# Project Report

On

**“Machine Learning Project for Student Dropout and Academic Success”**

**Course:** Machine Learning

**Course Code:** AIM 511

Under the guidance of

**Prof. Raghuram Bharadwaj**

Submitted by

**Shivam Padaliya (MT2024107)**

**Ojas Hegde (MT2024105)**

**Mohit Gupta (MT2024049)**

# Abstract

This study aims at applying the machine learning approaches in identifying student dropout and their academic achievement which can be classified into several classes such as dropout, average performers and high achievers. The data set that I used in this paper comprises of student's demographics, academic records, and behavioral indicators to create a multi-class Naive classification Bayes model. and The Softmax best Logistic classifier, Regression.

Support Vector Machine Also, (SVM), other was techniques compared like with K-Means were used to for performance clustering evaluation with metrics the such aim as of accuracy, understanding precision, the recall, similarities and and F1 differences score, of it the was student observed clusters. When it comes that SVM classifier produced the best results as the overall F1 score was 0.87, Softmax Logistic Regression with F1 score of 0.82 and Naive Bayes with F1 score of 0.78. The clustering analysis gave additional information about student clusters; however, it cannot be used to make predictions as well as for supervised student models. outcome prediction

thus The has results the show possibility that of SVM being is used suitable in identifying students who are likely to drop out and or underperform in their academic performance and hence intervene early. In the future, the work will be extended to ensemble models and deeper feature engineering to improve the predictive capacity.

# Introduction

It is equally important for educational institutions to be able to predict student dropout and academic success so that early intervention can be made and student success can be improved. In this study, machine learning models are applied to create a multi-class classification model that can classify students into three categories: drop out, average performers and high achievers. The main focus has been given to SVM classifier because it is quite efficient in handling complex data sets especially the multi-class classification problems. In order to assess how well the SVM classifier performs its task, the Bayes model and is SoftMax evaluated Logistic and Regression. compared Also, with in other order popular to models gain such further as understanding Naive of the data, several unsupervised learning methods, namely clustering, are used. All these findings are intended to help future research and practical applications in the area of student performance prediction.

Report on the Data Set The following are the activities that were done:

Data Pre-processing

Data cleaning: The data was cleaned by removing missing values, duplicates and ensuring that the data type of each column was appropriate for the data it contained. Encoding categorical data: Since the data contained categorical variables, these were encoded into numerical values that could be processed by the machine learning algorithms. Feature scaling: Since some features had widely different ranges of values were than on others, the the same features scale.

Choosing the Model The models compared include: Support Vector Machine (SVM): This model uses a hyperplane to create a margin that separates assumptions. the Softmax data Logistic classes. Regression: Naive This Bayes: model It is is similar a to probabilistic logistic classifier regression based but on is independence used for multi-class classification.

Model Assessment Accuracy: The accuracy score measures the percentage of predictions that the model got correct. Precision: It is the ratio of relevant results to the total number of results retrieved by the model for a particular class. Recall: Recall quantifies how often the model will correctly identify positive instances. F1-Score: The harmonic mean of precision and recall that balances the two metrics.

Conclusion Comparison From of the Classifiers results The of results the of study, the it models can are be shown concluded in that the the following SVM table:

Classifier performed well in predicting student performance, making it suitable the different for explore capability. groups integrating The real-time within additional application applications. the predictors of The student and student dataset, experimenting performance use which with prediction of could other in clustering be machine educational provided utilized learning institutions valuable for algorithms can insights personalized to contribute into educational enhance to strategies. the improved Future model's academic research accuracy outcomes, should and student predictive support services, and overall institutional effectiveness.

## ❖ Description Of Dataset

| | Marital status | Application mode | Application order | Course | Daytime/evening attendance | Previous qualification | Nacionality | Mother's qualification | Father's qualification | Mother's occupation |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 8 | 5 | 2 | 1 | 1 | 1 | 13 | 10 | 6 |
| 1 | 1 | 6 | 1 | 11 | 1 | 1 | 1 | 1 | 3 | 4 |
| 2 | 1 | 1 | 5 | 5 | 1 | 1 | 1 | 22 | 27 | 10 |
| 3 | 1 | 8 | 2 | 15 | 1 | 1 | 1 | 23 | 27 | 6 |
| 4 | 2 | 12 | 1 | 3 | 0 | 1 | 1 | 22 | 28 | 10 |

