Week-6

Naïve Bayes Overview and Project Announcement

V Semester - ML and AIUP, Aug-Oct 2024

Session Date and Time: 26th Oct 2024 10:45 AM IST – 12:00 Noon IST

Venkateswar Reddy Melachervu
Visiting Faculty and CTO, Brillium Technologies
Department of Computer Science Engineering – AI and ML (CSM)
Email: venkat.reddy.gf@gprec.ac.in

G.Pulla Reddy Engineering College (Autonomous)
G.Pulla Reddy Nagar, Nandyal Road, Kurnool, AP 518007, India
Website: https://www.gprec.ac.in

Disclaimer and Confidentiality Notice

The content of this guest lecture, including all discussions, materials, and demonstrations, code is intended for educational purposes only and reflects the views and opinions of the speaker. While every effort has been made to ensure the accuracy and relevance of the information presented, it should not be considered as legal, financial, or professional advice.

Brillium Technologies retains unrestricted ownership of all information shared during this session. Participants must not record, reproduce, distribute, or disclose any part of the lecture or materials without prior written permission from Brillium Technologies. Unauthorized use or distribution of the content may result in legal action.

Additionally, all trademarks, service marks, and intellectual properties referenced or used in this presentation are the property of their respective owners. No ownership or rights over such third-party content are claimed or implied by the author or Brillium Technologies or by GPREC.

By attending this lecture, you agree to respect the confidentiality of the information shared and refrain from using it in any unauthorized manner. Failure to comply with these terms may result in legal action.

Thank you for your understanding and cooperation.

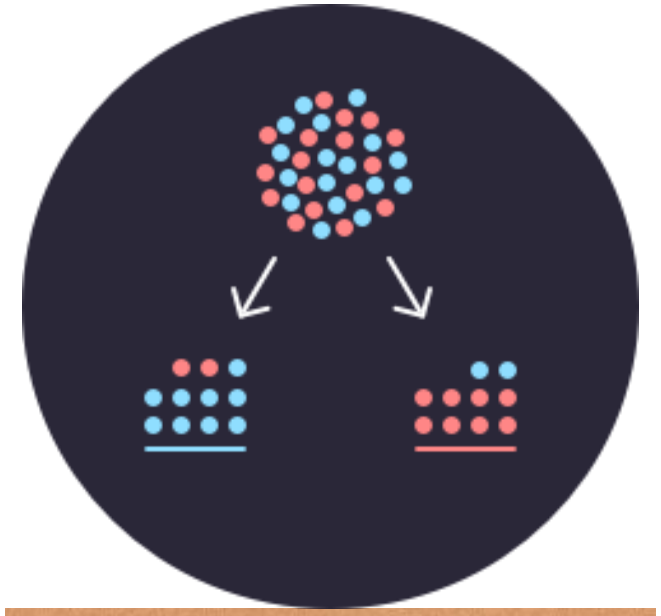
Lecture Outline

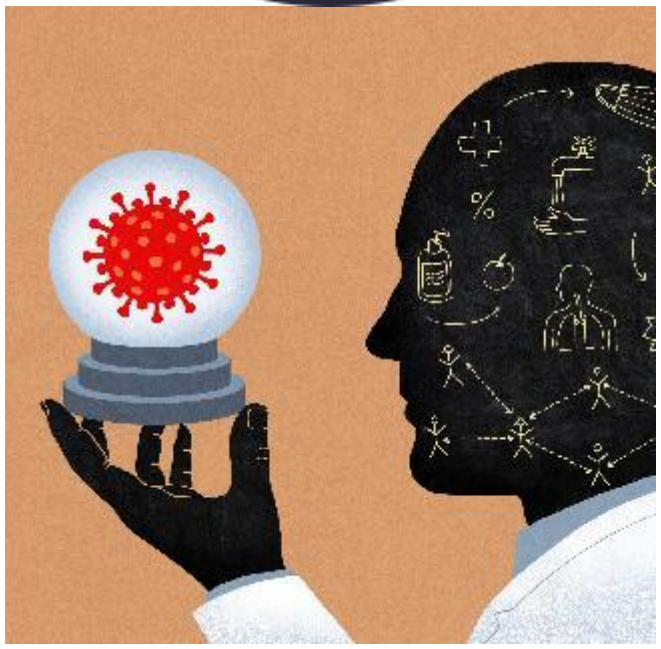
- Real-life Classification Scenario
- Probabilistic Models and Inferences
- Discriminative and Generative Models
- Bayes' Theorem
- Bayesian Probability and Rule
- Probability Distributions in Bayesian Inference
- Joint Probability of Multiple Dependent Events
- Email Spam Filtering
- Naïve Bayes Assumption
- Prior and Posterior Probabilities
- NB Classifier and Boundary Conditions
- Zero Frequency Problem and Laplace Smoothing
- Metrics
- Project and Dataset Details
- Implementation Steps
- TF-IDF
- Project Demo
- Q&A

