Comprehensive Analysis of Forest Cover Types: Data Preparation, Exploration, Modelling, and Evaluation

Abstract

This report presents a comprehensive data science project focused on predicting forest cover types based on cartographic variables. The project encompassed three main tasks: Problem Formulation, Data Acquisition and Preparation (Task 1), Data Exploration (Task 2), and Data Modelling (Task 3). Task 1 involved loading the dataset from the UCI repository, where a dataset consisting of 581,012 instances and 54 columns was selected. Task 2 involved in-depth data exploration, analysing numerical and categorical columns to understand their distributions and relationships. Additionally, the influence of slope on forest cover types was investigated. Task 3 focused on data modelling, comparing the performance of K-Nearest Neighbors (KNN) and Decision Tree (DT) classifiers on different training and test set splits. The KNN model outperformed the DT model consistently, exhibiting better accuracy, precision, recall, and F1 scores. Additionally, both models exhibited reasonable generalisation abilities without significant signs of overfitting or underfitting. The project provided valuable insights into the dataset, identified important variables, and recommended the best-performing model for predicting forest cover types.

Introduction

In this data science project, the overall workflow can be divided into three main tasks: Problem Formulation, Data Acquisition and Preparation (Task 1), Data Exploration (Task 2), and Data Modelling (Task 3).

Task 1 involved loading the dataset from the UCI repository. The dataset chosen for this project focuses on predicting forest cover type based on cartographic variables. The dataset contains information about a specific study area in the Roosevelt National Forest of northern Colorado. The dataset consists of 581,012 instances (observations) and 54 columns of data. The target variable is the forest cover type.

Task 2 focused on data exploration, where numerical and categorical columns were analysed. For numerical columns, descriptive statistics, histograms, and boxplots were generated using custom functions like explore_numerical_column(). Categorical columns were explored using descriptive statistics and bar plots with the help of explore_categorical_column(). Pairs of columns were also analysed using scatter plots, correlation coefficients, and boxplots to examine relationships between variables.

Task 3 involved data modelling. The dataset was split into different training and test sets to evaluate the performance of two chosen classification models. The scikit-learn package was used to implement the chosen models. Appropriate model parameters were selected. The models' performances were evaluated on both training and test sets using performance metrics. The two models were compared through graphical visualisations based on the evaluation results. Key observations, analyses, and conclusions from the comparison were provided, along with a recommendation on which model should be used.

This report describes the workflow, justified choices made during the process, and addressed any encountered issues and their resolutions in completing Task 1, Task 2, and Task 3.

Task 1: Problem Formulation, Data Acquisition and Preparation

In Task 1, the goal was to perform data preparation and exploration on the chosen dataset from the UCI repository. The workflow involved several key components. This section provides an overview of the workflow, key observations, choices made, and any encountered issues along with their resolutions.

Workflow

Loading the Dataset: The dataset was loaded using the pd.read_csv() method and the column names were manually specified. This was done by creating a list of column names based on the attribute information provided.

Checking for Missing Values: The df.info() function was used to examine the data types and the number of non-null values in each column. The code snippet df.isnull().any().sum() and df.isnull().any(axis=1).sum() was used to print the number of columns and rows with missing values, respectively.

Checking for Duplicate Rows: df.duplicated().sum() function was used to check for duplicate observations.

Class Distribution: Class imbalance was checked within the 'Cover_Type' column of the dataset.

Handling Categorical Features: It was observed that the dataset represented categorical features, specifically soil type and wilderness area, using binary columns. A new categorical column 'soil_type' was created based on binary soil type columns. Similarly, a 'wilderness_area' column was created. The original binary columns were then dropped from the DataFrame.

Key Observations and Analyses

Dataset Description and Size: The dataset focused on predicting forest cover type based on cartographic variables. It contains 581,012 instances (observations) and 54 columns of data, including 10 quantitative variables, 4 binary wilderness area variables, and 40 binary soil type variables.

Class Imbalance: The dataset exhibits a varied distribution of forest cover types. The two most prevalent types are Spruce/Fir and Lodgepole Pine, with 211,840 and 283,301 records, respectively. Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, and Krummholz cover types have lower representation in the dataset, with 35,754, 2,747, 9,493, 17,367, and 20,510 records, respectively. This distribution indicates that the dataset is skewed towards Spruce/Fir and Lodgepole Pine.

Missing Values: Upon checking for missing values, it was found that the dataset had no missing values. This indicates that the dataset was complete, and no further imputation or handling of missing values was required.

Duplicate Rows: Investigation into the presence of duplicate rows revealed that the dataset had no duplicate observations.

Justifications of Choices and Issues Encountered

Manually Specifying Column Names and Handling .data Format:
One of the initial challenges encountered was the absence of a header in the dataset, which was in the .data format. To address this, the choice was made to manually specify the column names during the loading process using the pd.read_csv() method. This ensured that the DataFrame had appropriate column names for further analysis and interpretation. By taking this approach, the dataset was successfully loaded despite the absence of a header.

Converting Categorical Features:
Another important decision was to convert the binary columns representing soil type and wilderness area into categorical variables. This choice was made to align with the requirements of Task 2 and enable meaningful exploration and interpretation of these features as distinct categories.

Task 2: Data Exploration

Method employed for exploring numerical columns:

- The explore_numerical_column() function calculates descriptive statistics, including measures such as mean, standard deviation, minimum, maximum, and quartiles. These statistics offer a summary of the distribution and central tendency of the data.
- The function then generates a histogram to visualise the distribution of the numerical values. This histogram helps in understanding the frequency distribution and potential patterns within the column.
- Additionally, a boxplot is created to provide an overview of the data's dispersion, skewness, and outliers.

Method employed for exploring categorical columns:

- The explore_categorical_column() function first presents the descriptive statistics, which include the count, unique categories, and summary statistics (such as count, unique, top, and frequency).
- Next, a bar plot is generated to visualise the frequency distribution of each category within the column. This allows for a quick understanding of the distribution and identification of dominant categories.

2.1 Exploring Columns

Column 1: "Elevation"

```
Descriptive Statistics:
Mean: 2959.37, Std: 279.98, Min: 1859.00, 25%: 2809.00, 50%: 2996.00, 75%: 3163.00, Max: 3858.00
```

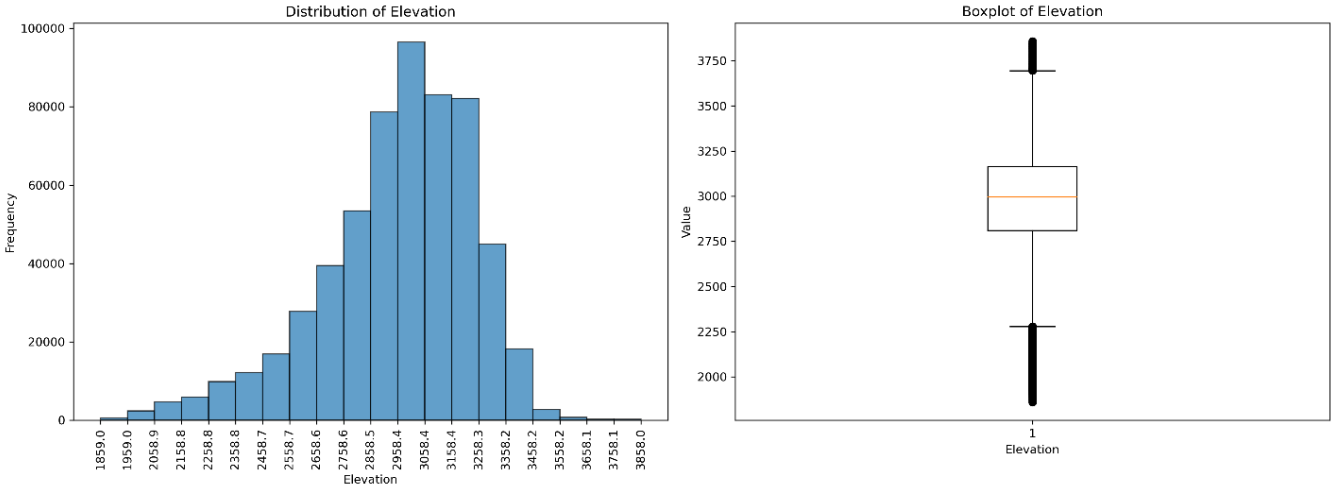

Figure 1: Descriptive Statistics, Histogram, and Boxplot of "Elevation"

Analysis: The majority of the data points have elevations close to the mean, with relatively low variability. The presence of outliers suggests the existence of some data points with exceptionally low and high elevations compared to the majority of the points.

