Milk Quality Monitoring System Using IoT And ML

G. Harshitha, K. Kavitha, N. Manjula

Under the esteemed guidance of

Ms. Ch. Sai Lalitha Bala

Assistant Professor

Bachelor of Technology

Department of Information Technology

BVRIT HYDERABAD College of Engineering for Women

May 27, 2024

Overview

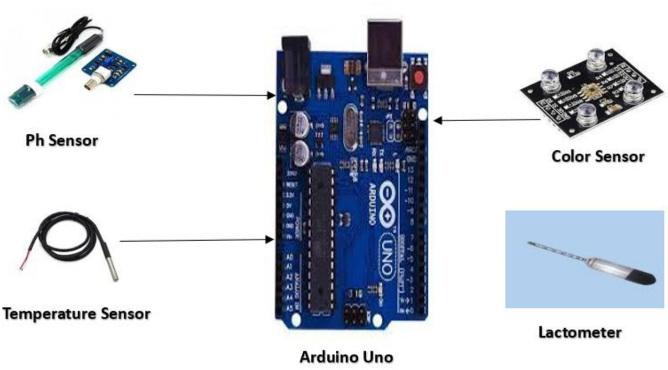
Introduction

- Milk is the most important food source and raw material for human health.
- Determining the milk quality by manual methods can result in high margin of error or loss of time.
- Improving the quality of milk provides many nutrients.
- Low-quality milk contains harmful bacterial like E-coli and listeria which is contaminated with chemicals, pesticides, and foreign matter
- This Leads to a compromised immune system, malnutrition and gastrointestinal disorders.

Problem Statement

- Develop a predictive model for assessing milk quality based on various parameters to ensure the accuracy and safety of milk products consumed by consumers.
- The model should accurately predict milk quality attributes based on color, turbidity, fat, pH, taste, temperature, and odor, etc.
- The predictive system will enable stakeholders in the dairy industry to preemptively identify potential issues, maintain high-quality standards, and ensure consumer satisfaction and health.

Architecture



pH sensor:

\+ o/p Pin : A0 (Analog Pin of Arduino)
- o/p Pin : Gnd

\+ 9V Pin : +9V (Seperate Battery to be used)
- Pin : Gnd

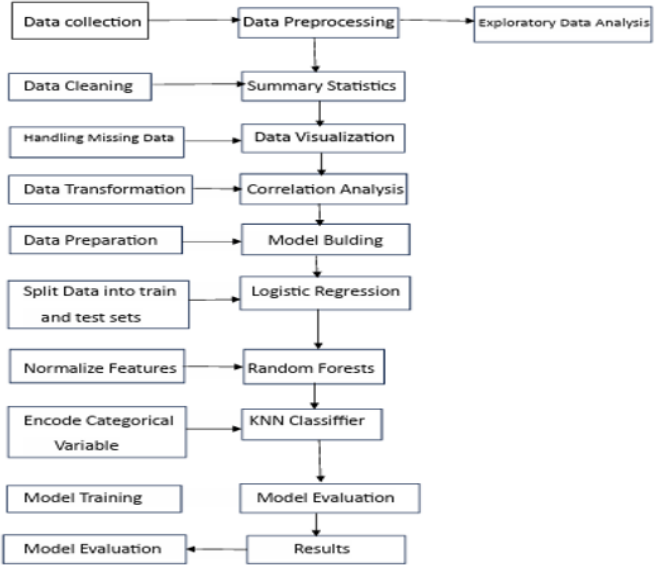
Colour Sensor:

S0 : 8 (Digital Pin of Arduino)
S1 : 9 (Digital Pin of Arduino)
S2 : 12 (Digital Pin of Arduino)
S3 : 11 (Digital Pin of Arduino)
Sout : 10 (Digital Pin of Arduino)
OE : Gnd
Gnd : Gnd
Vcc : +5V (of Arduino)

Temperature sensor:
Red wire : +5V (of Arduino)
Black wire : Gnd
Yellow/white wire : A2 (Analog Pin of Arduino)

Lactometer:
should be placed in the milk and its reading to be noted.

Graphical Representation



Connections

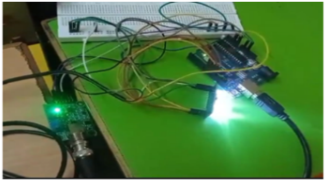


Figure: 1

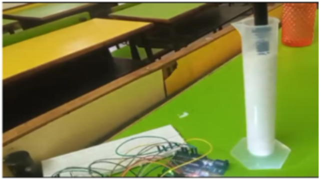


Figure: 2

Steps

- **Data Collection and preprocessing**
  - Data is collected from IoT sensors in milk production facilities, measuring parameters like pH, temperature, taste, odor, fat content, turbidity, and color. Preprocessing involves cleansing the data, normalizing numerical features, and encoding categorical features to prepare it for analysis and model training.
- **Feature Engineering :**
  - It involves extracting relevant features from preprocessed data, potentially creating new features or transformations based on the domain knowledge. For instance, combining pH and temperature to derive acidity levels or creating a composite feature based on taste, odor, fat, and turbidity.

