Group 4 - Section 02 (SECP2753)

Class Activity - Classification

KAVIVARTHAN A/L MANNIVANAN
A24CS0093

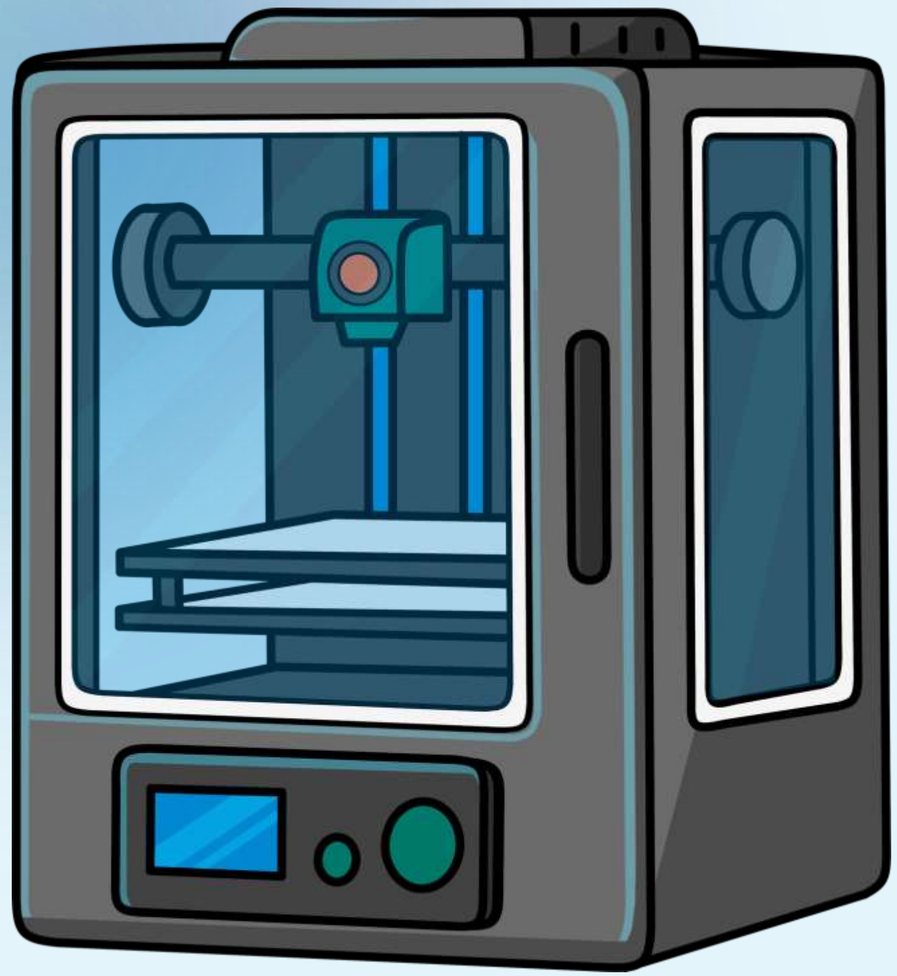
NIVEETHITA A/P PANDIA RAJAN
A24CS0148

SYARAH AQILAH BINTI YA'ACOB
A24CS0195




Part 1

Q1: What is classification in data mining?



Classification in data mining is a supervised learning approach used to assign data points into predefined classes based on their features. By analysing labelled historical data, classification algorithms learn patterns and relationships that enable them to categorize new, unseen data accurately.

https://www.geeksforgeeks.org/machine-learning/classification-in-data-mining/

Q2: Explain the importance and real-world applications of classification in data mining.

