# Question 1: Binary Classification with Custom Naive Bayes

## Objective:

The objective of this assignment is to develop a binary classification model using the

Naive Bayes algorithm. Students will gain hands-on experience in data loading,

preprocessing, visualization, and model evaluation, applying statistical fundamentals to

create an effective classifier

## Methodology:

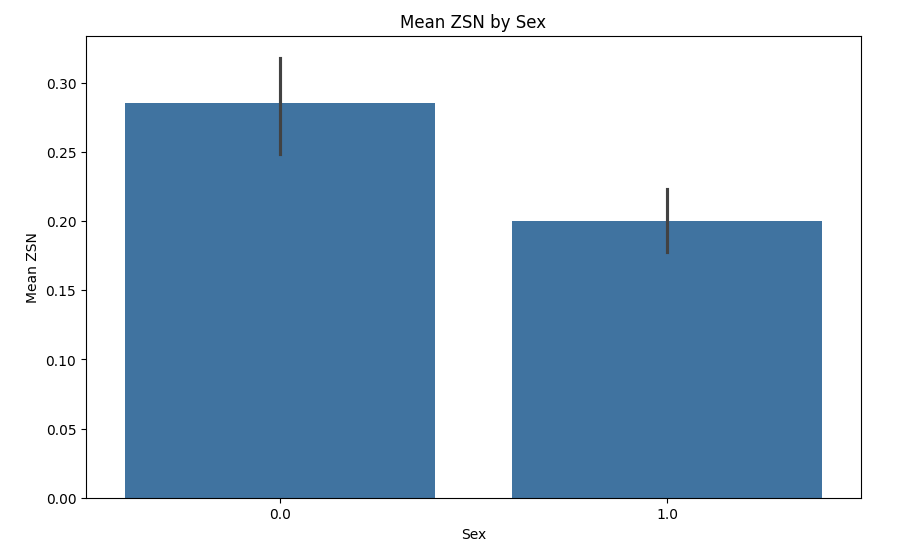
### Data preprocessing:

* The data was first loaded from an external source using a custom function, then the features and targets are concatenated into a single DataFrame for further processing or analysis.
* The SimpleImputer instance with strategy='mean' is used to replace missing values (NaNs) with the mean of each column specified in columns\_to\_impute.
* fit\_transform calculates the mean values for each specified column based on the available data and replaces the NaNs with these means.
* After imputation, the data DataFrame is updated in place with the imputed values.

