Classification of Forest Cover-type Based on Soil Characteristics Using Machine Learning

Zakia Sultana, Syed Mohtasib Mashruk, Rafeed Mahbub Rafi, Alistair Biswas
Mr. Md. Zahid Hossain, Mr. Md Rasheduzzaman, Prof. Dr. Md. Shamim Akhter
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology, Dhaka, 1208, Bangladesh.

***Abstract*—Forest coverage plays a vital role in biodiversity conservation, climate regulation, and ecosystem service provision. Accurate classification of forest cover types based on land characteristics is crucial for effective forest resource management. This project utilizes the "Covtype" dataset, containing detailed information on forest cover and land attributes, to develop machine learning models for classifying forest cover types. Following a comprehensive methodology involving environment setup, data exploration, preprocessing, and hyperparameter tuning, two models—Logistic Regression and Decision Tree—were employed. After rigorous cross-validation, the Decision Tree model demonstrated superior performance, achieving a test accuracy of 78.2%. The optimized model offers valuable insights for forest monitoring, conservation efforts, and environmental research by identifying key forest cover types, analyzing ecological diversity, and assessing anthropogenic impacts. This project ultimately contributes to sustainable forest management and ecological studies, benefiting forestry and environmental sectors.**

***Index Terms*—Forest cover classification, soil characteristics, machine learning, decision trees, logistic regression, gradient boosting, forest management, biodiversity conservation, ecological monitoring.**

I. Introduction

Different type of forest cover plays a crucial role in conserving, regulating climate, and provides essential ecosystem services. Accurate identification of forest cover types based on soil characteristics, lets us better monitor and manage forest resources, thereby preserved the natural environment. This understanding is super important for landscape planning, improving land use, prevent deforestation and create sustainable forest management strategies. Also, accurate on forest cover types are critical for scientific research in ecology and for detecting changes in ecosystems that may be linked to climate change.

Accurate classification of forest cover types can inform better agricultural practices and logging operations, balancing human impacts on forests. The 'Covtype' dataset [13], which includes detailed information on forest cover types and soil proprieties, gives a valuable resource for this purpose. By analyzing this data, we aim to develop machine learn models to classify forest cover types with high accuracy, supporting forest resource management and conservation efforts. This contribute to sustainable economic development by optimizing resource use and ensure long-term productivity of forest-related industries. Social development is also enhanced through the preservation of ecosystem services that forests provide, like clean air and water, which are crucial for human well-being.

Our research will focus on applying various machine learn algorithms, including decision trees, random forests, naive Bayes, logistic regression, gradient boosting, and multi-player-perception. These models will help improve the accuracy of forest cover classifications based on soil attributes. Understanding forest cover types through these models allows us to identify dominant tree species in specific areas and supports natural resource monitoring. The classifications model can detect changes in forest ecosystems, forecast issues such as tree diseases, and assess the impact of human activities. This provides forestry professionals with tools for better resource management, biodiversity measurement, and harvest planning. By ensure sustainable forest management, we support economic activities reliant on forest resources while maintaining ecosystem health.

Moreover, the classifications model we develop will be valuable for ecologists and researchers investigate forest ecosystems. It will facilitate the study of forest cover dynamics over time and support effort in forestry, nature protection, sustainable development, and environmental studies. By providing accurate and actionable insights, our model will contribute to more effective forest management and conservation practices, ensure the sustainability and health of forest ecosystems in the face of increasing human impact and climate alteration. This not only benefits the environment but also promotes social development by protecting the natural resources that communities depend on.

In summary, our project aims to leveraging machine learn technical—such as decision trees, random forest, missions Bayes, logistic regression, gradient boosting, and multi-player-perception—to create a robust classification model for forest cover types. This model will aid in the conservation of natural resources, the preservation of biodiversity, and the sustainable management of forest ecosystems. By supporting sustainable economic activities and enhancing social development through the preservation of essential ecosystem services, our work ultimately benefits various sectors, including forestry, agriculture, and environmental research.

The rest of the paper is organized as follows. Section II reviews the related work, Section III describes the dataset, Section IV visualizes the data in the dataset, Section V describes the methodology, Section VI shows the result analysis

of the models trained and finally Section VII concludes and mentions our future works regarding this project.

II. Related Work

The 'Covtype' dataset is easily accessible on the UCI ML repository, and so different scholars already proposed various methods in their work.

Blackard, J. A., & Dean, D. J. [1] proposed the Covtype dataset and benchmarked the dataset by comparing the accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. The study found that the neural network model predicted forest cover types more accurately than the discriminant analysis model.

