Data Mining Project

Project Report

Group 3
Gnanasudharsan Ashokumar
Meghana Rao
Meena Periasamy
Nirmalkumar Thirupallikrishnan Kesavan

ashokumar.g@northeastern.edu
rao.meg@northeastern.edu
periasamy.m@northeastern.edu
thirupallikrishnan.n@northeastern.edu

Percentage of Effort Contributed by Student 1: 25
Percentage of Effort Contributed by Student 2: 25
Percentage of Effort Contributed by Student 3: 25
Percentage of Effort Contributed by Student 4: 25

Signature of Student 1: Gnanasudharsan Ashokumar
Signature of Student 2: Meghana Rao
Signature of Student 3:Meena Periasamy
Signature of Student 4:Nirmalkumar Thirupallikrishnan Kesavan

Submission Date: 04/21/2025

Table of Contents

About the Dataset

The dataset used for this project is the Hazardous Fuel Treatment Reduction – Polygon Feature Layer, provided by the U.S. Forest Service through Data.gov. This spatial dataset is a comprehensive geospatial representation of hazardous fuel treatment activities performed across U.S. federal lands. It plays a crucial role in monitoring and evaluating efforts related to wildfire risk reduction and sustainable land management.

This dataset originates from the Forest Service's Natural Resource Manager (NRM) Forest Activity Tracking System (FACTS)—a centralized system used to track forest management activities including fire and fuel treatments, silviculture, and invasive species control. FACTS supports all levels of the Forest Service and is closely integrated with other NRM applications that manage living and non-living natural resource information.

The data layer specifically represents polygonal features of hazardous fuel treatments—activities carried out to manipulate vegetation in a way that reduces the potential intensity, severity, or ecological impact of wildland fires. Treatments may include prescribed burning, mechanical thinning, or other fuel reduction strategies that align with forest management plans. All activities recorded must meet measurable conditions using fire behavior prediction models or fire effects models.

Key features of the dataset:

- Geometry Type: Polygon
- Data Type: Geospatial layer (available as shapefile or geodatabase)
- Scope: National-level coverage across various U.S. Forest Service lands
- Content: Includes information on treatment type, method, year, acreage, and NEPA compliance details.

The dataset is particularly valuable for research in wildfire mitigation, forest ecology, resource planning, and environmental policy analysis. It allows data scientists and decision-makers to explore spatial patterns, evaluate the effectiveness of fuel reduction strategies, and identify priority areas for future interventions.

To access and download the dataset:

- Main dataset page: https://catalog.data.gov/dataset/hazardous-fuel-treatment-reduction-polygon-feature-layer-9c557

Problem Statement

Hazardous fuel treatment is a key strategy used by the U.S. Forest Service to manage wildfire risk. While extensive records are maintained on project plans and activities, the actual size of land treated (GIS_ACRES) often varies depending on multiple factors such as terrain, planning inputs, treatment method, and resource availability. Accurately predicting the number of acres treated in such projects can help improve operational efficiency and planning accuracy.

This project aims to develop a predictive model that estimates the number of acres actually treated in hazardous fuel reduction activities. Using features such as treatment type, method, planning year, land suitability, and cost inputs, the model provides data-driven estimates to support:

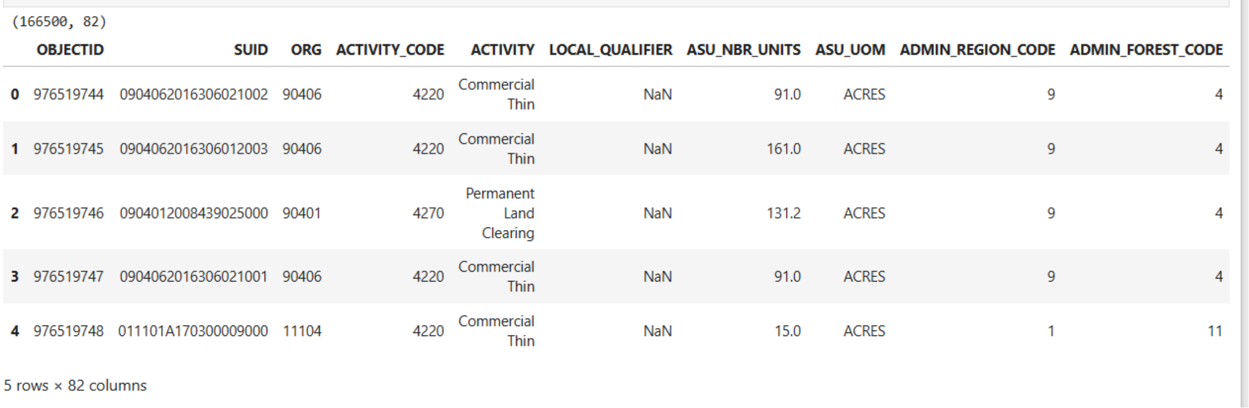
- Better resource allocation for future projects
- Improved forecasting of achievable treatment outcomes
- Identification of factors most influencing treatment scale

By building and comparing machine learning models—namely Random Forest and XGBoost—this analysis provides insights into which features most affect treatment size and offers a tool to aid forest management in strategic decision-making.

Data Preprocessing

Loading and Viewing the Dataset

The data preprocessing began by loading the dataset using the pandas library in Python. The dataset was sourced from a CSV file titled *Hazardous Fuel Treatment Reduction Polygon (Feature_Layer)(1).csv*, which contains detailed information about various fuel treatment activities conducted by the U.S. Forest Service. After loading, the first step was to examine the structure of the dataset using df.shape and df.head(). The dataset consists of 166,500 rows and 82 columns.



```
(166500, 82)
```

	OBJECTID	SUID	ORG	ACTIVITY_CODE	ACTIVITY	LOCAL_QUALIFIER	ASU_NBR_UNITS	ASU_UOM	ADMIN_REGION_CODE	ADMIN_FOREST_CODE
0	976519744	0904062016306021002	90406	4220	Commercial Thin	NaN	91.0	ACRES	9	4
1	976519745	0904062016306012003	90406	4220	Commercial Thin	NaN	161.0	ACRES	9	4
2	976519746	0904012008439025000	90401	4270	Permanent Land Clearing	NaN	131.2	ACRES	9	4
3	976519747	0904062016306021001	90406	4220	Commercial Thin	NaN	91.0	ACRES	9	4
4	976519748	011101A170300009000	11104	4220	Commercial Thin	NaN	15.0	ACRES	1	11