Importance:

- Helps organisation make accurate predictions that could improve outcomes and optimise their resources
- Able to identify and manage risks in company
- Helps to automate sorting tasks in large datasets

Real-World Applications:

- Classify medical images and diagnose diseases
- Improves Customer Relationship Management by predicting customer churn, identifying high-value customers, and personalising marketing campaigns
- Detect fraudulent transactions and prevent financial losses
- Classify images into different categories, such as objects, scenes, and actions



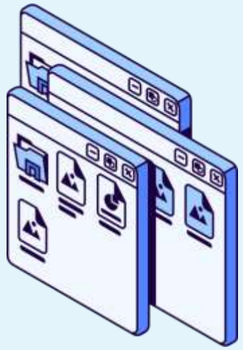
https://www.numberanalytics.com/blog/ultimate-guide-to-classification-in-data-mining-algorithms

Q3: Identify at least TEN (10) classification algorithms. Then, provide a concise explanation with an example for three (3) of them.

Classification Algorithms

1. Linear Regression
2. Logistic Regression
3. Decision Trees
4. SVM
5. Naive Bayes
6. KNN
7. K-Means
8. Random Forest
9. Dimensionality Reduction Algorithms
10. Gradient Boosting Algorithms

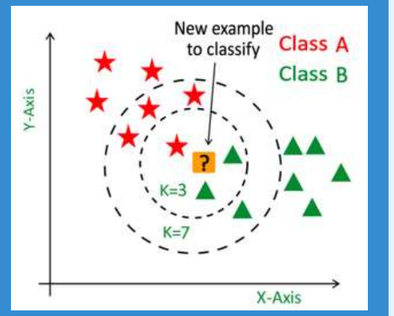




K-nearest neighbor (KNN)

This is a familiar classification technique that is used when there is little or no previous information about the dissemination of data. The technique was made to help in executing discriminant analysis in cases where dependable parametric estimates probability densities are difficult to define.

In the diagram, we need to determine which class the new example should be grouped with. K=3 means the new example will be identified as a Class B member. However, if K=7, the new example will belong to Class A



Linear Regression

These are similar to classical multilayer perception neural networks. In the Support Vector Machine, a predictor variable is called an attribute, and a transformed attribute defines the hyperplane, which we call a feature. Feature selection refers to the work involved in choosing the most appropriate representation. A vector refers to the features that describe a row of predictor values. Fig. 2 shows a support vector machine example.

The yellow broken-line is the ideal hyperplane which separates the two classes with the maximal margin. The other two lines only separate the classes with small margins

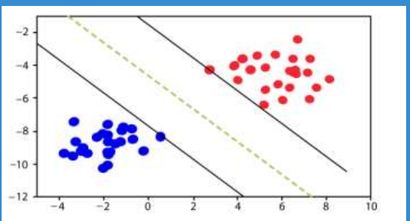


Fig. 2: Support vector machine example

Decision Trees

A decision tree is commonly used for inductive inference. It involves approximating discrete-valued functions for noisy data and capable of learning inconsistent phrases. Kesavaraj and Sukumaran (2013) contended that it is a model that estimates the value of the expected variable due to multiple input variables.

Generally, a decision tree can also be used in classification problems to categorize diseases, equipment malfunction, and also in loan applications. Quinlan (1993) recognized thatID3 or decision tree algorithm could be used to solve some problems in which the final results are changed to C4.5.

Some problems with decision trees are: deciding how intensely the decision tree is developed, selecting the best attribute, handling missing values, and managing differing costs. Another flaw is its unsuitability for small data set 18.

https://www.science-gate.com/IJAAS/Articles/2021/2021-8-2/1021833ijaas202102001.pdf

Q4: Identify and explain common evaluation metrics used in classification in data mining (such as accuracy, precision, recall, F1-score). Include relevant formulas.



ACCURACY

it indicates the proportion of correct predictions made by model out of all of its predictions. However, imbalanced datasets can cause this metrics to be misleading. The advantage is it gives a quick snapshot to evaluate

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

PRECISION

It indicates how many positive predictions made by the model are actually correct. It helps in field where it is risky to have false positive results.

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

RECALL @ SENSITIVITY

It measures how many positive cases are correctly identified. It helps in fields where it is risky to have false negatives more than false positives

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

F1 SCORE

It is the harmonic mean of precision and recall. It helps when there is a need to balance between precision and recall.

$$F1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

https://www.geeksforgeeks.org/machine-learning/metrics-for-machine-learning-model/

Q5: Compare Supervised Learning and Unsupervised Learning with examples.

