

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.Doi Number

Machine Learning Approaches to Milk Quality Assessment: Decision Trees Versus SVM

Ersan Ivanda Putra¹, Elisabet Lumban Tobing², Eunike Sinurat³,
Caroline Alexandra Santoso⁴, Evan Hendraloka⁵

^{1,2,3,4,5}Multimedia Nusantara University

Corresponding author:

Ersan Ivanda Putra (ersan.ivanda@student.umn.ac.id).

Elisabet Lumban Tobing (elisabet.lumban@student.umn.ac.id).

Eunike Sinurat (eunike.sinurat@student.umn.ac.id).

Caroline Alexandra Santoso (caroline.alexandra.santoso@student.umn.ac.id)

Evan Hendraloka (evan.hendraloka@student.umn.ac.id)

This work was supported in part by Universitas Multimedia Nusantara.

ABSTRACT The introduction outlines the growing significance of machine learning in food quality assessment, particularly in predicting milk quality, which is crucial for consumer satisfaction and food safety standards. It highlights the specific context of dairy farming in the Jabung area of Malang District and the importance of predictive modeling for improving dairy production processes. The study compares the performance of support vector machines (SVM) and decision trees in classifying milk quality based on key parameters such as pH, temperature, taste, odor, fat content, turbidity, and color. By evaluating the efficacy of these algorithms and conducting a comparative analysis, the research aims to provide insights into their suitability for milk quality prediction tasks. The ultimate goal is to contribute to the literature on machine learning applications in food quality assessment and offer practical guidance for enhancing dairy production and delivering high-quality milk products to consumers.

INDEX TERMS *Milk Quality, Sustainable, Compares, pH, SVM, Food Quality*

I. INTRODUCTION

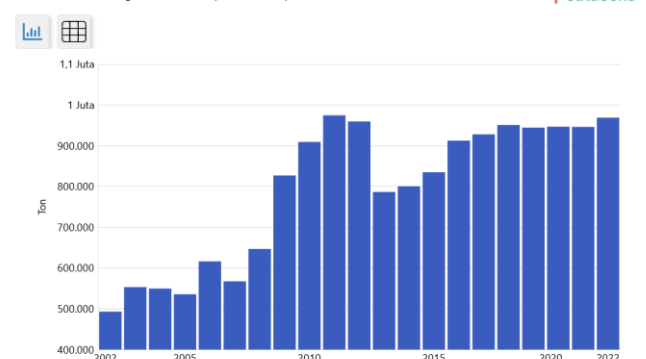
In recent years, the application of machine learning techniques has gained significant traction in various domains, including food quality assessment. One such critical area is the prediction of milk quality, which plays a pivotal role in ensuring consumer satisfaction and food safety standards. With the advancements in technology and the availability of comprehensive datasets, researchers have leveraged machine learning algorithms to develop predictive models for assessing milk quality [1].

The Livestock Service Office of Malang District indicates the Jabung area as a dairy cattle center in the eastern and southeastern regions of Malang, so efforts to develop dairy farming in the area must continue. Koperasi Agro Niaga (KAN) is one of the dairy cooperatives in Jabung district. As of July 2013, KAN Jabung had 1,600 farmers and a monthly milk production capacity of 1,108,504 liters or 29,560 kg/day. Because of this production capacity, KAN Jabung is classified as a large cooperative because it has a milk

production capacity of 20,000-40,000 kg/day (Yusdja, 2005) [1].

The classification of cow's milk quality is made by adapting the discipline in the field of computer science, namely data mining.

Produksi Susu Segar Indonesia (2002-2022)



Data mining is the process of finding meaningful relationships or patterns by examining archived data sets using pattern recognition techniques such as statistical and mathematical methods [2].

Picture.1 Bar Chart Produksi Susu Segar Indonesia (2002-2022)

In the graph above produces milk production from year to year in Indonesia from 2002 to 2022. It can be seen that the resulting graph from 2002 to 2022 is quite visible to increase as well. and the graph illustrates that Indonesia is able to produce fresh milk with good quality standards in accordance with production operations.

Freshness and quality of cow's milk reduce the nutrients in cow's milk. Several factors affect the quality of fresh cow's milk, such as the breed of dairy cow, feeding, feeding system, milking frequency, milking method, seasonal changes and cow's milk age is an equally important factor in terms of milk quality, because bacterial contamination of cow's milk begins after the milk leaves the cow's udder and the number of bacteria increases during longer lactation periods. Cow's milk producers often add water to their milk so that the milk can be sold at a lower price. Adding water to milk will weaken the purity and freshness of milk, as water does not add nutrients to milk. To avoid product fraud, the freshness of milk must be determined. Many ordinary people have a simple way to determine the freshness of cow's milk. A simple way that people often do is to smell cow's milk, but this method is still not completely accurate and it is difficult to detect whether the milk has been mixed with flavorings and colorings. In addition to smelling, an easy way to do this is by pouring milk and comparing the viscosity of cow's milk with pure cow's milk, but this method is very inefficient. Therefore, technology is needed that can detect the quality and freshness of cow's milk, so that we can know which cow's milk is nutritious and suitable for consumption [3].

This study focuses on comparing the performance of SVM and decision tree models in predicting milk quality based on a dataset comprising seven key parameters: pH, temperature, taste, odor, fat content, turbidity, and color. These parameters are known to influence the overall grade or quality of milk, making them essential factors in predictive analysis. The primary objective of this research is to evaluate the efficacy of SVM and decision tree algorithms in classifying milk into different quality grades based on the provided parameters. By conducting a comparative analysis, we aim to identify the strengths and limitations of each approach, thereby providing valuable insights into the suitability of these models for milk quality prediction tasks.

Through this comparative study, we seek to contribute to the existing body of literature on machine learning applications in food quality assessment and provide practical guidance for researchers and practitioners in selecting appropriate modeling techniques for milk quality prediction. Additionally, the findings of this study hold implications for

improving dairy production processes and ensuring the delivery of high-quality milk products to consumers.

Milk, being a staple and essential component of numerous diets worldwide, demands stringent quality control measures to maintain its nutritional integrity and safety [4]. Traditional methods of quality assessment often rely on manual inspection and chemical analysis, which can be time-consuming, labor-intensive, and prone to subjectivity. In contrast, machine learning-based approaches offer a more efficient and objective means of evaluating milk quality by harnessing the power of data-driven models.

The parameters included in our dataset—pH, temperature, taste, odor, fat content, turbidity, and color—are recognized as fundamental indicators of milk quality. pH levels, for instance, can influence microbial growth and enzymatic activity, while variations in temperature can impact the stability and shelf life of dairy products. Similarly, attributes such as taste, odor, fat content, turbidity, and color contribute to sensory attributes and overall consumer acceptability.

Support vector machines (SVM) and decision trees have emerged as prominent tools in the field of machine learning for classification tasks, each offering unique advantages and trade-offs [5]. SVM excel in handling high-dimensional data and capturing complex decision boundaries, making them well-suited for datasets with nonlinear relationships. Decision trees, on the other hand, offer interpretability and ease of understanding, making them particularly appealing for scenarios where model transparency is desired.

By conducting a comparative analysis of SVM and decision tree models in the context of milk quality prediction, we aim to address several research questions:

1. How do SVM and decision tree algorithms perform in classifying milk quality grades based on the provided parameters?
2. Which algorithm demonstrates superior predictive accuracy and robustness in differentiating between quality categories?
3. What insights can be gleaned from the comparison of SVM and decision tree models regarding the importance of individual parameters in determining milk quality?
4. How can the findings of this study inform stakeholders in the dairy industry about the most effective machine learning approaches for quality control and assurance?

Through a systematic evaluation of these questions, this research endeavors to advance our understanding of the applicability and effectiveness of machine learning techniques in the domain of milk quality assessment. Moreover, by shedding light on the comparative performance of SVM and decision tree models, this study

aims to facilitate informed decision-making among researchers and practitioners seeking to implement predictive analytics for enhancing milk production and quality management processes [6]. In recent years, the dairy industry has faced increasing pressure to meet stringent quality standards while also adapting to evolving consumer preferences and regulatory requirements. The ability to accurately predict milk quality is paramount for ensuring compliance with these standards and maintaining consumer trust in dairy products [7]. While traditional quality control measures remain essential, the integration of machine learning techniques offers a promising avenue for enhancing the efficiency and accuracy of milk quality assessment.