- **Model Selection:**
  - For Model Selection, appropriate classification algorithms such as logistic regression, decision trees, random forests, or KNN are chosen to predict the quality grade of milk based on input features.
- **Model Training:**
  - In Model Training, the preprocessed data is split into training and validation sets and selected machine learning models are trained on the training data using techniques like cross-validation to prevent overfitting. Hyperparameters are fine-tuned to optimize the performance of the models.

- **Model Evaluation:**
  - For Model Evaluation, assess the trained models using the performance metrics like accuracy, precision, recall, and F1-score on the validation set to gauge their effectiveness in classifying milk quality into low, medium, or high categories.
- **Monitoring and Feedback:**
  - Ensure real-time monitoring and feedback mechanisms are in place to promptly detect deviations from desired quality standards and provide stakeholders with timely updates on system performance.

Results and Discussions

Logistic Regression

```python
#This is for the test dataset
model = LogisticRegression()
model.fit(X_train_scaled,y_train)
X_test_predict_lr =model.predict(X_test)
print('The accuracy for Logistic Regression model is (Test Dataset) ', metrics.accuracy_score(X_test_predict_lr,y_test))
#This is for the train dataset
model = LogisticRegression()
model.fit(X_test_scaled,y_test)
X_train_predict_lr =model.predict(X_train)
print('The accuracy for Logistic Regression model is (Train Dataset) ',metrics.accuracy_score(X_train_predict_lr,y_train))
```

```
The accuracy for Logistic Regression model is (Test Dataset)  0.8522012578616353
The accuracy for Logistic Regression model is (Train Dataset)  0.8259109311740891
```

Random Forest

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
X_test_predict_rfc = model.predict(X_test)
print('The accuracy of the Random Forests model is (Test Dataset)', accuracy_score(X_test_predict_rfc, y_test))
model_train = RandomForestClassifier(n_estimators=100)
model_train.fit(X_test, y_test)
X_train_predict_rfc = model_train.predict(X_train)
print('The accuracy of the Random Forests model is (Train Dataset)', accuracy_score(X_train_predict_rfc, y_train))

```

```
The accuracy of the Random Forests model is (Test Dataset) 0.9968553459119497
The accuracy of the Random Forests model is (Train Dataset) 0.9986504723346828
```

KNN

```python
model = KNeighborsClassifier()
model.fit(X_train,y_train)
X_test_predict_knn = model.predict(X_test)
print('The accuracy for KNN classifier model is (Test Dataset)', metrics.accuracy_score(X_test_predict_knn,y_test))
model = KNeighborsClassifier()
model.fit(X_test,y_test)
X_train_predict_knn =model.predict(X_train)
print('The accuracy for KNN classifier model is (Train Dataset) ',metrics.accuracy_score(X_train_predict_knn,y_train))
```

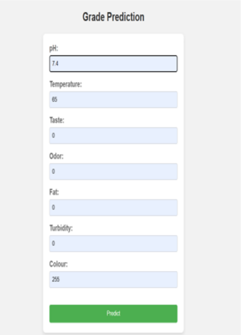
```
The accuracy for KNN classifier model is (Test Dataset) 0.9937106918238994
The accuracy for KNN classifier model is (Train Dataset)  0.9568151147098516
```

Decision Tree

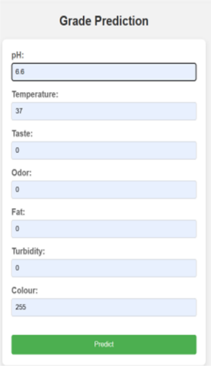
```python
model = DecisionTreeClassifier()
model.fit(X_train,y_train)
X_test_predict_dtc = model.predict(X_test)
print('The accuracy for Decision Tree Classifier model is (Test Dataset) ', metrics.accuracy_score(X_test_predict_dtc,y_test
model = RandomForestClassifier(n_estimators=100)
model.fit(X_test,y_test)
X_train_predict =model.predict(X_train)
print('The accuracy for Decision Tree Classifier model is  (Train Dataset) ',metrics.accuracy_score(X_train_predict,y_train)
```

