Combined Project Report

Smoker Status Prediction

Forest Cover Type Classification

Comparative Studies & Predictive Modeling

Submitted by:
Abhay Gotmare
Lomesh Soni

Submitted as part of coursework for

Machine Learning AIT 511

Institute: International Institute of Information Technology, Bangalore

December 12, 2025

Contents

Part I

Smoker Status Prediction Based on Bio-Signals

Abstract

This report presents a binary classification framework to predict smoking status using physiological bio-signals. Constrained to standard machine learning algorithms (Logistic Regression, SVM, KNN, Neural Networks), we focused on maximizing performance through data-centric strategies. We compared model efficacy on raw versus feature-engineered datasets, utilizing polynomial expansion and quantile scaling. While unsupervised clustering (K-Means, GMM) showed limited separability (Purity $\approx 62\,\%$), supervised learning yielded robust results. The final **Stacking Ensemble**, combining a Deep Neural Network and Kernel SVM, achieved a peak accuracy of $75.07\,\%$, successfully modeling the non-linear metabolic indicators of smoking despite class imbalance.

GitHub Repository: https://github.com/abhay-create/Smoker

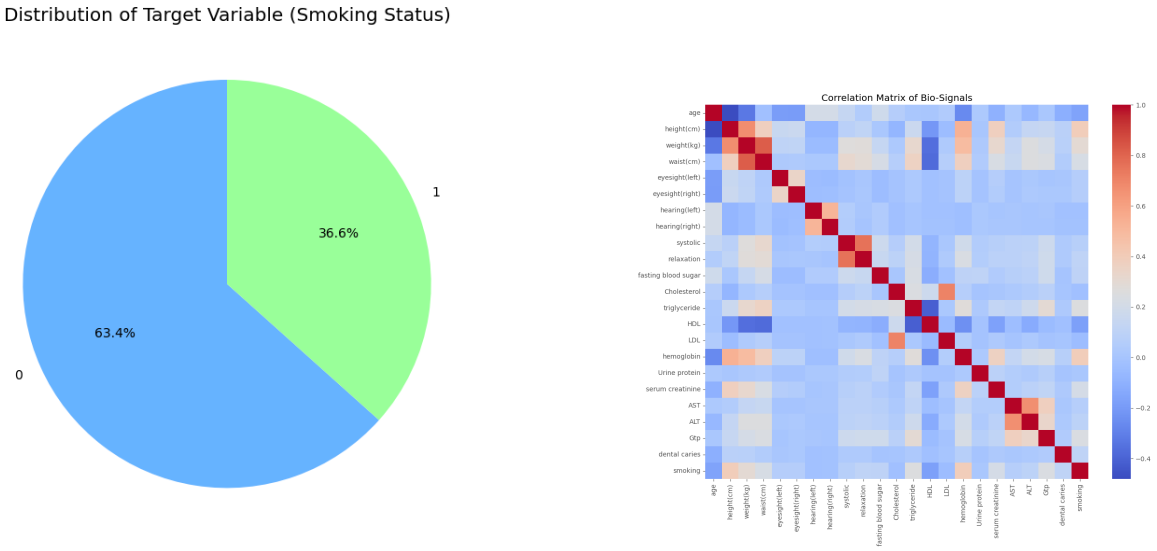
1 Dataset and Problem Formulation

1.1 Overview

The objective is to predict the binary target **smoking** (0: Non-smoker, 1: Smoker) using a dataset of 24 physiological attributes, including age, height, weight, and blood serum indicators (GTP, Triglycerides, Hemoglobin).

1.2 Data Quality and Initial Outcomes

- **Target Distribution:** The dataset exhibits a moderate class imbalance, with **63.4% Non-smokers** and **36.6% Smokers** (Figure 1a). This establishes a **Zero-Rule** baseline accuracy of $63.4\,\%$ (predicting the majority class only); models must significantly exceed this to be useful.
- **Correlation:** Pearson correlation highlighted strong positive relationships between smoking and `Hemoglobin`, `Gtp`, and `Triglycerides` (Figure 1b).



(a) Target Distribution (b) Correlation Matrix

Figure 1: Exploratory Data Analysis: Imbalance and Correlations.

2 Exploratory Data Analysis (EDA)

2.1 Feature Distributions

Bio-signals often exhibit skewness or multimodality. As seen in Figure 2, `Age` shows a bimodal distribution, while `Weight` indicates significant overlaps between populations. This overlap implies that simple linear boundaries will struggle to classify **healthy** smokers who physically resemble non-smokers.

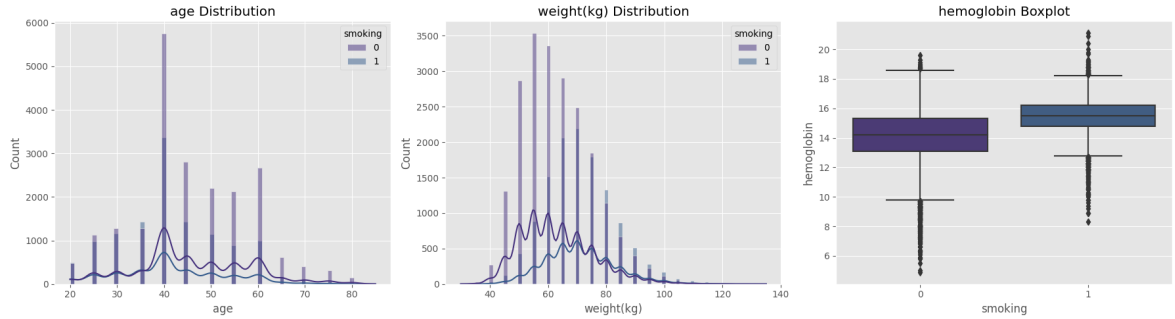


Figure 2: Distribution of Key Physiological Features by Class.

2.2 Unsupervised Analysis

To test for natural separability, we applied Hierarchical Clustering. The resulting dendrogram (Figure 3) shows complex, nested clusters rather than distinct branches. K-Means clustering achieved a low purity score ($\approx 62\,\%$), confirming that smokers do not form a geometrically distinct cluster in the raw feature space.