Kumar & Sinha [2] applied the Random Forests algorithm to classify forest cover types using the Covtype dataset. The study achieved a classification accuracy of 94.6% on ten-fold cross-validation, significantly improving upon the 70.8% accuracy of the original work presented at UCI.

De Almeida et al. [3] introduced the Dynse framework, which dynamically selects classifiers to handle concept drift in data streams. The framework adapts over time by adding classifiers trained on new batches to a pool, from which a custom ensemble is selected for each test instance, showing improved accuracy and speed in classification tasks.

Tsai et al. [4] developed techniques for mapping vegetation and land use types in Fanjingshan National Nature Reserve using Google Earth Engine. It utilized multi-seasonal Landsat image composites and elevation ancillary layers to minimize cloud cover and terrain issues, achieving over 70% mapping accuracies with limited training data.

Johnson & Abdelfattah [5] tested various machine learning classifiers for identifying forest cover types. The Random Forest classifier emerged as the most efficient and accurate, suggesting potential future improvements and applications of the algorithm.

Kumar et al. [6] focused on forest cover dynamics and prediction modeling using a logistic regression model in Bhanupratappur Forest Division, India. It considered variables like distance from forest edge and achieved high predictive accuracy (ROC=87%) for forest cover change between 1990 and 2010.

Rojanavasu et al. [7] presented a self-organized, distributed, and adaptive rule-based induction system that improved classification accuracy and execution speed on large datasets by employing a supervised learning classifier system on top of a self-organized map neural network.

Narayan et al. [8] proposed Maxdiff kd-trees for data condensation, which improved representation for high condensation ratios by using the Maxdiff criterion to separate out distant clusters before further splitting, thus enhancing the speed and representation of the data.

Oza [9] introduced online versions of bagging and boosting ensemble learning methods, which process training data in one pass, showing comparable performance to batch algorithms and offering advantages in situations where data arrive continuously or are too large for multiple passes.

Liu, T., Yang, K., & Moore, A. W. [10] present the ioc algorithm, an efficient nonparametric classification method for high-dimensional data. It is particularly useful in areas like computer vision and has shown to outperform other approximate nearest neighbor approaches.

Koggalage, R., & Halgamuge, S. [11] propose a method to reduce the number of training samples required for Support Vector Machine classification. By using clustering techniques like K-mean, the authors could identify non-relevant samples and exclude them from training without affecting the classification results.

Yang, Y, & Webb, G. I. [12] introduce Non-Disjoint Discretization (NDD) for Naive-Bayes classifiers, which forms overlapping intervals for numeric attributes. This approach seeks to improve probability estimation by adjusting the number and size of discretized intervals based on the number of training instances.

III. Dataset

The data set used for our research is the well-known 'Covtype' dataset. This dataset contains tree observations from four areas of the Roosevelt National Forest in Colorado. All observations are cartographic variables (no remote sensing) from 30 meter x 30 meter sections of forest. There are over half a million measurements total.

This dataset is part of the UCI Machine Learning Repository [13]. The original database owners are Jock A. Blackard, Dr. Denis J. Dean, and Dr. Charles W. Anderson of the Remote Sensing and GIS Program at Colorado State University.

A. *Dataset info*

We have been given a total of 54 attributes, these attributes contain Binary and Quantative attributes, and we need to predict which Forest Cover-Type is it from the given features.

Attribute	Value
No. of Instance	581011
No. of Attributes (Features)	54
Associate Task	Classification
Dataset Characteristics	Multivariate
Missing Values	None
Area	Life
Target Variable	Forest Cover Type

TABLE I: Dataset Information

Our dataset contains 581,011 observations and 54 attributes to predict the kind of forest cover, our goal variable. The target variable contains 7 different classes, resulting in a Multi-Class Classification problem. Table I shows the overall information of the 'ocvtype' Dataset.

The study included ten quantitative variables, four binary variables (*Wilderness Area*), and 40 binary variables (*Soil Type*).

The wilderness region has four binary columns that can only be present once per observation, as seen in the Table II above.

Name	Data Type
Elevation	Numerical
Aspect	Numerical
Slope	Numerical
Horizontal Distance to Hydrology	Numerical
Vertical Distance to Hydrology	Numerical
Horizontal Distance to Roadways	Numerical
Hillshade 9am	Numerical
Hillshade 3pm	Numerical
Horizontal Distance to Fire Points	Numerical
Wildness Area (X4)	Binary
Soil Type (X40)	Binary

TABLE II: Data Types of Each Feature

There are four categories of wilderness areas, and each observation/instance can only include one of them. The same goes for the *Soil Type* feature. In machine learning, we refer to these characteristics as 'one-hot encoded'.

