# Forest Cover Type Prediction

Chirag Chandrashekar
*Electrical, Computer and Energy Engineering*
*University of Colorado Boulder*
Boulder, Colorado
chch4032@colorado.edu

***Abstract*—Accurately predicting forest cover types is important because it helps us understand how ecosystems and environments are related. It also helps us manage and preserve forests effectively. This study aims to create reliable and precise forest cover type prediction models by analyzing feature correlations and comparing the effectiveness of different Data Analysis Techniques like logistic regression, neural networks, principal component analysis (PCA), and bagging.**

***Index Terms*—Principal Component Analysis (PCA), Logistic Regression, k-Nearest Neighbors (KNN), Neural Network, Bagging, Correlation Matrix, Confusion Matrix**

## I. Introduction

Maintaining the Earth's ecological balance, conserving biodiversity, regulating climate, and providing crucial resources for both humans and other organisms are all reasons why forests are essential. As a result, it's crucial to obtain accurate and timely information regarding forest cover types for effective forest management, ecological research, and conservation initiatives. Our study aims to develop trustworthy and precise forest cover type prediction models by analyzing feature correlations and evaluating the efficiency of different Data Analysis Techniques: logistic regression, neural networks, principal component analysis (PCA), and bagging. Our goal is to identify complex and non-linear relationships between various features, examine their impact on forest cover types, and compare the performance of different regression techniques to provide insights into their respective advantages and limitations. This study will enhance the accuracy and interpretability of prediction models, providing researchers with valuable insights.

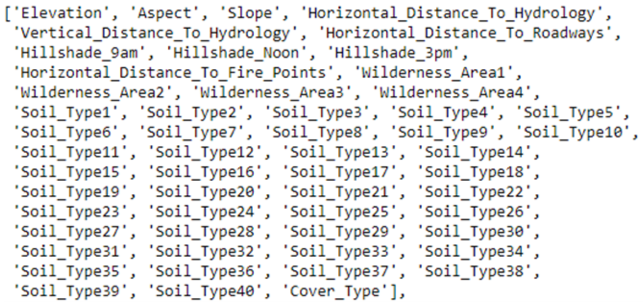
## II. Dataset Description

The dataset under consideration contains records representing diverse forest cover types observed across four separate wilderness areas located in the Roosevelt National Forest of Northern Colorado. Each observation corresponds to a 30m x 30m patch.

The dataset includes 581,012 samples and 55 distinct attributes. It provides valuable details such as elevation, distance from water sources and roads, and soil type. A comprehensive visualization of the dataset's features can be found in Figure 1. A brief overview of some noteworthy attributes within the dataset is provided below:

- The average elevation is 2,959.365 m, while the median value is 2,996 m.
- The median aspect is characterized by a 127-degree azimuth, with a median slope of 13 degrees.
- The mean horizontal distance to hydrology is 269.428 m, and the average vertical distance is 46.418 m.
- The highest elevation reaches 3,858 m.
- The mean horizontal distance to roadways measures 2,350.146 m, with the minimum distance being 0 m and the maximum spanning 7,117 m.

However, it lacks pre-defined train-test splits. To address this, we'll use stratified sampling to create our own divisions in an 80:20 ratio.



```
['Elevation', 'Aspect', 'Slope', 'Horizontal_Distance_To_Hydrology',
 'Vertical_Distance_To_Hydrology', 'Horizontal_Distance_To_Roadways',
 'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm',
 'Horizontal_Distance_To_Fire_Points', 'Wilderness_Area1',
 'Wilderness_Area2', 'Wilderness_Area3', 'Wilderness_Area4',
 'Soil_Type1', 'Soil_Type2', 'Soil_Type3', 'Soil_Type4', 'Soil_Type5',
 'Soil_Type6', 'Soil_Type7', 'Soil_Type8', 'Soil_Type9', 'Soil_Type10',
 'Soil_Type11', 'Soil_Type12', 'Soil_Type13', 'Soil_Type14',
 'Soil_Type15', 'Soil_Type16', 'Soil_Type17', 'Soil_Type18',
 'Soil_Type19', 'Soil_Type20', 'Soil_Type21', 'Soil_Type22',
 'Soil_Type23', 'Soil_Type24', 'Soil_Type25', 'Soil_Type26',
 'Soil_Type27', 'Soil_Type28', 'Soil_Type29', 'Soil_Type30',
 'Soil_Type31', 'Soil_Type32', 'Soil_Type33', 'Soil_Type34',
 'Soil_Type35', 'Soil_Type36', 'Soil_Type37', 'Soil_Type38',
 'Soil_Type39', 'Soil_Type40', 'Cover_Type'],
```

