Comparative Analysis of Models for Multiclass Forest Cover Classification

Prepared for

Dr. Sushree S. Behera

Assistant Professor, IIIT Bangalore

By

Vansh N. Doshi [MT2025729 - AI&DS]

Vishnu Dholu [MT2025044 - CSE]

Dec 10, 2025

Abstract

This project investigates the multiclass classification of forest cover types using cartographic data on a highly imbalanced dataset of 581,012 samples. By implementing domain-specific feature engineering—including physics-based terrain metrics and interaction terms—and addressing class sparsity with calculated weights, we evaluated a baseline Logistic Regression model against a custom "Wide and Deep" Neural Network. The analysis reveals that the Neural Network significantly outperforms the linear baseline, achieving 82% accuracy versus 60% and demonstrating superior recall for rare vegetation classes (e.g., 98% for Type 4). These findings confirm that deep learning architectures are essential for capturing the complex, non-linear interactions inherent in terrain-based ecological data.

Table of Contents

1. Executive Summary

This report presents a comparative analysis of machine learning models developed to solve the **Forest Cover Type Classification** problem. The objective was to predict the dominant vegetation type of forest patches using only cartographic variables, aiming to automate ecological mapping without direct field observation.

The analysis utilized the "Covertype" dataset, containing **581,012 samples** with **54 initial features** (reduced to 44 after feature engineering). The target variable comprises seven forest cover types with severe class imbalance; the two dominant classes account for nearly 85% of the data, while rare types represent less than 1%.

Two distinct modeling strategies were implemented: a baseline linear model (**Logistic Regression**) and a non-linear deep learning model (**Neural Network**). Both models utilized a preprocessing pipeline including physics-based feature engineering (e.g., hydrology elevation), rigorous class weighting to penalize minority class errors, and stratified validation. The Neural Network employed a deep, funnel-shaped architecture (512 $\rightarrow$ 32 neurons) with **ReLU activation**, Batch Normalization, and Dropout to prevent overfitting.

Key Findings:

The comparative analysis yielded the following results:

- **Performance Gap:** The Neural Network significantly outperformed the linear baseline, achieving an overall accuracy of **82%** compared to the Logistic Regression's **60%**.
- **Minority Class Detection:** While Logistic Regression achieved high recall for some minority classes, it suffered from very low precision (e.g., 11% for Type 5). In contrast, the Neural Network demonstrated superior generalization, achieving **98% Recall for Type 4** and **99% Recall for Type 7** while maintaining high precision across the board.

Final Recommendation:

Based on the experimental results, the adoption of the Deep Neural Network architecture (5-layer funnel with ReLU activation) is recommended. This model successfully captured the non-linear interactions between elevation, soil, and wilderness areas, offering the most robust balance of accuracy across both common and rare forest cover types.

2. Introduction and Problem Definition

2.1 Project Background

Effective forest resource management is critical for biodiversity conservation, wildfire prevention, and ecosystem sustainability. A fundamental component of this management is the accurate inventorying of forest cover types—specifically, identifying the dominant tree species (e.g., Spruce/Fir, Lodgepole Pine, Cottonwood) present in a given area. Understanding the spatial distribution of these vegetation types allows land managers to make informed decisions regarding timber harvesting, wildlife habitat protection, and reforestation efforts.

Traditionally, determining forest cover types requires intensive fieldwork or the interpretation of expensive remote sensing data (such as aerial photography or satellite imagery). Field assessments are time-consuming, labor-intensive, and often dangerous in rugged terrain, while remote sensing can be costly and limited by cloud cover or resolution.

This project explores a data-driven alternative: predicting forest cover types using only **cartographic variables**—such as elevation, slope, soil type, and distance to water sources. These variables are easily derived from Geographic Information Systems (GIS) and are available at a fraction of the cost of field surveys. By leveraging machine learning algorithms, we aim to automate the classification process, allowing for the scalable and rapid mapping of vast wilderness areas based on their physical geographical characteristics. This approach transforms a resource-intensive manual task into an efficient, automated computational problem.

2.2 Problem Statement

The core objective of this project is to develop a robust multiclass classification model capable of predicting the forest cover type for a given 30×30 meter cell of land. The prediction must be based exclusively on 54 cartographic variables derived from US Geological Survey (USGS) and USFS data, without reliance on direct observation or imagery.

The problem presents several significant challenges:

- **Multiclass Classification:** The model must distinguish between seven distinct forest cover types, ranging from common coniferous species to specific alpine vegetation.

- **Severe Class Imbalance:** The dataset is highly skewed. The two dominant classes (Spruce/Fir and Lodgepole Pine) constitute approximately 85% of the observations, while minority classes like Cottonwood/Willow and Aspen are extremely rare, representing less than 1% of the total data. This imbalance creates a high risk of model bias, where algorithms may ignore rare classes to maximize overall accuracy.
- **Complex Feature Interactions:** The relationship between terrain (e.g., elevation, aspect) and vegetation is non-linear and context-dependent. For instance, a specific elevation might support different tree types depending on the soil composition or the specific wilderness area.

The goal is to build a model that not only achieves high overall accuracy but also maintains high sensitivity (Recall) for the rare, minority forest types, ensuring a complete and ecologically valid classification map.

2.3 Dataset Overview

The analysis utilizes the "Covertype" dataset, a massive collection of cartographic data comprising **581,012 samples**. Each sample represents a 30×30 meter observation patch, characterized by **54 independent features** and a single target variable.

The features represent a mix of continuous and binary variables describing the physical terrain:

- **Topographical Features:** The dataset includes fundamental terrain metrics such as **Elevation** (ranging from approx. 1,800 to 3,800 meters), **Aspect** (azimuth), and **Slope** (degrees). Elevation is noted as the most discriminative feature for classification.
- **Distance Metrics:** To capture proximity to critical resources and landmarks, the dataset includes:
    - Horizontal_Distance_To_Hydrology & Vertical_Distance_To_Hydrology (distance to nearest water).
    - Horizontal_Distance_To_Roadways and Horizontal_Distance_To_Fire_Points.
    - *Note:* These distance features exhibit significant right-skewness and outliers, varying from 0 to over 7,000 meters.
- **Hillshade Indices:** Three features (Hillshade_9am, Hillshade_Noon, Hillshade_3pm) quantify the amount of sunlight the patch receives at specific times of day, ranging from 0 to 255.
- **Wilderness Areas:** Four binary columns indicate which USFS wilderness region the sample belongs to (e.g., Rawah, Neota, Comanche Peak, Cache la Poudre).
- **Soil Types:** The soil composition is represented by 40 binary columns (Soil_Type1 to Soil_Type40). This data is extremely sparse, with many soil types appearing in less than 1% of the samples.

Target Variable:

The target variable, Cover_Type, classifies the dominant forest vegetation into seven distinct integers. The classes are heavily imbalanced, with Type 1 (Spruce/Fir) and Type 2 (Lodgepole Pine) comprising nearly 85% of the data, while Type 4 (Cottonwood/Willow) represents less than 0.5%.

3. Exploratory Data Analysis (EDA)

3.1 Univariate Analysis

3.1.1 Target Variable Distribution

The analysis of the target variable, Cover_Type, reveals a severe class imbalance within the dataset. As illustrated in the bar chart below, the distribution is heavily skewed towards two dominant classes. **Cover Type 1 (Spruce/Fir)** and **Cover Type 2 (Lodgepole Pine)** collectively account for the vast majority of the samples, representing approximately 85% of the entire dataset.

In contrast, the remaining five classes are significantly underrepresented. **Cover Type 4 (Cottonwood/Willow)** is the most extreme minority class, containing only 2,747 samples (less than 0.5% of the total data), followed closely by **Type 5 (Aspen)** . This disparity presents a critical challenge for modeling, as standard algorithms will naturally bias their predictions towards the majority classes to minimize global error, potentially ignoring the ecologically significant minority forest types.