We will explore this Wilderness areas in detail:

- **Wilderness_Area1:** Rawah Wilderness Area
- **Wilderness_Area2:** Neota Wilderness Area
- **Wilderness_Area3:** Comanche Wilderness Area
- **Wilderness_Area4:** Cache La Poudre Wilderness Area

The Roosevelt National Forest's *Soil Type* feature has 40 columns representing 40 different types of soil obtained from four wilderness areas.

1) Cathedral family - Rock outcrop complex, extremely stony
2) Vanet - Ratake families complex, very stony
3) Haploborolis - Rock outcrop complex, rubbly
4) Ratake family - Rock outcrop complex, rubbly
5) Vanet family - Rock outcrop complex, rubbly
6) Vanet - Wetmore families - Rock outcrop complex, stony
7) Gothic family
8) Supervisor - Limber families complex
9) Troutville family, very stony
10) Bullwark - Catamount families - Rock outcrop complex, rubbly
11) Bullwark - Catamount families - Rock land complex, rubbly
12) Legault family - Rock land complex, stony
13) Catamount family - Rock land - Bullwark family complex, rubbly
14) Pachic Argiborolis - Aquolis complex
15) unspecified in the USFS Soil and ELU Survey
16) Cryaquolis - Cryoborolis complex
17) Gateview family - Cryaquolis complex
18) Rogert family, very stony
19) Typic Cryaquolis - Borohemists complex
20) Typic Cryaquepts - Typic Cryaquolls complex
21) Typic Cryaquolls - Leighcan family, till substratum complex
22) Leighcan family, till substratum, extremely bouldery
23) Leighcan family, till substratum, - Typic Cryaquolls complex
24) Leighcan family, extremely stony
25) Leighcan family, warm, extremely stony
26) Granile - Catamount families complex, very stony
27) Leighcan family, warm - Rock outcrop complex, extremely stony
28) Leighcan family - Rock outcrop complex, extremely stony
29) Como - Legault families complex, extremely stony
30) Como family - Rock land - Legault family complex, extremely stony
31) Leighcan - Catamount families complex, extremely stony
32) Catamount family - Rock outcrop - Leighcan family complex, extremely stony
33) Leighcan - Catamount families - Rock outcrop complex, extremely stony
34) Cryorthents - Rock land complex, extremely stony
35) Cryumbrepts - Rock outcrop - Cryaquepts complex
36) Bross family - Rock land - Cryumbrepts complex, extremely stony
37) Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony
38) Leighcan - Moran families - Cryaquolls complex, extremely stony
39) Moran family - Cryorthents - Leighcan family complex, extremely stony
40) Moran family - Cryorthents - Rock land complex, extremely stony

B. Forest Cover Type

This is the variable which we are going to predict and it has only one column which represents integer values from 1 to 7, where these digits represent type of forest cover for the observations. This is the variable which is not one-hot encoded like *Soil Type* and *Wilderness Area*, that is why it does not have 7 columns to represent each class for the observations. The names of these types of Forest Cover are:

1) Spruce/Fir
2) Lodgepole Pine
3) Ponderosa Pine
4) Cottonwood/Willow
5) Aspen
6) Douglas-fir
7) Krummholz

So, an observation which has '5' integer value in a *Forest Cover Type* column, it means that observation has *Aspen Forest Cover Type* for that observation.

As we have already discussed, there are 581,011 examples in the data set. Though it does not exactly qualify as big data, it is large enough to be manageable as an example and still highlight some issue of scale.

Table III shows the number of records for each Forest Type. As we can see, the distribution of these classes is unequal. Spruce and Lodgepole have the most records, whilst Cottonwood has the fewest. Unequal distribution will lead to unsatisfactory results.

Amount of distribution. We will do a thorough analysis and propose a solution for the project. It's worth noting that each

Forest Type	No. of Records
Spruce / Fir	211840
Lodgepole Pine	283301
Ponderosa Pine	35754
Cottonwood / Willow	2747
Aspen	9493
Douglas-Fir	17367
Krummholz	20510
Total Records	581011

TABLE III: Number of Records for Each Forest Type

observation is allocated to a forest type class, and there are no empty observations, which is beneficial for our analysis.

