Université Jean Monnet, Saint-Etienne

M1 MLDM

Machine Learning: Fundamentals and Algorithms

Classification of Forest Cover Types using Support Vector Machines

By:

	Part 1 (Practice)	**Part 2 (Go wild!)**
Ayomide Abayomi-Alli	32%	32%
Mahima Haridasan Sumathy	32%	32%
Sinjini Ghosh	32%	32%
generativeAI (ChatGPT)	4%	4%
total =	**100%**	**100%**

March 17, 2025

Abstract

Forest cover classification is crucial in environmental management and land use planning. In this study, we classify the Cover Type dataset from Roosevelt National Park, North Colorado, using Support Vector Machines (SVM). The dataset consists of 581012 instances and 54 features, including topographical, hydrological, and soil-related attributes. We explore the dataset, preprocess the features, and train an SVM classifier to predict the type of forest cover. We analyze feature correlations, class distributions, and the impact of class imbalance. The results demonstrate the effectiveness of SVM in accurately predicting forest cover types.

1 Introduction

The Roosevelt National park is located in northern Colorado and has a total area of 813,799 acres (3,293.33 km2). Originally part of the Medicine Bow Forest Reserve, the Roosevelt National Forest was established on May 22, 1902. In 1910, it was renamed the Colorado National Forest, and in 1932, it obtained its current name the Roosevelt National Forest, honoring President Theodore Roosevelt. Within Roosevelt National Forest, six officially recognized wilderness areas are part of the National Wilderness Preservation System.

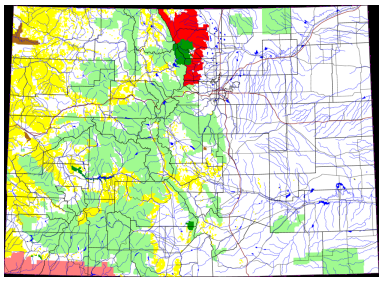

Figure 1: Map of the Roosevelt National Forest area in red, in north-central Colorado

2 Dataset Exploration

The dataset contains cartographic features of four wilderness areas located in Roosevelt National Park, North Colorado, United States. These areas represent forests with minimal human-caused disturbances. Each row in the dataset represents a specific geographical location or plot of land with various environmental and soil characteristics. The dataset has 581012 instances and 54 features, which also includes the one-hot encoded columns for soil type and wilderness area.

Name	Data Type	Measurement Unit	Description
Elevation	Integer	Metre	Elevation of the area in metres
Aspect	Integer	Degrees (0 to 360)	Direction a slope is facing
Slope	Integer	Degree	Slope in degrees
Horizontal Distance to Hydrology	Integer	Metre	Horizontal distance to the nearest surface water
Vertical Distance to Hydrology	Integer	Metre	Vertical distance to the nearest surface water
Horizontal Distance to Roadways	Integer	Metre	Horizontal distance to the nearest roadway
Hillshade 9AM	Integer	Index (0 to 255)	Hillshade index at 9 AM indicating the amount of sunlight received
Hillshade Noon	Integer	Index (0 to 255)	Hillshade index at Noon indicating the amount of sunlight received
Hillshade 3PM	Integer	Index (0 to 255)	Hillshade index at 3 PM indicating the amount of sunlight received
Horizontal Distance to Fire Points	Integer	Metre	Horizontal distance to wildfire ignition points
Wilderness Area (4 binary columns)	Categorical	Binary (0 or 1)	Wilderness area designation
Soil Type (40 binary columns)	Categorical	Binary (0 or 1)	Soil type designation

Table 1: Description of dataset features

There are seven forest cover types, label-encoded with numbers from 1 to 7.

Class imbalance is prevalent in the target variable. There are significantly more instances of Lodgepole Pine and Spruce/Fir type than the other types of vegetation, as illustrated in the Bar graph.