Aspect	Unsupervised Learning	Supervised Learning
Data Used	Uses unlabeled data without predefined outputs.	Uses labeled data with known input and output values.
How It Works	Discovers hidden patterns and structures in data automatically.	Learns by making predictions and correcting errors using training data.
Main Goal	Gain insights and identify patterns from large datasets.	Predict outcomes or classify new data accurately.

Aspect	Supervised Learning	Unsupervised Learning
Applications	Spam detection, sentiment analysis, weather forecasting, pricing prediction.	Recommendation systems, anomaly detection, customer segmentation, medical imaging.
Advantages	Usually more accurate because the model learns from labeled examples.	Can discover hidden relationships and patterns without labeled data.
Drawbacks	Training can be time-consuming and requires expert labeling.	Results may be inaccurate without human validation.

https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning

Q6: Differentiate between binary classification and multi-class classification. Provide ONE (1) example for each.

	BINARY CLASSIFICATION	MULTI-CLASS CLASSIFICATION
DATA REPRESENTATION	target variable is a 1D array with 0 or 1 (or -1 and 1) representing the two possible classes	target variable is still a 1D array, but it contains integer values representing multiple classes
ALGORITHM ADAPTATION	algorithms like logistic regression can be used directly without changes	some algorithms like logistic regression need an adaptation to handle multiple classes
LOSS FUNCTION	often use the binary cross-entropy loss	uses categorical cross-entropy loss



https://www.datasciencebyexample.com/2023/06/09/binary-classification-vs-multi-class-classification-vs-multi-label-classification/#google_vignette

Q6: Differentiate between binary classification and multi-class classification. Provide ONE (1) example for each.

BINARY CLASSIFICATION

Wisconsin Breast Cancer dataset to predict if a tumor is malignant or benign

```
# Load the breast cancer dataset
data = load_breast_cancer()
X = data.data
y = data.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)

# Train the logistic regression model
binary_model = LogisticRegression(max_iter=1000)
binary_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = binary_model.predict(X_test)

# Calculate the accuracy and print the report
binary_accuracy = accuracy_score(y_test, y_pred)
print("Binary Classification Accuracy:",
binary_accuracy)
print(classification_report(y_test, y_pred))
```

MULTI-CLASS CLASSIFICATION

MNIST digits dataset where each instance is a handwritten digit (0-9)

```
# Load the digits dataset
data = load_digits()
X = data.data
y = data.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)

# Train the logistic regression model
multi_class_model = LogisticRegression(max_iter=1000,
multi_class="ovr")
multi_class_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = multi_class_model.predict(X_test)

# Calculate the accuracy and print the report
multi_class_accuracy = accuracy_score(y_test, y_pred)
print("Multi-Class Classification Accuracy:",
multi_class_accuracy)
print(classification_report(y_test, y_pred))
```

https://www.datasciencebyexample.com/2023/06/09/binary-classification-vs-multi-class-classification-vs-multi-label-classification/#google_vignette

Q7: Discuss common challenges encountered in classification tasks.

- **Class Imbalance:** When one or more classes have significantly fewer examples than others, the model can be biased towards the majority class. Techniques such as oversampling (duplicating instances of the minority class), undersampling (removing instances of the majority class), and cost-sensitive learning (assigning higher misclassification costs to the minority class) can mitigate this issue.
- **Noise and Outliers:** Noisy data and outliers can distort the decision boundary and degrade model accuracy. Robust algorithms and data preprocessing techniques (e.g., outlier removal, smoothing) are crucial for handling these issues.
- **Overfitting:** Overfitting occurs when the model learns the training data too well, including its noise, and performs poorly on unseen data. Regularization techniques (e.g., L1 and L2 regularization), cross-validation, and early stopping are essential for preventing overfitting.
- **Underfitting:** Underfitting occurs when the model is too simple to capture the underlying patterns in the data, leading to poor performance on both training and unseen data. Using more complex models, feature engineering, or reducing regularization can address underfitting.
- **High Dimensionality:** Datasets with a large number of features can lead to the 'curse of dimensionality,' where the performance of many algorithms degrades. Feature selection and dimensionality reduction techniques (e.g., Principal Component Analysis (PCA), feature importance ranking) are employed to address this.
- **Missing Data:** Missing values in the dataset can cause problems with many classification algorithms. Imputation methods (e.g., replacing missing values with the mean or median) and algorithms robust to missing data (e.g., some tree-based methods) are used to handle this challenge.