A Practical Scenario of Real-life Classification Inference

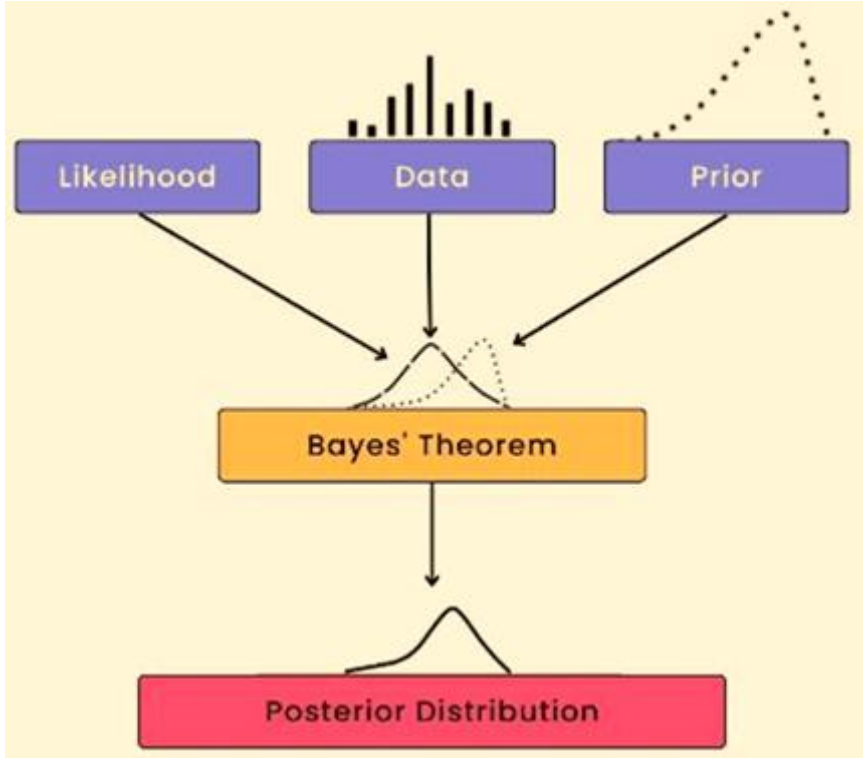
- You are a doctor and a patient comes in for a disease test
  - Disease is rare but the test is fairly accurate but NOT perfect
- Let
  - H be doctor's hypothesis – **patient has disease**
  - E be the evidence – **positive test result**
- Prior/historical information doctor has
  - **1%** people tested have the disease - $\boldsymbol{P(H) = 1\% = 0.01}$
    - **True Positive**
  - Test is **95% accurate/likelihood** for those who have disease - $\boldsymbol{P(E \mid H) = 0.95}$
    - **True Positives + False Negatives**
  - **1.1%** people tested have positive result across the whole population - $\boldsymbol{P(E) = 0.011}$
    - **True Positives + False Negatives**
- Doctor can not say patient has disease, even though the test is positive – why?
  - **Test result is NOT perfect** – contains **False Negatives**
- So, doctor needs to find the disease probability $\boldsymbol{P(H \mid E)}$ for the patient with
  - The prior/historical information and
  - Current evidence – test result

- **How?**





A Practical Scenario of Real-life Classification Inference

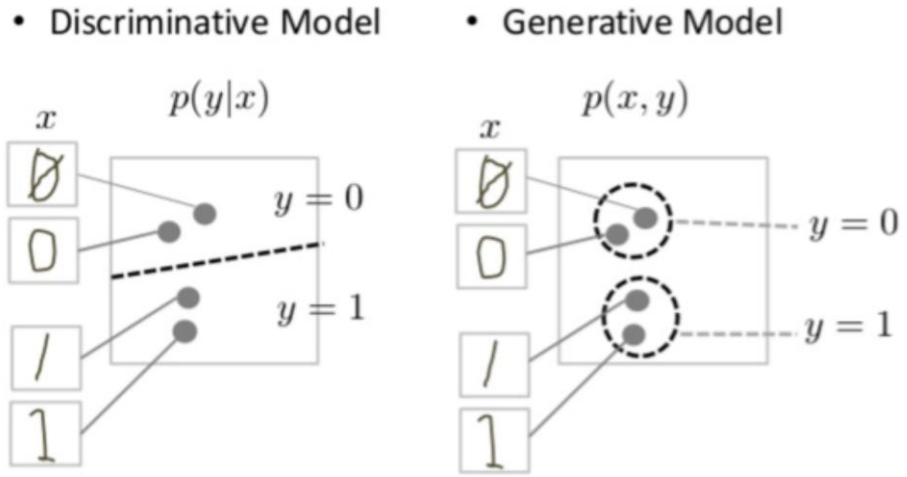


Bayesian Probability Comes to Doctor's Rescue!

Probabilistic Models and Inferences

- Deducing unknown or hidden information from known data by utilizing probability theory
- Process of calculating/updating probabilities for making predictions for unseen variables given observed data or new evidences
- Probabilistic models are defined in terms of probability distributions – where uncertainty and variability are modeled
  - Examples – Bayesian, HMM, Markov Random Fields
- Inference is about computing unknown probabilities or expected values for an outcome given some observed events/data

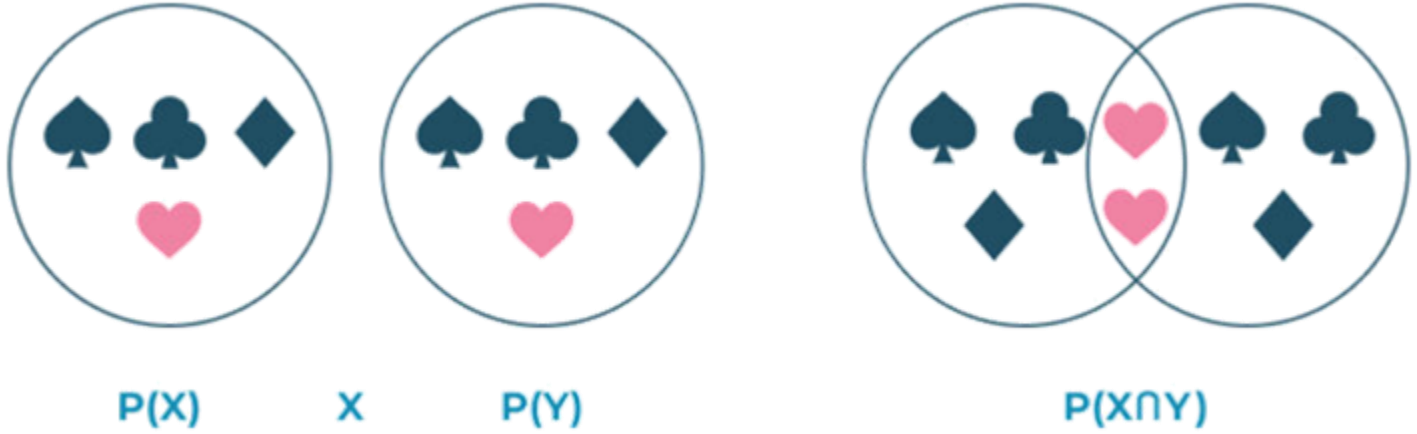
Probabilistic Generative and Discriminative Models