The parameters encompassed within our dataset represent multifaceted aspects of milk composition and characteristics, each contributing uniquely to the overall quality profile [8]. pH serves as a crucial indicator of acidity or alkalinity, affecting enzymatic activity and microbial proliferation. Temperature fluctuations can influence bacterial growth rates and alter the physical properties of milk, such as viscosity and solubility. Taste and odor, being sensory attributes, directly impact consumer perception and acceptability. Fat content contributes to the richness and mouthfeel of dairy products, while turbidity and color reflect visual aesthetics and cleanliness. Support vector machines (SVM) and decision trees offer complementary approaches to classification tasks, with distinct advantages and limitations [9]. SVMs excel in discerning complex decision boundaries by transforming the input data into a higher-dimensional space, effectively capturing intricate relationships between variables. However, SVMs may be computationally intensive and sensitive to parameter tuning, necessitating careful optimization for optimal performance. Decision trees, conversely, offer a transparent and intuitive representation of decision-making processes, making them particularly appealing for interpretability and knowledge extraction [10]. Yet, decision trees may be prone to overfitting, especially in the presence of noisy or high-dimensional data. By undertaking a comparative analysis of SVM and decision tree models within the context of milk quality prediction, our study endeavors to provide valuable insights into the efficacy and practical implications of these machine learning approaches. Specifically, we seek to evaluate the performance of SVM and decision tree algorithms in accurately categorizing milk samples into distinct quality grades based on the provided parameters. Additionally, we aim to elucidate the relative importance of individual features in determining milk quality and discern any discernible patterns or correlations within the dataset. Furthermore, by elucidating the strengths and limitations of SVM and decision tree models in the domain of milk quality assessment, this research aims to empower stakeholders in the dairy industry with actionable knowledge for optimizing quality control processes, enhancing product consistency, and mitigating risks associated with quality deviations [11].

Ultimately, our findings hold the potential to catalyze advancements in dairy production practices and contribute to the overarching goal of ensuring the provision of safe, nutritious, and high-quality milk products to consumers worldwide.

1.2 PROBLEM STATEMENT

1. How does the performance of Support Vector Machines (SVM) and decision tree models compare in predicting milk quality?
2. How to determine milk with good quality?
3. Why are SVM and decision tree chosen as models to predict milk quality?
5. What are the implications of the findings for the dairy industry and consumers?

1.3 SOLUTION STATEMENT

1. Conduct a comprehensive performance evaluation of the SVM and decision tree models using relevant evaluation metrics such as accuracy, precision, recall and F1-score. Compare their
2. performance to determine which model is better at predicting milk quality.
3. Define milk quality standards based on key parameters that affect quality, such as pH, temperature, taste, aroma, fat content, turbidity, and color. Then, use a machine learning model (SVM or decision tree) to classify the milk based on these standards. Milk that meets or exceeds these standards can be considered good quality.
4. SVM and decision tree were chosen because they are both effective models in handling classification problems and can perform well even in the case of complex and non-linear data. SVM is able to find the optimal hyperplane that maximizes the margin between different classes, while decision tree divides the feature space into different regions based on simple decision rules.
5. The use of machine learning technology in milk quality assessment can improve efficiency and objectivity compared to traditional methods such as manual inspection and chemical analysis.

II. METHODOLOGY

The method used in this scientific writing is a literature review. The databases used are Kaggle and Google Scholar. Research methodology steps that can be taken in preparing and working on a project, starting from problem identification to system evaluation. That is, knowing the problem of milk quality based on the background explanation and problem formulation. Explore the literature by reading journals, reports on milk quality in the health world, and related empirical studies to gain a deeper understanding of health quality issues and risk factors [12]. Collect the data we get on the website with the name 'milknew.csv' which contains the columns pH, Temperature, Taste, Odor, Fat, Turbidity, Color, Grade.

After the data is collected, the next step is to analyze the data using statistical and machine learning techniques to identify patterns and relationships between various variables that affect milk quality. The results of this analysis are then interpreted to provide recommendations and solutions that can be implemented to improve milk quality. System evaluation is carried out by comparing the analysis results with applicable quality standards and validating the model used.

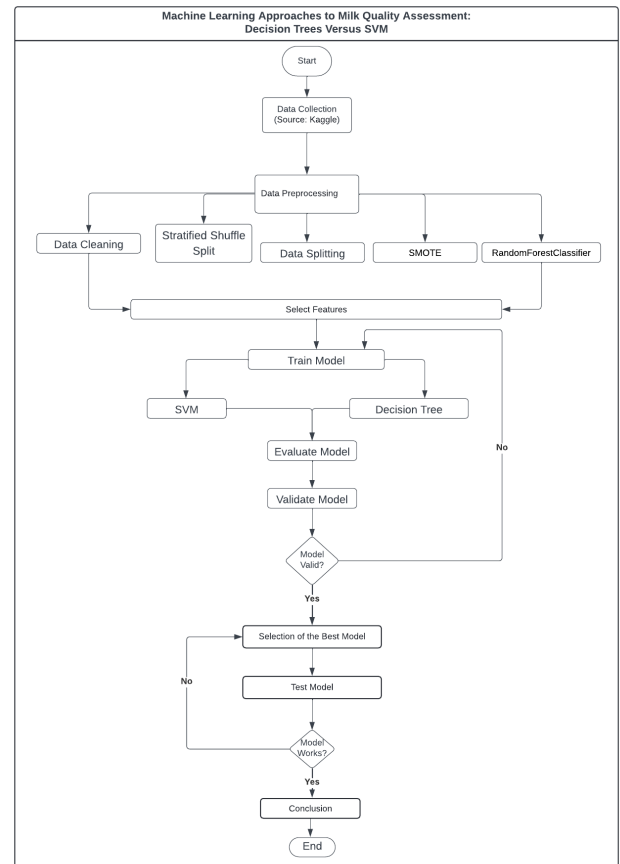
2.1. Flowchart

This flowchart outlines the process of using machine learning approaches to assess milk quality, specifically comparing Decision Trees and Support Vector Machines (SVM). Below is a detailed step-by-step explanation of each component in the flowchart:

1. **Start:** This is the initial step where the process begins.
2. **Data Collection (Source: Kaggle):** The first major task is to gather the dataset required for the analysis. In this case, the data source is Kaggle, a well-known platform for datasets and data science competitions.
3. **Data Preprocessing:** Once the data is collected, it needs to be prepared for analysis. This includes several sub-steps:
 - **Data Cleaning:** This involves removing or correcting any errors, handling missing values, and ensuring the data is in a suitable format for analysis.
 - **Stratified Shuffle Split:** This technique is used to split the data into training and testing sets while preserving the distribution of the target variable. It's particularly useful for classification tasks to ensure each split is representative of the overall dataset.
 - **Data Splitting:** This is the actual process of dividing the dataset into training and testing subsets.

Training data is used to train the model, and testing data is used to evaluate its performance.

- **SMOTE (Synthetic Minority Over-sampling Technique):** This is used to address class imbalance by generating synthetic samples for the minority class, ensuring that the model has enough data to learn from all classes.
- **RandomForestClassifier:** A method sometimes used to rank the importance of different features in the dataset, which can help in selecting the most relevant features for model training.



Picture 2. Flowchart

4. **Select Features:** After preprocessing, relevant features are selected based on their importance and relevance to the target variable.
5. **Train Model:** With the data ready and features selected, two types of models are trained:
 - **SVM (Support Vector Machine):** A supervised learning model used for classification and regression tasks, which finds the hyperplane that best separates the classes in the feature space.
 - **Decision Tree:** A model that splits the data into branches to make decisions based on the feature values, which is easy to understand and interpret.

6. **Evaluate Model:** The trained models are evaluated on the test data to assess their performance. Evaluation metrics might include accuracy, precision, recall, F1 score, etc.
7. **Validate Model:** Validation involves checking the models to ensure they perform well on unseen data and are not overfitting. This step might involve cross-validation or other techniques to verify the model's generalizability.
8. **Model Valid?:** A decision point where the validity of the models is assessed. If the models are valid, the process continues; otherwise, adjustments are made, and the models are retrained.
9. **Selection of the Best Model:** Once the models are validated, the best-performing model is selected based on predefined criteria like accuracy, robustness, and other performance metrics.
10. **Test Model:** The selected model undergoes final testing on a separate test set to ensure it works as expected and delivers consistent results.
11. **Model Works?:** Another decision point to check if the model performs satisfactorily. If not, further adjustments and retraining may be required.
12. **Conclusion:** If the model works as intended, the process concludes, and the results are documented.
13. **End:** The process ends with the final conclusion, summarizing the outcomes of the analysis and model performance.

