AIT511: Course Project 2

Team: MT2025013
Member: Aditya Dave

Course: Machine Learning

December 10, 2025

Contents

1 Abstract

This project addresses two supervised machine learning classification problems based on Kaggle competitions: Smoker Status Prediction and Forest Cover Type Classification. Both datasets are analyzed using three classification models taught after the mid-semester course evaluation, namely Logistic Regression, Support Vector Machine (SVM), and Artificial Neural Networks (ANN). The study begins with an in-depth exploratory data analysis (EDA), followed by data visualization and preprocessing steps including handling missing values, feature scaling, and feature transformation. Hyperparameter tuning is performed for each model using appropriate optimization techniques to improve predictive performance. Each model is then trained on the processed data and evaluated using accuracy as the primary performance metric. A comparative analysis of the models is presented to identify the most effective algorithm for each dataset. The report also includes a detailed discussion of the observed results and the underlying reasons for differences in model performance. The analysis is first conducted on the Smoker Status Prediction dataset and subsequently extended to the Forest Cover Type Classification dataset.

Smoker Status Prediction using Bio-Signals

2 Introduction

We always hear and see advertisements that smoking is injurious to health, but people always ignore this statement. In medical science, generally people lie about their smoking habits, and to claim their insurance, they try to hide their smoking habits; for that, we need to create an algorithm that can detect whether a person is a smoker or not using the person's lifestyle and attributes. In this study, we will analyze the data and each feature, and we will figure out which features are important to indicate the smoking status properly and confidently.

3 Dataset Description

In the smoker status prediction dataset, we have 38,984 data points, 22 features, and 1 target column.

Feature	**Description**	**Type**
age	Age in 5-years gap	Numerical
height(cm)	Height in centimeters	Numerical
weight(kg)	Weight in kilograms	Numerical
waist(cm)	Waist circumference length	Numerical
eyesight(left)	Left eye vision	Numerical
eyesight(right)	Right eye vision	Numerical
hearing(left)	Left ear hearing	Categorical
hearing(right)	Right ear hearing	Categorical
systolic	Systolic blood pressure	Numerical
relaxation	Diastolic blood pressure	Numerical
fasting blood sugar	Fasting blood sugar level	Numerical
Cholesterol	Total cholesterol	Numerical
triglyceride	Triglyceride level	Numerical
HDL	High-density lipoprotein cholesterol	Numerical
LDL	Low-density lipoprotein cholesterol	Numerical
hemoglobin	Hemoglobin level	Numerical
Urine protein	Urine protein level	Categorical
serum creatinine	Serum creatinine level	Numerical
AST	Glutamic oxaloacetic transaminase type	Numerical
ALT	Glutamic oxaloacetic transaminase type	Numerical
Gtp	γ-GTP level	Numerical
dental caries	Dental cavity condition	Categorical
smoking	Smoking status (Target)	Categorical

Table 1: Feature description

4 Exploratory Data Analysis (EDA)

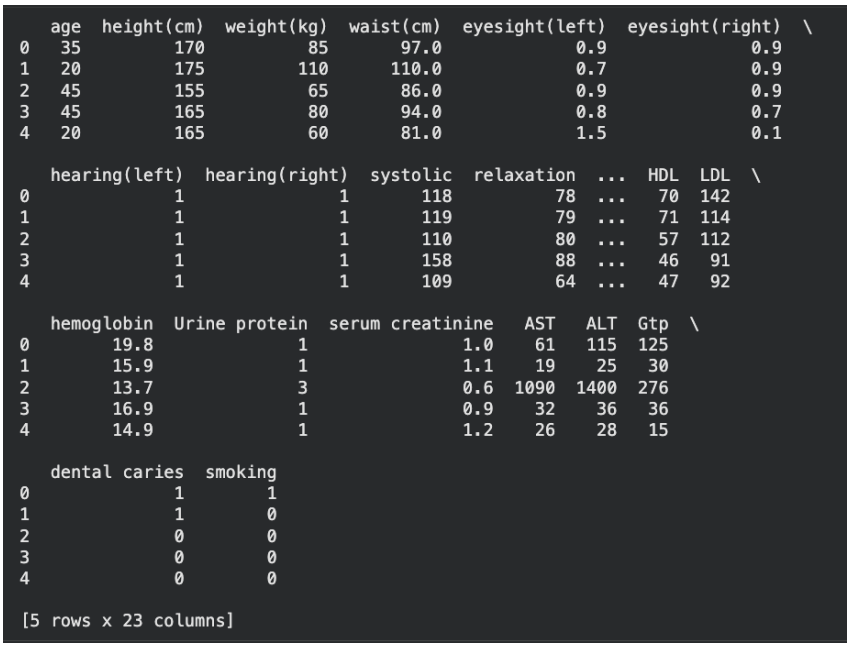
In the Exploratory Data Analysis (EDA), we will deep dive into each feature, what are the values of each feature, how well it is distributed, and its relation with the target feature.

4.1 Dataset Overview

In this section, we will analyze shape and structure of the data.

4.1.1 Dataset Dimensions

The training dataset contains data points with 22 features, excluding the target feature. The training dataset contains 38,984 data points.



	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	eyesight(right)	\
0	35	170	85	97.0	0.9	0.9	
1	20	175	110	110.0	0.7	0.9	
2	45	155	65	86.0	0.9	0.9	
3	45	165	80	94.0	0.8	0.7	
4	20	165	60	81.0	1.5	0.1	

	hearing(left)	hearing(right)	systolic	relaxation	...	HDL	LDL	\
0	1	1	118	78	...	70	142	
1	1	1	119	79	...	71	114	
2	1	1	110	80	...	57	112	
3	1	1	158	88	...	46	91	
4	1	1	109	64	...	47	92	

	hemoglobin	Urine protein	serum creatinine	AST	ALT	Gtp	\
0	19.8	1	1.0	61	115	125	
1	15.9	1	1.1	19	25	30	
2	13.7	3	0.6	1090	1400	276	
3	16.9	1	0.9	32	36	36	
4	14.9	1	1.2	26	28	15	

	dental caries	smoking
0	1	1
1	1	0
2	0	0
3	0	0
4	0	0

[5 rows x 23 columns]

Figure 1: Dataset first 5 rows

4.1.2 Feature Types

In the dataset, it is important to divide features into numerical and categorical. Here, in the categorical, we do not have strings like "Yes" or "No"; we only have numbers, so here we need to first calculate how many unique values each feature contains, and based on that, we need to make a threshold beyond which every feature will be treated as a numerical feature and at or below the threshold will be treated as a categorical feature.

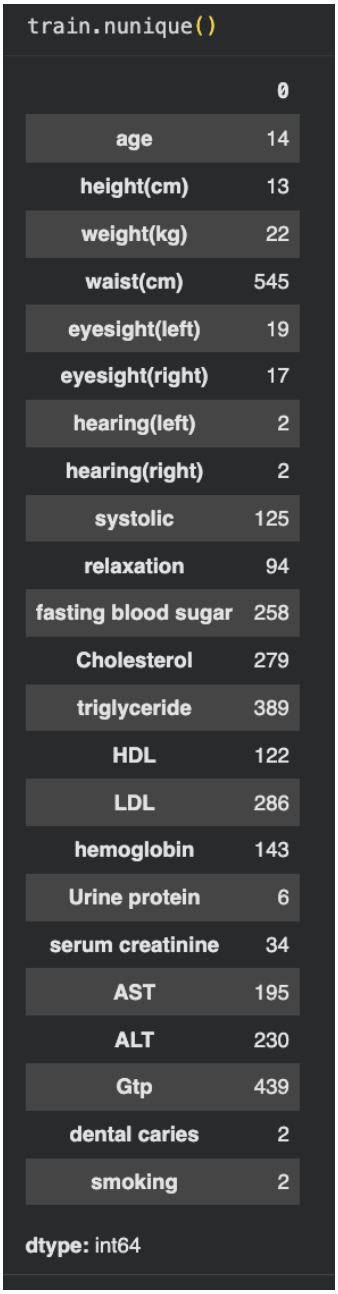


Figure 2: Unique values in each feature

Here we will keep the threshold as 6, so all the features equal to or less than 6 will be considered as categorical variables and others as numerical features.

Feature Type	**Feature Names**
Numerical	age, height(cm), weight(kg), waist(cm), eyesight(left), eyesight(right), systolic, relaxation, fasting blood sugar, Cholesterol, triglyceride, HDL, LDL, hemoglobin, serum creatinine, AST, ALT, Gtp
Categorical	hearing(left), hearing(right), Urine protein, dental caries

Table 2: Numerical and Categorical Features

4.1.3 Range of the feature values

We will divide data into two parts, one for the numerical values and one for the categorical values. Purpose behind doing this is we can see if there is any outlier in the numerical value or not. Same for the categorical value.

	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	\
min	20.000000	130.000000	30.000000	51.000000	0.100000	
max	85.000000	190.000000	135.000000	129.000000	9.900000	
mean	44.153943	164.684465	65.930319	82.081501	1.013849	
var	145.727571	84.563974	165.841735	86.686033	0.246259	

	eyesight(right)	systolic	relaxation	fasting blood sugar	\
min	0.100000	71.000000	40.000000	46.000000	
max	9.900000	233.000000	146.000000	423.000000	
mean	1.009553	121.498730	76.017599	99.261511	
var	0.247872	186.896752	93.548934	419.609253	

	Cholesterol	triglyceride	HDL	LDL	hemoglobin	\
min	55.000000	8.000000	4.000000	1.000000	4.900000	
max	445.000000	999.000000	359.000000	1860.000000	21.100000	
mean	196.964562	126.806048	57.257537	115.182090	14.624463	
var	1326.181497	5150.288354	213.102232	1862.713032	2.441136	

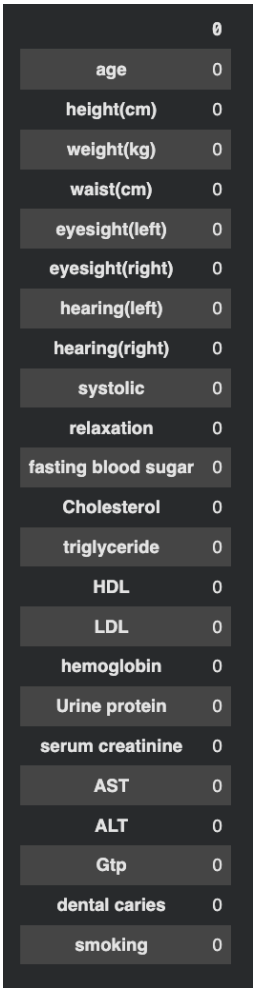
	serum creatinine	AST	ALT	Gtp
min	0.100000	6.000000	1.000000	2.000000
max	11.600000	1090.000000	2914.000000	999.000000
mean	0.886467	26.195536	27.139929	39.952401
var	0.049301	351.959374	999.391792	2496.574768

Figure 3: Range for the numerical values

The numerical features show a wide range of values, which helps in identifying outliers in the dataset. Age ranges from 20 to 85 years, showing that data includes both young and old individuals. Weight, waist, and blood pressure values have large ranges, indicating the presence of both normal and unhealthy individuals. Fasting blood sugar and cholesterol-related features such as triglyceride, HDL, and LDL also show very high maximum values, which suggest the existence of extreme medical conditions. Liver-related features like AST, ALT, and GTP have very high maximum values, clearly indicating strong outliers related to smoking effects. Overall, these variations show that normalization and outlier handling are important before applying machine learning models.

4.1.4 Check for duplicates and missing values

In the dataset, we don't have null values, but we have 5517 duplicate rows. Before visualizing the dataset, we need to remove this data. So after importing the data, first we will remove duplicate data points.