https://www.clrn.org/what-is-classification-problem-in-machine-learning/

Q8: What is a confusion matrix and how is it used to evaluate classification models? Provide a labeled example

- It is a table used to measure how well a classification model is performing
- Compares the predictions made by the model with the actual results and shows where the model was right or wrong
- Shows where the model confuses one class for another
- It works for both binary classification and multi-class classification
- Helps you understand where the model is making mistakes so you can improve it
- It breaks down the predictions into four categories, which are true positive(TP), true negative(TN), false positive (FP), and false negative(FN)
- Confusion Matrix also helps to calculate key measures (accuracy, precision and recall) to give a better idea of performance especially when the data is imbalanced

Labeled Example: A hospital uses an AI model to predict whether patients have a disease

	Predicted: Has Disease	**Predicted: Healthy**
Actual: Has Disease	TP = 40	FN = 10
Actual: Healthy	FP = 5	TN = 45

TP= 40 means 40 sick pa tients were correctly detected
FN = 10 means 10 sick patients were wrongly predicted healthy
FP = 5 means 5 healthy patients were wrongly predicted as sick
TN = 45 means 45 healthy patients were correctly detected as healthy

https://www.geeksforgeeks.org/machine-learning/confusion-matrix-machine-learning/

Q9: Explain the concept of overfitting and underfitting in classification models. How can each issue be addressed?

Overfitting

Overfitting occurs when a machine learning model fits too closely to the training data and learns irrelevant details or “noise” in the dataset. As a result, the model performs very well on training data but fails to make accurate predictions on new or unseen data. Overfitting usually happens when the model is too complex or trained for too long. A common way to address overfitting is by using early stopping, reducing model complexity, or using techniques such as cross-validation to evaluate model performance.

Underfitting

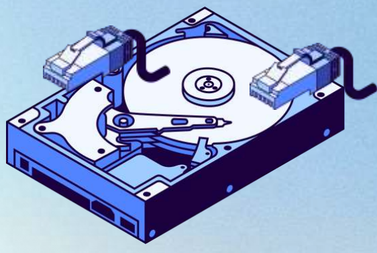
Underfitting occurs when the model has not been trained long enough or when the input variables are too simple to identify meaningful relationships in the data. An underfitted model cannot capture the dominant trend of the dataset and therefore performs poorly on both training and testing data. Underfitting can be addressed by increasing training time, adding more relevant features, or using a more complex model.





https://www.ibm.com/think/topics/overfitting

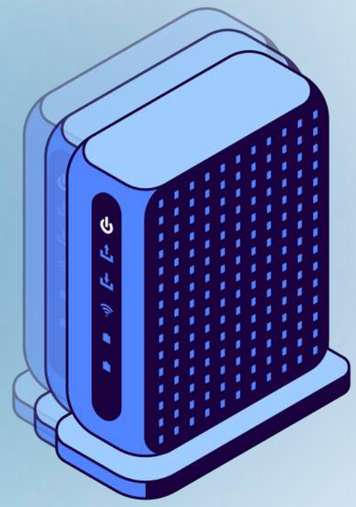
Q10: Describe the role of cross-validation in evaluating the performance of a classification algorithm.



- Cross-validation is a technique used to evaluate the performance of a classification algorithm more reliably
- Instead of testing the model only once, the dataset is divided into multiple parts and the model is tested several times
- The most common approach used is k-fold cross validation where the data is split into k equal parts then trained on k-1 of these folds and tested on the remaining fold
- The training process repeats k times so that each fold adopts the role of test set exactly once
- The results from all iterations are then averaged to provide a more comprehensive and reliable performance metric.