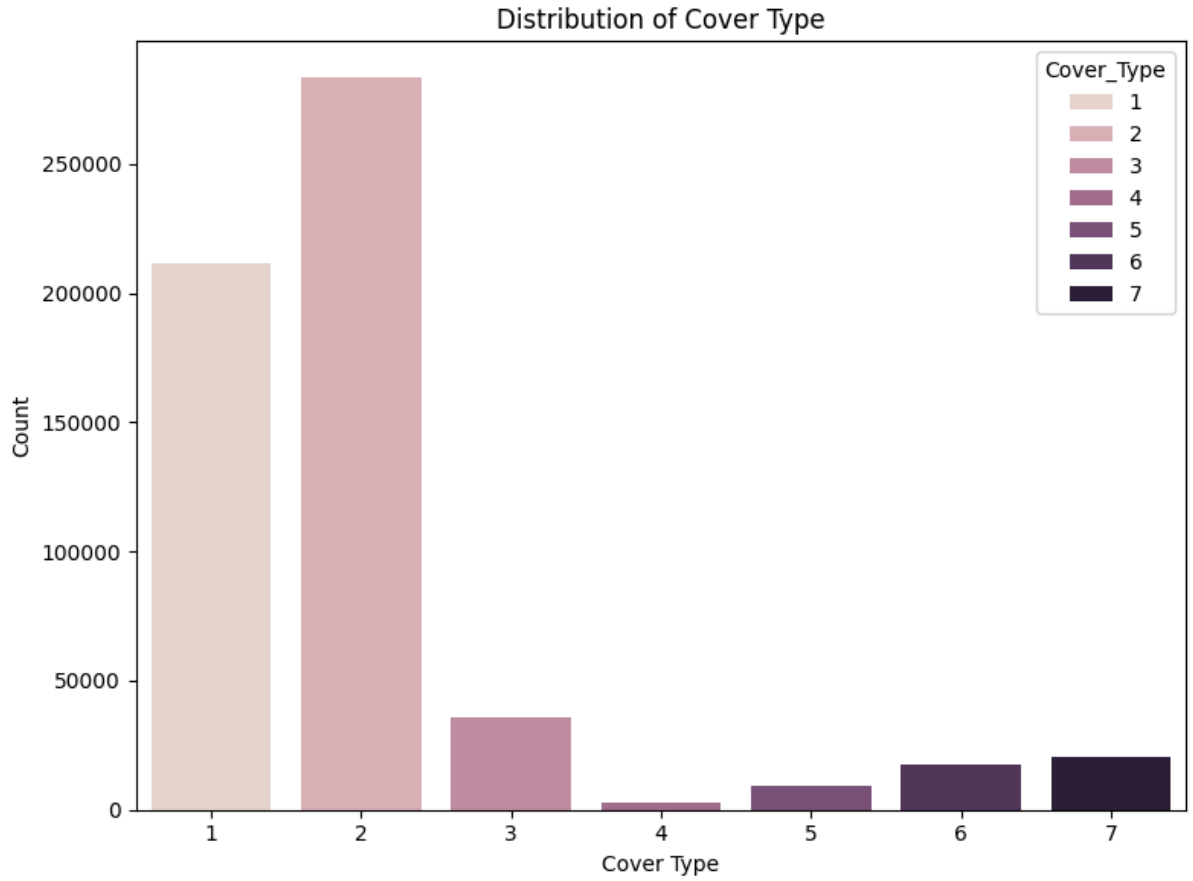

Figure 3.1: Distribution of samples across the seven Forest Cover Types. Note the overwhelming dominance of Types 1 and 2 compared to the sparsity of Types 4, 5, and 6.

3.1.2 Numerical Feature Distributions

The distribution of numerical features was examined using histograms to identify data characteristics that could impact model performance. The analysis highlights two critical issues: varying scales and significant skewness.

First, the input features operate on vastly different magnitudes. For example, **Elevation** ranges from approximately 1,800 to 3,800 meters, whereas **Slope** is concentrated between 0 and 60 degrees, and **Hillshade** indices are bounded between 0 and 255 . Without normalization, algorithms relying on distance calculations or gradient descent (like Neural Networks and Logistic Regression) would be disproportionately influenced by features with larger numerical values .

Second, several features exhibit severe **right-skewness** (positive skew). Specifically, Horizontal_Distance_To_Hydrology, Horizontal_Distance_To_Roadways, and Horizontal_Distance_To_Fire_Points show long "tails" extending to the right, indicating that while most samples are close to these resources, a small subset is located very far away. This non-Gaussian distribution can degrade the performance of linear models, necessitating mathematical transformations (e.g., Log Transformation) to compress the outliers and stabilize the variance.

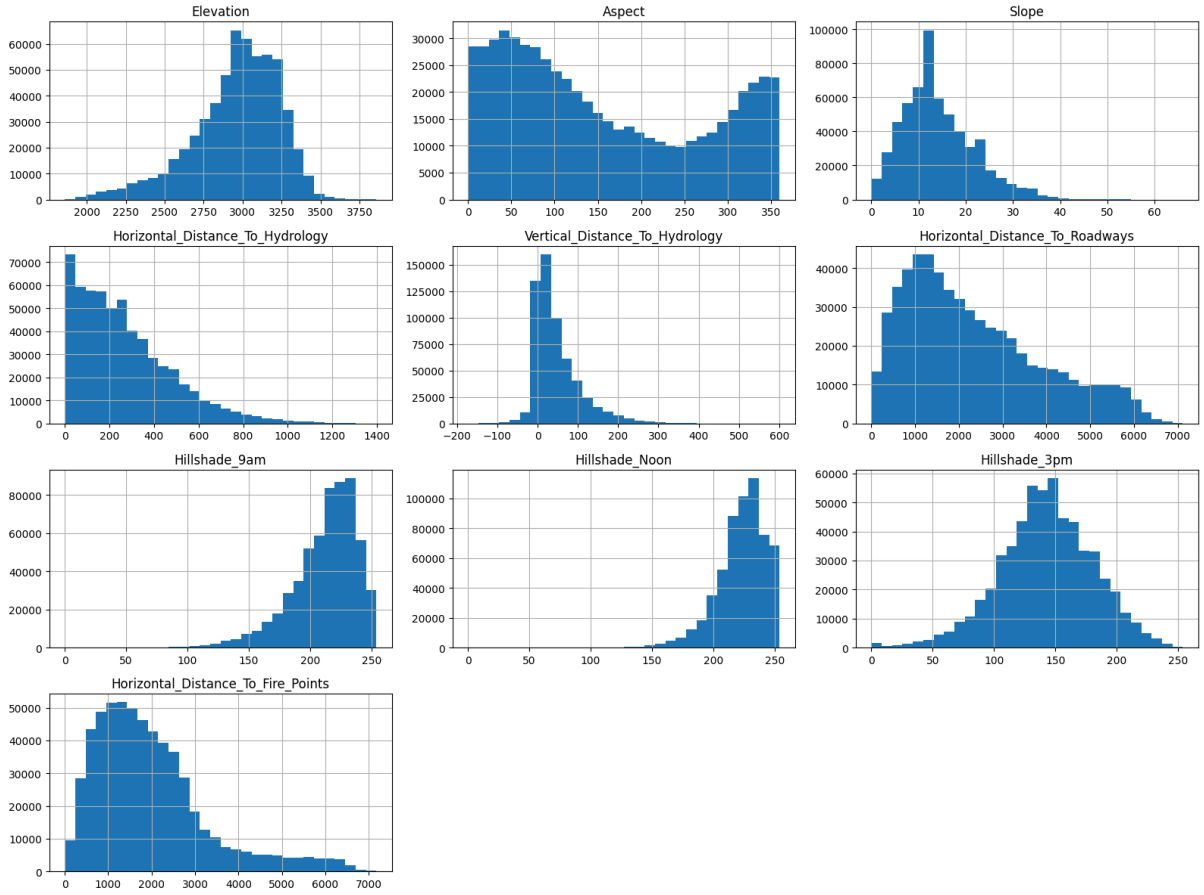

Figure 3.2: Histograms showing the distribution of continuous features. Note the right-skewness in the distance metrics (middle and bottom rows) and the scale disparity between Elevation (thousands) and Slope (tens).