This flowchart outlines a comprehensive and methodical approach to evaluate milk quality using machine learning. It demonstrates the steps from data collection to model training, evaluation, validation, and final testing. By comparing Decision Trees and SVM, the flowchart ensures a rigorous analysis to determine the best model for assessing milk quality. Each step is designed to ensure data quality, proper model training, and robust evaluation, culminating in a reliable conclusion about the effectiveness of the models used.

1) Dataset

Research datasets from

Dataset: Kaggle

We used the data to analyze milk quality. We analyze milk quality based on the grade of the milk.

2) Preprocessing

The data preprocessing stage is the initial process of analyzing data. This process aims to correct errors in the data to facilitate the future research process.

3) Data Model Development

4) Validation Model

- Split Counting

This process divides the dataset into two: training data and test data [6]. This process is an important part of training and testing data.

- Correlation Checking

Simple correlation is a statistical method used to measure the strength of the relationship between two variables and to determine the form of the relationship between them which is quantitative in nature. The strength of the relationship between the two variables in question is that the two variables have a weak relationship, either close or not close. While the form of meaningful relationship between two variables is in the form of positive linear correlation or negative linear correlation which includes the technique of measuring association [4].

- Decision Tree Implemented

Decision tree is one of the methods that is quite easy for humans to interpret. A decision tree is a prediction model using a tree structure or hierarchical structure. The concept of a decision tree is to transform data into a decision tree and decision rules.

- SVM Implemented

The Support Vector Machine (SVM) algorithm itself is an algorithm that tries to find the maximum hyperplane, the hyperplane is a function that can separate two classes. In this process, SVM maximizes the margin or distance between the training pattern and the decision boundary [6].

5) Testing Data

- Generalize

The ability of a machine learning model to accurately predict previously seen data. The results will capture general patterns from the training data and show the model can apply to new data well

- Check Data

Checking the data that has been obtained and which has been analyzed using the methods we use.

III. RESULT AND DISCUSSION

3.1 Preprocessing

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.6	35	1	0	1	0	254	high
1	6.6	36	0	1	0	1	253	high
2	8.5	70	1	1	1	1	246	low
3	9.5	34	1	1	0	1	255	low
4	6.6	37	0	0	0	0	255	medium
...
1054	6.7	45	1	1	0	0	247	medium
1055	6.7	38	1	0	1	0	255	high
1056	3.0	40	1	1	1	1	255	low
1057	6.8	43	1	0	1	0	250	high
1058	8.6	55	0	1	1	1	255	low

1059 rows x 8 columns

Picture 3. Initial Preprocessing Result

In the picture above is the output result in the initial preprocessing stage that we did. This stage is a mandatory stage that is reading the data that we chose as this research. This involves loading the dataset into a suitable programming environment, such as Python or R, and performing an initial inspection of the data. During this inspection, we check for the presence of any missing values, assess the data types of each column, and perform basic descriptive statistics to understand the distribution and central tendencies of the data. These preliminary steps are crucial as they set the foundation for more detailed data cleaning and transformation processes that follow.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   pH          1059 non-null   float64
1   Temperature 1059 non-null   int64
2   Taste       1059 non-null   int64
3   Odor        1059 non-null   int64
4   Fat         1059 non-null   int64
5   Turbidity   1059 non-null   int64
6   Colour      1059 non-null   int64
7   Grade       1059 non-null   object
dtypes: float64(1), int64(6), object(1)
memory usage: 66.3+ KB
```

Picture 4. Columns of Data

In the picture above is the output result in the initial preprocessing stage that we did. This stage is a mandatory stage that is reading the data that we chose for this research. The dataset, as loaded into a pandas DataFrame, contains 1059 entries with 8 columns. Below is a detailed description of each column:

1. pH: This column contains 1059 non-null values with a data type of float64, indicating it represents the pH levels of the milk samples.

2. Temperature: This column also has 1059 non-null values with a data type of int64, representing the temperature at which the milk was tested.

3. Taste: This column contains 1059 non-null values of int64 type, indicating a numerical representation of the taste attribute.

4. Odor: This column has 1059 non-null int64 values, representing the odor characteristic of the milk.

5. Fat: This column includes 1059 non-null values with an int64 type, representing the fat content in the milk samples.

6. Turbidity: This column has 1059 non-null int64 values, indicating the turbidity level of the milk.

7. Colour: This column contains 1059 non-null values of int64 type, representing the color attribute of the milk samples.

8. Grade: This column has 1059 non-null values with an object data type, indicating the categorical classification or grade of the milk.

3.2 View Missing Values

```
<boud method DataFrame.sum of
0   False  False  False  False  False  False  False  False
1   False  False  False  False  False  False  False  False
2   False  False  False  False  False  False  False  False
3   False  False  False  False  False  False  False  False
4   False  False  False  False  False  False  False  False
...   ...   ...   ...   ...   ...   ...   ...
1054  False  False  False  False  False  False  False  False
1055  False  False  False  False  False  False  False  False
1056  False  False  False  False  False  False  False  False
1057  False  False  False  False  False  False  False  False
1058  False  False  False  False  False  False  False  False