- Cross-validation helps evaluate how well the model generalizes to unseen data and reduces the risk of overfitting
- Without cross validation, the evaluation may depend too heavily on a single train test split, causing the model to appear better or worse than it actually is.
- It plays a role in preventing issues like model overfitting while using limited data efficiently
- It helps evaluation to be more stable and trustworthy since the model is tested on different subset
- Other approaches include Leave-One-Out Cross-Validation, Stratified Cross-Validation, Repeated K-Fold Cross-Validation, and the Holdout Method

https://www.geeksforgeeks.org/machine-learning/cross-validation-machine-learning/



Q11: How can feature selection improve classification model performance? Mention at least two methods.

How Feature Selection Improve Performance?

- Reduced overfitting: Overfitting happens when a model cannot generalize past its training data. Removing redundant features decreases overfitting and makes a model better able to generalize to new data.
- Shorter training times: By focusing on a smaller subset of features, algorithms take less time to train. Model creators can test, validate and deploy their models faster with a smaller set of selected features.
- Lower compute costs: A smaller dataset made of the best features makes for simpler predictive models that occupy less storage space. Their computational requirements are lower than those of more complex models.
- Greater interpretability: Explainable AI is focused on creating models that humans can understand. As models grow more complex, it becomes increasingly difficult to interpret their results. Simpler models are easier to monitor and explain.

What are the methods used?

1. Wrapper methods

Wrapper methods train the machine learning algorithm with various subsets of features, adding or removing features and testing the results at each iteration. The goal of all the wrapper methods is to find the feature set that leads to optimal model performance.
Wrapper methods that test all possible feature combinations are known as greedy algorithms. Their search for the overall best feature set is compute-intensive and time-consuming, and so is best for datasets with smaller feature spaces.

Wrapper methods include:

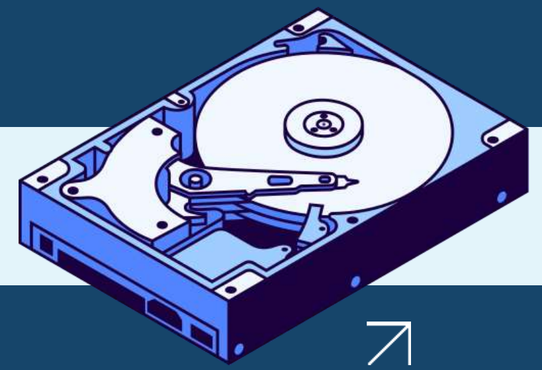
- **Forward selection:** Starts with an empty feature set and gradually adds new features until the optimal set is found. Model selection takes place when the algorithm's performance fails to improve after any specific iteration.
- **Backward selection:** Trains a model with all the original features and iteratively removes the least important feature from the feature set.
- **Exhaustive feature selection:** Tests every possible combination of features to find the overall best one by optimizing a specified performance metric. A logistic regression model that uses exhaustive feature selection tests every possible combination of every possible number of features.

2. Filter methods

Filter methods are a group of feature selection techniques that are solely concerned with the data itself and do not directly consider model performance optimization. Input variables are assessed independently against the target variable to determine which has the highest correlation. Methods that test features one by one are known as univariate feature selection methods.

some common filter methods are:

- **Information gain:** Measures how important the presence or absence of a feature is in determining the target variable by the degree of entropy reduction.
- **Mutual information:** Assesses the dependence between variables by measuring the information obtained about one through the other.
- **Chi-square test:** Assesses the relationship between two categorical variables by comparing observed to expected values.



https://www.ibm.com/think/topics/feature-selection


Part 2

Decision Tree

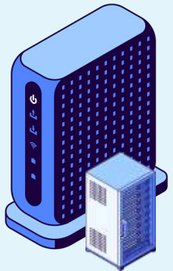
BASIC CONCEPT

Decision trees is one of the most intuitive and interpretable learning algorithms. The decision and the possible consequence is modelled as a tree structure, where each internal node represents a decision based on a feature, each branch represents the outcome of that decision, and each leaf node represents a final prediction. This mirrors how humans break down complex decisions into series of simpler questions.

