 

**AUTOMATED GARBAGE BIN FILL-LEVEL CLASSIFICATION SYSTEM**

**24EEE431 – AI & EDGE COMPUTING**

**Report**

*Submitted b*y

* + - * KRISSHANTH R - CB.EN.U4ELC22021
      * U S PRAVEENKUMAR - CB.EN.U4ELC22056
      * PRANAV KARTHIKEYAN - CB.EN.U4ELC22062
      * PRANAV VENKATESHWARAN - CB.EN.U4ELC22127

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AMRITA SCHOOL OF ENGINEERING,

AMRITA VISHWA VIDYAPEETHAM,

COIMBATORE - 641112

March, 2025

**AIM OF THE PROJECT:**

The goal of this project is to create an intelligent system capable of accurately labeling garbage bins as "Full" or "Not Full" based on sensor data and machine learning techniques. Prediction of the fill level of the bins through this system facilitates smart waste management by optimizing collection routes and reducing operation costs.

**PROBLEM STATEMENT AND SOLUTION:**

Dustbin emptying is beset with tracking and timely dustbin emptying. Static scheduling or manual monitoring has been employed previously, which are bound to lead to overflows or wasteful collections. The system leverages sensor readings (fill level, battery, temperature, location, and temporal features) to automatically label bins as "Full" or "Not Full" using three machine learning algorithms: CatBoost, K-Nearest Neighbors (KNN) Classification, and Decision Tree Classification.

Main objectives are:

• Accurate Classification: Employ precision, recall, accuracy, and F1-score as primary performance measurements.

• Consistent Evaluation: Evaluate the performance not just using ROC curves, AUC values, and Cohen's Kappa Score but also.

• Efficiency: Monitor and optimize overall run time for the system in order to make it real-time or near real-time.

Decision-making by the system helps in best routing of waste collection and thereby lowers operational cost as well as environmental cost.

**PROCEDURE:**

Brief treatment of the operation procedures include:

1. **The sensor readings:**

The data forms collected from the sensors merging with the dustbins along with any vital historical information storage collected that referred to status of battery, temperature, geo location, and fill level grades in litres or percentage steeped as outputs. Data Preprocessing: Cleaning: removing extraneous characters and transforming all string values like temperature to float values.

1. **Feature Engineering:**

Time features like hour and day of the week against filling rates will create categories for geographical locations of the bins on categorical fields for model purpose when encoding.

No treatment: The minority class 'Full' was resampled using SMOTE.

1. **Model Training:**

Training with the subsequent three classifiers:

1. CatBoost: The gradient-boosting classifier of heterogeneous data; CatBoost.
2. k Neighbours Classifier - instance-based classification.
3. Decision Tree: Interpretable rules.
4. **Performance evaluation of the model:**

All these models were cross validated on the following metrics:

1. Core metrics-Precision, Recall, Accuracy, F1-Score, Confusion Matrix.
2. Secondary metrics-ROC Curve, AUC Score, Cohen's Kappa Score.
3. Total run time efficiency.

The graphical output consists of heatmaps of confusion matrix and ROC curves besides hard copies of all the measurements.

**FLOW EXPLANATION:**

1. **Data Preparation and Cleaning**:

The data is captured as Excel file from a system input. Attributes are trimmed from front and back sides and entire rows with NaN are deleted. All characters not numbers would be cleaned off using regular expressions for numeric features such as temperature, battery level, latitude, and longitude.

1. **Feature Engineering and Selection:**

The selected columns are WEEK-NO, Total (Litres), LATITUDE, LONGITUDE, and LOCATION (one-hot encoded). After adding these features, Add-Hour, day-of-week, and weekend flag features were computed, and fill rate was calculated on the features generated for concatenation into CatBoost.

1. **Training-Test Split and Scaling:**

The cleaned data were subjected to training-test partitioning at 80:20. StandardScaler was then applied to standardize the features.

1. **Imbalance Handling:**

This SMOTE procedure was applied to the training set to reduce any class imbalance between the two classes, i.e., Full or Not Full.

1. **Train Test and Scoring:**

Run all classifiers on the SMOTE-oversampled train set (Cat Boost, KNN, and Decision Tree) for comparative performance reports based on common classification metrics and visualization below: Confusion Matrix-Provided in heatmap.

1. **ROC Curve and AUC:**

Used to visualize the ratio of true positive to false-positive. Cohen's Kappa Score will show that the value for this will show whether all the agreement levels are greater than what is expected by chance.

1. **Comparison Time for Running:**

Awaits comparison for performance indices of the system Results Reporting: To set forth for comparison and therefore to consider a performance model for the application.

1. **Running Time**:

Measured to ensure the system’s efficiency.

