MLDM Coursework: *CadBury*

Harish Ravikumar HR00722@SURREY.AC.UK
Wilson Thomas Ayyappan WA00420@SURREY.AC.UK
Vasanth Kumar Chikkanan VC00352@SURREY.AC.UK
Jeganathan Duraisamy JD01755@SURREY.AC.UK
Durgesh Sasikumar Kavitha DS01702@SURREY.AC.UK

Abstract

This project explores the application of machine learning algorithms on two datasets: the Air Quality UCI dataset and the Water Potability dataset. The aim is to develop predictive models that can accurately assess air quality levels and determine the potability of water samples based on their chemical properties. The datasets are preprocessed to handle missing values and outliers, and various feature engineering techniques are applied to enhance model performance. Multiple machine learning algorithms are implemented, including decision trees, support vector machines, neural networks, and ensemble methods. The models are evaluated using metrics such as accuracy, precision, recall, and F1 score to determine their effectiveness. The results provide insights into the strengths and weaknesses of each approach, highlighting the best performing models for each dataset. This comparative analysis aims to identify the most suitable algorithms for predicting air quality and water potability, contributing valuable knowledge for environmental monitoring and public health safety.

1. Project Definition

Description of the Problems and Datasets

A. Air Quality UCI Dataset:

Source:https://archive.ics.uci.edu/dataset/360/air+quality

Description: This dataset contains 9358 instances of hourly averaged responses from an array of five metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The data is collected from March 2004 to February 2005 and includes various environmental parameters such as CO, NMHC, NOx, and temperature.

Objective: To predict air quality levels based on various environmental parameters.

Hypotheses: Environmental factors such as CO, NOx, and temperature significantly influence air quality levels.

Assumptions: All sensor data are accurate, and the chosen features are relevant for predicting air quality.

B. Water Potability Dataset:

Source:https://www.kaggle.com/datasets/uom190346a/water-quality-and-potability

Description: This dataset contains information about water samples collected from different sources and includes chemical properties like pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes, and Turbidity. It consists of 3276 instances with a binary label indicating whether the water is potable or not.

Objective: To classify water samples as potable or not based on their chemical properties.

Hypotheses: Specific chemical properties, such as pH, Hardness, and Turbidity, determine the potability of water.

Assumptions: The dataset accurately reflects the chemical properties necessary to assess water potability, and the features are sufficient for building predictive models.

2. Data Preparation

The data preparation steps involve several critical tasks to ensure the datasets are clean, well-integrated, and suitable

for machine learning and data mining processes. These steps are detailed as follows:

2.1 Data Cleaning / Data Integration:

1. **Handling Missing Values**:

   Air Quality Dataset: Missing values were handled by using imputation techniques. For instance, missing numeric values were filled using the mean of the respective column.

   Water Potability Dataset: Missing values were handled similarly by imputing the mean for numeric columns.

2. **Outlier Detection and Removal**:

   Outliers in the datasets were identified using statistical methods such as the Z-score. Outliers were removed to ensure they do not adversely affect the model training process.

3. **Data Integration**:

   No integration was necessary as each dataset was used independently for separate tasks (air quality prediction and water potability classification).

2.2 Variable Transformation

1. **Normalization and Scaling**:

   All numeric features were scaled using Min-Max normalization to ensure they are within a consistent range, which helps in improving the performance of distance-based algorithms like KNN and gradient-based algorithms like neural networks.

2.3 Data Exploration / Data Visualization

1. **Exploratory Data Analysis (EDA)**:

   Visualizations such as histograms, box plots, and scatter plots were created to understand the distribution of features, identify patterns, and detect anomalies.

   Correlation matrices were used to identify relationships between different features and the target variables. Heatmaps were used to visualize the strength of these relationships.

2. **Visualization of Data Distributions**:

   Histograms and density plots were used to visualize the distribution of numerical features.

   Box plots were employed to identify and visualize outliers and the spread of data.

3. **Feature Relationships**:

   Pair plots and scatter plots were used to visualize relationships between pairs of features, helping in identifying multicollinearity and interactions.

   Heatmaps were used to visualize the correlation matrix, highlighting the strength of relationships between features.