5 rows × 82 columns

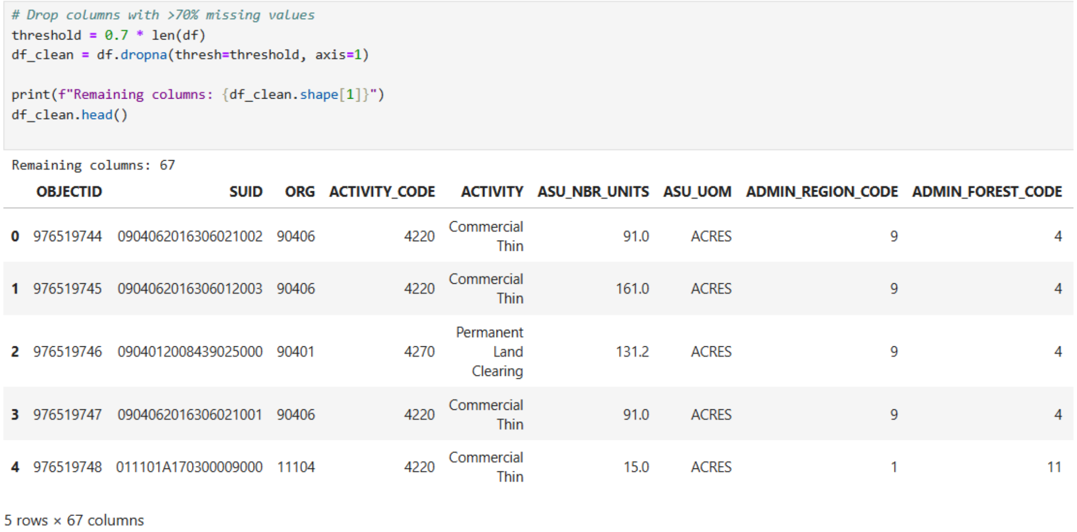
Key Columns and their description:

Column Name	Description
OBJECTID	A unique identifier for each record or activity
SUID	Possibly a unique spatial or submission ID
ORG	Code indicating the organization or district doing the activity
ACTIVITY_CODE	A numerical code identifying the type of treatment
ACTIVITY	Text description of the activity (e.g., "Commercial Thin", "Permanent Land Clearing")
LOCAL_QUALIFIER	Often NaN; might contain extra classification of the activity
ASU_NBR_UNITS	Number of acres treated
ASU_UOM	Unit of measure, which is consistently "ACRES" here
ADMIN_REGION_CODE	Code for the region under which the activity falls
ADMIN_FOREST_CODE	Code for the specific forest unit

Handling Missing Values

After loading the dataset, the next step was to identify and handle missing values. This was done by calculating the number of missing entries in each column using df.isnull().sum(), followed by sorting them in descending order. This step provided a clear picture of which fields had the most missing data. To reduce noise and focus on more reliable data, columns with more than 70% missing values were removed. The threshold was calculated as 70% of the total number of rows, and any column with fewer non-missing values than this was dropped. This was implemented using the df.dropna(thresh=threshold, axis=1) command.

As a result, the number of columns in the dataset was reduced from 82 to 67. Removing high-missing-value fields helped streamline the dataset, making it more manageable and reducing potential errors during modeling.

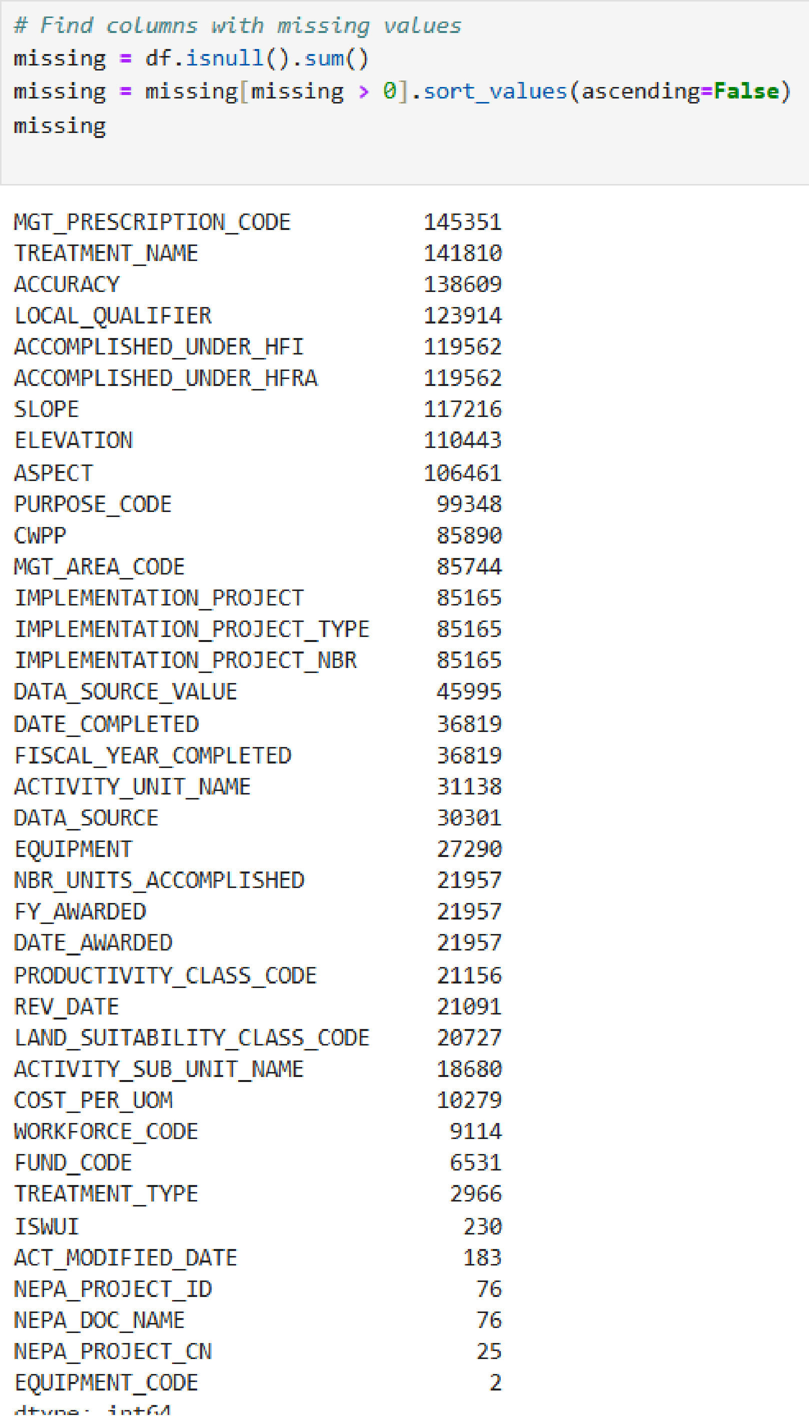


```
# Drop columns with >70% missing values
threshold = 0.7 * len(df)
df_clean = df.dropna(thresh=threshold, axis=1)

print(f"Remaining columns: {df_clean.shape[1]}")
df_clean.head()
```

```
Remaining columns: 67
```

	OBJECTID	SUID	ORG	ACTIVITY_CODE	ACTIVITY	ASU_NBR_UNITS	ASU_UOM	ADMIN_REGION_CODE	ADMIN_FOREST_CODE
0	976519744	0904062016306021002	90406	4220	Commercial Thin	91.0	ACRES	9	4
1	976519745	0904062016306012003	90406	4220	Commercial Thin	161.0	ACRES	9	4
2	976519746	0904012008439025000	90401	4270	Permanent Land Clearing	131.2	ACRES	9	4
3	976519747	0904062016306021001	90406	4220	Commercial Thin	91.0	ACRES	9	4
4	976519748	011101A170300009000	11104	4220	Commercial Thin	15.0	ACRES	1	11

