

# **Machine Learning Project 2 Report**

**Team Name:** MT2025065

**Member:** Keyur Sanjaykumar Padiya

**Course:** Machine Learning

**Date:** December 12, 2025

**Dataset 1: Smoker Status Prediction**

**Dataset 2: Forest Cover Type Prediction**

# Contents

<b>I</b>	<b>Dataset 1: Smoker Status Prediction</b>	<b>4</b>
<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Data Processing</b>	<b>4</b>
2.1	Exploratory Data Analysis (EDA) . . . . .	4
2.1.1	Initial Analysis . . . . .	4
2.1.2	Target Variable Analysis . . . . .	5
2.1.3	Numerical Feature Analysis . . . . .	6
2.1.4	Outlier Analysis . . . . .	10
2.1.5	Feature Importance and Correlation . . . . .	11
2.1.6	Target Value Analysis . . . . .	12
2.2	Data Preprocessing Pipeline . . . . .	13
2.2.1	Data Loading and Exploration . . . . .	13
2.2.2	Handle Categorical Features . . . . .	13
2.2.3	Handle Numerical Features . . . . .	13
2.2.4	Train-Validation Split . . . . .	14
<b>3</b>	<b>Models Used</b>	<b>15</b>
3.1	Logistic Regression . . . . .	15
3.2	Neural Network (MLP Classifier) . . . . .	15
3.3	Support Vector Machine (SVM) . . . . .	15
<b>4</b>	<b>Hyperparameter Tuning</b>	<b>16</b>
4.1	Logistic Regression Tuning . . . . .	16
4.2	Support Vector Machine (SVM) Tuning . . . . .	16
4.3	Neural Network Tuning . . . . .	17
<b>5</b>	<b>Performance Evaluation</b>	<b>18</b>
<b>6</b>	<b>Conclusion</b>	<b>19</b>
<b>7</b>	<b>GitHub Repository Link</b>	<b>19</b>
<b>II</b>	<b>Dataset 2: Forest Cover Type Prediction</b>	<b>20</b>
<b>1</b>	<b>Introduction</b>	<b>20</b>
<b>2</b>	<b>Data Processing</b>	<b>20</b>
2.1	Exploratory Data Analysis (EDA) . . . . .	20
2.1.1	Initial Analysis . . . . .	20
2.1.2	Target Variable Analysis . . . . .	21
2.1.3	Numerical Feature Analysis . . . . .	22
2.1.4	Outlier Analysis . . . . .	24
2.1.5	Feature Importance and Correlation . . . . .	24
2.1.6	Target Value Analysis . . . . .	26

2.2	Data Preprocessing Pipeline . . . . .	28
2.2.1	Data Loading and Cleaning . . . . .	28
2.2.2	Handle Categorical Features . . . . .	28
2.2.3	Handle Numerical Features . . . . .	28
2.2.4	Train-Validation Split . . . . .	28
<b>3</b>	<b>Models Used</b>	<b>29</b>
3.1	Logistic Regression . . . . .	29
3.2	Support Vector Machine (SVM) . . . . .	29
3.3	Neural Network (MLP Classifier) . . . . .	29
<b>4</b>	<b>Hyperparameter Tuning</b>	<b>30</b>
4.1	Logistic Regression Tuning . . . . .	30
4.2	Support Vector Machine (SVM) Tuning . . . . .	30
4.3	Neural Network Tuning . . . . .	31
<b>5</b>	<b>Performance Evaluation</b>	<b>32</b>
<b>6</b>	<b>Conclusion</b>	<b>32</b>
<b>7</b>	<b>GitHub Repository Link</b>	<b>32</b>

## Part I

# Dataset 1: Smoker Status Prediction

## 1 Introduction

The objective of this project is to analyze a bio-signal dataset to predict the smoker status of individuals. By performing Exploratory Data Analysis (EDA) and preprocessing the data, we aim to build a foundation for machine learning modeling. The dataset includes various physiological and biochemical markers, such as height, weight, eyesight, and serum levels (e.g., cholesterol, hemoglobin), which are critical for understanding the health impact of smoking.

## 2 Data Processing

### 2.1 Exploratory Data Analysis (EDA)

An extensive initial analysis was conducted to understand the data structure, quality, and distributions. The analysis confirmed that the dataset is robust, with a substantial number of samples and no missing values.

#### 2.1.1 Initial Analysis

The dataset is divided into training and testing sets. The training set comprises 38,984 samples with 23 columns, while the test set contains 16,708 samples. The features are predominantly numerical, consisting of 22 numerical attributes and 0 categorical attributes (excluding the target).

Attribute	Count
Total Samples (Train)	38,984
Total Features	22
Missing (Null) Values	0
Duplicate Rows (Removed)	5,517

Table 1: High-Level Data Summary

A check for missing values confirmed that there are zero null values across both training and test datasets, indicating high data quality. However, 5,517 duplicate rows were identified and removed to prevent model bias and ensure unbiased evaluation.

### 2.1.2 Target Variable Analysis

The target variable, `smoking`, is a binary classification target (0 for non-smoker, 1 for smoker). Analysis of the class distribution reveals a class imbalance.

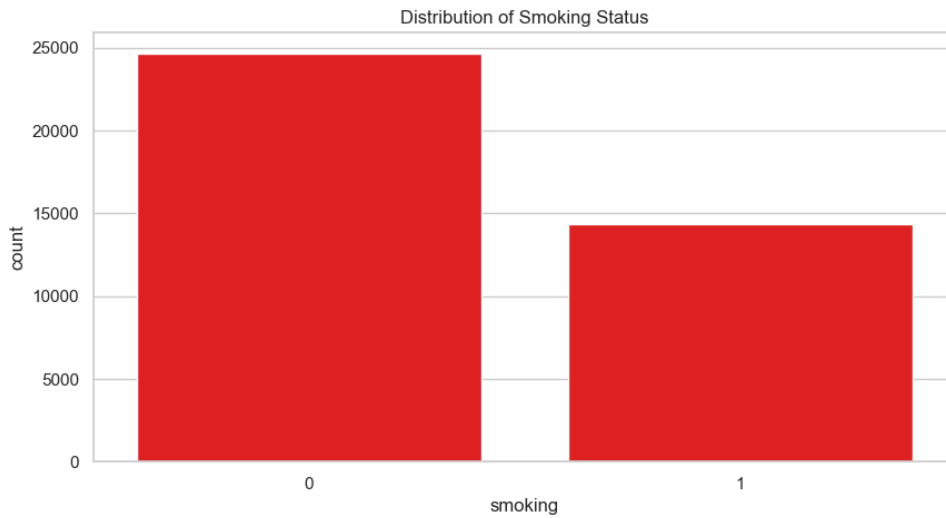


Figure 1: Distribution of the Target Variable (Smoking Status)

As shown in Figure 1, Non-smokers constitute approximately 63.3% of the data, while smokers make up 36.7%. The minority-to-majority class ratio is 0.580. While this imbalance is not extreme, it suggests that stratified splitting techniques should be employed during model training to maintain representative class proportions.

### 2.1.3 Numerical Feature Analysis

We analyzed the distributions of all 22 numerical features. To ensure clarity, the features are visualized across four separate figures, following their order in the dataset.

**Group 1: Demographics and Physical Attributes** Figure 2 displays the distributions for age, height, weight, waist, and eyesight (left/right).

- **Physical Stats:** height, weight, and waist exhibit roughly normal distributions. Waist shows a slight skew to the right, indicating a subset of individuals with higher abdominal measurements.
- **Eyesight:** Both left and right eyesight features are heavily right-skewed. The distribution peaks sharply at standard vision values (around 1.0-1.2) with a long tail extending towards higher values, representing individuals with poorer vision.

