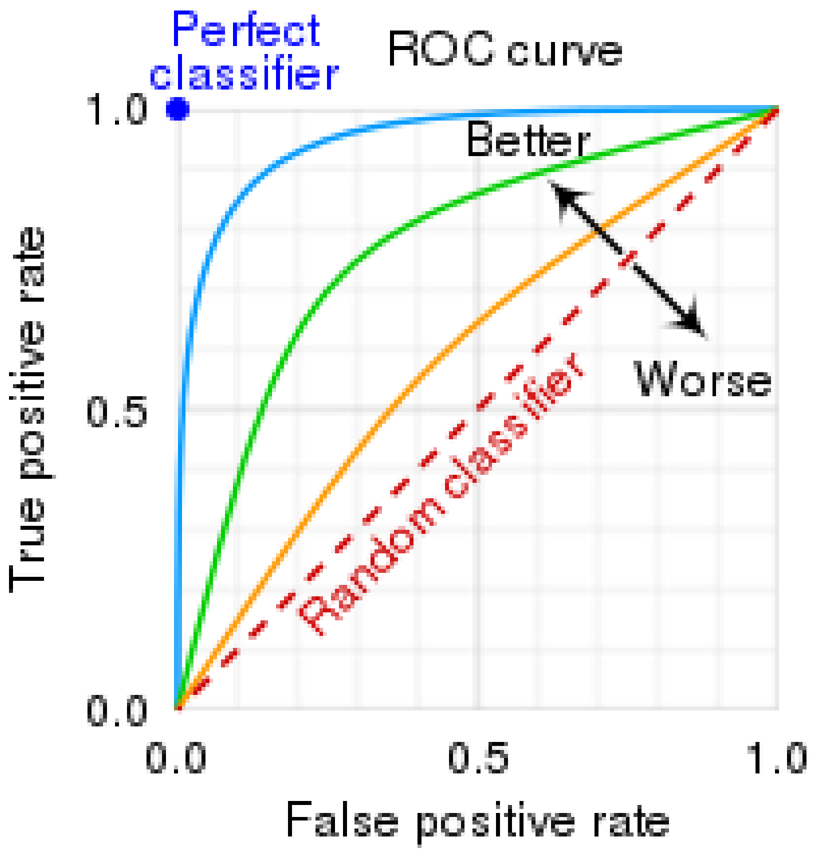


True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$



- Example: In a medical diagnostic test for a disease, the ROC curve plots the trade-off between sensitivity (true positive rate) and specificity (true negative rate) as the threshold for classifying a sample as positive or negative is varied.

5. **Information and Entropy**:
    - **Information**: In information theory, information quantifies the "surprise" or "uncertainty" associated with the outcome of a random variable. It is inversely proportional to the probability of occurrence of an event. Mathematically, it is defined as I(x) = -log(P(x)), where P(x) is the probability of event x.
      Example: Consider a fair coin flip. If it comes up heads, the information gained is -log(0.5) = 1 bit, because heads and tails are equally likely.
    - **Entropy**: Entropy measures the average amount of information (or uncertainty) contained in a set of messages or random variables. Mathematically, it is defined as H(X) = -Σ P(x) log(P(x)), where X is a random variable and P(x) is the probability distribution over all possible values of X.
      Example: In a fair six-sided die, each outcome has a probability of 1/6. Therefore, the entropy of the die is -Σ (1/6) log(1/6) = log(6) ≈ 2.58 bits.

These concepts are fundamental in various fields, especially in machine learning and information theory, and understanding them thoroughly is essential for developing and evaluating models effectively.

▼ Dependency and uncertainty in training data, linear correlation, mutual information, false neighbourhood, anomaly detection, machine learning-based anomaly detection.

1. **Dependency and uncertainty in training data**: Dependency refers to the relationship between variables in a dataset, indicating how changes in one variable affect another. Uncertainty in training data reflects the level of confidence or reliability associated with the data points. Understanding dependencies and uncertainties is crucial for building accurate and robust models.

Example: Consider a dataset containing information about housing prices. The price of a house (dependent variable) may depend on various factors such as size, location, number of bedrooms, etc. These factors (independent variables) exhibit dependencies among themselves and with the house price. However, there might be uncertainties associated with some features, like missing values in the dataset or inaccuracies in the reported data.

2. **Linear correlation**: Linear correlation measures the strength and direction of the relationship between two continuous variables. It is typically quantified using correlation coefficients such as Pearson correlation coefficient or Spearman rank correlation coefficient.

   Example: In a study examining the relationship between hours of study and exam scores, a strong positive linear correlation might be observed. That is, as the number of hours spent studying increases, the exam scores also tend to increase. A correlation coefficient close to 1 indicates a strong positive linear correlation

3. **Mutual information**: Mutual information measures the amount of information that one random variable contains about another random variable. It quantifies the degree of dependency between variables and is particularly useful for feature selection and dimensionality reduction.

   Example: Consider a dataset containing both the temperature and humidity readings. Mutual information quantifies how much knowing the temperature can reduce uncertainty about humidity and vice versa. If knowing the temperature gives a lot of information about humidity and vice versa, then the mutual information between these variables would be high.

4. **False neighborhood**: In the context of anomaly detection, false neighborhood refers to instances where normal data points are incorrectly identified as anomalies. Minimizing false positives is essential for building effective anomaly detection systems.

   Example: In network intrusion detection, false positives occur when legitimate network activities are incorrectly flagged as malicious. For instance, if a benign software update is mistaken for a malicious intrusion attempt, it results in a false positive. Minimizing false positives is crucial to ensure that security systems do not unnecessarily disrupt normal operations.