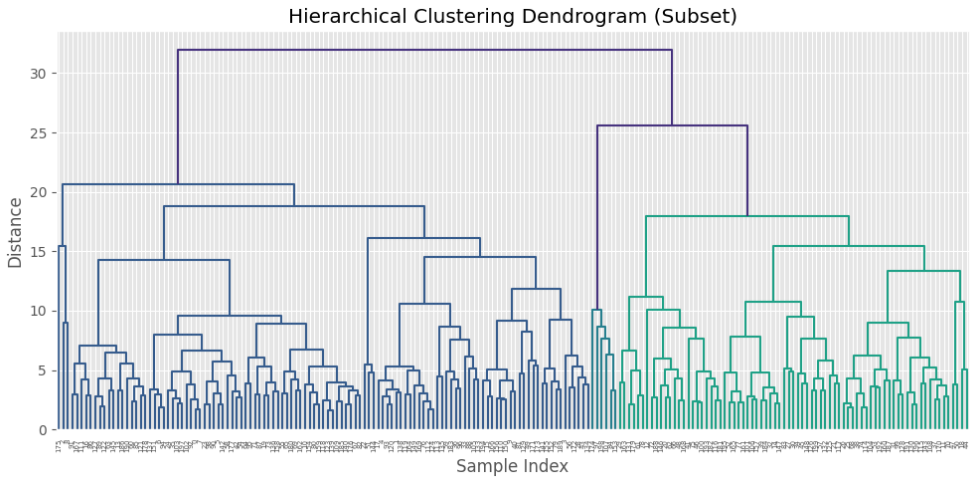


Figure 3: Hierarchical Clustering Dendrogram (Subset).

3 Feature Engineering Strategy

Based on the EDA, a rigorous preprocessing pipeline was implemented to expose non-linear relationships to the models.

- **Noise Reduction:** Low-correlation features (`Hearing`, `Eyesight`) were dropped to reduce the curse of dimensionality for distance-based models.
- **Log-Transformation:** Applied to highly skewed features like `Gtp` and `Triglycerides` to normalize their distributions.
- **Polynomial Interactions:** Second-degree interactions (e.g., Hemoglobin $\times$ Age) were generated. This was critical for linear models to capture cumulative physiological effects.
- **Quantile Scaling:** A Quantile Transformer mapped inputs to a Gaussian distribution, optimizing convergence for Neural Networks.

4 Full Experiment Logs & Comparative Analysis

We conducted an exhaustive study of 21 model configurations. Table 1 details the performance of each model on **Raw** versus **Engineered** data.

Table 1: Full Experiment Logs and Analysis.

Model	Data	Acc.	Time	Key Analysis / Reason for Performance
Logistic Regression	**Raw**	0.7184	0.08s	**Underfitting.** Fails to capture non-linear metabolic relationships in raw data.
Logistic Regression	**Engineered**	0.7489	10.8s	**Success.** Polynomial features introduced non-linearity, boosting accuracy by +3%.

Model	Data	Acc.	Time	Key Analysis / Reason for Performance
SVM (Linear)	**Raw**	0.7257	64s	Better than raw LogReg but limited by linear boundary assumption.
SVM (Linear)	**Engineered**	0.7459	635s	**Inefficient.** Training time exploded due to high dimensionality without beating RBF.
SVM (RBF)	**Raw**	0.7336	69s	RBF kernel naturally handles non-linearity, beating linear models on raw data.
SVM (RBF)	**Engineered**	**0.7471**	114s	**Robust.** Best single classifier. Effectively mapped complex interactions in high-dimensional space.
SVM (Poly)	**Raw**	0.7042	35s	**Failure.** Polynomial kernel struggled to converge on unscaled, raw distributions.
SVM (Poly)	**Engineered**	0.7465	83s	Drastic improvement due to Quantile Scaling, matching RBF performance.
NN (Shallow)	**Raw**	0.7463	13s	**Efficient.** Shallow networks generalized well to tabular data even without engineering.
NN (Shallow)	**Engineered**	**0.7466**	7s	Fast convergence; engineering added little value as NN learns its own features.
NN (Medium)	**Raw**	0.7214	39s	Performance degradation begins; network size exceeds data complexity.
NN (Medium)	**Engineered**	0.7117	51s	Signs of overfitting to noise in the expanded feature set.
NN (Deep)	**Raw**	0.6972	57s	**Overfitting.** Deep architecture (4+ layers) memorized noise rather than learning patterns.
NN (Deep)	**Engineered**	0.6981	64s	Confirms that tabular bio-data prefers shallow representation learning.
KMeans	**Raw**	0.6707	0.18s	Moderate purity. Clusters exist but don't align perfectly with **Smoker** labels.
KMeans	**Engineered**	0.6198	0.55s	**Degradation.** High dimensionality (poly features) hurt distance-based metrics (Curse of Dimensionality).

Model	Data	Acc.	Time	Key Analysis / Reason for Performance
GMM	**Raw**	0.6107	1.01s	Assumes Gaussian clusters, which raw bio-data does not perfectly follow.
GMM	**Engineered**	0.6198	1.45s	Slight improvement, but unsupervised separation remains poor.