5 rows × 35 columns

| Curricular units 2nd sem (credited) | Curricular units 2nd sem (enrolled) | Curricular units 2nd sem (evaluations) | Curricular units 2nd sem (approved) | Curricular units 2nd sem (grade) | Curricular units 2nd sem (without evaluations) | Unemployment rate | Inflation rate | GDP | Target |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.000000 | 0 | 10.8 | 1.4 | 1.74 | Dropout |
| 0 | 6 | 6 | 6 | 13.666667 | 0 | 13.9 | -0.3 | 0.79 | Graduate |
| 0 | 6 | 0 | 0 | 0.000000 | 0 | 10.8 | 1.4 | 1.74 | Dropout |
| 0 | 6 | 10 | 5 | 12.400000 | 0 | 9.4 | -0.8 | -3.12 | Graduate |
| 0 | 6 | 6 | 6 | 13.000000 | 0 | 13.9 | -0.3 | 0.79 | Graduate |

Dataset Features

Size of Dataset: 4424 rows and 35 columns.

## ❖ **Features and Target:**

- Features: 34 columns (mix of numerical and categorical variables) related to demographic, academic, and economic factors.
- Target Variable: "Target" (categorical), representing three classes:
  - "Graduate" (2209 instances)
  - "Dropout" (1421 instances)
  - "Enrolled" (794 instances)

## ❖ Data Types:

- Numerical Features: 5 columns (e.g., "Curricular units 1st sem (grade)", "GDP").
- Categorical Features: 29 columns (e.g., "Marital status", "Application mode").
- Target: Object type (categorical).

## ❖ Missing Data

- No missing values were detected in any column.

## ❖ Imbalanced Target Variable

- The target variable shows an imbalance, with the "Graduate" class being the majority. The proportions are as follows:
  - Graduate: 49.9%
  - Dropout: 32.1%
  - Enrolled: 18.0%

# ❖ Feature Descriptions

1. Marital Status: Indicates the student's marital status (e.g., single, married).
2. Application Mode: Represents the mode through which the student applied (e.g., online, offline, direct).
3. Application Order: Reflects the preference order of the program during application.
4. Course: The specific academic program the student enrolled in.
5. Daytime Evening Attendance: Indicates whether the student attends classes during the day or evening.
6. Previous Qualification: The academic qualification attained before enrollment.
7. Previous Qualification Grade: Grade achieved in the previous academic qualification.
8. Nationality: The nationality of the student.
9. Mother's Qualification: Educational level of the student\u2019s mother.
10. Father's Qualification: Educational level of the student\u2019s father.
11. Mother's Occupation: Mother\u2019s profession.
12. Father's Occupation: Father\u2019s profession.
13. Displaced: Indicates if the student has been displaced from their hometown.
14. Educational Special Needs: Specifies if the student has any special educational needs.
15. Debtor: Indicates whether the student has unpaid tuition fees.
16. Tuition Fees Up to Date: Whether tuition fees are paid on time.

17. Gender: Student\u2019s gender (e.g., male, female).
18. Scholarship Holder: Indicates if the student is receiving a scholarship.
19. Age at Enrollment: Age of the student during enrollment.
20. Curricular Units 1st Sem (Enrolled): Number of units the student enrolled in during the first semester.
21. Curricular Units 1st Sem (Approved): Number of units the student passed in the first semester.
22. Curricular Units 1st Sem (Grade): Average grade for units taken in the first semester.
23. Curricular Units 2nd Sem (Enrolled): Number of units the student enrolled in during the second semester.
24. Curricular Units 2nd Sem (Approved): Number of units the student passed in the second semester.
25. Curricular Units 2nd Sem (Grade): Average grade for units taken in the second semester.
26. Unemployment Rate: Unemployment rate in the student's region during enrollment.
27. GDP: Gross Domestic Product of the student's country or region during enrollment.
28. Inflation Rate: Inflation rate at the time of enrollment.
29. Target: The outcome classification of the student:
    - Graduate: Successfully completed the program.
    - Dropout: Did not complete the program.
    - Enrolled: Still pursuing the program.