Aspect	Discriminative Models	Generative Models
Objective	Model the conditional probability $P(Y \mid X)$	Model the joint probability $P(X, Y)$
Focus	Learn the boundary that separates different classes	Understand the underlying distribution of data
Examples	Logistic Regression, SVM, Neural Networks, Random Forests	Naive Bayes, Gaussian Mixture Models, HMM, VAEs
Advantages	Better classification performance, directly optimizes decision boundaries	Can generate new data, handles missing data well
Disadvantages	Cannot generate new data, requires a lot of labeled data	Requires assumptions about data distribution, less accurate for classification tasks
Usage	Classification, regression	Data generation, density estimation

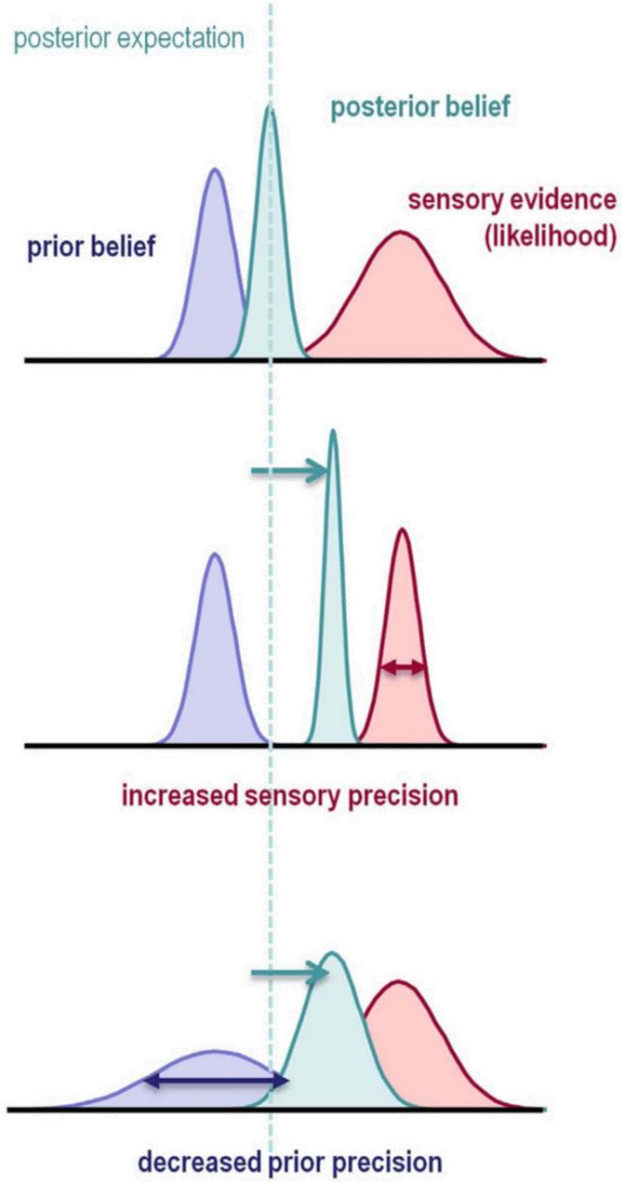
Bayes' Theorem

- When the inference involves predicting a class or a value based on two or more events/data points happening together, the probability is called **joint probability**
- **Joint Probability** of events A and B occurring for an outcome Y - $P(A \cap B \mid Y)$
- From Probability theory, **Joint Probability** $P(A \cap B)$
  - $= P(A) \times P(B)$ – if A, B are independent events
  - $= P(A) \times P(B|A) = P(B) \times P(A|B)$ – if A, B are dependent events - A key to understand Bayes' Theorem
- Deriving **Bayes' theorem** from joint probability
  - $P(A) \times P(B|A) = P(B) \times P(A|B)$
  - $\Rightarrow P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$
- A is **Hypothesis/Cause** and B is **Evidence/Observed Outcome**



Bayesian Probability and Rule

- Statistical inference method that updates the probability for a hypothesis/event as more evidence or information becomes available for that hypothesis/event
- Rooted in Bayes' theorem - The updated probability of an event based on prior knowledge/probabilities and current evidence
- $P(H \mid E) = \frac{P(E \mid H) \times P(H)}{P(E)}$
- $P(H \mid E)$
  - Called **Posterior Probability** of Hypothesis being true
  - The updated probability of hypothesis/event being true given new evidence E
  - Read as - **Probability of Hypothesis is being True Given The New Evidence**
- $P(E \mid H)$
  - Called **Evidence Likelihood**
  - Read as - **The probability of observing the Evidence E given the Hypothesis H**
- $P(H)$
  - Called **Prior Probability** of Hypothesis
  - The initial/prior probability of Hypothesis H before seeing the current evidence
- $P(E)$
  - Called **Marginal Probability/Likelihood of Evidence**
  - Marginal term comes from statistics which refers to the process of summing over one variable values for each instance value of other
  - Summing up (marginalizing) all probabilities of observing Evidence E across all possible hypotheses

