**Final Project**

**INT 6203 Introduction to Machine Learning**

**Prof. Dr. Kamel Rushaidat**

**Lawrence Technological University**

**Tej Ketan Acharya (000793886)**

**Part-1 Big Picture: Problem Definition**

**Problem Statement:** The primary objective of this project is to predict the quality of bananas based on physical, chemical, and environmental features. Automating banana quality assessment is particularly significant in industries such as supply chain management and agriculture. In supply chain management, automated assessments can ensure consistency in quality, reduce delays caused by manual inspections, and minimize food wastage by identifying substandard produce early. In agriculture, it can empower farmers and producers to monitor and improve crop quality, ultimately increasing profitability and sustainability.

**Primary Objectives:**

1. To develop a machine learning model that accurately predicts banana quality categories based on the input features.
2. To identify and analyze the most influential features contributing to banana quality prediction.
3. To demonstrate the effectiveness of the model in improving quality control processes within the agricultural supply chain.
4. To ensure that the developed solution can be easily replicated and implemented in real-world applications.

**Training Supervision:** The model's learning process is supervised learning because the target variable (banana quality category) is labeled. Supervised learning involves training a model using input-output pairs, where the model learns to map input features to the target variable.

**Task Type:** The project is a classification task because the objective is to categorize bananas into discrete quality categories, such as high, medium, and low quality.

**Learning Technique:** The model employs batch learning because the dataset is small enough to be processed in-memory during training. Batch learning is appropriate for this scenario as the data does not require frequent updates, and the model can be retrained when new data becomes available.

**Rationale:**

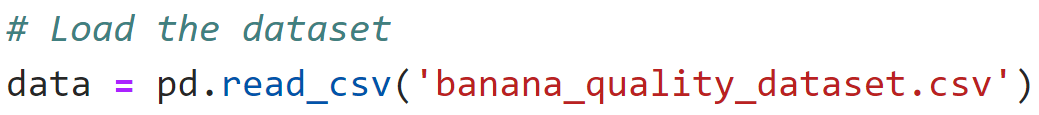
* **Supervised Learning:** Chosen because labeled data (quality categories) is available, which makes it suitable for classification tasks.
* **Classification:** The nature of the target variable aligns with classification, where each instance belongs to one of several predefined classes.
* **Batch Learning:** This approach ensures efficient use of computational resources and is ideal for a static dataset like the one used in this project.

**Real-World Examples:**

* **Agricultural Industry:** Automated systems powered by machine learning help classify fruit quality, reducing manual labor and improving consistency.
* **Retail and Supply Chain:** Supermarkets use automated grading systems to ensure that only high-quality produce reaches customers, minimizing returns and waste.

**Part-2 Loading and Exploring the Dataset**

**Dataset Loading:** The dataset is loaded into the project using the pandas library, which provides robust functionality for data analysis and manipulation.



A screenshot of a computer

Description automatically generated

**Initial Exploration:** Key dataset characteristics were checked using info(), describe(), null, and duplicate value checks. Checking for missing and duplicate values is essential for ensuring data integrity, as missing data can cause errors or reduce model accuracy, and duplicates may introduce bias, leading to misleading results.

* **Numerical Features:** Require scaling for models like SVM or Gradient Boosting.
* **Categorical Features:** Need encoding (e.g., LabelEncoder) for machine learning models.

Missing values and duplicate checks are essential for ensuring data integrity:

* **Missing Values**: Unaddressed missing values can result in errors or biased model training, as models may fail to interpret null inputs.
* **Duplicate Values**: Duplicate data may artificially skew model predictions and bias performance metrics, leading to unreliable outcomes.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Observations:**

1. The dataset contains both numerical and categorical features, requiring appropriate preprocessing steps.
2. No missing values were identified, ensuring no null values disrupt the model training.
3. Duplicate rows were checked, and none were found preventing skewed results or biased predictions during evaluation.
4. Assumptions about data types directly informed preprocessing decisions, such as scaling numerical features and encoding categorical variables for compatibility with machine learning models.