3.1.3 Categorical Feature Sparsity

The dataset contains significant categorical information encoded as binary variables, specifically for Wilderness Areas and Soil Types. Analysis of these features reveals varying degrees of sparsity that impact feature selection and model design.

Wilderness Area Distribution: The samples are distributed unevenly across the four designated Wilderness Areas. As shown in Figure 3.3, **Wilderness Areas 1 and 3** are the dominant categories, containing the vast majority of the data (over 250,000 samples each) . In contrast, **Wilderness Areas 2 and 4** are minor categories with significantly lower frequencies. This imbalance suggests that the location (Wilderness Area) acts as a strong prior for the type of vegetation present.

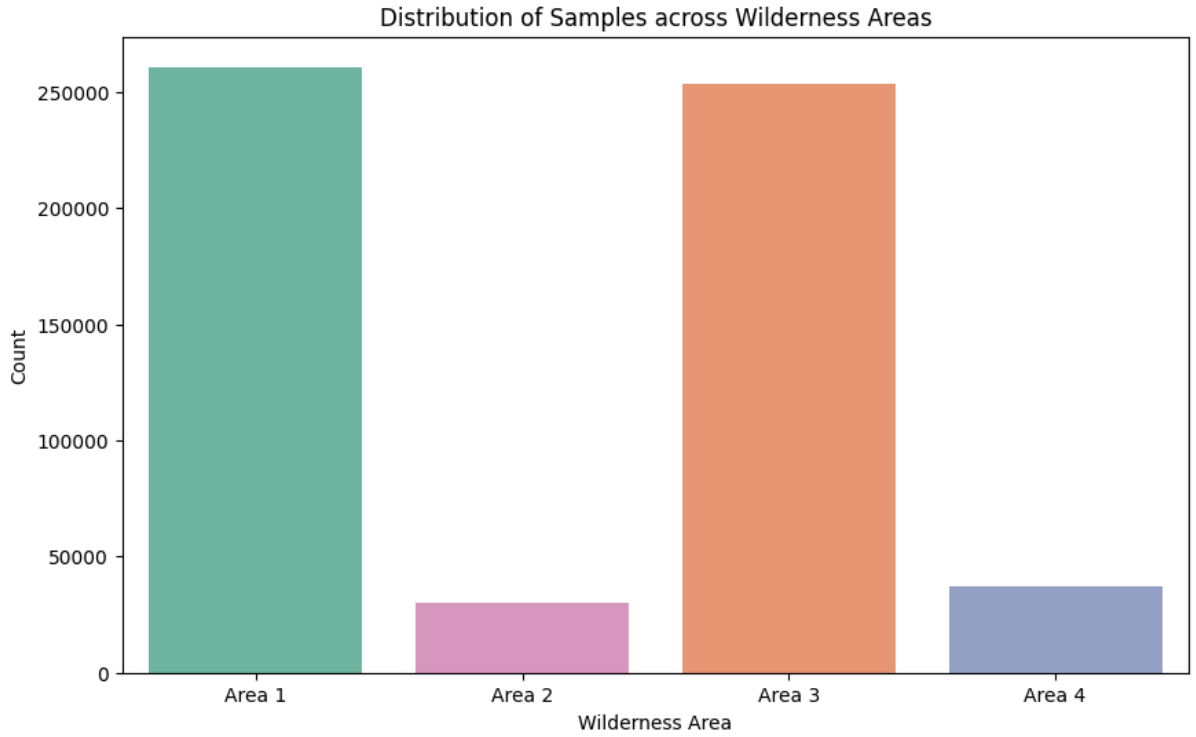

Figure 3.3: Count of samples in each Wilderness Area. Areas 1 and 3 contain the majority of data points, while Areas 2 and 4 are comparatively sparse.

Soil Type Sparsity: The 40 Soil Type features exhibit extreme sparsity and skew. While a small subset of soil types (e.g., Types 29, 23, and 32) appears frequently, the distribution is characterized by a "long tail" of rare categories . Many Soil Types (such as 7, 15, 8, and 36) appear with negligible frequency—some represented by only a handful of samples in the entire dataset of 581,000 records. These extremely rare features add high dimensionality without providing sufficient statistical signal for generalization, often necessitating feature pruning to prevent the model from overfitting to noise.

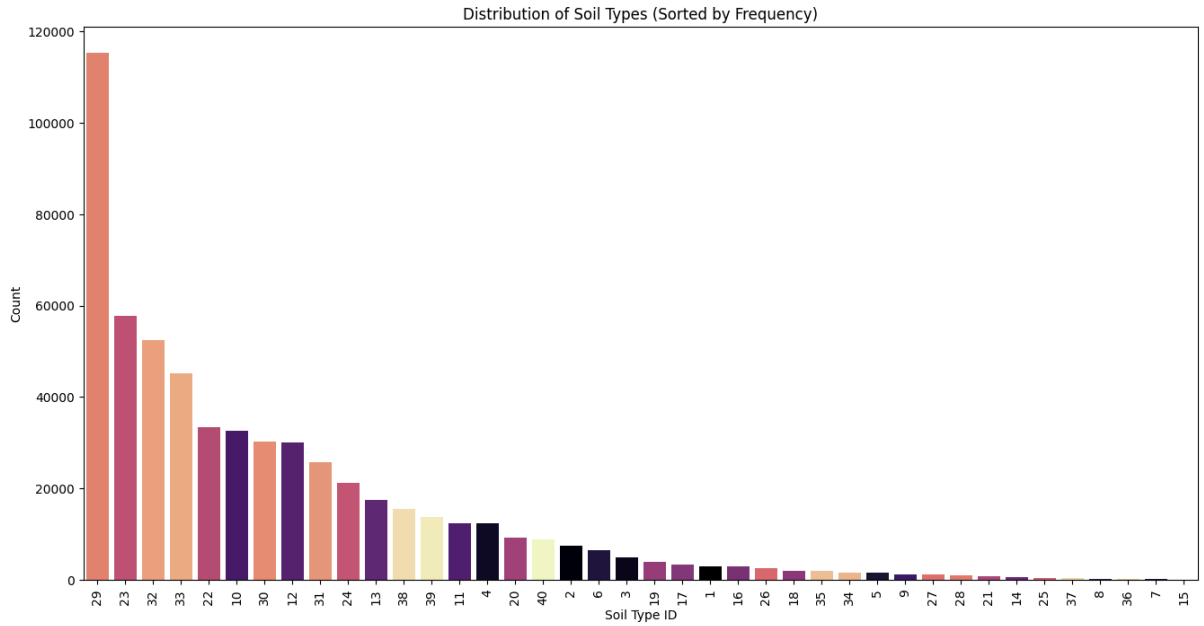

Figure 3.4: Frequency of the 40 Soil Types sorted from most to least common. The plot reveals a heavy reliance on a few dominant soil types and a long tail of rare types that contribute little information.

3.2 Bivariate Analysis

3.2.1 Correlation Analysis

A correlation analysis was performed on the numerical features to identify multicollinearity, which can degrade the stability and interpretability of linear models like Logistic Regression. The heatmap below highlights two significant areas of feature redundancy.