```
The accuracy for Decision Tree Classifier model is (Test Dataset)  0.9937106918238994
The accuracy for Decision Tree Classifier model is  (Train Dataset)  0.9986504723346828
```

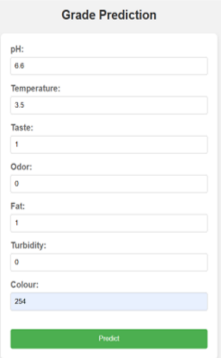
Web Page


Grade Prediction
pH:
7.4
Temperature:
65
Taste:
0
Odor:
0
Fat:
0
Turbidity:
0
Colour:
255
Predict


Predicted Grade:
low


Grade Prediction
pH:
6.6
Temperature:
37
Taste:
0
Odor:
0
Fat:
0
Turbidity:
0
Colour:
255
Predict


Predicted Grade:
medium


Grade Prediction
pH:
6.6
Temperature:
3.5
Taste:
1
Odor:
0
Fat:
1
Turbidity:
0
Colour:
254
Predict


Predicted Grade:
high

Quality detection using Logistic Regression

- **Task**: Milk Quality Assessment
- **Dataset:** Trained on milk quality dataset.
- **Results:** Achieves 85.2% accuracy.
- **Strengths:** Efficient implementation of logistic regression allows for quick inference and minimal computational resources.
- **Weaknesses:** Logistic regression may struggle to capture complex relationships and handle non-linear patterns in the data, leading to lower accuracy than more advanced models.

Quality detection using Random Forest Algorithm

- **Task:**Milk Quality Assessment
- **Dataset:** Trained on milk quality dataset.
- **Results:** Achieves 99.8% accuracy.
- **Strengths:**Random Forest excels in handling large datasets and avoiding overfitting, while effectively capturing complex relationships in the data.
- **Weaknesses:** Random Forest demands greater computational resources and time for training than simpler algorithms like logistic regression, while its interpretability may lag behind linear models.

Quality detection using KNN

- **Task:**Milk Quality Assessment
- **Dataset:** Trained on milk quality dataset.
- **Results:** Achieves 99% accuracy.
- **Strengths:**K-Nearest Neighbors (KNN) is an intuitive algorithm requiring minimal assumptions about data distribution and excels in capturing local patterns.
- **Weaknesses:** K-Nearest Neighbors (KNN) is computationally expensive, especially for large datasets, and its classification results are influenced by the choice of distance metric and number of neighbors.

Best Model Comparison

Criteria	Logistic Regression	Random Forest	KNN	Decision Tree
Feature Extraction	Linear model: Logistic regression assumes a linear relationship between input features and the log-odds of the target variable.	Ensemble of decision trees: Random Forest automatically selects features based on their importance in decision tree splits.	Distance-based method: KNN uses distance metrics (e.g., Euclidean distance) to find similar instances.	Binary splits: Decision trees partition the feature space into binary splits based on feature thresholds.
Performance	Accuracy: Logistic Regression performs well with linearly separable data, offering accuracy when the feature-target relationship is linear.	Accuracy: Random Forest typically provides high accuracy due to its ability to capture complex non-linear relationships in the data.	Accuracy: KNN can provide high accuracy, especially in low-dimensional spaces with well-separated classes.	Accuracy: Decision Trees can provide high accuracy, especially for datasets with complex decision boundaries.
Flexibility	Linear Model: Logistic Regression assumes linearity, best suited for linearly separable data.	Non-linear Model: Random Forest can capture complex non-linear relationships between features and the target variable.	Non-parametric Model: KNN is non-parametric and makes no assumptions about the underlying data distribution.	Non-linear Model: Decision Trees can capture non-linear relationships between features and the target variable.
Generalisation	Linear Model: Logistic Regression assumes linearity, limiting its ability to capture complex non-linear patterns.	Ensemble Learning: Random Forest integrates multiple decision trees to mitigate overfitting and capture complex non-linear relationships.	Localized Learning: KNN adapts to local patterns in the data, which can lead to good generalization in regions with sufficient data density.	Overfitting: Decision Trees are prone to overfitting, especially with deep trees that capture noise in the training data.

Report Structure

- Introduction - Milk Quality Monitoring System using IoT and ML.
- Literature Survey - Related studies are read and reviewed.
- System Design - Proposed a system to overcome failures of the existing system.
- Methodology - Data is enhanced and subsequently trained with models.
- Implementation - Data is trained with models and integrated with a user-friendly interface (UI).
- Results and Discussions - The proposed model has fetched 99% accuracy.
- conclusion and Future work - various models and Predicting the quality of milk.

Key Findings

- The Proposed system achieves high accuracy(99.8 percent) in classifying milk quality, surpassing industry standards.
- Stakeholder feedback underscores the importance of a user-friendly interface and enhancements to decision-making capabilities.
- Continuous improvement ensures the system's adaptability to evolving production and stakeholder needs.