Column 2: "Aspect"

```
Descriptive Statistics:
Mean: 155.66, Std: 111.91, Min: 0.00, 25%: 58.00, 50%: 127.00, 75%: 260.00, Max: 360.00
```

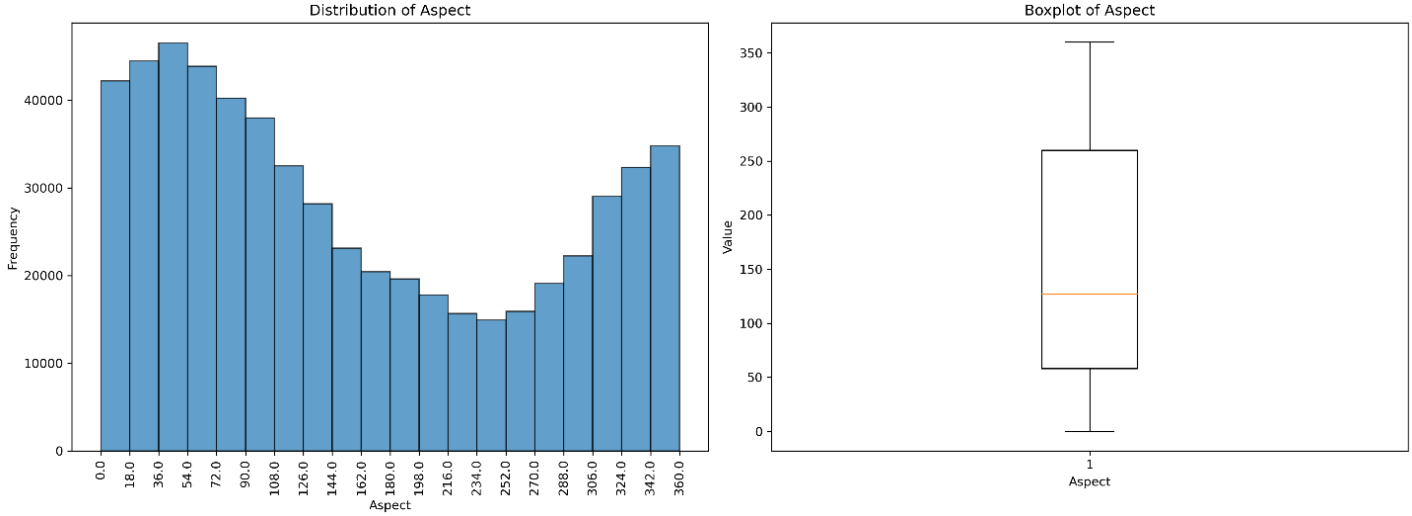

Figure 2: Descriptive Statistics, Histogram, and Boxplot of "Aspect"

Analysis: The "Aspect" values range from 0 to 360 degrees, covering a full circle. This indicates that the dataset includes slopes facing all possible directions. The high frequency of aspect values is observed in the range of 0.0 to 90.0, indicating that a significant portion of the dataset has aspect angles within this range. Since no outliers are present in the boxplot, it indicates that the aspect values are within a reasonable range and there are no extreme or unusual values.

Column 3: "Slope"

```
Descriptive Statistics:
Mean: 14.10, Std: 7.49, Min: 0.00, 25%: 9.00, 50%: 13.00, 75%: 18.00, Max: 66.00
```

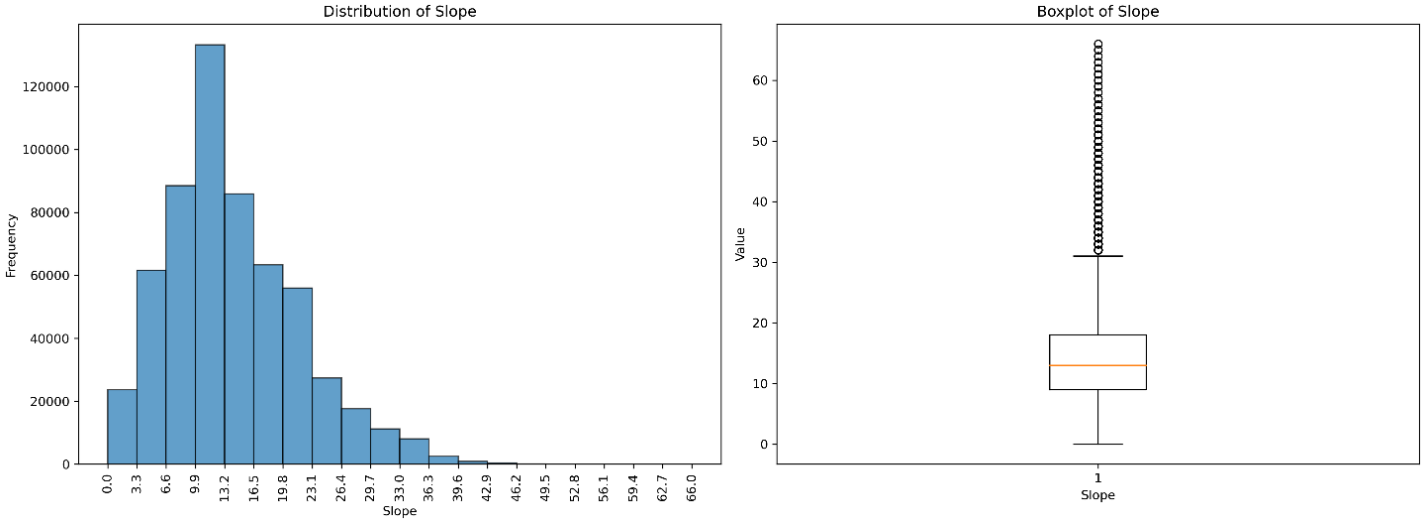

Figure 3: Descriptive Statistics, Histogram, and Boxplot of "Slope"

Analysis: The mean and standard deviation values, as well as the histogram, suggest that the majority of the slope values are relatively moderate, but there is some variability present. The histogram shows a generally decreasing trend in the frequency as the slope steepness increases, with a peak around the 9.9 to 13.2 degrees range. Additionally, the boxplot reveals that outliers are present at the maximum end of the data.

Column 4: "Horizontal_Distance_To_Hydrology"

```
Descriptive Statistics:
Mean: 269.43, Std: 212.55, Min: 0.00, 25%: 108.00, 50%: 218.00, 75%: 384.00, Max: 1397.00
```

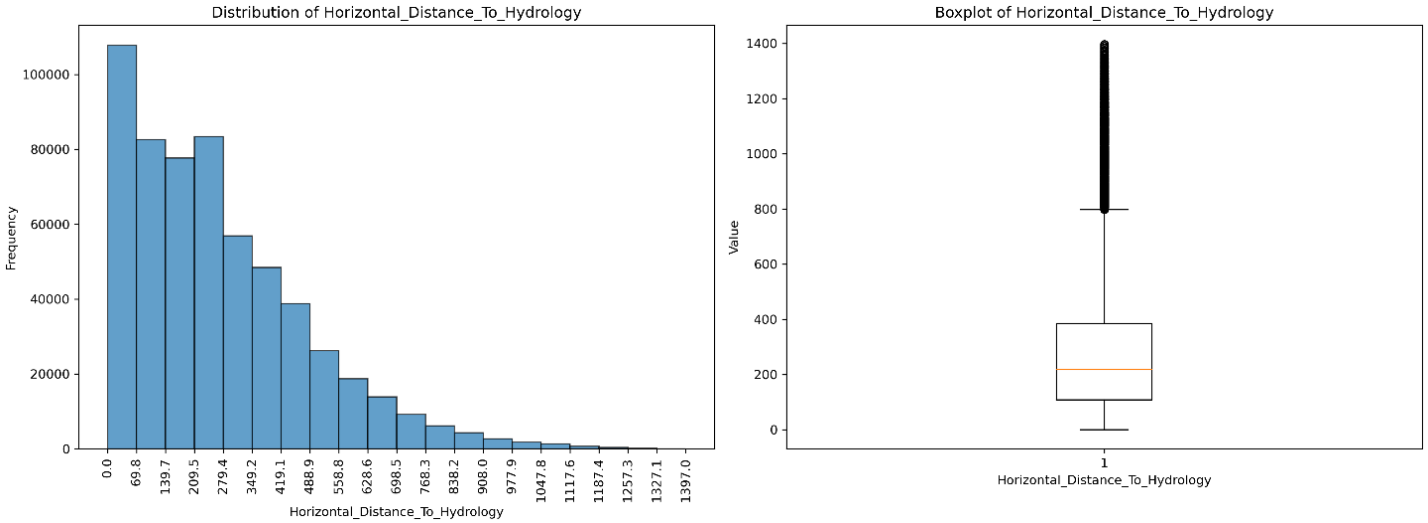

Figure 4: Descriptive Statistics, Histogram, and Boxplot of "Horizontal_Distance_To_Hydrology"