[1059 rows x 8 columns]>
```

Picture 5. Missing Values

In the picture above is the result of displaying Missing Values. This is so that when doing the analysis there are no problems. In the picture above is the output result at the initial preprocessing stage that we did. This stage is a mandatory stage, namely reading the data we chose for this research. The dataset loaded into the pandas DataFrame contains 1059 entries with 8 columns. Below is a detailed description of each column:

1. pH : This column contains 1059 non-null values with a data type of float64, indicating it represents the pH levels of the milk samples.

2. Temperature : This column also has 1059 non-null values with a data type of int64, representing the temperature at which the milk was tested.

3. Taste : This column contains 1059 non-null values of int64 type, indicating a numerical representation of the taste attribute.

4. Odor: This column has 1059 non-null int64 values, representing the odor characteristic of the milk.

5. Fat : This column includes 1059 non-null values with an int64 type, representing the fat content in the milk samples.

6. Turbidity : This column has 1059 non-null int64 values, indicating the turbidity level of the milk.

7. Colour : This column contains 1059 non-null values of int64 type, representing the color attribute of the milk samples.

8. Grade : This column has 1059 non-null values with an object data type, indicating the categorical classification or grade of the milk.

The dataset is fully populated with no missing values across any of the columns. The memory usage of the DataFrame is approximately 66.3 KB. This initial inspection helps in understanding the structure and composition of the data, allowing us to plan further steps for detailed data cleaning and transformation effectively.

3.3 Stratified Shuffle Split

<p>Original dataset distribution: Counter({'low': 429, 'medium': 374, 'high': 256})</p> <p>Training set distribution: Counter({'low': 343, 'medium': 299, 'high': 205})</p> <p>Testing set distribution: Counter({'low': 86, 'medium': 75, 'high': 51})</p>

Table 1. Stratified Shuffle Split

The provided data distributions represent how your original dataset, training set, and testing set are divided among three categories: 'low', 'medium', and 'high'. Here's a breakdown and explanation of these distributions:

Original Dataset Distribution

- Low: 429 instances
- Medium: 374 instances
- High: 256 instances

This is the initial count of instances in each category before any splitting into training and testing sets. It shows the overall balance of your dataset.

Training Set Distribution

- Low: 343 instances
- Medium: 299 instances
- High: 205 instances

The training set is a subset of the original dataset used to train your model. This distribution shows that the majority class is 'low', followed by 'medium', and then 'high'.

Testing Set Distribution

- Low: 86 instances
- Medium: 75 instances
- High: 51 instances

The testing set is another subset of the original dataset used to evaluate the performance of your model. It also maintains the same order of majority to minority classes as the original dataset.

Explanation of Distribution

1. Consistency in Distribution: The proportion of 'low', 'medium', and 'high' instances is roughly maintained across the training and testing sets. This suggests that the splitting process was likely stratified. Stratified splitting ensures that each subset has a similar distribution of classes to the original dataset, which is crucial for maintaining the representativeness of the data.

2. Training vs. Testing Set Size: The training set is larger than the testing set. Typically, a common practice is to allocate around 70-80% of the data for training and 20-30% for testing. In this case:

- Training set: $343 + 299 + 205 = 847$ instances
- Testing set: $86 + 75 + 51 = 212$ instances
- Total: 847 (training) + 212 (testing) = 1059 instances

This means approximately:

- Training set: $847 / 1059 \approx 80\%$
- Testing set: $212 / 1059 \approx 20\%$

3. Purpose of Splitting:

- Training Set: Used to build and train the model, learning the patterns within the data.
- Testing Set: * Used to evaluate the model's performance on unseen data, providing an unbiased assessment of its generalization ability.

the provided distributions indicate a stratified split of your dataset into training and testing sets, ensuring that each subset accurately reflects the class proportions of the original dataset. This approach is crucial for building a reliable and generalizable machine learning model.

3.4 SMOTE

Class distribution after SMOTE: Counter({'high': 429, 'low': 429, 'medium': 429})

Table 2. SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is a method used to address class imbalance in datasets. By generating synthetic examples for the minority classes, SMOTE helps to balance the class distribution, which can improve the performance of machine learning models, especially those sensitive to imbalanced data.

Original Dataset Distribution

- Low: 429 instances
- Medium: 374 instances
- High: 256 instances

After SMOTE Application

- Low: 429 instances
- Medium: 429 instances
- High: 429 instances

Explanation of Distribution After SMOTE

1. Purpose of SMOTE:

- SMOTE is applied to balance the class distribution by creating synthetic samples of the minority classes ('medium' and 'high' in this case) until all classes have the same number of instances as the majority class ('low').

2. Balancing the Classes:

- Low: This class already had 429 instances, so no synthetic samples are needed.
- Medium: Initially had 374 instances. SMOTE generates $429 - 374 = 55$ synthetic instances to reach 429 instances.
- High: Initially had 256 instances. SMOTE generates $429 - 256 = 173$ synthetic instances to reach 429 instances.

3. Synthetic Instances:

- SMOTE creates synthetic instances by interpolating between existing minority class instances. This is done by selecting a point from the minority class and creating new points along the line segments joining the point and its nearest neighbors. This

process increases the number of instances in the minority classes, leading to a balanced dataset.

Importance of a Balanced Dataset

1. Model Performance:

- Machine learning models, especially those that do not handle imbalanced data well, can perform poorly when there is a significant class imbalance. The model might become biased towards the majority class.
- Balancing the dataset ensures that the model has an equal opportunity to learn from all classes, improving its ability to generalize and perform well on unseen data.

2. Evaluation Metrics:

- With a balanced dataset, evaluation metrics like accuracy, precision, recall, and F1-score become more reliable, as they are not skewed by class imbalance.

After applying SMOTE, the dataset now has an equal number of instances (429) in each class ('low', 'medium', and 'high'). This balanced distribution helps in building a more robust machine learning model that can fairly learn and predict across all classes, avoiding biases towards the majority class.

3.5 Random Forest

Classification Report (with balanced class weights):				
	precision	recall	f1-score	support
high	1.00	0.98	0.99	51
low	1.00	1.00	1.00	86
medium	0.99	1.00	0.99	75
accuracy			1.00	212
macro avg	1.00	0.99	0.99	212
weighted avg	1.00	1.00	1.00	212

Picture 6. Classification Report

- Accuracy: 1.00

The overall model is 100% accurate, meaning all predictions for all classes on the test data are correct.

- Macro Avg:
 - ❖ Precision: 1.00
 - ❖ Recall: 0.99
 - ❖ F1-score: 0.99

This is an unweighted average of the metrics per class, giving each class equal weight.

- Weighted Avg:
- ❖ Precision: 1.00
- ❖ Recall: 1.00
- ❖ F1-score: 1.00

This is a weighted average of the metrics per class, taking into account the proportion of each class in the test data.

This report shows that the RandomForestClassifier model with counterbalanced class weights performs very well on the test data. All performance metrics show very high values, indicating that the model has excellent predictive ability and shows no significant bias towards any class. The perfect accuracy indicates that no prediction errors were made by the model on the test data.

3.6 Modeling Data

a. Decision Tree

```
array([[0., 0., 6.5, 0. ],
       [0., 1., 6.5, 1. ],
       [0., 0., 6.8, 0. ],
       [1., 1., 6.8, 1. ],
       [1., 1., 8.6, 0. ],
       [1., 0., 9.5, 1. ],
       [1., 1., 9., 1. ],
       [1., 0., 6.7, 1. ],
       [1., 1., 5.6, 0. ],
       [1., 1., 6.5, 1. ]])
```

Picture 7. Decision Tree Array

```
X = milk[['Odor', 'Fat ', 'pH', 'Taste']].values
X[0:10]
```

Table 4. Decision Tree 1

1. Feature Selection: This code selects columns that are considered important for further analysis or machine learning models. In this case, the columns selected are related to milk characteristics such as Odor, Fat, pH, and Taste.

2. Conversion to Numpy Array: Converts the DataFrame into a numpy array for easy math operations and integration with machine learning algorithms that generally use numpy format.

3. Data Preview: By viewing the first 10 rows (X[0:10]), a researcher or analyst can quickly check how the selected data looks like and make sure that the data is as expected before proceeding to further analysis steps.

```
756    medium
962    medium
922    medium
729     high
878     low
1048    low
408     low
761    medium
940     low
1032    high
Name: Grade, dtype: object
```

Picture 8. Dataframe Pandas

```
y = milk["Grade"]
y[0:10]
```

Table 3. Decision Tree 2

The image above displays the results of coding to numeric values, this also aims to:

1. Understanding the Initial Distribution of Data: Provides an initial overview of the distribution and variation of data in the Grade column.
2. Data Validation: Ensures data from the Grade column is correctly imported into Series y.
3. Preparation for Analysis or Modeling: Knowing the data type and grade distribution helps in data preparation for further analysis or use in machine learning models.
4. Category Handling: Knowing that the Grade column contains categorical data, the next step may involve encoding the categories into a numerical format for machine learning algorithms.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3, random_state=42)
```

Table 5. Decision Tree 3

In the coding table above to separate the data into training and testing data, this aims as a step before further modeling.

Input Parameters:

- X and y are variables representing the dataset:
- X is an array or DataFrame that contains the input of the dataset.
- y is an array or Series containing the labels or targets (outputs) of the dataset.

Data Separation:

- The `train_test_split` function separates dataset `X` and `y` into four subsets:
- `X_train`: Subset of `X` used to train models (data training).
- `X_test`: Subset of `X` used to test the model (testing data).
- `y_train`: Subset of `y` corresponding to `X_train` (label for training data).
- `y_test`: Subset of `y` corresponding to `X_test` (label for testing data).

Parameter `test_size`:

- `test_size=0.3` means that 30% of the dataset will be used as test data, while the remaining 70% will be used as training data.

Random_state parameter:

- `random_state=42` is a seed for random number generators used by `train_test_split` to ensure that data separation is consistent and reproducible whenever code is run. You can use any number as seed.

Function Description

- `train_test_split`: This function randomly divides the dataset into two parts, training data and testing data, based on the proportions specified by `test_size`.

Examples of Use

- Suppose you have a dataset with features stored in the `X` variable and labels stored in the `y` variable. Using the above code, the dataset is divided into two parts: one part for training the model (`X_train` and `y_train`) and one part for testing the model (`X_test` and `y_test`).

```
[ 'low' 'medium' 'medium' 'high' 'medium' ]
280      low
178     medium
402     medium
805      high
4       medium
Name: Grade, dtype: object
```

Picture 9. Pandas Series

The image shows a Pandas Series named `'Grade'` that contains categorical data representing grades classified as 'low', 'medium', and 'high'. The Series has both the values and the corresponding indices displayed. Here's a detailed explanation of what the image represents:

Explanation of the Image

1. Series Values and Index:

- The first line ``['low' 'medium' 'medium' 'high' 'medium']`` indicates the unique categories in the `'Grade'` column.
- The following lines show the index and the corresponding grade value.

2. Entries in the Series:

- ❖ Index 280: Grade is 'low'
- ❖ Index 178: Grade is 'medium'
- ❖ Index 402: Grade is 'medium'
- ❖ Index 805: Grade is 'high'
- ❖ Index 4: Grade is 'medium'

3. Name and Data Type:

- Name: The Series is named `'Grade'`.
- dtype: The data type of the Series is `'object'`, indicating that the Series contains string values.

Interpretation

- This snippet of data seems to be a small part of a larger dataset where the `'Grade'` column classifies items (perhaps students, products, or some other entities) into different categories of performance or quality: 'low', 'medium', and 'high'. The indices (280, 178, 402, 805, 4) correspond to the positions of these entries in the original DataFrame.