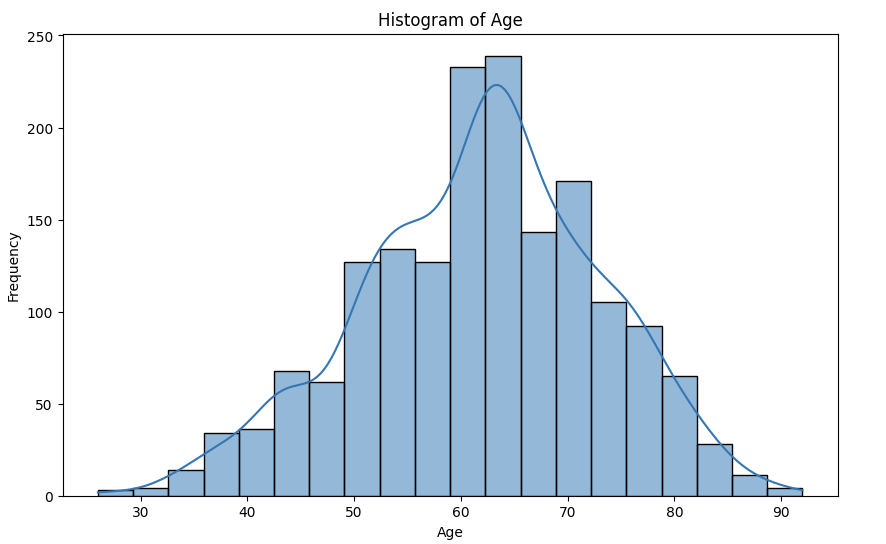


### Data visualization:

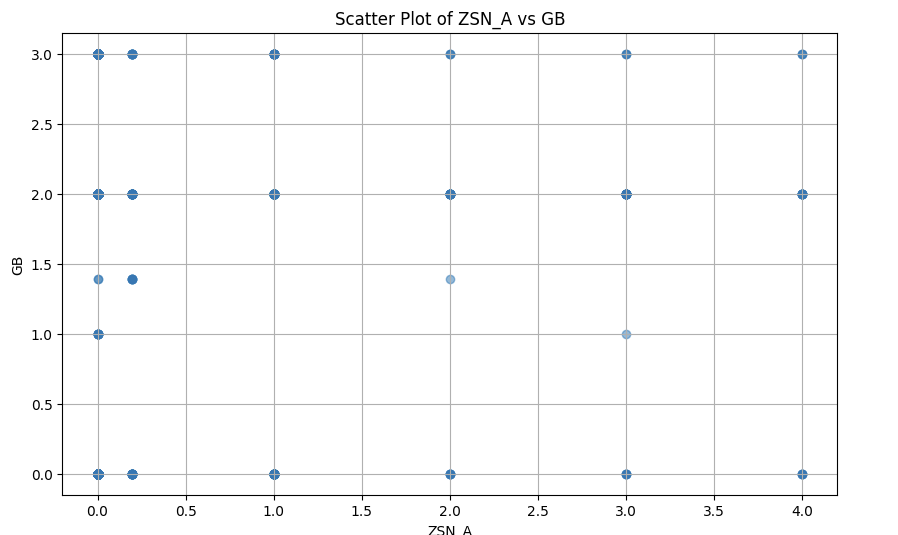
Exploring the distribution of ZSN across sexes reveals that female individuals tend to have higher mean ZSN levels compared to men. This insight provides initial clues on potential associations between sex and ZSN levels.



The histogram of age showcases the distribution of ages within our dataset, indicating peaks around the 60 age group. Understanding the age distribution is essential for contextualizing other demographic and health-related variables.



The scatter plot illustrates the relationship between ZSN\_A and GB(Presence of an essential hypertension )



### Binary Classification Model Development:

For the Classification Model I had three iterations of Model development. Each iteration of the code enhances the modeling process by adding new capabilities, such as threshold adjustment and feature selection, aiming to improve the model's predictive performance and interpretability.

**Approach 1:**

Data Validation: Check if the number of samples in X\_train matches y\_train to ensure consistency in data dimensions.

Data Scaling: Use MinMaxScaler to scale the features (X\_train and X\_test) to a specified range

Model Initialization: Initialize a Gaussian Naive Bayes classifier .

Model Training: Train the Naive Bayes classifier on the scaled training data (X\_train\_scaled).

Prediction: Make predictions on both the scaled training data (X\_train\_scaled) and the scaled test data (X\_test\_scaled).

Evaluation: Calculate accuracy scores (accuracy\_score) to evaluate the performance of the model on both training and test datasets.

**Approach 2:**

Data Scaling: Similar to Code 1, use MinMaxScaler to scale the features (X\_train and X\_test).

Model Initialization & Training: Initialize and train a Gaussian Naive Bayes classifier on the scaled training data (X\_train\_scaled).

Threshold Adjustment: Adjust the classification threshold to control the balance between precision and recall.

Prediction: Make predictions on the test data (X\_test\_scaled) using the adjusted threshold.

Evaluation: Calculate accuracy and precision scores (accuracy\_score, precision\_score) to assess the model's performance with the adjusted threshold.

**Approach 3:**

Data Scaling: Use MinMaxScaler to scale the features (X\_train and X\_test).

Feature Selection: Employ SelectKBest with f\_classif scoring to select the top K most relevant features.

Model Construction: Create a pipeline (Pipeline) that includes feature selection (feature\_selection) followed by a Gaussian Naive Bayes classifier (nb\_classifier\_tuned).

Pipeline Training: Fit the pipeline on the scaled training data (X\_train\_scaled), which includes feature selection and model training.

Threshold Adjustment & Prediction: Apply threshold adjustment to the pipeline's predictions on the test data (X\_test\_scaled).