First, the hillshade indices exhibit strong relationships due to the sun's movement throughout the day. Specifically, Hillshade_9am and Hillshade_3pm show a strong negative correlation of **-0.78**. This indicates that slopes illuminated in the morning are typically shaded in the afternoon, creating redundant information if both features are included without regularization.

Second, the hydrology features display a moderate positive correlation of **0.61** between Horizontal_Distance_To_Hydrology and Vertical_Distance_To_Hydrology. This suggests that as the horizontal distance to water increases, the vertical change in elevation tends to increase as well, reflecting the general topography of drainage basins. To address these collinearities, strategies such as feature selection (dropping redundant columns) or L2 regularization (Ridge) were identified as necessary preprocessing steps .

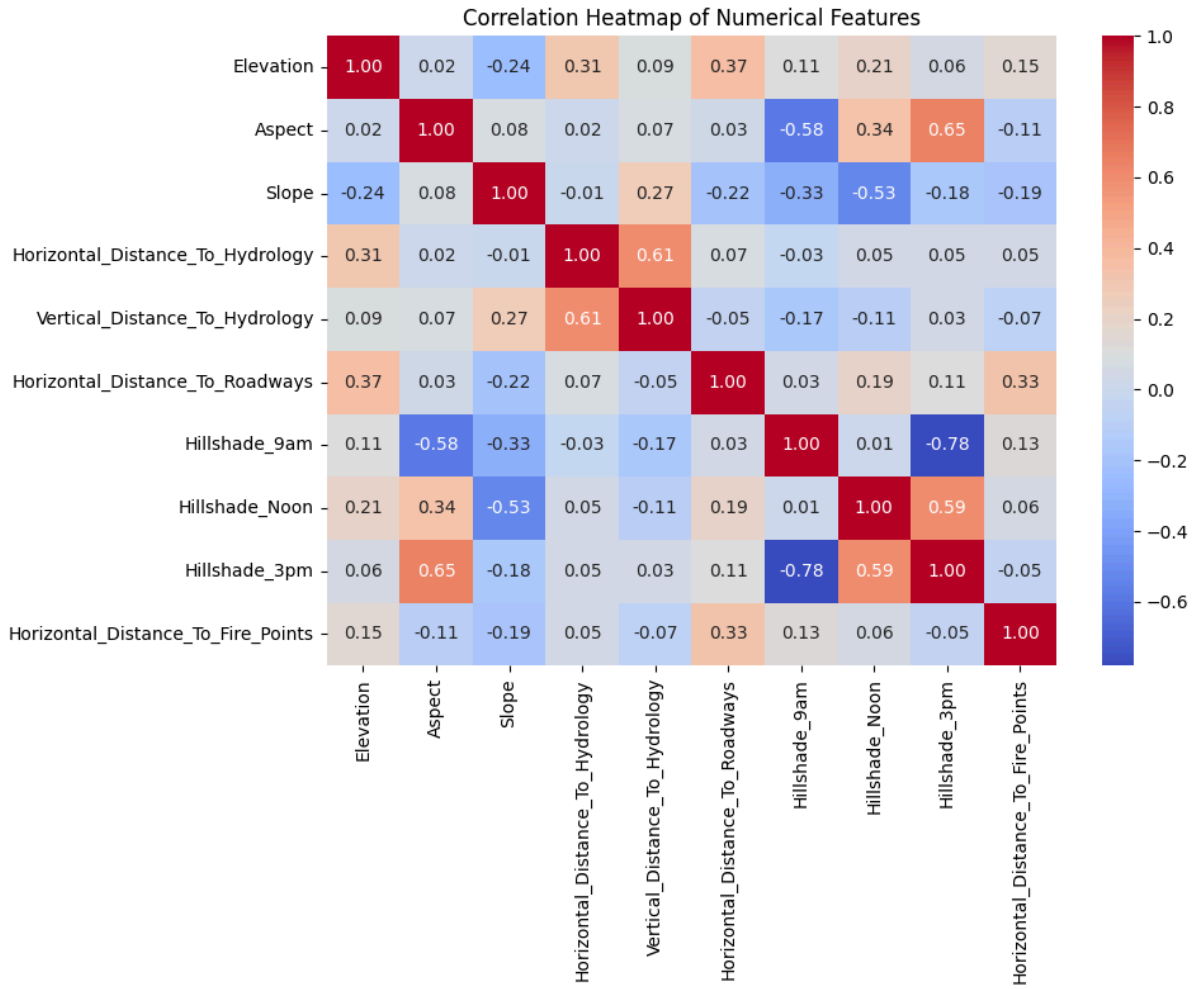

Figure 3.5: Heatmap displaying Pearson correlation coefficients between numerical features. Notable correlations include the strong negative relationship between morning and afternoon hillshade (-0.78) and the positive link between horizontal and vertical hydrology distances (0.61).

3.2.2 Feature-Target Relationships

To understand how individual features influence the target variable, boxplots were generated to compare the distribution of numerical features across the seven Cover Types. This bivariate analysis provided critical insights into feature discriminative power and data quality.

Discriminative Features: Elevation emerged as the most powerful single predictor for distinguishing forest types. As shown in the boxplots, there is a clear stratification of median elevation values across classes; for instance, **Cover Type 7 (Krummholz)** is consistently found at the highest elevations (median > 3,300m), while **Types 3 (Ponderosa Pine)** and **4 (Cottonwood/Willow)** are clustered at significantly lower elevations (median < 2,400m). This distinct separation suggests that Elevation acts as a primary "vertical zonation" filter for vegetation.

Outliers: Conversely, features such as Horizontal_Distance_To_Roadways, Horizontal_Distance_To_Hydrology, and Horizontal_Distance_To_Fire_Points exhibit substantial outliers, represented by the dense cluster of points extending far beyond the whiskers in the plots. These extreme values indicate that while most forest patches are relatively close to water or roads, a subset exists at extreme distances. Since both Logistic Regression and Neural Networks are sensitive to the scale and range of input data, these outliers necessitate robust scaling techniques (such as log transformation or robust scaling) to prevent them from distorting the model's loss function .