Analysis: The mean and standard deviation values, as well as the histogram, suggest that the majority of the data points are relatively close to water features, but there is a wide distribution. However, there are some outliers that represent locations with greater distances to water sources.

Column 5: "Vertical_Distance_To_Hydrology"

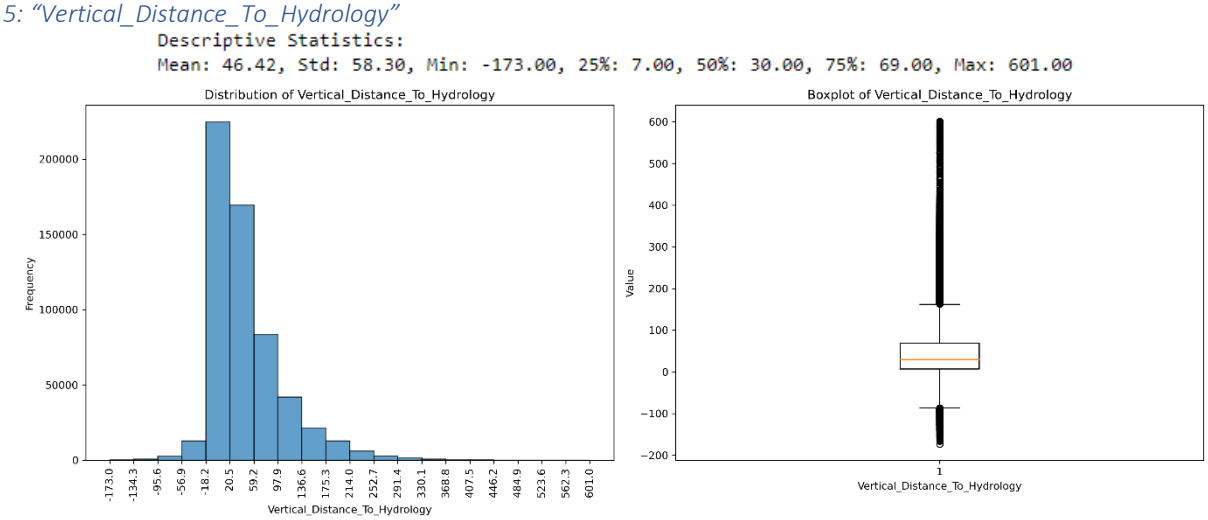

Figure 5: Descriptive Statistics, Histogram, and Boxplot of "Vertical_Distance_To_Hydrology"

Analysis: The highest frequency of data points is observed in the bins with distances ranging from -18.2 to 20.5 units, indicating that the majority of points are close to or at a similar elevation as the water features. The outliers indicated by the boxplot represent data points with extreme vertical distances to water features.

Column 6: "Horizontal_Distance_To_Roadways"

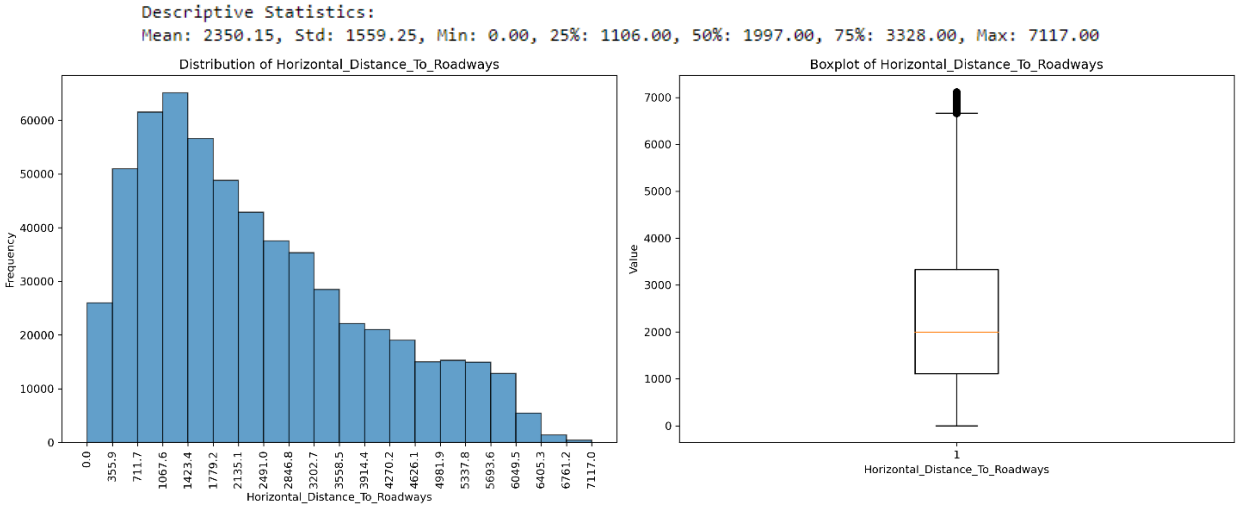

Figure 6: Descriptive Statistics, Histogram, and Boxplot of "Horizontal_Distance_To_Roadways"

Analysis: The mean and standard deviation values, as well as the histogram, suggest that the majority of the data points are at a moderate distance from roadways, but there is considerable variability present. The outliers indicated by the boxplot represent data points that are significantly further away from roadways than the majority of the dataset.

Column 7: "Horizontal_Distance_To_Fire_Points"

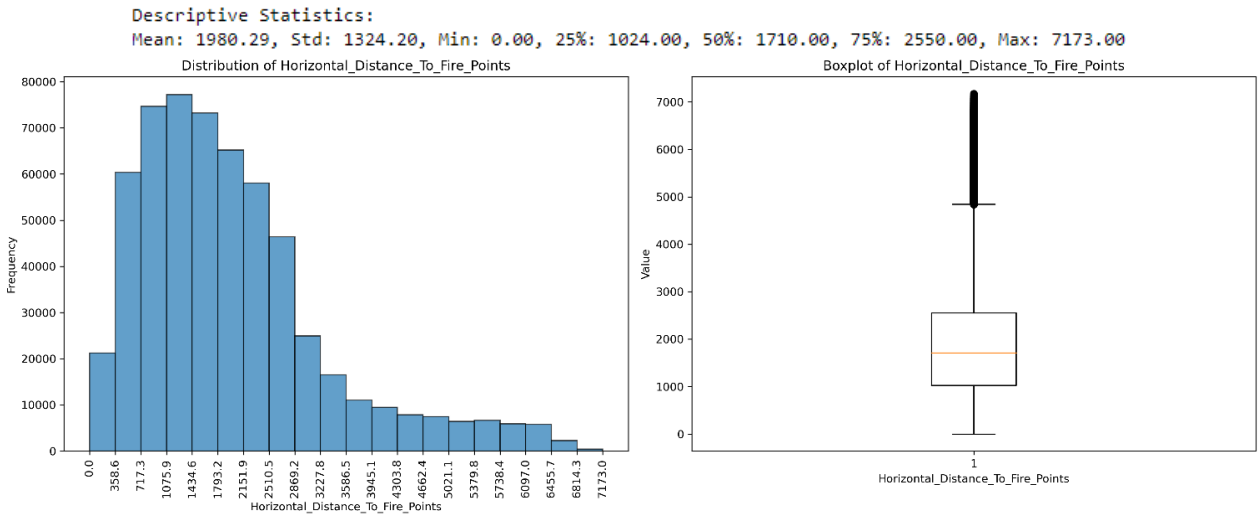

Figure 7: Descriptive Statistics, Histogram, and Boxplot of "Horizontal_Distance_To_Fire_Points"

Analysis: The mean and standard deviation values, as well as the histogram, suggest that the majority of the data points have a moderate distance to the nearest fire point, but there is also considerable variability present in the distances. Outliers detected in the boxplot represent data points with unusually long distances to the nearest fire point.