Let's analyze the statistics for the specified features in Table IV :

Feature Name	Mean	Std Dev
Elevation	2959.36	279.98
Aspect	155.65	111.91
Slope	14.10	7.49
Horizontal Distance to Hydrology	269.43	212.55
Vertical Distance to Hydrology	46.42	58.30
Horizontal Distance to Roadways	2350.15	1559.25
Hillshade 9am	212.15	26.77
Hillshade Noon	223.32	19.77
Hillshade 3pm	142.53	38.27
Horizontal Distance to Fire Points	1980.29	1324.19

TABLE IV: Mean and Standard Deviation of Each Feature

IV. Dataset Visualization

In this project, we analyzed the distribution of forest cover types within a dataset. The dataset consists of seven distinct cover types, with the majority of instances belonging to class 2 (283,301 instances) and class 1 (211,840 instances). The remaining classes contain significantly fewer instances, with class 3 comprising 35,754 instances, class 4 comprising 2,747 instances, class 5 comprising 9,493 instances, class 6 comprising 17,367 instances, and class 7 comprising 20,510 instances. This imbalance in class distribution is an important factor to consider in the model training process, as it may impact classification performance.

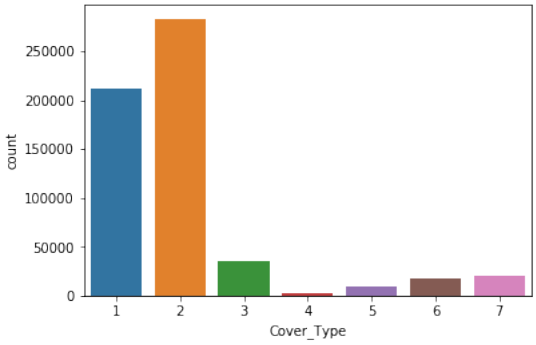


Fig. 1: Number of instances belonging to each class

A. Correlation

In this analysis, we explored the correlation between the variables to understand their relationships and potential impact on the classification task. Correlation measures the strength and direction of a linear relationship between two variables, with values ranging from -1 to 1. A positive correlation indicates that as one variable increases, the other tends to increase as well, while a negative correlation implies an inverse relationship.

By examining the correlation matrix, we can identify pairs of features that are highly correlated. Highly correlated features may introduce redundancy and multicollinearity in the model, potentially affecting its performance. Conversely, low or no correlation suggests that the variables provide unique information to the model. Understanding these relationships helps in feature selection and engineering, ensuring that we optimize the predictive power of the model while avoiding overfitting or irrelevant input.