5 rows × 67 columns

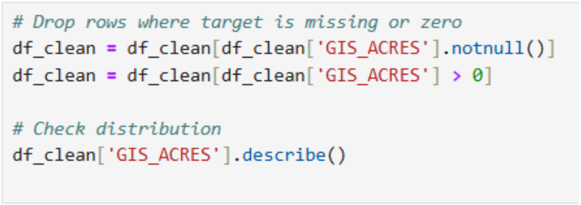


```
# Find columns with missing values
missing = df.isnull().sum()
missing = missing[missing > 0].sort_values(ascending=False)
missing
```

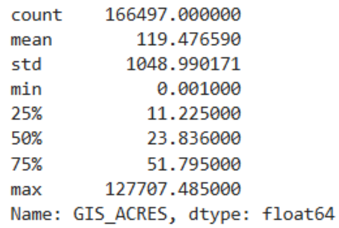
```
MGT_PRESCRIPTION_CODE          145351
TREATMENT_NAME                 141810
ACCURACY                       138609
LOCAL_QUALIFIER                123914
ACCOMPLISHED_UNDER_HFI         119562
ACCOMPLISHED_UNDER_HFRA        119562
SLOPE                          117216
ELEVATION                      110443
ASPECT                         106461
PURPOSE_CODE                    99348
CWPP                            85890
MGT_AREA_CODE                   85744
IMPLEMENTATION_PROJECT          85165
IMPLEMENTATION_PROJECT_TYPE     85165
IMPLEMENTATION_PROJECT_NBR      85165
DATA_SOURCE_VALUE               45995
DATE_COMPLETED                  36819
FISCAL_YEAR_COMPLETED           36819
ACTIVITY_UNIT_NAME              31138
DATA_SOURCE                     30301
EQUIPMENT                       27290
NBR_UNITS_ACCOMPLISHED          21957
FY_AWARDED                      21957
DATE_AWARDED                    21957
PRODUCTIVITY_CLASS_CODE         21156
REV_DATE                        21091
LAND_SUITABILITY_CLASS_CODE     20727
ACTIVITY_SUB_UNIT_NAME          18680
COST_PER_UOM                    10279
WORKFORCE_CODE                   9114
FUND_CODE                        6531
TREATMENT_TYPE                   2966
ISWUI                             230
ACT_MODIFIED_DATE                 183
NEPA_PROJECT_ID                    76
NEPA_DOC_NAME                      76
NEPA_PROJECT_CN                    25
EQUIPMENT_CODE                      2
dtype: int64
```

Refining the dataset

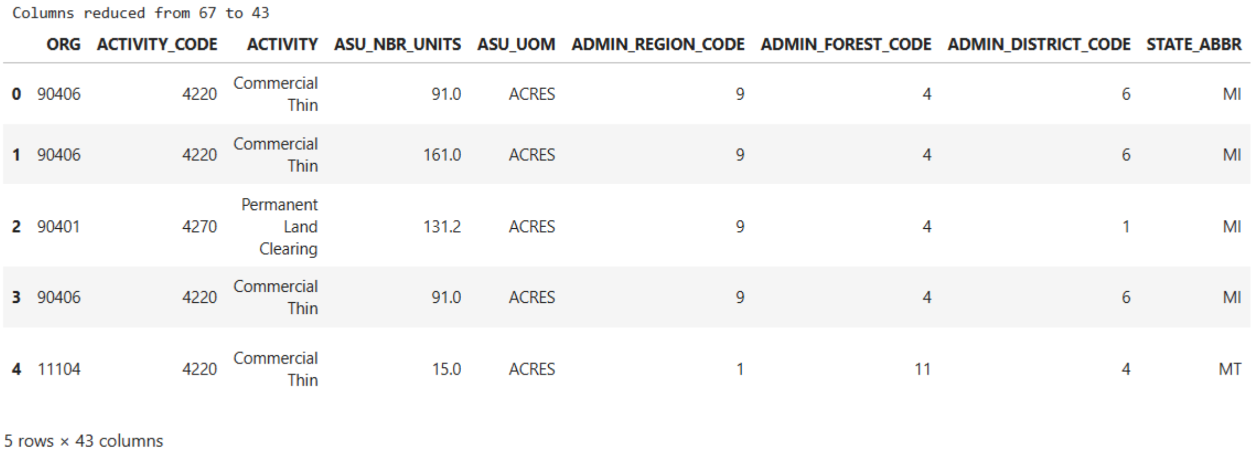
To prepare the dataset for analysis, rows where the target variable GIS_ACRES was missing or equal to zero were removed. This column represents the treated area size, and such rows do not contribute meaningful information to the analysis. Next, a set of columns that were identifiers, timestamps, or unused metadata were dropped to simplify the dataset. These included fields like OBJECTID, SUID, and various date columns that did not offer analytical value. After this step, the dataset was reduced from 67 to 43 columns.



```python
# Drop rows where target is missing or zero
df_clean = df_clean[df_clean['GIS_ACRES'].notnull()]
df_clean = df_clean[df_clean['GIS_ACRES'] > 0]

# Check distribution
df_clean['GIS_ACRES'].describe()
```



```
count    166497.000000
mean        119.476590
std        1048.990171
min           0.001000
25%          11.225000
50%          23.836000
75%          51.795000
max      127707.485000
Name: GIS_ACRES, dtype: float64
```



```
Columns reduced from 67 to 43
```

	ORG	ACTIVITY_CODE	ACTIVITY	ASU_NBR_UNITS	ASU_UOM	ADMIN_REGION_CODE	ADMIN_FOREST_CODE	ADMIN_DISTRICT_CODE	STATE_ABBR
0	90406	4220	Commercial Thin	91.0	ACRES	9	4	6	MI
1	90406	4220	Commercial Thin	161.0	ACRES	9	4	6	MI
2	90401	4270	Permanent Land Clearing	131.2	ACRES	9	4	1	MI
3	90406	4220	Commercial Thin	91.0	ACRES	9	4	6	MI
4	11104	4220	Commercial Thin	15.0	ACRES	1	11	4	MT

5 rows × 43 columns

In the final step, a smaller subset of relevant columns was selected to prepare the dataset for analysis. These columns represent key variables such as treatment area (GIS_ACRES), units of measurement, treatment methods, state, ownership, and land classification codes. To ensure clean input for modeling, rows with missing values in these selected fields were dropped.