HOW IT WORKS

1. Evaluate all possible splits across all features
2. For each potential split, calculate impurity (Gini impurity or Information Gain)
3. Select the split that provides the greatest improvement (lowest impurity or highest information gain)
4. Repeat the process for every node, building the tree from top to bottom
5. Stop when stopping criteria met (max depth reached / node achieves perfect classification / no remaining attributes / no samples left)

ADVANTAGES	DISADVANTAGES
Provides transparent view of decision-making process, easy to understand and explain predictions	Tend to overfit especially when trees grow deep
Not sensitive to feature scaling → requires minimal preprocessing	Can be unstable — small changes in training data can lead to dramatically different tree structures
Non-parametric — makes no assumptions about underlying data distribution	Greedy splitting approach may miss global optimal solution
Can handle both classification and regression	Can be biased towards high cardinality features (features with many unique values)



Decision Tree

SUITABLE TYPES OF DATA

- Mixed (numerical + categorical)
- Requires no feature scaling
- Small to medium datasets
- Large datasets
- Missing values

REAL-WORLD APPLICATION EXAMPLE

1. **Banking: Credit Scoring and Loan Approval**

Banks use decision trees to automate the initial assessment of personal loan applications. [1]

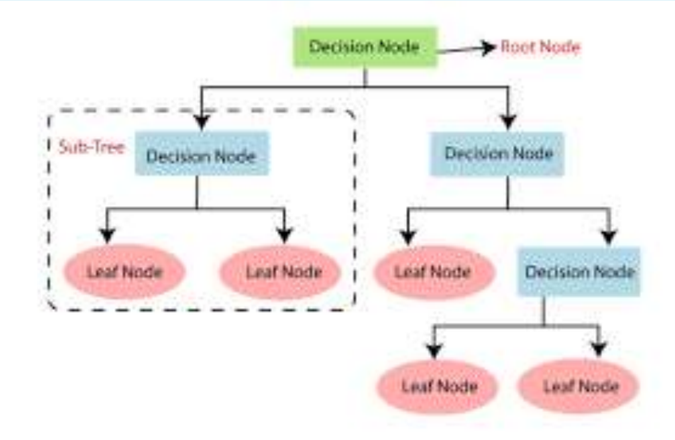
- **Root Node:** Does the applicant have a credit score above 700?
- **Internal Nodes:** Factors like annual income, employment history, and existing debt.
- **Leaf Nodes:** The final decision—"**Approve**" or "**Reject**". [1, 2, 3, 4, 5]

1. R**etail: Purchase Prediction**

Retailers predict customer behavior to optimize sales and inventory. [1, 2]

- **Scenario:** Predicting if a customer will buy a specific product based on their **age** and **income**.
- **Decision Logic:** If a customer is over age 30 and has a high income, they are classified as "**Likely to Buy**". [1, 2, 3]

SIMPLE ILLUSTRATION OR DIAGRAM



Naive Bayes

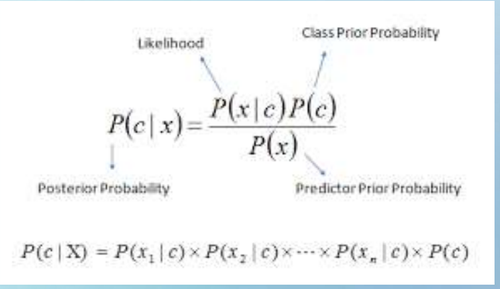
BASIC CONCEPT

Naïve Bayes is a probabilistic learning algorithm that applies Bayes' theorem with the following 'naïve' assumptions: (1) All features are conditionally independent given the class value. (2) All features are equally important. The intuition behind Naïve Bayes is straightforward: given a set of features, what is the probability that this instance belongs to each class

HOW IT WORKS

1. Convert data into a frequency table
2. Create a likelihood table by finding the probabilities
3. Use the Naïve Bayesian equation to calculate posterior probability for each class
4. The class with the highest posterior probability is the prediction outcome