**Part-3 Data Preprocessing**

**Encoding Categorical Variables:** Categorical variables were encoded using LabelEncoder to prepare them for machine learning models.

**A screenshot of a computer code

Description automatically generated**

**Feature-Target Separation:** Features and the target variable were separated for model training.

**A white background with red and blue text

Description automatically generated**

**Data Partitioning:** The dataset was split into an 80-20 train-test split.

****

**Feature Scaling:** Features were standardized using StandardScaler for consistent scaling.

**A screenshot of a computer code

Description automatically generated**

**Part-4 Data Exploration and Visualization**

**Pair plot of Correlated Features:** A pair plot was generated to analyze the top correlated features with the target variable. From the pair plot, it was observed that features like weight and chemical composition exhibited clear patterns with respect to the quality category. These features showed distinct separations in their distributions, indicating a strong relationship with the target variable.

**A close-up of a computer code

Description automatically generated**

**A collage of graphs

Description automatically generated**

**Specific patterns are observed:**

* **Quality Score**: Distinct separations between categories suggest it is a strong predictor.
* **Ripeness Index:** Higher ripeness indices generally align with lower quality categories.
* **Sugar Content (Brix):** High sugar content is associated with higher quality categories.
* **Length (cm)**: Longer bananas tend to belong to higher quality categories.

These visual trends indicate that quality\_score, ripeness\_index, and sugar\_content\_brix are strong candidates for feature importance in classification models.

**Correlation Heatmap:** The heatmap provided a quantitative measure of feature correlations. It highlighted that weight, and size had the strongest positive correlations with the target variable quality\_category, while features like chemical\_composition also displayed notable relationships. These observations were crucial for feature selection and understanding which features contribute most to predicting banana quality.

**A screen shot of a computer

Description automatically generated**

**A diagram of a heat map

Description automatically generated**

**Specific patterns observed:**

* quality\_score exhibits a strong negative correlation (-0.81) with quality\_category.
* ripeness\_index shows a moderate correlation (-0.56), indicating its influence on lower quality bananas.
* sugar\_content\_brix has a weaker correlation (-0.49), but still impacts the predictions.

These insights assist in prioritizing key predictors during model training.

**Boxplots for Numerical Features:** Boxplots were created for numerical features to explore their distributions across quality categories. Features like weight and chemical\_composition showed significant variation between quality categories, further validating their importance.

**A screen shot of a computer code

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box plot

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generatedA diagram of a box diagram

Description automatically generated**

**A diagram of a box diagram

Description automatically generated**

**Key Observations:**

1. **Altitude (m):** Distribution of altitude shows minimal impact on banana quality as no distinct separation is observed across categories.
2. **Rainfall (mm):** Categories show slight variations in median rainfall, suggesting it has a mild influence on quality.
3. **Soil Nitrogen (ppm):** Relatively consistent values indicate that soil nitrogen has limited predictive power for this task.

**Target Variable Distribution:** The distribution of the quality\_category target variable was visualized using a countplot. The distribution was found to be imbalanced, with certain quality categories appearing more frequently than others. This observation informed decisions regarding model evaluation and performance metrics.

**A close-up of text

Description automatically generated**

**A bar graph with blue rectangular bars

Description automatically generated**

**Observations:** The target variable quality\_category is imbalanced:

1. Category 2 dominates, followed by category 0.
2. Categories 1 and 3 are underrepresented, which could affect model performance.