Column 8: "Hillshade_9am"

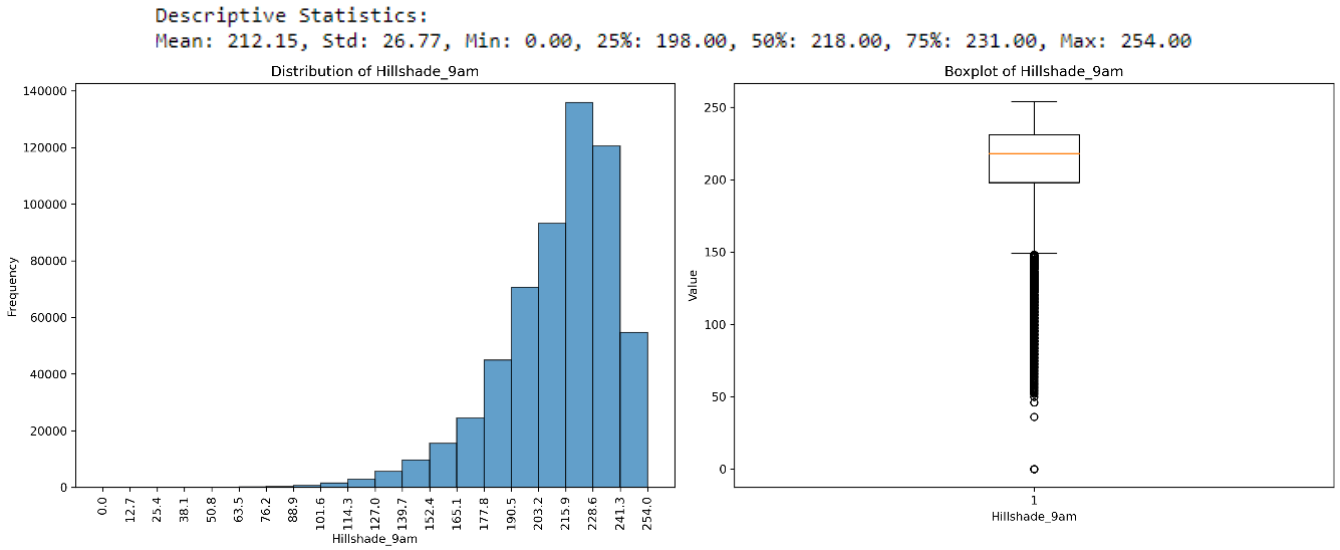

Figure 8: Descriptive Statistics, Histogram, and Boxplot of "Hillshade_9am"

Analysis: The mean and standard deviation values, as well as the histogram, suggest that the majority of the data points have a relatively high hillshade index, indicating a bright terrain during the morning hours. The outliers may indicate areas with shaded or obstructed terrain that block sunlight.

Column 9: "soil_type"

```
Descriptive Statistics:
count      581012
unique         29
top          Como
freq       145417
Name: soil_type, dtype: object
Number of Unique Categories: 29
```

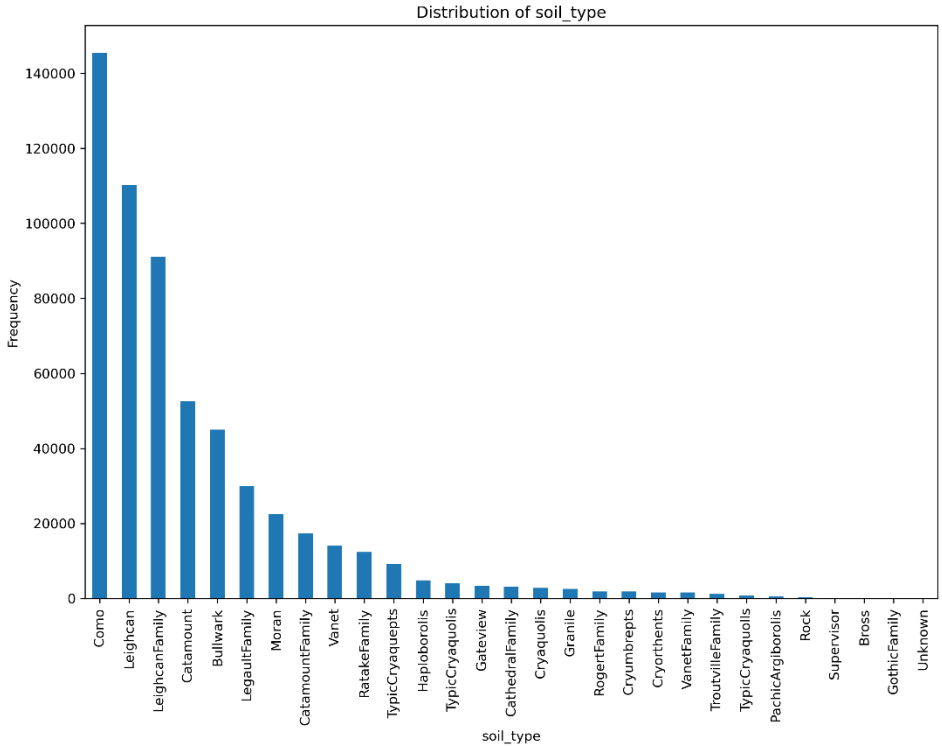

Figure 9: Descriptive Statistics, and Bar Chart of "soil_type"

Analysis: The "soil_type" column represents 29 unique soil types. The key observation from the distribution of soil types in the dataset is that "Como" and "Leighcan" are the most prevalent soil types, while several other soil types have relatively lower frequencies.

Column 10: "wilderness_area"

```
Descriptive Statistics:
count     581012
unique         4
top        Rawah
freq      260796
Name: wilderness_area, dtype: object
Number of Unique Categories: 4
```

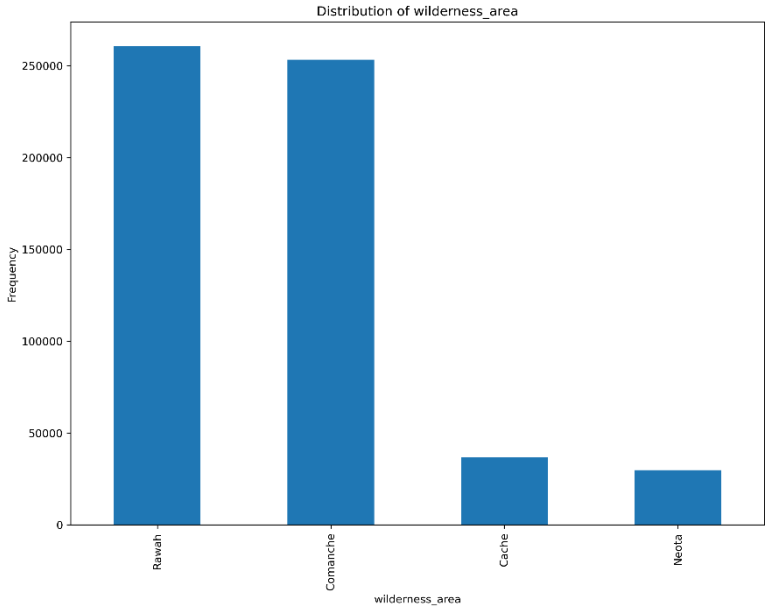

Figure 10: Descriptive Statistics, and Bar Chart of "wilderness_area"

Analysis: The "wilderness_area" column represents 4 unique wilderness areas, with Rawah being the most frequent. The distribution of wilderness areas suggests the presence of distinct natural environments within the study area, and the frequency counts highlight the potential imbalance in the representation of wilderness areas.