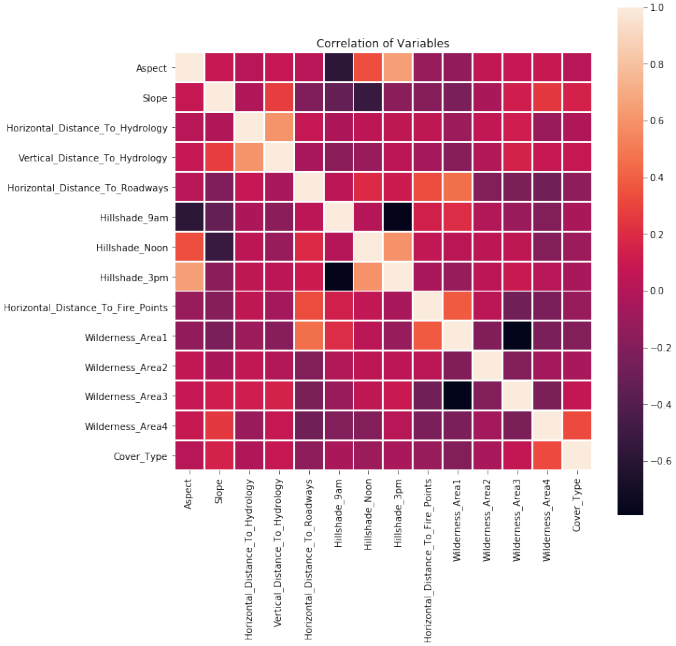


Fig. 2: Correlation of variables

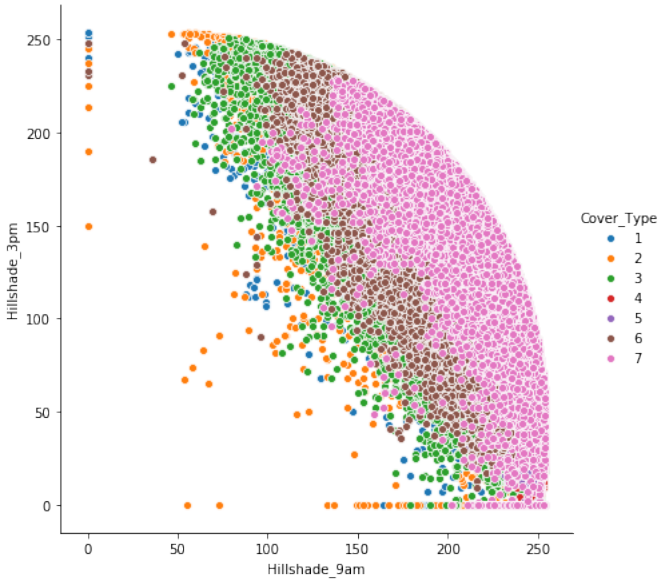
B. Scatter Plot (pairplot)

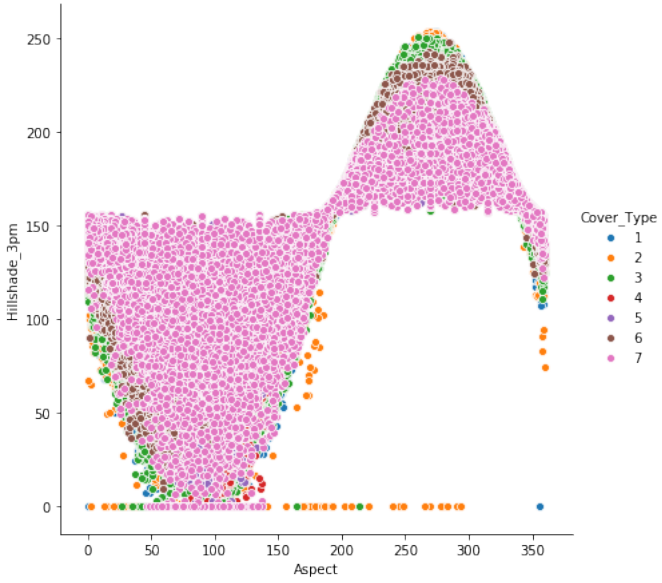
The scatter plots (pairplot) provide a visual representation of the relationships between different feature pairs, highlighting how data points are distributed across the various classes. Upon analysis, we observe considerable overlap in the class distribution, indicating that the classes are not easily separable based on these features alone.

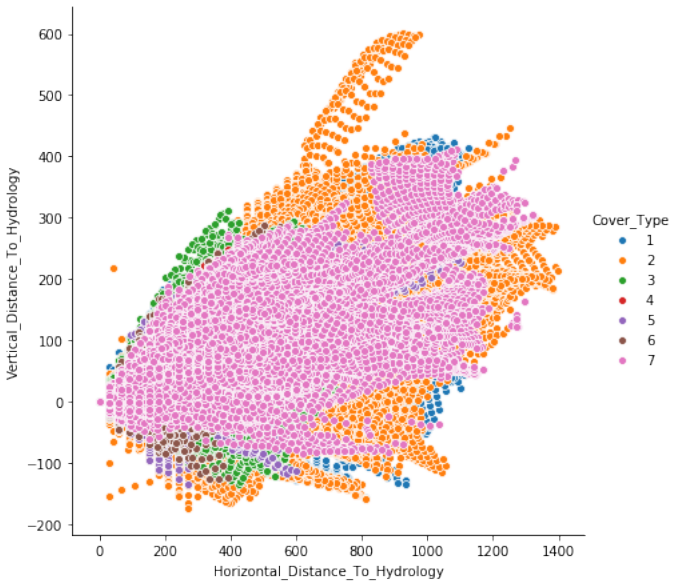
A distinct ellipsoid pattern emerges from the Hillshade-related features (Hillshade_9am, Hillshade_Noon, Hillshade_3pm), suggesting a strong correlation among these attributes. Additionally, the Aspect and Hillshade attributes form a sigmoid-like pattern, revealing non-linear relationships between these variables.

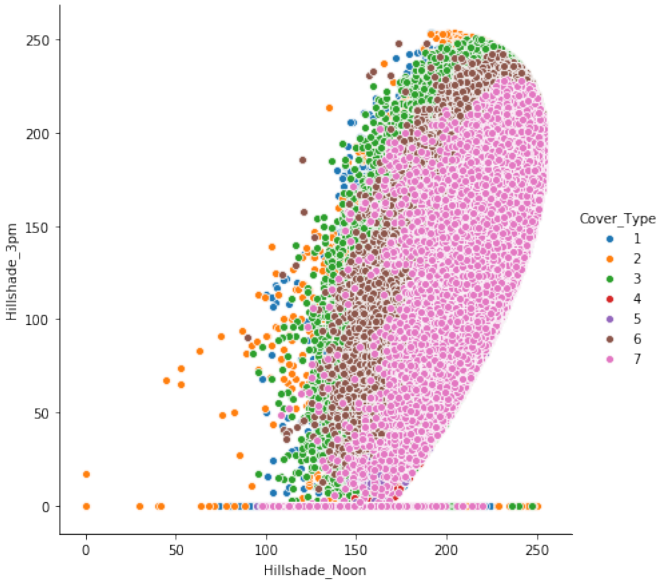
Interestingly, the horizontal and vertical distance to hydrology features exhibit an almost linear pattern, indicating