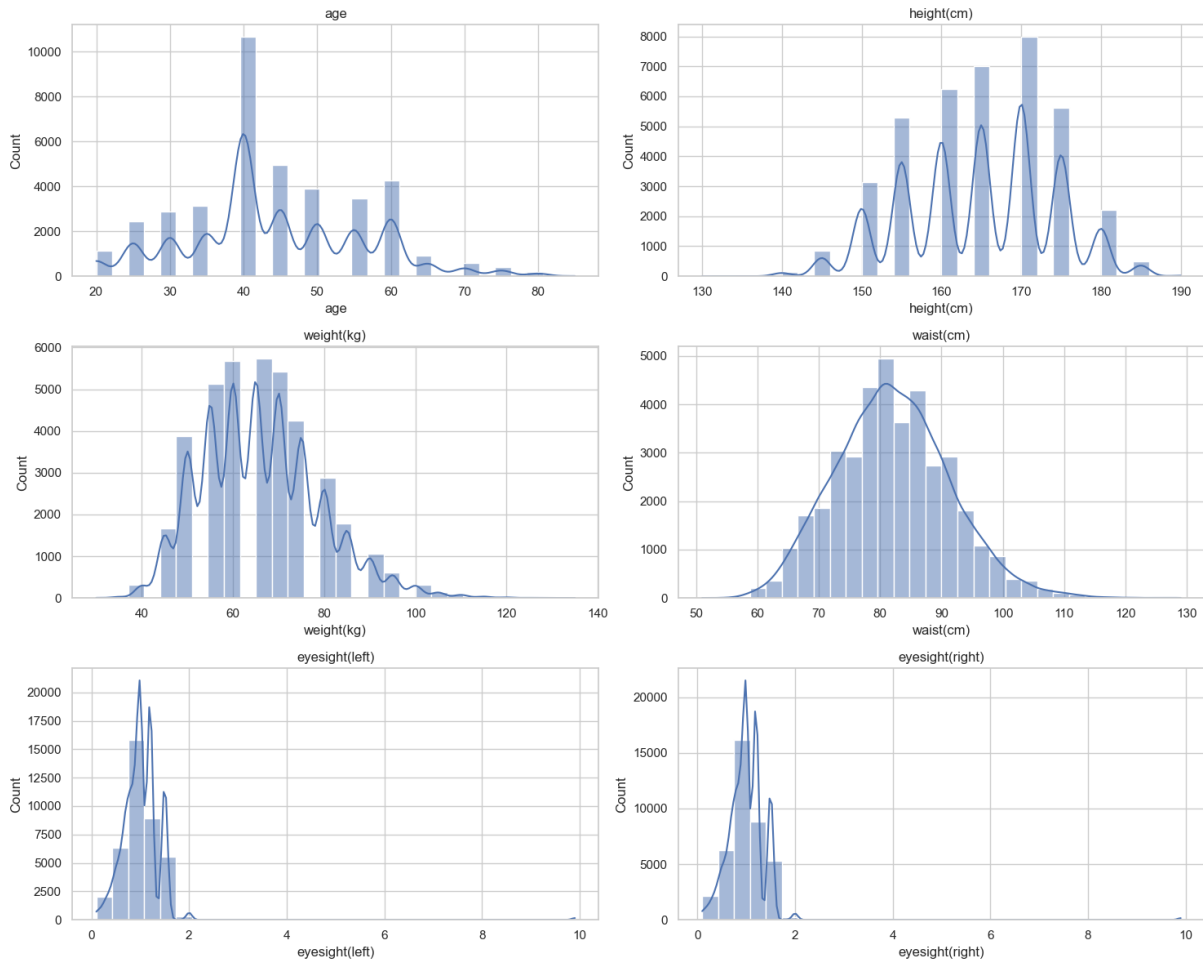


Figure 2: Distribution of Group 1 (Age, Height, Weight, Waist, Eyesight)

**Group 2: Sensory and Blood Pressure** Figure 3 covers **hearing** (left/right), blood pressure (systolic, relaxation), fasting blood sugar, and cholesterol.

- **Hearing:** Both hearing features appear as discrete bars rather than continuous curves, indicating they are likely binary or categorical variables (e.g., 1 for normal, 2 for impaired).
- **Blood Pressure:** **systolic** and **relaxation** follow a classic normal distribution (bell curve).
- **Metabolic:** **fasting blood sugar** is extremely right-skewed, while **Cholesterol** follows a relatively normal distribution centered around the population mean.

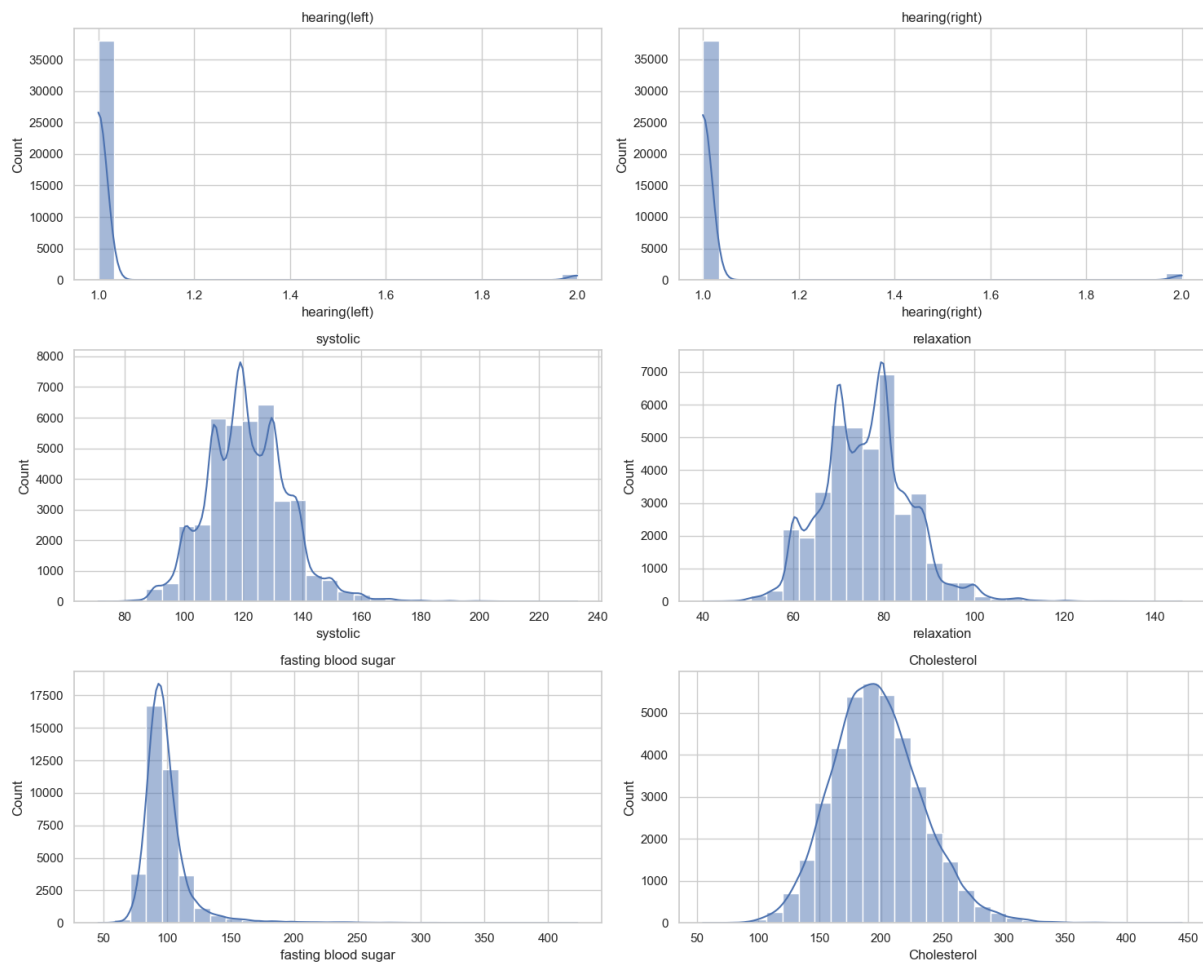


Figure 3: Distribution of Group 2 (Hearing, Blood Pressure, Sugar, Cholesterol)

**Group 3: Lipids and Kidney Function** Figure 4 analyzes triglyceride, HDL, LDL, hemoglobin, urine protein, and serum creatinine.

- **Lipids & Hemoglobin:** HDL, LDL, and hemoglobin display normal distributions. Triglyceride is notably right-skewed, suggesting the presence of high-value outliers.
- **Kidney Markers:** Urine protein is highly skewed, with the vast majority of samples clustered at the lowest value. Serum creatinine also shows a right skew, typical for this biological marker in a general population.