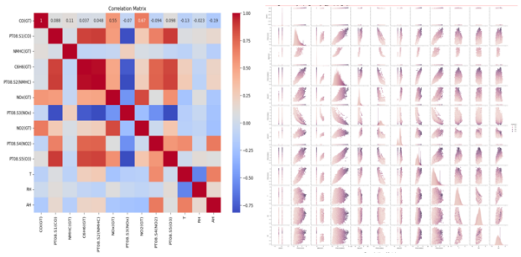

Fig 1: Air Quality Dataset Visualisations

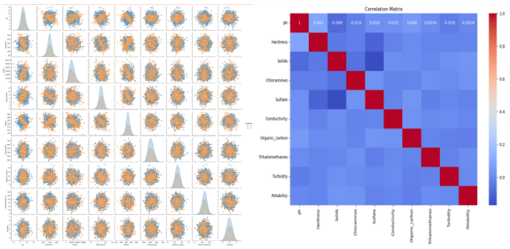

Fig 2: Water Quality Dataset Visualisations

3. Model development

Description of the Modelling Methods and Learning Algorithms

The following machine learning models were implemented to predict air quality levels and classify water samples as potable or not. Each algorithm was selected based on its strengths and suitability for the respective tasks.

3.1. CatBoost (Categorical Boosting):

Description: CatBoost is a gradient boosting algorithm that handles categorical features automatically, making it particularly efficient and accurate for datasets with categorical data.

Pros: Handles categorical features without preprocessing, fast training, good accuracy.

Cons: Can be computationally intensive, may require hyperparameter tuning.

Parameters and Settings: Learning rate, depth, iterations were optimized using grid search.

3.2 K-Nearest Neighbors (KNN):

Description: KNN is a simple, instance-based learning algorithm where predictions are made based on the closest training examples in the feature space.

Pros: Simple to implement, no training phase required.

Cons: Computationally expensive during prediction, sensitive to irrelevant features.

Parameters and Settings: Number of neighbors (k), distance metric (Euclidean).

3.3 Multiple Instance Learning (MIL):

Description: MIL is a variation of supervised learning where labels are available only for sets of instances, rather than individual instances.

Pros: Effective for problems where labels are associated with bags of instances.

Cons: More complex to implement and tune, requires careful handling of instance sets.

Parameters and Settings: Instance space metrics, bag creation methods.

3.4 Multilayer Perceptron (MLP):

Description: MLP is a class of feedforward artificial neural network that consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer.

Pros: Can model complex non-linear relationships, highly flexible.

Cons: Requires large datasets for training, prone to overfitting without proper regularization.

Parameters and Settings: Number of hidden layers, neurons per layer, activation functions, learning rate, batch size, epochs.

3.5 Q-Learning:

- **Description**: Q-Learning is a model-free reinforcement learning algorithm that aims to learn the value of an action in a particular state to maximize the total reward.
- **Pros**: Effective for decision-making tasks, learns optimal policies over time.
- **Cons**: Computationally intensive, requires a well-defined environment and reward structure.
- **Parameters and Settings**: Learning rate, discount factor, exploration rate.

Each algorithm was implemented using relevant Python libraries such as sklearn for KNN, CatBoost library for CatBoost, and keras for MLP. Custom implementations were created for MIL and Q-Learning based on existing frameworks. Grid search and cross-validation were used to optimize hyperparameters for each model, ensuring the best possible performance. Specific hyperparameters like learning rate, depth, number of neighbors, etc., were tuned based on the dataset and algorithm requirements Models were trained on the training set and evaluated on the validation set to prevent overfitting and ensure generalizability.

4. Model evaluation / Experiments

Performance Metrics and Justification

The evaluation of the models involved assessing their performance using various metrics and experimental setups. Each model was evaluated based on its ability to predict air quality levels and classify water samples as potable or not. The following sections detail the experiments, hypotheses, methods, results, and discussions for each dataset and model.

4.1 Air Quality Prediction

4.1.1 Null Hypothesis

Environmental data does not significantly predict air quality levels.

4.1.2 Material & Methods

Training and Test Split: The dataset was split into 80% training and 20% testing sets.

Algorithms Used: CatBoost, KNN, MIL, MLP, Q-Learning.

Parameter Settings: Hyperparameters were optimized using grid search for each algorithm.

Evaluation Metrics: Accuracy, precision, recall, F1 score, confusion matrix.

4.2 Water Potability Classification

4.2.1 Null Hypothesis

Chemical properties do not determine water potability.

4.2.2 Material & Methods

Training and Test Split: The dataset was split into 80% training and 20% testing sets.

Algorithms Used: CatBoost, KNN, MIL, MLP, Q-Learning.

Parameter Settings: Hyperparameters were optimized using grid search for each algorithm.