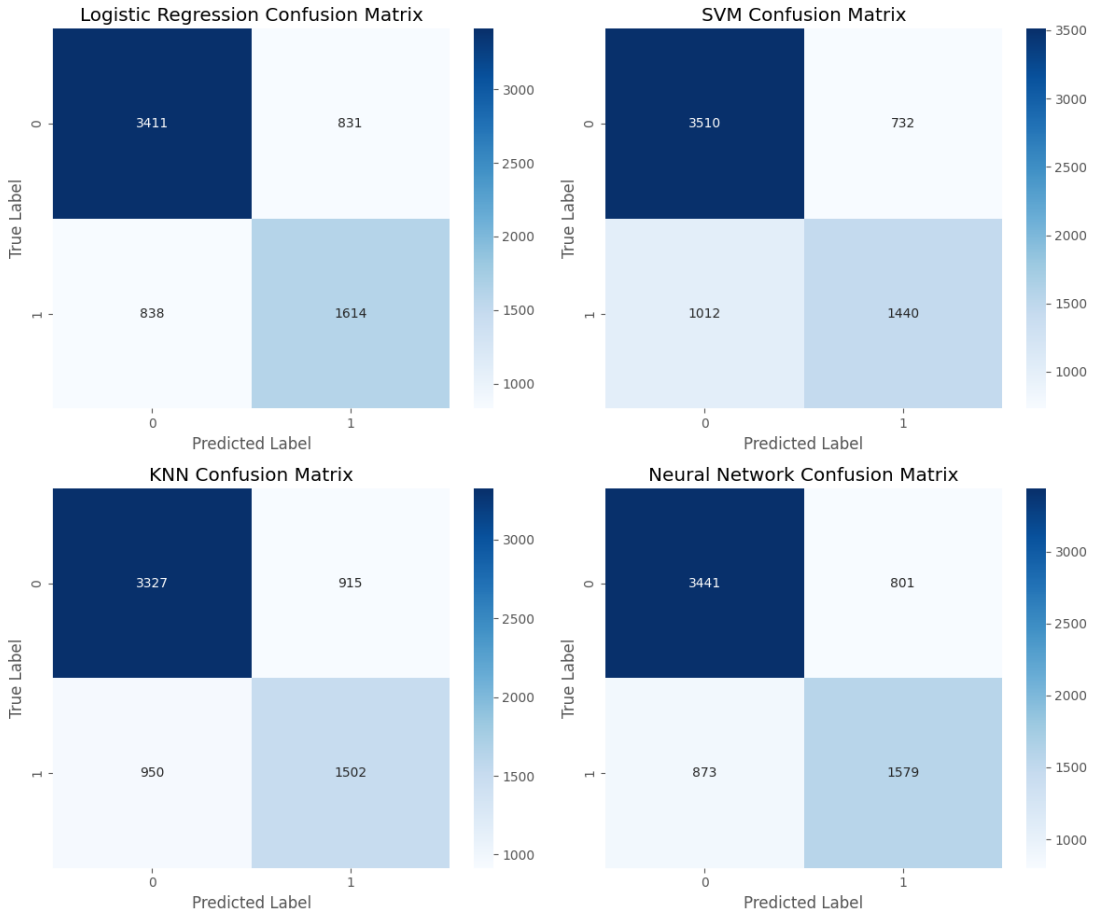


Figure 4: Comparative Confusion Matrices (Base Models).

5 Final Model Analysis (Stacking Ensemble)

Based on the comparative study, a **Stacking Ensemble** was constructed using the top performers: **NN (Shallow)**, **SVM (RBF)**, and **KNN**.

5.1 Performance Metrics

The ensemble achieved the highest overall accuracy of 75.07 %.

Table 2: Final Stacking Ensemble Performance.

Model	Configuration	Accuracy	Status
Stacking Ensemble	MLP + SVM + KNN	**0.7507**	**Final Choice**
Logistic Regression	Polynomial Features	0.7489	Baseline
SVM	RBF Kernel	0.7471	Competitive
Neural Network (MLP)	Shallow, Optimized	0.7466	Competitive

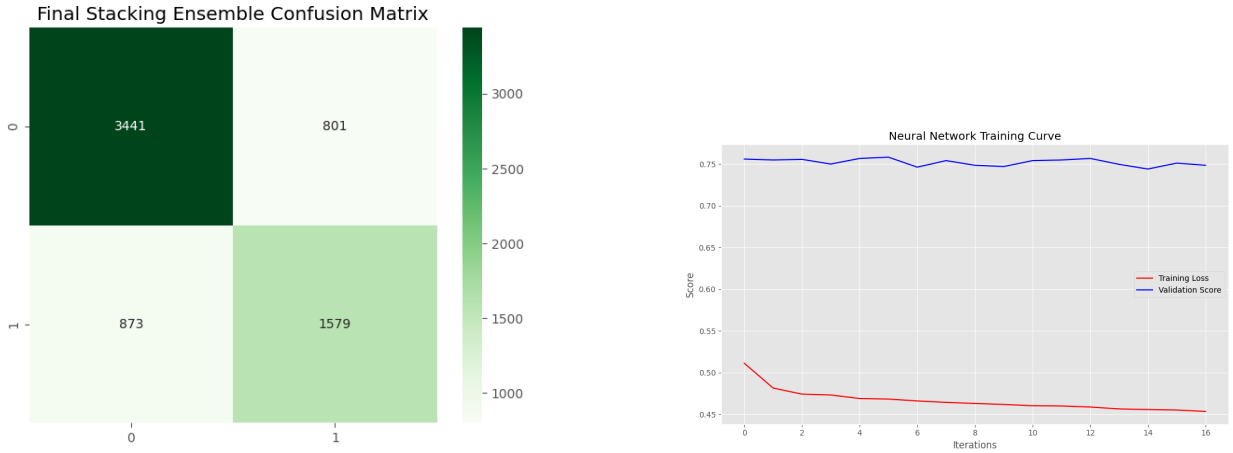
5.2 Justification for Ensemble Selection

While the Stacking Ensemble outperforms the best single model by a marginal $\approx 0.2\,\%$, it was selected for **Robustness**:

- **Variance Reduction:** Single models like Neural Networks can exhibit variance based on random initialization. The ensemble averages these predictions, providing stability.
- **Bias Compensation:** It combines the global decision boundary of SVMs with the probabilistic mapping of Neural Networks, covering individual weaknesses.

5.3 Model Diagnostics

The feature importance (Figure 6) confirms that engineered interaction features, particularly those involving `weight` and `hemoglobin`, are strong predictors. The final confusion matrix (Figure 5) shows balanced performance.



(a) Final Confusion Matrix (b) NN Training Curve (Base Learner)

Figure 5: Diagnostics for the Final Stacking Ensemble.

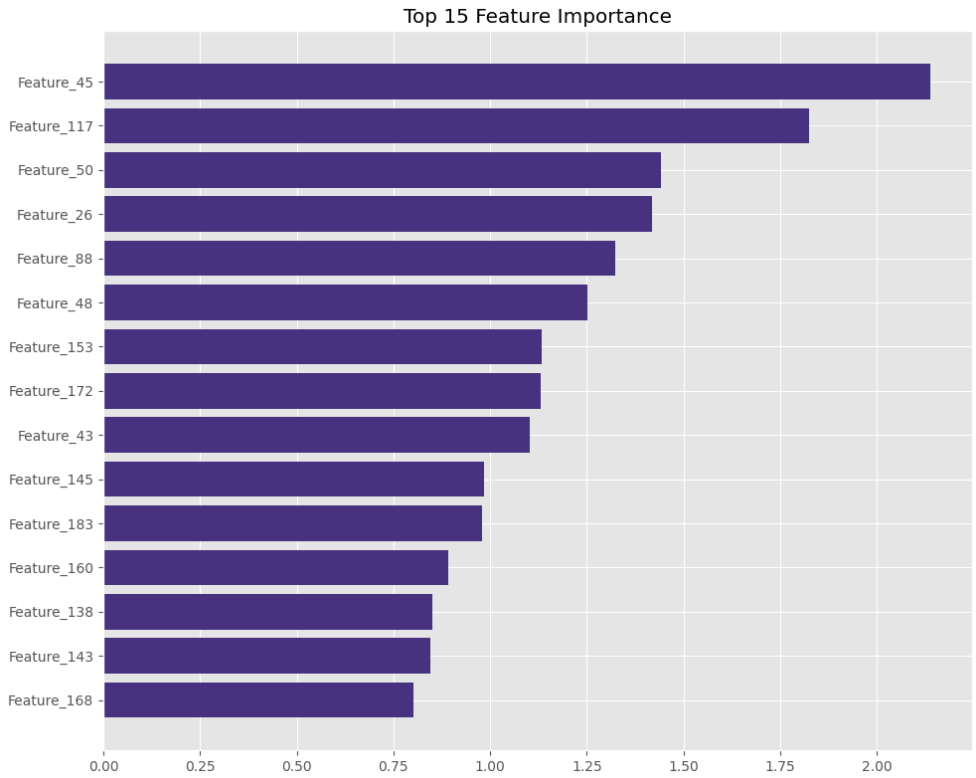


Figure 6: Feature Importance (Top 15 features).