	0
age	0
height(cm)	0
weight(kg)	0
waist(cm)	0
eyesight(left)	0
eyesight(right)	0
hearing(left)	0
hearing(right)	0
systolic	0
relaxation	0
fasting blood sugar	0
Cholesterol	0
triglyceride	0
HDL	0
LDL	0
hemoglobin	0
Urine protein	0
serum creatinine	0
AST	0
ALT	0
Gtp	0
dental caries	0
smoking	0

Figure 4: Missing values in each attributes

Figure 5: Duplicate rows in the dataset

4.2 Target Variable Analysis

The target feature smoking is a binary variable with two classes: 0 (non-smoker) and 1 (smoker). From the distribution, it is clear that the number of non-smokers is higher than the number of smokers, but both classes are well represented. This indicates a moderate class imbalance, not a severe one. Since both categories have a sufficient number of samples, the distribution is acceptable for training classification models, especially when using stratified train–test splitting. However, the imbalance should still be considered during model evaluation.

Category	**Count**
Non-smokers (0)	21209
Smokers (1)	12258

Table 3: Distribution of Smoking Status

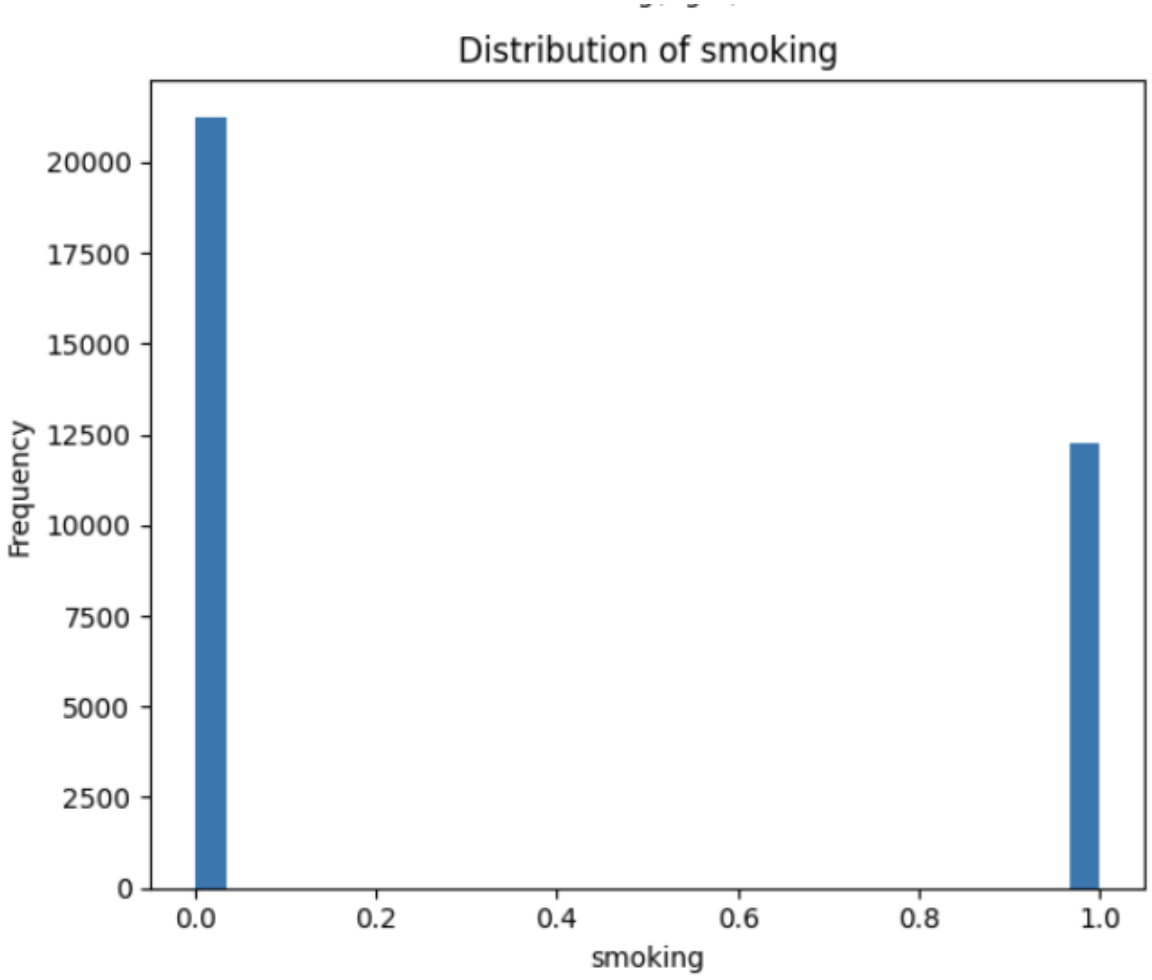


Figure 6: Target values distribution

4.3 Univariate Analysis

4.3.1 Numerical Features

In the univariate analysis, we will look at each feature very closely, so we will start with numerical features.

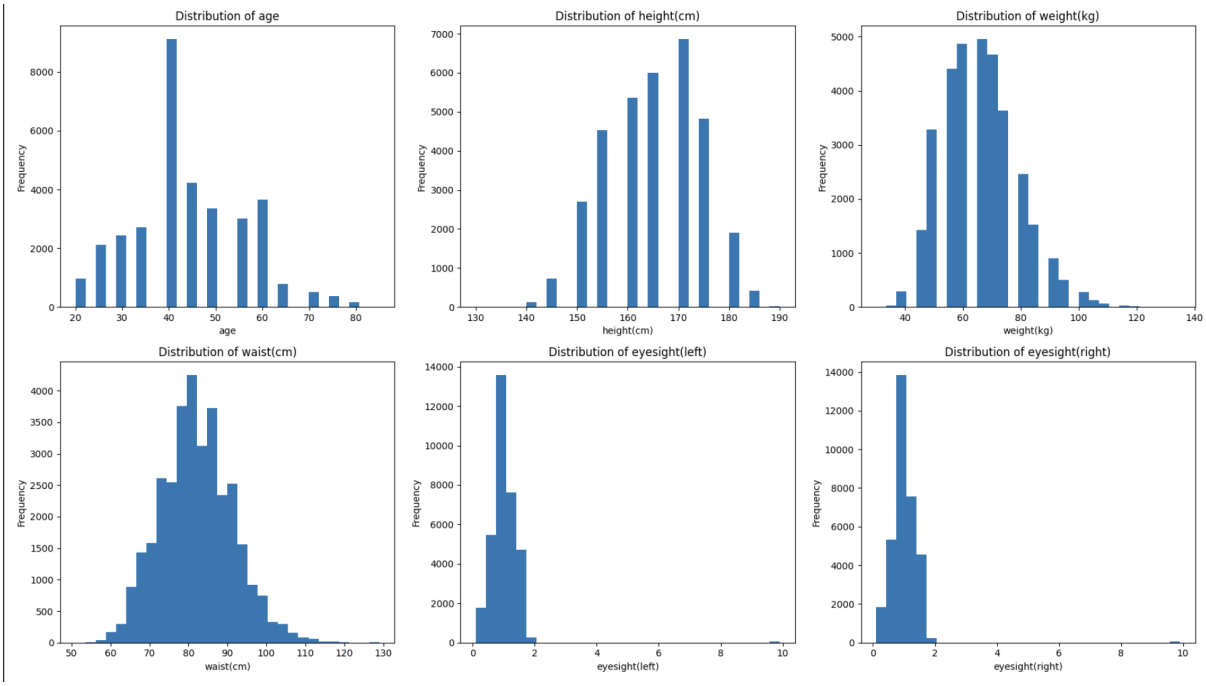


Figure 7: Numerical Features Distribution

The age distribution is slightly right-skewed, with most individuals falling between 40 and 60 years. Height follows an almost normal distribution centered around the average human height range. Weight shows a right-skewed pattern with most values between 50 and 80 kg and a few higher outliers. Waist circumference is mainly concentrated between 70 and 90 cm and appears normally distributed, with some larger values indicating obesity. For eyesight, both left and right eyes have most values clustered near 1.0, while a few extreme outliers are observed at higher values.

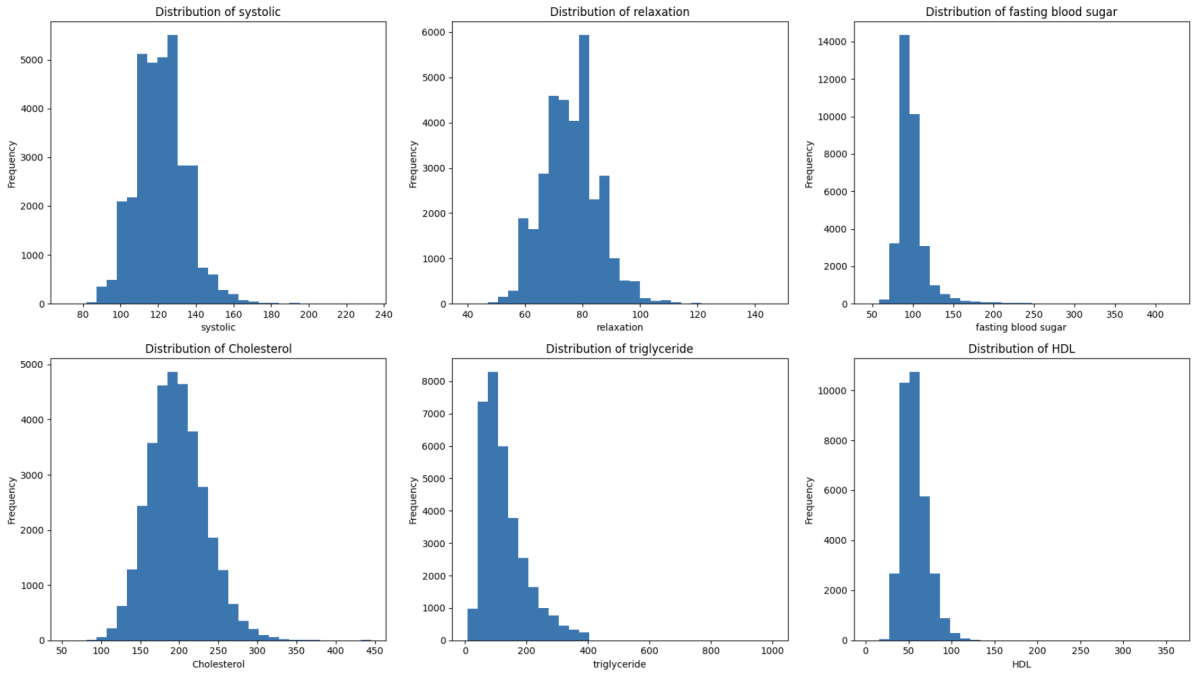


Figure 8: Numerical Features Distribution

The systolic and relaxation (diastolic) blood pressure values show near-normal distributions, with most individuals falling in the healthy to slightly elevated range, while a few high-value outliers indicate hypertension. Fasting blood sugar is highly right-skewed, where most values are near the normal range but some extreme values represent diabetic conditions. Cholesterol levels appear approximately normally distributed around the average range, with a few high outliers indicating hypercholesterolemia. Triglyceride values are strongly right-skewed with several extreme outliers, suggesting abnormal lipid metabolism in some individuals. HDL shows a moderate spread with most values concentrated in the normal range and fewer high-value outliers.