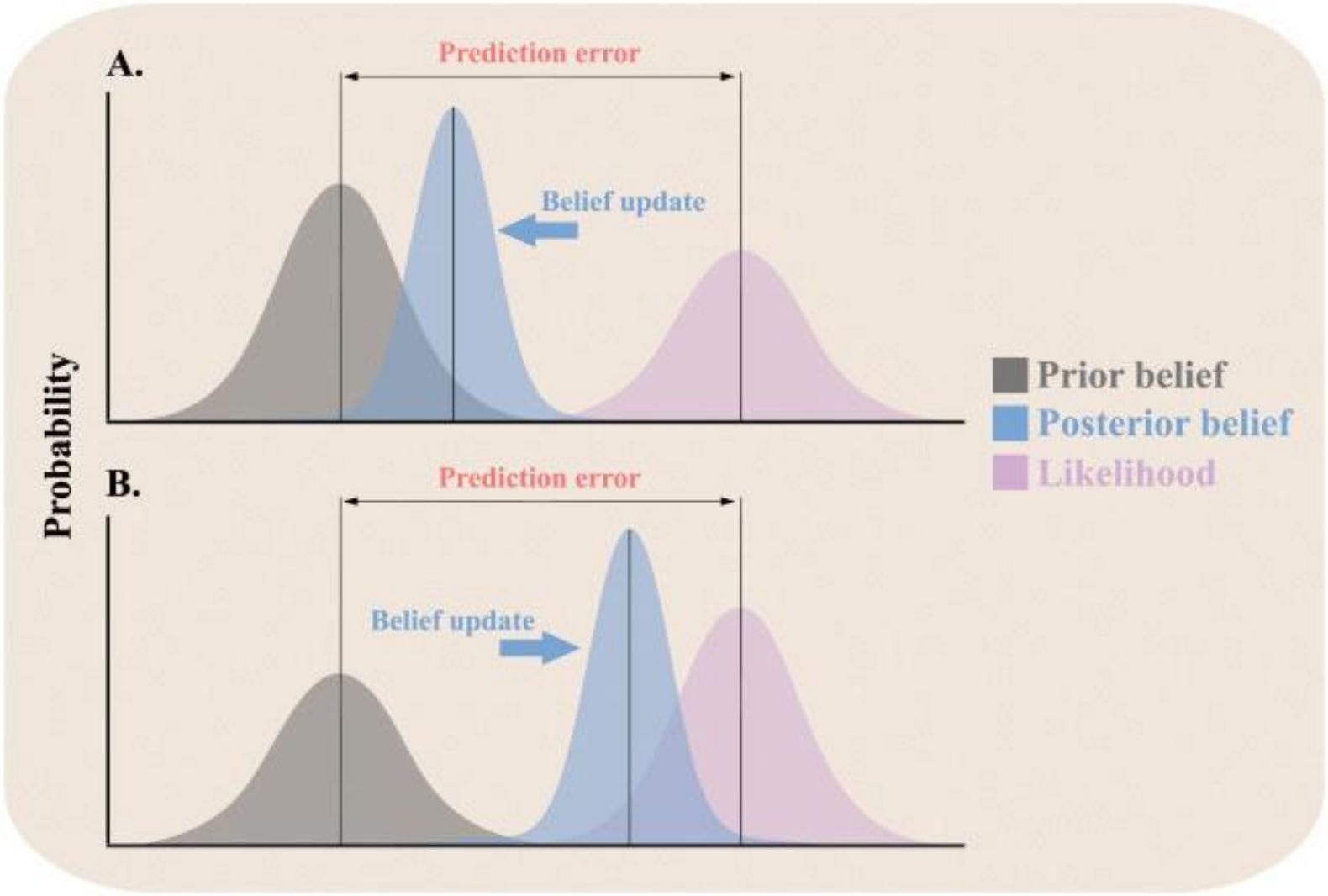


Bayesian Probability Comes to Doctor's Rescue!

- Disease probability $\boldsymbol{P(H \mid E)}$ for the patient with prior information and current evidence – test result
- $Patient\ Disease\ Probability\ = \frac{\boldsymbol{Test\ Accuracy \times Prior\ Patient\ Disease\ Probability\ in\ Total\ Population}}{\boldsymbol{Probability\ of\ Positive\ Test\ Result\ across\ Total\ Population}}$
- $\boldsymbol{P(H \mid E) = \frac{P(E \mid H) \times P(H)}{P(E)} = \frac{0.95 \times 0.01}{0.011} = 0.864 = 86.4\%}$

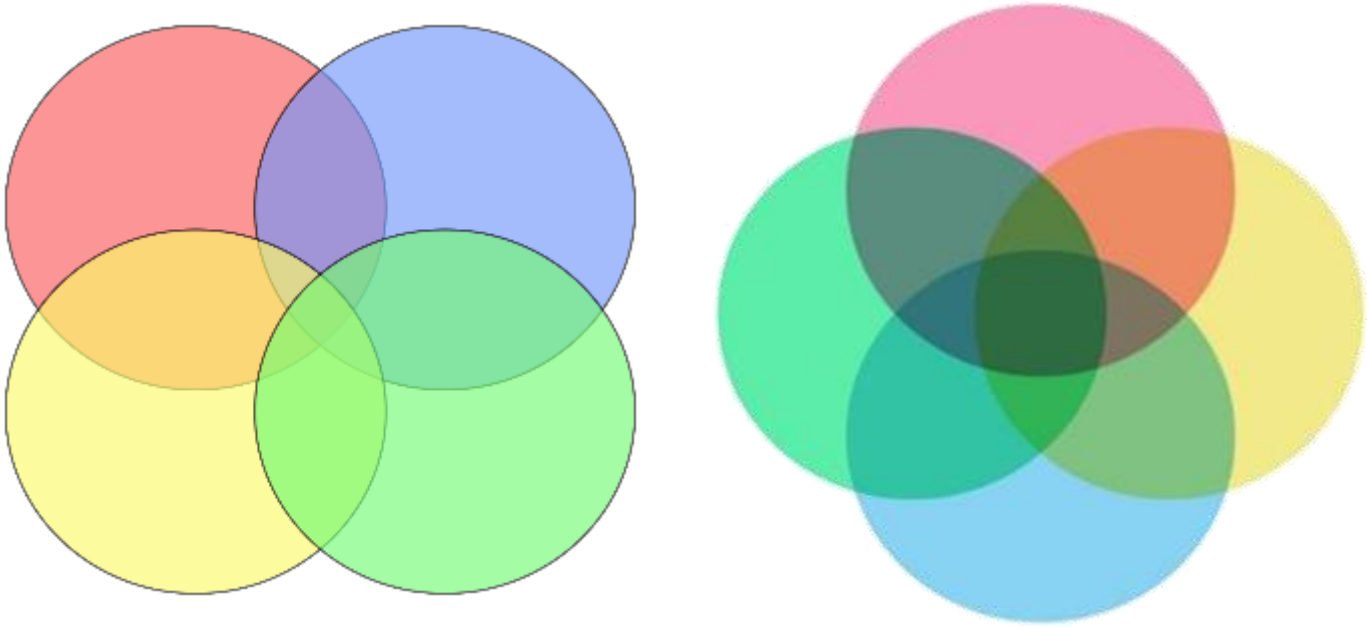
Though test result is positive, the probability **patient has the disease is only 86.4% - given the test result accuracy is NOT perfect!**