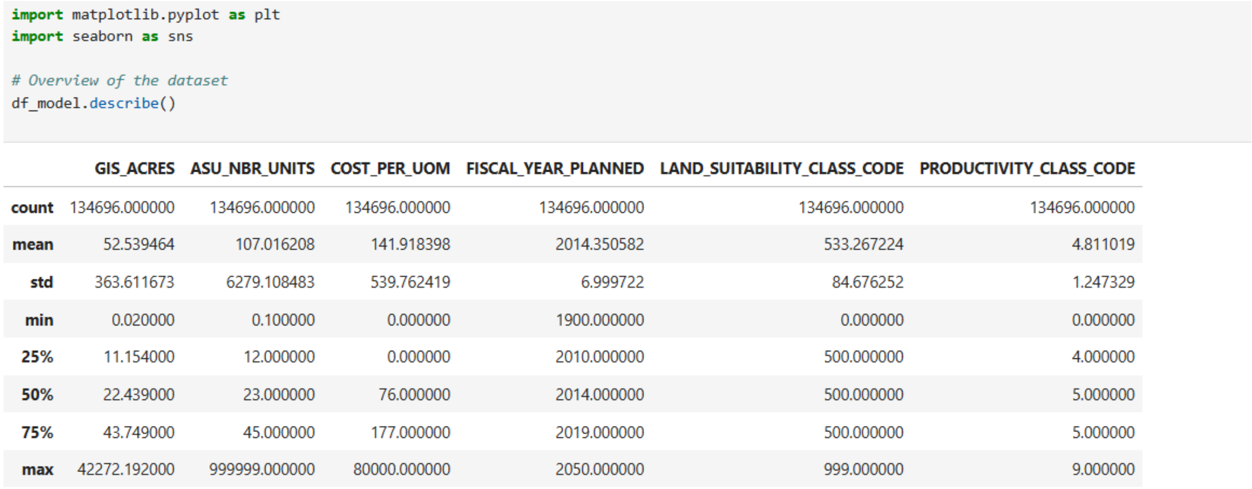
```
Final cleaned dataset shape: (134696, 10)
```

	GIS_ACRES	ASU_NBR_UNITS	COST_PER_UOM	STATE_ABBR	TREATMENT_TYPE	METHOD	FISCAL_YEAR_PLANNED	OWNERSHIP_CODE
0	33.600	91.0	0.0	MI	Thinning	Logging Methods	2020	FS
1	29.642	161.0	0.0	MI	Thinning	Logging Methods	2020	FS
2	131.233	131.2	0.0	MI	Machine Pile	Not Applicable	2014	FS
3	32.758	91.0	0.0	MI	Thinning	Logging Methods	2020	FS
4	10.571	15.0	200.0	MT	Thinning	Mechanical	2007	FS

Data Visualization

Before performing any modeling, an overview of the selected variables was generated to understand their statistical distribution. The summary statistics from the describe() function helped identify patterns and possible issues. For instance, GIS_ACRES, which represents the treated area size, ranges from very small plots (0.02 acres) to extremely large ones (over 42,000 acres), with a median of about 22 acres. This suggests the dataset includes a wide range of treatment scales.

This initial summary helped identify variables with high variance and potential outliers, setting the foundation for visual exploration and deeper insights in the next steps.

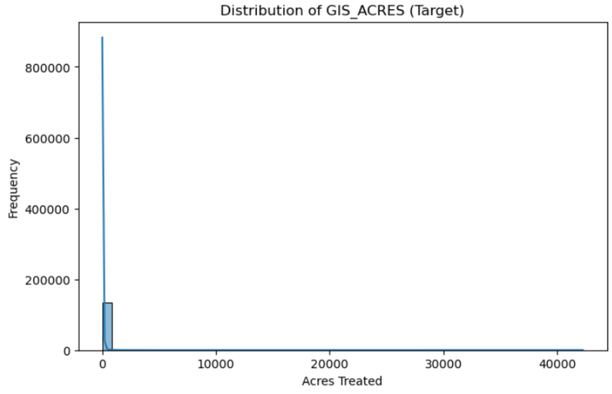


```
import matplotlib.pyplot as plt
import seaborn as sns

# Overview of the dataset
df_model.describe()
```

	GIS_ACRES	ASU_NBR_UNITS	COST_PER_UOM	FISCAL_YEAR_PLANNED	LAND_SUITABILITY_CLASS_CODE	PRODUCTIVITY_CLASS_CODE
count	134696.000000	134696.000000	134696.000000	134696.000000	134696.000000	134696.000000
mean	52.539464	107.016208	141.918398	2014.350582	533.267224	4.811019
std	363.611673	6279.108483	539.762419	6.999722	84.676252	1.247329
min	0.020000	0.100000	0.000000	1900.000000	0.000000	0.000000
25%	11.154000	12.000000	0.000000	2010.000000	500.000000	4.000000
50%	22.439000	23.000000	76.000000	2014.000000	500.000000	5.000000
75%	43.749000	45.000000	177.000000	2019.000000	500.000000	5.000000
max	42272.192000	999999.000000	80000.000000	2050.000000	999.000000	9.000000

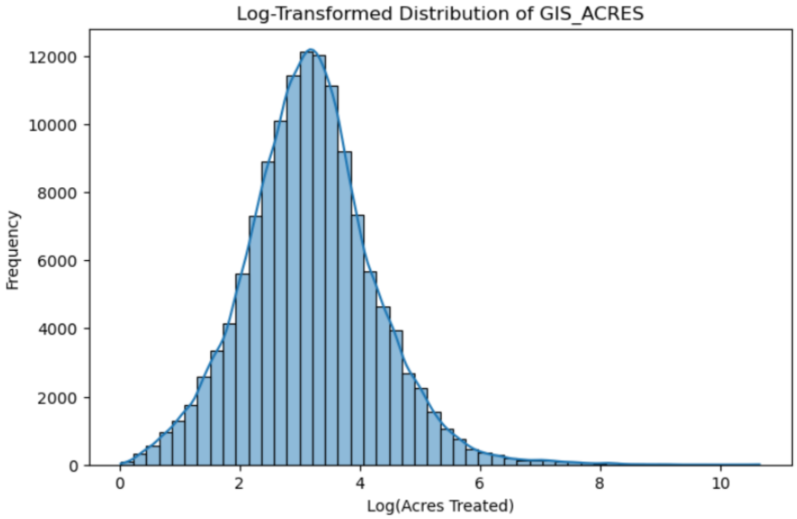
Distribution of Treatment Area (GIS_ACRES)

A histogram was used to visualize the distribution of GIS_ACRES, which represents the size of treated land areas. The plot shows a right-skewed distribution, where most activities involve small-scale treatments, often under 50 acres. A few records show very large treatment sizes, some exceeding 40,000 acres. These outliers may affect certain analyses and could require transformation or separate handling in later steps.