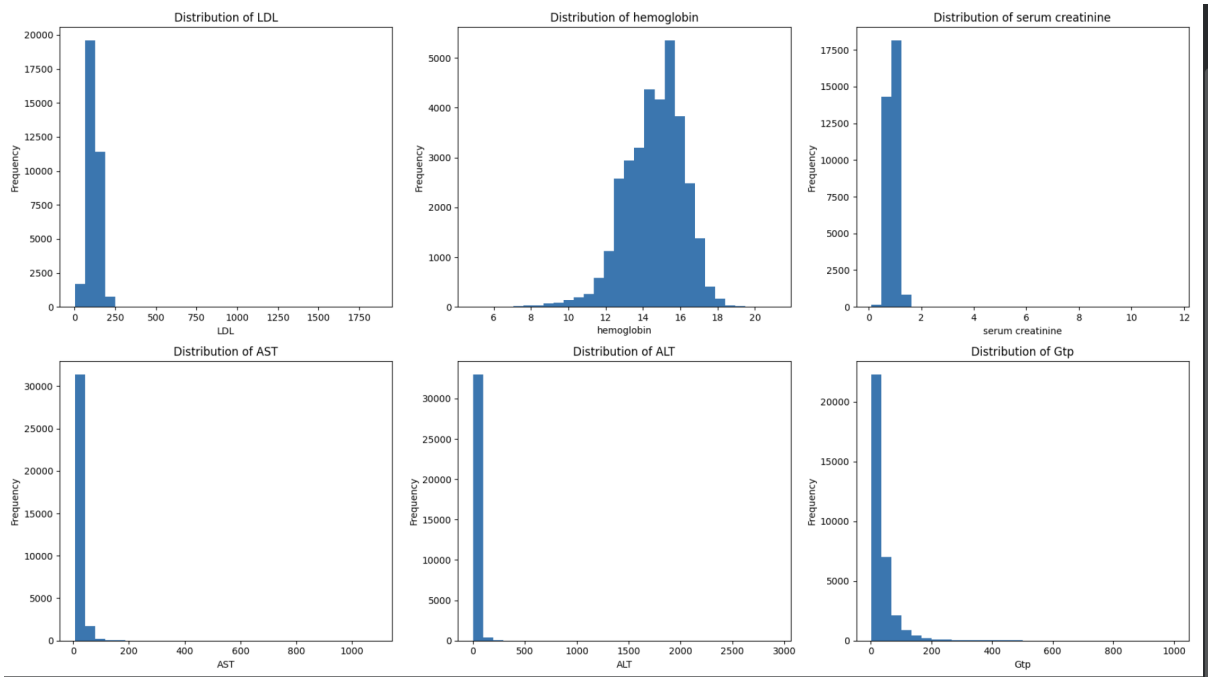


Figure 9: Numerical Features Distribution

The LDL distribution is highly right-skewed, with most values in the lower range and a few extreme outliers at very high levels. Hemoglobin follows an almost normal distribution centered around the normal healthy range, indicating a balanced spread of values. Serum creatinine values are tightly clustered near the lower range, with very few higher outliers representing possible kidney-related issues. Both AST and ALT enzyme levels show strong right-skewness with extremely large outliers, clearly indicating abnormal liver function in some individuals. Similarly, GTP is also heavily right-skewed with a few very high values, which strongly reflects the biological impact of smoking on liver health.

Overall, most numerical features in the dataset show either normal or right-skewed distributions. Features such as height, waist, hemoglobin, and blood pressure appear approximately normally distributed, which is suitable for machine learning models. However, many medical features like fasting blood sugar, triglyceride, LDL, AST, ALT, and GTP are highly right-skewed and contain extreme outliers. These outliers represent serious medical conditions but can negatively affect model performance if not handled properly. To overcome this issue, normalization and log transformation can be applied to reduce skewness, and outlier handling techniques such as IQR-based capping or removal can be used. After proper preprocessing, the data becomes more stable and suitable for accurate model training.

4.3.2 Categorical Features

In the categorical feature, we only have 4 features and 1 target; first we will analyze the distribution of features across categories.

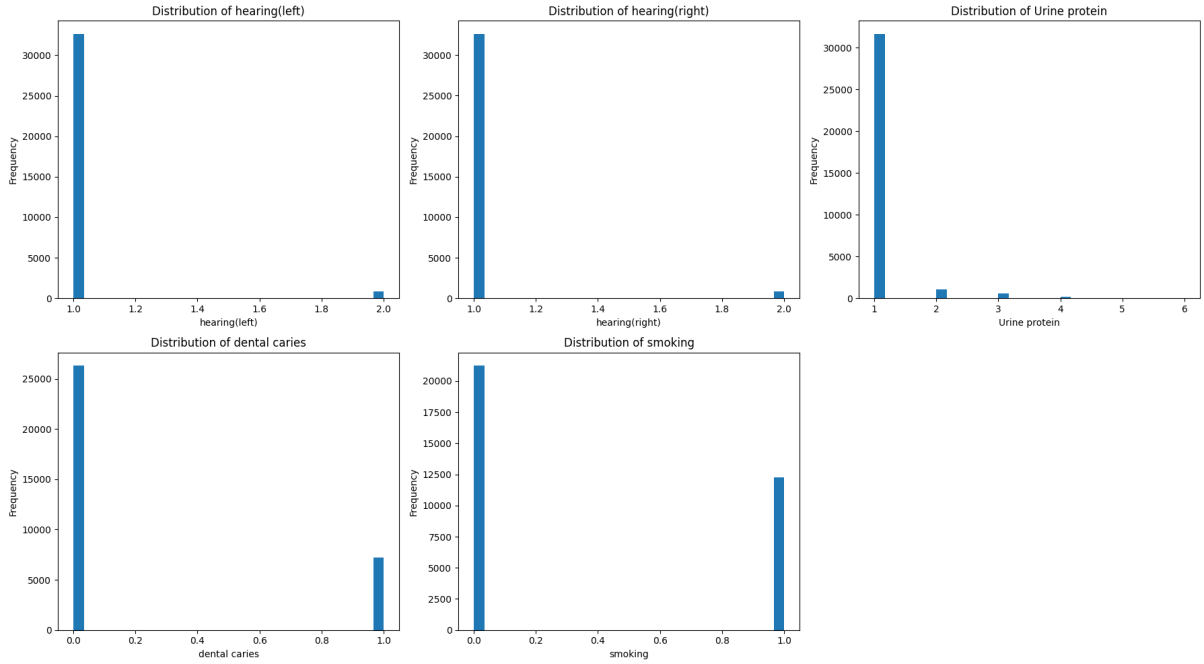


Figure 10: Categorical features Distribution

The distributions of hearing (left and right) show that most individuals fall into the normal hearing category, with only a very small number having impaired hearing. This indicates a strong class imbalance in both hearing features. The urine protein feature is also highly imbalanced, with most individuals having normal levels and very few showing higher levels, which may indicate kidney-related issues in rare cases. Similarly, the dental caries feature shows that most individuals do not have dental problems, while a smaller portion shows the presence of cavities. Overall, these categorical features are highly imbalanced but still medically meaningful, as the minority classes represent important health conditions that can contribute to smoking-related risk prediction.

Since the categorical features such as hearing (left and right), urine protein, and dental caries are highly imbalanced, they should be handled carefully during preprocessing.Additionally, proper scaling is not required for categorical variables, but their impact on the model can be controlled using regularization and by selecting appropriate evaluation metrics.

4.4 Correlation and Multicollinearity

It is essential to know about the correlation between features and the target column. I am showcasing 3 heat maps. One is between numerical features, one is between categorical features, and one is between features and the target.

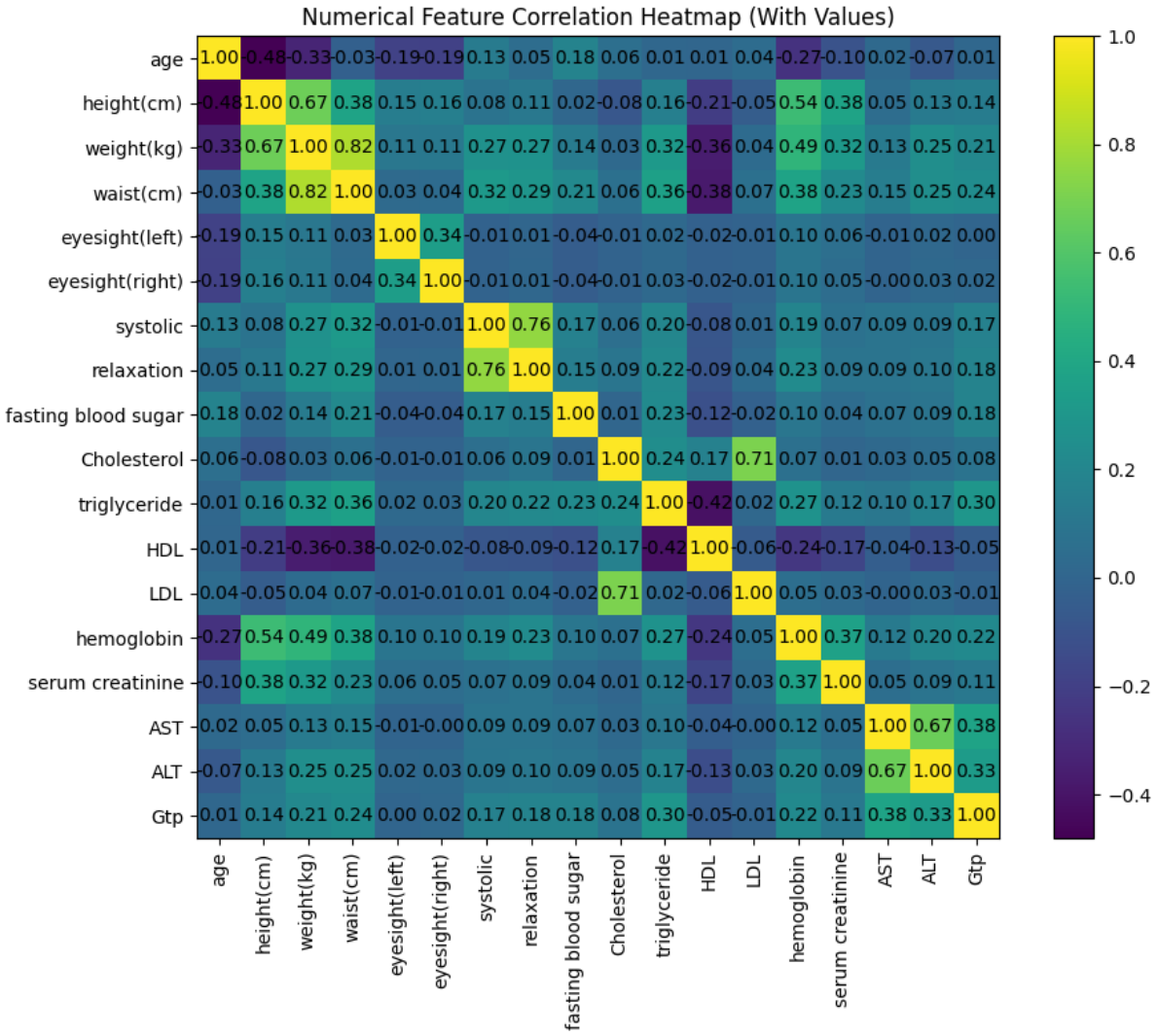


Figure 11: Correlation Matrix for Numerical Data

The correlation matrix shows that weight and waist are highly positively correlated, which is expected as both are indicators of body size. Systolic and relaxation blood pressure also have a strong positive correlation, showing that these two measurements increase together. Cholesterol and LDL exhibit a high positive correlation, which confirms their direct medical relationship. AST and ALT show a strong positive correlation, indicating that both are liver-related enzymes and behave similarly. Most other feature pairs show low correlation, suggesting that many numerical features contribute independent information to the model.