ADVANTAGES	DISADVANTAGES
Most notable strength is speed and efficiency — training only requires calculating class priors and feature distributions	Ignores correlation between features — works poorly when strong correlations exist between features across classes
Works well even when data is scarce	Can struggle with continuous features that don't follow assumed Gaussian distributiontree structures
Highly scalable — can handle high-dimensional data effectively	Sensitive to feature preprocessing, especially with high cardinality categorical features
Simple to implement and understand	The "naïve" independence assumption is rarely true in real-world data



Decision Tree

SUITABLE TYPES OF DATA

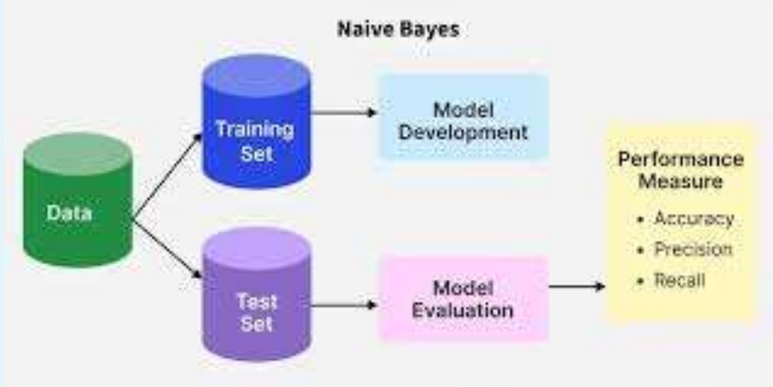
- Text/document classification
- High-dimensional data
- Small datasets
- Categorical features
- Numerical features that are normally distributed

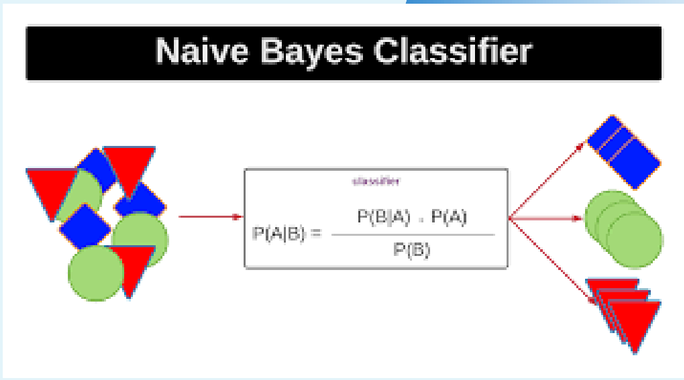
REAL-WORLD APPLICATION EXAMPLE

Spam Filtering

- **Goal:** Classify incoming emails as "Spam" or "Not Spam" (Ham).
- **Data:** Historical emails with identified labels (Training Data).
- **Mechanism:** The algorithm builds a model identifying that words like "Free," "Offer," "Urgent," or "Click" appear more frequently in spam emails.
- **Application:** When a new email arrives, the algorithm calculates the probability of it being spam given the words it contains and flags it if the probability exceeds a threshold.
- **Why It's Used:** It is fast, efficient, handles high-dimensional data well, and performs well for text classification. [1, 2, 3, 4, 5]