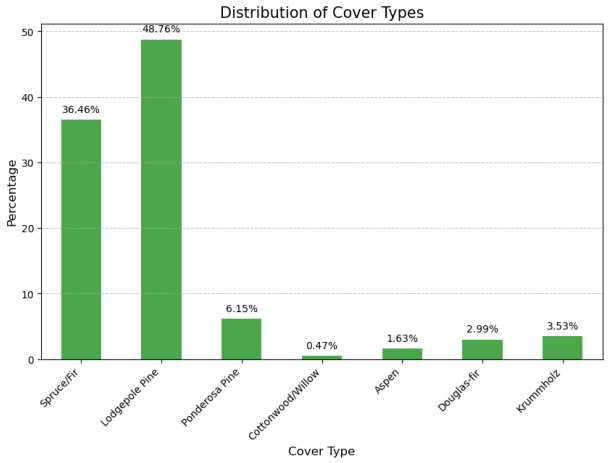

Figure 2: Number of Observations in Each Cover Type Category

There is an imbalance in the categorical features (wilderness area and soil type). Wilderness areas Rawah and Comanche Peak seem to dominate over the other two types. Similarly, soil type 29 significantly dominates over the other soil types.

The dataset's continuous features exhibit varying degrees of skewness, with some being left-skewed (e.g., Elevation, Hillshade 9am, Hillshade Noon) and others right-skewed (e.g., Horizontal and Vertical Distance to Hydrology, Horizontal Distance to Fire Points), suggesting that most locations are clustered near roads, water bodies, or fire points, while some are much farther away. Features like Slope and Aspect show moderate skewness, indicating a concentration of lower values with occasional high-value outliers. To address these distribution imbalances, scaling is necessary.

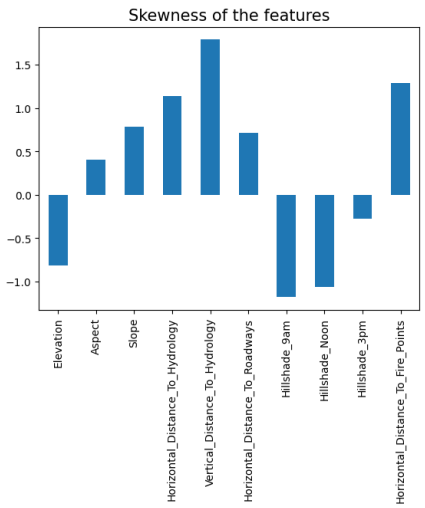

Figure 3: Display of Skewness in certain features

There is significant negative correlation between Hillshade index at 9 AM and Hillshade index at 3 PM, which is expected since sunlight exposure in the morning is inversely related to that in the afternoon. There is a considerable positive correlation between Vertical distance to the nearest hydrology and Horizontal distance to the nearest hydrology, meaning areas that are horizontally farther from water bodies also tend to be vertically farther. We also observe that hillshade index at noon and slope are inversely proportional indicating that steeper slopes receive less sunlight at noon.

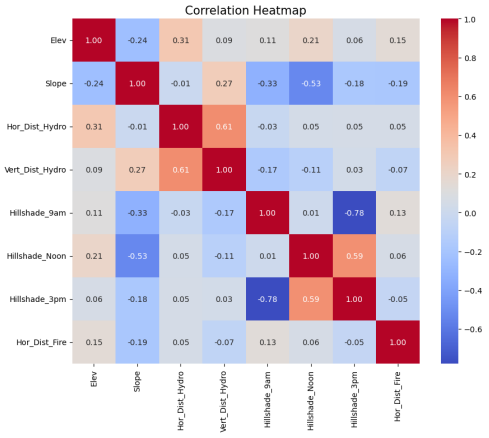

Figure 4: Correlation Matrix

3 Data Pre-Processing

Several Pre-processing steps were applied to enhance the performance of the model

3.1 Missing Values and Duplicate values

No missing values or duplicate values were detected in the dataset.

3.2 Feature Scaling

Numerical features in the dataset vary significantly in their magnitude. To ensure uniformity and prevent certain features from dominating the model due to their scale, we applied **Min-Max Scaling**, which normalizes the data within a fixed range of $[0, 1]$. This transformation helps improve the stability and efficiency of SVM.

Mathematically, Min-Max Scaling is defined as:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where:

- X is the original feature value,
- $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the feature, respectively,
- X' is the transformed value after scaling.

This ensures that all numerical features lie within the $[0, 1]$ range, preventing the features with large magnitudes from introducing bias in the model.

3.3 Splitting into Train and Test sets

The dataset is divided into 70% train set and 30% test split.

3.4 Class Imbalance

Class imbalance is addressed using two different techniques: SMOTE-NC (Synthetic Minority Oversampling technique for Nominal and Continuous) and Random Undersampling.