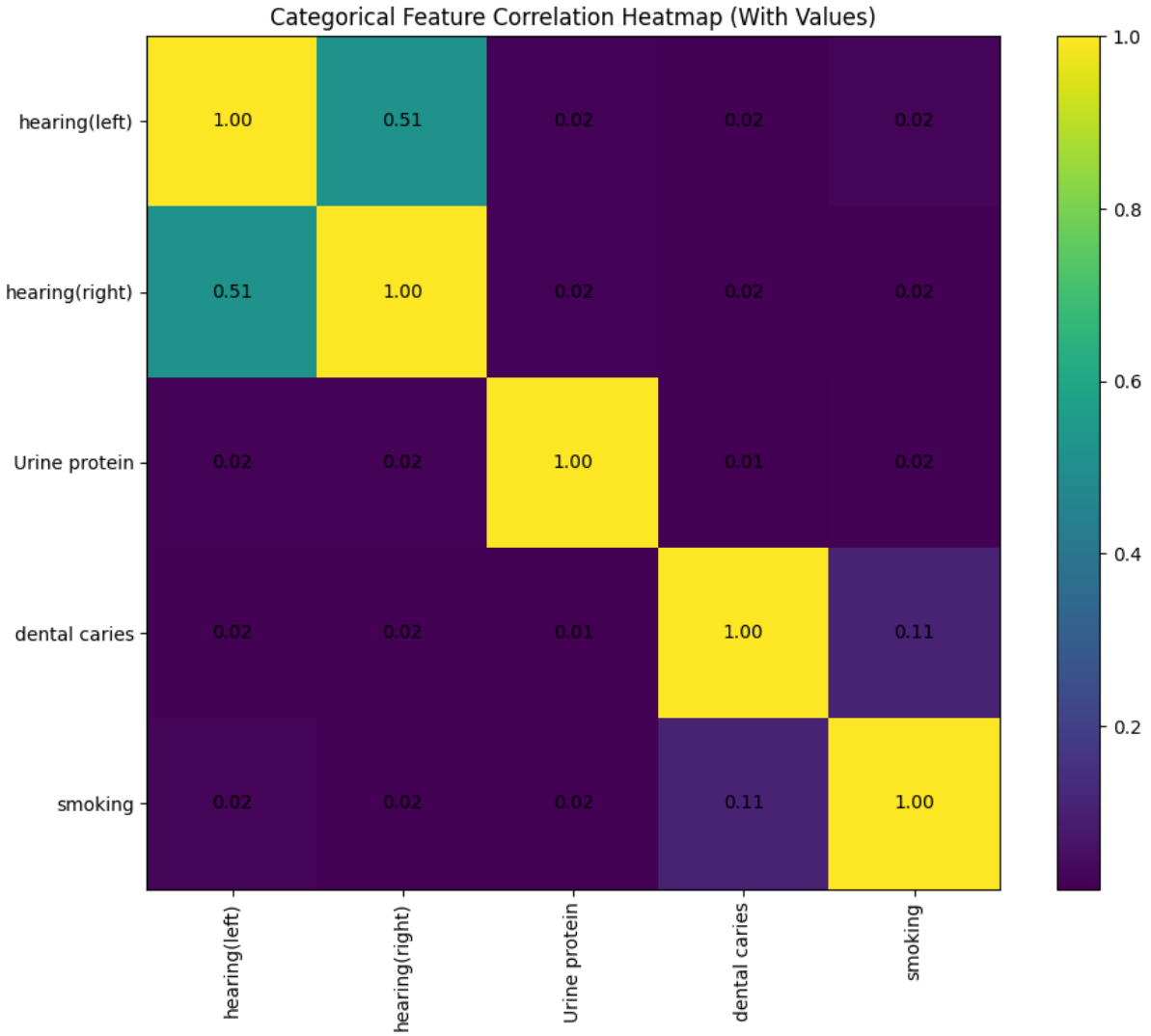
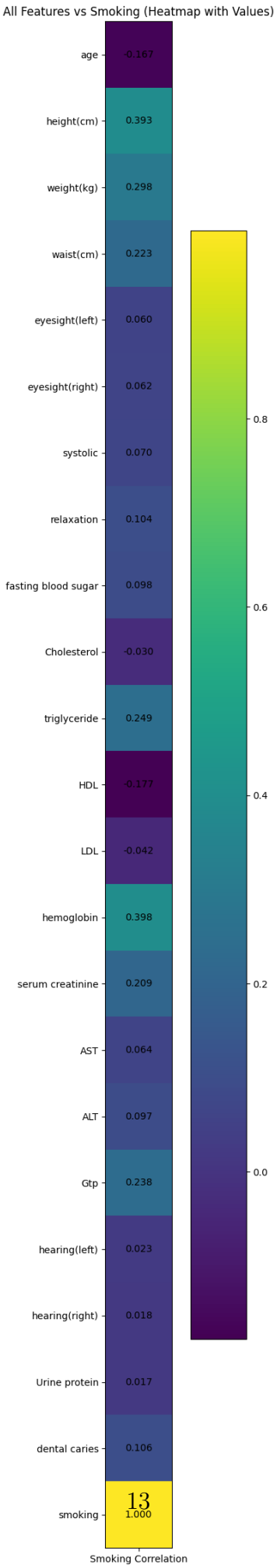


Figure 12: Correlation Matrix for categorical data

The categorical feature correlation matrix shows that hearing (left) and hearing (right) have a moderate positive correlation, which is expected since both measure related hearing ability. All other categorical features, including urine protein and dental caries, show very low correlation with each other, indicating that they are largely independent. The correlation between dental caries and smoking is slightly positive but still weak, suggesting a limited direct relationship in the dataset. Overall, the low correlations among most categorical features reduce the risk of multicollinearity and make them suitable for use in classification models.


All Features vs Smoking (Heatmap with Values)
age
-0.167
height(cm)
0.393
weight(kg)
0.298
waist(cm)
0.223
eyesight(left)
0.060
eyesight(right)
0.062
systolic
0.070
relaxation
0.104
fasting blood sugar
0.098
Cholesterol
-0.030
triglyceride
0.249
HDL
-0.177
LDL
-0.042
hemoglobin
0.398
serum creatinine
0.209
AST
0.064
ALT
0.097
Gtp
0.238
hearing(left)
0.023
hearing(right)
0.018
Urine protein
0.017
dental caries
0.106
smoking
13
1.000
Smoking Correlation
0.8
0.6
0.4
0.2
0.0

From the correlation plot, hemoglobin shows the highest positive correlation with smoking, indicating that smokers tend to have higher hemoglobin levels. Height, weight, triglyceride, waist, serum creatinine, and GTP also show moderate positive correlation, suggesting that these features are important indicators of smoking status. HDL shows a negative correlation, which means HDL levels tend to decrease in smokers. Age also shows a slight negative correlation, while most other features have very weak correlation with smoking, indicating a limited direct linear relationship.

From the numerical correlation heatmap, we observe strong relationships between medically related features such as weight–waist, systolic–relaxation, cholesterol–LDL, and AST–ALT. The categorical correlation plot shows that most categorical features are weakly correlated with each other and with smoking, except hearing left and right. From the feature-to-target correlation plot, hemoglobin, triglyceride, GTP, weight, and waist are identified as the most influential features for predicting smoking. Based on these results, feature selection, scaling, and removal of highly correlated redundant features can improve model performance.

5 Data Preprocessing

In the preprocessing of this dataset, we will perform feature engineering to reduce dimensionality and make meaningful features, and then we will normalize the features with StandardScaler.

5.1 Feature Engineering

Feature engineering was performed to improve the predictive capability of the machine learning models by extracting meaningful physiological indicators from the raw bio-signal attributes. The following derived features were created:

- **Body Mass Index (BMI)** was calculated using:

$$\text{BMI} = \frac{\text{weight (kg)}}{(\text{height (cm)}/100)^2}$$

  BMI provides a standardized measure of body fat and is strongly related to smoking-related metabolic health.

- **Waist-to-Height Ratio** was computed as:

$$\text{Waist-to-Height Ratio} = \frac{\text{waist (cm)}}{\text{height (cm)}}$$

  This ratio is a reliable indicator of central obesity and cardiovascular risk.

- **Average Eyesight** was obtained using:

$$\text{Average Eyesight} = \frac{\text{eyesight(left)} + \text{eyesight(right)}}{2}$$

  This reduces redundancy while preserving overall vision information.

- **Average Hearing** was computed as:

$$\text{Average Hearing} = \frac{\text{hearing(left)} + \text{hearing(right)}}{2}$$

  This provides a combined representation of auditory ability.

- **Blood Pressure Category** was created by discretizing systolic blood pressure into medical risk classes:
  - Normal (0): < 120 mmHg
  - Prehypertension (1): 120–139 mmHg
  - Hypertension (2): ≥ 140 mmHg

- **Cholesterol Ratio** was calculated as:

$$\text{Cholesterol Ratio} = \frac{\text{HDL}}{\text{LDL}}$$

  This ratio reflects the balance between good and bad cholesterol and is an important cardiovascular health indicator.

After generating these features, the original attributes used in their computation (weight, height, waist, eyesight, hearing, systolic blood pressure, HDL, and LDL) were removed to prevent redundancy and multicollinearity. This resulted in a compact feature set with higher clinical interpretability and improved learning efficiency for the classification models.

5.2 Feature Normalization

Since the numerical features in the dataset are measured on different scales (for example, age ranges in tens while biochemical features such as triglycerides, AST, ALT, and GTP can reach several hundreds), feature normalization is necessary to ensure that no single feature dominates the learning process. Without normalization, machine learning models such as Logistic Regression, Support Vector Machine, and Neural Networks may become biased toward high-magnitude features and produce unstable or inaccurate results. In this study, **Standard Scaling (Z-score Normalization)** was applied, where each feature is transformed to have zero mean and unit variance using the following formula:

$$Z = \frac{X - \mu}{\sigma}$$

where X is the original feature value, μ is the mean, and σ is the standard deviation of the feature. This normalization step improves model convergence, stabilizes gradient-based learning, and significantly enhances the performance of distance-based and optimization-based classifiers used in this project.

6 Model Development

In this section, first we will find optimal parameters for all 3 models: logistic regression, support vector machine, and neural networks. Then we will compare the accuracy of each model and what could be the reason behind their accuracies.

6.1 Hyperparameter Tuning using Grid Search Cross-Validation

Grid Search Cross-Validation (GridSearchCV) is a systematic method used to identify the optimal combination of hyperparameters by exhaustively searching through a predefined parameter space. It evaluates each combination using cross-validation and selects the set that provides the highest accuracy. In this study, GridSearchCV was applied to Logistic Regression, Support Vector Machine, and Neural Network models.

6.1.1 Logistic Regression Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
C	Controls regularization strength	0.1, 1, 10
max_iter	Maximum training iterations	200, 400

Table 4: Logistic Regression Grid Search Parameters

Best Parameters Obtained:

$$\{C = 0.1,\ \text{max\_iter} = 200\}$$

6.1.2 Support Vector Machine (SVM) Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
C	Controls margin and misclassification penalty	0.1, 1

Table 5: SVM Grid Search Parameters

Best Parameters Obtained:

$$\{C = 0.1\}$$

6.1.3 Neural Network Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
hidden_layer_sizes	Number of neurons in hidden layer	(8,), (16,)
max_iter	Maximum number of training epochs	150

Table 6: Neural Network Grid Search Parameters

Best Parameters Obtained:

$$\{\text{hidden\_layer\_sizes} = (16,),\ \text{max\_iter} = 150\}$$

6.1.4 Limitations of Grid Search and Transition to Random Search

Although Grid Search provides accurate hyperparameter selection, it is computationally expensive and time-consuming because it evaluates every possible parameter combination. This becomes inefficient as the number of hyperparameters increases. Therefore, to reduce training time while still achieving high model performance, we switch to **Random Search Cross-Validation**, which samples random combinations from the parameter space and provides faster optimization with comparable accuracy.

6.2 Hyperparameter Tuning using Random Search Cross-Validation

Random Search Cross-Validation (RandomizedSearchCV) is an efficient alternative to Grid Search, where a fixed number of random hyperparameter combinations are sampled from a given range. Instead of testing all possible combinations, Random Search explores the search space more efficiently and significantly reduces computational cost while maintaining strong model performance. In this study, RandomizedSearchCV was applied to Logistic Regression, Support Vector Machine, and Neural Network models.