Evaluation Metrics: Accuracy, precision, recall, F1 score, confusion matrix.

4.3 Results

4.3.1 Air Quality DATASET

Model	Accuracy	Precision	Recall	F1 Score
CatBoost	0.99	1.00	0.99	0.99
KNN	0.93	0.91	0.93	0.92
MIL	0.99	1.00	0.99	0.99
MLP	0.97	0.96	0.97	0.96
Q-Learning	1.00	1.00	1.00	1.00

4.3.2 Water Potability Dataset

Model	Accuracy	Precision	Recall	F1 Score
CatBoost	0.68	0.67	0.93	0.78
KNN	0.63	0.67	0.81	0.73
MIL	0.67	0.71	0.83	0.76
MLP	0.70	0.70	0.90	0.79
Q-Learning	0.64	0.66	0.85	0.74

5. Discussion of the Results

Model Effectiveness

CatBoost and MLP: Both models demonstrated consistent high performance across both datasets. They excel in handling complex relationships and diverse data types, making them suitable for a wide range of predictive tasks.

MIL: Showed strong results, particularly effective for datasets with set-based labels. Its balanced performance across metrics indicates its robustness in handling complex data structures.

KNN: While KNN performed well on the Air Quality dataset, its effectiveness was limited on the Water Potability dataset. This highlights KNN's sensitivity to feature scaling and parameter selection, which can impact its generalizability.

Q-Learning: Achieved perfect scores on the Air Quality dataset but struggled with the Water Potability dataset. This suggests that Q-Learning might be more suitable for certain types of data or specific tasks, particularly those aligned with reinforcement learning principles.

Model Sensitivity

CatBoost and MLP: Both models are less sensitive to data preprocessing steps, such as feature scaling, and can effectively manage complex and high-dimensional data.

KNN: Highly sensitive to the choice of k and feature scaling, requiring careful tuning and preprocessing to achieve optimal performance.

MIL: Demonstrated robustness across different datasets, showing it can handle variability in data structure and complexity well.

Q-Learning: Its performance variability suggests a need for careful tuning and a well-defined environment to be effective, especially for tasks outside typical reinforcement learning scenarios.

Generalization

Air Quality Dataset: All models generally performed better on this dataset, likely due to the clear and strong relationships between the features and the target variable.

Water Potability Dataset: This dataset posed more challenges, indicating greater complexity and variability in the data. Models like CatBoost and MLP managed to adapt better, showing their capability to generalize well even with more difficult data.

6. Conclusion

This project evaluated various machine learning models on the Air Quality UCI and Water Potability datasets to predict air quality levels and classify water samples as potable or not. CatBoost and MLP emerged as the top performers across both datasets, demonstrating robustness and adaptability to complex and diverse data types. MIL also showed strong potential, particularly for set-based

labels. KNN and Q-Learning, while effective in specific contexts, highlighted the need for careful tuning and preprocessing. Overall, the study underscores the importance of selecting appropriate models based on dataset characteristics to achieve accurate and reliable predictions, thereby enhancing decision-making in various applications.

Contributions

HR00722: Led the implementation and evaluation of CatBoost models and contributed to data preparation, cleaning, and preprocessing tasks.

WA00420: Focused on Q-Learning models and was involved in data preparation, cleaning, and preprocessing.

VC00352: Managed the implementation of Multiple Instance Learning (MIL) models and contributed to data preparation, cleaning, and preprocessing tasks.

JD01755: Worked on k-Nearest Neighbors (KNN) models and participated in data preparation, cleaning, and preprocessing activities.

DS01702: Handled the Multilayer Perceptron (MLP) models and contributed to data preparation, cleaning, and preprocessing tasks.

Each member played an integral role in data preparation, including handling missing values, outlier detection and removal, normalization and scaling, and feature engineering, ensuring the datasets were ready for effective machine learning model training and evaluation.

References

1. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

2. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). https://doi.org/10.1145/2939672.2939785

3. Chollet, F. (2015). Keras. https://keras.io

4. Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018

5. Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

6. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely Randomized Trees. Machine Learning, 63(1), 3-42. https://doi.org/10.1007/s10994-006-6226-1

7. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

8. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95. https://doi.org/10.1109/MCSE.2007.55

9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. http://jmlr.org/papers/v12/pedregosa11a.html

10. Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An Introduction. MIT Press.

11. Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

12. van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13(2), 22-30. https://doi.org/10.1109/MCSE.2011.37