Evaluation: Calculate accuracy and precision scores (accuracy\_score, precision\_score) based on the adjusted predictions from the pipeline.



#### Considerations:

Feature Selection Impact: Feature selection (using SelectKBest) likely helped in focusing on the most

relevant features for the model, potentially improving performance by reducing noise and overfitting.

Threshold Adjustment: Adjusting the classification threshold influenced the precision of the model's predictions.

## Model Insights:

**Accuracy**: The accuracy of approximately 70.06% indicates that the classifier correctly predicts the class label for about 70% of the instances in the test set.

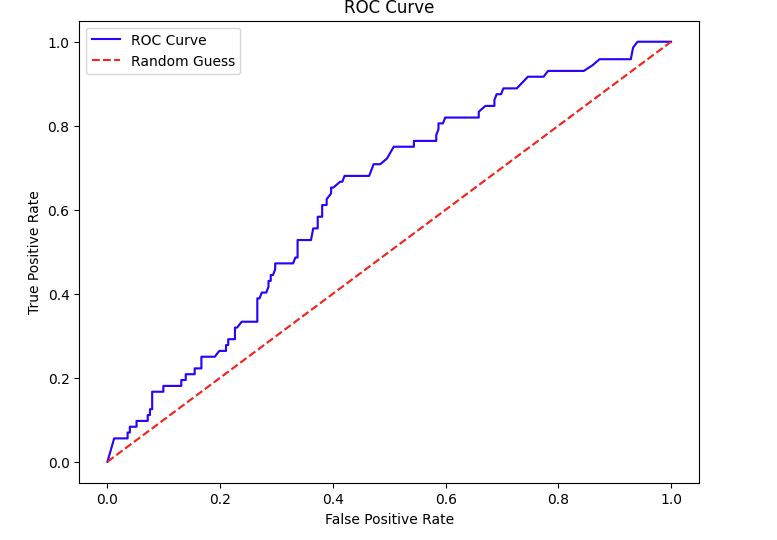
**Precision**: The precision of around 28.07% suggests that when the classifier predicts a positive class label, it is correct only about 28.07% of the time. This metric is particularly important when the cost of false positives is high.

**Recall**: The recall score of about 22.22% indicates that the classifier correctly identifies only about 22.22% of the actual positive instances in the dataset. Recall is crucial in scenarios where missing positive instances (false negatives) is costly.

**F1 Score:** The F1 score, which is the harmonic mean of precision and recall (approximately 24.81% here), provides a balanced measure of the model's performance.

**ROC Curve**

The Receiver Operating Characteristic (ROC) curve in the image represents the performance of a binary classification model at various classification thresholds.

****

**Limitations and Potential Improvements:**

**Hyperparameter Tuning:** Optimize the hyperparameters of the Naive Bayes model. For example, in Gaussian Naive Bayes, adjusting priors or var\_smoothing could lead to improved performance.

**Utilize Different Algorithms:** Experiment with different algorithms beyond Naive Bayes to see if they yield better results given the specific characteristics of the data.

**Cross-Validation:** Use cross-validation techniques to assess model performance more robustly and ensure that the evaluation metrics are not biased by the specific train-test split.

**Error Analysis:** Dive deeper into the types of errors (false positives and false negatives) the model is making to gain insights into areas where the model is struggling and iterate on the features or model choice accordingly.

## Conclusion:

In conclusion, the evaluation of the Naive Bayes classifier reveals insights into its performance and areas for improvement. The classifier achieved an accuracy of approximately 70.06%, indicating its ability to correctly classify instances. However, the precision, recall, and F1 score highlight specific challenges, particularly in accurately identifying positive instances

# Question 2: Emotion Analysis using SVM and K-Means Clustering

## Objective:

Develop an understanding of emotion recognition in text using Support Vector

Machines (SVM) for classification and K-Means clustering for pattern discovery. This

assignment will help you grasp the nuances of supervised and unsupervised learning

techniques in Natural Language Processing (NLP)