6.2.1 Logistic Regression Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
C	Controls regularization strength	10^{-2} to 10^{2} (log scale)
max_iter	Maximum training iterations	200, 300, 400

Table 7: Logistic Regression Random Search Parameters

Best Parameters Obtained:

$$\{C = 0.0774,\ \text{max\_iter} = 400\}$$

6.2.2 Support Vector Machine (SVM) Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
C	Controls margin and misclassification penalty	10^{-2} to 10^{1} (log scale)

Table 8: SVM Random Search Parameters

Best Parameters Obtained:

$$\{C = 10.0\}$$

6.2.3 Neural Network Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
hidden_layer_sizes	Number of neurons in hidden layer	(8,), (16,)

Table 9: Neural Network Random Search Parameters

Best Parameters Obtained:

$$\{\text{hidden\_layer\_sizes} = (16,)\}$$

6.2.4 Limitations of Random Search and Transition to Optuna

Although Random Search is faster than Grid Search and explores a wider hyperparameter space, it still relies on random sampling and does not learn from previous trials. As a result, it may miss optimal regions of the parameter space and can require many iterations to achieve the best performance. To overcome these limitations and perform intelligent hyperparameter optimization, we further adopt **Optuna**, which uses adaptive Bayesian optimization to efficiently search for optimal hyperparameters with fewer trials and higher accuracy.

6.3 Hyperparameter Optimization using Optuna

Optuna is an advanced hyperparameter optimization framework based on Bayesian optimization. Unlike Grid Search and Random Search, Optuna intelligently selects new hyperparameter values based on the performance of previous trials, which allows it to converge faster toward optimal solutions with fewer evaluations. In this study, Optuna was applied to Logistic Regression, Support Vector Machine, and Neural Network models.

6.3.1 Logistic Regression Hyperparameters

Hyperparameter	**Purpose**	**Search Range**
C	Controls regularization strength	10^{-2} to 5 (log scale)

Table 10: Logistic Regression Optuna Parameters

Best Parameters Obtained:

$$\{C = 0.4478\}$$

6.3.2 Support Vector Machine (SVM) Hyperparameters

Hyperparameter	**Purpose**	**Search Range**
C	Controls margin and misclassification penalty	10^{-2} to 5 (log scale)

Table 11: SVM Optuna Parameters

Best Parameters Obtained:

$$\{C = 0.0125\}$$

6.3.3 Neural Network Hyperparameters

Hyperparameter	**Purpose**	**Search Range**
hidden_layer_sizes	Number of neurons in hidden layers	(32,), (64,), (64, 32)
alpha	Regularization strength	10^{-5} to 10^{-2} (log scale)
learning_rate_init	Initial learning rate	10^{-4} to 10^{-2} (log scale)
activation	Activation function	relu, tanh

Table 12: Neural Network Optuna Parameters

Best Parameters Obtained:

$$\{\text{hidden\_layer\_sizes} = (16,)\}$$

6.3.4 Final Conclusion on Hyperparameter Optimization

Among the three tuning techniques, Grid Search provided exhaustive but slow optimization, while Random Search improved efficiency with limited intelligence. Optuna proved to be the most efficient approach as it adaptively guided the search toward better hyperparameter regions, resulting in faster convergence and improved model performance. Therefore, Optuna is chosen as the final and most reliable hyperparameter tuning method for this study.

6.4 Accuracy Comparition

Tuning Method	Logistic Regression	SVM	Neural Network
Grid Search	0.71	0.71	0.74
Random Search	0.71	0.71	0.74
Optuna	0.71	0.71	0.75

Table 13: Model Accuracy Comparison (Rounded to 2 Decimal Places)

6.5 Interpretation of Model Performance

From the accuracy results, it is observed that all three models—Logistic Regression, SVM, and Neural Network—perform in a similar range of approximately 71% to 75%. Logistic Regression achieves an accuracy of about 71% across all tuning methods, indicating that the relationship between the features and the target is not purely linear, which limits its performance. The SVM model also shows a similar accuracy of around 71%, suggesting that even with margin-based optimization, the dataset does not exhibit a strong separable boundary for classification. The Neural Network consistently outperforms both Logistic Regression and SVM, especially with Optuna tuning where it reaches the highest accuracy of 75%. This improvement is due to the Neural Network's ability to capture non-linear patterns and complex feature interactions present in the bio-signal data. The slight improvement achieved using Optuna indicates that intelligent hyperparameter optimization helps the model converge to a better solution compared to Grid Search and Random Search. Overall, the results show that the dataset contains complex non-linear relationships, making the Neural Network the most suitable model among the three for smoker status prediction.

7 Conclusion

In this study, smoker status prediction was performed using Logistic Regression, Support Vector Machine, and Neural Network models with proper data preprocessing and feature engineering. The Neural Network achieved the highest accuracy of 75%, indicating its ability to learn complex non-linear patterns, while Logistic Regression and SVM achieved around 71% accuracy. This confirms that advanced models with optimized hyperparameters are more effective for accurate smoker detection.

Forest Cover Type

8 Introduction

Forests play a crucial role in maintaining ecological balance, supporting biodiversity, and regulating the climate. However, different types of trees grow under different environmental and geographical conditions such as soil type, elevation, slope, and distance from water sources. Identifying the correct forest cover type manually over large forest regions is time-consuming and inefficient. To overcome this problem, machine learning techniques can be used to automatically predict the type of tree cover based on surrounding environmental characteristics. In this study, we analyze the Forest Cover Type dataset from the Roosevelt National Forest and examine each feature to understand how different geographical and soil-related factors influence the growth of various tree species and help in accurate forest cover type classification.

9 Dataset Description

In the forest cover type, we have 581012 data points and 54 features and a target column. But out of 54, we have 40 columns titled as soil1, soil2, so I created one column titled as soil type. So after converging features, the shape of the data is 581012 x 13 including the target variable.

Feature	**Description**	**Type**
Elevation	Elevation above sea level (in meters)	Numerical
Aspect	Direction the slope faces (in degrees)	Numerical
Slope	Steepness of the terrain (in degrees)	Numerical
Horizontal Distance To Hydrology	Horizontal distance to the nearest water source	Numerical
Vertical Distance To Hydrology	Vertical distance to the nearest water source	Numerical
Horizontal Distance To Roadways	Horizontal distance to the nearest roadway	Numerical
Hillshade 9am	Hillshade index at 9 AM based on sun angle	Numerical
Hillshade Noon	Hillshade index at 12 PM based on sun angle	Numerical
Hillshade 3pm	Hillshade index at 3 PM based on sun angle	Numerical
Horizontal Distance To Fire Points	Horizontal distance to the nearest fire ignition point	Numerical
Wilderness Area	Designated wilderness area category	Categorical
Soil Type	Type of soil present in the region	Categorical
Cover Type	Forest cover type (Target variable)	Categorical

Table 14: Feature Description of Forest Cover Type Dataset

10 Exploratory Data Analysis (EDA)

In the Exploratory Data Analysis (EDA), we will deep dive into each feature, what are the values of each feature, how well it is distributed, and its relation with the target feature.

10.1 Dataset Overview

In this section, we will analyze shape and structure of the data.

10.1.1 Dataset Dimensions

The training dataset contains data points with 12 features, excluding the target feature. The training dataset contains 581012 data points.

	elevation	aspect	slope	hydro_h	hydro_v	road_dist	hill_9am	hill_noon	hill_3pm	fire_dist	wilderness	soil	cover
0	2596	51	3	258	0	510	221	232	148	6279	1	29	5
1	2590	56	2	212	-6	390	220	235	151	6225	1	29	5
2	2804	139	9	268	65	3180	234	238	135	6121	1	12	2
3	2785	155	18	242	118	3090	238	238	122	6211	1	30	2
4	2595	45	2	153	-1	391	220	234	150	6172	1	29	5

Figure 14: Dataset first 5 rows

10.1.2 Feature Types

In the dataset, it is important to divide features into numerical and categorical. Here, in the categorical, we do not have strings like "Yes" or "No"; we only have numbers, so here we need to first calculate how many unique values each feature contains, and based on that, we need to make a threshold beyond which every feature will be treated as a numerical feature and at or below the threshold will be treated as a categorical feature.

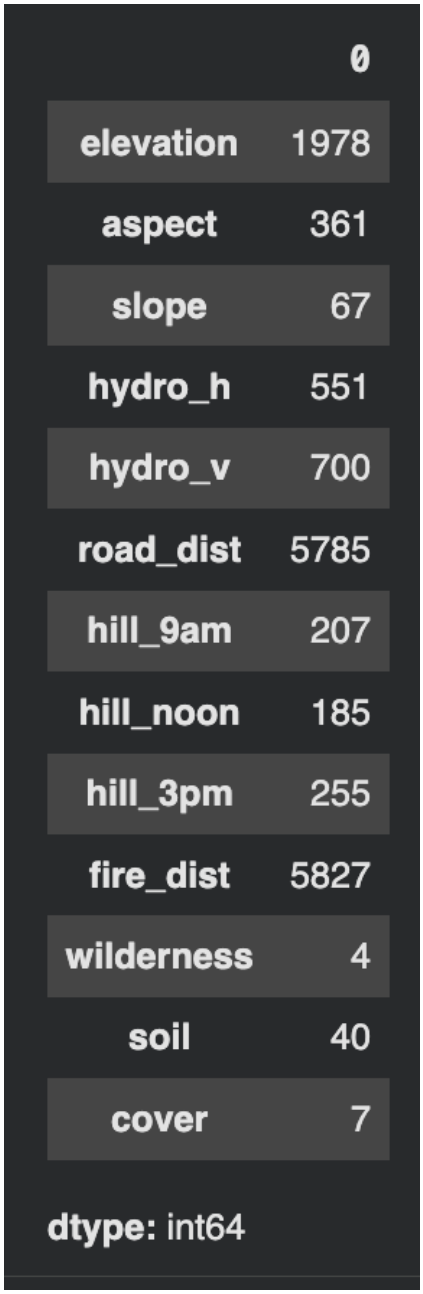


Figure 15: Unique values in each feature

Here we will keep the threshold as 40, so all the features equal to or less than 40 will be considered as categorical features and others as numerical features.

Feature Type	**Feature Names**
Numerical	elevation, aspect, slope, hydro_h, hydro_v, road_dist, hill_9am, hill_noon, hill_3pm, fire_dist
Categorical	wilderness, soil, cover

Table 15: Numerical and Categorical Features

10.1.3 Range of the feature values

We will divide data into two parts, one for the numerical values and one for the categorical values. Purpose behind doing this is we can see if there is any outlier in the numerical value or not. Same for the categorical value.

	elevation	aspect	slope	hydro_h	hydro_v
min	1859.000000	0.000000	0.000000	0.000000	-173.000000
max	3858.000000	360.000000	66.000000	1397.000000	601.000000
mean	2959.365301	155.656807	14.103704	269.428217	46.418855
var	78391.451413	12524.680949	56.073765	45177.228564	3398.334030

	road_dist	hill_9am	hill_noon	hill_3pm	fire_dist
min	0.000000e+00	0.000000	0.000000	0.000000	0.000000e+00
max	7.117000e+03	254.000000	254.000000	254.000000	7.173000e+03
mean	2.350147e+03	212.146049	223.318716	142.528263	1.980291e+03
var	2.431276e+06	716.626947	390.801387	1464.939588	1.753493e+06

Figure 16: Range for the numerical values

The numerical feature summary shows that the forest regions lie at high elevations, ranging from about 1859 to 3858 meters, with an average elevation near 2960 meters, indicating predominantly mountainous terrain. Aspect spans the full range from 0 to 360 degrees, confirming that slopes face all possible directions, while slope values reach up to 66 degrees, showing the presence of both flat and very steep areas. The hydrology distances indicate that some locations are very close to water sources, while others are far away, with horizontal hydrology distance extending up to about 1400 meters and vertical distance varying significantly, including negative values indicating locations below nearby water sources. Road and fire distances also vary widely, reaching over 7000 meters, which suggests a mix of easily accessible and remote forest areas. Hillshade values at different times of the day lie between 0 and 254, reflecting varying sunlight exposure due to terrain orientation. Overall, the large variances across most features indicate high geographical diversity in the forest environment, which is important for distinguishing different forest cover types.