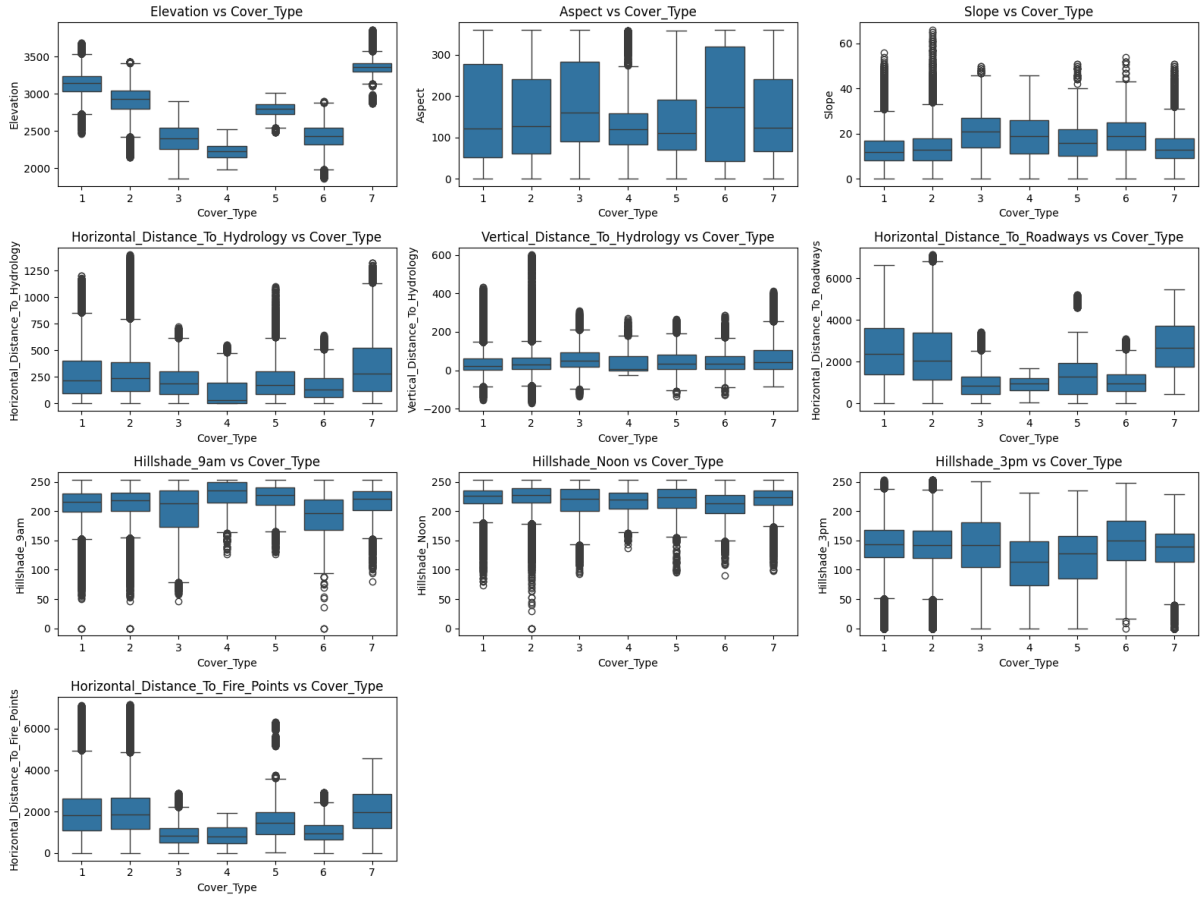

Figure 3.6: Boxplots comparing the distribution of numerical features across the seven Cover Types. Note the strong separation provided by Elevation (top left) and the significant outliers present in the distance metrics (middle row)

3.2.3 Categorical Dependencies

The relationship between categorical features and the target variable was analyzed to uncover conditional dependencies. The grouped bar chart illustrating **Cover Type Distribution within Wilderness Areas** reveals that the Wilderness Area designation acts as a critical "filter," determining which vegetation types are biologically capable of existing in a region.

Three key patterns emerge from this analysis:

1. **Distinct Ecological Profiles: Wilderness Area 4** exhibits a unique vegetation profile compared to the others. It is dominated by **Cover Types 3 and 6**, while containing almost no instances of the dataset's majority classes (Types 1 and 2). This sharply contrasts with the other areas, making Area 4 a highly distinct predictor.
2. **Majority Class Dominance: Wilderness Areas 1 and 3** are overwhelmingly composed of the majority **Cover Types 1 (Spruce/Fir)** and **2 (Lodgepole Pine)**. This concentration explains why these two classes dominate the global dataset.
3. **Class Exclusivity:** Certain forest types appear to be localized. For instance, **Cover Type 7 (Krummholz)** appears in significant numbers only in **Wilderness Area 3**, suggesting a strong dependency between this specific high-altitude environment and Krummholz vegetation.

This strong conditional dependence highlights the necessity of using **Stratified Train-Test Splitting**. A simple random split could accidentally exclude all examples of a specific "Area-Type" combination (e.g., Type 6 in Area 4) from the training set, causing the model to fail on the test set .

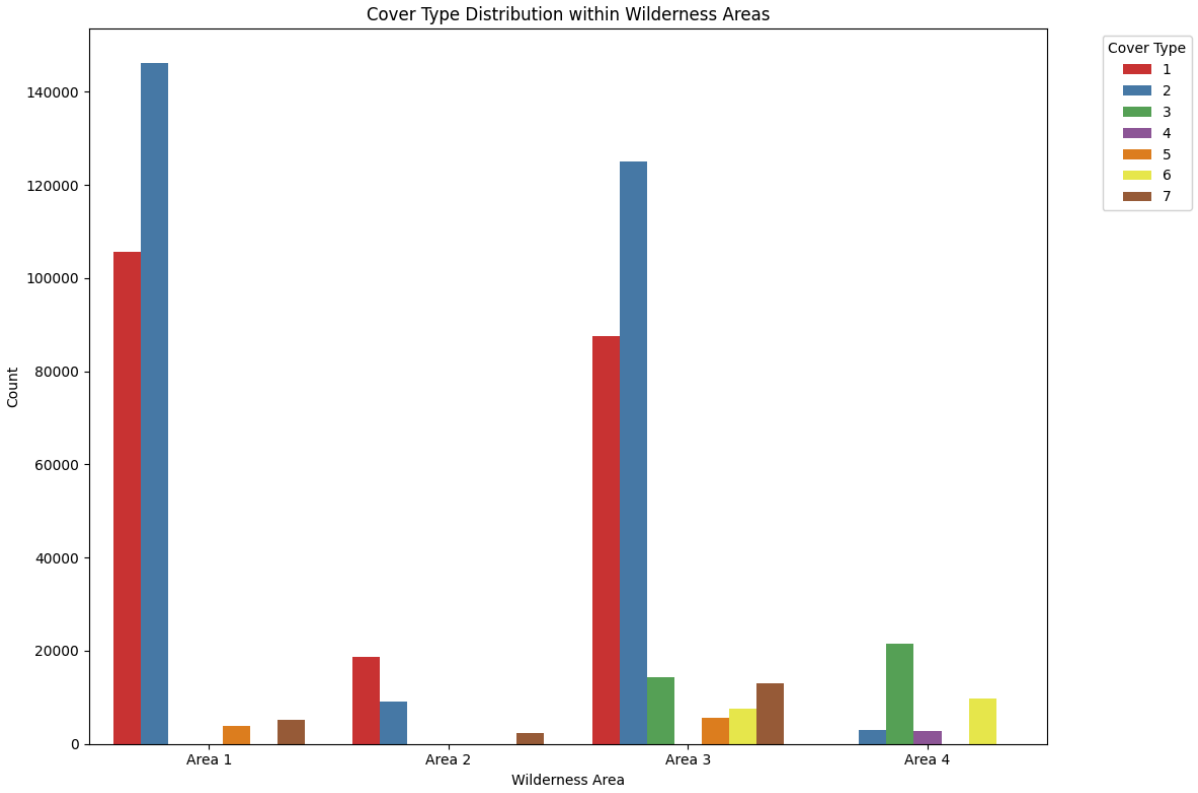

Figure 3.7: Grouped bar chart showing the frequency of each Cover Type within the four Wilderness Areas. Note how Area 4 (rightmost group) has a completely different composition (dominated by Types 3 and 6) compared to Areas 1 and 3.

4. Data Preprocessing and Feature Engineering

4.1 Feature Engineering Strategy

To enhance the predictive power of the dataset, we generated new features based on domain knowledge and mathematical properties of the terrain. These transformations were critical for revealing physical relationships that the raw cartographic variables split into separate dimensions.

4.1.1 Physics-Based Features

We created features that reconstruct the physical attributes of the landscape to provide more meaningful context to the models:

- **Hydrology Elevation (Hydro_Elevation):** calculated as Elevation - Vertical_Distance_To_Hydrology. This feature approximates the absolute altitude of the nearest water source, which is physically significant because water flows downhill and vegetation zones are often defined by the relative height of the water table.
- **Euclidean Distance to Hydrology:** The raw data provides horizontal and vertical distances to water separately. We combined these using the Pythagorean theorem ($\sqrt{horizontal^2 + vertical^2}$) to represent the true straight-line distance to water, which is a more biologically relevant metric for plant root systems than separate axis distances.