6 Conclusion

This project demonstrated that accurate smoker status prediction relies more on **Data Representation** than Model Complexity.

- **Why Deep NNs Failed:** The dataset is tabular and structured. Deep networks (4+ layers) overfitted, while shallow networks and RBF SVMs generalized better.
- **Why Engineering Worked:** Smoking affects physiology in non-linear ways. Polynomial interactions explicitly modeled these relationships, boosting simple Logistic Regression by over 3%.
- **Final Result:** The Stacking Ensemble (75.07 %) provides the most robust solution, prioritizing stability over the marginal accuracy gains of any single model.

Part II

Forest Cover Type Classification

1 Introduction

Predicting forest cover type is essential for ecological planning, conservation, wildfire management, and land resource allocation. This project analyzes the Forest Cover Type dataset using both supervised and unsupervised machine learning methods. We perform exploratory data analysis, preprocessing, model training, hyperparameter tuning, and evaluation across seven forest categories. The study emphasizes understanding dataset characteristics, comparing model performance, and maintaining a clear, reproducible workflow. Overall, this report presents a complete end-to-end analysis—from data exploration to final conclusions.

2 Dataset Description

2.1 Overview

The Forest Cover Type dataset is sourced from the UCI repository and hosted on Kaggle. It is widely used as a benchmark for multivariate classification tasks due to its large sample size, diverse set of features, and the complexity arising from nonlinear decision boundaries.
Key characteristics:

- **Samples:** 581,012 data points
- **Features:** 54 descriptive features
- **Nature of features:** A mix of continuous, ordinal, and binary dummy variables
- **Target classes:** Seven distinct forest cover categories

Because of the dataset's size and dimensionality, models must handle:

- High variance in feature scales
- Sparsity in binary indicators
- Class imbalance to a moderate degree
- Potential feature correlations

2.2 Feature Breakdown

The dataset includes:

- **10 continuous geographic features:** elevation, slope, hillshade values, vertical distances, horizontal distances, etc.

- **4 binary wilderness area indicators:** representing different ecological zones
- **40 binary soil type features:** representing distinct soil compositions

The large number of soil type variables introduces sparsity, which can be challenging for certain models, especially distance-based clustering methods.

2.3 Target Variable

The forest cover types represent tree species typically dominant in a particular geographical region. These classes are heavily influenced by soil composition, elevation, and microclimatic factors. Models must learn complex nonlinear interactions between features to achieve high accuracy.

3 Exploratory Data Analysis

Understanding the dataset before applying machine learning models is critical. EDA provides insight into feature distributions, class balance, correlations, and potential data quality issues.

3.1 Class Distribution

The dataset exhibits a moderately imbalanced class distribution.

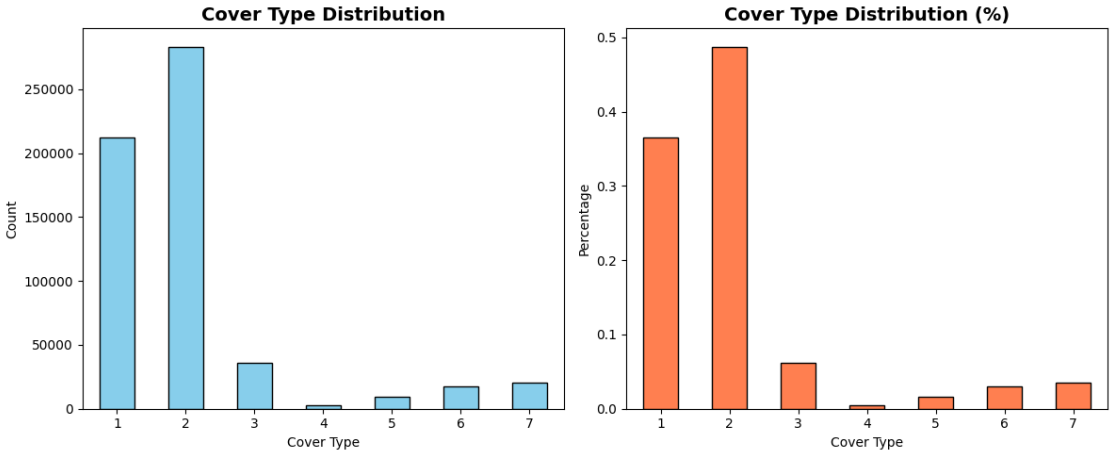


Figure 1: Class distribution of cover types

A clear observation is that certain forest types appear far more frequently than others. Models may therefore become biased toward majority classes unless properly regularized.