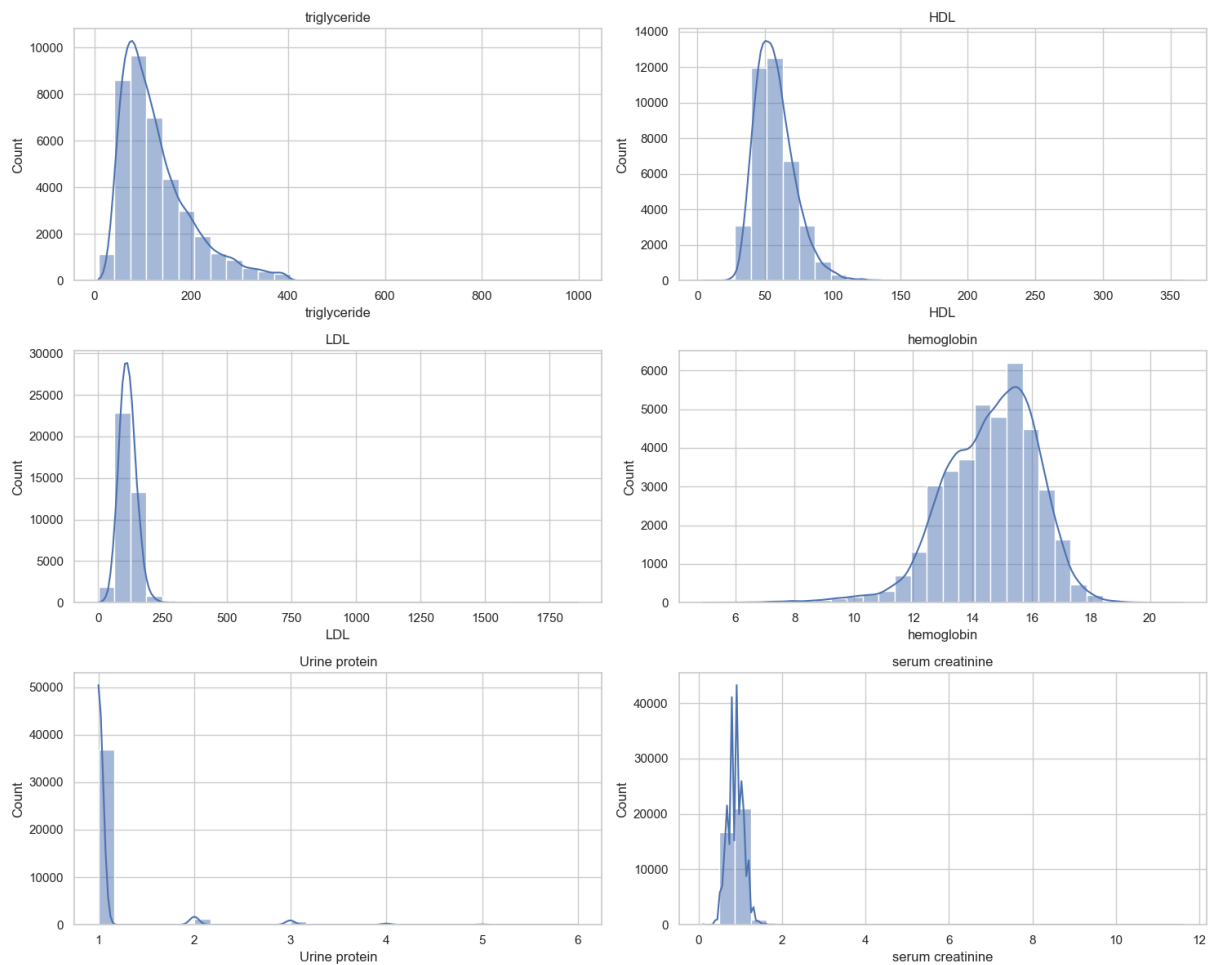


Figure 4: Distribution of Group 3 (Lipids, Hemoglobin, Kidney Markers)

**Group 4: Liver Enzymes and Dental Health** Figure 5 shows the remaining features: AST, ALT, Gtp, and dental caries.

- **Liver Enzymes:** All three liver function markers (AST, ALT, Gtp) are heavily right-skewed. This indicates that while most individuals have low enzyme levels, there is a significant tail of individuals with elevated levels, which is often correlated with smoking or alcohol consumption.
- **Dental Caries:** This feature displays a binary distribution (0 or 1), representing the presence or absence of cavities.

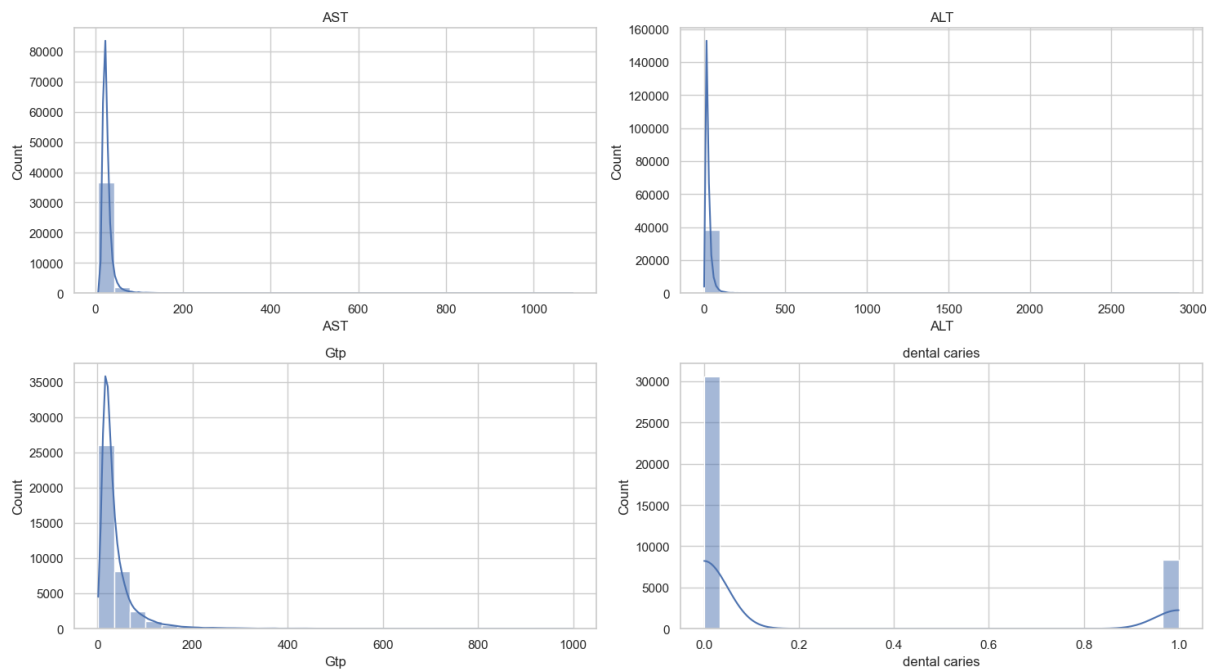


Figure 5: Distribution of Group 4 (Liver Enzymes, Dental Caries)

### 2.1.4 Outlier Analysis

Specific attention was given to the `triglyceride` feature to evaluate outlier detection methods. We compared Z-score and Interquartile Range (IQR) methods.

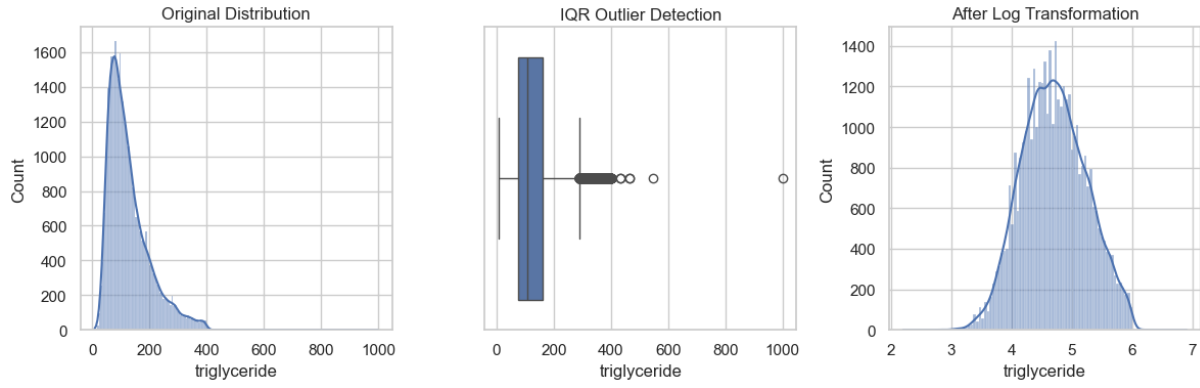


Figure 6: Outlier Analysis on Triglyceride Feature

As illustrated in Figure 6, the Z-score method identified 641 outliers, while the IQR method identified 1,607. Given the biological nature of the data, "extreme" values might be genuine health indicators rather than errors. Therefore, Robust Scaler (which uses median and IQR) was selected for preprocessing to handle these outliers effectively without discarding valuable data.

### 2.1.5 Feature Importance and Correlation

A correlation heatmap was generated to identify linear relationships between features and the target variable.

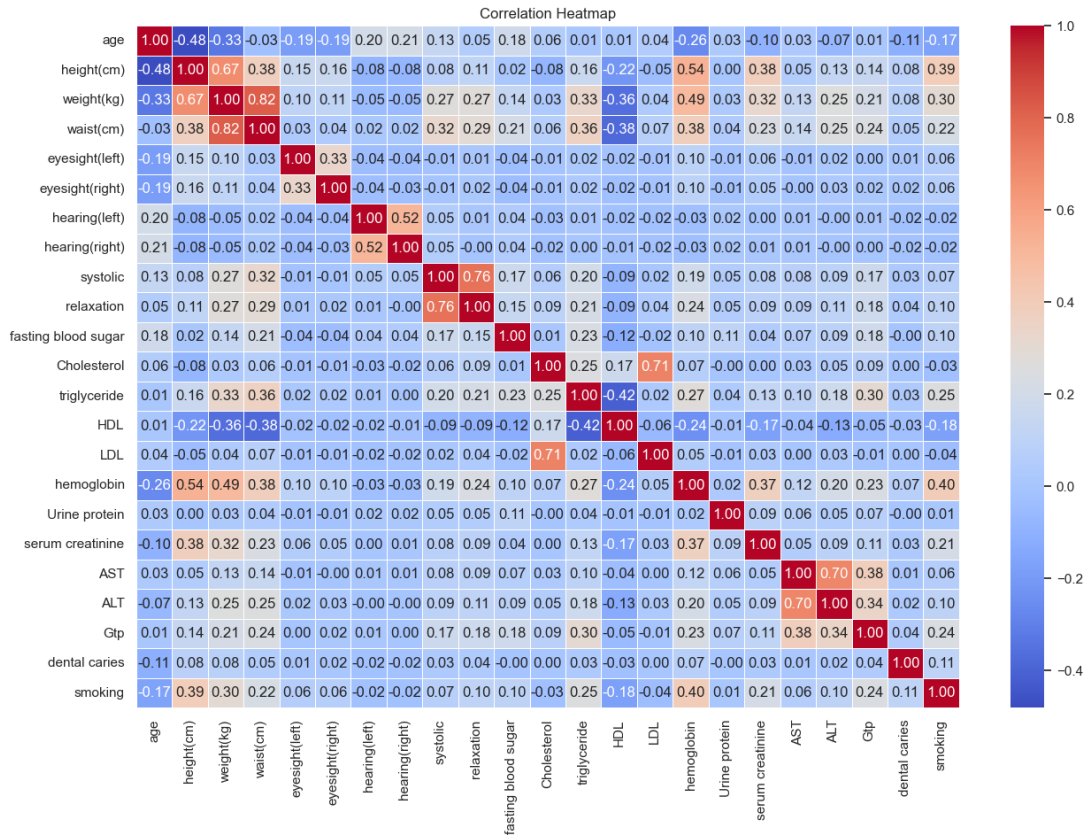


Figure 7: Correlation Heatmap

Figure 7 highlights several key insights:

- **Top Predictors:** hemoglobin (0.40), height (0.39), and weight (0.30) show the strongest positive correlation with smoking status.
- **Multicollinearity:** Strong correlations exist between `weight` and `waist(cm)`, as well as `systolic` and `relaxation` (blood pressure metrics). Tree-based ensembles are generally capable of handling this multicollinearity.
- `Gtp` and `triglyceride` also show notable correlations, reinforcing their potential predictive power.