Contextual Use

This type of data is typically used in various analyses such as:

- ❖ Classification Tasks: Training machine learning models to predict the grade based on other features.
- ❖ Statistical Analysis: Understanding the distribution of grades and identifying any patterns or correlations with other variables.
- ❖ Data Visualization: Creating plots to visually represent the distribution and frequency of each grade category.

By examining the indices and values, you can understand how the grades are distributed across different entries in the dataset and use this information for further analysis or modeling.

Decision Tree Accuracy: 0.875				
	precision	recall	f1-score	support
high	0.71	0.83	0.77	12
low	0.92	1.00	0.96	23
medium	0.92	0.79	0.85	29
accuracy			0.88	64
macro avg	0.85	0.88	0.86	64
weighted avg	0.88	0.88	0.87	64

Picture 10. Decision Tree Accuracy

The image shows a classification report for a Decision Tree model, providing detailed metrics on its performance. Here's a brief explanation of the results to determine whether the Decision Tree model is performing well:

Key Metrics

1. Precision: The ratio of correctly predicted positive observations to the total predicted positives.
2. Recall: The ratio of correctly predicted positive observations to all observations in the actual class.
3. F1-score: The weighted average of precision and recall.
4. Support: The number of actual occurrences of the class in the dataset.

Class-specific Performance

- High:

- ❖ Precision: 0.71
- ❖ Recall: 0.83
- ❖ F1-score: 0.77
- ❖ Support: 12

- Low:

- ❖ Precision: 0.92
- ❖ Recall: 1.00
- ❖ F1-score: 0.96
- ❖ Support: 23

- Medium:

- ❖ Precision: 0.92
- ❖ Recall: 0.79
- ❖ F1-score: 0.85
- ❖ Support: 29

Overall Performance

- ❖ Accuracy: 0.88
- ❖ This means 88% of the total predictions were correct.

- Macro Average:

- ❖ Precision: 0.85
- ❖ Recall: 0.88
- ❖ F1-score: 0.86
- ❖ This is the average performance metric for each class without considering class imbalance.

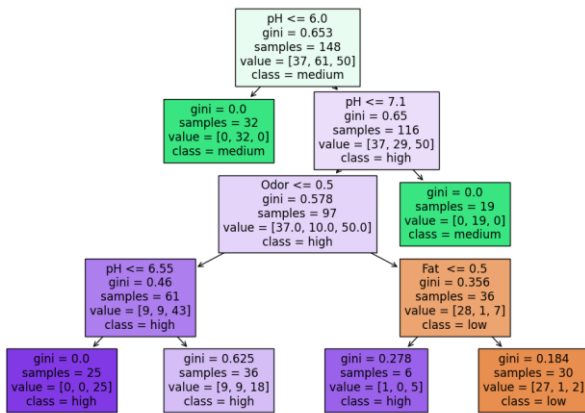
- Weighted Average:

- ❖ Precision: 0.88
- ❖ Recall: 0.88
- ❖ F1-score: 0.87
- ❖ This average takes into account the support (number of true instances) of each class.

Summary

- ❖ Low class performance is excellent, with perfect recall and very high precision and F1-score.
- ❖ Medium class performance is also very good, with high precision and decent recall.
- ❖ High class performance is slightly weaker compared to others, with lower precision and F1-score but still acceptable recall.
- ❖ Overall Accuracy of 88% suggests the model is performing well.
- ❖ Macro and Weighted Averages** are also high, indicating balanced performance across all classes.
- ❖ However, there is some room for improvement in the "high" class performance.

the Decision Tree model shows good overall performance, especially for the "low" and "medium" classes, but could benefit from improvements in predicting the "high" class more accurately.



Picture 11. Decision Tree Structure

The image shows a visualization of a Decision Tree classifier used to predict the class labels ('low', 'medium', 'high') based on features such as pH, Odor, and Fat. Here's an analysis of the Decision Tree and an assessment of its performance:

Decision Tree Structure

1. Root Node:

- `pH <= 6.0`
- Gini impurity: 0.653
- Total samples: 148
- Class distribution: [37, 61, 50] (low, medium, high)
- Predicted class: medium

2. Left Subtree (pH <= 6.0):

- Splits into a pure node:
- All samples (32) are 'medium' (Gini = 0.0).

3. Right Subtree (pH > 6.0):

- Further splits based on `pH <= 7.1` and then `Odor <= 0.5` and `Fat <= 0.5`.
- These nodes show more detailed splits:
- Some nodes are pure (Gini = 0.0), and others have varying levels of impurity.

Assessing the Tree

1. Purity of Leaves:

- Several leaf nodes have a Gini impurity of 0, indicating pure classification.
- However, other nodes still have some impurity, meaning they contain samples of different classes.

2. Sample Distribution:

- Nodes like `pH <= 6.0` and `Odor <= 0.5` result in relatively pure splits.

- Some nodes like `Fat <= 0.5` still have some mixed class labels.

3. Depth and Complexity:

- The tree is reasonably shallow with a depth of 4, which suggests it's not overly complex and thus less likely to overfit.
- This level of simplicity is generally desirable as it indicates that the model is not overly complex, which can help in generalizing to new data.

Overall Performance (In Context of Previous Classification Report)

Precision, Recall, and F1-score:

Previous metrics showed:

- High class: Lower precision (0.71) but decent recall (0.83).
- Low and Medium classes had high precision and recall.
- The visualization shows that while the Decision Tree does well in splitting and classifying many of the samples correctly, it might struggle with certain splits (e.g., `pH > 6.55`).

The Decision Tree model appears to perform reasonably well based on both the visual representation and the classification report metrics. The key points are:

Strengths:

- Good overall accuracy (0.88).
- Effective splits with many pure nodes.
- Good balance between depth and complexity.

Weaknesses:

- Some nodes still have mixed classes indicating room for improvement in certain splits.
- The "high" class has lower precision, suggesting the model struggles more with these predictions.

The Decision Tree results are generally good but not perfect. The model handles most splits effectively, and the overall accuracy is high. There is still some impurity in certain nodes, especially for the "high" class, which aligns with the slightly lower precision observed in the classification report. For further improvement, you could consider:

- Pruning the Tree: To remove unnecessary nodes and reduce complexity.
- Tuning Hyperparameters: Adjusting the depth or other parameters of the tree.

- Using Ensemble Methods: Combining the Decision Tree with other models to improve robustness and performance, such as using Random Forests or Gradient Boosting.

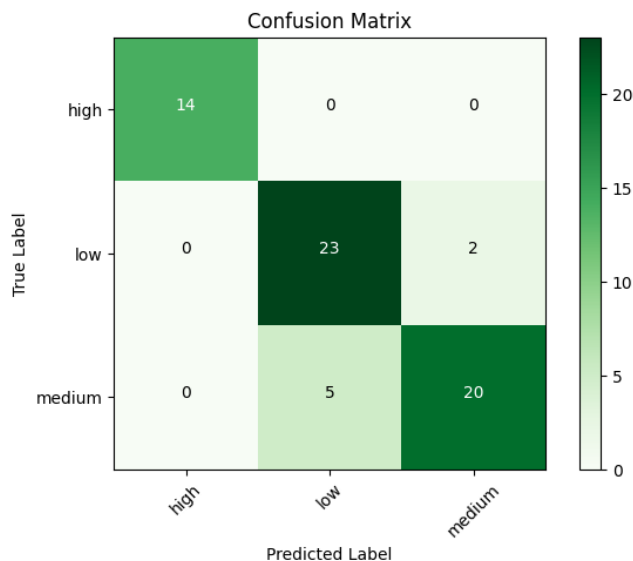
b. Confusion Matrix By Decision Tree

Confusion Matrix, without normalization

```
[[14  0  0]
 [ 0 23  2]
 [ 0  5 20]]
```

	precision	recall	f1-score	support
high	1.00	1.00	1.00	14
low	0.82	0.92	0.87	25
medium	0.91	0.80	0.85	25
accuracy			0.89	64
macro avg	0.91	0.91	0.91	64
weighted avg	0.89	0.89	0.89	64

Picture 12. Decision Tree Confusion Matrix



Picture 13. Decision Tree Confusion Matrix Label

The image shows a confusion matrix, which is used to evaluate the performance of a classification model. The matrix displays the number of correct and incorrect predictions made by the model compared to the actual labels in the test data.

Breakdown of the Confusion Matrix:

- Y-axis (True Label): Represents the actual labels of the data.
- X-axis (Predicted Label): Represents the labels predicted by the model.

The labels in this matrix are "high," "low," and "medium."