2.2 Exploring Pairs of Columns

Pair 1: Elevation vs. Slope

```
Descriptive Statistics:
Correlation between Elevation and Slope: -0.242696639313851
Covariance between Elevation and Slope: -508.83617083620175
Median Elevation: 2996.0
Median Slope: 13.0
```

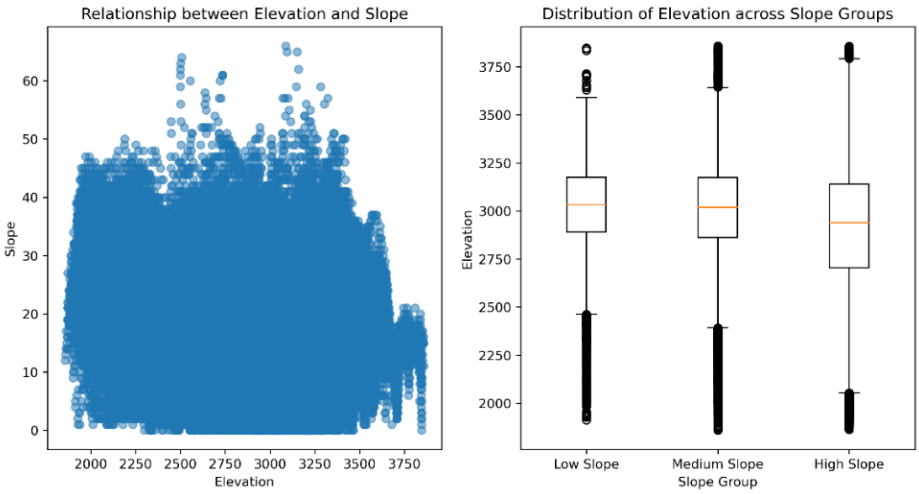

Figure 11: Descriptive Statistics, Scatter Plot, and Grouped Boxplot of "Elevation vs. Slope"

Method: Descriptive statistics, including the correlation coefficient and covariance, were calculated to provide quantitative measures of the relationship. A scatter plot was used to visually demonstrate the relationship between Elevation and Slope. The grouped box plot provided a visual representation of the distribution of Elevation across different slope groups.

Analysis: The correlation coefficient and scatter plot indicated a slight tendency for higher elevations to be associated with lower slopes. In the grouped box plot, the interquartile range (IQR) for high slopes indicated a wider range of Elevation values within the high slope category compared to low and medium slopes. The similar 3rd quartile values across the slope groups suggest that the upper range of Elevation is relatively consistent, while the lower range (1st quartile) is smaller for high slopes.

Pair 2: Elevation vs. Aspect

```
Descriptive Statistics:
Mean Elevation: 2959.365300544567
Mean Aspect: 155.65680743254873
Standard Deviation (Elevation): 279.9847342506334
Standard Deviation (Aspect): 111.91372100329212
Correlation between Elevation and Aspect: 0.015734938290299576
```

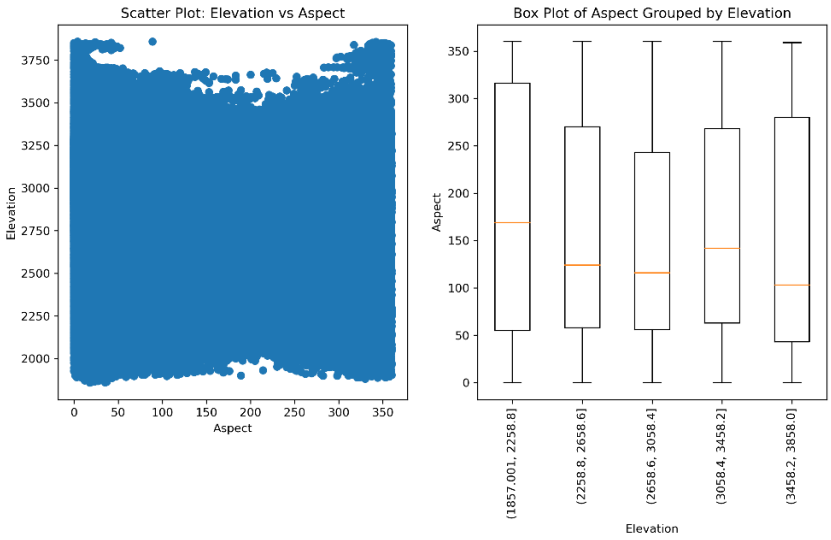

Figure 12: Descriptive Statistics, Scatter Plot, and Grouped Boxplot of "Elevation vs. Apect"

Method: Descriptive statistics, including the mean, standard deviation, correlation coefficient, were calculated to provide quantitative measures of the relationship. The scatter plot visually displays the relationship between elevation and aspect. The grouped box plot provides additional information by grouping elevation into categories and comparing the aspect values within each group.

Analysis: The scatter plot and the correlation coefficient being close to zero implies that changes in elevation are not strongly associated with changes in aspect. The grouped box plot reveals that lower elevation values tend to have relatively higher aspect values. This can be seen from the box plot, where the box corresponding to the lower elevation category (or group) exhibits a higher median aspect value compared to the boxes of higher elevation categories.

Pair 3: Horizontal_Distance_To_Roadways vs. wilderness_area

wilderness_area	count	mean	std	min	25%	50%	75%	max
Cache	36968.0	733.017123	404.869554	0.0	400.0	696.0	1050.0	1776.0
Comanche	253364.0	1937.118438	994.475116	0.0	1180.0	1831.0	2610.0	5463.0
Neota	29884.0	1008.166477	560.678672	30.0	573.0	914.0	1401.5	2632.0
Rawah	260796.0	3134.408055	1743.157038	0.0	1572.0	3155.0	4589.0	7117.0

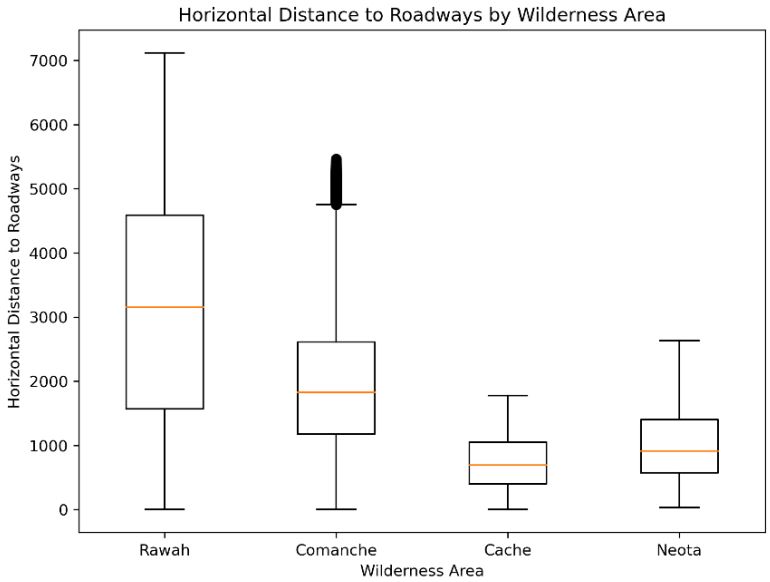

Figure 13: Descriptive Statistics, Grouped Boxplot of "Horizontal_Distance_To_Roadways vs. wilderness_area"

Method: In this exploration, descriptive statistics and graphical visualisations were employed. The boxplot, allowed for a visual comparison of the central tendency, spread, and outliers of horizontal distances to roadways for each wilderness area.

Analysis: Rawah wilderness area tends to have the highest average distance and the widest range of distances, while the Cache wilderness area has the lowest average distance.