**The core hardware components are:**

* End points of input parameters: for instance, bins status, battery, temperature and geolocation coordinates **t**h**a**tmust b**e** fed into the system.
* To recap again, one can never clean enough data, i.e., not purify it enough for any meaningful number in respect to a numeric attribute to be considered seriously towards some treatment alone being really a signal towards some outcome of a precious user attribute.
* Another th**i**ng **c**ould be tre**atin**gthe **o**utcom**e**s-non-fu**l**filling outcomes as classification by this concept-by full bins and empty bins, and so not as continuous but as binary classification.
* Evaluated model metrics include accuracy, precision and recall, confusion matrices, F1-score, area under the ROC curve separating positive from negative instances at all thresholds cumulative AUC, Cohen's kappa degree of agreement and time taken for the training.
* Visualising Data: There would also include heat maps of confusion matrices and noise charts along with ROC cuts, other than the above specifications.
* To bring in these, the nonfunctionals which came in form of performance indicators were to be put in**t**o the overall picture of judging the system.
* This model predictions have such scalability because they will take up more and redesigned data that will be generated later in future, each single improvement and development s**t**ati**n**g very much not a one-time single shot, withstands challenge into the future.
* Noise data that was supposed possible to affect model performance very little would use this in training on clean-data pre-processing toward improving the robustness of SMOTE.
* **Tools & Technolog**y**:**

**Programming** l**anguage:** Python **Librar**y**: P**andas, Numpy, Scikit-learn, Catboost, Imbalanced-learn, Matplotlib, and Seaborn.

**Data Source:** It is a CSV export of the historical data of sensors from Excel, mainly.

**Measurement of Evaluation:** It's just how many full bins were called too full, and this, by definition, reflects accuracy-the ratio of positive classification that turned out to be positive.

* Then recall tells how many of the actual, really positives the model caught; recall is real.
* In addition, accuracies tell how accurate the classification should have been.
* F1-Score is the harmonic mean of precision and recall.
* True-positive or false-positives and-negatives comprise feelings-based bases, multiplicity engaged through confusion matrix.
* AUC and ROC are mainly disjunct parameters of performance metrics, applicable under probabilistic outcome condition for appraising a model.
* Cohen's Kappa: Chance agreement correction.
* Running time measurement is essentially time-efficiency measured in time.

**SOLUTION**

**Models Developed:**

1. **CatBoost Classifier**
   * **Advantages:** Excellent handling of categorical data; robust performance with minimal parameter tuning.
   * **Implementation:** Uses engineered features (temporal, spatial) with CatBoost to produce high-quality predictions.
   * **Evaluation:** Metrics include high accuracy and balanced precision/recall, along with ROC-AUC and Cohen’s Kappa.
2. **K-Nearest Neighbors (KNN) Classifier**
   * **Advantages:** Simple and intuitive; effective with proper scaling and balanced data via SMOTE.
   * **Implementation:** Uses a KNN classifier with k=5; performance is evaluated with confusion matrix and ROC metrics.
   * **Evaluation:** Provides a baseline model with competitive precision and recall on oversampled data.
3. **Decision Tree Classifier**
   * **Advantages:** Highly interpretable; useful for understanding feature splits and decision boundaries.
   * **Implementation:** Uses a decision tree with controlled depth (max\_depth=5) to prevent overfitting.
   * **Evaluation:** Metrics include confusion matrix, ROC-AUC, and Cohen’s Kappa, supporting the model’s explainability.

**Comparative Analysis:**

* CatBoost tends to excel in scenarios with a mix of categorical and numerical data.
* KNN provides a straightforward approach when features are well-scaled.
* Decision Tree offers clear interpretability and faster inference.
* All models were enhanced by preprocessing (cleaning, feature extraction, scaling) and by using SMOTE to correct class imbalance.
* The evaluation metrics and visualizations (confusion matrix heatmaps, ROC curves) guide the selection of the best model for deployment.

**COMPARISON TABLE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sno.** | **CatBoost Algorithm** | **kNN Algorithm** | **Decision Tree Classifier** |
| 1. |  |  |  |
| 2. | AUC = 1.00 | AUC = 0.45 | AUC = 0.52 |
| 3. | Accuracy: 0.99 | Accuracy: 0.88 | Accuracy: 0.63 |
| 4. | Precision: 1.0 | Precision : 0.11 | Precision: 0.09 |
| 5. | Recall: 0.989 | Recall: 0.05 | Recall : 0.36 |
| 6. | F1-Score: 0.99 | F1-score : 0.07 | F1-score : 0.14 |
| 7. | Cohen’s Kappa Score: 0.99 | Cohen’s Kappa Score: 0.01 | Cohen's Kappa Score: 0.01 |
| 8. | Training time = 6.3s | Training time = 4.67s | Training time = 4.35s |

**CONCLUSION**

The hazard of waste bin fill level classification successfully marries the sensor reading features with the modern machine-learning techniques for waste bin status prediction. The proposed system focuses on data quality through the implementation of a thorough preprocessing method and proper feature engineering to keep the input data clean.

Bigger models allow comparing disparate completely level algorithms based upon their strengths. For instance, algorithms such as CatBoost, KNN, and Decision Tree are used here.

Therefore, performance measures are made up of precision, recall, accuracy, F1 score, ROC-AUC, Cohen's Kappa, and runtime evaluation that proved by all the efficacy and efficiency of the system. Hence the system presents a fullness-oriented model for optimizing the waste collection route of bins and minimizing operational costs, thereby beautifying life in the city.

Some next improvements would be better feature selection, database extensions, and testing for another ensemble or stacking classifiers. Thus, a good foundation has been laid for an intelligent sensor-based waste management system.