## ❖ Inferences from Features

1. Marital Status: Married students may face additional responsibilities, increasing the likelihood of dropping out.
2. Application Mode: Certain modes might attract higher-performing students or those with specific characteristics.
3. Application Order: A lower order might indicate a less preferred course, potentially correlating with dropout rates.
4. Course: Specific courses may have varying dropout and success rates depending on difficulty or demand.
5. Daytime Evening Attendance: Evening students might have jobs, leading to time constraints impacting academic performance.
6. Previous Qualification: Students with higher prior qualifications are likely to perform better.
7. Previous Qualification Grade: Higher grades in previous education might predict better academic performance.
8. Nationality: International students might face cultural and language barriers affecting their performance.

9. Mother's and Father's Qualifications: Higher parental education levels are often linked to better academic performance.
10. Mother's and Father's Occupations: Certain occupations might correlate with financial stability and educational support.
11. Displaced: Students away from their hometown may face adjustment challenges.
12. Educational Special Needs: May require additional support, impacting dropout and success rates.
13. Debtor: Financial difficulties might correlate with dropouts.
14. Tuition Fees Up to Date: Indicates financial discipline, potentially linked to academic commitment.
15. Gender: Gender disparities in academic performance might exist in certain contexts.
16. Scholarship Holder: Scholarship recipients may show better commitment due to financial support.
17. Age at Enrollment: Non-traditional (older) students might face challenges in adapting to academic environments.
18. Curricular Units (Enrolled/Approved/Grades): Direct indicators of academic performance and persistence.
19. Unemployment Rate: Higher unemployment might correlate with students focusing more on education as a fallback option.
20. GDP: Regions with higher GDPs might offer better educational infrastructure, aiding student success.
21. Inflation Rate: High inflation might indicate economic stress, impacting dropout rates.

## ❖ Data Set Preprocessing

Preprocessing is essential to prepare the dataset for ML models. This particular
chapter explains the transformations we have done on the data set.

### ❖ Data Cleaning

This dataset is nearly clean thanks to the rigorous data preprocessing performed by the contributors of the [data](https://www.mdpi.com/2306-5729/7/11/146). They've addressed anomalies, outliers, and missing values. However, a few minor cleaning steps are still needed. First, I'll modify some column names to make them more consistent and easier to work with later.

Correct a column name that has a typo and replace single quotes with underscores.

Replace white space in the column names with underscore

Remove the parenthesis

I'll change the data types of columns that should be categorical from 'int' to 'category' to ensure that classification models treat these columns as categorical data rather than numerical.

## ❖ Exploratory Data Analysis

There are 34 features in this dataset. I will examine their relationship with the target variable, which is a three-class categorical data. The features that have no association with the label will be the potential variables to be removed from modeling.

Chi-Square Independence Test for Categorical Variables

The Chi-Square independence test will be implemented to check the association between the categorical variables and the dependent variable, with the hypothesis as follows:

H0: the two variables are independent

H1: the two variables are dependent

$\alpha = 0.05$

I will reject the null hypothesis and accept the alternative hypothesis if the p-value is less than 0.05, meaning the two variables are dependent. If the p-value is greater than or equal to 0.05, I fail to reject the null hypothesis, meaning the two variables are independent.

## ❖ Feature Selection and Statistical Analysis

During the analysis, most features exhibited p-values close to zero, suggesting a statistically significant association with the target variable. However, three features—**'Nationality'**, **'International'**, and **'Educational_special_needs'**—had relatively high p-values (0.24, 0.53, and 0.73, respectively). These values indicate a lack of significant association with the target variable. As a result, these features were excluded from further modeling to enhance the performance and interpretability of the predictive model.

## ❖ Accuracy

Accuracy is one of the most common evaluation metrics for classification tasks. It
measures the proportion of correctly classified instances out of the total instances.
It is calculated using the formula:
Accuracy =
Number of Correct Predictions
Total Number of Predictions
Accuracy was used to evaluate the Logistic Regression, Support Vector Machine
(SVM), and Naive Bayes models. While high accuracy indicates good performance,
it may not always be reliable for imbalanced datasets.

