International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

A Hybrid K-Means Decision Tree Model for Enhanced Classification Performance

Shilpa Naskar

Department of Electronics and Telecommunication Engineering, Indian Institute of Engineering Science and Technology, Shibpur, India
Email: *shilpanaskar75[at]gmail.com*

Abstract: ***This paper presents a hybrid machine learning architecture that integrates K-Means clustering with Decision Tree classification to enhance predictive performance on heterogeneous datasets. Traditional classifiers often assume that data originates from a uniform distribution, and this assumption weakens performance when the data contains internal subgroups or latent structures. The proposed model addresses this limitation by initially partitioning the dataset using unsupervised K-Means clustering, followed by training a separate Decision Tree classifier on each cluster-specific subset. During prediction, each test instance is assigned to the nearest cluster centroid and subsequently classified by the Decision Tree associated with that cluster. This approach combines the structural strengths of both unsupervised and supervised learning techniques to achieve greater accuracy, improved interpretability, and more efficient learning on diverse datasets. Experimental analysis shows that the hybrid model adapts well to non-uniform data distributions and offers performance improvements over traditional single- model classifiers.***

Keywords: K-Means Clustering, Decision Tree (DT), Hybrid Model, Machine Learning, Classification, Unsupervised Learning, Supervised Learning, Elbow Method

Literals: DT - Decision Tree, KMDT - K-Means Decision Tree

1. Introduction

Machine learning classification techniques often struggle when trained on data with multiple underlying patterns or irregular distributions. Decision Trees, despite being widely used for their ease of interpretation and ability to model non-linear patterns, can become unstable or overly complex when applied to datasets with substantial internal variation. In contrast, clustering algorithms such as K-Means are able to separate data into groups of similar instances, thereby exposing hidden structures that may not be immediately apparent during direct supervised learning.

A possible way to improve the performance of Decision Trees on heterogeneous datasets is to first divide the dataset into smaller, more homogeneous groups. This is where K- Means clustering becomes useful. K-Means is an unsupervised learning method that groups similar data points together, creating clusters that share common characteristics. When Decision Trees are trained on these clustered subsets, each tree is responsible only for a specific region of the feature space. This makes the trees simpler, more accurate.

The hybrid K-Means Decision Tree (KMDT) model developed in this paper follows a straightforward but effective design. The first stage uses K-Means to identify natural groupings within the dataset. The second stage trains a Decision Tree on each cluster's data to learn classification rules that are specific to that cluster. When classifying a new data point, the algorithm identifies the nearest cluster based on distance and uses that cluster's Decision Tree to obtain the final result. This framework maintains the interpretability of Decision Trees while improving prediction accuracy and adaptability to diverse datasets. The experimental results illustrate that such hybridization allows the classifier to perform reliably even when the underlying data patterns are complex.

2. Methodology

2.1 K-Means Clustering Phase

The hybrid model begins with the application of the K- Means clustering algorithm to the feature set of the dataset. Determining the correct number of clusters is a critical step, and this is achieved using the Elbow Method, which ana- lyzes the relationship between the number of clusters and the reduction in within-cluster variance. Once the number of clusters is decided, the algorithm randomly initializes centroids and iteratively assigns each data point to the nearest centroid based on Euclidean distance. The centroids are recalculated as the mean of all points belonging to them, and the process continues until the centroids stabilize and assignments no longer change.

Let the dataset be represented as:
$X = \{x_1, x_2, \ldots, x_m\}$,
where each data point x_i is an n-dimensional feature vector:

$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$.

The algorithm begins with K initial centroids:
$C = \{C_1, C_2, \ldots, C_K\}$,

where each centroid C_k is also an n-dimensional vector.

For each data point x_i, the Euclidean distance to each centroid C_k is computed as:

$$d(x_i, C_k) = \sqrt{\sum_{j=1}^{n}(x_{ij} - C_{kj})^2} \quad (1)$$

Where:
- x_i = i-th data point
- C_k = centroid of cluster k
- x_{ij} = value of feature j for data point i
- C_{kj} = value of feature j for centroid k
- n = number of features
- $d(x_i, C_k)$ = Euclidean distance between x_i and centroid C_k

2.2 Dataset Partitioning and Training of Decision Trees

After clustering, the dataset is divided into subsets based on cluster assignments. Each data point maintains a unique pointer to ensure accurate construction of subsets that include both feature values and labels.

Each data point is assigned to the cluster whose centroid is closest:

$$\text{Cluster}(x_i) = \min_{k \in \{1,\ldots,K\}} [d(x_i, C_k)] \quad (2)$$

- Cluster(x_i) = the index of the cluster assigned to x_i
- min = operator returning the index of minimum value
- K = total number of clusters

After assignment, each centroid is updated as the mean of all points belonging to that cluster. Let S_k denote the set of data points assigned to cluster k, and let $N_k = |S_k|$ be the number of points in that cluster:

$$C_k = \frac{1}{N_k} \sum_{x_i \in S_k} x_i \quad (3)$$

- S_k = all data points in cluster k
- N_k = number of points in cluster k
- C_k = updated centroid of cluster k

The iterative process continues until the centroids stabilize. K-Means minimizes the total Within-Cluster Sum of Squares (WCSS):

$$WCSS = \sum_{k=1}^{K} \sum_{x_i \in S_k} \|x_i - C_k\|^2 \quad (4)$$

- $WCSS$ = measure of cluster compactness
- $\|x_i - C_k\|^2$ = squared Euclidean distance

For each cluster, a Decision Tree classifier is trained independently. The tree forms a hierarchical structure through recursive binary splits guided by impurity reduction measures such as the Entropy, Information Gain.