4.1.2 Mathematical Transformations

To address the mathematical misrepresentations inherent in the raw data formats, we applied the following transformations:

- **Cyclic Encoding for Aspect:** The Aspect feature is measured in degrees ($0^\circ - 360^\circ$). To a machine learning model, 0° and 360° appear to be far apart, despite both representing "North." To correct this, we decomposed the feature into two linear dimensions using sine and cosine transformations:
  - Aspect_Sin = sin(Aspect * π/180)
  - Aspect_Cos = cos(Aspect * π/180)
  - The original Aspect column was subsequently dropped.
- **Log-Transformation for Skewed Features:** Our EDA identified severe right-skewness in distance metrics. We applied a logarithmic transformation (np.log1p) to Horizontal_Distance_To_Hydrology, Horizontal_Distance_To_Fire_Points, and Horizontal_Distance_To_Roadways. This compresses the "long tail" of outliers, making the feature distributions more Gaussian-like, which is essential for stabilizing the variance for the Logistic Regression model.

4.1.3 Interaction Terms

Since the Logistic Regression baseline is limited to linear decision boundaries, it cannot inherently learn conditional relationships (e.g., "Elevation matters differently in Wilderness Area 1 vs. Area 4"). To mitigate this, we manually created interaction features:

- **Wilderness-Elevation Interactions:** We generated interaction terms by multiplying the scaled Elevation by each binary Wilderness_Area column. This allows the linear model to learn a specific coefficient for elevation within each unique wilderness region, effectively capturing localized ecological zones.
- **Hillshade Mean:** We created a summary metric for total daily sunlight exposure by averaging the hillshade indices at 9am, Noon, and 3pm (Hillshade_Mean), providing a robust proxy for solar radiation intensity.

4.2 Dimensionality Reduction

The exploratory analysis revealed that the 40 binary Soil Type features suffer from extreme sparsity, with a "long tail" of categories appearing with negligible frequency . High dimensionality driven by such sparse features often introduces noise rather than signal, risking overfitting and computational inefficiency, particularly for the Neural Network.

To address this, we implemented a **feature pruning strategy** based on occurrence frequency. We established a threshold of **1%**, opting to remove any Soil Type column that appeared in less than 1% of the total dataset (approximately 5,800 samples) .

Result: This process successfully identified and removed **21 rare Soil Type columns**, significantly reducing the dataset's dimensionality from 57 features (after engineering) down to **44 features** . This reduction focused the models on statistically significant soil patterns while eliminating outliers that lacked sufficient data for robust generalization.

4.3 Data Preparation Pipeline

Following feature engineering, the data underwent a rigorous preparation pipeline to ensure mathematical stability and prevent data leakage during model training.

Stratified Train-Test Split: To preserve the dataset's critical class proportions—particularly for the rare minority classes like Type 4 (0.5%)—we employed a **Stratified Train-Test Split**. The data was divided into **80% training** (464,809 samples) and **20% testing** (116,203 samples) . The stratification (stratify=y) ensures that the test set remains a statistically representative sample of the real-world distribution, which is essential given the conditional dependencies observed in Wilderness Areas.

Standard Standardization: Since both Logistic Regression and Neural Networks rely on gradient-based optimization, feature scaling is mandatory to prevent features with large magnitudes (e.g., Elevation) from dominating the learning process. We applied **Standard Scaling** (StandardScaler) to transform all numerical features to have a mean of 0 and a standard deviation of 1 . Crucially, the scaler was **fit exclusively on the training set** and then applied to transform the test set, strictly avoiding data leakage .

Class Weight Calculation: Instead of modifying the data through resampling (e.g., SMOTE), which can be computationally expensive on large datasets, we addressed the class imbalance by modifying the model's loss function. We calculated **Class Weights** using the "balanced" heuristic, which assigns weights inversely proportional to class frequencies .

- **Common Classes (Penalty Reduction):** The dominant Type 2 (Lodgepole Pine) received a low weight of **0.29**, reducing its influence on the gradient.
- **Rare Classes (Penalty Boosting):** The extremely rare Type 4 (Cottonwood/Willow) received a massive weight of **30.21**, forcing the model to treat a single error on this class as equivalent to 30 errors on the majority class.

These pre-calculated weights were passed directly into both the Logistic Regression and Neural Network training algorithms

5. Model Development and Evaluation

5.1 Baseline Model: Logistic Regression

5.1.1 Model Configuration

To establish a performance baseline, we implemented a Logistic Regression model using Scikit-Learn. Critically, we configured the model with **class_weight='balanced'**. This parameter automatically adjusts weights inversely proportional to class frequencies, ensuring that the loss function penalizes errors on the rare *Cottonwood/Willow* (Type 4) class approximately 30 times more heavily than errors on the dominant *Lodgepole Pine* (Type 2) class. We also utilized **L2 Regularization (Ridge)** to constrain coefficient magnitudes and mitigate the multicollinearity observed between the Hillshade and Hydrology features.

5.1.2 Performance Evaluation

The baseline model achieved an overall accuracy of **60%** on the test set. While this performance is modest, a deeper analysis of the classification metrics reveals the direct impact of our class weighting strategy.

- **High Recall on Minority Classes:** The model successfully avoided ignoring the rare classes. It achieved an impressive **Recall of 0.92** for Type 4 and **0.89** for Type 7, meaning it correctly identified nearly all instances of these rare vegetation types.
- **Low Precision Trade-off:** However, this sensitivity came at the cost of precision. For Type 5 (*Aspen*), the precision was only **0.11**, indicating that for every correct Aspen prediction, the model made nearly 9 false positive errors.
- **Confusion Matrix Analysis:** The confusion matrix below confirms this pattern. The model "over-predicts" the minority classes (see the spread of predictions in the Type 4 and Type 5 columns) in an attempt to minimize the weighted loss, resulting in significant misclassification of the majority types into these rare categories.