Pair 4: Aspect vs. soil_type

soil_type	Bross	Bullwark	Catamount	CatamountFamily	CathedralFamily	Como	Cryaquolis	Cryorthents	Cryumbrepts	Gateview	...	Rock	RogertFam
Aspect_Group													
0-90	NaN	21591.0	15466.0	1963.0	1022.0	61194.0	1057.0	207.0	896.0	1189.0	...	266.0	998
90-180	88.0	5568.0	11323.0	5361.0	962.0	41813.0	513.0	559.0	698.0	977.0	...	31.0	761
180-270	31.0	1957.0	12026.0	7787.0	789.0	14965.0	457.0	696.0	171.0	569.0	...	NaN	50
270-360	NaN	15928.0	13704.0	2320.0	258.0	27445.0	818.0	149.0	126.0	687.0	...	1.0	90

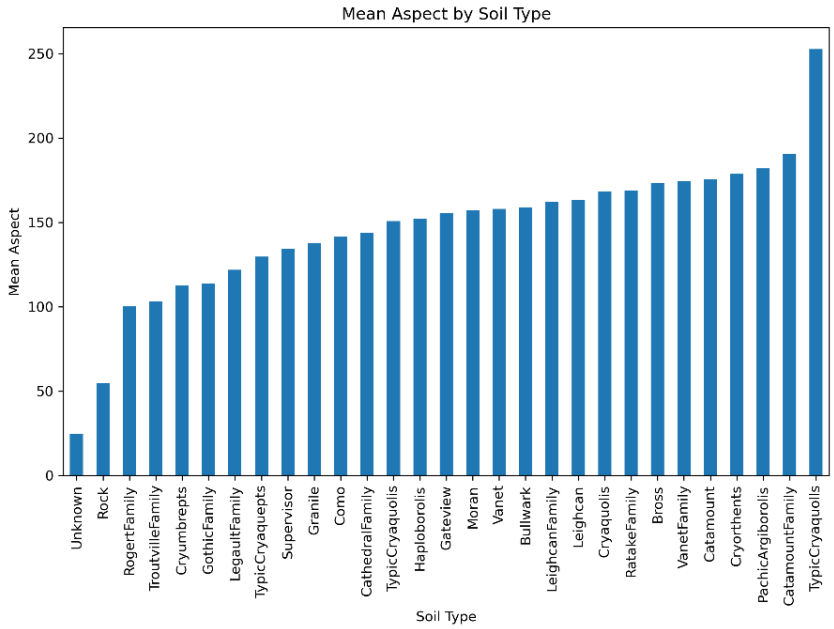

Figure 14: Descriptive Statistics, and Bar Chart of "Horizontal_Distance_To_Roadways vs. wilderness_area"

Method: In this exploration, the analysis involved grouping the data based on aspect ranges and counting the frequency of each soil type within each aspect group. Additionally, the mean aspect was calculated for each soil type, and a bar plot was generated to visualise the mean aspect by soil type.

Analysis: The frequency table shows the distribution of soil types across different aspect groups. For example, in the 0-90 aspect group, the Bullwark soil type has the highest frequency (21,591), followed by Catamount (15,466) and Bullwark (15,466). The mean aspect values for each soil type provide insights into the average aspect associated with different soil types. For instance, the soil type with the highest mean aspect is TypicCryaquolls.

Pair 5: Aspect vs. Hillshade_Noon

```
Descriptive Statistics:
Mean Aspect: 155.65680743254873
Mean Hillshade at Noon: 223.31871630878535
Correlation between Aspect and Hillshade at Noon: 0.33610296426695485
```

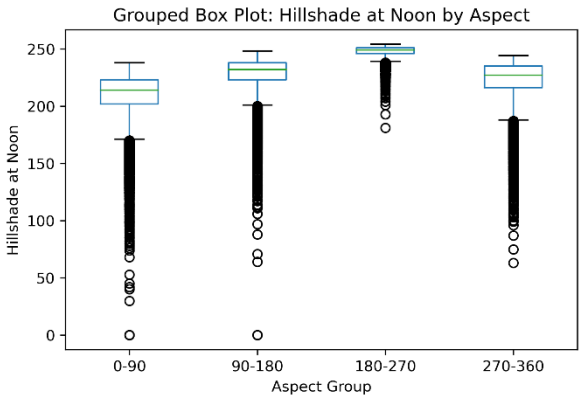

Figure 15: Descriptive Statistics, and Grouped Boxplot of "Aspect vs. Hillshade_Noon"

Method: First, the mean aspect value, the mean hillshade at noon, and the correlation coefficient were computed to assess the strength and direction of the relationship between the two variables. Furthermore, the data were grouped based on aspect ranges, and a grouped box plot was created to visualise the distribution of hillshade values within each aspect group.

Analysis: A moderate positive correlation between aspect and hillshade at noon, suggested that higher aspect values tend to be associated with higher hillshade values during noon. The 0-90 group had the lowest hillshade values, while the 180-270 group had the highest and most stable hillshade values. The 90-180 and 270-360 groups had intermediate hillshade values. These findings indicate that the aspect angle influences the intensity of shade during noon hours, with different aspect ranges exhibiting distinct patterns of hillshade values.

2.3 Investigating the Influence of Slope on Forest Cover Types:

Question: What is the Relationship between Slope and Different Forest Cover Types?

Method: The exploration of the relationship between slope and forest cover types includes creating a scatter plot with distinct markers and colours for each cover type, calculating summary statistics such as minimum, maximum, mean, and median slopes for each cover type.

	Slope			
	min	max	mean	median
Cover_Type				
Aspen	0	51	16.641315	16
Cottonwood/Willow	0	46	18.528941	19
Douglas-fir	0	54	19.048886	19
Krummholz	0	51	14.255924	13
Lodgepole Pine	0	66	13.550499	13
Ponderosa Pine	0	50	20.770208	21
Spruce/Fir	0	56	13.127110	12

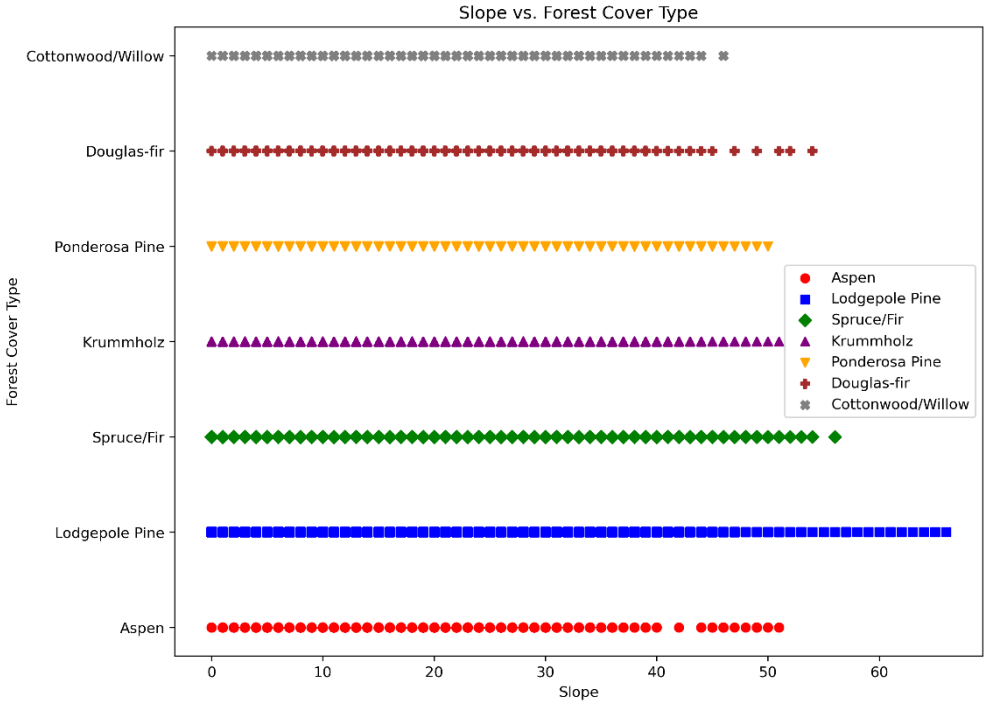

Figure 16: Descriptive Statistics, and Scatter plot of "Slope vs. Cover_Type"

Observations:

- The slope values range from 0 to a maximum of 66 degrees.
- Among the forest cover types, Lodgepole Pine and Krummholz exhibit the widest range of slopes, with a maximum of 66 and 51 degrees, respectively.
- The cover types with the lowest median slope values are Spruce/Fir and Krummholz, with median slopes of 12 and 13 degrees, respectively.
- Ponderosa Pine has the highest median slope value of 21 degrees.
- The mean slope values range from approximately 13 to 20 degrees across different cover types.