3.5 Dimensionality reduction

Dimensionality reduction was performed using two different methods. We trained the SVM classifier both on the data whose dimensionality was reduced by Feature selection and the data whose dimensionality was reduced by PCA. The methods are described below:

- Feature Selection: We trained an XGBClassifier model on the training data and evaluated its performance on the test set using the F1-score (macro average). Then, we performed

feature selection by iterating through the feature importances of the model, selecting features at different thresholds, and re-training the model at each threshold. We evaluated the performance of each model based on the F1-score and found that the highest F1-score was achieved when selecting 35 features.

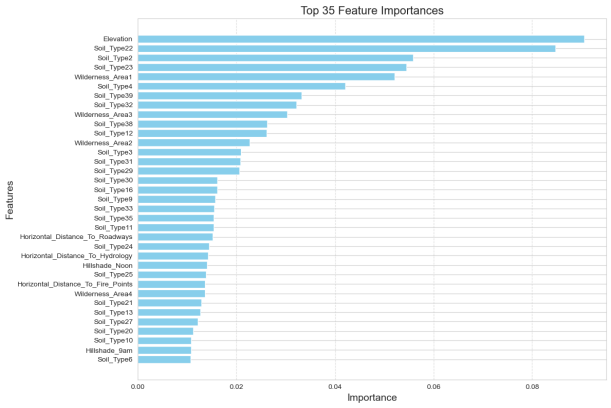

Figure 5: Top 35 Important Features

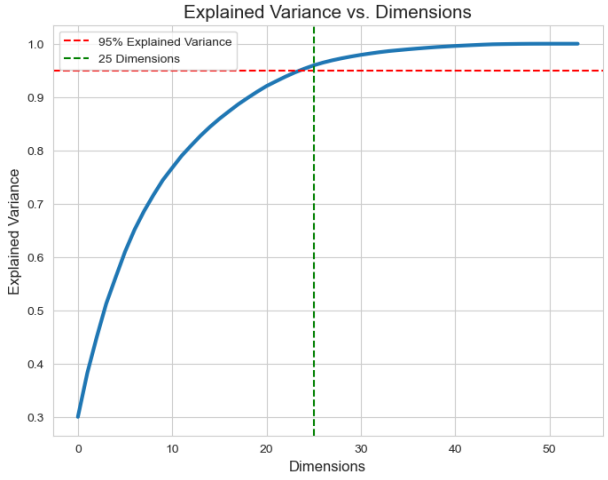

Figure 6: Plot of Explained Variance versus Number of Principal Components

- Principal Component Analysis (PCA): We applied Principal Component Analysis (PCA) to the scaled training data and calculated the cumulative explained variance for each principal component. By analyzing the cumulative variance, we determined that the optimal number of principal components (PCs) required to explain at least 95% of the variance was 25.

4 Support Vector Machine

For the forest cover, we have two different sorts of datasets. We aimed to evaluate SVM's performance on both balanced and unbalanced datasets.

4.1 Hyperparameter tuning

We performed hyperparameter tuning for the **SVM model** using **GridSearchCV**. The hyperparameters optimized were the regularization parameter C, the kernel coefficient γ, and the choice of kernel (`linear` or `rbf`). The parameter grid for the search was as follows:

- C: {0.1, 1, 10, 100}
- γ: {1, 0.1, 0.01, 0.001}
- `kernel`: {`linear`, `rbf`}

The **best combination of hyperparameters** found through the grid search was:

- $C = 100$
- $\gamma = 1$
- `kernel` = `rbf`

These optimal hyperparameters were used for all the 8 variations of the SVM models tested in the study. This ensures that the performance of each model was evaluated under the same set of hyperparameters, allowing for a fair comparison of the different dimensionality reduction and data balancing techniques.

4.2 Model Performance Comparison

In this study, we evaluated several SVM models with different dimensionality reduction techniques and data balancing methods. The models were assessed based on their test accuracy and F1 score, as well as their train accuracy and F1 score. Below are the key findings:

4.2.1 Dimensionality Reduction Method

The models using **Feature Selection** generally outperformed those utilizing **PCA**, particularly in terms of **test F1 score**. Feature Selection provided slightly better results in most cases, leading to higher model performance on the test set.