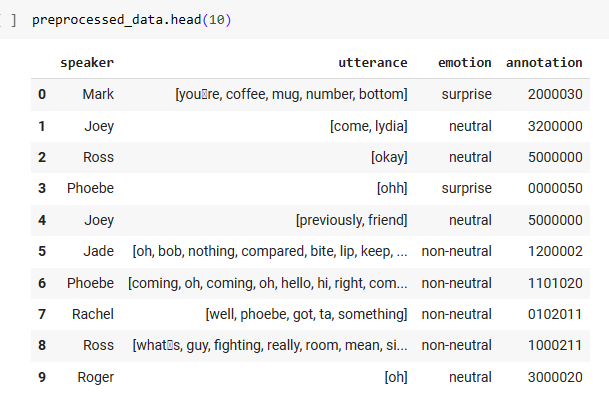
## Methodology:

### Data preprocessing:

For Data preprocessing we used some preprocessing techniques used in NLP like Tokenization, stemming/lemmatization, and stop words removal.

* Tokenization is the process of splitting a text or a sentence into meaningful units, called tokens. These tokens could be words, numbers, punctuation marks, or other elements, depending on the tokenizer used.
* Stemming and lemmatization are techniques used to reduce words to their base or root form. This helps in normalizing words so that variations of the same word are treated as identical during analysis.
* Stop words are common words (e.g., "the", "is", "at", "which") that occur frequently in a text but generally do not contribute much to its overall meaning or context.

Preprocessed data:



# 

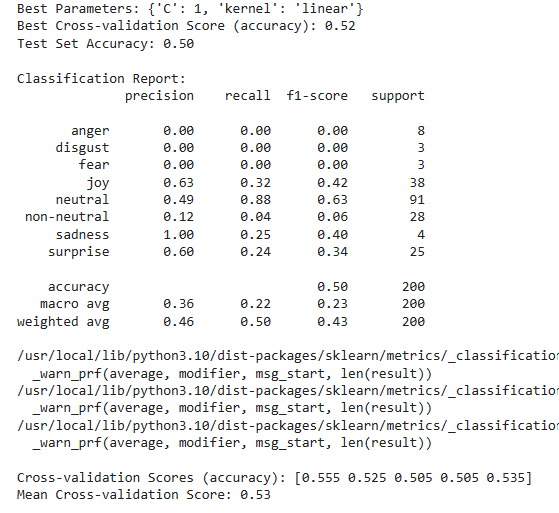
### Use of TF-IDF to convert text into numerical data:

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic used in natural language processing and information retrieval to represent the importance of a term (word) within a document relative to a collection of documents. It converts text data into numerical vectors that can be used as input for machine learning algorithms. Using TF-IDF, text data is transformed into a numerical format that retains important information about the relative importance of terms within documents and across a corpus, making it suitable for various text analysis tasks.

### SVM for Emotion Classification

SVM stands for Support Vector Machine, which is a supervised machine learning algorithm used for classification and regression tasks. It's particularly effective for classification tasks in high-dimensional spaces, making it a popular choice for text classification, image classification, and other complex problems.