### 2.1.6 Target Value Analysis

To validate the predictive power of specific bio-signals, boxplots were created to visualize the separation between smokers and non-smokers.

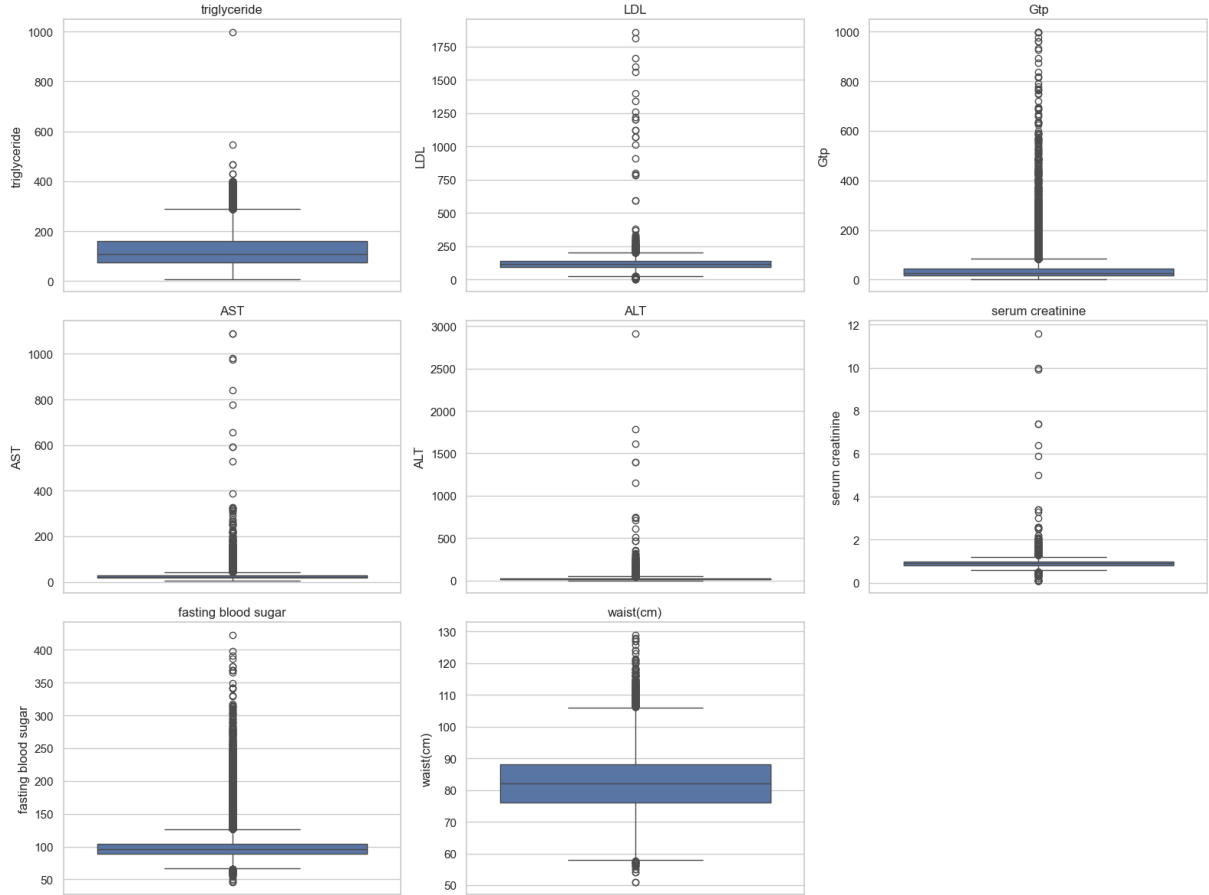


Figure 8: Bio-signal Distributions by Smoking Status

Figure 8 demonstrates clear distinctions in the central tendency for features like hemoglobin and Gtp between the two classes. Smokers tend to have higher median values for these indicators, confirming their relevance as strong predictors for the machine learning model.

## 2.2 Data Preprocessing Pipeline

Data preprocessing is a critical step to ensure the model receives clean, normalized, and structured data. Based on the findings from the EDA, we implemented a pipeline involving data cleaning, feature type handling, robust scaling, and stratified splitting.

### 2.2.1 Data Loading and Exploration

The initial step involved loading the raw training and test datasets. We inspected the data for integrity issues such as missing values and duplicates.

Metric	Value
Original Training Samples	38,984
Test Samples	16,708
Missing Values Detected	0
Duplicate Rows Identified	5,517
<b>Action Taken</b>	<b>Duplicates Removed</b>
Final Training Samples	33,467

Table 2: Data Cleaning Summary

As detailed in Table 2, while the dataset contained no missing values, a significant number of duplicate rows (5,517) were found. These were removed to ensure the model learns generalized patterns rather than memorizing repeated samples.

### 2.2.2 Handle Categorical Features

We explicitly checked for categorical variables (data type 'object') to determine if One-Hot Encoding or Label Encoding was necessary.

- **Identified Categorical Features:** 0 (None)
- **Action:** No encoding required.

The dataset consists entirely of numerical bio-signals, simplifying the pipeline as no complex text processing or cardinality reduction was needed.

### 2.2.3 Handle Numerical Features

Given the presence of significant outliers identified in Section 2.1.4 (specifically in `triglyceride` and `Gtp`), standard mean-variance scaling (`StandardScaler`) is less suitable, as it can be heavily influenced by extreme values. Instead, we employed `RobustScaler`.

This scaling technique utilizes the median and the Interquartile Range (IQR) to scale the data, rather than the mean and standard deviation. By focusing on the central 50% of the data, `RobustScaler` ensures that the outliers found in the bio-signal data do not distort the scaling of the majority of the samples. This preserves the integrity of the feature distributions, allowing the model to learn effective patterns without being biased by extreme physiological measurements.

### 2.2.4 Train-Validation Split

To evaluate the model effectively, the cleaned training data was split into a training set and a validation set. We used Stratified Sampling to preserve the class imbalance ratio (approx. 63:37) in both subsets.

Dataset	Samples	Features	Role
Training Set ( $X_{train}$ )	26,773	22	Model Fitting
Validation Set ( $X_{val}$ )	6,694	22	Hyperparameter Tuning
Test Set (Unseen)	16,708	22	Final Evaluation

Table 3: Final Dataset Split Dimensions

Table 3 summarizes the final shapes of the data fed into the model. This 80/20 split provides a substantial amount of data (over 26k samples) for training while reserving a statistically significant portion (over 6.6k samples) for validation.

## 3 Models Used

To address the binary classification task of predicting smoker status, we selected three distinct machine learning algorithms. Each model was chosen for its unique properties and capability to handle the tabular bio-signal data.

### 3.1 Logistic Regression

Logistic Regression was selected as the baseline model due to its simplicity, interpretability, and efficiency. It models the probability of the target variable as a function of the features using the logistic function. This model serves as a benchmark to determine if complex non-linear models offer significant improvements over a linear decision boundary.

### 3.2 Neural Network (MLP Classifier)

We implemented a Multi-Layer Perceptron (MLP), a type of Artificial Neural Network. Neural networks are capable of capturing complex, non-linear interactions between features through multiple layers of neurons and non-linear activation functions. Given the biological nature of the data, where factors like liver enzymes and blood sugar may interact in complex ways, an MLP is well-suited to learn these latent patterns.

### 3.3 Support Vector Machine (SVM)

The Support Vector Machine was chosen for its effectiveness in high-dimensional spaces and its ability to create complex decision boundaries using kernel functions. By mapping the input data into a higher-dimensional feature space using the Radial Basis Function (RBF) kernel, the SVM can handle non-linear separations that Logistic Regression might miss.

## 4 Hyperparameter Tuning

Extensive hyperparameter tuning was conducted for each model to maximize performance. We utilized both `GridSearchCV` for exhaustive search and `Optuna` for efficient Bayesian optimization.

### 4.1 Logistic Regression Tuning

The initial baseline accuracy was approximately 71.8%. We performed multiple rounds of tuning:

1. **Grid Search:** We exhaustively tested solvers (`liblinear`, `lbfgs`) and penalties (11, 12).
2. **Optuna Optimization:** We expanded the search space for the regularization parameter  $C$  (using a log-uniform distribution) and tolerance.
3. **Data Transformation:** A significant breakthrough occurred when we applied Log Transformation to skewed features (`triglyceride`, `Gtp`, etc.) combined with `RobustScaler`.