4.2.2 Impact of Data Balancing

When the data was balanced using **SMOTE-NC** or **Random Undersampling**, the **train accuracy** and **train F1 scores** showed significant improvement. However, this often came at the cost of slightly lower **test accuracy** and **test F1 scores**, indicating a possible overfitting issue with the balanced data techniques. The **unbalanced data** models, particularly those using **Feature Selection**, tended to maintain better overall performance on both test and train data.

4.2.3 Cost Sensitive SVM

Cost-Sensitive SVMs are designed to address the challenges posed by imbalanced datasets, where traditional SVMs may underperform. In standard SVMs, the objective is to find a hyperplane that maximizes the margin between classes while minimizing misclassification errors, treating all errors equally. However, in many real-world scenarios, misclassification costs differ between classes. CS-SVM introduces class-specific cost parameters, C_+ for the positive class and C_- for the negative class, to penalize misclassifications differently. The objective function of CS-SVM is:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C_+ \sum_{i\in\mathcal{I}_+} \xi_i + C_- \sum_{i\in\mathcal{I}_-} \xi_i \qquad (2)$$

subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \qquad (3)$$

Here, $\mathbf{w}$ represents the weight vector, b is the bias term, ξ_i are the slack variables accounting for misclassifications, and C is the regularization parameter controlling the trade-off between maximizing the margin and minimizing classification errors. $\mathcal{I}_+$ and $\mathcal{I}_-$ denote the sets of indices for positive and negative samples, respectively. This allows the model to assign higher penalties to misclassifications of the minority class, improving its sensitivity to that class.

4.2.4 Dimensionality Reduction Comparison

In terms of **dimensionality reduction**, **PCA** showed a slightly lower performance compared to **Feature Selection**, as indicated by the relatively lower test accuracy and F1 scores. This suggests that **Feature Selection** may be more effective in retaining relevant information for classification.

4.2.5 Best Performing Models

The best overall performance was observed with the **Cost Sensitive SVM** using **Feature Selection** on **unbalanced data**, which achieved a high test accuracy of 82.07% and test F1 score of 82.31%. For **balanced data**, the **SVM with SMOTE-NC and Feature Selection** performed well, with a test accuracy of 77.77% and test F1 score of 78.17%. While these results were slightly lower than those with unbalanced data, they showed the benefits of balancing the data to handle class imbalances.

	Model	Dimensionality reduction method	Balanced/Unbalanced data	Test Accuracy	Test F1 Score	Train Accuracy	Train F1 Score:
0	SVM	Feature Selection	Unbalanced	0.7954	0.7897	0.8187	0.8147
1	SVM	Feature Selection	Balanced with SMOTE-NC	0.7777	0.7817	0.9055	0.9043
2	SVM	Feature Selection	Balanced with Undersampling	0.7109	0.7219	0.9381	0.9378
3	Cost Sensitive SVM	Feature Selection	Unbalanced	0.8207	0.8231	0.8670	0.8685
4	SVM	PCA	Unbalanced	0.7970	0.7923	0.8381	0.8350
5	SVM	PCA	Balanced with SMOTE-NC	0.7696	0.7743	0.9108	0.9097
6	SVM	PCA	Balanced with Random Undersampling	0.7012	0.7114	0.9402	0.9400
7	Cost Sensitve SVM	PCA	Unbalanced	0.8204	0.8209	0.9257	0.9257

Figure 7: Comparison of performance of the models implemented

5 Conclusion

This study demonstrated the effectiveness of Support Vector Machines in classifying forest cover types, with a focus on improving the model's performance on datasets with imbalance of classes. Several techniques were employed to improve model performance including **feature selection, dimensionality reduction, class imbalance handling, and hyperparameter tuning**. **Feature Selection outperformed PCA**, while **Cost-Sensitive SVM** achieved the best results, with **82.07% test accuracy** on unbalanced data. We did not specify a multi-class classification method, so **Scikit-learn's SVC used One-Versus-One (OvO) by default**, training separate SVM models for each class pair.

Overall, **Feature Selection + Cost-Sensitive SVM proved most effective**, avoiding overfitting while maintaining high accuracy. Future work can explore **ensemble models or deep learning** for further improvements.

References

1. Cost-Sensitive SVM for imbalanced classification.
2. Feature importance and feature selection using XGBoost.
3. Scikit-learn Documentation on SVC.
4. Imblearn documentation on SMOTENC
5. Imblearn documentation on Random Undersampling
6. UCI ML Repository