Fig. 1. Features of Dataset

## III. Data Pre-processing

### A. Data Cleaning

Prior to implementing any modeling or inference methodologies on the dataset, an analysis and cleaning procedure was executed. The data underwent the subsequent pre-processing stages:

- Removed multiple entries by the same user.
- Handled Nan values by substituting them with Zeros.

### B. Correlation Matrix

A correlation matrix is a valuable tool for assessing the linear relationship between pairs of features within a dataset. By analyzing the correlation matrix for first 10 attributes of the dataset, better analysis of the dataset can be done. In this process, each cell of the matrix represents the correlation coefficient (ranging from -1 to 1) between the corresponding pair of features. A correlation close to 1 indicates a strong positive relationship, while a value close to -1 signifies a strong negative relationship. Conversely, a correlation coefficient near

0 suggests little to no linear relationship between the features. Following the evaluation of the correlation matrix for the features, Figure 2 presents the heat map showcasing the associations between the relevant attributes. Highly correlated features values are shown in Figure 3.

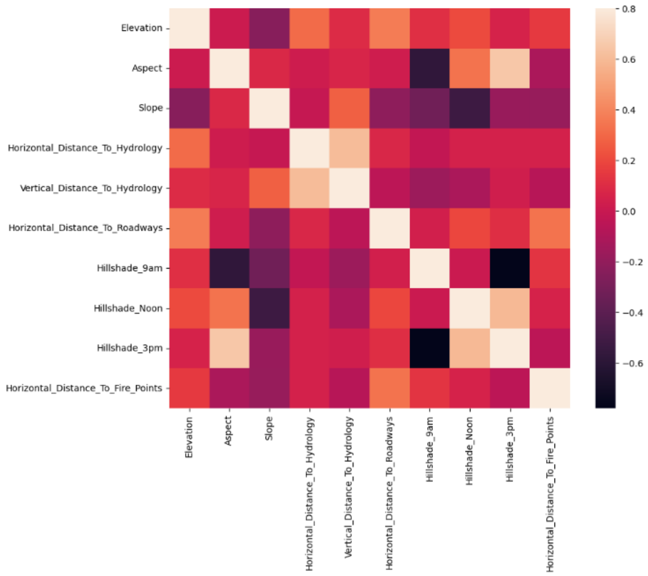
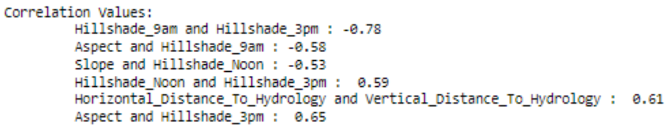


Fig. 2. Heat Map



```
Correlation Values:
        Hillshade_9am and Hillshade_3pm : -0.78
        Aspect and Hillshade_9am : -0.58
        Slope and Hillshade_Noon : -0.53
        Hillshade_Noon and Hillshade_3pm :  0.59
        Horizontal_Distance_To_Hydrology and Vertical_Distance_To_Hydrology :  0.61
        Aspect and Hillshade_3pm :  0.65
```

Fig. 3. Highly Correlated Features

Better visualization of the correlation matrix can be done using the pair plot depicted in Figure 4, Figure 5 and Figure 6.