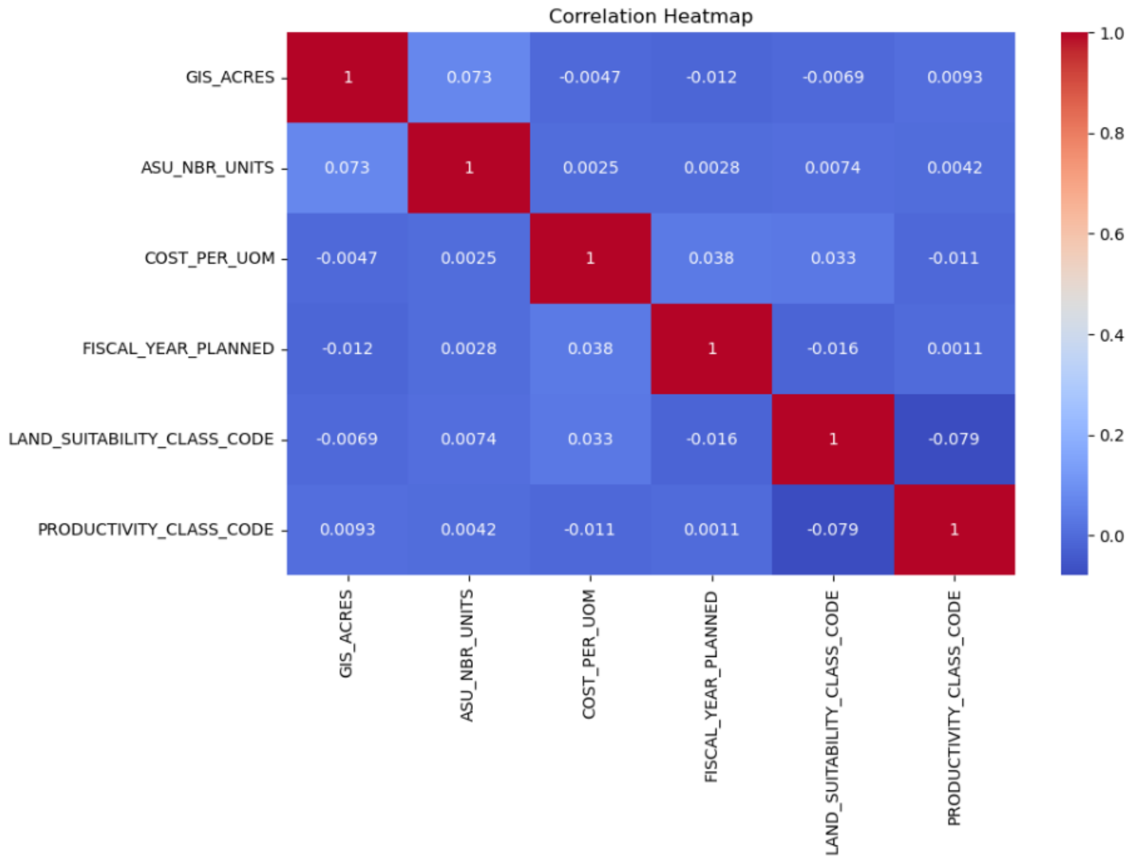
Log-Transformed Target Variable

To address the skewed distribution of GIS_ACRES, a log transformation was applied using log1p, which safely handles zero values. The resulting distribution of LOG_GIS_ACRES is more balanced and closer to a normal shape. Most records are now centered around a log value of 3 to 4, with fewer extreme values at either end. This transformation helps reduce the influence of outliers and improves model performance for techniques sensitive to scale, such as linear regression or clustering.



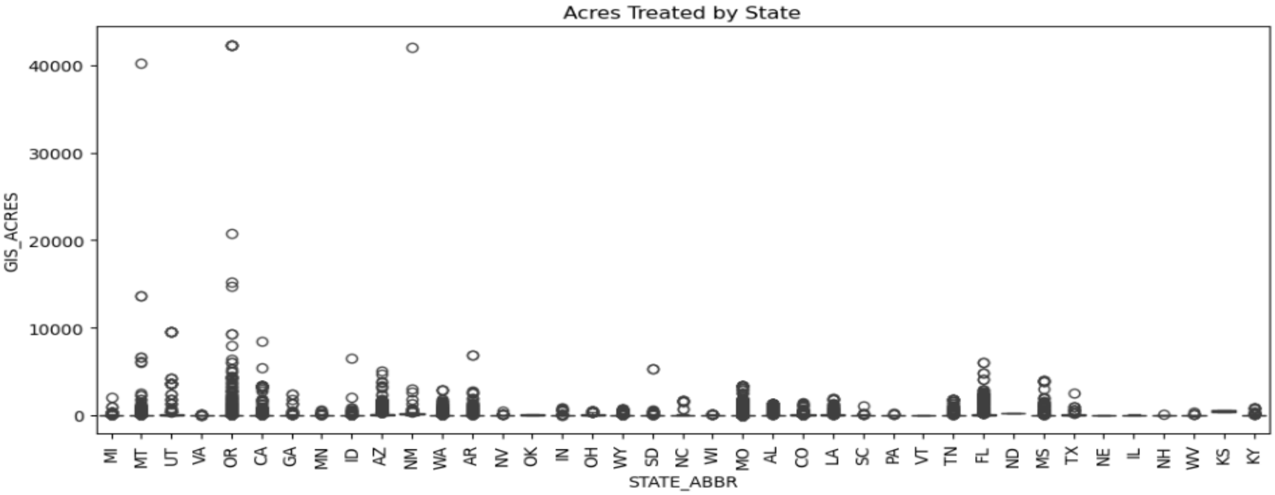
Correlation Analysis

A heatmap was created to examine correlations between the numerical features in the dataset. The matrix shows that most variables have very low correlation with each other, including with the target variable GIS_ACRES. The highest correlation observed is between GIS_ACRES and ASU_NBR_UNITS (0.073), though this is still quite weak. Other variables like COST_PER_UOM, FISCAL_YEAR_PLANNED, and land or productivity classification codes also show minimal linear relationships.



