

a) Compare and contrast Linear Regression and Logistic Regression Machine Learning algorithms

Certainly! Linear regression and logistic regression are both fundamental machine learning algorithms used for different types of tasks, particularly regression and classification, respectively. Let's compare and contrast them:

- **Nature of the Dependent Variable:**

- Linear Regression: The dependent variable (the variable we are trying to predict) in linear regression is continuous. It can take any real value within a given range.
- Logistic Regression: The dependent variable in logistic regression is categorical. It typically represents binary outcomes, such as "yes" or "no", "spam" or "not spam", etc.

- **Output Type:**

- Linear Regression: The output of linear regression is a continuous value that represents the best estimate of the dependent variable given the input features.
- Logistic Regression: The output of logistic regression is a probability value between 0 and 1, representing the likelihood of belonging to a particular class. It's often interpreted as the probability of the positive class.

- **Algorithm Objective:**

- Linear Regression: The objective of linear regression is to find the best-fitting linear relationship between the independent variables (features) and the dependent variable. This is typically done by minimizing the sum of squared differences between the observed and predicted values.
- Logistic Regression: The objective of logistic regression is to model the probability that a given input belongs to a specific category or class. It does

this by fitting a logistic curve to the data, which allows it to predict probabilities.

- **Cost Function:**

- Linear Regression: In linear regression, the cost function is typically the Mean Squared Error (MSE) or Mean Absolute Error (MAE), which measures the average squared (or absolute) difference between the observed and predicted values.
- Logistic Regression: In logistic regression, the cost function is the Log Loss (or Cross-Entropy Loss), which measures the difference between the actual and predicted probabilities. It penalizes incorrect classifications with a higher cost.

- **Decision Boundary:**

- Linear Regression: In linear regression, there is no explicit decision boundary since it's a regression algorithm dealing with continuous values.
- Logistic Regression: Logistic regression uses a decision boundary to classify instances into different classes based on the calculated probabilities. The decision boundary is typically a straight line (for binary classification) or a hyperplane (for multi-class classification).

- **Applications:**

- Linear Regression: Common applications include predicting house prices, stock prices, sales forecasting, etc.
- Logistic Regression: Common applications include binary classification tasks such as spam detection, credit risk analysis, disease diagnosis, etc.

In summary, while both linear regression and logistic regression are based on similar principles of fitting a line or curve to the data, they are used for different types of tasks (regression vs. classification) and have different mathematical formulations and objectives.

b) Describe a reliable method that can be used to test whether an entity has learned or not.

Certainly! Let's expand on the steps mentioned for testing for learning using a validation dataset:

- **Splitting the Data:**

- Splitting the dataset into multiple subsets helps ensure that the model's performance is evaluated on data it hasn't seen during training, thus simulating real-world scenarios. The typical split involves dividing the dataset into a training set, a validation set, and optionally a test set.
- The training set is used to train the model's parameters, the validation set is used to tune hyperparameters and evaluate performance during training, and the test set is reserved for a final evaluation of the trained model's performance.

- **Training the Model:**

- During the training phase, the model is exposed to the training data and learns patterns and relationships between the input features and the target variable.
- The model's parameters are adjusted iteratively through optimization algorithms (e.g., gradient descent) to minimize the error between the actual and predicted values.

- **Validation:**

- The validation set serves as a proxy for unseen data and is used to evaluate the model's performance during training.
- Performance metrics such as accuracy, precision, recall, F1-score, etc., are computed on the validation set to assess how well the model generalizes to unseen data.
- By monitoring the model's performance on the validation set, adjustments can be made to the model's architecture, hyperparameters, or training process to improve its performance.

- **Adjustment:**

- If the model's performance on the validation set is unsatisfactory, adjustments can be made to improve it.
- This may involve tweaking hyperparameters (e.g., learning rate, regularization strength), changing the model architecture (e.g., adding or removing layers), or experimenting with different optimization algorithms.
- The training process is then repeated using the updated configurations until satisfactory performance is achieved on the validation set.

- **Final Evaluation (Optional):**

- Once the model is trained and its hyperparameters are optimized, a final evaluation can be performed on the test set to assess its performance on unseen data.
- The test set provides an unbiased estimate of the model's performance and helps gauge its generalization ability to new, unseen instances.
- It's important to note that the test set should only be used once to avoid overfitting to the test data.