Parameter	Optimal Value
Solver	<code>liblinear</code>
Penalty	11
C (Inverse Regularization)	0.0650
Max Iterations	200
Tolerance	0.0022

Table 4: Best Hyperparameters for Logistic Regression

This final configuration significantly improved the accuracy to **73.50%**.

### 4.2 Support Vector Machine (SVM) Tuning

SVM training is computationally expensive, so we employed a multi-stage Optuna search strategy.

1. **Initial Search:** Tuned  $C$  and  $\gamma$  on the full dataset.
2. **Refined Search:** To speed up convergence, we trained on subsets (40% and 60%) of the data to narrow down the parameter space for  $C$  (range 0.5 to 50) and  $\gamma$  (range  $1e-4$  to 0.05).
3. **Final Optimization:** Included additional parameters like `shrinking` and `decision_function_shape`.

Parameter	Optimal Value
Kernel	<code>rbf</code>
C	8.8986
Gamma	0.0144
Tolerance	0.0037
Shrinking	<code>True</code>

Table 5: Best Hyperparameters for SVM

This rigorous tuning process resulted in a validation accuracy of **74.69%**.

### 4.3 Neural Network Tuning

The Neural Network tuning focused on the architecture (number of hidden layers and neurons) and the solver.

- **Architecture Search:** We tested various configurations, including (128), (128, 64), and (128, 64, 32). Deeper networks did not necessarily yield better results, suggesting the dataset complexity could be captured by fewer layers.
- **Solver & Activation:** The `adam` solver with `logistic` (sigmoid) activation function provided the most stable convergence.

Parameter	Optimal Value
Hidden Layer Sizes	(128, 64)
Activation	<code>logistic</code>
Solver	<code>adam</code>
Alpha (L2 Penalty)	0.0001
Learning Rate	<code>constant</code>

Table 6: Best Hyperparameters for Neural Network

The optimal configuration achieved a validation accuracy of **75.71%**, outperforming the baseline.

## 5 Performance Evaluation

After extensive preprocessing and hyperparameter tuning, we compared the best-performing configurations for each of the three models. The Neural Network achieved the highest validation accuracy, demonstrating its ability to effectively model the non-linear relationships in the bio-signal data.

Model	Best Validation Accuracy (%)
Logistic Regression	73.50
Support Vector Machine (SVM)	74.69
<b>Neural Network (MLP)</b>	<b>75.71</b>

Table 7: Performance Comparison Across All Models

While Logistic Regression provided a solid baseline, the non-linear models (SVM and NN) showed a clear advantage. The Neural Network’s superior performance suggests that the interactions between physiological features like GTP, Hemoglobin, and Triglycerides are best captured by a multi-layer perceptron architecture.

## 6 Conclusion

This study successfully demonstrated the efficacy of machine learning in predicting smoker status from bio-signal data, emphasizing the critical role of domain-aware preprocessing. By addressing significant outliers and skewed distributions through *RobustScaler* and log transformations, we established a high-quality dataset for modeling. Among the algorithms evaluated, the **Neural Network** emerged as the superior model, marginally outperforming the Support Vector Machine and the baseline Logistic Regression. These results confirm that while linear baselines are effective, capturing non-linear biological interactions via advanced architectures like neural networks yields the most predictive power for complex medical datasets.

Overall, this project highlights the critical role of domain-aware preprocessing and model selection in analyzing medical datasets.

## 7 GitHub Repository Link

The complete source code, including data preprocessing scripts, model training notebooks, and analysis reports, is available at:

[https://github.com/KeyPad717/smoke\\_status\\_prediction](https://github.com/KeyPad717/smoke_status_prediction)

## Part II

# Dataset 2: Forest Cover Type Prediction

## 1 Introduction

The objective of this project is to analyze the Forest Cover Type dataset to predict the forest cover type based on cartographic variables. Unlike the previous binary classification task, this is a multi-class classification problem involving 7 distinct tree cover types. By performing comprehensive Exploratory Data Analysis (EDA) and robust data preprocessing, we aim to establish a reliable foundation for machine learning modeling. The dataset includes environmental features such as elevation, aspect, slope, and soil type, which are critical for determining vegetation patterns.

## 2 Data Processing

### 2.1 Exploratory Data Analysis (EDA)

An extensive initial analysis was conducted to understand the dataset's high-dimensional structure. The analysis confirmed that the dataset consists of a mix of continuous and binary features representing environmental measurements.

#### 2.1.1 Initial Analysis

The dataset is substantial, containing 581,012 samples and 55 columns. Of these, 54 are predictive features and 1 is the target variable (`Cover_Type`).

Attribute	Count
Total Samples	581,012
Total Features	55
Continuous Numerical Features	10
Binary Categorical Features	44
Missing (Null) Values	0
Duplicate Rows	0

Table 1: High-Level Data Summary (Forest Cover)

A notable characteristic of this dataset is the presence of 44 binary columns representing `Wilderness_Area` (4 columns) and `Soil_Type` (40 columns). An integrity check confirmed zero missing values and zero duplicate rows, indicating high data quality.

### 2.1.2 Target Variable Analysis

The target variable, `Cover_Type`, contains 7 distinct classes. Analysis of the distribution reveals a significant class imbalance.

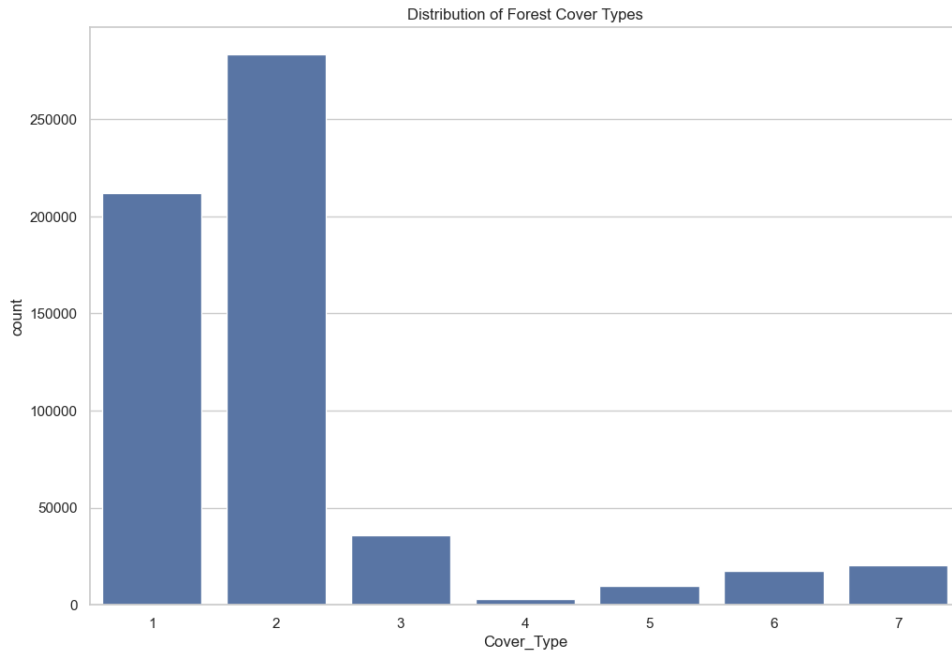


Figure 1: Distribution of Forest Cover Types

As shown in Figure 1, the dataset is dominated by Class 1 (Spruce/Fir) at 36.46% and Class 2 (Lodgepole Pine) at 48.76%. Together, these two classes account for over 85% of the data. In contrast, Class 4 (Cottonwood/Willow) makes up only 0.47% of the data. This severe imbalance suggests that stratified splitting is essential to ensure minority classes are adequately represented during training.

### 2.1.3 Numerical Feature Analysis

We analyzed the distributions of the 10 continuous numerical features to understand their spread and potential outliers. The analysis is visualized across two figures.

**Elevation, Aspect, and Slope:** Figure 2 displays the first set of features.

- **Elevation:** This feature follows a somewhat bimodal distribution but is generally bell-shaped, likely reflecting the altitude preferences of the two dominant tree classes. It has a slight left skew.
- **Aspect & Slope:** These features show multimodal distributions, indicating distinct topographical patterns in the wilderness areas surveyed.
- **Horizontal Distance to Hydrology:** This feature is right-skewed, showing that most trees are located near water sources, with fewer samples at greater distances.
- **Vertical Distance to Hydrology:** This distribution is centered near zero but has a wide spread, capturing elevation changes relative to water bodies.