Probability Distributions in Bayesian Inference



Joint Probability for An Outcome of Dependent Multiple Events

- Let $A_1, A_2, A_3 \ldots A_n$ represent n dependent events for an outcome Y
- The joint probability of all these events for outcome $Y = P(A_1 \cap A_2 \cap \cdots \cap A_n | \, Y)$
- $P(A_1 \cap A_2 \cap \cdots \cap A_n | \, Y) =$
  - $P(A_1) \times P(A_2|A_1) \times P(A_3|A_2 \cap A_1) \times \cdots \times P(A_n|A_{n-1} \cap A_{n-2} \cap \cdots \cap A_2 \cap A_1)$



Inference gets complex, messy, and compute intensive for higher dimensional input features/events!

Is there a simple yet effective way?

Email Spam Filtering

- Let's consider a spam filter algorithm – a received email is a SPAM or GENUINE
- Consider the feature vector $\bar{v}$ of size N – number of words in Oxford English Dictionary
  - N ≈ 6,00,000
- For each received email, create the feature vector $\bar{v}$ of size N
  - $\bar{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}_{6,00,000 \times 1}$
- $v_j = 1$ if the email contains j^{th} of the dictionary, else $v_j = 0$
- Received email contains the sentence - ***Enter the Diwali festive bash with biggest discounts!***
- Position/index of each word from above sentence in Oxford English Dictionary
  - **Enter** - 32,500, **the** – 5,75,000
  - **Diwali** – 1,25,000, **Festive** – 2,32,500
  - **Bash** – 37,500, **with** – 5,90,000
  - **Biggest** – 47,500, **Discounts** – 1,05,000
- Label $Y = \mathbf{1}$ **for SPAM** and $\mathbf{0}$ **for GENUINE**

Email Spam Filter

- $\bar{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_{32,500} = Enter \\ \vdots \\ v_{37,500} = Bash \\ \vdots \\ v_{47,500} = Biggest \\ \vdots \\ v_{1,05,000} = Discounts \\ \vdots \\ v_{1,25,000} = Diwali \\ \vdots \\ v_{2,32,500} = Festive \\ \vdots \\ v_{5,75,000} = The \\ \vdots \\ v_{5,90,000} = With \\ \vdots \\ v_{6,00,000} \end{bmatrix} = \begin{bmatrix} v_1 = 0 \\ \vdots \\ v_{32,499} = 0 \\ v_{32,500} = 1 \\ v_{32,501} = 0 \\ \vdots \\ v_{37,500} = 1 \\ \vdots \\ v_{47,500} = 1 \\ \vdots \\ v_{1,05,000} = 1 \\ \vdots \\ v_{1,25,000} = 1 \\ \vdots \\ v_{2,32,500} = 1 \\ \vdots \\ v_{5,75,000} = 1 \\ \vdots \\ v_{5,90,000} = 1 \\ \vdots \\ v_{6,00,000} = 0 \end{bmatrix}$

Email Spam Filter – Naïve Bayes Assumption

- Joint probability of $\bar{v}\ \forall\ v_i \in \{0,1\}$ is SPAM or GENUINE $u \in \{0,1\}$ – $\boldsymbol{p(\bar{x} = \bar{v} \mid y = u)}$
  - $P(x_1 = v_1, x_2 = v_2, \ldots, x_{6,00,000} = v_{6,00,000} \mid y = u)$
  - $= P(v_1 \cap v_2 \cap \cdots \cap v_{6,00,000} \mid y = u)$
  - $= P(v_1) \times P(v_2|v_1) \times P(v_3|v_2 \cap v_1) \times \cdots \times P(v_{6,00,000}|v_{5,99,999} \cap v_{5,99,998} \cap \cdots \cap v_2 \cap v_1)$
- **Naïve Bayes independence assumption**
  - **Features $(x_1, x_2 \ldots x_{6,00,000})$ are conditionally independent given the outcome/class $y \in \{0, 1\}$**
- Applying Naïve Bayes assumption and generalising
  - $\boldsymbol{p(v_2|v_1) = p(v_2), p(v_3|v_2 \cap v_1) = p(v_3)}$ etc. from probability axioms
  - $p(\bar{x} = \bar{v} \mid y = u) = P(v_1) \times P(v_1) \times P(v_2) \times \cdots \times P(v_N)$ - Conditional independence
  - $\Rightarrow p(\bar{x} = \bar{v} \mid y = u) = \prod_{j=1}^{N} \boldsymbol{p(x_j = v_j \mid y = u)}$