By following this systematic process of training, validation, adjustment, and optional final evaluation, practitioners can effectively test whether their machine learning model has learned meaningful patterns from the training data and how well it generalizes to unseen data.

c) Distinguish between hold-out and N-fold cross-validation methods of evaluating Machine Learning models.

Certainly! Let's delve deeper into both hold-out validation and N-fold cross-validation:

Hold-out Validation:

In hold-out validation, the dataset is typically split into two subsets: a training set and a validation set. The model is trained on the training set, and its performance is evaluated on the validation set. Here's a more detailed explanation:

- **Splitting the Data:** The dataset is randomly divided into two disjoint subsets: a training set and a validation set. The training set usually comprises a larger portion of the data, while the validation set is smaller.
- **Training the Model:** The model is trained on the training set using various algorithms and hyperparameters. The goal is to find the optimal parameters that minimize the error on the training data.
- **Evaluating Performance:** After training, the model's performance is evaluated on the validation set using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score). This step helps assess how well the model generalizes to unseen data.
- **Adjustment and Iteration:** Based on the performance on the validation set, the model's hyperparameters can be adjusted, or different algorithms can be tried. This process may involve multiple iterations until a satisfactory model is obtained.

Advantages of Hold-out Validation:

- Simple to implement.
- Computationally efficient, especially for large datasets.

- Provides a straightforward estimate of model performance.

Disadvantages of Hold-out Validation:

- The performance estimate may have high variance depending on how the data is split.
- Limited data for training and validation, especially if the dataset is small.

N-fold Cross-Validation:

N-fold cross-validation is a more robust method for evaluating model performance, especially when dealing with limited data. Here's how it works:

- **Dividing the Data into Folds:** The dataset is divided into N roughly equal-sized subsets or folds. Each fold contains approximately the same proportion of samples and maintains the original class distribution.
- **Training and Validation:** The model is trained N times, with each iteration using a different fold as the validation set and the remaining N-1 folds as the training set.
- **Performance Evaluation:** After each training iteration, the model's performance is evaluated on the validation fold using appropriate metrics.

The performance metrics from each iteration are averaged to obtain an overall estimate of the model's performance.

- **Aggregating Results:** The final performance estimate is calculated by averaging the performance metrics from all N iterations.

Advantages of N-fold Cross-Validation:

- Utilizes the entire dataset for both training and validation, providing a more reliable estimate of model performance.
- Helps reduce the variance in the performance estimate compared to hold-out validation.
- Suitable for small datasets where splitting the data into separate training and validation sets may lead to biased results.

Disadvantages of N-fold Cross-Validation:

- Computationally more expensive compared to hold-out validation, especially for large datasets or complex models.
- May not be suitable for time-sensitive applications due to increased training time.

In summary, while hold-out validation is simpler and computationally efficient, N-fold cross-validation provides a more robust estimate of model performance, particularly when dealing with limited data or when a reliable performance estimate is crucial.

(d) Demonstrate how the Euclidean distance between nominal attributes that involve color (e.g., the distance between the values red, green and blue) are determined in an instance based knowledge representation scheme.

Certainly! Let's delve deeper into representing colors as vectors and computing Euclidean distance for color attributes:

Color Representation as Vectors:

In a typical RGB color model, colors are represented as combinations of three primary colors: red, green, and blue. Each color component (red, green, blue) is represented by an intensity value ranging from 0 to 255. Therefore, a color can be represented as a three-dimensional vector in RGB space, where each dimension corresponds to one of the color components.

For example, the color red can be represented as the vector (255, 0, 0), indicating maximum intensity of red and zero intensity of green and blue. Similarly, the color green can be represented as the vector (0, 255, 0), and the color blue as the vector (0, 0, 255).

Euclidean Distance for Color Attributes:

Once colors are represented as vectors in RGB space, the Euclidean distance between two colors can be computed using the standard Euclidean distance formula. The Euclidean distance between two points (vectors) in a multi-dimensional space is calculated as the square root of the sum of squared differences between their corresponding components.

For example, the Euclidean distance between two colors represented as vectors $v = (v_1, v_2, v_3)$ $v = (r_1, g_1, b_1)$ and $w = (w_1, w_2, w_3)$ $w = (r_2, g_2, b_2)$ can be calculated as follows:

Euclidean

$$\text{Distance} = \sqrt{(v_1 - w_1)^2 + (v_2 - w_2)^2 + (v_3 - w_3)^2}$$

$$\text{Euclidean Distance} = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}$$

This distance metric allows us to quantify the similarity or dissimilarity between colors in RGB space. Colors with smaller Euclidean distances are considered more similar, while colors with larger distances are considered more dissimilar.