Interpretation of the Matrix:

- First Row (high):

- There are 14 instances that are actually "high" and predicted as "high" (correct).
- No instances of "high" were predicted as "low" or "medium" (no prediction errors for the "high" class).

- Second Row (low):

- There are 23 instances that are actually "low" and predicted as "low" (correct).
- There are 2 instances that are actually "low" but predicted as "medium" (prediction errors).

-Third Row (medium):

- There are 20 instances that are actually "medium" and predicted as "medium" (correct).
- There are 5 instances that are actually "medium" but predicted as "low" (prediction errors).

Colors:

- The color of the cells in the matrix indicates the number of instances: the darker the color, the higher the number of instances in that cell.
- The dark green in the (low, low) cell indicates the highest count (23), while lighter shades of green in other cells indicate fewer instances.

Conclusion:

- The model performs very well for the "high" class with perfect accuracy.
- The model has some difficulty distinguishing between the "low" and "medium" classes.

c. SVM

```

milk['Grade'] = milk['Grade'].replace('low', 0)
milk['Grade'] = milk['Grade'].replace('medium', 1)
milk['Grade'] = milk['Grade'].replace('high', 2)
milk.info()

# KOLOM "Temperature" : dalam 'C
# KOLOM "Taste"       : (0 = Bad) & (1 = Good)
# KOLOM "Odor"        : (0 = Bad) & (1 = Good)
# KOLOM "Fat"         : (0 = Low) & (1 = High)
# KOLOM "Turbidity"   : (0 = Low) & (1 = High) ->
Cloudiness(keruh)
# KOLOM "Grade"       : (0 = Low), (1 = Medium), &
(2 = High)

```

Table 6. SVM 1

Value Replacement in the 'Grade' Column:

- `milk['Grade'] = milk['Grade'].replace('low', 0)`: Replaces the value 'low' with 0.
- `milk['Grade'] = milk['Grade'].replace('medium', 1)`: Replaces the value 'medium' with 1.
- `milk['Grade'] = milk['Grade'].replace('high', 2)`: Replaces the value 'high' with 2.

These steps convert the categorical values in the 'Grade' column into numerical values, which are often required for analysis or modeling purposes in machine learning.

```

array([[0. , 0. , 6.5, 0. ],
       [0. , 1. , 6.5, 1. ],
       [0. , 0. , 6.8, 0. ],
       [1. , 1. , 6.8, 1. ],
       [1. , 1. , 8.6, 0. ],
       [1. , 0. , 9.5, 1. ],
       [1. , 1. , 9. , 1. ],
       [1. , 0. , 6.7, 1. ],
       [1. , 1. , 5.6, 0. ],
       [1. , 1. , 6.5, 1. ]])

```

Picture 14. SVM Array

```

X = milk[['Odor', 'Fat ', 'pH', 'Taste']].values
X[0:10]

```

Table 7. SVM 2

The image and code shows Python code that extracts the columns "Odor," "Fat," "pH," and "Taste" from a DataFrame named `Milk_balance`, and stores their values in a variable `X`. Then, the first ten rows of the array `X` are displayed.

The array contains numerical values for each of the selected columns from the DataFrame, with each row representing one data sample.

```

756    1
962    1
922    1
729    2
878    0
1048   0
408    0
761    1
940    0
1032   2
Name: Grade, dtype: int64

```

Picture 15. Distribution of grades

Its output provide a snapshot of the distribution of grades in the dataset, which is crucial for understanding the data and evaluating any predictive models built using this data.

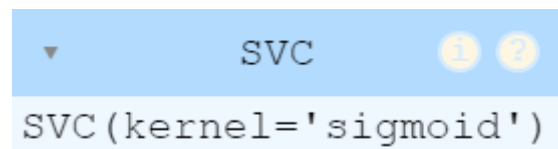
```

y = milk["Grade"]
y[0:10]

```

Table 8. SVM 3

- `milk["Grade"]`: has the 'Grade' column of the milk dataset with the label to be predicted
- `y[0:10]`: displays the first 10 rows of the Grade column. Index [0:10] is slicing, which takes elements from index 0 to 9 (the first 10 rows).



```

SVC(kernel='sigmoid')

```

Picture 16. Sigmoid Kernel

The sigmoid kernel is another common and versatile kernel function for SVMs. It is defined as $\mathbf{K}(\mathbf{x}, \mathbf{y}) = \tanh(\alpha * \mathbf{x} * \mathbf{y} + \beta)$, where \mathbf{x} and \mathbf{y} are the input vectors, α and β are parameters, and \tanh is the hyperbolic tangent function.

```

array([0, 1, 1, 0, 1], dtype=int64)

```

Table 9. SVM 4

The array `array([0, 1, 1, 0, 1], dtype=int64)` represents categorical 'Grade' data that has been converted into numerical form for analysis or modeling purposes. These values allow the use of statistical techniques and machine learning algorithms that require numerical input.

SVM Accuracy: 0.53125					
	precision	recall	f1-score	support	
0	0.50	0.44	0.47	25	
1	0.55	0.92	0.69	25	
2	0.00	0.00	0.00	14	
accuracy			0.53	64	
macro avg	0.35	0.45	0.38	64	
weighted avg	0.41	0.53	0.45	64	
Confusion Matrix, without normalization					
[[11 14 0]					
[2 23 0]					
[9 5 0]]					

Picture 17. Performance metrics

The image displays the performance metrics and confusion matrix of a Support Vector Machine (SVM) classifier. Here's a brief explanation:

1. Accuracy: The SVM model has an accuracy of 53.13%, meaning it correctly predicts 53.13% of the instances.

2. Classification Report: This includes precision, recall, and F1-score for each class (0, 1, and 2):

- Class 0: Precision is 0.50, recall is 0.44, and F1-score is 0.47, with 25 instances.
- Class 1: Precision is 0.55, recall is 0.92, and F1-score is 0.69, with 25 instances.
- Class 2: Precision, recall, and F1-score are all 0.00, with 14 instances.

3. Macro Average: The unweighted average of the precision, recall, and F1-score across all classes:

- ❖ Precision: 0.35
- ❖ Recall: 0.45
- ❖ F1-score: 0.38

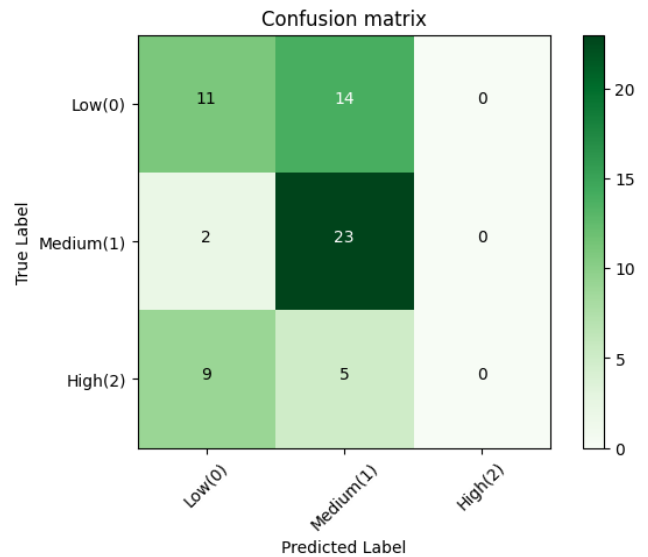
4. Weighted Average: The weighted average of the precision, recall, and F1-score, accounting for the number of instances in each class:

- ❖ Precision: 0.41
- ❖ Recall: 0.53
- ❖ F1-score: 0.45

5. Confusion Matrix: Shows the counts of true positives, false positives, and false negatives for each class:

- ❖ Class 0: 11 correct, 14 misclassified as class 1, 0 as class 2
- ❖ Class 1: 2 misclassified as class 0, 23 correct, 0 as class 2
- ❖ Class 2: 9 misclassified as class 0, 5 as class 1, 0 correct

Overall, the model performs well for class 1, moderately for class 0, and poorly for class 2.



Picture 18. Heatmap Representation

The image shows a heatmap representation of the confusion matrix for a classification model. The confusion matrix provides a visual summary of the model's performance by comparing the true labels to the predicted labels.

- Axes Labels:

- The x-axis represents the predicted labels: Low (0), Medium (1), and High (2).
- The y-axis represents the true labels: Low (0), Medium (1), and High (2).

- Matrix Values:

For class Low (0):

- 11 instances correctly classified as Low.
- 14 instances misclassified as Medium.
- 0 instances misclassified as High.

For class Medium (1):

- 2 instances misclassified as Low.
- 23 instances correctly classified as Medium.
- 0 instances misclassified as High.