Naïve Bayes Classifier – Prior Probabilities

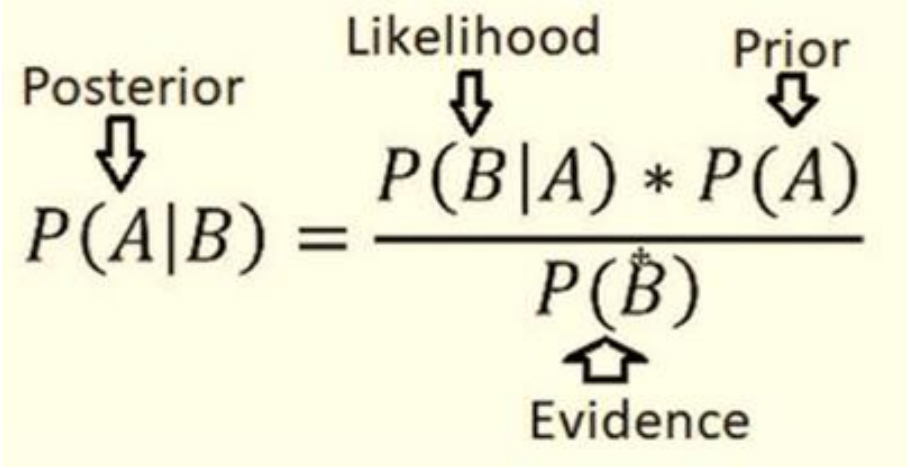
- Consider M training samples - $[\bar{x}(i), y(i)]_M \; i = 1,2, \ldots M$
- M – total emails with some being SPAM and the other GENUINE
- Prior Probability - j^{th} word occurs in SPAM email $(y = 1)$
  - $p(x_j = 1 \mid y = 1) = \frac{Number\ of\ SPAM\ Emails\ with\ j^{th}\ Word}{Total\ Number\ of\ SPAM\ Emails} = \frac{\sum_{i=1}^{M} 1(x_j(i)=1, y(i)=1)}{\sum_{i=1}^{M} 1(y(i)=1)}$
  - $1(x_j(i) = 1, y(i) = 1)$ – indicator function
    - $1(x_j(i) = 1, y(i) = 1)$ = 1 when $x_j(i) = 1\ and\ y(i) = 1$, else 0
  - $1(y(i) = 1)$ – indicator function
    - $1(y(i) = 1)$ = 1 when $y_j(i) = 1$, else 0
- Prior Probability - j^{th} word DOES NOT occur in SPAM email $(y = 0)$
  - $p(x_j = 0 \mid y = 1) = 1 - p(x_j = 1 \mid y = 1)$

Naïve Bayes Classifier – Prior Probabilities

- Prior Probability - j^{th} word occurs in GENUINE email $(y = 0)$
  - $p(x_j = 1 \mid y = 0) = \frac{Number\ of\ GENUINE\ Emails\ with\ j^{th}\ Word}{Total\ Number\ of\ GENUINE\ Emails} = \frac{\sum_{i=1}^{M} 1(x_j(i)=1, y(i)=0)}{\sum_{i=1}^{M} 1(y(i)=0)}$
  - $1(x_j(i) = 1, y(i) = 0)$ – indicator function
    - $1(x_j(i) = 1, y(i) = 0)$ = 1 when $x_j(i) = 1\ and\ y(i) = 0$, else 0
  - $1(y(i) = 0)$ – indicator function
    - $1(y(i) = 0)$ = 1 when $y_j(i) = 0$, else 0
- Prior Probability - j^{th} word DOES NOT occur in GENUINE email $(y = 0)$
  - $p(x_j = 0 \mid y = 0) = 1 - p(x_j = 1 \mid y = 0)$
- Prior Probability – An email being SPAM
  - $p(y = 1) = \frac{Number\ of\ SPAM\ Emails}{Total\ Number\ of\ Emails} = \frac{\sum_{i=1}^{M} 1(y(i)=1)}{M}$
- Prior Probability – An email being GENUINE
  - $p(y = 0) = 1 - p(y = 1)$

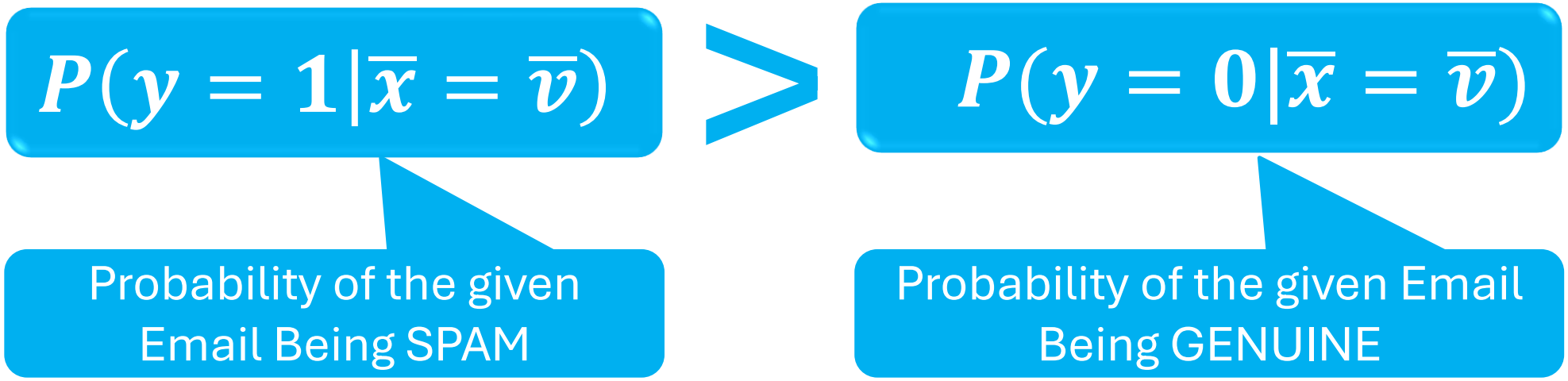
Naïve Bayes Classifier – Posterior Probabilities

- Apply Bayes' Theorem/rule to update the probabilities
- Posterior probability of an email $\bar{v}$ is SPAM
  - $p(y = 1|\bar{x} = \bar{v}) = \frac{p(\bar{x}=\bar{v}|y=1)\times p(y=1)}{p(\bar{x}=\bar{v})}$
- Posterior probability of an email $\bar{v}$ is GENUINE
  - $p(y = 0|\bar{x} = \bar{v}) = \frac{p(\bar{x}=\bar{v}|y=0)\times p(y=0)}{p(\bar{x}=\bar{v})}$



Naïve Bayes Classifier – Classification

- Email is classified as SPAM if:



- Else as GENUINE

Naïve Bayes Classifier – Boundary Conditions

- Email is classified as SPAM if - $\boldsymbol{p(y=1|\bar{x}=\bar{v}) > p(y=0|\bar{x}=\bar{v})}$
- $\Rightarrow \frac{p(\bar{x}=\bar{v}|y=1)\times p(y=1)}{\cancel{p(\bar{x}=\bar{v})}} > \frac{p(\bar{x}=\bar{v}|y=0)\times p(y=0)}{\cancel{p(\bar{x}=\bar{v})}}$
- Conditions
  - SPAM Email
    - $\boldsymbol{p(\bar{x}=\bar{v}|y=1)\times p(y=1) > p(\bar{x}=\bar{v}|y=0)\times p(y=0)}$
    - $\boldsymbol{\Rightarrow \sum_{j=1}^{N} p(x_j=v_j|y=1)\times p(y=1) > \sum_{j=1}^{N} p(x_j=v_j|y=0)\times p(y=0)}$
  - GENUINE Email
    - $\boldsymbol{p(\bar{x}=\bar{v}|y=1)\times p(y=1) \leq p(\bar{x}=\bar{v}|y=0)\times p(y=0)}$
    - $\boldsymbol{\Rightarrow \sum_{j=1}^{N} p(x_j=v_j|y=1)\times p(y=1) \leq \sum_{j=1}^{N} p(x_j=v_j|y=0)\times p(y=0)}$
- Taking the natural logarithm and simplifying
- SPAM Email
  - $\boldsymbol{\sum_{j=1}^{N}\left[ln\left(p(x_j=v_j|y=1)\right)\right] + ln(p(y=1)) > \sum_{j=1}^{N}\left[ln\left(p(x_j=v_j|y=0)\right)\right] + ln(p(y=0))}$
- GENUINE Email
  - $\boldsymbol{\sum_{j=1}^{N}\left[ln\left(p(x_j=v_j|y=1)\right)\right] + ln(p(y=1)) \leq \sum_{j=1}^{N}\left[ln\left(p(x_j=v_j|y=0)\right)\right] + ln(p(y=0))}$

Naïve Bayes Classifier

Feature Vector (Training Data)	Inference (Predicted Value)	Algorithm Name
Continuous - $x \in \mathbb{R}^n$	Continuous – $y \in \mathbb{R}^n$	Linear Regression
Continuous - $x \in \mathbb{R}^n$	Discrete – $y \in \{0,1\}$	Logistic Regression
Discrete - $x \in \{0,1\}$	**Discrete - $y \in \{0,1\}$**	**Naïve Bayes**
Discrete - $x \in \{0,1\}$	Continuous - $y \in \mathbb{R}^n$	Linear Regression with One-Hot Encoding

- Simple yet powerful probabilistic classifier
- Effectiveness
  - Works well – features are **relatively independent or weakly correlated**
    - Examples : Email spam filter, disease diagnosis, sentiment analysis
  - Fails but works effectively - features are **relatively dependent or correlated**
    - Examples : Document classification, image recognition, speech recognition

Zero Frequency Problem

- Suppose an email is received with the word **petrichor**
- **Petrichor** was NOT present in any training emails – SPAM or GENUINE
- The word is found around 4,65,000th position in Oxford English Dictionary
- Prior probabilities of **petrichor**
  - Probability of **petrichor** in SPAM emails
    - $p(x_{4,65,000} = 1 \mid y = 1) = \frac{No.of\ SPAM\ Emails\ with\ the\ word\ \text{petrichor}}{Total\ Number\ of\ SPAM\ Emails} = \frac{\sum_{i=1}^{M} 1(x_{4,65,000}(i)=1, y(i)=1)}{\sum_{i=1}^{M} 1(y(i)=1)} = \frac{0}{\sum_{i=1}^{M} 1(y(i)=1)} = \mathbf{0}$
  - Probability of **petrichor** in GENUINE emails
    - $p(x_{4,65,000} = 1 \mid y = 0) = \frac{Number\ of\ GENUINE\ Emails\ with\ the\ word\ \text{petrichor}}{Total\ Number\ of\ GENUINE\ Emails} = \frac{\sum_{i=1}^{M} 1(x_{4,65,000}(i)=1, y(i)=0)}{\sum_{i=1}^{M} 1(y(i)=0)} = \frac{0}{\sum_{i=1}^{M} 1(y(i)=0)} = \mathbf{0}$
- Posterior Probabilities
  - $\bar{x} \in \mathbb{R}^N \ \forall\ x_i \in \{0,1\}\ u \in \{0,1\}$
  - $\Rightarrow p(\bar{x} = \bar{v} \mid y = u) = P(v_1) \times P(v_1) \times P(v_2) \times \cdots \times P(v_N) = P(v_1) \times P(v_1) \times P(v_2) \times \mathbf{0} \ldots \times P(v_N) = 0$
  - $\Rightarrow p(y = u|\bar{x} = \bar{v}) = \frac{p(\bar{x}=\bar{v}|y=u) \times p(y=u)}{p(\bar{x}=\bar{v})} = \frac{0 \times p(y=u)}{p(\bar{x}=\bar{v})} = \mathbf{0}$

If Naive Bayes encounters a feature value that was not present in the training data, it assigns a zero probability to the entire class, making the prediction impossible

Petrichor : The pleasant, earthy smell that accompanies the first rain after a long period of warm, dry weather

Laplace Smoothing aka Additive Smoothing

- Zero Frequency Problem is mitigated by using Laplace smoothing

$$p(x_j = 1 \mid y = 1) = \frac{1 + \sum_{i=1}^{M} 1(x_{4,65,000}(i) = 1, y(i) = 1)}{2 + \sum_{i=1}^{M} 1(y(i) = 1)}$$

Laplace Smoothing

$$p(x_j = 1 \mid y = 0) = \frac{1 + \sum_{i=1}^{M} 1(x_{4,65,000}(i) = 1, y(i) = 0)}{2 + \sum_{i=1}^{M} 1(y(i) = 0)}$$