## ❖ Confusion Matrix

The confusion matrix provides a detailed breakdown of the model's predictions compared
to the actual outcomes. It includes four components:
• True Positives (TP): Correctly predicted positive cases.
• True Negatives (TN): Correctly predicted negative cases.
• False Positives (FP): Incorrectly predicted positive cases.
• False Negatives (FN): Incorrectly predicted negative cases.
From the confusion matrix, other metrics such as precision, recall, and F1-score
can be derived for more nuanced performance analysis.

### Precision, Recall, and F1-Score

Although not directly used in this project, these metrics are often derived from the confusion matrix:

- **Precision:** Measures the accuracy of positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall:** Measures the model's ability to identify all positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-Score:** The harmonic mean of precision and recall.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

## ❖ Comparison of Metrics Across Models

The performance of the following models was evaluated using accuracy as the primary metric:

1. Logistic Regression
2. Support Vector Machine (SVM)
3. Naive Bayes
4. Clustering

Each model's performance was computed on a test dataset split from the original data. The model with the highest accuracy was considered the best-performing model.

# 1.Clustering

The purpose of clustering is to group the data points in such a way that data points in the same cluster are more similar to each other than to points in other cluster. Similarity is measured using distance metrics like euclidean distance. For this project we have used clustering because we want to see if we can discover patterns in the input data

## ❖ K-means clustering

For this project we have used k-means clustering. K-means is a hard clustering algorithm which means points have to belong to only one cluster, while in soft clustering points can belong to multiple clusters. The reason we chose to use a hard clustering algorithm is because we want clear groupings which are simple and easy to interpret. This makes it easier to analyze the clusters as once we have found the clusters we can take simple statistical analytics like mean and mode.

## The K-Means algorithm:

1. Initialize centroids to random points in the data
2. For each data point:

a. Calculate distance to all centroids
b. Assign data point to nearest centroid

3. For each cluster

. Compute mean of all data points

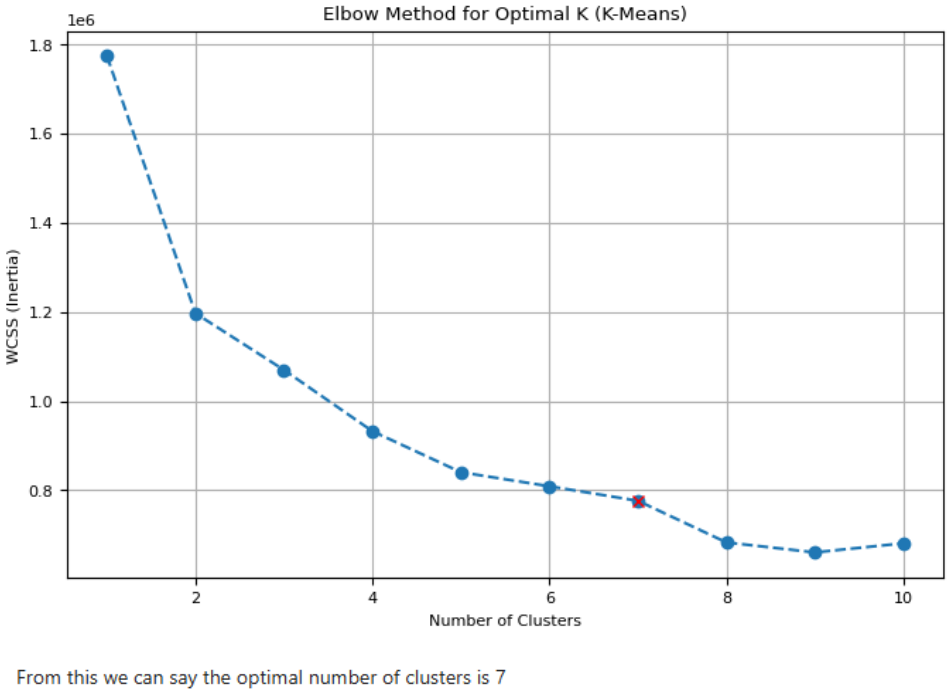
a. Set the new centroid to that mean

4. The algorithms stops when the max number of iterations has been reached or the centroids move by a distance so small that we choose to terminate the algorithm

## ❖ Finding the best ‘K’ in k-means algorithm

We use the elbow method to find the best k. The elbow method is based on a term called inertia or within cluster sum of squares. It is the variance within the clusters, summed up. You could say it measures the *compactness of the clusters*.