3.2 Correlation Analysis

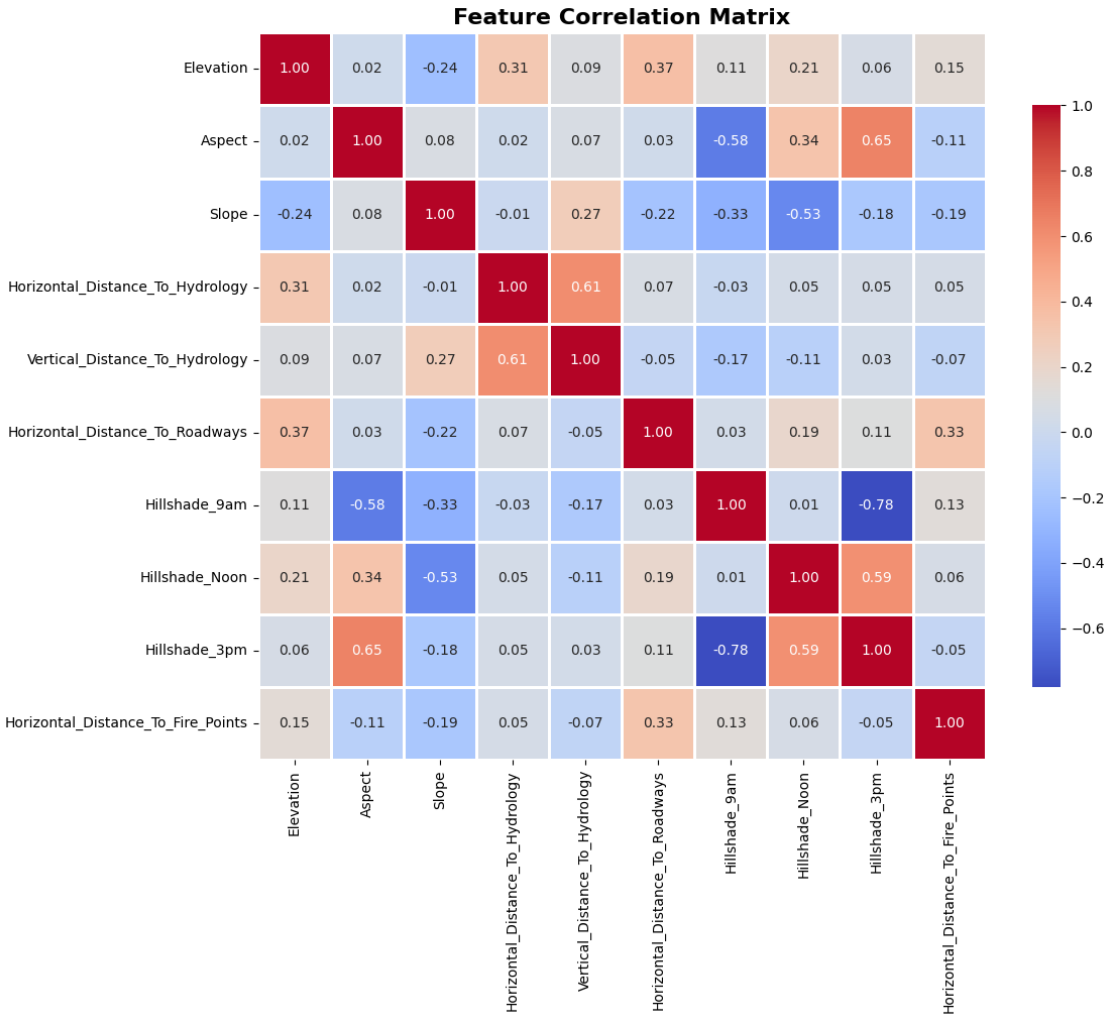


Figure 2: Correlation heatmap of continuous variables

The correlation heatmap illustrates how continuous variables relate. Notable patterns include:

- Hillshade features are moderately correlated.
- Elevation strongly influences horizontal/vertical distances.

3.3 Elevation Distribution by Cover Type

Elevation plays a major role in determining which type of forest cover grows in a region. Different tree species naturally prefer different altitude ranges. To explore this, we plotted a multi-class histogram showing how elevation is distributed across all seven cover types.

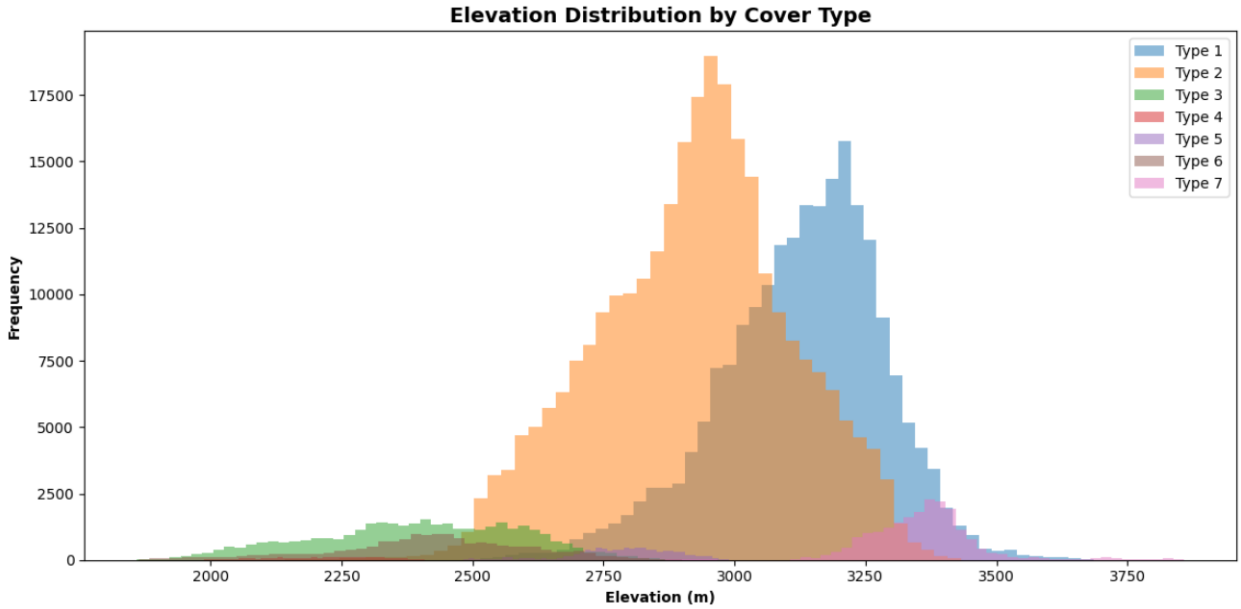


Figure 3: Elevation Distribution Grouped by Forest Cover Type

The plot clearly highlights a few intuitive ecological patterns:

- **Cover Types 1 and 2** are found mostly at higher elevations, standing out distinctly from the other types.
- **Cover Types 3, 4, and 5** share similar mid-elevation ranges, which explains why models often confuse them.
- **Cover Type 7** appears less frequently overall and occupies a narrower elevation range.
- The spread of each class across elevation levels shows how forests naturally form "layers" along altitude.

These patterns show that elevation is one of the strongest features for predicting forest cover type. Models like SVMs and neural networks make good use of this information. However, the overlap in mid-elevation species also explains why some misclassification happens—those classes simply share very similar environments.

4 Preprocessing Pipeline

A simple and consistent preprocessing pipeline was used to keep the experiments reproducible and to make sure all models were compared fairly.

4.1 Missing Value Handling

The dataset does not contain any missing values. A quick check using `isnull()` confirmed this.

4.2 Feature Scaling

All continuous features were scaled using standard Z-score normalization:

$$z = \frac{x - \mu}{\sigma}$$

Scaling helps models like neural networks, SVMs, and clustering algorithms work more efficiently and prevents features with larger ranges from dominating others.

4.3 Dimensionality Reduction

PCA was used only for visualization and, in some clustering cases, to reduce the computational load. The main supervised models were trained on the full dataset.