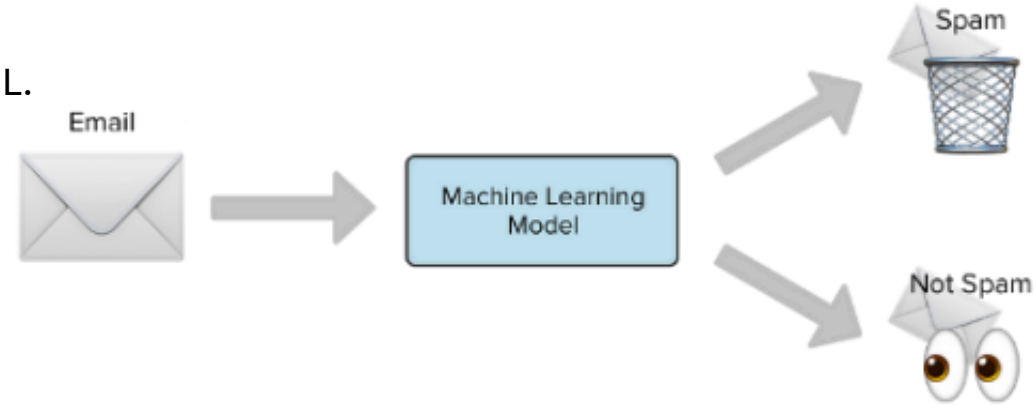
- Equivalent of adding:
  - 2 SPAM emails to training set
    - One containing all words in dictionary
    - One blank
  - 2 GENUINE emails to training set
    - One containing all words in dictionary
    - One blank

Accuracy and Efficiency

- Accuracy
  - Accuracy represents the proportion of correctly classified samples (both true positives and true negatives) out of the total number of samples.
  - $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- Precision
  - Precision measures how many of the samples predicted as positive are actually positive. It focuses on the quality of the positive predictions.
  - $Precision = \frac{TP}{TP+FP}$
- Recall aka Sensitivity aka TPR
  - Recall measures how many of the actual positive samples were correctly classified. It focuses on the ability to find all positive instances
  - $Recall = \frac{TP}{TP+FN}$
- F1-Score
  - The F1-score is the harmonic mean of precision and recall. It is useful when the dataset has an uneven class distribution.
  - $F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$
- Confusion Matrix
  - A confusion matrix is a table used to describe the performance of a classification model
  - It provides insight into the following
    - True Positives (TP): Correctly predicted positive instances
    - True Negatives (TN): Correctly predicted negative instances
    - False Positives (FP) (Type I Error): Incorrectly predicted positive instances
    - False Negatives (FN) (Type II Error): Incorrectly predicted negative instances

Project – Naïve Bayes Spam Classifier

- Objective
  - The objective of this project is to build an email spam filter using the **Naive Bayes (NB) algorithm**, which classifies emails as either spam or genuine (ham)
  - The project will utilize the **SpamAssassin dataset,** a popular dataset for spam filtering tasks from Apache SpamAssasin project
  - You will preprocess the dataset, vectorize the email content, train a Naive Bayes model, and evaluate its performance using standard classification metrics
- Dataset
  - Number of Emails: 10,745 (Spam and Genuine)
  - Spam to Genuine Ratio: Includes a balanced or imbalanced mixture of spam and genuine emails
  - File Format: Emails are stored as plain text files (.txt) in two main directories: spam and genuine
- Features
  - The features for the Naive Bayes model will be derived from the email text content
  - Email Headers: Contain metadata such as subject, sender, and date
  - Email Body: The main content of the email, which may be plain text or HTML.
  - Email Subject: Can be an important indicator of spam emails (e.g., "Free," "Win," etc.)
- Data Source
  - The SpamAssassin Public Corpus
  - Published by SpamAssassin.org, a well-known open-source spam filtering project
  - Data Download Link - SpamAssasin public corpus



Naïve Bayes Spam Classifier - Implementation Steps

1. Load, preprocess, and clean email data from spam and genuine directories:
   - Load email files
   - Preprocess and clean text data
   - Return processed data as a DataFrame with labels
2. Vectorize data using TF-IDF
   - Transform text data into numerical features using the TF-IDF vectorizer
   - Return the transformed feature matrix
3. Train the Naive Bayes model
   - Instantiate and train the Naive Bayes classifier on the training dataset
4. Evaluate the trained model
   - Use the test set to make predictions
   - Print the accuracy, classification report, and confusion matrix for model performance
5. Plot confusion matrix with watermark and classification report with watermark
6. Calculate and save word frequencies
   - Calculate word frequencies across the dataset
   - Save frequencies in a text file (word_frequency.txt) for further analysis or reference
7. Main function
   - Define paths for spam and genuine email directories
   - Prepare data by loading and preprocessing
   - Split data into training and testing sets
   - Vectorize the data
   - Train the Naive Bayes model
   - Evaluate and plot the results with watermarks
   - Calculate and save word frequencies
8. Run the main function if the script is executed directly

Term Frequency – Inverse Document Frequency Vectorization

- For transforming text data into numeric features
- Statistical measure to evaluate the importance of a word within a document relative to the collection corpus
- Term Frequency
  - Measures how often a word appears in a document
  - $TF_{word,doc} = \frac{Count\ of\ word\ in\ the\ doc}{Total\ words\ in\ the\ doc}$
  - Higher TF indicates its importance in the document
- Inverse Document Frequency
  - Measures how unique or rare a word is across collection/corpus
  - $IDF_{word} = \log\left(\frac{Total\ docs}{Number\ of\ docs\ containing\ the\ word}\right)$
  - Higher IDF implies word is rare ⇒ Potentially more informative
- TF-IDF
  - $TF - IDF_{word,doc} = TF_{word,doc} \times IDF_{word}$
- TF-IDF vectorization helps create a feature space
  - Reflecting the **relevance of words** in each document
  - Down-weighting common but less informative words
  - Enables ML model to make better predictions in text classification tasks like spam detection

```
# Vectorize the data
vectorizer
= TfidfVectorizer(stop_words='english',
max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Project Implementation Demo

Interested in building a Gen AI application?

Reach out to venkat@brillium.in

AAML-IPL Brought You in Partnership with:

Brillium Technologies

Sector 7, HSR Layout, Bengaluru 560102, Karnataka, India
Website: www.brillium.in | Email: connect@brillium.in