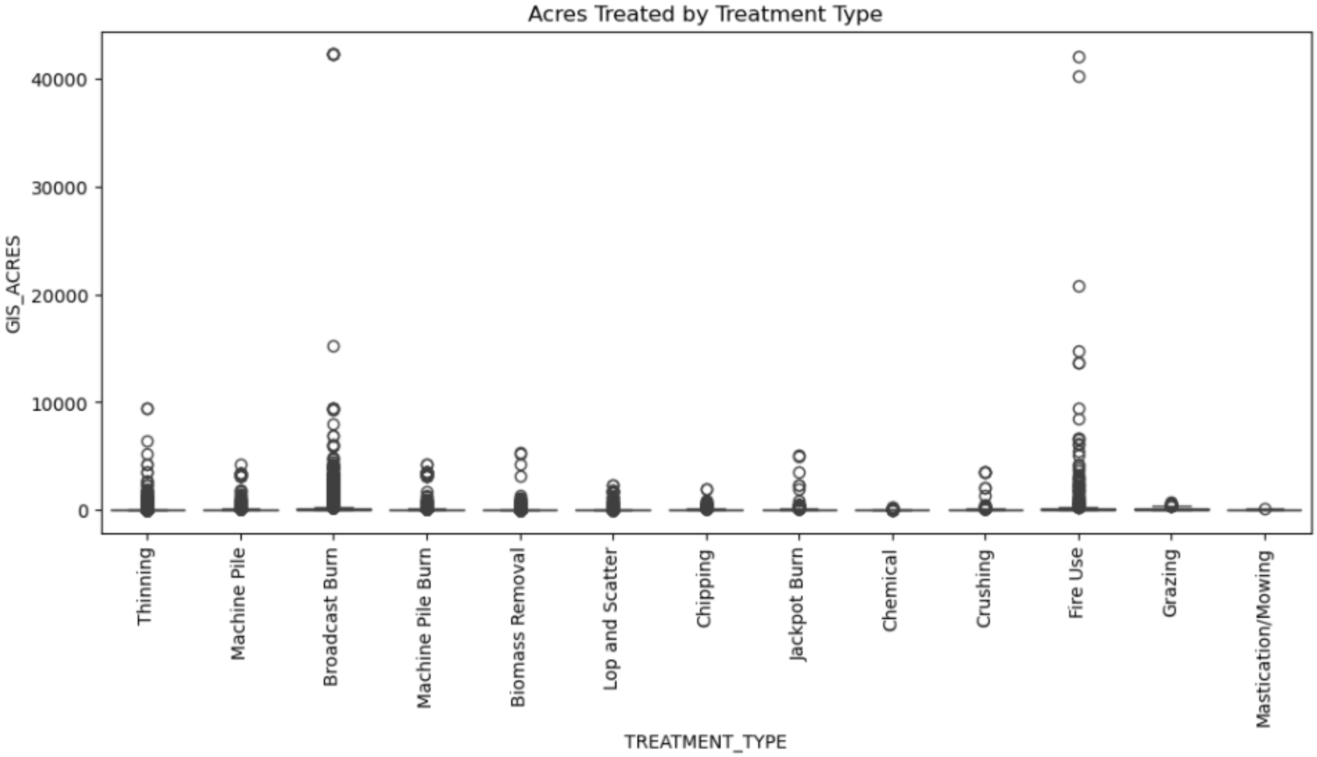
Acres Treated by State

A boxplot was used to compare the distribution of GIS_ACRES across different states. This helped identify how treatment sizes vary geographically. Most states show a high number of smaller-scale treatments, with a few large outliers extending beyond 10,000 acres. States like Oregon, California, and Virginia show some of the largest outliers, suggesting that large-scale treatments are more common in these regions.



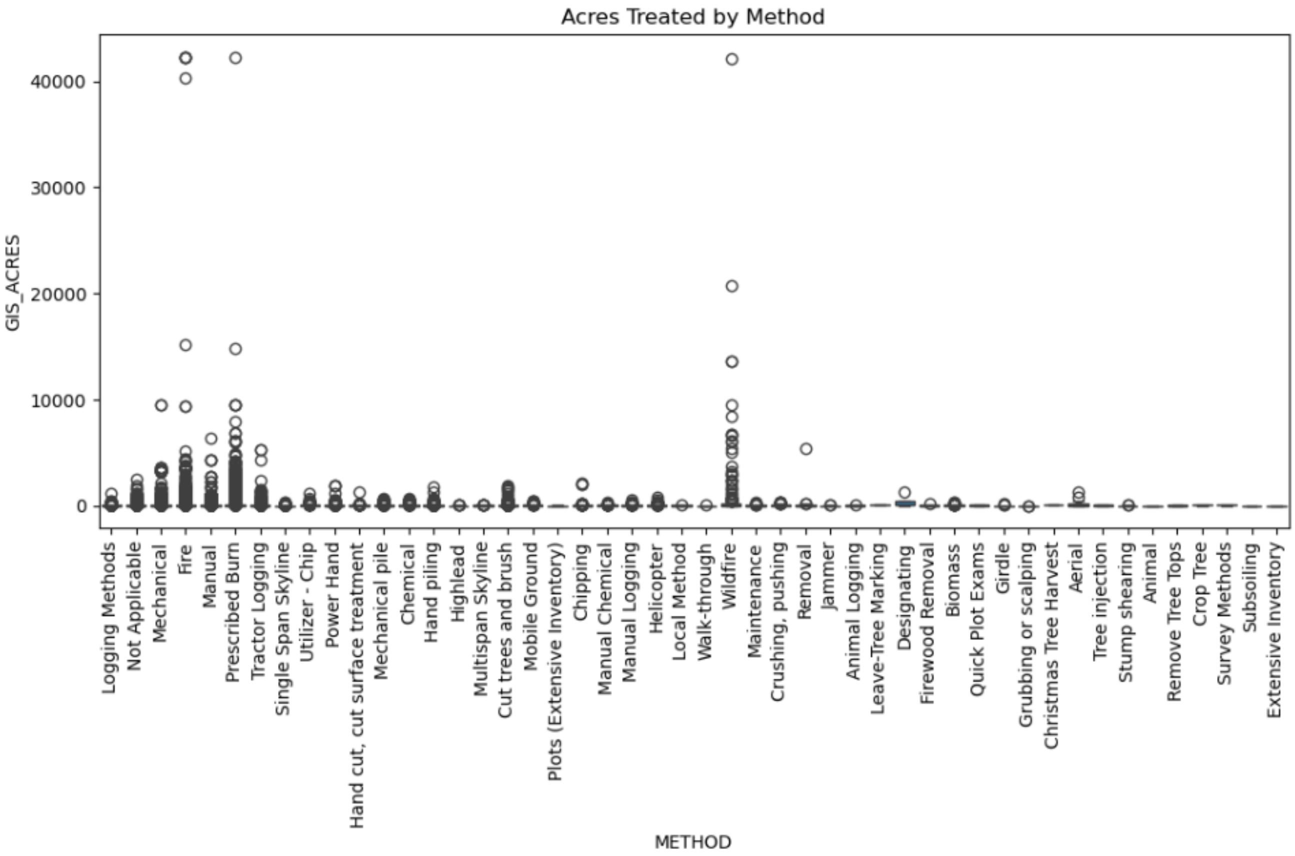
Acres Treated by Treatment Type

A boxplot was used to compare the size of treated areas (GIS_ACRES) across different treatment types. Most types, such as thinning, pile burning, and chipping, show consistent patterns of small-to-moderate scale treatments. However, some treatment types—especially "Broadcast Burn" and "Fire Use"—include extreme outliers where more than 40,000 acres were treated.