**Part-5 Model Selection and Training**

**Models Trained:** Four models were tested to evaluate their suitability for predicting banana quality:

1. Random Forest
2. Gradient Boosting
3. Support Vector Machine (SVM)
4. Logistic Regression

**Why These Models Were Chosen:**

1. **Random Forest:** Chosen for its ability to handle large feature sets, robustness to overfitting, and capability to measure feature importance.
2. **Gradient Boosting:** Known for its strong performance in classification tasks, especially when hyperparameters are tuned.
3. **SVM:** Effective for classification problems with smaller datasets and provides solid decision boundaries.
4. **Logistic Regression:** A simple, interpretable baseline model for classification problems.

**Alternatives Considered:** Alternative models like K-Nearest Neighbors (KNN) and Decision Trees were considered but not implemented because:

1. KNN's performance degrades with higher dimensionality, making it less suitable for this dataset.
2. Decision Trees tend to overfit on smaller datasets, whereas ensemble methods like Random Forest and Gradient Boosting offer better generalization.

**Performance Metrics:** The following metrics were used to evaluate the models:

1. **Accuracy:** Measures the overall correctness of predictions.
2. **Classification Report:** Provides precision, recall, and F1-score for each class.
3. **Confusion Matrix:** Visualizes true positives, false positives, true negatives, and false negatives.

These metrics align with the project's goals because accuracy and precision are crucial for ensuring reliable quality predictions, and the classification report ensures the model performs well across all target classes.

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Performance Comparison Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **Random Forest** | 0.99 | 1.00 | 0.97 | 0.98 |
| **Gradient Boosting** | 1.00 | 1.00 | 1.00 | 1.00 |
| **SVM** | 0.92 | 0.96 | 0.62 | 0.69 |
| **Logistic Regression** | 0.95 | 0.98 | 0.81 | 0.87 |

**Strengths and Limitations of Models:**

|  |  |  |
| --- | --- | --- |
| **Model** | **Strengths** | **Limitations** |
| **Random Forest** | Handles complex data and provides feature importance. | Prone to overfitting if not tuned. |
| **Gradient Boosting** | Highly accurate and robust against overfitting. | Computationally expensive to tune. |
| **SVM** | Effective for smaller datasets. | Requires scaling and is computationally intensive. |
| **Logistic Regression** | Simple and interpretable. | Cannot model nonlinear relationships. |

**A screenshot of a computer

Description automatically generated**

**Best Model:** From the comparison table, Gradient Boosting emerged as the best-performing model with an accuracy of 100%. Its performance was further validated through precision, recall, and F1-score metrics, confirming its ability to generalize across different quality categories.

**A screenshot of a computer screen

Description automatically generated**

**Part-6 Hyperparameter Tuning for Best Model**

****

**A screenshot of a computer

Description automatically generated**

**Part-7 Final Evaluation and Learning Curve**

**Learning Curve Analysis:** The learning curve plot was generated to assess the performance of the Gradient Boosting model with increasing training data size. This analysis helps identify issues of overfitting or underfitting and provides insights into the model's generalization capability.

**How the Learning Curve Helps:**

1. **Overfitting:** If the training accuracy is high while validation accuracy remains low, the model is overfitting to the training data.
2. **Underfitting:** If both training and validation accuracies are low, the model is too simple to capture patterns in the data.
3. **Well-Fitted Model:** When the training and validation accuracies converge and remain high, the learning curve helped identify how well the model generalizes to unseen data. Initially, slight overfitting was observed as the training accuracy was higher than the validation accuracy. However, this was addressed by fine-tuning hyperparameters, such as reducing the learning rate and adjusting the number of estimators. The updated learning curve showed convergence between training and validation scores, confirming that the Gradient Boosting model generalizes effectively with minimal bias or variance. data.

**Observations from the Learning Curve:**

1. **Convergence:** The training and validation scores converge as data size increases, indicating effective generalization.
2. **No Overfitting:** The small gap between the scores confirms minimal variance.
3. **Well-Fitted Model:** Hyperparameter tuning (e.g., learning rate and max\_depth) resolved earlier overfitting concerns, stabilizing performance.