Example:

Let's consider two colors: red $((255,0,0)(255,0,0))$ and green $((0,255,0)(0,255,0))$. Using the Euclidean distance formula:

Euclidean

$$\begin{aligned} \text{Distance} &= (255-0)^2 + (0-255)^2 + (0-0)^2 \\ &= 255^2 + (-255)^2 + 0^2 = 65025 + 65025 \\ &= 130050 \approx 360.62 \end{aligned}$$

$$\begin{aligned} \text{Distance} &= (255-0)^2 + (0-255)^2 + (0-0)^2 \\ &= 255^2 + (-255)^2 + 0^2 = 65025 + 65025 \\ &= 130050 \approx 360.62 \end{aligned}$$

So, the Euclidean distance between red and green colors is approximately 360.62 units in RGB space.

By representing colors as vectors and computing Euclidean distances between them, we can effectively quantify the similarity or dissimilarity between colors, which is useful in various applications such as image processing, color matching, and computer graphics.

(e) Using relevant examples, discuss four applications of Machine Learning.

Certainly! Let's dive deeper into each of these applications of machine learning:

1. Image Recognition:

- **Object Detection:** Machine learning algorithms, especially CNNs, can detect and localize objects within images, enabling applications like self-driving cars, surveillance systems, and augmented reality.
- **Facial Recognition:** Machine learning models can recognize and identify faces in images or videos, leading to applications such as biometric security systems, photo tagging on social media platforms, and personalized user experiences.
- **Image Classification:** Machine learning algorithms can classify images into predefined categories, such as identifying handwritten digits, identifying diseases from medical images, or classifying species from wildlife photographs.

2. Natural Language Processing (NLP):

- **Sentiment Analysis:** NLP techniques enable sentiment analysis, which involves determining the sentiment or emotion expressed in a piece of text. This is used in social media monitoring, customer feedback analysis, and market research.
- **Language Translation:** Machine learning models, including sequence-to-sequence models and transformers, are used for language translation tasks, enabling communication across different languages in applications like Google Translate.
- **Chatbots:** NLP-powered chatbots use machine learning algorithms to understand and respond to natural language inputs, providing customer support, virtual assistants, and automated messaging services.

3. Recommendation Systems:

- **Content-Based Filtering:** Machine learning algorithms analyze user preferences and recommend items similar to those the user has liked or interacted with before, such as movies, music, products, or articles.
- **Collaborative Filtering:** Recommendation systems analyze user behavior and preferences across a large user base to identify patterns and make

recommendations based on similarities between users or items.

- **Hybrid Approaches:** Some recommendation systems combine content-based and collaborative filtering techniques to provide more accurate and personalized recommendations.

4. Healthcare:

- **Disease Diagnosis:** Machine learning algorithms analyze medical data, including patient records, medical images, and genetic information, to assist in disease diagnosis, early detection, and prognosis prediction.
- **Personalized Treatment Plans:** Machine learning models can analyze patient data to recommend personalized treatment plans, medication dosages, and therapeutic interventions based on individual characteristics and responses.
- **Medical Imaging Analysis:** Machine learning algorithms analyze medical images, such as X-rays, MRI scans, and histopathology slides, to assist radiologists and pathologists in identifying abnormalities, tumors, and other medical conditions.

These applications demonstrate how machine learning techniques are transforming various industries and

improving processes, decision-making, and user experiences across different domains.

Question 2: (15 Marks)

a) Using illustrations, analyze goals of transfer learning. (4 marks)

Certainly! Let's provide more detail on each part:

a) Goals of Transfer Learning:

- **Improve Learning Efficiency:** Transfer learning reduces the need for large amounts of labeled data for training a model on the target task. Instead of starting from scratch, the model leverages knowledge from the source task, which helps in learning the target task more efficiently.
- **Enhance Model Generalization:** By transferring knowledge from a related source task, transfer learning helps the model generalize better to new, unseen data in the target task. The learned representations or patterns from the source task contribute to building a more robust and generalizable model for the target task.
- **Adaptation to New Domains:** Transfer learning allows models trained on one domain to adapt and perform well in a different but related domain. This adaptability increases the applicability of machine learning models across

various domains, saving time and resources required for training separate models for each domain.