Takeaways:

- Slope appears to be a distinguishing factor among different forest cover types. Each cover type shows a specific range of slopes that can be used to differentiate them.
- Forest cover types such as Lodgepole Pine and Krummholz tend to occur across a wide range of slope values, indicating their adaptability to different terrain conditions.
- Spruce/Fir and Krummholz have relatively lower median slope values, suggesting a preference for flatter terrains compared to other cover types.
- Ponderosa Pine stands out with a higher median slope value, indicating a preference for steeper slopes.

Task 3: Data Modelling

3.1: Data Splitting Workflow and Reproducibility Measures

The following sections describe the workflow, the justifications for the choices made, and any encountered issues along with their resolutions.

Data Preparation:

- Initially, the dataset contained binary columns representing two categorical features: soil type and wilderness area. However, since the requirement was to use binary features for KNN and Decision Tree models, the original dataframe was utilised instead of the modified one from Task 2.
- Additionally, the numerical class names were mapped to their respective actual class names using a mapping dictionary. This step was performed to make the class labels more interpretable.

Sampling:

- To make the large dataset more manageable, a 1% sample of the observations was obtained using the train_test_split function from the scikit-learn library. This reduced the data size while still providing a representative subset for further processing.
- Additionally, stratified sampling was employed to address the imbalanced class distribution in the dataset.

Splitting into Three Suites:

- The next step involved splitting the 1% sample into three different suites: Suite1, Suite2, and Suite3, each with a different ratio of training and test data.
  Suite1: 50% for training and 50% for testing.
  Suite2: 60% for training and 40% for testing.
  Suite3: 80% for training and 20% for testing.

Standardisation:

- After the split into training and test sets for each suite, standardisation was performed on the feature data.
- Standardisation (or scaling) is a crucial preprocessing step that transforms the features to have zero mean and unit variance. It helps in mitigating the impact of different scales and units.

Encountered Issues and Resolutions:

- The use of a smaller sample size helped overcome the computational challenges posed by the large dataset (more than half a million rows). By working with a 1% sample, the computational requirements were reduced while still maintaining a representative subset for analysis and modelling.
- Furthermore, the implementation of stratified sampling ensured that the class proportions remained consistent between the training and test sets, enhancing the generalisability of the classification models (Muralidhar, 2021). This approach addressed the potential bias that could arise from imbalanced class distributions, resulting in more reliable and accurate model performance.

Ensuring Reproducibility:

- The random_state parameter was set to a specific value (1 in this case) during the sampling and splitting process. This ensures that the same random seed is used each time the code is run, resulting in consistent splits and reproducible results (Bansal, 2020). By setting the same random seed, the same splits and standardised data can be obtained.

3.2: Model Training and Evaluation

This section describes the workflow and key components involved in training and evaluating machine learning models using the scikit-learn package. It provides justifications for the choices made during the implementation and discusses any encountered issues along with the corresponding solutions.

Workflow Overview:

Model Selection:
Two machine learning models were chosen for training and evaluation: the K-Nearest Neighbors (KNN) classifier and the Decision Tree classifier.

Function Implementation:
Two essential functions were implemented to streamline the process. The train_and_evaluate_model() function was responsible for training the models, calculating evaluation metrics, and returning the results. The print_confusion_matrix() function was used to visualise the confusion matrices.

Model Training and Evaluation:
The train_and_evaluate_model() function was called for each suite and model type. It employed a cross-validation strategy to select the best hyperparameters for the models. Different values of K were tested for the KNN model, while different maximum depth values were explored for the Decision Tree model.

Performance Evaluation:
After training the models with the best hyperparameters, performance evaluation metrics were calculated. These included the confusion matrix, accuracy, precision, recall, and F1 score. The function train_and_evaluate_model() returned the trained models, confusion matrices, and evaluation metrics.

Visualisation:
The print_confusion_matrix() function was used to generate visualisations of the confusion matrices for both the training and test sets. Heatmaps were created, representing the distribution of instances across different categories, with text annotations displaying the actual counts.

Justifications and Issues Faced:

Model Selection: The choice of the KNN and Decision Tree models was based on their suitability for classification tasks and their popularity in the scikit-learn library. These models offer different approaches to classification, allowing for a comparative analysis of their performance.

Parameter Selection: Cross-validation was used to select the optimal hyperparameters for the models. Testing different values of K for the KNN model and different maximum depth values for the Decision Tree model allowed for finding the configurations that yielded the best performance. Cross-validation ensures that the models generalise well to unseen data and mitigates overfitting.

3.3: Model Comparison

Methodology:

The model comparison analysis involved confusion matrices for training and test sets, a summary table, and three key visualisations.

- The confusion matrices were generated for each model on both the training and test sets across the three suites. Figure 17 shows confusion matrices for training and test sets of KNN and DT Models in Suite 1.

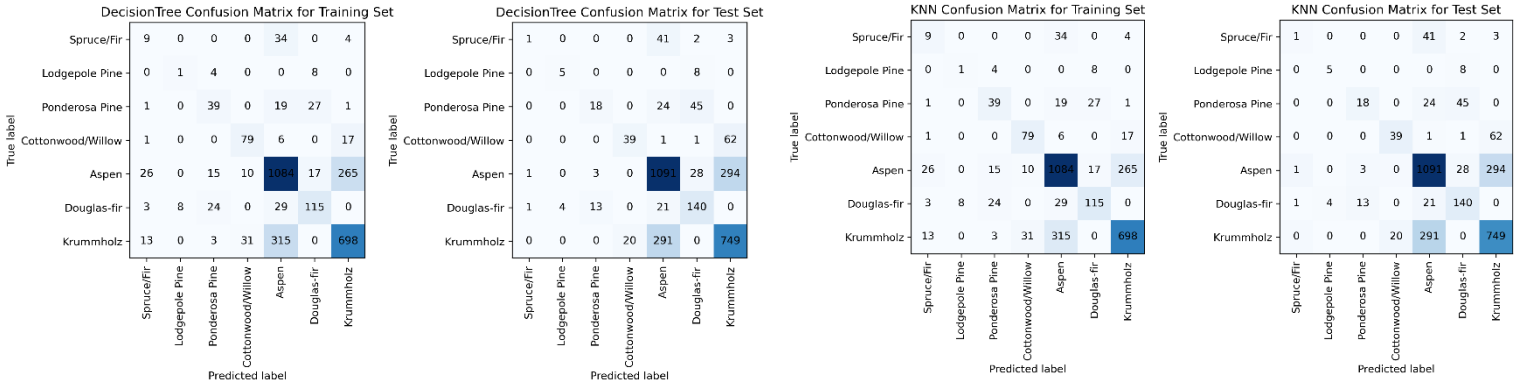

Figure 17: Confusion Matrices for Training and Test Sets of KNN and DT Models in Suite 1

- The summary table presented various performance metrics such as accuracy, precision, recall, and F1 score for both the K-Nearest Neighbors (KNN) and Decision Tree (DT) models across different suites. It provided a comprehensive overview of the models' performance.

	Suite	KNN Train Accuracy	KNN Test Accuracy	KNN Train Precision	KNN Test Precision	KNN Train Recall	KNN Test Recall	KNN Train F1 Score	KNN Test F1 Score	DT Train Accuracy	DT Test Accuracy	DT Train Precision	DT Test Precision	DT Train Recall	DT Test Recall	DT
0	Suite 1	0.844750	0.696834	0.845736	0.696674	0.844750	0.696834	0.844842	0.695984	0.740103	0.701308	0.739573	0.691779	0.740103	0.701308	0.73
1	Suite 2	0.855135	0.707527	0.856251	0.705881	0.855135	0.707527	0.855136	0.705186	0.728916	0.694624	0.727475	0.675213	0.728916	0.694624	0.71
2	Suite 3	0.860155	0.711952	0.861150	0.714042	0.860155	0.711952	0.860253	0.712684	0.766566	0.717971	0.771312	0.719396	0.766566	0.717971	0.75

Figure 18: Summary Table showcasing performance metrics of KNN and DT models across suites

- The bar plots specifically focused on the accuracy metric, visually representing the training and test accuracies of the models in each suite.