a more straightforward relationship between these two variables. These insights are critical for feature selection, as they highlight which variables may hold stronger predictive power for distinguishing between classes.









V. Methodology

A. Setting up the Environment

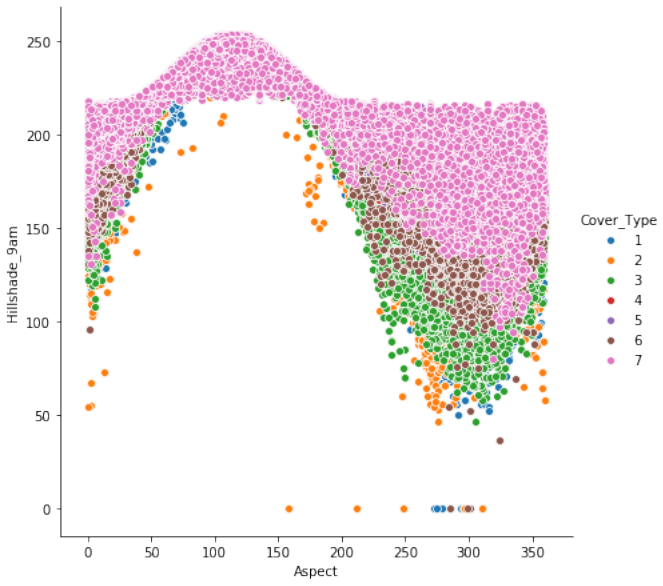
PySpark was the primary tool used for handling large datasets and building machine learning models. Spark was used to process the large dataset efficiently. Then PySpark SQL was used to load and manipulate data in a distributed manner For machine learning algorithms and model building, PySpark MLlib was used. The environment used for running and documenting code was Google Colaboratory.

B. Exploring the Data

1) Data Overview: The Covertype dataset was read using PySpark's `SparkSession.read()` function, allowing the dataset to be loaded into a distributed DataFrame. `spark.read.csv()` was used to load the dataset. Then the schema was inferred automatically, and the data types were inspected with `.printSchema()`. The numbers of observations (rows) in the DataFrame were counted and stored in a variable for future calculation.

2) Exploratory Data Analysis (EDA):

- **Correlation Matrix:** PySpark was used to calculate the correlation between numerical features, and the results were visualized using Matplotlib to plot a heatmap.
- **Scatterplot (Pairplot):** Matplotlib was used to create scatterplots for a more detailed visual exploration of



the relationships between features, especially elevation, slope, and soil type.

C. Preprocessing and Splitting the Data

In this part, we created stages to convert the feature and label variables to appropriate type for analysis. Then, we created a preprocessing pipeline to be used for future transformation of similar DataFrame.

We started by creating the list of categorical and numerical features. Then, we encoded/assembled them using the following process:

- **StringIndexer:** Categorical features like soil type and wilderness area were indexed using PySpark's StringIndexer, which assigns numerical labels to these categorical variables.
- **OneHotEncoder:** To avoid ordinal relationships among categorical features, OneHotEncoder was used, converting the indexed features into binary vectors. Logistic regression model needs the feature indexing as well as oneHotEncoding of categorical variables while decision tree performs better with feature indexing without oneHotEncoding.
- **VectorAssembler:** All features (numerical and encoded categorical) were combined into a single vector using VectorAssembler, preparing the dataset for the machine learning model.

The dataset was split into training and testing sets using `randomSplit()`. PySpark was used to partition the data into 80% training and 20% test sets, ensuring a balanced representation of all classes. We confirmed the split by counting the number of rows in each DataFrame.

D. Hyperparameter Tuning for Logistic Regression

We performed the hyperparameter tuning using accuracy as model performance evaluator and evaluate the logistic regression models thus created. We started by creating an instance of `MulticlassClassificationEvaluator` to calculate the accuracy metric. Then, we created a `LogisticRegression` instance, a grid search process was implemented to optimize hyperparameters, particularly the `regParam` (regularization parameter) and `elasticNetParam` (elastic net mixing). We then identified the optimal model under the grid search parameters above and calculate the CV Score, Lambda and Alpha of the optimal model.