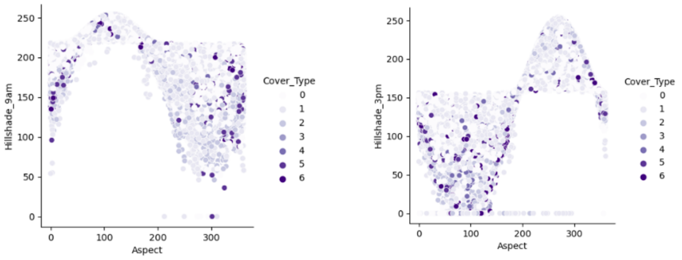


Fig. 4. Pair Plot

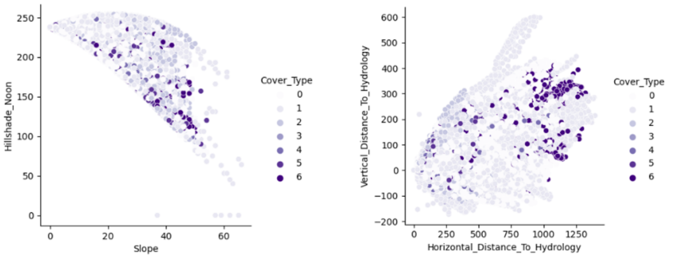


Fig. 5. Pair Plot

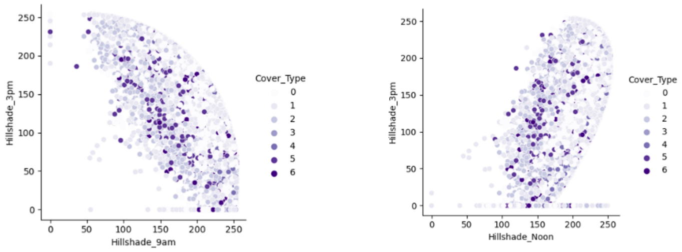


Fig. 6. Pair Plot

### C. Standardizing the data

Standardization of data is a crucial preprocessing step in machine learning that ensures the features within a dataset are on a similar scale. This process involves transforming the original feature values such that they have a mean of 0 and a standard deviation of 1. Standardization is particularly important when working with algorithms that are sensitive to the magnitude of feature values or rely on distance metrics, such as linear regression, support vector machines, and k-Nearest Neighbors. By ensuring that all features have comparable scales, standardization prevents the model from being biased towards features with larger numerical ranges, leading to improved performance and faster convergence. Moreover, standardization can mitigate the impact of outliers, enhancing the model's ability to learn patterns within the data and generalize to unseen instances. It is essential to perform standardization separately for the training and testing data to avoid information leakage and maintain the integrity of the validation process.

## IV. Data Analysis Techniques

### A. Logistic Regression

A logistic regression model was employed to make predictions, subsequently determining the accuracy of the training and testing data. The model achieved a training accuracy of 72.46% and a testing accuracy of 72.32%, highlighting its effectiveness in classifying the target variable.

### B. Neural Network

A neural network technique was utilized to estimate the accuracy of the training and testing data. This sophisticated computational model, inspired by the structure and functionality of biological neural networks, was implemented to learn and adapt to the patterns within the data. Through the iterative training process, the neural network model successfully minimized the error between its predictions and the true target values. As a result, the model achieved a training accuracy of 87.65% and a testing accuracy of 87.17%, demonstrating its effectiveness in classifying the target variable and its ability to generalize to unseen data.

An alternative approach involved developing a neural network implementation from scratch. However, this custom implementation was hitting the maximum implementation time, hence not yielding any desirable result.

### C. k-Nearest Neighbors (KNN)

k-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm that predicts the target variable based on the similarity of instances within its neighborhood. KNN method was attempted for implementation; however, it did not yield successful results in providing accuracy for the training and testing data due to its extensive computational time. The algorithm's high computational complexity, particularly for large datasets, limited its effectiveness in this specific application.