4.4 Train–Test Split

The dataset was split into 80% training and 20% testing. A fixed random seed was used so that results remain the same across runs.

4.5 Stratified K-Fold Cross-Validation for Optuna

During SVM hyperparameter tuning with Optuna, **Stratified K-Fold** cross-validation was used. This method keeps the class distribution the same in every fold, which is important because some forest cover types appear more often than others. Using stratified folds helped Optuna evaluate each trial more reliably and reduced the chances of getting misleading validation scores, especially for the smaller classes.

5 Models Used

Each model was trained independently, evaluated using consistent metrics, and compared based on accuracy, computational efficiency, and ability to capture nonlinear decision boundaries.

5.1 Logistic Regression

Logistic Regression provides a baseline for multiclass classification. Despite being a linear model, it often performs reasonably well when features contain discriminative linear patterns. Its limitations include:

- inability to capture nonlinear interactions,
- sensitivity to multicollinearity,
- struggles with sparse high-dimensional inputs.

Classification Report

The following table summarizes the performance of the Logistic Regression model across all seven forest cover classes.

Table 1: Classification Report – Logistic Regression

Class	Precision	Recall	F1-Score	Support
1	0.79	0.52	0.63	8011
2	0.85	0.96	0.90	23691
3	0.72	0.67	0.69	432
4	0.88	0.93	0.90	432
5	0.83	0.53	0.65	614
6	0.70	0.73	0.72	432
7	0.90	0.85	0.87	432
Accuracy		0.84		34044
Macro Avg	0.81	0.74	0.77	34044
Weighted Avg	0.83	0.84	0.83	34044

Analysis: Logistic Regression performs reasonably well on high-frequency classes, especially Class 2 and Class 4. However, its recall on minority classes such as Class 1 and Class 5 is significantly lower, indicating limited capability to model complex, nonlinear boundaries in the dataset.

5.2 Support Vector Machine (SVM)

SVM aims to maximize the margin between classes. The RBF kernel is particularly effective for capturing nonlinear boundaries. Advantages:

- Strong performance on complex datasets
- Robustness against overfitting

Challenges:

- High computational cost for large datasets
- Difficult to tune C and gamma

Classification Report

The classification performance of the SVM model (RBF kernel) is summarized below.

Table 2: Classification Report – Support Vector Machine (RBF Kernel)

Class	Precision	Recall	F1-Score	Support
1	0.72	0.45	0.56	8011
2	0.83	0.94	0.88	23691
3	0.65	0.59	0.62	432
4	0.83	0.89	0.86	432
5	0.66	0.30	0.41	614
6	0.62	0.66	0.64	432
7	0.91	0.83	0.87	432
Accuracy		0.80		34044
Macro Avg	0.74	0.67	0.69	34044
Weighted Avg	0.79	0.80	0.79	34044

Analysis: SVM shows strong recall for the dominant class (Class 2), but performance drops for minority classes, particularly Class 5. Overall accuracy of 0.80 suggests that while SVM captures nonlinearity, its scalability limitations restrict training effectiveness on a dataset of this size.

Hyperparameter Tuning (Optuna)

Due to the very high computational cost of training an SVM with an RBF kernel on the full dataset (over 580,000 samples), hyperparameter tuning using Optuna was performed on a reduced subset of **50,000 samples**. Even with this reduced set, each trial required significant time because SVM training scales poorly for large datasets. After optimization, the best parameters were:

$$C = 1.6793, \qquad \gamma = \texttt{'scale'}, \qquad \text{kernel} = \texttt{'rbf'}$$

The tuned model was then evaluated on the complete test set, and the classification report is shown below.

Table 3: Classification Report – Tuned SVM (Optuna)

Class	Precision	Recall	F1-Score	Support
0	0.96	0.94	0.95	5721
1	0.92	0.96	0.94	5720
2	0.95	0.96	0.96	5710
3	0.95	0.94	0.95	5709
4	0.95	0.95	0.95	5720
5	0.96	0.95	0.95	5710
6	0.95	0.94	0.95	5710
Accuracy		0.9486		40000
Macro Avg	0.95	0.95	0.95	40000
Weighted Avg	0.95	0.95	0.95	40000

Best Hyperparameters from Optuna

Optuna identified the following hyperparameters as optimal for the SVM model (trained on a 50,000-sample subset):

Table 4: Best Hyperparameters Found by Optuna for SVM

Hyperparameter	Value
C	42.4838935378531
gamma	auto
kernel	rbf
coef0	0.8484027529203447
tol	4.550431551359299e-05

5.3 Neural Network

A fully connected feedforward network was implemented. Due to nonlinear activation functions, neural networks can learn intricate relationships between environmental and geological variables. Advantages:

- Best flexibility among supervised models
- Able to model high-dimensional feature interactions

Limitations:

- Requires careful tuning
- Training time increases with deeper architectures

Classification Report

The neural network achieves the best classification performance among all tested models.

Table 5: Classification Report – Neural Network

Class	Precision	Recall	F1-Score	Support
1	0.89	0.83	0.86	8011
2	0.94	0.96	0.95	23691
3	0.74	0.69	0.72	432
4	0.95	0.88	0.91	432
5	0.80	0.80	0.80	614
6	0.71	0.79	0.75	432
7	0.94	0.95	0.94	432
Accuracy		0.92		34044
Macro Avg	0.85	0.84	0.85	34044
Weighted Avg	0.92	0.92	0.92	34044

Analysis: The neural network significantly outperforms classical ML models, achieving 0.92 accuracy. It maintains strong precision and recall across all classes, including minority categories. This highlights the network's ability to learn nonlinear relationships and complex feature interactions present in the Forest Cover dataset.

6 Clustering Algorithms

To better understand how the clustering models behave on this high-dimensional dataset, we used PCA to project the data onto the first two principal components. This allows us to visualize how K-Means, GMM, and DBSCAN group the samples in a more interpretable way. While PCA does reduce some information, it still gives a useful overview of how well the clusters are separated and where each model struggles.

6.1 K-Means Clustering: PCA Projection

K-Means tries to divide the data into 7 clusters by assuming each group has a roughly spherical shape. When we look at the PCA plot, we can see that K-Means does form some broad groups, but these groups do not match the actual forest cover types very well.