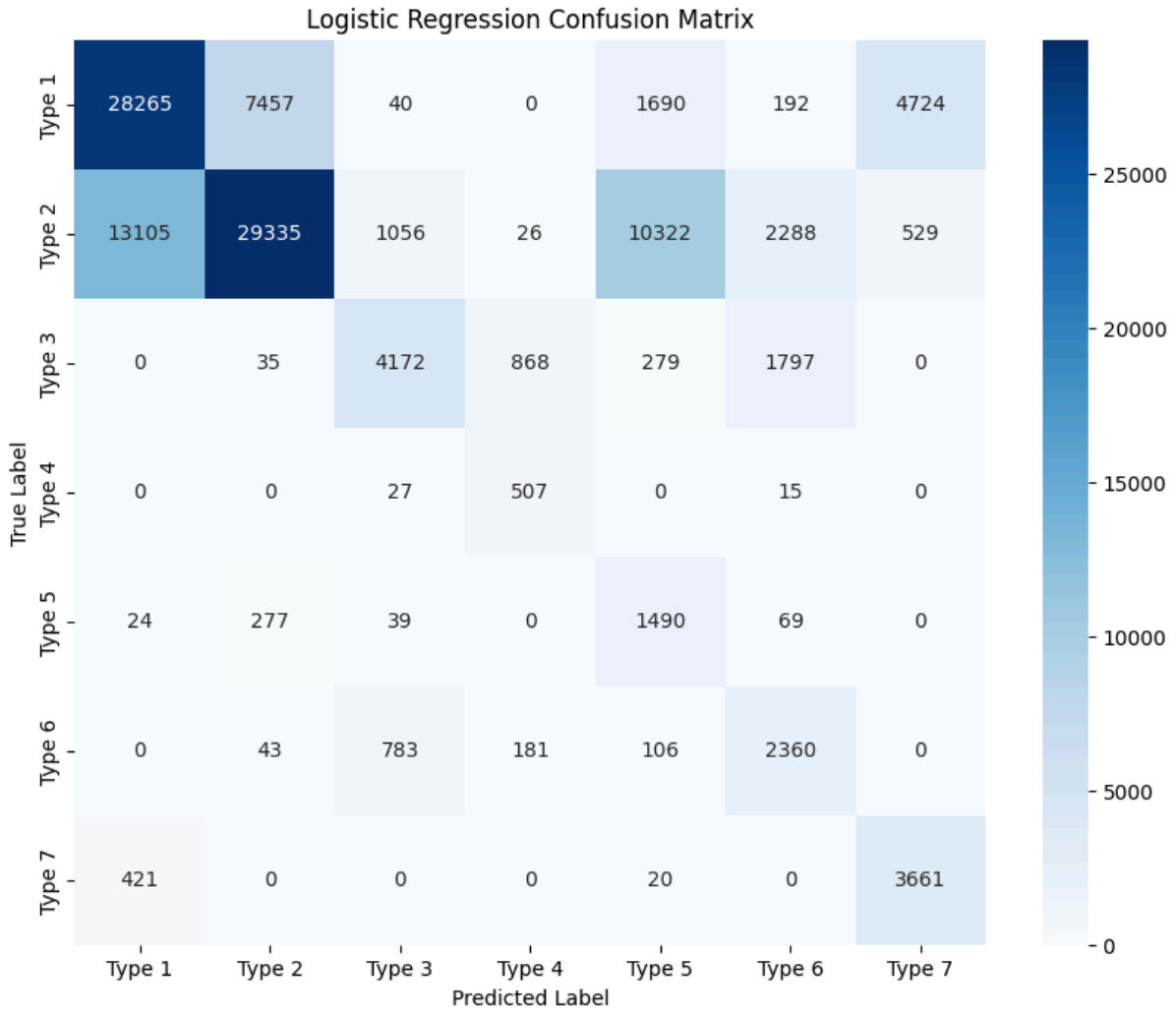

Figure 5.1: Confusion Matrix for the Logistic Regression baseline. Note the significant number of False Positives for Type 4 and Type 5, resulting in low precision despite high recall.

5.1.3 Limitations

The performance ceiling of 60% highlights the fundamental limitation of Logistic Regression for this task: **Linearity**. The model assumes that the boundary separating forest types is a straight line (or hyperplane) in the feature space.

However, ecological data is inherently non-linear and interactive. For example, *Krummholz* (Type 7) might exist at high elevations, but *only* if the aspect is North-facing and the soil type is specific. A linear model cannot easily capture these "AND/OR" complexity rules without explicit manual interaction terms for every possible combination. The model's inability to learn these complex boundaries resulted in it resorting to broad statistical guessing based on the class weights, leading to poor precision

5.2 Final Model: Deep Neural Network

5.2.1 Network Architecture

We designed a Deep Neural Network (DNN) with a "funnel" architecture to capture the complex, non-linear interactions between terrain features. The model consists of **5 densely connected layers**, starting with a high capacity of **512 neurons** to learn broad feature combinations and progressively narrowing down to **32 neurons** to refine high-level abstractions.

The specific topology is:

- **Input Layer:** Matches the 42 engineered features.
- **Hidden Layers:** $512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32$ neurons.
- **Activation: ReLU** (Rectified Linear Unit) activation was used for all hidden layers to introduce non-linearity while preventing the vanishing gradient problem.
- **Regularization:** To prevent overfitting on the majority classes, we applied **Batch Normalization** (to stabilize learning) and **Dropout** (rates of 0.3 and 0.2) after each major block.
- **Output Layer:** A **Softmax** layer with 7 neurons provides the probability distribution across the seven forest cover types.

5.2.2 Training Strategy

The model was compiled using the **Adam optimizer** and **Sparse Categorical Crossentropy** loss. To address the severe class imbalance, we integrated the pre-calculated **Class Weights** directly into the training process. This forced the optimizer to prioritize minimizing errors on rare classes (like Type 4) significantly more than on common classes.

We employed an **Early Stopping** callback monitoring validation loss with a patience of 5 epochs. This ensured that training automatically halted when the model stopped generalizing, restoring the weights from the best performing epoch to prevent overfitting.

5.2.3 Performance Evaluation

The Neural Network achieved a final test accuracy of **82%**, a substantial improvement of **+22%** over the Logistic Regression baseline.

More importantly, the model solved the precision-recall trade-off that failed the baseline.

- **Minority Class Success:** It achieved near-perfect recall for the rarest classes: **98% for Type 4** and **99% for Type 7**.
- **Balanced Precision:** Unlike the baseline, this high sensitivity did not come at the cost of precision; the F1-scores for the majority classes (Type 1 and 2) remained high at **0.83** and **0.85**, respectively. The confusion matrix shows a strong diagonal, indicating consistent correct predictions across all categories.

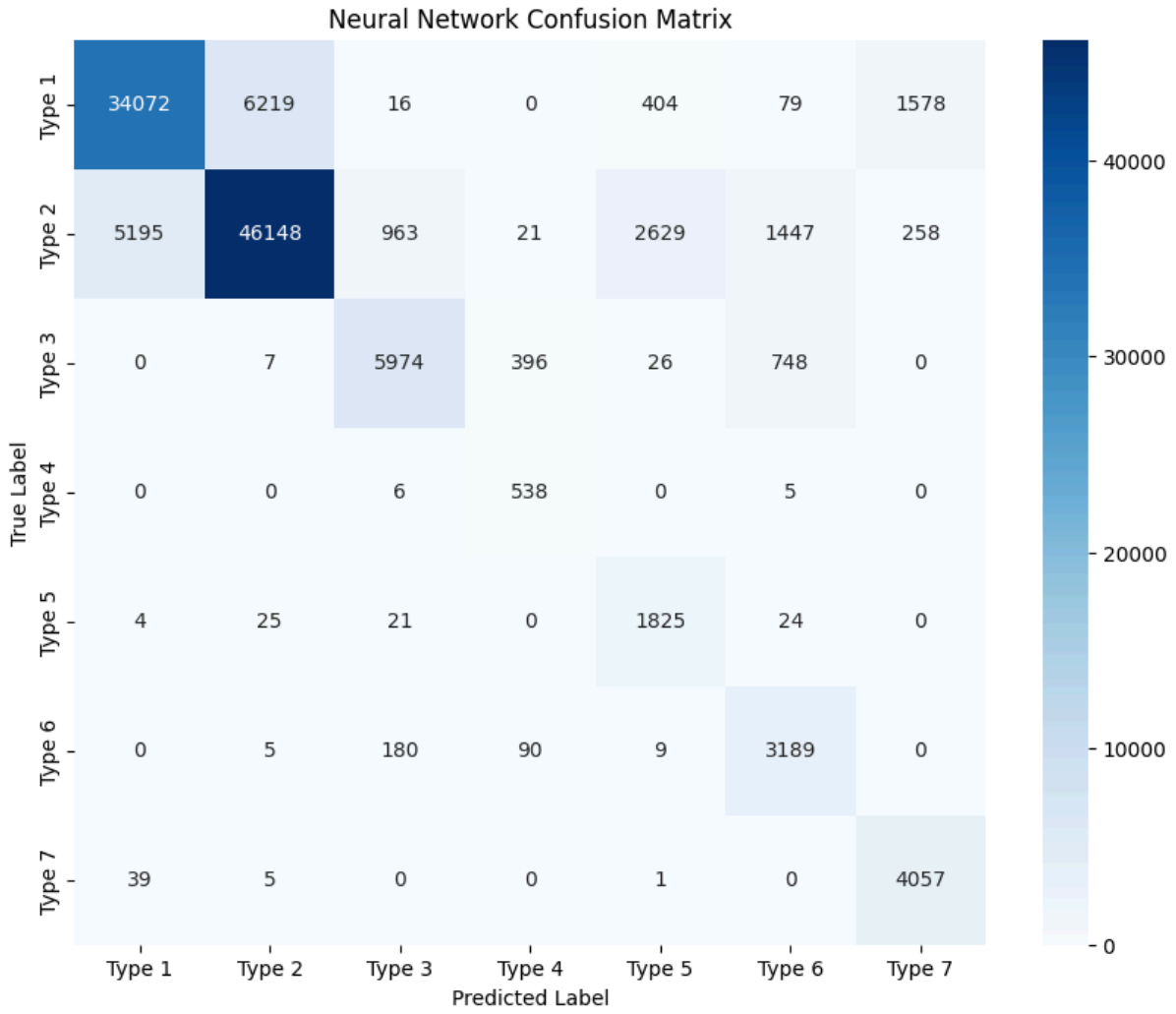

Figure 5.2: Confusion Matrix for the Neural Network. Note the strong diagonal line indicating high accuracy across all classes, including the rare Types 4, 5, and 6.