### D. Principal Component Analysis

The Principal Component Analysis (PCA) technique was employed as a preprocessing step for the dataset, with the objective of enhancing the accuracy of the test and training data. PCA is a dimensionality reduction method that transforms the original features into a new set of uncorrelated components, capturing the maximal amount of variance in the data. By reducing the dimensionality, PCA can address issues related to multicollinearity and overfitting, thereby improving model performance. After implementing PCA and training the model on the transformed data, the obtained accuracies were 53.85% for the training data and 53.92% for the test data. These results reflect the model's capability in classifying the target variable and its generalization to unseen data, albeit with a relatively moderate level of accuracy. This accuracy was noticed when the features where reduced from 55 to 2.

PCA was executed with varying component values to examine the impact on accuracy, and Figure 7 provides an effective visualization of the accuracy fluctuations as the number of components changes.

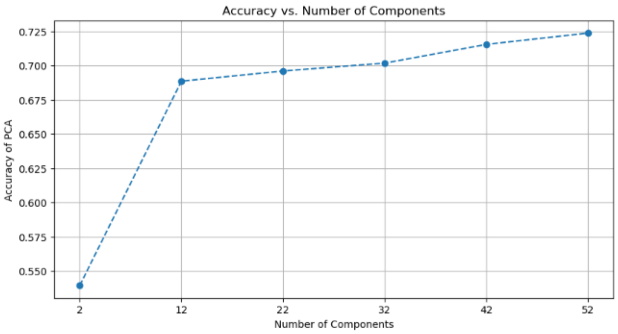
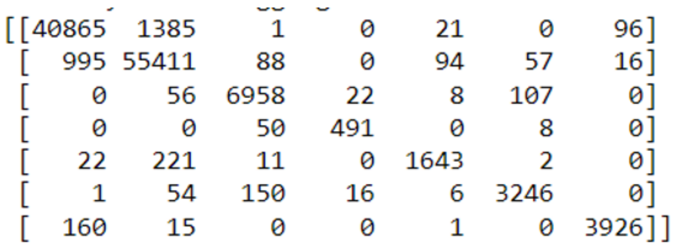


Fig. 7. PCA Analysis

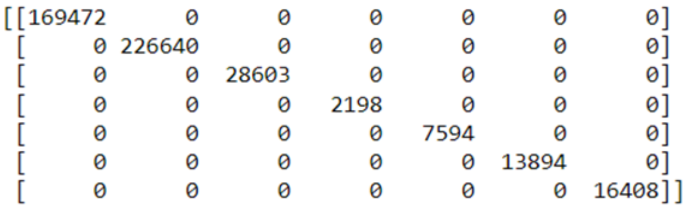
### E. Bagging

Bagging, or Bootstrap Aggregating, is an ensemble learning technique that aims to improve the performance and stability of a model by combining the outputs of multiple base models, which are trained on different subsets of the original data generated using random sampling with replacement. This approach reduces overfitting and enhances the model's generalization capabilities. The Bagging method was employed to determine the accuracy of the test and training data, achieving a test accuracy of 96.85% and a perfect training accuracy of 100%. The confusion matrices for the predictions of the training and testing data are depicted in Figure 9 and Figure 8 respectively, providing a comprehensive representation of the model's classification performance.



```
[[40865  1385     1     0    21     0    96]
 [  995 55411    88     0    94    57    16]
 [    0    56  6958    22     8   107     0]
 [    0     0    50   491     0     8     0]
 [   22   221    11     0  1643     2     0]
 [    1    54   150    16     6  3246     0]
 [  160    15     0     0     1     0  3926]]
```

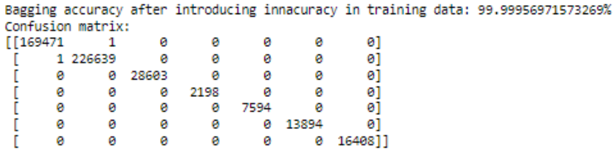
Fig. 8. Confusion Matrix for Test Data



```
[[169472      0      0      0      0      0      0]
 [     0 226640      0      0      0      0      0]
 [     0      0  28603      0      0      0      0]
 [     0      0      0   2198      0      0      0]
 [     0      0      0      0   7594      0      0]
 [     0      0      0      0      0  13894      0]
 [     0      0      0      0      0      0  16408]]
```