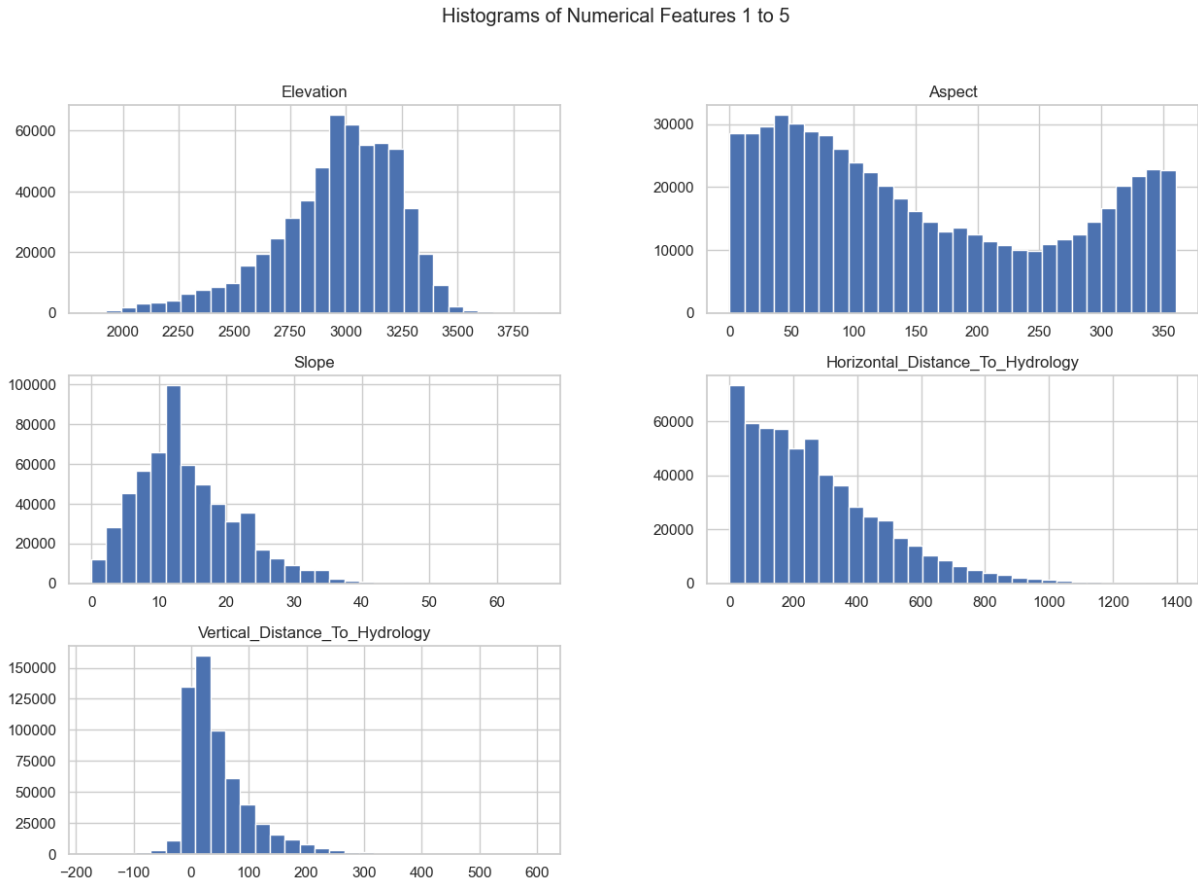


Figure 2: Distribution of Elevation, Aspect, Slope, and Hydrology Distances

**Hillshade and Fire Point Distances:** Figure 3 shows the remaining continuous features.

- **Horizontal Distance to Roadways:** This feature is slightly right-skewed but fairly spread out, indicating variable proximity to access roads.
- **Hillshade (9am, Noon, 3pm):** These features exhibit left-skewness (especially 9am and Noon), meaning most forest areas receive ample sunlight during these times. Hillshade at 3pm shows a more normal distribution.
- **Horizontal Distance to Fire Points:** This feature is right-skewed, with a long tail indicating some forest areas are very remote from fire points.

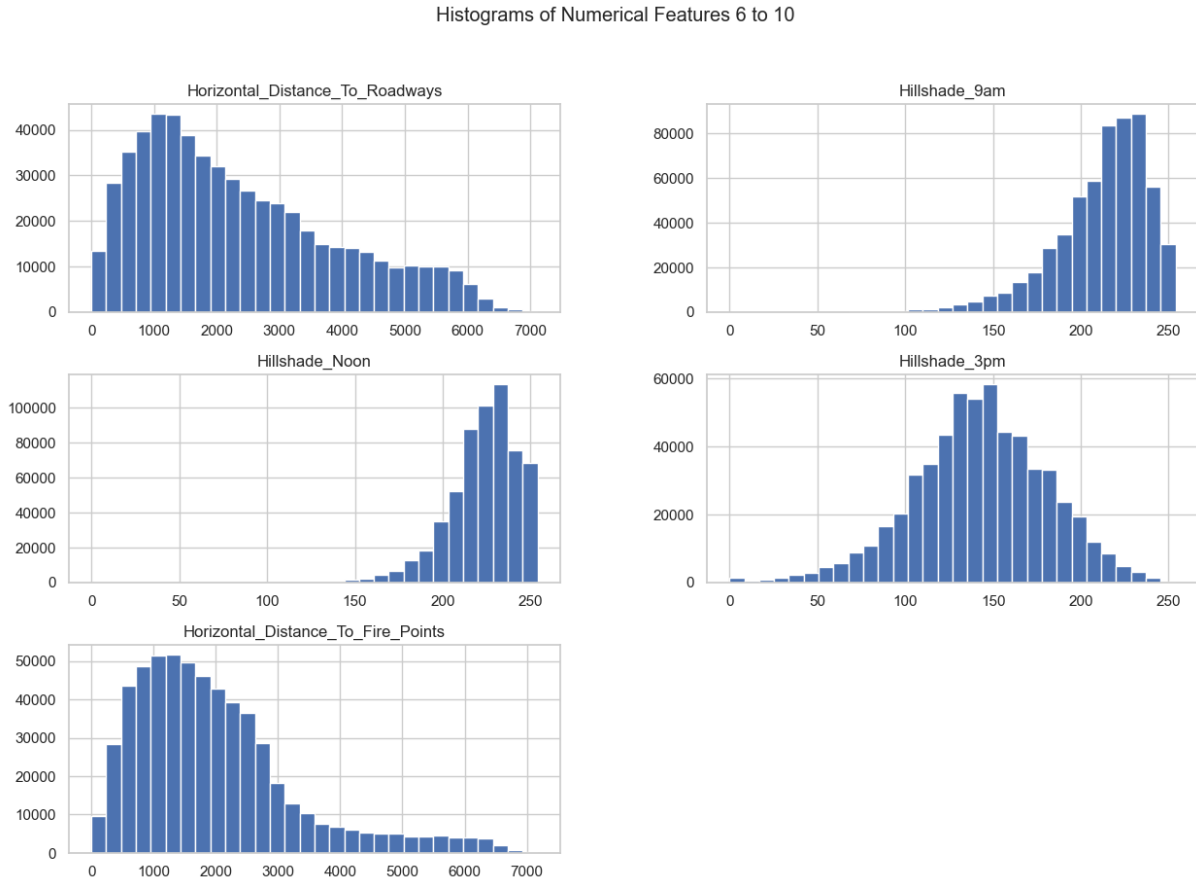


Figure 3: Distribution of Roadway Distance, Hillshade, and Fire Points

### 2.1.4 Outlier Analysis

We compared Z-score and Interquartile Range (IQR) methods to detect outliers in the continuous features.

Feature	Z-score Outliers	IQR Outliers	Skewness
Elevation	5,832	0	-0.82
Vertical_Dist_To_Hydrology	10,059	31,463	1.79
Horz_Dist_To_Roadways	336	69	0.71
Horz_Dist_To_Fire_Points	10,853	31,157	1.29
Hillshade_9am	7,516	17,433	-1.18

Table 2: Outlier Detection Summary (Subset)

The analysis (Table 2) shows that IQR outlier counts are significantly higher than Z-score counts for skewed features like **Vertical\_Distance\_To\_Hydrology** and **Hillshade\_9am**. Since these "extreme" values represent valid environmental variations (e.g., steep cliffs or distant water sources) rather than noise, we selected **RobustScaler** for preprocessing. This scaler uses the median and IQR, making it resilient to these outliers without removing valuable data points.

### 2.1.5 Feature Importance and Correlation

We conducted a two-part correlation analysis to identify relationships between features (multicollinearity) and their predictive power regarding the target variable.

**Feature-to-Feature Correlation:** Figure 4 illustrates the linear relationships between numerical features.

- **Hillshade Multicollinearity:** A strong negative correlation (-0.78) exists between **Hillshade\_9am** and **Hillshade\_3pm**. This is physically intuitive, as slopes facing the morning sun (east) will naturally be in shadow in the afternoon (west).
- **Hydrology:** There is a logical positive correlation between Horizontal and Vertical distances to hydrology, as moving away from water sources often involves elevation changes.
- **Elevation (-0.27):** This is the strongest numerical predictor. The negative correlation suggests that as elevation decreases, the likelihood of certain cover types (like Cottonwood/Willow) increases, or vice-versa for high-altitude species like Krummholz.
- **Roadways (-0.15):** A moderate negative correlation indicates that certain tree types are more prevalent in accessible areas near roads, while others thrive in remote locations.
- **Slope (0.15):** Shows a positive correlation, implying that steeper terrains favor specific vegetation types.