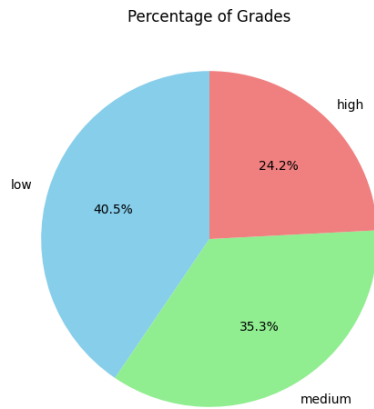
- For class High (2):

- 9 instances misclassified as Low.
- 5 instances misclassified as Medium.
- 0 instances correctly classified as High.

The heatmap color intensity indicates the count of instances, with darker shades representing higher counts. The model performs well in identifying Medium (1) class but struggles with High (2) class, misclassifying them predominantly into Low (0) and Medium (1) classes.

IV. VISUALIZATION

a. Pie Chart



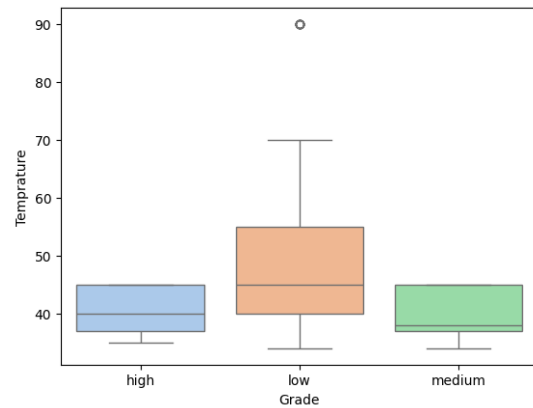
Picture 19. Pie Chart Percentage of Age

The pie chart above shows the percentage distribution of the three score categories: Low, Medium, and High. The following is an explanation of the pie chart results:

- Low (40.5%):
 - The Low category is the category with the highest percentage, which is 40.5% of the total data. This means that almost half of the whole sample has a value that falls into the Low category.
- Medium (35.3%):
 - The Medium category has a percentage of 35.3%. This indicates that more than one-third of the total sample has values that fall into the Medium category.
- High (24.2%):
 - The High category had the lowest percentage of 24.2%. This means that about a quarter of the total sample had scores that fell into the High category.

this pie chart shows that the majority of the sample falls into the Low and Medium categories, with the Low category being the most dominant. The High category is the least numerous compared to the other two categories.

b. Box Plot



Picture 20. Boxplot 1

This box plot visualizes the distribution of temperatures for three different grades: high, low, and medium. Here's a detailed explanation of the components and what they represent:

1. Boxes: Each box represents the interquartile range (IQR), which contains the middle 50% of the data.

- The bottom of the box represents the first quartile (Q1), the 25th percentile.

- The top of the box represents the third quartile (Q3), the 75th percentile.

- The line inside the box represents the median (Q2), the 50th percentile.

2. Whiskers: The lines extending from the boxes, known as whiskers, represent the range of the data excluding outliers.

- The lower whisker extends from the bottom of the box (Q1) to the smallest value within $1.5 * \text{IQR}$ below Q1.

- The upper whisker extends from the top of the box (Q3) to the largest value within $1.5 * \text{IQR}$ above Q3.

3. Outliers: Any data points beyond the whiskers are considered outliers and are plotted as individual points. In this plot, there is one outlier in the "low" grade category.

Interpretation by Grade:

- High Grade:

- Temperature range: Approximately 38 to 50.

- Median temperature: Around 42.

- IQR: Roughly 40 to 44.

- No outliers.

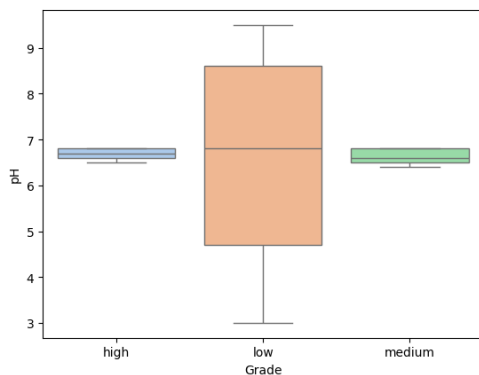
- Low Grade:

- Temperature range: Approximately 30 to 70.
- Median temperature: Around 50.
- IQR: Roughly 40 to 60.
- One outlier around 90.
- Medium Grade:
 - Temperature range: Approximately 38 to 48.
 - Median temperature: Around 43.
 - IQR: Roughly 40 to 46.
 - No outliers.

Key Observations:

- The "low" grade has the widest range of temperatures and includes an outlier around 90.
- The median temperature for "low" grade is higher than the medians for "high" and "medium" grades.
- The "high" and "medium" grades have similar median temperatures and ranges, but "medium" has a slightly wider IQR.

This box plot allows for a clear comparison of temperature distributions across the three grades, highlighting differences in central tendency and variability.



Picture 21. Boxplot 2

This box plot visualizes the distribution of pH values for three different grades: high, low, and medium. Here's a detailed explanation of the components and what they represent:

1. Boxes: Each box represents the interquartile range (IQR), which contains the middle 50% of the data.

- The bottom of the box represents the first quartile (Q1), the 25th percentile.

- The top of the box represents the third quartile (Q3), the 75th percentile.

- The line inside the box represents the median (Q2), the 50th percentile.

2. Whiskers: The lines extending from the boxes, known as whiskers, represent the range of the data excluding outliers.

- The lower whisker extends from the bottom of the box (Q1) to the smallest value within $1.5 * \text{IQR}$ below Q1.

- The upper whisker extends from the top of the box (Q3) to the largest value within $1.5 * \text{IQR}$ above Q3.

Interpretation by Grade:

- High Grade:

- pH range: Approximately 6.7 to 7.4.

- Median pH: Around 7.

- IQR: Roughly 6.9 to 7.2.

- No outliers.

- Low Grade:

- pH range: Approximately 3 to 9.

- Median pH: Around 7.

- IQR: Roughly 5.5 to 8.

- No outliers.

- Medium Grade:

- pH range: Approximately 6.7 to 7.3.

- Median pH: Around 7.

- IQR: Roughly 6.8 to 7.1.

- No outliers.

Key Observations:

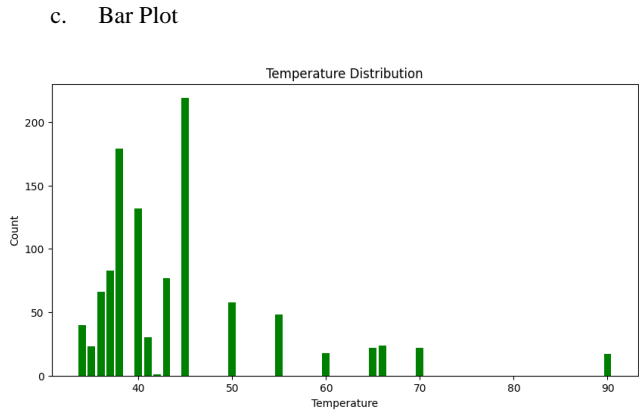
- The "low" grade has the widest range of pH values, spanning from 3 to 9, and the IQR indicates more variability in the middle 50% of the data.

- The median pH for all three grades is around 7, indicating a neutral pH is common across all grades.

- The "high" and "medium" grades have similar and narrower ranges and IQRs, indicating less variability compared to the "low" grade.

- No outliers are present in any of the grades, suggesting that all pH values fall within 1.5 times the IQR from the first and third quartiles.

This box plot allows for a clear comparison of pH distributions across the three grades, highlighting differences in variability while showing similar central tendencies.



Picture 22. Barplot

This bar plot visualizes the distribution of temperature values. Here's a detailed explanation of the components and what they represent:

Components of the Plot:

- 1. Bars: Each bar represents the count of occurrences of temperature values within specific intervals (bins). The height of each bar corresponds to the number of observations in that bin.
- 2. X-axis (Temperature): The horizontal axis shows the range of temperature values. It spans from 30 to 90.
- 3. Y-axis (Count): The vertical axis indicates the number of occurrences (frequency) for each temperature bin.