#### 



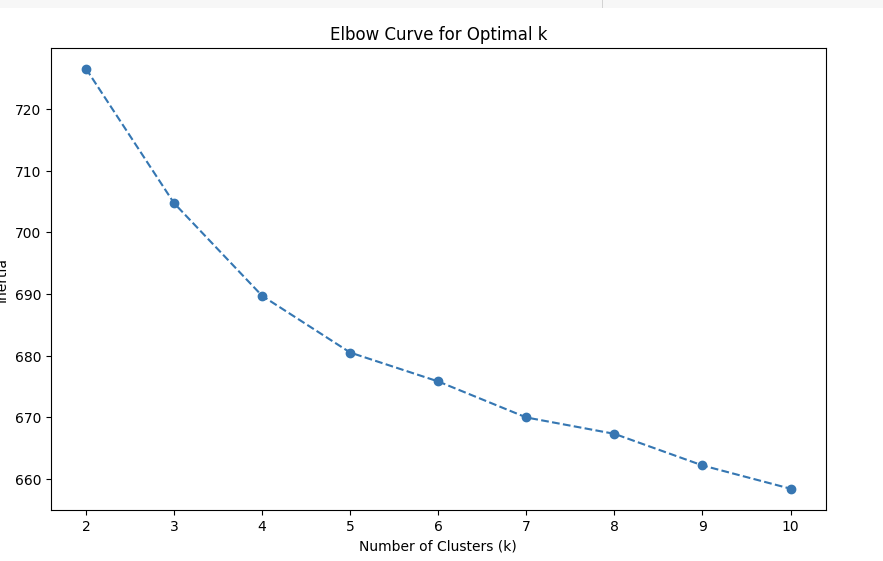
### K-Means Clustering

K-Means is an unsupervised machine learning algorithm used for clustering data into groups based on similarities. The goal of K-Means is to partition the data into K clusters, where each data point belongs to the cluster with the nearest mean .

**Identifying optimal number of clusters with elbow method:**

The Elbow Curve is a plot of the sum of squared distances (inertia) of data points to their nearest cluster centroid against different values of K (number of clusters).

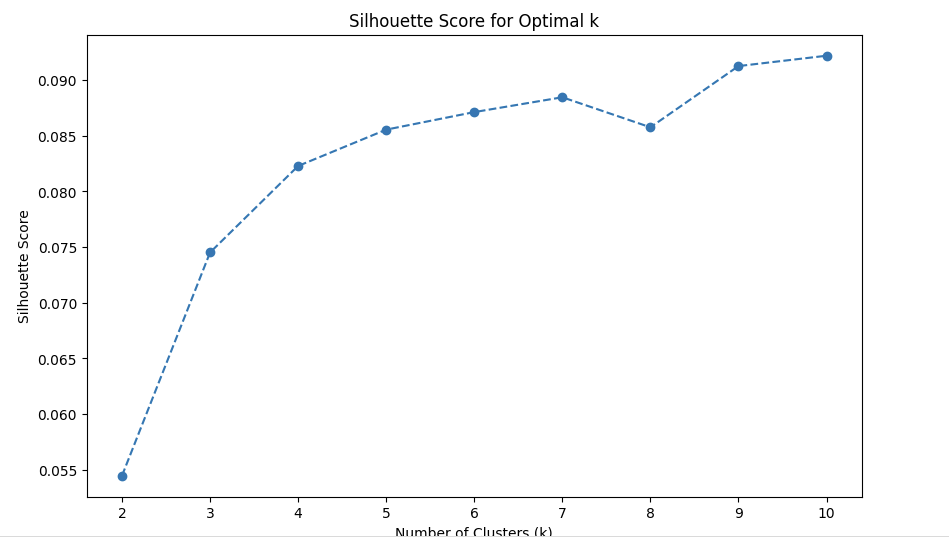
The optimal number of clusters (K) is typically chosen at the point where the inertia reduction significantly slows down, forming an "elbow" in the curve. From the graph we can deduce that inertia reduction significantly slows down at K=5



**Identifying optimal number of clusters with Silhouette Score method:**

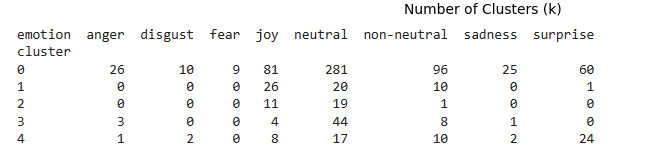
The Silhouette Score is a metric used to evaluate the quality of clusters formed by a clustering algorithm. It measures how well-separated the clusters are and provides insight into the cohesion and separation of data points within each cluster. The Silhouette Score initially increases as k grows, indicating better clustering. It reaches a peak (optimal point) at k=6, suggesting that 6

clusters provide the best separation. Beyond k=6, the score declines, indicating that additional clusters do not improve the clustering significantly.



Thus, after analyzing data using elbow method and silhouette score, we will choose K as 5

**Interpret the clusters to find patterns corresponding to different emotions:**

****

## Model Insights:

### SVM Performance Analysis:

**Best Parameters:**

The SVM model's best hyperparameters are identified as {'C': 1, 'kernel': 'linear'}, suggesting that a linear kernel with a regularization parameter (C) of 1 performed optimally on the training data.