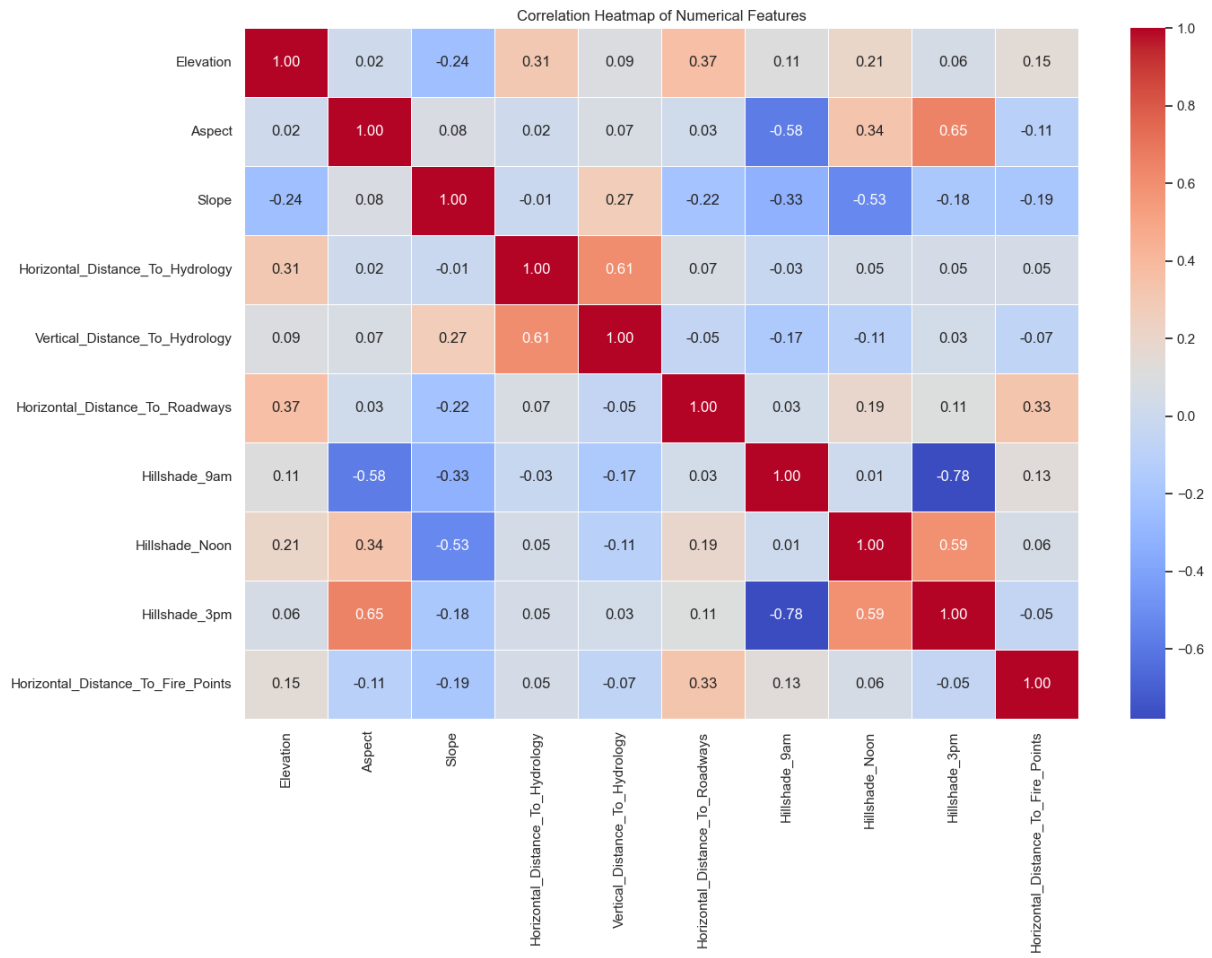


Figure 4: Correlation Heatmap (Numerical Features)

### 2.1.6 Target Value Analysis

To confirm the discriminatory power of the features, we visualized their distributions across the 7 forest cover types using boxplots. This helps identify which features best separate the classes.

**Group 1: Elevation and Hydrology** Figure 5 compares distributions for Elevation, Slope, and Hydrology distances.

- **Elevation Separation:** This provides the clearest separation among classes. Cover Type 7 (Krummholz) consistently appears at the highest elevations (medians > 3000m), while Type 4 (Cottonwood/Willow) is distinctively found at low elevations.
- **Slope:** While the medians are similar, the spread varies. Type 1 and 2 show broader distributions, indicating they can grow on varied terrain, whereas Type 3 is more restricted.

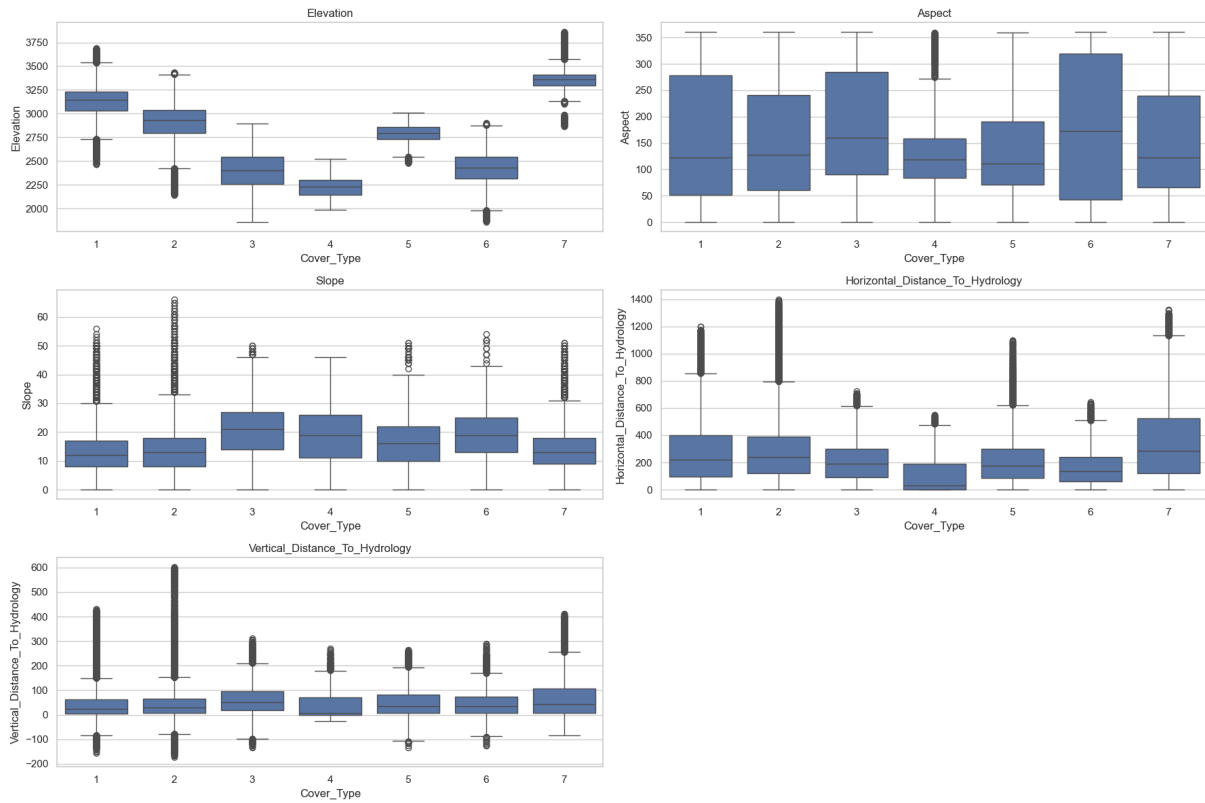


Figure 5: Bio-signal Distributions by Cover Type (Group 1)

**Group 2: Roadways, Hillshade, and Fire Points** Figure 6 examines distances to infrastructure and sunlight metrics.

- **Roadways:** Cover Types 3 and 4 tend to be located significantly closer to roadways, likely due to their lower elevation habitats which are more accessible.
- **Fire Points:** Class 2 (Lodgepole Pine) shows a wide range of distances to fire points, whereas Class 4 is generally clustered closer to them.
- **Hillshade:** The Hillshade at 9am shows distinct variations, helping to differentiate shade-tolerant species from those requiring direct morning sunlight.