- **Address Data Scarcity:** In scenarios where labeled data for the target task is limited or expensive to acquire, transfer learning is particularly beneficial. By utilizing knowledge from the source task, transfer learning overcomes the data scarcity issue and enables effective learning on the target task with fewer labeled examples.

b) Measures by which Transfer Might Improve Learning:

- **Feature Representation:** Transfer learning improves learning by transferring knowledge about feature representations learned from the source task to the target task. This facilitates the extraction of relevant features from the input data, leading to better performance on the target task.
- **Model Initialization:** Transfer learning initializes the parameters of the model using pre-trained weights from the source task. This initialization provides a good starting point for training on the target task, enabling the model to converge faster and achieve better performance.

- **Regularization:** Transfer learning acts as a form of regularization by constraining the model to learn representations that are useful for both the source and target tasks. By leveraging the shared knowledge between tasks, transfer learning prevents overfitting and improves the model's ability to generalize to new data.

c) Distinguishing Inductive Transfer from Bayesian Transfer:

Inductive Transfer:

- **Definition:** Inductive transfer involves transferring empirical knowledge in the form of learned patterns, representations, or rules from the source task to the target task.
- **Focus:** It focuses on transferring experience-based knowledge gained from the source task to improve learning on the target task.
- **Examples:** Transferring feature representations learned from a pre-trained convolutional neural network (CNN) for image classification or transferring decision rules learned from one classification task to another.

Bayesian Transfer:

- **Definition:** Bayesian transfer involves transferring probabilistic knowledge, such as prior distributions or likelihood functions, from the source task to the target task.
- **Focus:** It leverages Bayesian inference to update prior knowledge with evidence from the target task and make predictions or decisions based on the updated posterior distribution.
- **Examples:** Transferring prior knowledge about the distribution of model parameters in a Bayesian framework or transferring prior beliefs about class distributions in a classification task.

In summary, transfer learning offers various benefits such as improved learning efficiency, enhanced model generalization, adaptation to new domains, and addressing data scarcity. It achieves these goals through measures like transferring feature representations, model initialization, and regularization. Additionally, transfer learning can be classified into inductive transfer, focusing on empirical knowledge transfer, and Bayesian transfer, focusing on probabilistic knowledge transfer.

b) Elucidate three measures by which transfer might improve learning. (6 marks)

c) Using illustrations, distinguish inductive transfer from Bayesian transfer. (5 marks)

Question 3: (15 Marks)

a) Explain three learning styles in machine learning applications. (3 marks)

Certainly! Let's delve deeper into each learning style in machine learning applications:

1. Supervised Learning:

- **Definition:** Supervised learning is a type of machine learning where the model is trained on labeled data, meaning each input example is paired with a corresponding output label. The model learns to map input features to output labels based on the provided labeled examples.
- **Example:** Consider a spam email classification task where the input data consists of email content and the output labels indicate whether the email is spam or not. The model learns from a labeled dataset containing examples of both spam and non-spam emails, and it learns to classify new emails into one of these categories based on their content.
- **Applications:** Supervised learning is widely used in various applications such as image classification, speech recognition, sentiment analysis, and medical diagnosis.

2. Unsupervised Learning:

- **Definition:** Unsupervised learning is a type of machine learning where the model is trained on unlabeled data, meaning there are no predefined output labels. The goal of unsupervised learning is to discover hidden patterns, structures, or relationships within the data without explicit guidance.
- **Example:** Consider a customer segmentation task where the input data consists of customer demographic information and purchase history. The goal is to identify distinct groups or segments of customers based on similarities in their behavior or characteristics. Unsupervised learning algorithms like k-means clustering or hierarchical clustering can be used to partition the data into meaningful clusters.

- **Applications:** Unsupervised learning is used in applications such as clustering, anomaly detection, dimensionality reduction, and recommendation systems.

3. Reinforcement Learning:

- **Definition:** Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent takes actions in the environment, receives feedback in the form of rewards or penalties, and learns to maximize cumulative rewards over time.
- **Example:** Consider training a computer program to play a game like chess or Go. The program acts as an agent that makes moves on the game board, and the game environment provides feedback in the form of winning or losing the game. The agent learns from experience by trying different strategies and observing the outcomes of its actions.
- **Applications:** Reinforcement learning is used in applications such as autonomous driving, robotics, game playing, and recommendation systems.