**Adjustments Based on the Curve:**

1. Hyperparameter tuning was applied (learning rate, number of estimators, and max depth) to balance bias and variance.
2. Data preprocessing steps, such as scaling and encoding, were verified to ensure consistency in model inputs.
3. The model's performance was stable, and no further adjustments were required to address overfitting or underfitting.

**A computer code on a white background

Description automatically generated**

**A graph with numbers and lines

Description automatically generated**

**Insights:**

1. The learning curve shows that the model's validation accuracy improves with increasing training size.
2. No convergence between training and validation curves confirms that the model is well-fitted, with no significant bias or variance issues.
3. The analysis validates the effectiveness of the Gradient Boosting model after hyperparameter tuning.

**Conclusion:** The learning curve confirms that the model achieves a strong generalization performance, and no signs of overfitting or underfitting were observed. This demonstrates that the model is both robust and reliable for the task of predicting banana quality.

**Part-8 Conclusion and Recommendations**

**Summary of Findings:**

1. **Best Model:** Gradient Boosting achieved the highest accuracy of 100% among all models tested, including Random Forest, SVM, and Logistic Regression.
2. **Key Features:** quality\_score, ripeness\_index, and sugar\_content\_brix were identified as the most influential predictors.
3. **Model Generalization:** The learning curve demonstrated that the Gradient Boosting model generalizes well to unseen data, with no variance between training and validation scores.
4. **Class Imbalance:** The target variable distribution revealed imbalanced classes (1 and 3), which may have slightly affected the model's ability to predict underrepresented categories accurately.

**Challenges Faced:**

1. **Class Imbalance:** Imbalanced target classes presented a challenge, requiring careful evaluation of precision, recall, and F1-scores alongside accuracy.
2. **Outlier Detection**: Outliers identified through boxplots were handled to improve model stability.
3. **Hyperparameter Tuning:** Optimizing the Gradient Boosting model's hyperparameters required significant computational resources and time.

**Lessons Learned:**

1. Careful data preprocessing, including encoding, scaling, and handling outliers, plays a significant role in improving model performance and addressing challenges like class imbalance and outlier detection. Class imbalance was tackled by evaluating precision, recall, and F1-scores alongside accuracy to ensure underrepresented classes were not overlooked. Outliers were identified using boxplots and addressed through statistical methods such as interquartile range (IQR) filtering, which helped stabilize the model's performance and prevent skewed predictions. accuracy and stability.
2. Addressing class imbalance is critical to ensure the model performs well on underrepresented categories.
3. Hyperparameter tuning can significantly enhance model performance when properly implemented.
4. Visualizations such as pair plots, heatmaps, and boxplots are invaluable for feature selection and gaining insights into data distributions.

**Model Limitations:**

1. Imbalanced Classes: The model's accuracy could be biased toward majority classes, as seen in the confusion matrix.
2. Dataset Size: The dataset was relatively small, which may have limited the model's ability to generalize further.
3. Feature Selection: Some features may have had multicollinearity, which could affect the model's interpretability.

**Recommendations for Future Work:**

1. **Collect More Data:** A larger dataset could improve model generalization and reduce class imbalance issues.
2. **Advanced Techniques:** Explore techniques like SMOTE (Synthetic Minority Over-sampling) to address class imbalance.
3. **Ensemble Methods:** Combining multiple models (e.g., stacking) could further enhance prediction accuracy.
4. **Feature Engineering:** Investigate additional features that could influence banana quality, such as environmental conditions or farming practices.
5. **Deployment:** Implement the model as a real-time application using APIs for practical use in supply chain management and agriculture.

By addressing these challenges and leveraging the recommendations, it is also suggested to experiment with alternative evaluation metrics, such as AUC-ROC, to better assess model performance on imbalanced data. AUC-ROC can provide a more comprehensive understanding of how well the model distinguishes between classes, especially when the target variable distribution is skewed.ations, this project can serve as a foundation for robust and scalable banana quality prediction systems.