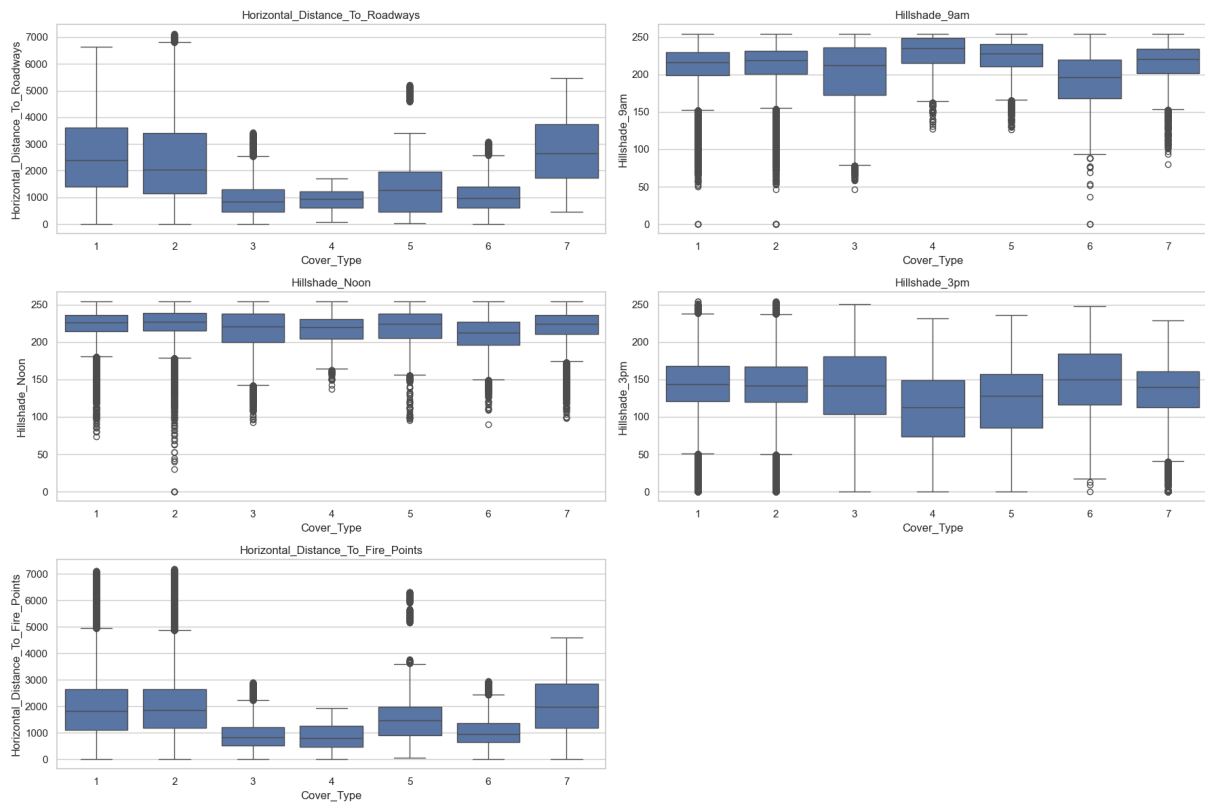


Figure 6: Bio-signal Distributions by Cover Type (Group 2)

## 2.2 Data Preprocessing Pipeline

Based on the EDA findings, we established a pipeline to prepare the data for multi-class classification.

### 2.2.1 Data Loading and Cleaning

The dataset was loaded and checked for integrity. Unlike the previous dataset, the Forest Cover dataset was clean with 0 missing values and 0 duplicate rows. Therefore, no row removal operations were required.

### 2.2.2 Handle Categorical Features

We identified 44 categorical columns:

- **Wilderness\_Area** (4 columns)
- **Soil\_Type** (40 columns)

These features are already pre-encoded as binary (0/1) integers (One-Hot Encoded representation). Consequently, no further encoding steps (like Label Encoding or One-Hot Encoding) were necessary, simplifying the pipeline.

### 2.2.3 Handle Numerical Features

Scaling was applied exclusively to the 10 continuous numerical features (**Elevation**, **Aspect**, **Slope**, etc.). The 44 binary features were left untouched to preserve their categorical nature.

Given the significant outliers identified in Section 2.1.4, we employed **RobustScaler**. This technique removes the median and scales the data according to the quantile range (IQR), ensuring the model is not biased by extreme environmental measurements.

### 2.2.4 Train-Validation Split

To strictly maintain the class proportions of the 7 forest cover types (especially the minority Class 4), we employed a Stratified Split.

Dataset	Samples	Features	Role
Training Set ( $X_{train}$ )	464,809	54	Model Fitting
Validation Set ( $X_{val}$ )	116,203	54	Hyperparameter Tuning

Table 3: Final Dataset Split Dimensions (Forest Cover)

The 80/20 split resulted in a training set of over 464,000 samples, providing ample data for models to learn the complex decision boundaries required for this multi-class problem.

## 3 Models Used

To address the multi-class classification task of predicting forest cover types, we selected three distinct machine learning algorithms. Each model offers different strengths in handling high-dimensional and non-linear data.

### 3.1 Logistic Regression

Logistic Regression was chosen as the baseline model. We utilized the `multinomial` option to handle the 7 target classes naturally. This linear model serves as a benchmark to determine whether the boundaries between forest cover types can be approximated by linear hyperplanes or if more complex non-linear structures are required.

### 3.2 Support Vector Machine (SVM)

The Support Vector Machine was selected for its ability to handle complex decision boundaries. We experimented with both Linear and Radial Basis Function (RBF) kernels. The RBF kernel allows the model to map features into a higher-dimensional space, which is crucial for separating classes like Spruce/Fir and Lodgepole Pine that may have overlapping feature distributions.

### 3.3 Neural Network (MLP Classifier)

We implemented a Multi-Layer Perceptron (MLP) to capture deep non-linear relationships. Given the large dataset size (>500k samples), Neural Networks are well-suited to learn intricate patterns without hitting the performance plateaus often seen with simpler models. We varied the depth and width of the network to optimize performance.

## 4 Hyperparameter Tuning

Extensive hyperparameter tuning was conducted for each model, leveraging `Optuna` for efficient parameter space exploration.

### 4.1 Logistic Regression Tuning

The baseline Logistic Regression achieved an accuracy of approximately 72.48%.

1. **Parameter Selection:** We tested different solvers and penalties. The `lbfgs` solver was selected for its efficiency on large multi-class datasets.
2. **Optuna Optimization:** We performed an extensive search for the regularization parameter  $C$  and tolerance `tol` on subsamples (150k and 300k rows) to speed up the process.
3. **Result:** Despite rigorous tuning, the accuracy plateaued around **72.48%**. The Optuna search confirmed that linear models struggle to capture the full complexity of the data.

Parameter	Optimal Value
Solver	lbfgs
Penalty	l2
C	26.89
Max Iterations	562
Tolerance	2.433e-06

Table 4: Best Hyperparameters for Logistic Regression

### 4.2 Support Vector Machine (SVM) Tuning

Training Support Vector Machines on a dataset of this magnitude (around 580k samples) is computationally expensive, as training time scales quadratically with the number of samples. Consequently, extensive automated hyperparameter tuning (e.g., via Optuna) was infeasible. Instead, we adopted a strategic **manual tuning approach**:

1. **Kernel Selection:** We initially compared Linear vs. RBF kernels. The Linear kernel yielded an accuracy of  $\approx 72.5\%$ , similar to the Logistic Regression baseline, indicating that the data is not linearly separable.
2. **Non-Linearity (RBF):** Switching to the Radial Basis Function (RBF) kernel with standard parameters ( $C = 1.0$ ,  $\gamma = 'scale'$ ) resulted in a significant performance jump to **83.22%**. This confirmed that mapping features to a higher-dimensional space to capture non-linear relationships (e.g., between elevation and soil type) was the critical factor for improvement.

Parameter	Optimal Value
Kernel	<code>rbf</code>
C	1.0 (Standard)
Gamma	<code>scale</code>
Tolerance	0.001

Table 5: Best Hyperparameters for SVM

### 4.3 Neural Network Tuning

The Neural Network showed the most promise, responding well to increased model depth and data size. We utilized an iterative **manual tuning strategy** rather than a "black-box" search. This approach allowed us to isolate the impact of specific architectural changes (like adding depth or changing activation functions) to ensure the model was learning features rather than just memorizing noise.

- **Architecture Search:** We tested architectures ranging from shallow (1 layer) to deep (4 layers). A structure of (256, 128, 64) neurons provided a significant boost.
- **Activation Function:** The `tanh` activation function consistently outperformed `relu` and `logistic` functions for this dataset, improving accuracy to over 89%.
- **Data Scaling:** Increasing the training subset from 30% to 100% resulted in a steady accuracy climb from 92% to 95%.

Parameter	Optimal Value
Hidden Layer Sizes	(256, 128, 64)
Activation	<code>tanh</code>
Solver	<code>adam</code>
Learning Rate	<code>adaptive</code>
Alpha	0.0005
Max Iterations	1500

Table 6: Best Hyperparameters for Neural Network

The final Neural Network configuration achieved a remarkable validation accuracy of **95.42%**.

## 5 Performance Evaluation

After training on the full dataset with optimized parameters, we compared the validation accuracy of the three models. The Neural Network was the clear winner, demonstrating superior capability in handling the complex, high-dimensional interactions of the forest cover data.

Model	Best Validation Accuracy (%)
Logistic Regression	72.48
Support Vector Machine (SVM)	83.22
<b>Neural Network (MLP)</b>	<b>95.42</b>

Table 7: Performance Comparison Across All Models (Forest Cover)

The results highlight a clear hierarchy: linear models (LR) are insufficient; kernel methods (SVM) provide a significant improvement by handling non-linearity; but deep learning (NN) fully unlocks the predictive potential of the dataset, correctly classifying the vast majority of samples including the minority classes.

## 6 Conclusion

This project demonstrated the critical importance of model selection when dealing with complex, large-scale environmental datasets. Through rigorous EDA, we identified key features like Elevation and addressed outliers using Robust Scaling. While Logistic Regression provided a baseline, it was limited by the non-linear nature of the data. The Support Vector Machine improved performance significantly (83.22%) using an RBF kernel. However, the **Neural Network** achieved exceptional performance (95.42%) by leveraging a deep architecture with `tanh` activation to capture intricate feature interactions. This underscores the value of Deep Learning for large, multi-class tabular datasets where feature relationships are highly non-linear.

## 7 GitHub Repository Link

The complete source code, including data preprocessing scripts, model training notebooks, and analysis reports, is available at:

[https://github.com/KeyPad717/Forest\\_Cover\\_Type\\_Prediction](https://github.com/KeyPad717/Forest_Cover_Type_Prediction)