10.1.4 Check for duplicates and missing values

In the dataset, we do not have null values or duplicate values. So we are good to go without removing any data point from the dataset.

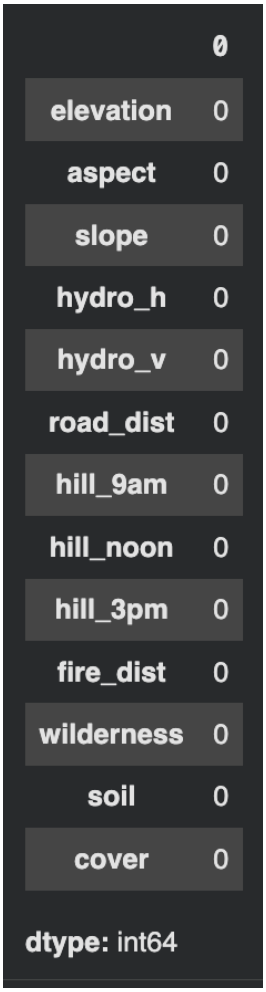


Figure 17: Missing values in each attributes

Figure 18: Duplicate rows in the dataset

10.2 Target Variable Analysis

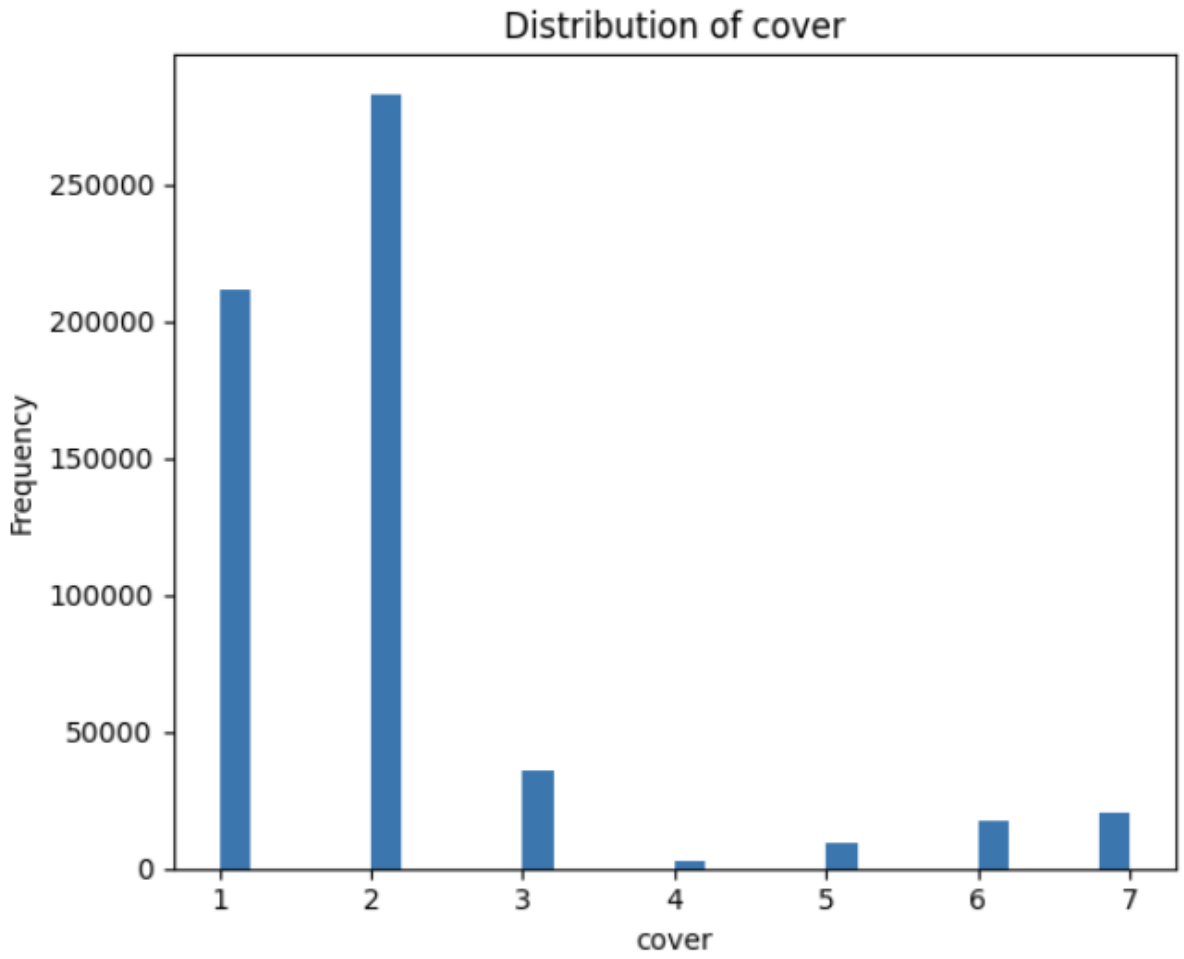


Figure 19: Target values distribution

Cover Type	Number of Samples
1	211840
2	283301
3	35754
4	2747
5	9493
6	17367
7	20510

Table 16: Distribution of Forest Cover Types

The distribution of the target variable is highly imbalanced, with Cover Type 2 and Cover Type 1 being the most dominant classes. Cover Types 4, 5, 6, and 7 have significantly fewer samples, especially Cover Type 4, which is extremely underrepresented. This class imbalance can affect model performance and may require techniques such as class weighting or resampling.

10.3 Univariate Analysis

10.3.1 Numerical Features

In the univariate analysis, we will look at each feature very closely, so we will start with numerical features.

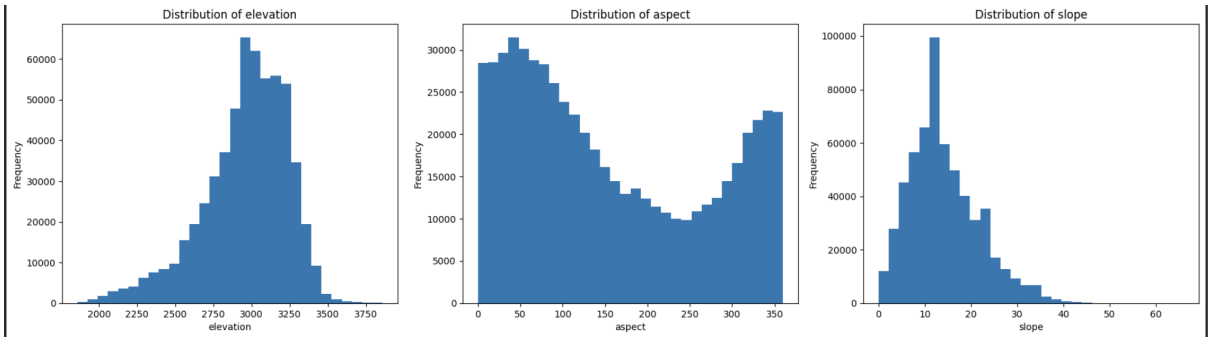


Figure 20: Numerical Features Distribution

The elevation distribution shows that most forest areas lie between approximately 2600 and 3300 meters, indicating that the dataset mainly represents high-altitude mountainous regions. The aspect distribution is spread across the full range of 0 to 360 degrees, with slightly higher concentrations around certain directional ranges, showing that the forest slopes face multiple directions rather than being limited to a single orientation. The slope distribution is right-skewed, with the majority of areas having gentle to moderate slopes below 20 degrees, while very steep slopes are relatively rare. These patterns highlight that tree growth in the Roosevelt National Forest is influenced by a combination of high elevation, varied slope direction, and mostly moderate terrain steepness.

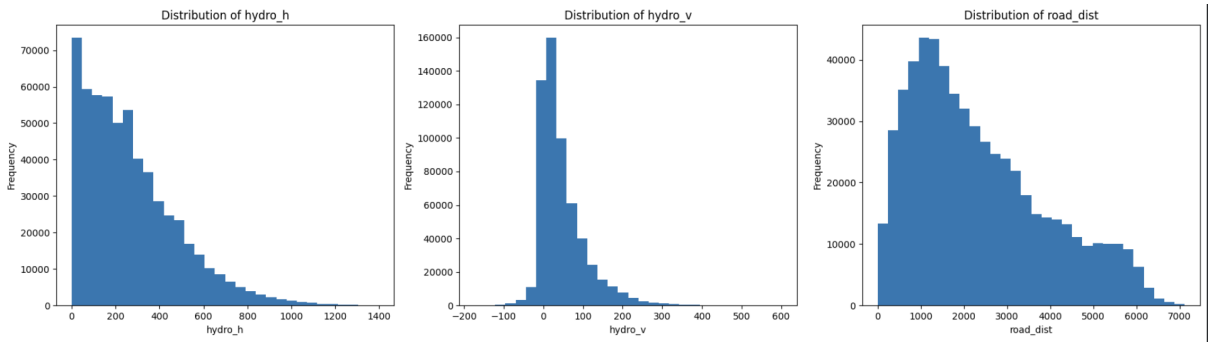


Figure 21: Numerical Features Distribution

The horizontal distance to hydrology (hydro h) shows a strong right-skewed distribution, indicating that most forest areas are located close to water sources, while only a smaller proportion lies far away, which highlights the importance of water availability for vegetation growth. The vertical distance to hydrology (hydro v) is concentrated around zero with both positive and negative values, showing that many locations are at similar elevation levels to nearby water bodies, while some regions are either above or below them. The distance to roadways (road dist) displays a wide spread with higher frequencies at lower to moderate distances and gradually decreasing toward larger values, indicating

that many forest regions are relatively accessible while some remain remote. These patterns suggest that proximity to water and human infrastructure plays a significant role in shaping forest cover distribution.

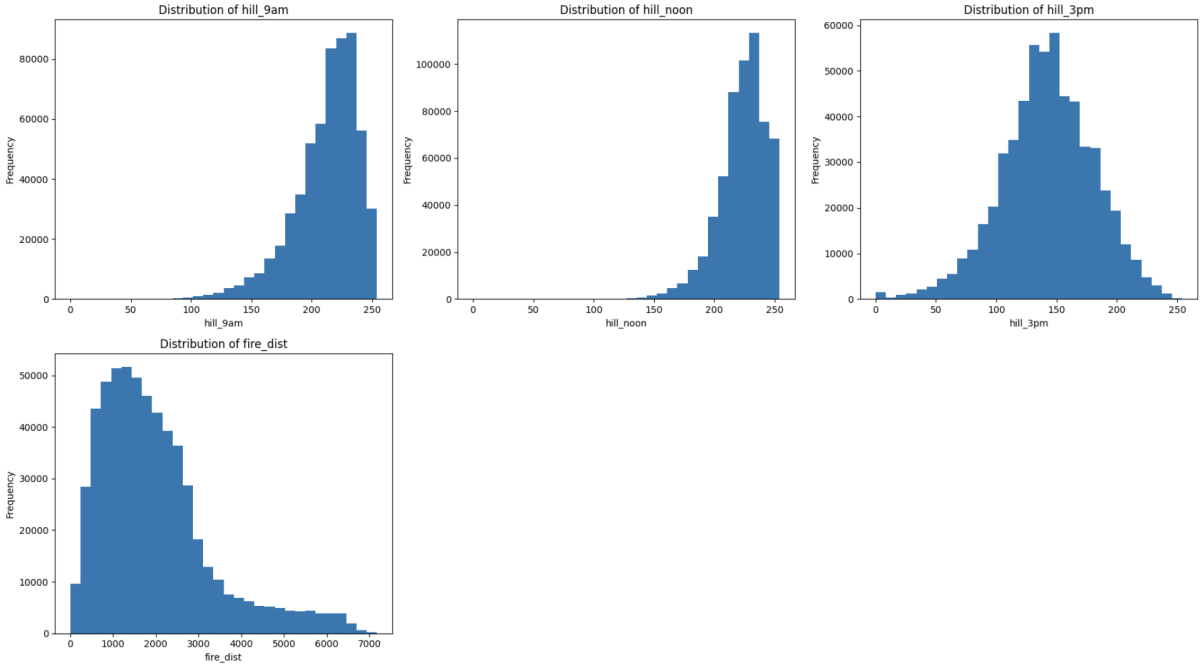


Figure 22: Numerical Features Distribution

The hillshade features at 9 AM, noon, and 3 PM show distinct patterns of sunlight exposure across the forest terrain. Hillshade at 9 AM and noon is highly concentrated toward higher values, indicating that a large portion of the forest receives strong morning and midday sunlight, while hillshade at 3 PM shows a more symmetric distribution around mid-range values, reflecting varying afternoon illumination due to terrain orientation. The distance to fire points (fire dist) exhibits a right-skewed distribution, with most areas located closer to fire ignition points and progressively fewer regions at very large distances. These patterns suggest that sunlight exposure and proximity to historical fire locations are important environmental factors influencing forest cover type.