We take various values of k and compute inertia. As k increases, inertia decreases. There is a point where the rate at which inertia decreases noticeably slows down, which we call elbow point because it looks like an elbow. We can also find it mathematically by taking second derivative of the inertia with respect to k.




*The elbow method used on our dataset*

## ❖ Result of clustering

In our project we find the optimal k is 7 using the elbow method.
Cluster Sizes:
Cluster
4 944
2 866
3 642
6 591
1 591
5 527
0 188

If we analyze the largest Cluster:
--- Cluster 4 ---

Categorical Variables:
  Marital_status: Mode = 1
  Application_mode: Mode = 1
  Application_order: Mode = 1
  Course: Mode = 12
  Daytime/evening_attendance: Mode = 1
  Previous_qualification: Mode = 1
  Mother_qualification: Mode = 1
  Father_qualification: Mode = 1
  Mother_occupation: Mode = 5
  Father_occupation: Mode = 5
  Displaced: Mode = 1
  Debtor: Mode = 0
  Tuition_fees_up_to_date: Mode = 1
  Gender: Mode = 0
  Scholarship_holder: Mode = 0
  Target_encoded: Mode = 2.0

Numerical Variables:
  Age: Mean = 21.40, Median = 20.00
  GDP: Mean = 0.22, Median = 0.79
  avg_enrolled: Mean = 6.32, Median = 6.00
  avg_approved: Mean = 4.71, Median = 5.00
  avg_grade: Mean = 10.80, Median = 12.38
  avg_without_evaluations: Mean = 0.15, Median = 0.00

This data set was encoded by the providers of the data set so detailed analysis is difficult as we do not know what numerical value is assigned to what category. We can see for example that this cluster is mostly graduated, and the mean gdp of this cluster is higher than mean gdp of whole data set

# 2.Logistic Regression (Softmax)

## ❖ Introduction

Logistic regression is commonly used for binary classification. By using softmax instead of sigmoid function, we can use it for multi class classification.

Implementation of this model uses the gradient descent algorithm to optimize the model parameters.

## ❖ Logistic Regression Algorithm:

**Input Data**:

- Let X be the input data matrix of size m×n, where m is the number of samples and n is the number of features.
- Let y be the true labels, where each label belongs to one of C classes

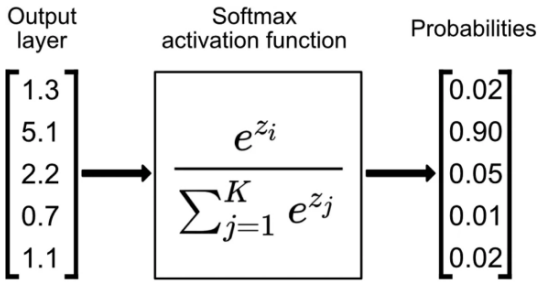
**Model Parameters**:

- The model learns a weight matrix W of size n×C and a bias vector b of size C.

**Compute the Z**:

- For each sample we compute the linear equation z = XW + b

**Apply the Softmax Function**:




- The **softmax function** transforms the z into probabilities for each class:
- The softmax function takes a vector of C real values and outputs a vector of C real values that sum up to one.
- We can consider this the probability of the data point being assigned to that class
- This ensures:
    - The output probabilities for each sample sum to 1.
    - The probabilities are proportional to the exponentiated z.

❖ **Loss Function (Cross-Entropy)**:

- For multiclass classification, the **cross-entropy loss** compares the predicted probabilities of xi with the true class which is a one-hot encoded matrix of the y.

- $$\text{Loss} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{C} y_{ij}\log(y_{\text{pred},ij})$$
- Where yij = 1 if xi belongs to class j or 0 otherwise

❖ **Optimization**:

- Use gradient descent to minimize the loss function
- Parameters are updated according to equations:

W -= learning rate * dW, b -= learning rate * db

❖ **Prediction**:

- For a new input x, compute probabilities using Softmax(XW + b)
- The predicted class is the class with maximum probability

## ❖ Performance

Accuracy - Number of correct predictionsTotal no of predictions
Accuracy = 0.671264367816092 = **67.12%**

Confusion matrix
[[223 43 18]
[ 52 82 25]
[ 35 113 279]]

Confusion matrix[i][j] represents tuple of true class i is predicted as class j

# 3. Support Vector Machine (SVM)

## ➢ Introduction of the Model

- Support Vector Machines are powerful supervised learning models that are designed to achieve the optimal hyperplane that separates two classes with maximum separation.
- Implementation of this model focuses on a linear SVM with hinge loss and uses L2 regularization to overcome overfitting.