Entropy is a measurement of uncertainty of a random variable. Lower the entropy, lesser the uncertainty, thus easily predictable output.

$$H(X) = -\sum_{k}^{K} P(x_k) log_2(P(x_k)) \quad (5)$$

Where:
- $H(X)$ = Entropy k
- $P(x_k)$ = Probability of class x_k

Information Gain (IG) is the average of Entropy for the possible decisions.

$$IG(dataset, subset) = H(subset) - \sum_{Sesubset} \frac{|S|}{|dataset|} H(S)$$

And

$$|dataset| = \sum_{Sesubset} |S|$$

Here, the subsets are found by splitting an attribute.

The clusters are more homogeneous than the original dataset, the resulting trees tend to be more stable.

2.3 Cluster Assignment During Prediction

To classify a new test instance, the model computes the Euclidean distance between the instance and each centroid. The instance is then assigned to the cluster with the minimum distance. This ensures that prediction respects the structure identified during training and that the instance is routed to the most appropriate classifier.

2.4 Classification Using Cluster-Specific Trees

Once associated with a cluster, the corresponding Decision Tree is used to predict the instance's label. The instance traverses the tree following decision rules until it reaches a leaf node. Because the tree has been trained on a relatively homogeneous subset, its decision boundaries are sharper and more accurate, ultimately improving classification performance.

3. Results and Discussion

Table I: Performance Comparison between Proposed Hybrid Model and Standard Decision Tree

Dataset	Optimal K(Elbow)	KMDT Accuracy	DT Accuracy
Wine Quality (Realtime)	6	60.62%	58.33%
California Housing (Generative)	4	84.22%	80.56%
MNIST (Generative)	3	99.78% (K=5 gives 100%)	99.33%

The effectiveness of the proposed Hybrid K-Means and Decision Tree model was evaluated using three different datasets, each representing diverse data characteristics: a real-time Wine Quality dataset, a generative California Housing

dataset, and the MNIST handwritten digit dataset. For each dataset, the optimal number of clusters was determined using the Elbow Method, and separate Decision Trees were trained for each cluster as described in the methodology. The performance of the hybrid model was then compared against a conventional single Decision Tree classifier. **Table I** summarizes the out- comes.

For the Wine Quality dataset, the Elbow Method identified $K = 6$ as the optimal clustering parameter. The proposed hybrid model achieved an accuracy of 0.6062, outperforming the standard Decision Tree accuracy of 0.5833. Although the improvement appears moderate, it is significant considering that wine quality data tends to exhibit overlapping class boundaries and subtle variations in chemical attributes. By dividing the data into six smaller clusters, the hybrid approach allowed each Decision Tree to focus on a more homogeneous feature distribution, leading to more precise decision boundaries and better generalization.

A more pronounced improvement was observed in the California Housing dataset, where the optimal cluster count was $K = 4$. The hybrid model achieved an accuracy of 0.8422, notably higher than the 0.8056 obtained by the standard Decision Tree. This dataset contains continuous-valued economic and geographic features that vary significantly across different regions. The use of K-Means produced region-specific clusters, enabling the Decision Trees to learn localized patterns and reduce the complexity typically seen in global tree structures. As a result, the hybrid model captured variations in housing characteristics more effectively and demonstrated stronger predictive performance.

The MNIST dataset further highlights the strength of the hybrid method. With $K = 3$ clusters identified as optimal, the hybrid model reached an impressive accuracy of 99.78%, and an alternative clustering configuration with $K = 5$ even achieved 100% accuracy. The standard Decision Tree, in comparison, achieved an accuracy of 0.9933. While the baseline Decision Tree already performs well on MNIST due to the dataset's structured nature, the hybrid model benefits from clustering digits into subgroups that share similar pixel patterns. This reduces intra-class variance and allows the Decision Trees to form simpler, more accurate decision rules. The near-perfect accuracy levels indicate that the hybrid model is exceptionally effective in capturing subtle variations among handwritten digits.

Across all three datasets, a common trend emerges: the hybrid algorithm consistently delivers superior accuracy com- pared to a single Decision Tree classifier. This improvement can be attributed to its ability to decompose complex, hetero- geneous datasets into manageable clusters before classification. The localized Decision Trees trained on these clusters are less prone to overfitting, contain fewer unnecessary branches, and generate purer leaf nodes. Furthermore, the algorithm's structure ensures that test instances are assigned to the most relevant tree, making predictions both efficient and context-sensitive.