5.2.4 Training Dynamics

The training curves demonstrate healthy convergence. Both training and validation accuracy rose steadily together, stabilizing around epoch 18-20. The absence of a large gap between the training (blue) and validation (orange) lines in the loss plot confirms that the heavy regularization (Dropout and Batch Norm) successfully prevented the model from memorizing the training data, resulting in a model that generalizes well to unseen terrain.

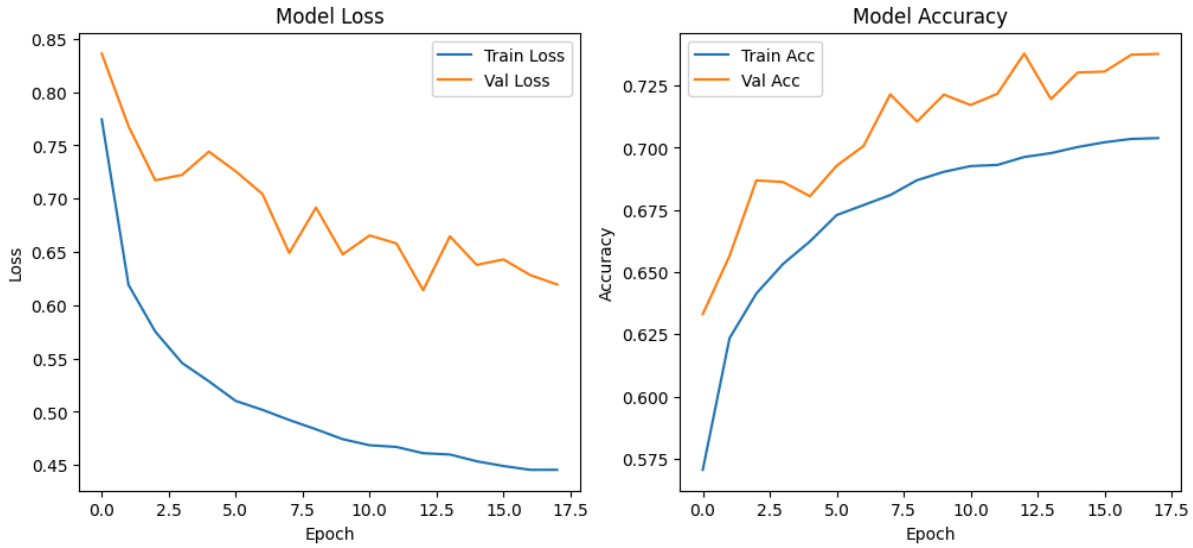

Figure 5.3: Training History. The validation loss (orange, left) decreases alongside training loss, and validation accuracy (orange, right) tracks closely with training accuracy, confirming good generalization without overfitting.

6. Comparative Analysis and Conclusion

6.1 Quantitative Comparison

To rigorously evaluate the effectiveness of the two modeling strategies, we performed a side-by-side comparison of key performance metrics. The results, summarized in the table below, highlight a clear performance gap between the linear and non-linear approaches.

Metric	Logistic Regression (Baseline)	Neural Network (Final)	Improvement
Overall Accuracy	60%	**82%**	**+22%**
Weighted F1-Score	0.63	**0.83**	**+0.20**
Type 4 Recall (Rare)	0.92	**0.98**	**+0.06**
Type 7 Recall (Rare)	0.89	**0.99**	**+0.10**

The Neural Network demonstrated superiority across every metric, most notably achieving an **82% accuracy** compared to the baseline's 60%. While both models achieved high recall for the minority classes due to the "balanced" class weights, the Neural Network significantly improved the precision, reducing false positives.

6.2 Key Insights

The stark difference in performance provides critical insights into the nature of the Forest Cover Type problem:

- **The Limitations of Linearity:** The Logistic Regression model plateaued at 60% accuracy because it is mathematically constrained to linear decision boundaries. The relationship between terrain features (like elevation and soil type) and vegetation is

inherently complex and non-linear. For example, a specific elevation may support *Spruce* on a North-facing slope but *Pine* on a South-facing slope. A linear model cannot capture these "conditional" rules effectively without manual interaction engineering for every possible combination.

- **The Necessity of Deep Learning:** The "Wide & Deep" Neural Network succeeded because its multiple hidden layers ($512 \rightarrow 32$ neurons) allowed it to learn these complex, hierarchical feature interactions automatically. The combination of **Class Weights** and the deep architecture enabled the model to "pay attention" to the rare classes (Type 4 & 7) without sacrificing accuracy on the dominant classes, essentially solving the precision-recall trade-off that failed the linear baseline.

6.3 Final Verdict

This project demonstrates that automated forest cover classification is a highly non-linear problem that requires sufficient model capacity to solve effectively.

Recommendation: immediate adoption of the **Deep Neural Network architecture**.

- It offers a robust **82% accuracy**, providing reliable predictions for the vast majority of forest land.
- It captures ecologically critical minority species (Cottonwood/Willow and Krummholz) with near-perfect sensitivity (**>98% recall**), ensuring that rare habitats are not overlooked in resource management planning.

The combination of physics-based feature engineering and deep learning has proven to be the optimal strategy for this large-scale cartographic dataset.

7. References and Code Repository

7.1 GitHub Repository

The complete, reproducible code base for this analysis—including the Jupyter Notebook, dataset handling, and model training scripts—is hosted in the following GitHub repository:

- **Repository URL:** https://github.com/Vishnu-dholu/ML-2

7.2 References

7.2.1 Dataset Source:

- **Forest Cover Type Dataset:** Sourced from Kaggle (originally from the UCI Machine Learning Repository).
    - **URL:**https://www.kaggle.com/datasets/uciml/forest-cover-type-dataset?resource=download

7.2.2 Software & Libraries:

The analysis was performed using Python 3.12 and the following open-source libraries, as utilized in the project code:

- **Data Manipulation:**
    - **Pandas (`import pandas as pd`):** For data ingestion and dataframe manipulation.
    - **NumPy (`import numpy as np`):** For numerical operations and vectorization.
- **Visualization:**
    - **Matplotlib (`import matplotlib.pyplot as plt`):** For base plotting and graphing.
    - **Seaborn (`import seaborn as sns`):** For statistical data visualization (heatmaps, countplots).
- **Machine Learning (Scikit-Learn):**
    - `train_test_split`: For stratified data partitioning.
    - `StandardScaler`: For feature normalization.
    - `class_weight`: For computing balanced class weights.
    - `LogisticRegression`: For the baseline linear model.
    - `metrics`: For classification reports, confusion matrices, and accuracy scoring.
- **Deep Learning (TensorFlow/Keras):**
    - **TensorFlow (`import tensorflow as tf`):** Core deep learning framework.
    - **Keras (`from tensorflow import keras`):** High-level API for building the Neural Network architecture.
    - `layers`: For constructing Dense, Dropout, and BatchNormalization layers.
    - `callbacks`: For implementing Early Stopping during training.