SIMPLE ILLUSTRATION OR DIAGRAM





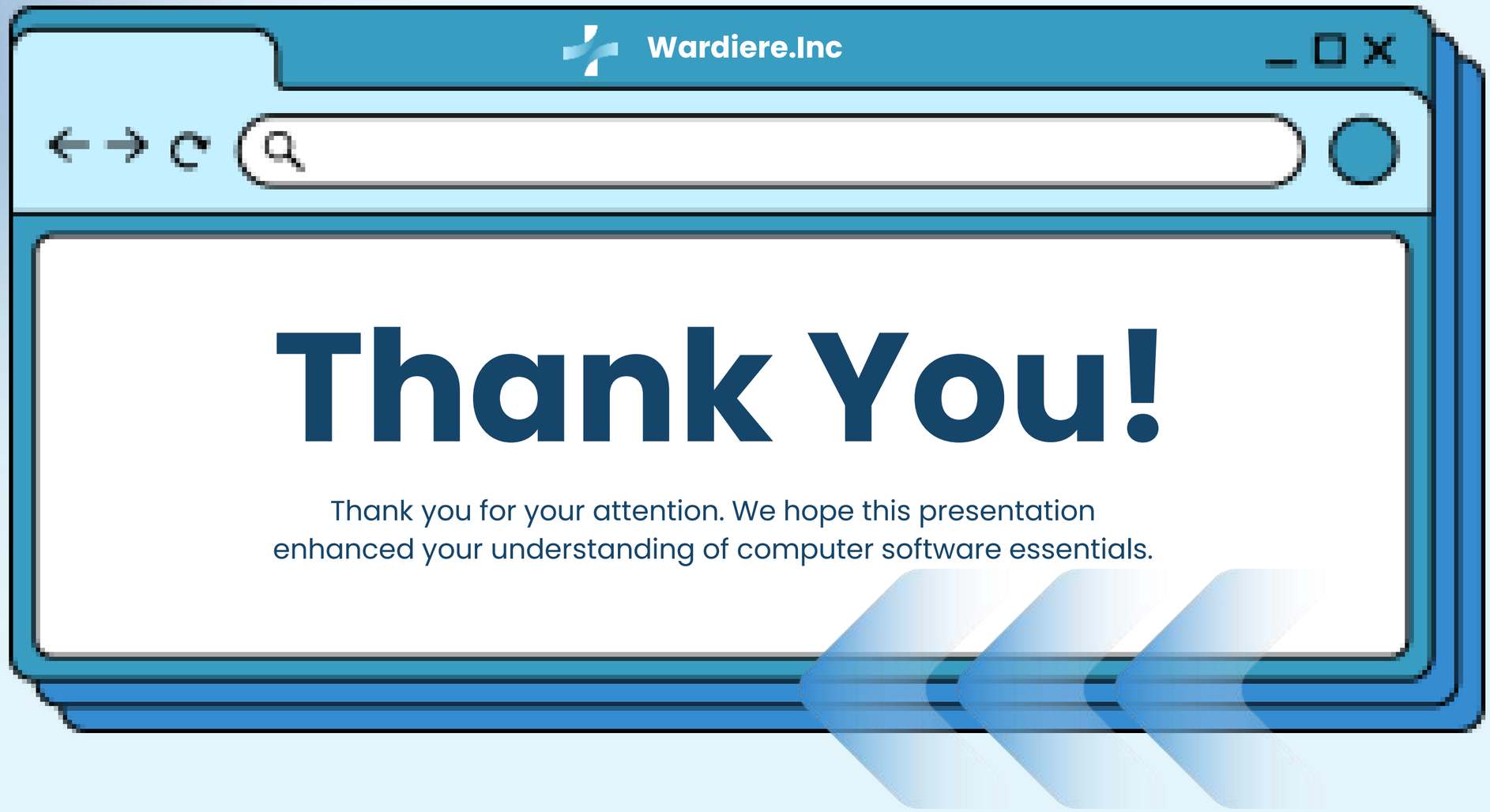
References

Rosidi, N. (2024, August 29). Demystifying decision trees for the real world. KDnuggets. https://www.kdnuggets.com/demystifying-decision-trees-for-the-real-world

What is a decision tree? types, use cases & examples. LatentView Analytics. (2026, April 15). https://www.latentview.com/glossary/decision-tree/

E, V. (2025, September 23). Naive Bayes algorithm: Origins, real-world applications, and case studies. DEV Community. https://dev.to/vamshi_e_eebe5a6287a27142/naive-bayes-algorithm-origins-real-world-applications-and-case-studies-4dhd

Cadili, R. (2024, November 25). An introduction to the Naive Bayes classifier and how to use it. Medium. https://medium.com/low-code-for-advanced-data-science/an-introduction-to-the-naive-bayes-classifier-and-how-to-use-it-bba78826ef8f



Thank You!

Thank you for your attention. We hope this presentation enhanced your understanding of computer software essentials.