In summary, supervised learning focuses on learning from labeled data to predict output labels, unsupervised learning focuses on discovering patterns or structures in unlabeled data, and reinforcement learning focuses on learning optimal decision-making policies through interaction with an environment. Each learning style has its own set of algorithms, techniques, and applications, catering to different types of learning tasks and scenarios.

b) Consider the data shown in table 1.2. Derive two classification rules and two association rules from this data. (4 marks)

c) Illustrate four ways in which knowledge derived from a machine learning scheme can be represented using clusters. (8 marks)

Question 4: (15 Marks)

a) With an illustration, briefly describe the instance-based knowledge representation i.e., the nearestneighbor classification method. (7 marks)

Sure, let's first understand how to apply the 1R algorithm and calculate errors for the rules in the table. Then, we'll compute the total error for each attribute.

Applying the 1R Algorithm:

The 1R algorithm is a simple rule-based classification algorithm that generates rules based on the most common class label for each attribute. Here's how to apply it:

- **For each attribute, determine the most common class label.**
- **Generate a rule using the most common class label for each attribute.**
- **Calculate the error of each rule by counting the instances where the predicted class label does not match the actual class label.**

Now, let's create a hypothetical dataset and apply the 1R algorithm to illustrate the process:

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rain	Mild	High	False	Yes
Rain	Cool	Normal	False	Yes
Rain	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rain	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rain	Mild	High	True	No

Step 1: Determine Most Common

Class Label for Each Attribute:

- **Outlook:** The most common class label is "Sunny," with 5 instances of "No" and 4 instances of "Yes."

- **Temperature:** The most common class label is "Mild," with 4 instances of "Yes" and 3 instances of "No."
- **Humidity:** The most common class label is "High," with 7 instances of "No" and 4 instances of "Yes."
- **Windy:** The most common class label is "False," with 6 instances of "Yes" and 2 instances of "No."

Step 2: Generate Rules:

- **Rule for Outlook:** If Outlook is "Sunny," predict "No." Otherwise, predict "Yes."
- **Rule for Temperature:** If Temperature is "Mild," predict "Yes." Otherwise, predict "No."
- **Rule for Humidity:** If Humidity is "High," predict "No." Otherwise, predict "Yes."
- **Rule for Windy:** If Windy is "False," predict "Yes." Otherwise, predict "No."

Step 3: Calculate Errors:

To calculate errors, compare the predicted class label based on the rule with the actual class label for each instance. Count the instances where the predicted label does not match the actual label.

For example:

- For Outlook, the rule predicts "No" for "Sunny" instances. So, the error is 1 because one instance labeled "Sunny" is actually "Yes."
- For Temperature, the rule predicts "Yes" for "Mild" instances. So, the error is 3 because three instances labeled "Mild" are actually "No."
- For Humidity, the rule predicts "No" for "High" instances. So, the error is 2 because two instances labeled "High" are actually "Yes."
- For Windy, the rule predicts "Yes" for "False" instances. So, the error is 2 because two instances labeled "False" are actually "No."

After calculating errors for each attribute, you can sum them up to find the total error for each attribute. Let me know if you need further clarification!

Question 4: (15 Marks)

a) With an illustration, briefly describe the instance-based knowledge representation i.e., the nearestneighbor classification method. (7 marks)

b) Consider the following rules derived from contact lens data that tells the kind of contact lens to prescribe, given certain information about a patient.

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Using these rules, construct a decision tree that describes the structure of the lens data. (8 marks)

Certainly! Let's construct a decision tree based on the given rules derived from contact lens data:

Given Rules:

- If the age of the patient is young and the spectacle prescription is myope, prescribe soft contact lenses.
- If the age of the patient is young and the spectacle prescription is hypermetrope, prescribe hard contact lenses.
- If the age of the patient is pre-presbyopic and the astigmatic condition is no, prescribe soft contact lenses.
- If the age of the patient is pre-presbyopic and the astigmatic condition is yes, prescribe hard contact lenses.
- If the age of the patient is presbyopic, prescribe either soft or hard contact lenses depending on the tear production rate.

Construction of Decision Tree:

To construct the decision tree, we'll start with the root node and split the data based on each attribute (age, spectacle prescription, and astigmatic condition) according to the given rules.