Fig. 9. Confusion Matrix for Training Data

To gain a deeper understanding of the model's sensitivity, a slight inaccuracy was introduced to the 'Aspects' feature of the training data to observe the effect on accuracy. Figure 10 presents the corresponding accuracy and confusion matrix, illustrating the model's performance under these altered conditions.



```
Bagging accuracy after introducing innacuracy in training data: 99.99956971573269%
Confusion matrix:
[[169471      1      0      0      0      0      0]
 [     1 226639      0      0      0      0      0]
 [     0      0  28603      0      0      0      0]
 [     0      0      0   2198      0      0      0]
 [     0      0      0      0   7594      0      0]
 [     0      0      0      0      0  13894      0]
 [     0      0      0      0      0      0  16408]]
```

Fig. 10. Confusion Matrix for Training Data

## V. Conclusion

In conclusion, this paper presented a comprehensive study on the application of various machine learning techniques to analyze and predict outcomes based on the given dataset. The methods employed included logistic regression, neural networks, Principal Component Analysis (PCA), and Bagging. The study demonstrated the effectiveness of these techniques in handling different aspects of the data analysis process, such as dimensionality reduction, model generalization, and ensemble learning.

The results obtained from the logistic regression and neural network models revealed their capability in classifying the target variable, with varying levels of accuracy for training and testing data. The PCA technique was utilized to explore the impact of dimensionality reduction on the model's performance, and it was observed that the number of components could influence the model's accuracy. The Bagging method showcased its potential in improving model performance and stability by combining the outputs of multiple base models.

Furthermore, the sensitivity analysis was conducted by introducing a small inaccuracy to the 'Aspects' feature in the training data, providing insight into the model's robustness under altered conditions. The confusion matrices for the training and testing data predictions were also analyzed to evaluate the classification performance.

The significance of choosing the right machine learning methods and comprehending their capabilities and limitations for successful data analysis and prediction is emphasized in this paper. To improve model performance and produce more reliable predictions, future research can explore supplementary techniques, hyperparameter tuning, and feature engineering.

## Acknowledgment

I want to express my sincerest gratitude to Prof. Mahesh Varanasi, Altug Karakurt, and my fellow peers for the invaluable guidance and support they provided throughout the completion of this paper. Their expertise, constructive feedback, and encouragement have been instrumental in refining the research and enhancing the overall quality of the work presented. The collaboration and intellectual discussions fostered by the academic community have greatly contributed to my understanding of the subject matter and growth as a researcher. I extend a special thanks to the professor for sharing their knowledge and fostering a stimulating learning environment, the teaching assistants for their dedication and assistance, and the peers for their camaraderie and collaborative spirit.

## References

[1] https://www.kaggle.com/datasets/uciml/forest-cover-type-dataset

[2] Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

[3] https://scikit-learn.org/stable/install.html

[4] https://www.tensorflow.org

[5] https://www.obviously.ai/post/data-cleaning-in-machine-learning

[6] https://builtin.com/data-science/correlation-matrix

[7] https://machinelearningmastery.com/logistic-regression-for-machine-learning/

[8] https://www.ibm.com/topics/neural-networks

[9] Howley, T., Madden, M.G., O'Connell, ML., Ryder, A.G. (2006). The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data. In: Macintosh, A., Ellis, R., Allen, T. (eds) Applications and Innovations in Intelligent Systems XIII. SGAI 2005. Springer, London. https://doi.org/10.1007/1-84628-224-1_16

[10] https://www.researchgate.net/profile/Bhavna-Reddy/post/how_text_classification_is_based_on_rocchios_method/attachment/59d623706cda7b8083a1e0d1/AS%3A331938180157440%401456151637583/download/knn+document+classification+%281%29.pdf