E. Hyperparameter Tuning for Decision Trees

In this part, we performed the hyperparameter tuning using accuracy as model performance evaluator and evaluate the Decision Tree models thus created. We started by creating a `DecisionTreeClassifier` instance, create a grid search parameters and then cross validate the models within some arbitrary search parameters of the maximum depth of the tree and the minimum number of instances each leaf node should have. Then, we identified the optimal model under the grid search parameters above and calculate the CV Score, Lambda and Alpha of the optimal model. Finally, we calculated and displayed each of the features and their importance as evaluated by the optimal decision tree model.

F. Identifying and Evaluating the Final Model

We compared the optimal logistic regression and optimal decision tree model to identify the best model between them. We then evaluated the best model's performance without of sample (test) data. The evaluation metrics used were:

- **Confusion Matrix:** For each model, a confusion matrix was computed using PySpark to identify true positives, false positives, true negatives, and false negatives for each cover type. This matrix helped visualize the model's classification performance for all cover types.
- **Precision and Recall:** The precision and recall values were calculated for each forest cover type. These metrics helped assess how well the models identified each type of forest correctly without overclassifying incorrect types.
- **Final Model Selection:** Based on the performance metrics (accuracy, precision, recall), the best-performing model was selected as the final model for forest type classification.

VI. Result Analysis

The results from both models—Logistic Regression (LR) and Decision Tree (DT)—have been compared to determine the best-performing model for our classification task.

A. Logistic Regression Model

The optimal Logistic Regression model was identified with an **alpha of 0.0** and a **lambda of 1e-05**. After cross-validation using a 5-fold approach, the model achieved a **maximum cross-validation (CV) score of 0.6741** on the training data. This indicates that the Logistic Regression model predicted correctly about 67.4% of the time during training. Despite the

tuning of hyperparameters, the performance of the Logistic Regression model remained suboptimal, particularly when compared to the Decision Tree model.

B. Decision Tree Model

The optimal Decision Tree model was selected with a **maximum depth of 16** and a **minimum instances per node of 1**. It achieved a **maximum CV score of 0.7775**, meaning it predicted correctly approximately 77.8% of the time on the training data. This clearly outperforms the Logistic Regression model in terms of training accuracy. The deeper structure and non-linear decision boundaries of the Decision Tree allow it to capture complex relationships in the data that Logistic Regression, being a linear model, cannot.

C. Test Set Evaluation

Given the superior performance of the Decision Tree during training, it was selected for further evaluation on the test set. The Decision Tree model achieved a **test set accuracy of 0.7818** (78.2%), closely aligning with its training performance, which suggests that the model generalizes well to unseen data.

D. Confusion Matrix

The confusion matrix for the Decision Tree model provides deeper insight into its classification performance:

Actual/Predicted	1	2	3	4	5	6	7
1	311	90	2	0	13	5	36
2	101	220	12	0	58	11	3
3	1	10	306	23	13	78	0
4	0	1	22	423	0	25	0
5	9	26	3	0	364	4	0
6	1	4	58	13	5	319	0
7	19	7	0	0	2	0	404

The confusion matrix shows that most classes are well predicted, but some degree of misclassification still exists, especially between certain cover types. For example, class 1 has some overlap with class 2 and class 7, and class 3 exhibits confusion with classes 4 and 6. This suggests that while the Decision Tree model performs well overall, improvements could potentially be made to minimize these misclassifications, possibly through additional tuning or the use of ensemble methods.

E. Precision and Recall Analysis

Based on the test set, the precision and recall scores for each label type are as follows:

Label	Precision	Recall
1	0.7036	0.6805
2	0.6145	0.5432
3	0.7593	0.7100
4	0.9216	0.8981
5	0.8000	0.8966
6	0.7217	0.7975
7	0.9120	0.9352

Some observations based on the precision and recall values of the final model:

- Label 7 (Cover Type 7) has the highest recall value (0.9352), indicating that this cover type is most likely to be correctly classified by our final model.
- Label type 2 (Cover Type 2) has the lowest recall score (0.5432), meaning that the probability of identifying a cover type as type 2 out of all true type 2 instances is only about 54.32%, making it the most likely to be misclassified.
- Label type 5 (Cover Type 5) has the highest difference in precision and recall values (difference: 0.0966). There is about an 80% probability that the model correctly identifies Cover Type 5, while there is about an 89.7% probability that it correctly identifies a cover type as Cover Type 5 among all actual Cover Type 5 instances.