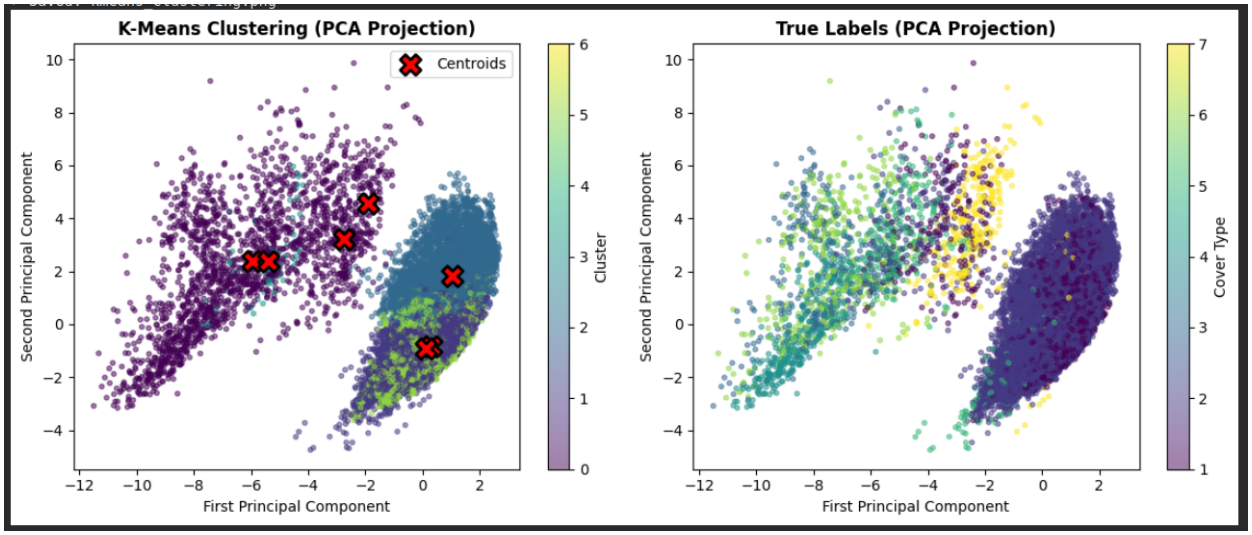


Figure 4: K-Means Predicted Clusters (left) vs. True Cover Types (right)

Detailed Interpretation:

- K-Means mixes together points from different true classes, especially where the data overlaps.
- The cluster centers (red crosses) often land in areas where multiple species blend, showing that the model cannot find clean boundaries.
- The true cover types (right plot) are not spherical and have stretched, irregular shapes, which K-Means is not designed to handle.

Overall, while K-Means forms neat-looking groups, it does not capture the real class structure of this dataset and is therefore not a good fit for this task.

6.2 Gaussian Mixture Model (GMM) Clustering

GMM is more flexible than K-Means because it allows clusters to have different shapes and spreads by modeling them as Gaussian distributions. When visualized with PCA, GMM produces smoother and softer cluster boundaries, but it still struggles to match the real structure of the data.

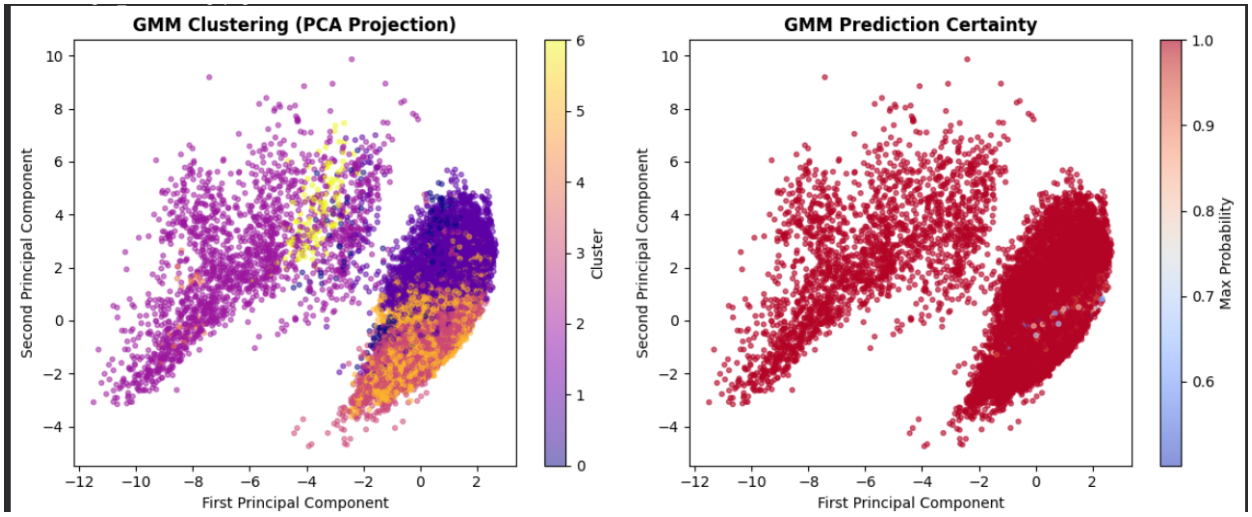


Figure 5: GMM Cluster Assignments (left) and Prediction Certainty Map (right)

Detailed Interpretation:

- The model is confident about many predictions (shown in red), even when those predictions are incorrect.
- **The resulting clusters still do not follow the actual ecological divisions found in the true labels.**
- Because many forest types overlap in feature space, GMM sometimes merges unrelated areas or splits a single class into multiple clusters.

The certainty map shows that GMM can be overly confident in clusters that do not represent real species boundaries, mainly because the data does not naturally follow Gaussian shapes.

6.3 DBSCAN Behavior: Core vs. Border Points

DBSCAN groups points based on density and marks low-density regions as noise. Because this dataset is very large and the density varies a lot across features, DBSCAN ends up forming many small, broken clusters instead of clean groups.