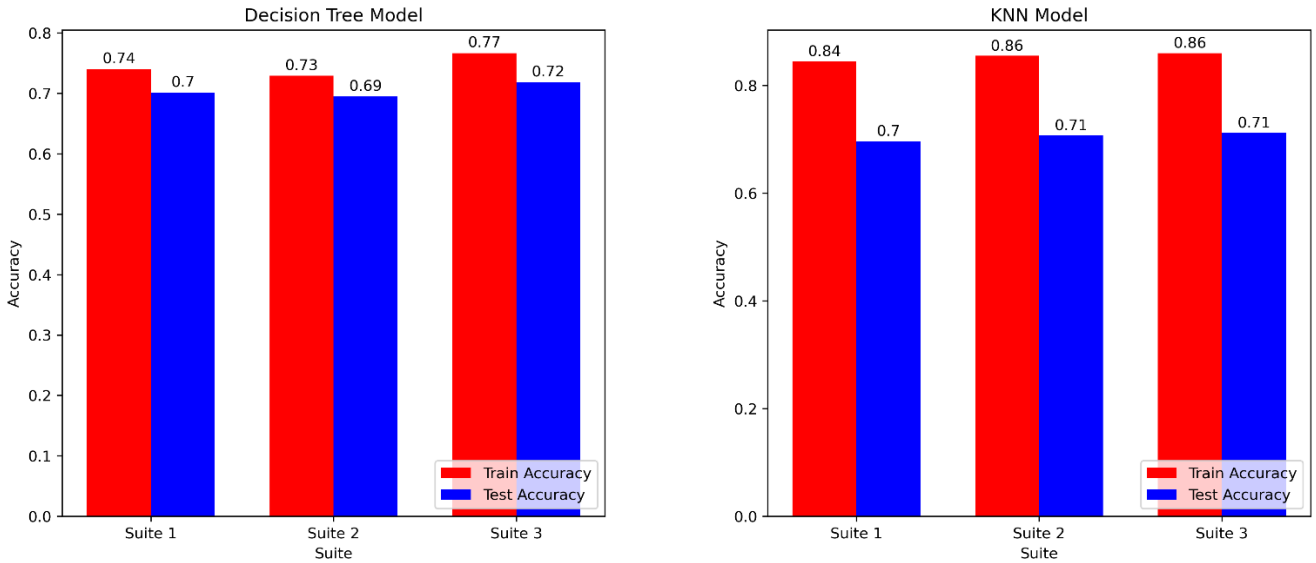

Figure 19: Bar plot comparing the training and test accuracies of KNN and Decision Model across suites

- Additionally, the line plot showcased the trends of precision, recall, and F1 scores for both models across the suites, allowing for a comparison of their performance over different scenarios.

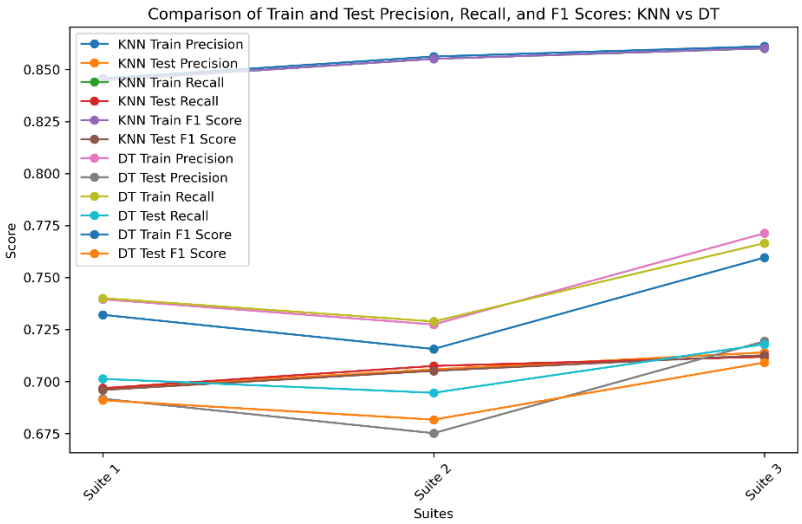

Figure 20: Line Plot comparing precision, recall, and F1 scores of KNN and DT models

Key Observations and Analysis:

Based on the summary table, the bar plots, and the line plot comparing the performance metrics of the KNN and DT models, the following observations can be made:

Performance Metrics:

- Accuracy: The KNN model shows a slightly higher accuracy than the DT model across all suites. Suite 3 has the highest accuracy for both models, indicating that the models perform relatively better on that particular split.

- Precision: The KNN model exhibits slightly higher precision values than the DT model for both training and test sets in all suites.

- Recall: The recall values are slightly higher for the KNN model in all suites.

- F1 Score: The F1 scores follow a similar trend as precision and recall. The KNN model consistently outperforms the DT model in terms of F1 score for both training and test sets across all suites.

Overfitting: Overfitting occurs when a model performs well on the training data but fails to generalise to new, unseen data. Signs of overfitting can include a significant difference between the performance of the model on the training and test sets (Raj, 2022).

- In Suite 1, both the KNN and DT models show similar performance metrics between the training and test sets, suggesting no significant signs of overfitting.

- In all three suites, the KNN model shows slightly higher performance on the training set compared to the test set, indicating a potential slight overfitting. However, the difference is not substantial, suggesting a relatively good generalisation ability of the KNN model.

- In Suite 3, the DT model shows slightly higher performance on the training set compared to the test set, indicating a potential slight overfitting. Similar to the KNN model, the difference is not significant, suggesting a reasonable generalisation ability of the DT model.

Underfitting: Underfitting occurs when a model fails to capture the underlying patterns and complexities in the data, resulting in low performance on both the training and test sets. Based on the performance metrics, there are no clear indications of underfitting for either the KNN or DT models. The models demonstrate reasonable performance on both the training and test sets, with relatively high accuracy, precision, recall, and F1 scores.

Suite-wise Comparison: Suite 3 consistently exhibits higher performance metrics for both models compared to Suite 1 and Suite 2. This indicates that a larger training dataset (80%) in Suite 3 leads to better model performance.

Recommendation:

Based on the observations and analysis, it is recommended to use the KNN model for classification tasks in this scenario. The KNN model consistently outperforms the DT model in terms of accuracy, precision, recall, and F1 scores.

Conclusions

In conclusion, this report provided a comprehensive overview of the project, encompassing problem formulation, data acquisition, data preparation, data exploration, data modelling, and model evaluation.

Task 1 involved formulating the problem and acquiring the dataset. Data preparation steps ensured the dataset's integrity by addressing missing values, duplicate rows, and handling categorical features. The observations revealed no missing values or duplicate observations, and the dataset exhibited class imbalance with a bias towards Spruce/Fir and Lodgepole Pine forest cover types.

Task 2 focused on data exploration, utilising various visualisations and descriptive statistics to understand the distributions, relationships, and patterns within the data. The analysis provided valuable insights into the characteristics of different features, including numerical and categorical columns. Additionally, the investigation into the influence of slope on forest cover types revealed that slope is a distinguishing factor for differentiating forest cover types

Task 3 involved data modelling, where K-Nearest Neighbors (KNN) and Decision Tree (DT) classifiers were trained and evaluated. The analysis showed that the KNN model consistently outperformed the DT model across different suites, showcasing better accuracy, precision, recall, and F1 scores. Additionally, both models exhibited reasonable generalisation abilities without significant signs of overfitting or underfitting.

References

Bansal, J. (2020, November 27). *How to use random seeds effectively*. Medium. https://towardsdatascience.com/how-to-use-random-seeds-effectively-54a4cd855a79

Muralidhar, K. (2021, February 22). *What is stratified cross-validation in machine learning?* Medium. https://towardsdatascience.com/what-is-stratified-cross-validation-in-machine-learning-8844f3e7ae8e

Raj, S. (2022, December 28). *Overfitting and underfitting in machine learning + [example]*. KnowledgeHut. https://www.knowledgehut.com/blog/data-science/overfitting-and-underfitting-in-machine-learning