Based on the results of cross-validation and test set performance, the **Decision Tree model** clearly outperforms the Logistic Regression model, with a higher accuracy on both the training and test sets. Therefore, the Decision Tree model is selected as the optimal model for this classification task.

VII. Conclusion and Future Work

In this research, we aimed to develop an accurate model for classifying forest cover types using the "Covtype" dataset, applying machine learning techniques. After preprocessing the data and performing hyperparameter tuning, two models—Logistic Regression and Decision Tree—were trained and evaluated. The Decision Tree model outperformed Logistic Regression, achieving a cross-validation score of 77.8% and a test accuracy of 78.2%, demonstrating strong predictive power and generalization capabilities. Additionally, the precision and recall analysis revealed that Cover Type 7 had the highest recall, while Cover Type 2 exhibited the most misclassification, which can be further investigated.

While the Decision Tree model achieved satisfactory results, there is room for future improvements. Potential areas of future work include:

- **Exploring Ensemble Methods:** Implementing advanced ensemble techniques such as Random Forest or Gradient Boosting could help reduce misclassification and further improve model accuracy.
- **Feature Engineering:** Additional domain-specific feature engineering could be conducted to better capture the relationships between land traits and forest cover types.
- **Handling Imbalanced Data:** Investigating techniques like oversampling, undersampling, or class-weight adjustments to address class imbalances, especially for Cover Types with lower precision and recall values, could improve classification outcomes.
- **Deploying the Model:** The final Decision Tree model could be deployed in a real-world environment to assist in forest management and monitoring, offering practical utility for conservation efforts.

This project contributes to the broader goal of leveraging machine learning for environmental monitoring and sustainable forest management. Future work in these areas could further enhance the accuracy, reliability, and applicability of forest cover classification models.

REFERENCES

[1] Blackard, J. A., & Dean, D. J. (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Computers and electronics in agriculture, 24(3), 131-151.

[2] Kumar, A., & Sinha, N. (2020). Classification of forest cover type using random forests algorithm. In Advances in data and information sciences (pp. 395-402). Springer, Singapore.

[3] De Almeida, P. R. L., et al. (2016). Handling concept drifts using dynamic selection of classifiers. In 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE.

[4] Tsai, Y. H., Stow, D., Chen, H. L., Lewison, R., An, L., & Shi, L. (2018). Mapping vegetation and land use types in Fanjingshan National Nature Reserve using google earth engine. Remote Sensing, 10(6), 927.

[5] Johnson, P., & Abdelfattah, E. (2018, November). Applying machine learning models to identify forest cover. In 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 471-474). IEEE.

[6] Kumar, R., Nandy, S., Agarwal, R., & Kushwaha, S. P. S. (2014). Forest cover dynamics analysis and prediction modeling using logistic regression model. Ecological Indicators, 45, 444-455.

[7] Rojanavasu, P., et al. (2009). A self-organized, distributed, and adaptive rule-based induction system. IEEE Transactions on Neural Networks, 20(3), 446 459.

[8] Narayan, B. L., Murthy, C. A., & Pal, S. K. (2006). Maxdiff kd-trees for data condensation. Pattern Recognition Letters, 27(3), 187.

[9] Oza, N. C. (2005). Online bagging and boosting. In Proceedings of IEEE International Conference on Systems Man and Cybernetics (Vol. 3, pp. 2340 2345). New Jersey, USA. Special Session on Ensemble Methods for Extreme Environments.

[10] Liu, T., Yang, K., & Moore, A. W. (2004). The ioc algorithm: Efficient many-class nonparametric classification for high-dimensional data. In Proceedings of the 2004 ACMSIGKDD International Conference on Knowledge Discovery and Data Mining- 634). New York, USA: ACM Press.

[11] Koggalage, R., & Halgamuge, S. (2004). Reducing the number of training samples for fast support vector machine classification. Neural Information Processing-Letters and Reviews, 2.

[12] Yang, Y, & Webb, G. I. (2002). Non-disjoint discretization for naive-bayes classifiers. In A. G. Hoffmann Sammut (Ed.) Proceedings of the 19th International Conference on Machine Learning (ICML'02) (pp 666 - 673). San Francisco, USA: Morgan Kaufmann Publishers Inc.

[13] Blackard,Jock. (1998). Covertype. UCI Machine Learning Repository. https://doi.org/10.24432/C50K5N.