Key Observations:

- 1. Clustered Temperatures Around 40-45:
 - There is a significant clustering of temperature values between 40 and 45, with the highest bar (around 43-44) having over 200 occurrences.
 - This indicates that temperatures in this range are the most common in the dataset.
- 2. Smaller Peaks at Higher Temperatures:
 - There are smaller peaks at temperatures around 50, 60, and 70.
 - These smaller bars suggest that while there are some observations in these ranges, they are much less frequent compared to the 40-45 range.
- 3. Rare High Temperatures:
 - There are very few occurrences of temperatures around 80 and 90, indicating that these values are outliers or rare events in the dataset.

- Interpretation:
- The plot shows a skewed distribution with most temperature values concentrated in the lower range (40-45), tapering off as temperatures increase.
 - The presence of a few high temperature values around 80 and 90, which are far less frequent, suggests the possibility of outliers in the data.
 - This distribution aligns with the box plot of temperature for the "low" grade, which had a wider range and a notable outlier around 90.

This bar plot provides a clear picture of how temperature values are distributed across the dataset, highlighting areas of high concentration and indicating the presence of less frequent, higher temperature values.

TABLE I
UNITS FOR MAGNETIC PROPERTIES

#	COLUMN	NON-NULL COUNT	DTYPE
0	PH	1059 NON-NULL	FLOAT64
1	TEMPRATURE	1059 NON-NULL	INT64
2	TASTE	1059 NON-NULL	INT64
3	ODOR	1059 NON-NULL	INT64
4	FAT	1059 NON-NULL	INT64
5	TURBIDITY	1059 NON-NULL	INT64
6	COLOUR	1059 NON-NULL	INT64
7	GRADE	1059 NON-NULL	OBJECT

V. CONCLUSION

The milk dataset, consisting of 1059 entries with 8 columns, was a complete basis for analysis and contained no missing values. With each model providing a different perspective, the use of Decision Tree and SVM helped gain a better understanding of the relationship between features and milk categories. The performance evaluation of both models helps in selecting the most suitable model for practical applications, given the specific needs of the milk dataset used. The results of this analysis can be used to improve data processing and modeling and provide important insights for the decision-making process related to milk quality.

VI. ACKNOWLEDGMENT

We would like to express our sincere gratitude to Dr. Irmawati, S.Kom., M.M.S.I., our Data Modelling lecturer, for her unwavering support and insightful feedback throughout the duration of this course. Her expert guidance has been instrumental in the development of this research, and her dedication has made our learning experience both rewarding and inspiring.

REFERENCES

- [1] Kartika Budi Utami, Lilik Eka Radiati, and Pugu Hurjowardojo, "Kajian kualitas susu sapi perah PFH (studi kasus pada anggota Koperasi Agro Niaga di Kecamatan Jabung Kabupaten Malang)," *Jurnal Ilmu-Ilmu Peternakan*, vol. 24, no. 2, pp. 58–66, 2014, Accessed: Oct. 31, 2019. [Online]. Available: <https://jiip.ub.ac.id/index.php/jiip/article/view/174/243>
- [2] M. Frizzarin et al., "Predicting cow milk quality traits from routinely available milk spectra using statistical machine learning methods," *Journal of Dairy Science*, vol. 104, no. 7, pp. 7438–7447, Jul. 2021, doi: 10.3168/jds.2020-19576.
- [3] R. I. Abraham, B. Hidayat, and S. Darana, "Identifikasi Kualitas Kesegaran Susu Sapi Melalui Pengolahan Citra Digital Berdasarkan Metode Content-based Image Retrieval (cbir) Dengan Klasifikasi Decision Tree," *eProceedings of Engineering*, vol. 5, no. 2, Aug. 2018, Accessed: May 13, 2024. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/6594/6491>
- [4] "BAB II KERANGKA TEORI." Accessed: May 19, 2024. [Online]. Available: <http://repository.uinfabengkulu.ac.id/2305/3/BAB%20I.pdf>
- [5] A. H. Nasrullah, "IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI PRODUK LARIS "; *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, vol. 7, no. 2, pp. 45–51, Sep. 2021, doi: <https://doi.org/10.35329/jiik.v7i2.203>.
- [6] F. Abdusyukur, "PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) UNTUK KLASIFIKASI PENCEMARAN NAMA BAIK DI MEDIA SOSIAL TWITTER," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 1, pp. 73–82, May 2023, doi: <https://doi.org/10.34010/komputa.v12i1.9418>.
- [7] F. Mu, Y. Gu, J. Zhang, and L. Zhang, "Milk source identification and milk quality estimation using an electronic nose and machine learning techniques," *Sensors*, vol. 20, no. 15, p. 4238, Jul. 2020, doi: 10.3390/s20154238.
- [8] H. Anwar, T. Anwar, and S. Murtaza, "Review on food quality assessment using machine learning and electronic nose system," *Biosensors and Bioelectronics*, vol. 14, p. 100365, Sep. 2023, doi: 10.1016/j.biosx.2023.100365.
- [9] L. F. M. Mota et al., "Evaluating the performance of machine learning methods and variable selection methods for predicting difficult-to-measure traits in Holstein dairy cattle using milk infrared spectral data," *Journal of Dairy Science*, vol. 104, no. 7, pp. 8107–8121, Jul. 2021, doi: 10.3168/jds.2020-19861.
- [10] Y.-T. Wang et al., "A novel approach to temperature-dependent thermal processing authentication for milk by infrared spectroscopy coupled with machine learning," *Journal of Food Engineering*, vol. 311, p. 110740, Dec. 2021, doi: 10.1016/j.jfoodeng.2021.110740.
- [11] M. Frizzarin, T. F. O'Callaghan, T. B. Murphy, D. Hennessy, and A. Casa, "Application of machine-learning methods to milk mid-infrared spectra for discrimination of cow milk from pasture or total mixed ration diets," *Journal of Dairy Science*, vol. 104, no. 12, pp. 12394–12402, Dec. 2021, doi: 10.3168/jds.2021-20812.
- [12] R. Alaiz-Rodríguez and A. C. Parnell, "A Machine Learning Approach for Lamb Meat Quality Assessment Using FTIR Spectra," in *IEEE Access*, vol. 8, pp. 52385–52394, 2020, doi: 10.1109/ACCESS.2020.2974623.
- [13] A. E. Orche, A. Mamad, O. Elhamdaoui, A. Cheikh, M. E. Karbane, and M. Bouatia, "Comparison of machine learning classification methods for determining the geographical origin of raw milk using vibrational spectroscopy," *Journal of Spectroscopy*, vol. 2021, pp. 1–9, Dec. 2021, doi: 10.1155/2021/5845422.
- [14] J. S. Farah et al., "Differential scanning calorimetry coupled with machine learning technique: An effective approach to determine the milk authenticity," *Food Control*, vol. 121, p. 107585, Mar. 2021, doi: 10.1016/j.foodcont.2020.107585.
- [15] R. Muñoz, M. Cuevas-Valdés, and B. De La Roza-Delgado, "Milk quality control requirement evaluation using a handheld near infrared reflectance spectrophotometer and a bespoke mobile application," *Journal of Food Composition and Analysis*, vol. 86, p. 103388, Mar. 2020, doi: 10.1016/j.jfca.2019.103388.
- [16] C. Piras et al., "Speciation and milk adulteration analysis by rapid ambient liquid MALDI mass spectrometry profiling using machine learning," *Scientific Reports*, vol. 11, no. 1, Feb. 2021, doi: 10.1038/s41598-021-82846-5.
- [17] N. Slob, C. Catal, and A. Kassahun, "Application of machine learning to improve dairy farm management: A systematic literature review," *Preventive Veterinary Medicine*, vol. 187, p. 105237, Feb. 2021, doi: 10.1016/j.prevetmed.2020.105237.

- [18] M. Said, A. Wahba, and D. Khalil, "Semi-supervised deep learning framework for milk analysis using NIR spectrometers," *Chemometrics and Intelligent Laboratory Systems*, vol. 228, p. 104619, Sep. 2022, doi: 10.1016/j.chemolab.2022.104619.
- [19] T. Bobbo, S. Biffani, C. Taccioli, M. Penasa, and M. Cassandro, "Comparison of machine learning methods to predict udder health status based on somatic cell counts in dairy cows," *Scientific Reports*, vol. 11, no. 1, Jul. 2021, doi: 10.1038/s41598-021-93056-4.
- [20] E. Hosseini, J. B. Ghasemi, B. Daraei, G. Asadi, and N. Adib, "Near-infrared spectroscopy and machine learning-based classification and calibration methods in detection and measurement of anionic surfactant in milk," *Journal of Food Composition and Analysis*, vol. 104, p. 104170, Dec. 2021, doi: 10.1016/j.jfca.2021.104170.