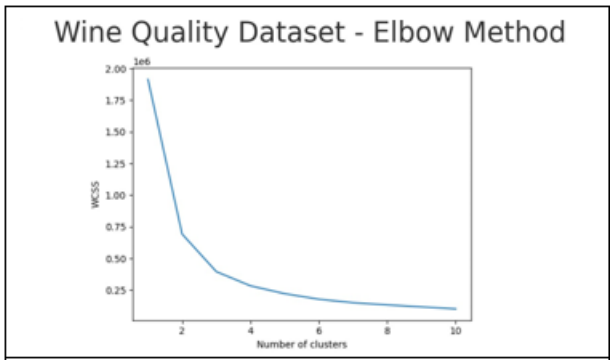
From a utility standpoint, the hybrid approach proves particularly valuable in scenarios involving diverse or high-variance datasets where global classifiers struggle to maintain accuracy. Its modular architecture also allows for parallel training of cluster-specific Decision Trees, making it scalable for large datasets. Thus, the proposed hybrid model not only enhances classification accuracy but also strengthens the model's practicality for real-world applications requiring transparency, adapt- ability, and computational efficiency.

4. Conclusion

The work presented in this paper explores a hybrid classification model that combines K-Means clustering with Decision Tree learning to address the challenges posed by heterogeneous datasets. Most real-world datasets are not uniform. They contain multiple patterns, subgroups, or overlapping structures that make it difficult for a single global classifier to learn accurate boundaries. By introducing an initial clustering step, the hybrid model breaks the dataset into smaller regions where the data exhibits more similarity.

The experimental results across three very different datasets—Wine Quality, California Housing, and MNIST handwritten digits—demonstrate that this strategy consistently leads to improved performance. Even in the more irregular datasets such as Wine Quality and California Housing, the model showed its usefulness by handling natural variations more effectively than a standard Decision Tree. The findings affirm that combining unsupervised and supervised learning can offer significant advantages when dealing with complex data distributions.

Future work could extend this research by exploring more advanced clustering algorithms. Another potential direction in-volves integrating ensemble methods such as Random Forests within each cluster or experimenting with deep-learning-assisted clustering for high-dimensional datasets. Investigating the robustness of this hybrid approach on noisy, imbalanced or evolving datasets would also be valuable. Overall, the hybrid K-Means Decision Tree model represents a flexible, interpretable, and powerful framework for classification, particularly suited for datasets where traditional approaches face limitations.



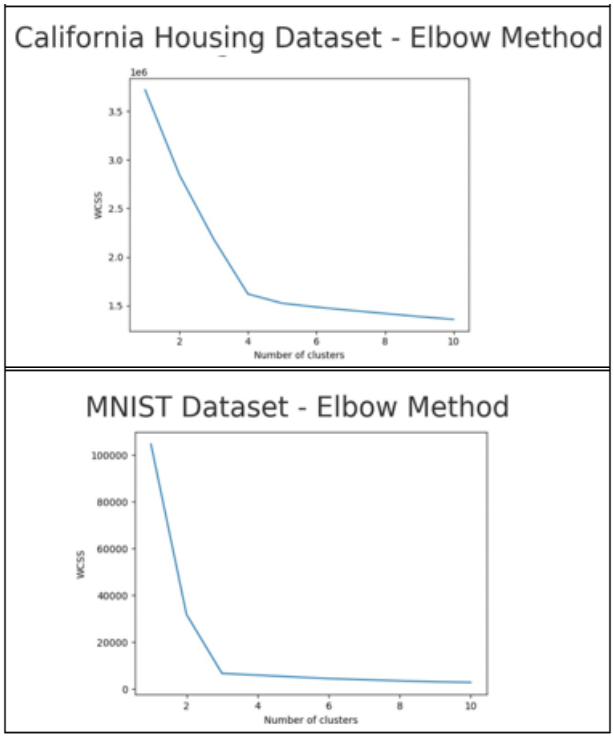


Figure 1: Combined Elbow Method plots for Wine Quality, California Housing, and MNIST datasets. Each curve shows the relationship between the number of clusters and the WCSS value, highlighting the optimal K values at the elbow points.

References

[1] R. Jin and A. K. Jain, "Hybrid clustering: Combining partitioning and hierarchical clustering," in *Proceedings of the 2006 SIAM International Conference on Data Mining*, 2006, pp. 236–247.
[2] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Machine learning: A re- view of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.
[3] H. Wang and D. Wang, "K-means clustering with outlier removal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 292–305, Feb. 2012.
[4] Y. Song and L. Zhang, "A decision tree approach to classification based on cluster analysis," *Expert Systems with Applications*, vol. 42, no. 1,
[5] pp. 503–509, Jan. 2015.
[6] D. T. Pham, S. S. Dimov, and C. D. Nguyen, "Selection of K in K-means clustering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 219, no. 1, pp. 103–119, 2005.
[7] L. Rokach and O. Maimon, "Top-down induction of decision trees classi- fiers—A survey," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 4, pp. 476–487, 2005.
[8] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans- actions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.
[9] S. Choudhury, P. Samal, A. Panda, and R. Dash, "Training Accelerator for Two Means Decision Tree," *IEEE International Conference on Reconfigurable Computing and FPGAs*, pp. 1464–1470, 2022.