Overall, The numerical analysis shows that the forest region is mainly located at high elevations, which strongly influences the type of vegetation found in the area. Most regions have gentle to moderate slopes and receive substantial sunlight throughout the day, making hillshade an important factor in tree growth. The majority of forest areas are located relatively close to water sources, roads, and fire points, highlighting the combined influence of natural resources and human activity on forest distribution. Additionally, many of the distance-based features follow a right-skewed distribution, indicating that while most regions are easily accessible, a smaller number of areas remain highly remote, contributing to ecological diversity.

10.3.2 Categorical Features

In the categorical feature, we only have 2 features and 1 target; first we will analyze the distribution of features across categories.

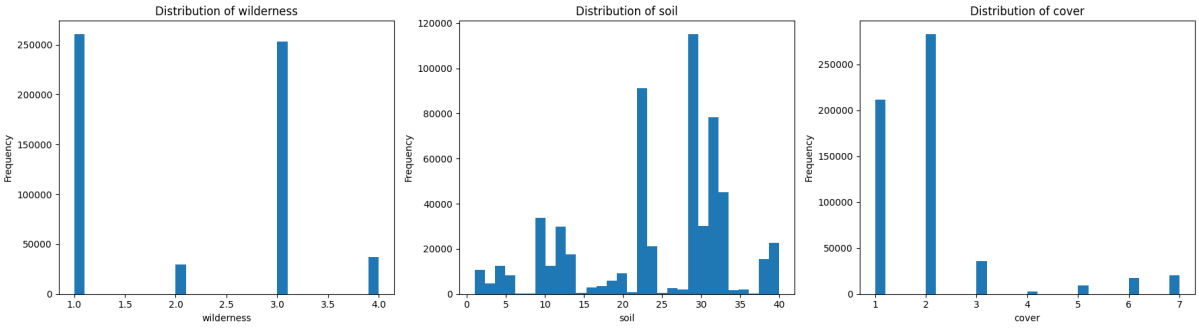


Figure 23: Categorical features Distribution

The wilderness feature shows that most observations belong to Wilderness Areas 1 and 3, indicating that these protected regions dominate the dataset, while Areas 2 and 4 are comparatively underrepresented. The soil type distribution is highly uneven, with a few soil categories having very high frequencies while many others appear only rarely, reflecting strong variability in soil composition across the forest. The cover type distribution is also clearly imbalanced, where Cover Types 1 and 2 are the most dominant, whereas Cover Types 4, 5, 6, and 7 have significantly fewer samples. This overall imbalance across categorical features suggests that some classes may strongly influence the model during training and may require techniques such as class weighting or resampling for better classification performance.

During preprocessing, numerical features should be normalized to bring them to a common scale, while categorical features such as wilderness and soil should be properly encoded. Since the dataset shows strong class imbalance in the target variable, techniques like class weighting or resampling should be applied to improve model learning. Additionally, redundant or highly correlated features can be merged or removed to reduce dimensionality and noise.

10.4 Correlation and Multicollinearity

It is essential to know about the correlation between features and the target column. I am showcasing 3 heat maps. One is between numerical features, one is between categorical features, and one is between features and the target.

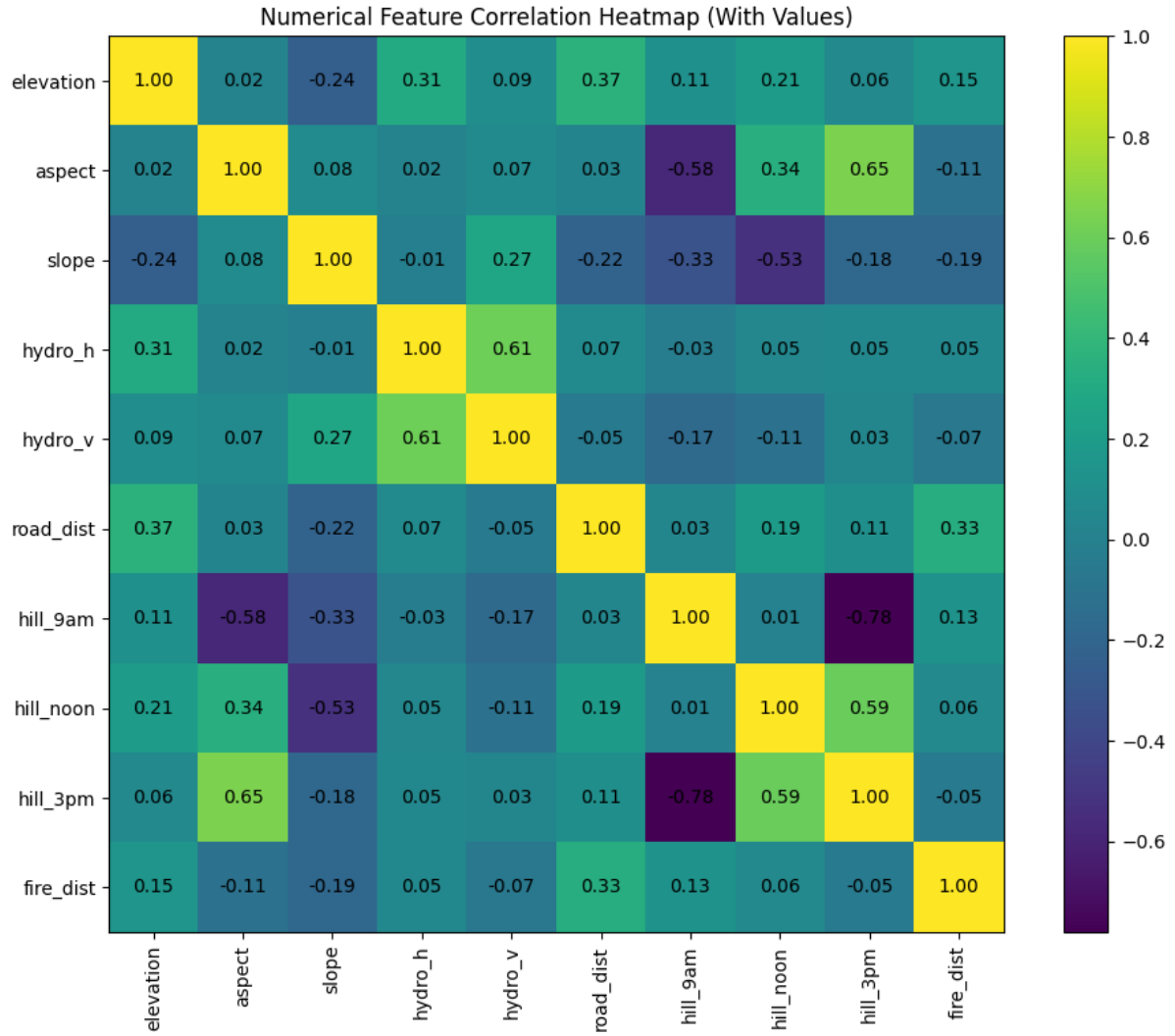


Figure 24: Correlation Matrix for Numerical Data

This correlation heatmap shows several meaningful relationships among the numerical features. The strongest positive correlation is observed between horizontal and vertical hydrology distances, indicating that areas closer horizontally to water sources also tend to be closer vertically. The hillshade features show strong interrelationships, especially a strong negative correlation between hillshade at 9am and 3pm and a positive correlation between hillshade at noon and 3pm, which reflects sunlight angle variations during the day. Aspect is moderately correlated with hillshade at 3pm, showing terrain orientation affects sunlight exposure. Elevation has weak to moderate correlations with road distance and hydrology, suggesting accessibility and water proximity change with height. Overall, most features are not highly correlated, which is beneficial for machine learning as it reduces multicollinearity.