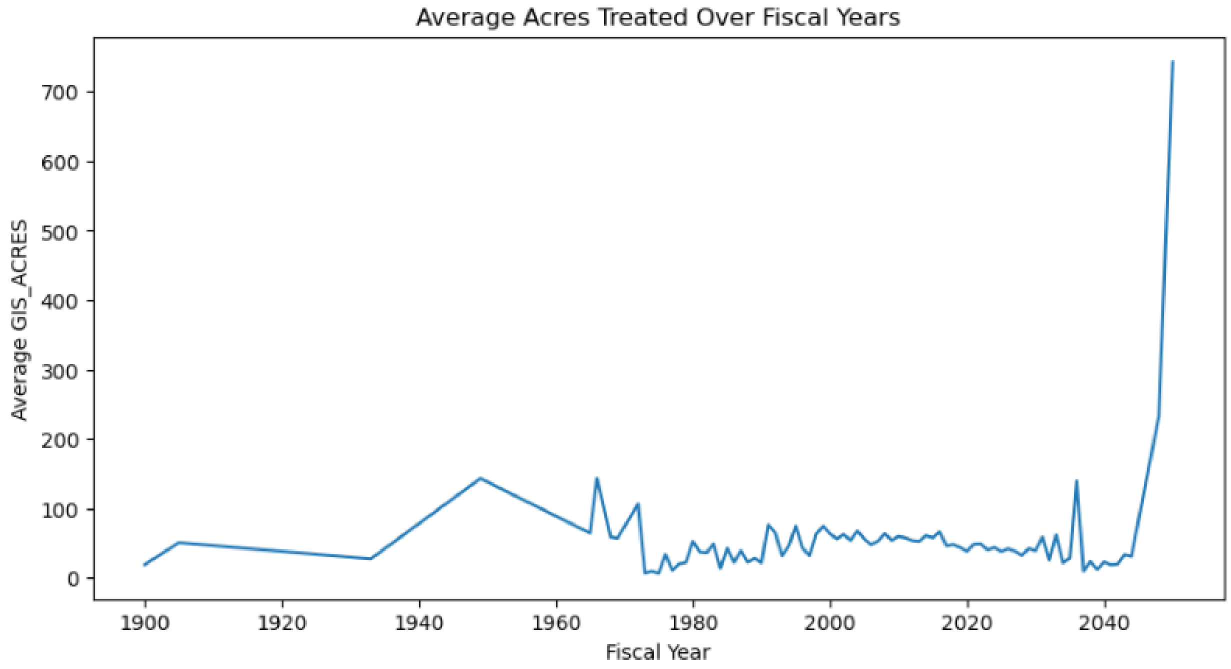
Acres Treated by Method

This boxplot compares the acres treated (GIS_ACRES) across different treatment methods. The most common methods, such as “Logging Methods,” “Mechanical,” and “Manual,” show a wide range in treatment sizes, including several large outliers above 10,000 acres. Fire-related methods like “Prescribed Burn” also show high variability. On the other hand, many methods have consistently small treatment areas with minimal variation, suggesting their use in more targeted or localized applications.



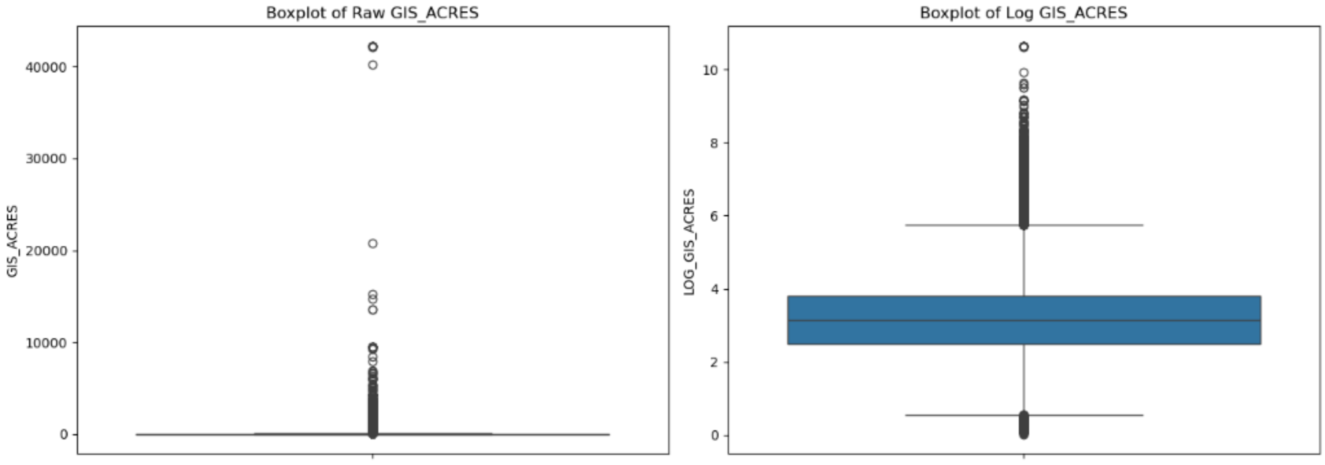
Average Acres Treated Over Time

A line plot was created to observe how the average treatment size (GIS_ACRES) changed over fiscal years. The graph shows some fluctuations over time, with relatively stable averages from the mid-1980s to early 2000s.



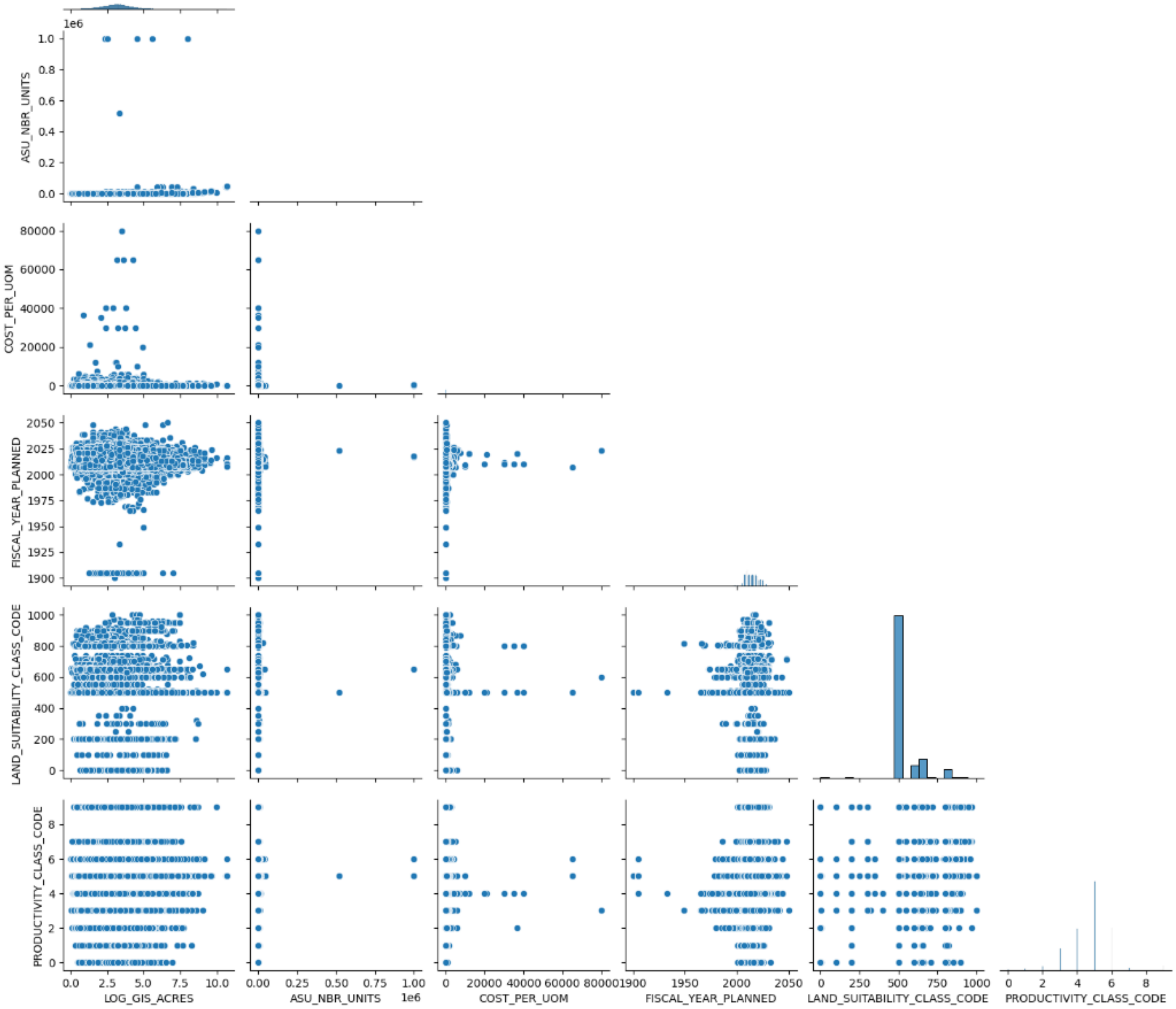
Comparison of Raw and Log-Transformed Treatment Areas

Two boxplots were used to compare the original and log-transformed values of GIS_ACRES. The raw values show many extreme outliers, which stretch the scale and compress the rest of the data. In contrast, the log-transformed version is more compressed, with the bulk of values falling within a tighter range. While some outliers remain, the log scale helps reduce their influence and brings the distribution closer to normal.