- **Root Node:**

- Split based on age:
 - Young
 - Pre-presbyopic
 - Presbyopic

- **Split for Young Patients:**

- If the spectacle prescription is myope, prescribe soft contact lenses.
 - If the spectacle prescription is hypermetrope, prescribe hard contact lenses.

- **Split for Pre-presbyopic Patients:**

- If the astigmatic condition is no, prescribe soft contact lenses.
 - If the astigmatic condition is yes, prescribe hard contact lenses.

- **Split for Presbyopic Patients:**

- Further split based on tear production rate:
 - Low tear production rate: Prescribe hard contact lenses.
 - Normal or high tear production rate: Prescribe soft contact lenses.

Decision Tree Structure:

Age

**/ | **

Young Pre-presbyopic Presbyopic

| | |

**Soft CL Astigmatic? Tear Production
Rate**

/ \

No / \ Yes

/ \

Soft CL Hard CL

This decision tree represents the structure of the lens data based on the given rules. It provides a systematic approach to classify patients and prescribe the appropriate type of contact lenses based on their age, spectacle prescription, astigmatic condition, and tear production rate.

New chat

Previous 7 Days

James Recognized, Assistant

Upgrade plan

James Mburu

ChatGPT 3.5

tyr like the recent

ChatGPT

Sure, let's solve this step by step using the Naive Bayes algorithm:

Given Data:

Age	Spectacle Prescription	Astigmatism	Tear Production Rate	Recommended Lenses
Young	Myope	Yes	Normal	Soft
Young	Hypermetrop	Yes	Reduced	Hard
Pre-presbyopic	Myope	No	Reduced	Soft
Pre-presbyopic	Hypermetrop	No	Normal	Soft
Presbyopic	Myope	Yes	Reduced	Soft
Presbyopic	Hypermetrop	Yes	Normal	Hard

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

ChatGPT 3.5

New Patient:

Age	Spectacle Prescription	Astigmatism	Tear Production Rate	Recommended Lenses
Young	Myope	Yes	Normal	?

a) Applying Naive Bayes Algorithm:

1. Calculate Prior Probabilities:

- $P(Hard) = \frac{3}{6} = 0.5$
- $P(Soft) = \frac{3}{6} = 0.5$

2. Calculate Likelihoods for Each Attribute:

- For Age = Young:
 - $P(Age = Young|Hard) = \frac{2}{3}$
 - $P(Age = Young|Soft) = \frac{1}{3}$
- For Spectacle Prescription = Myope:
 - $P(Myope|Hard) = \frac{1}{3}$
 - $P(Myope|Soft) = \frac{2}{3}$
- For Astigmatism = Yes:

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

ChatGPT 3.5

For Spectacle Prescription: Myope

- $P(Myope|Hard) = \frac{1}{3}$
- $P(Myope|Soft) = \frac{2}{3}$

For Astigmatism = Yes:

- $P(Yes|Hard) = \frac{2}{3}$
- $P(Yes|Soft) = \frac{1}{3}$

For Tear Production Rate = Normal:

- $P(Normal|Hard) = \frac{1}{3}$
- $P(Normal|Soft) = \frac{2}{3}$

3. **Apply Naïve Bayes Algorithm:**

$$P(Hard|Young, Myope, Yes, Normal)$$

$$= P(Young|Hard) \times P(Myope|Hard) \times P(Yes|Hard) \times P(Normal|Hard) \times P(Hard)$$

$$= \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} \times 0.5$$

$$= \frac{4}{162}$$

b) Calculate the Probability of Prescribing the Hard Lens:

The probability of prescribing the hard contact lens is the sum of the probabilities of prescribing

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Type here to search

21°C 20:57 15/04/2024

ChatGPT 3.5

$$= P(Young|Hard) \times P(Myope|Hard) \times P(Yes|Hard) \times P(Normal|Hard) \times P(Hard)$$

$$= \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} \times 0.5$$

$$= \frac{4}{162}$$

b) Calculate the Probability of Prescribing the Hard Lens:

The probability of prescribing the hard contact lens is the sum of the probabilities of prescribing the hard lens and the soft lens.

$$P(Hard) = P(Hard|Young, Myope, Yes, Normal) \times P(Hard)$$

$$P(Hard) = \frac{4}{162} \times 0.5$$

$$P(Hard) = \frac{2}{162}$$

So, the probability of the optician prescribing the hard contact lens for the new patient is $\frac{2}{162}$.

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Type here to search

21°C 20:57 15/04/2024