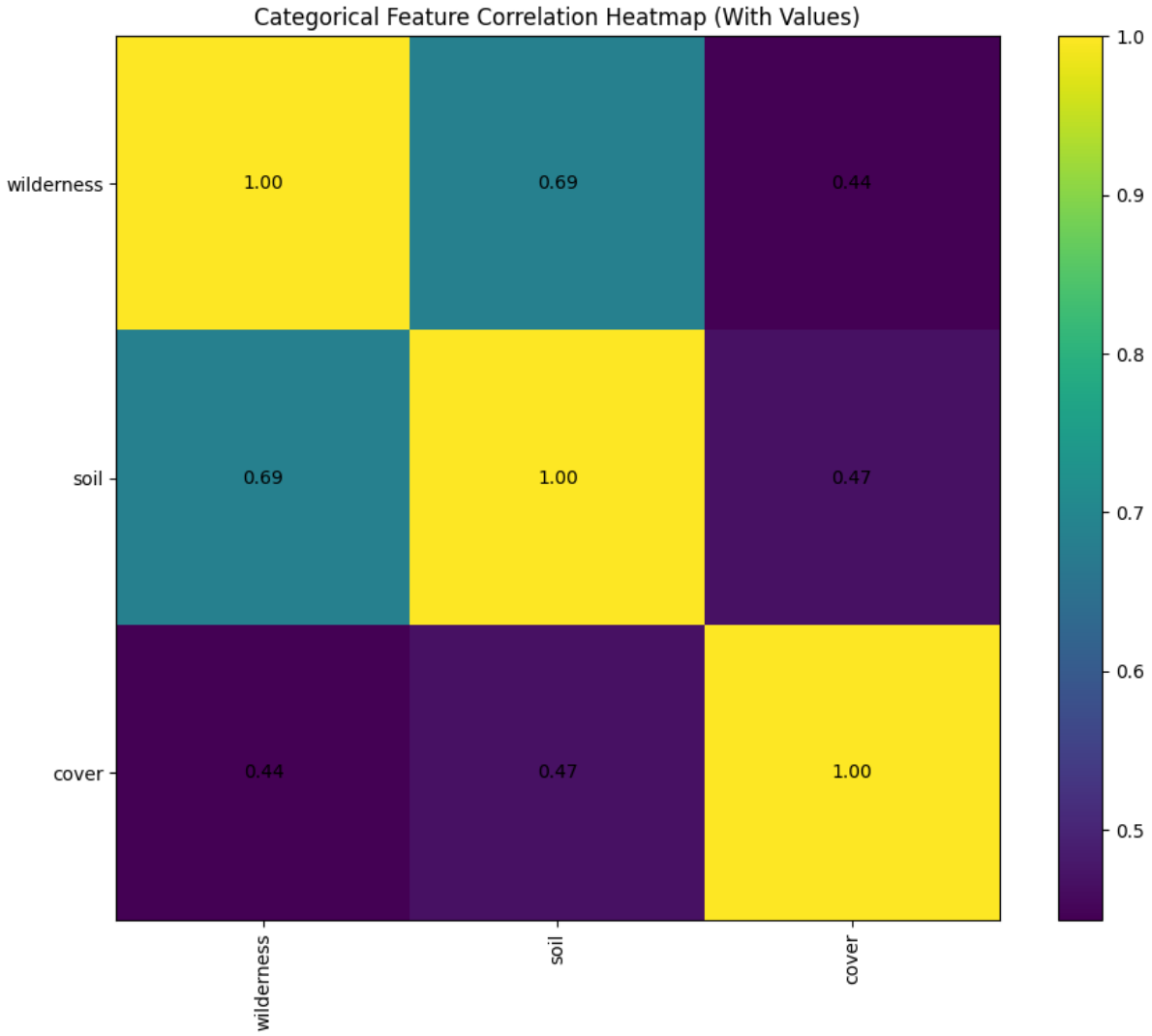
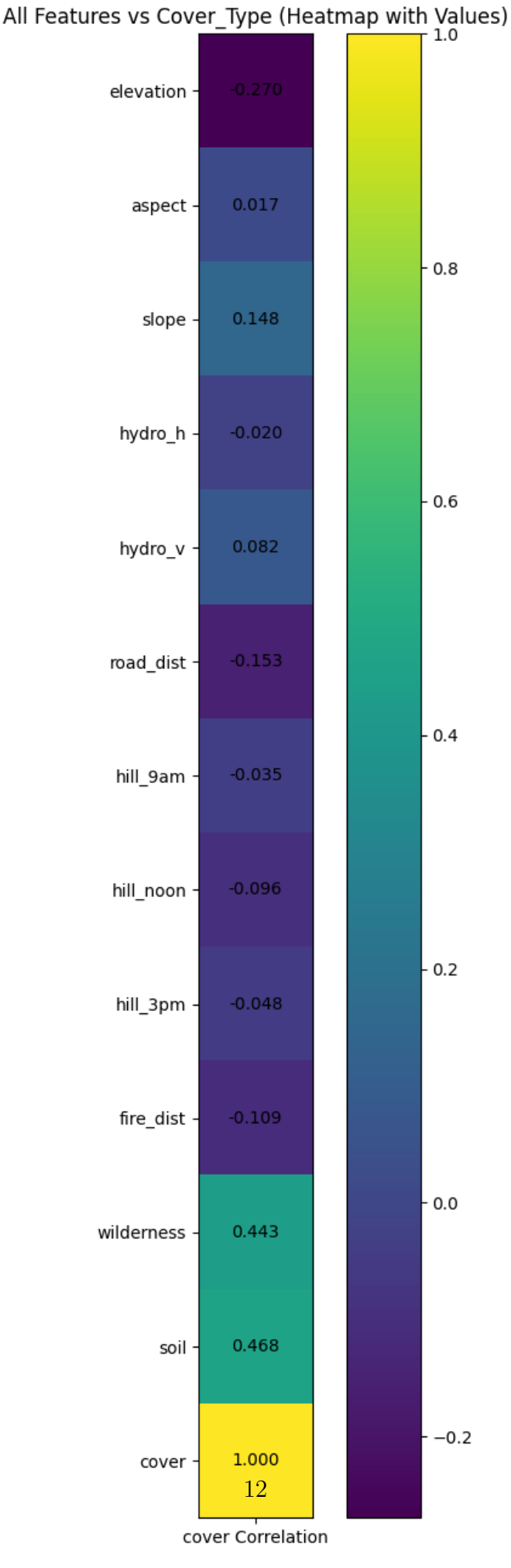


Figure 25: Correlation Matrix for categorical data

This categorical correlation heatmap shows a moderate positive relationship between wilderness and soil, indicating that certain soil types are strongly associated with specific wilderness areas. The cover type has a moderate correlation with both soil and wilderness, which suggests that tree species distribution is influenced by both terrain category and soil conditions. None of the correlations are extremely high, meaning that while these categorical features are related, they still provide independent and useful information for predicting forest cover type, which is beneficial for building a robust classification model.


All Features vs Cover_Type (Heatmap with Values)
elevation
aspect
slope
hydro_h
hydro_v
road_dist
hill_9am
hill_noon
hill_3pm
fire_dist
wilderness
soil
cover
-0.270
0.017
0.148
-0.020
0.082
-0.153
-0.035
-0.096
-0.048
-0.109
0.443
0.468
1.000
12
cover Correlation
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

This heatmap shows that among all features, the categorical variables—wilderness and soil have the strongest positive correlation with the cover type, indicating that forest cover is highly influenced by ecological region and soil characteristics. Most numerical features such as elevation, road distance, fire distance, and hillshade values show weak negative correlations, suggesting they still affect cover type but not in a strongly linear way. Features like aspect and hydro distances have almost no linear relationship with cover type. Overall, this indicates that categorical environmental factors play a more dominant role than individual numerical features, and nonlinear models may be better suited to capture these complex relationships.

Overall, during preprocessing, numerical features should be scaled or normalized to handle wide value ranges and improve model stability, while categorical features like wilderness and soil should be properly encoded using label or one-hot encoding. Additionally, highly skewed features can be log-transformed, and weak or redundant features can be removed to improve model performance.

11 Preprocessing

11.1 Standard Scaling

In this data, we only have 12 features, so we do not need to do much on the feature side. As ranges are different, we only need to do standard scaling.

12 Model Development

In this section, first we will find optimal parameters for all 3 models: logistic regression, support vector machine, and neural networks. Then we will compare the accuracy of each model and what could be the reason behind their accuracies.

12.1 Hyperparameter Tuning using Grid Search Cross-Validation for Forest Cover Type

Grid Search Cross-Validation (GridSearchCV) is a systematic approach used to determine the optimal set of hyperparameters by exhaustively evaluating all possible combinations within a specified range. Each combination is assessed using cross-validation, and the one yielding the highest classification accuracy is selected. In this study, GridSearchCV was applied to Logistic Regression, Support Vector Machine (SVM), and Neural Network models for the Forest Cover Type classification task.

12.1.1 Logistic Regression Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
C	Controls regularization strength	0.1, 1, 5

Table 17: Logistic Regression Grid Search Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{C = 5\}$$

12.1.2 Neural Network Hyperparameters

Hyperparameter	**Purpose**	**Range Used**
hidden_layer_sizes	Number of neurons in hidden layer	(16,), (32,)
max_iter	Maximum number of training epochs	150

Table 18: Neural Network Grid Search Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{\text{hidden\_layer\_sizes} = (32,)\}$$

12.1.3 Support Vector Machine (SVM) Hyperparameters

Hyperparameter	Purpose	Range Used
C	Controls margin and misclassification penalty	0.1, 1, 5

Table 19: SVM Grid Search Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{C = 5\}$$

12.1.4 Limitations of Grid Search and Transition to Random Search

Although Grid Search provides an exhaustive search of hyperparameter combinations and ensures optimal selection within the given range, it is computationally expensive and time-consuming, especially for large datasets like the Forest Cover Type dataset. As the number of hyperparameters increases, the total number of model evaluations grows rapidly. To address this limitation and reduce computational cost while maintaining good performance, we transition to **Random Search Cross-Validation**, which explores the hyperparameter space more efficiently by sampling random parameter combinations.

12.2 Hyperparameter Tuning using Random Search Cross-Validation for Forest Cover Type

Random Search Cross-Validation (RandomizedSearchCV) is an efficient alternative to Grid Search, where only a fixed number of random hyperparameter combinations are evaluated instead of all possible combinations. This approach significantly reduces computational cost while still providing strong model performance. In this study, RandomizedSearchCV was applied to Logistic Regression, Support Vector Machine, and Neural Network models for the Forest Cover Type classification task.

12.2.1 Logistic Regression Hyperparameters

Hyperparameter	Purpose	Range Used
C	Controls regularization strength	10^{-2} to 10^{1} (log scale)

Table 20: Logistic Regression Random Search Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{C = 10.0\}$$

12.2.2 Neural Network Hyperparameters

Hyperparameter	Purpose	Range Used
hidden_layer_sizes	Number of neurons in hidden layer	(16,), (32,), (64,)

Table 21: Neural Network Random Search Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{\text{hidden\_layer\_sizes} = (64,)\}$$

12.2.3 Support Vector Machine (SVM) Hyperparameters

Hyperparameter	Purpose	Range Used
C	Controls margin and misclassification penalty	10^{-2} to 10^{1} (log scale)

Table 22: SVM Random Search Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{C = 10.0\}$$

12.2.4 Limitations of Random Search and Transition to Optuna

Although Random Search significantly reduces the computational cost compared to Grid Search and explores a broader hyperparameter space, it still relies on random sampling and does not learn from previous trials. As a result, it may fail to explore the most promising regions of the parameter space efficiently. To overcome this limitation and achieve intelligent and adaptive hyperparameter optimization, we further adopt **Optuna**, which uses Bayesian optimization to guide the search toward optimal hyperparameter values with fewer trials and improved model performance.

12.3 Hyperparameter Optimization using Optuna for Forest Cover Type

Optuna is an advanced hyperparameter optimization framework that uses Bayesian optimization to intelligently search for the best hyperparameter values. Unlike Grid Search and Random Search, Optuna learns from previous trials and focuses on promising regions of the search space, resulting in faster convergence and improved model performance. In this study, Optuna was applied to Logistic Regression, Support Vector Machine (SVM), and Neural Network models for the Forest Cover Type classification task.

12.3.1 Logistic Regression Hyperparameters

Hyperparameter	Purpose	Search Range
C	Controls regularization strength	10^{-2} to 5 (log scale)

Table 23: Logistic Regression Optuna Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{C = 0.2305\}$$

Final Accuracy Achieved:

0.7124

12.3.2 Neural Network Hyperparameters

Hyperparameter	Purpose	Search Range
hidden_layer_sizes	Number of neurons in hidden layers	(32,), (64,), (64, 32)
alpha	Regularization strength	10^{-5} to 10^{-2} (log scale)
learning_rate_init	Initial learning rate	10^{-4} to 10^{-2} (log scale)
activation	Activation function	relu, tanh
solver	Optimization algorithm	adam, lbfgs
batch_size	Batch size for training	32, 64, 128

Table 24: Neural Network Optuna Parameters (Forest Cover Type)

Best Parameters Obtained:

$\{\text{hidden\_layer\_sizes} = (64, 32),\ \alpha = 0.000335,\ \text{learning\_rate\_init} = 0.002428,\ \text{activation} = \text{tanh},\ \text{solver} =$

Final Accuracy Achieved:

0.8576

12.3.3 Support Vector Machine (SVM) Hyperparameters

Hyperparameter	Purpose	Search Range
C	Controls margin and misclassification penalty	10^{-2} to 5 (log scale)

Table 25: SVM Optuna Parameters (Forest Cover Type)

Best Parameters Obtained:

$$\{C = 3.0850\}$$

Final Accuracy Achieved:

0.6696

12.3.4 Final Observation on Optuna Performance

Among all models tuned using Optuna, the Neural Network achieved the highest accuracy of **85.76%**, significantly outperforming Logistic Regression and SVM. This highlights Optuna's ability to efficiently tune complex deep learning models and confirms that Neural Networks are highly effective for modeling the non-linear relationships present in the Forest Cover Type dataset.

12.4 Accuracy Comparition

Tuning Method	Logistic Regression	SVM	Neural Network
Grid Search	0.71	0.67	0.78
Random Search	0.71	0.77	0.80
Optuna	0.71	0.67	0.86

Table 26: Model Accuracy Comparison (Rounded to 2 Decimal Places)

12.5 Interpretation of Model Performance

The results show that among all three models optimized using Optuna, the Neural Network achieved the highest accuracy of 85.76%, indicating its strong ability to capture complex non-linear relationships present in the forest dataset. Logistic Regression achieved moderate performance at around 71%, as it is limited to learning linear decision boundaries. The SVM model showed comparatively lower accuracy, suggesting that it was less effective in handling the high-dimensional and multi-class nature of the forest cover data. Overall, these results confirm that Neural Networks are the most suitable model for forest cover type classification in this study.

GitHub Repository: https://github.com/adityadave29/ML_Project_2