## ➢ Core Ideas of the Model

- **Learning Rate (lr):**
  - To ensure stable convergence while iteratively minimizing loss, while updating weights and bias during gradient descent, Learning rate is set as 0.001.
- **Regularization Parameter (lambda param):**
  - It adds a penalty term to prevent overfitting and control the magnitude of the weight vector.
  - It also helps in maintaining the balance between maximizing the margin between two classes and minimizing misclassifications.
- **Optimization Process:**
  - Gradients are calculated, and the weights (w) and bias (b) are updated step by step to reduce the hinge loss.
  - Two cases for updating weights and bias:
    - Condition Met: If $y_i \times (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1$, only the regularization term is updated.
    - Condition Not Met: If $y_i \times (\mathbf{w} \cdot \mathbf{x}_i - b) < 1$, both the regularization and misclassification terms are updated.
- **Hinge Loss:**
  - Hinge loss measures the model's ability to correctly classify points while maximizing the margin.
  - Defined as:

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i \cdot (\mathbf{w}\cdot\mathbf{x}_i - b)\right)$$

- **Prediction Rule:**
  - The sign of the decision boundary ($\mathbf{w}\cdot\mathbf{x} - b$) determines class labels:
    - Positive sign: Class 1.
    - Negative sign: Class -1.

# ➢ Performance

- **Evaluation Metric Calculations:**
  - The predicted values ($y_{\text{pred}}$) are compared to the actual labels ($y_{\text{test}}$) to compute the performance metrics.
  - The formulas for each metric are as follows:
    - **Accuracy:**

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

    - **Precision:**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

    - **Recall:**

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

    - **F1-Score:**

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Model Performance:**
  - After training the model with 1000 iterations, a learning rate of 0.001, and a regularization parameter of 0.01, the SVM achieved the following metrics on the test dataset:
    - **Accuracy:** 26.7%
    - **Auc Score : 27.7 %**
    - **F1-Score:** 7.7%

# 4.Naïve Bayes

## ❖ Naive Bayes Model:

- **Introduction of the Model**
  - Naive Bayes is a fast and simple classification algorithm that uses probability to predict the class of a data point. It assumes that all features are independent, making it efficient and effective for tasks like spam detection and text classification.

  - While it works well with large datasets and independent features, it may struggle with complex data where features are highly dependent. Variants like Gaussian Naive Bayes handle continuous data effectively.

- **Core Ideas of the Model**
  - **Prior Probabilities:**
    - The prior probability for each class is calculated as:

$$P(\text{class}_c) = \frac{\text{Number of Samples in Class } c}{\text{Total Number of Samples}}$$

  - **Likelihood Probabilities:**
    - The likelihood of each feature value given a class is calculated as:

$$P(x_j|\text{class}_c) = \frac{\text{Count of } x_j \text{ in class } c + 1}{\text{Total Samples in class } c + \text{Number of Unique Values in } x_j}$$

    - Laplace smoothing (+1 to each count) ensures that no probability is zero.
  - **Posterior Probabilities:**
    - The posterior probability is proportional to the product of the prior and the likelihoods:

$$P(\text{class}_c|x) \propto P(\text{class}_c) \prod_{j=1}^{n} P(x_j|\text{class}_c)$$

    - The model predicts the class with the highest posterior probability.
  - **Incremental Training:**
    - The dataset is divided into small batches of data, and the model is updated incrementally to handle large datasets.

- **Performance**

**Evaluation Metric Calculations:**

- The predicted values ($y_{pred}$) are compared to the actual labels ($y_{test}$) to compute the performance metrics.
- The formulas for each metric are as follows:
  - **Accuracy:**

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

  - **Precision:**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

  - **Recall:**

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

  - **F1-Score:**

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

## Model Performance

The Naive Bayes model, trained using chunk-based updates, achieved the following metrics on the test dataset:

**Accuracy:**66%