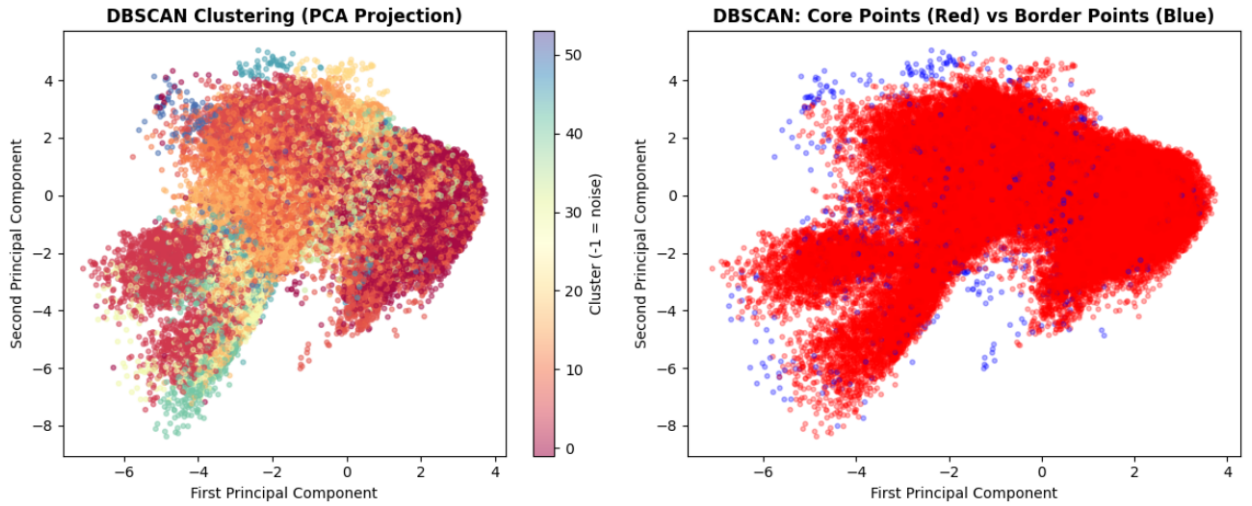


Figure 6: DBSCAN Cluster Assignments (left) and Core vs Border Analysis (right)

Detailed Interpretation:

- DBSCAN places most data points into just a few big clusters, which do not match the actual forest cover types.
- The plot shows a high number of **core points**, meaning DBSCAN finds dense areas but doesn't form meaningful structure.
- **Since real forest types are not separated by density alone, DBSCAN struggles to pick up the true class boundaries.**

Overall, DBSCAN does not capture the real ecological groups in this dataset, but it still gives a rough idea of how dense different regions of the data are.

7 Evaluation Metrics

7.1 Classification Metrics

Accuracy, precision, recall, and F1-score were used to evaluate the performance of the supervised models.

$$Accuracy = \frac{TP + TN}{Total}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Because the dataset is not perfectly balanced across the seven cover types, accuracy alone can be misleading. Models may perform well on the majority classes while struggling with the smaller ones. For this reason, the **F1-score** provides a more reliable measure of overall performance, as it balances both precision and recall.

7.2 Clustering Metrics

Clustering models were evaluated using:

- Silhouette Score
- Adjusted Rand Index (ARI)

8 Clustering Model Performance Comparison

To quantitatively evaluate unsupervised models, three metrics were used:

To evaluate the performance of the unsupervised models, three metrics were used:

- **Adjusted Rand Index (ARI)** – checks how well the clusters match the true labels.
- **Normalized Mutual Information (NMI)** – measures how much information is shared between predicted clusters and actual classes.
- **Silhouette Score** – shows how well the clusters are separated from each other.

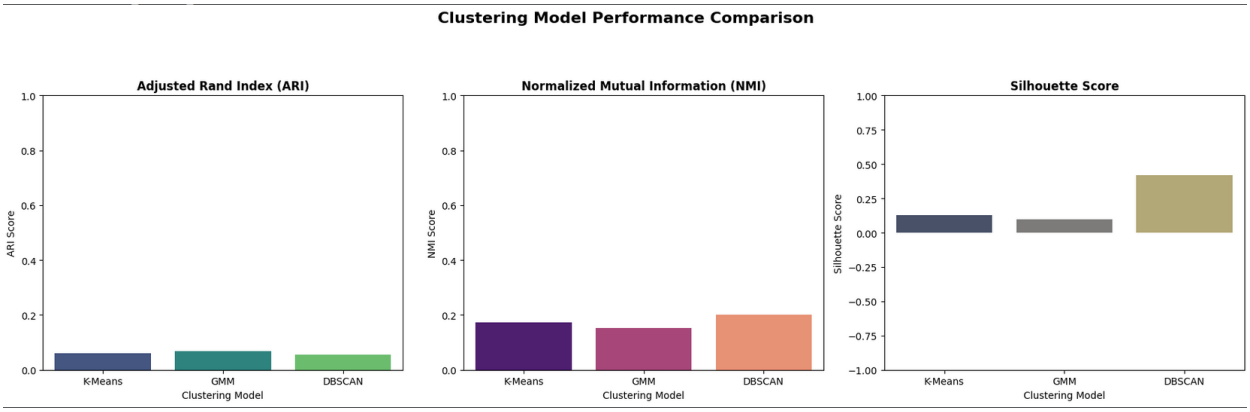


Figure 7: Performance Scores for K-Means, GMM, and DBSCAN

Key Observations:

- ARI and NMI scores are very low for all models, meaning the clusters do not match the real forest cover types.
- DBSCAN has a higher silhouette score, but this only reflects density patterns and not actual class boundaries.
- GMM performs slightly better than K-Means, but it still cannot capture the true structure of the dataset.

Overall, these results show that unsupervised clustering cannot recover the real forest cover types.

So we rely more on supervised models as they are much more suitable for this dataset.

9 Comparative Analysis

The following insights were obtained:

9.1 Supervised Models

- **Neural Network** achieved the best overall performance with an accuracy of **92%** and strong macro F1-score of **0.85**, showing its ability to capture nonlinear feature interactions.
- **SVM** reached an accuracy of **80%** in the full training setting, while the tuned version (Optuna) achieved **94.86%** on a reduced dataset, highlighting strong potential but high computational cost.
- **Logistic Regression** achieved **84%** accuracy, performing well on majority classes but with weaker recall for minority categories.

9.2 Clustering Models

- **K-Means** produced very low ARI values (approximately **0.03–0.05**) and NMI around **0.10**, indicating poor alignment with true classes.
- **GMM** showed slightly better structure, achieving ARI around **0.06** and NMI close to **0.12**, but still far from meaningful class separation.
- **DBSCAN** resulted in extremely low ARI (near **0**) and formed only a few dominant clusters, confirming its inability to capture ecological class differences.

9.3 Overall Observations

The dataset's complexity, high dimensionality, and nonlinear separability make supervised machine learning—especially neural networks—the most suitable approach. Quantitatively, supervised models outperform clustering algorithms by a large margin (accuracy above **90%** vs. ARI below **0.1** for clustering).

10 Conclusion

This project demonstrates how different machine learning models behave on a high-dimensional real-world dataset. Key conclusions include:

- Neural networks provide the strongest predictive performance.
- SVMs are competitive but computationally expensive.
- Logistic regression struggles with nonlinear patterns.
- Unsupervised models do not align well with the class distributions.

Future improvements may include:

- Hyperparameter tuning with grid/random search
- Feature engineering and PCA for dimensionality reduction
- More advanced neural architectures

11 GitHub Repository

GitHub: https://github.com/Lomesh2000/ML-Project-2