**Cross-validation Score:**

The mean cross-validation accuracy score is 53%, which is relatively low. This indicates that the SVM model might be struggling to generalize well across different folds of the training data. The complexity of the emotion classification task could be a contributing factor.

**Test Set Performance:**

The accuracy on the test set is 50%, which is slightly lower than the cross-validation score. This suggests that the model's performance is consistent but not very high, likely due to the inherent complexity of accurately classifying emotions based on the available features.

**Classification Report:**

The classification report reveals that certain emotion classes like "anger," "disgust," and "fear" have very low precision and recall. This implies challenges in accurately predicting these specific emotions, possibly due to their nuanced and subjective nature in expression.

### K-Means Clustering Analysis:

**Identifying Optimal Number of Clusters (K=5):**

The K-Means clustering analysis suggests that dividing the data into 5 clusters best represents the underlying patterns in the dataset, based on the elbow method.

**Cluster Interpretation:**

Each cluster represents a distinct grouping of emotional expressions:

Cluster 0: Mixed group with dominant neutral and joyful expressions.

Cluster 1: Primarily joyful expressions with some neutral and non-neutral ones.

Cluster 2: Mostly neutral expressions mixed with some joy.

Cluster 3: Predominantly neutral with minor occurrences of joy and anger.

Cluster 4: Represents a diverse mix of multiple emotional states without clear dominance.

### Compare and Contrast:

**Methodology:**

SVM: Supervised learning algorithm used for classification. Trained on labeled data to learn patterns and make predictions.

K-Means: Unsupervised learning algorithm used for clustering. Identifies similarities in data points to group them into clusters based on feature similarity.

**Objective:**

SVM: Classifies emotional expressions into predefined categories (e.g., joy, anger, neutral) based on learned patterns from training data.

K-Means: Identifies inherent structures or patterns within the dataset without predefined labels, grouping data points into clusters based on similarities.

**Output:**

SVM: Predicts emotion labels for new data points based on learned patterns and relationships.

K-Means: Assigns each data point to a cluster, representing similar emotional characteristics within each cluster.

**Task Complexity:**

SVM tackles a specific classification task with predefined emotional classes, aiming for accurate prediction within these categories.

K-Means explores the dataset's underlying structure, identifying natural groupings or clusters based on feature similarity.

**Performance Evaluation:**

SVM's performance is measured by its ability to accurately classify emotions based on learned patterns, evaluated using standard classification metrics.

K-Means' effectiveness is judged by how well it captures inherent groupings or emotional trends within the dataset, typically assessed through clustering metrics and visualization.

### 

### Insights into Emotional Trends:

**SVM:**

Captures detailed emotional nuances by assigning specific labels (e.g., joy, anger) to input data based on learned patterns.

Provides insights into the model's ability to distinguish between different emotional states, highlighting challenges in recognizing certain emotions (e.g., anger, fear) based on available features.

SVM's detailed labeling of emotional nuances can enhance sentiment analysis systems in customer feedback analysis, social media monitoring, and market research. By accurately identifying specific emotions like joy, anger, or fear, businesses can gain deeper insights into customer sentiments and tailor their strategies accordingly.

**K-Means:**

Reveals underlying emotional trends or patterns within the dataset by grouping similar emotional expressions into clusters.

Identifies common emotional themes or groupings (e.g., dominant joy in Cluster 1, mixed expressions in Cluster 4), shedding light on inherent emotional structures in the dataset.

Clustering emotional expressions using K-Means can reveal underlying emotional patterns in patient data or mental health assessments. This information can assist healthcare professionals in better understanding emotional states, identifying trends in emotional well-being, and customizing treatment plans for individuals based on their emotional profiles.

## Conclusion:

Integrating SVM predictions with K-Means clusters can offer a holistic view of emotional expressions, validating SVM's predictions against natural groupings identified by K-Means.

Insights from both methods can guide emotion understanding and inform model improvements, such as refining SVM features or adjusting cluster definitions based on emotional patterns.