5. **Anomaly detection**: Anomaly detection is the task of identifying rare or unusual events, patterns, or observations in a dataset that deviate significantly from the norm. Anomalies may indicate potential fraud, errors, or other interesting phenomena.

   Example: In credit card fraud detection, anomalies might include unusually large transactions, transactions from unfamiliar locations, or a sudden increase in transaction frequency. Anomaly detection algorithms aim to identify these irregular patterns that deviate from the normal behavior of legitimate transactions.

6. **Machine learning-based anomaly detection**: Machine learning techniques are often employed for anomaly detection tasks. These techniques involve training models on normal data to learn its underlying patterns and then identifying deviations from these patterns as anomalies. Common machine learning algorithms used for anomaly detection include isolation forest, one-class SVM, and autoencoders.

   Example: Anomaly detection in IoT devices could involve using machine learning algorithms to monitor sensor data. For instance, in a network of temperature sensors, anomalies could indicate a malfunctioning sensor or an unusual environmental condition. By training a machine learning model on normal sensor data, deviations from the learned patterns can be flagged as anomalies.

▼ Clustering, low-dimensional representation of high-dimensional data, data compression, (Principal Component Analysis (PCA), SVD, t-SNE, Self Organizing Maps)

1. **Clustering**: Clustering is the task of dividing a dataset into groups, or "clusters," where data points in the same group are more similar to each other than to those in other groups. Common clustering algorithms include k-means, hierarchical clustering, and DBSCAN.

2. **Low-dimensional representation of high-dimensional data**: Many real-world datasets have high dimensionality, which can make analysis and visualization challenging. Techniques for reducing dimensionality aim to

represent the data in a lower-dimensional space while preserving its essential structure and relationships.

3. **Data compression**: Data compression refers to techniques for reducing the size of data, typically to save storage space or transmission bandwidth. In the context of machine learning and data analysis, data compression can also be useful for speeding up computation and reducing the complexity of models.
4. **Principal Component Analysis (PCA)**: PCA is a popular technique for dimensionality reduction. It identifies the directions (principal components) in which the data varies the most and projects the data onto a lower-dimensional subspace defined by these components.
5. **Singular Value Decomposition (SVD)**: SVD is a matrix factorization technique that decomposes a matrix into three other matrices, capturing the underlying structure of the original matrix. It is widely used in various machine learning tasks, including dimensionality reduction, data compression, and collaborative filtering.
6. **t-distributed Stochastic Neighbor Embedding (t-SNE)**: t-SNE is a nonlinear dimensionality reduction technique particularly well-suited for visualization of high-dimensional data.

   In high-dimensional data, each data point is represented by a vector of features. t-SNE aims to find a lower-dimensional representation of this data while keeping similar data points close to each other in the lower-dimensional space.

   To achieve this, t-SNE first constructs a probability distribution over pairs of high-dimensional data points. This probability distribution represents the similarity between data points in the original high-dimensional space. Then, it constructs a similar probability distribution over pairs of points in the low-dimensional space.

   The algorithm then adjusts the positions of the points in the low-dimensional space in order to minimize the difference between these two probability distributions. In other words, it tries to find a configuration of points in the lower-dimensional space where the relationships between points (as measured by similarities) are as similar as possible to the relationships between points in the original high-dimensional space.

By doing so, t-SNE effectively preserves the local structure of the data, meaning that nearby points in the original high-dimensional space remain close to each other in the lower-dimensional embedding. This makes it particularly useful for visualizing complex, high-dimensional data in a way that retains meaningful relationships between data points.

https://www.youtube.com/watch?v=NEaUSP4YerM&ab_channel=StatQuestwithJoshStarmer

7. **Self-Organising Maps (SOM)**: SOM is a type of artificial neural network that learns to organize high-dimensional data into a low-dimensional, usually two-dimensional, grid. It is particularly useful for visualizing and understanding the structure of complex datasets, as it preserves the topology of the input space. (how dimensionality reduction happens in Self-Organizing Maps (SOM)?????)

   https://www.youtube.com/watch?v=_IRcxgG0FL4&ab_channel=MaheshHuddar

▼ Communication over a noisy channel (basic types of noisy channels, information conveyed by a channel, the noisy-channel coding theorem)

<Refer to chapter “Communication over a noisy channel”>

Basic types of noisy channel: https://youtu.be/Ev-NQPB1-iU?si=hwIO5Jq0gMtdXYKm

Coding theorem: https://www.youtube.com/results?search_query=the+noisy-channel+coding+theorem

▼ Coding theory, Huffman coding, concepts of arithmetic coding, Hamming codes

Huffman Coding: Huffman coding is a popular algorithm used for lossless data compression. It works by assigning variable-length codes to input characters, with shorter codes assigned to more frequent characters. This ensures that the most common characters are represented by shorter codes, leading to overall compression of the data. The algorithm constructs a binary tree called a Huffman tree, where each leaf node represents a character and each internal node represents the combined frequency of its child nodes. Huffman coding is widely used in file compression algorithms like ZIP.

<Explain with example>

Arithmetic coding

Method that goes hand in hand with the philosophy that compression of data from a source entails probabilistic modelling of that source.

As of 1999, the best compression methods for text files use arithmetic coding

Several state-of-the-art image compression systems use it too.

<Refer to chapter “Stream Codes”>

https://youtu.be/ouYV3rBtrTI?si=8R0TgrY7IXKy6uH8

Hamming codes:

<Refer to chapter “Stream Codes”>

https://youtu.be/1A_NcXxdoCc?si=Mt72aZqNvbcry3P8