Pairwise Relationships

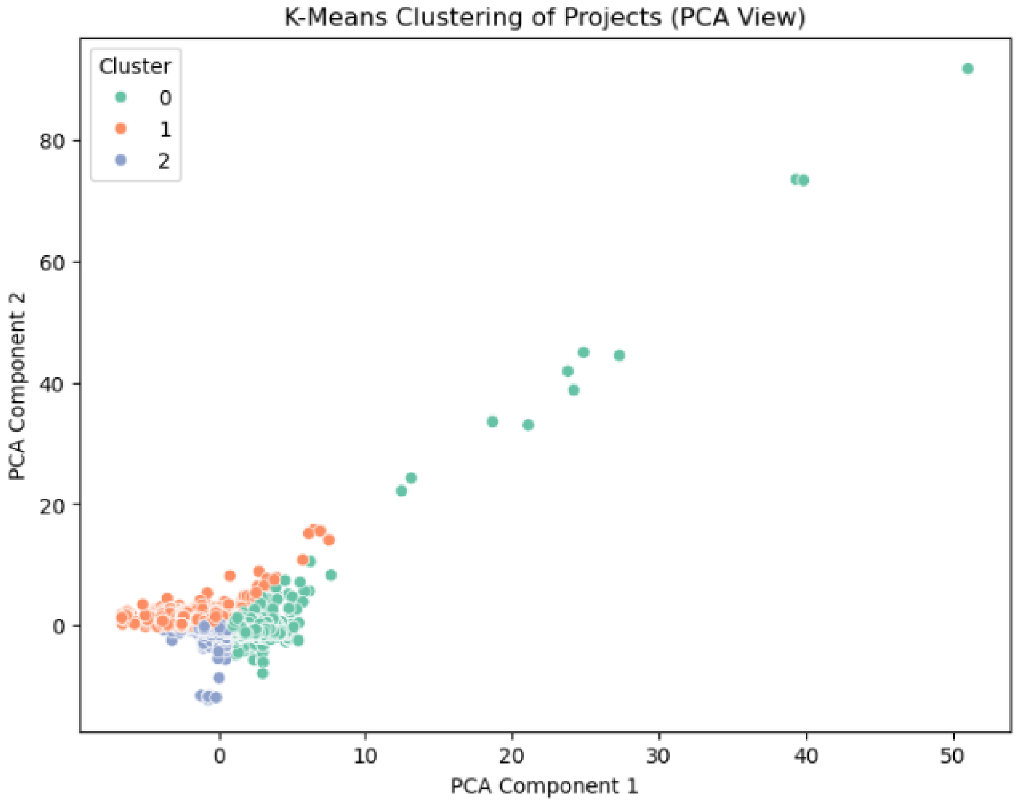
A pairplot was created to explore relationships between the main numerical variables, using the log-transformed GIS_ACRES as the target. Most variable pairs show weak or no clear linear trend, which aligns with earlier correlation analysis. A few patterns are visible—for example, a weak positive association between LOG_GIS_ACRES and ASU_NBR_UNITS. Other features like FISCAL_YEAR_PLANNED and COST_PER_UOM show a wide spread without a strong directional trend.



Clustering and Dimensionality Reduction

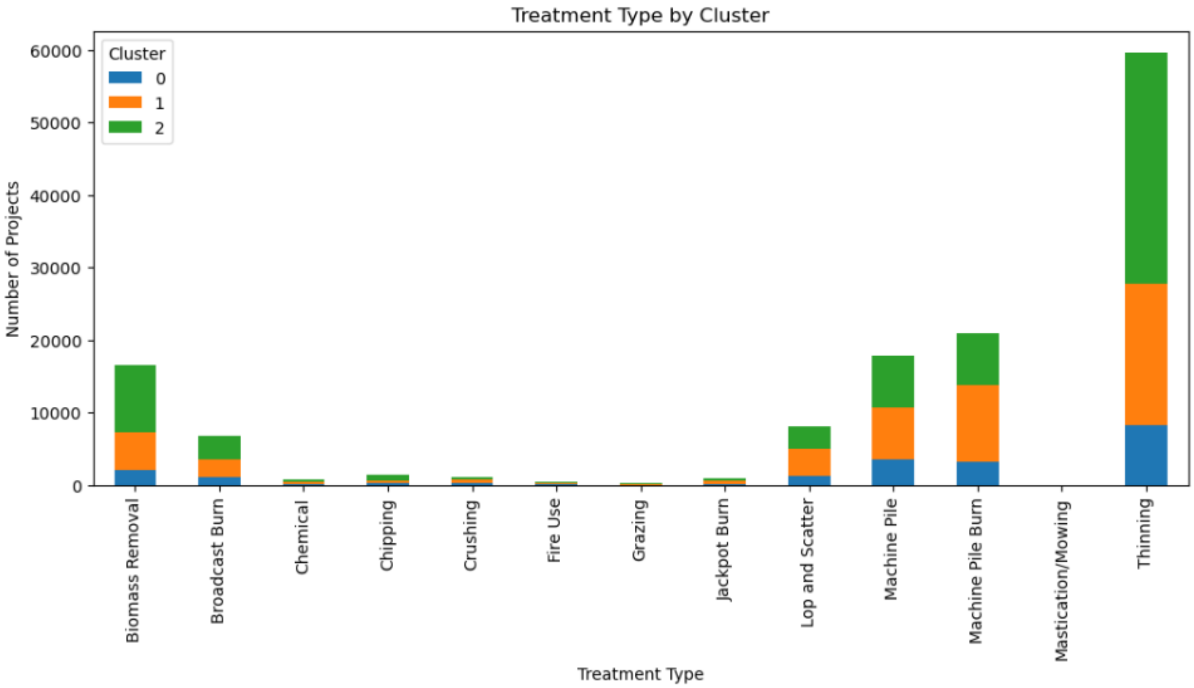
K-Means clustering was applied to group similar fuel treatment records based on five numerical features. These include treatment cost, planning year, and land suitability metrics. Before clustering, the data was standardized to bring all features to a similar scale. Since high-dimensional data is difficult to visualize, Principal Component Analysis (PCA) was used to reduce the features to two components for plotting.

The scatter plot shows the K-Means results with three clusters, revealing how projects are grouped in the reduced feature space. Although PCA simplifies the relationships, the clusters suggest that certain combinations of features—such as high cost or specific land classifications—may drive similarity between treatments.



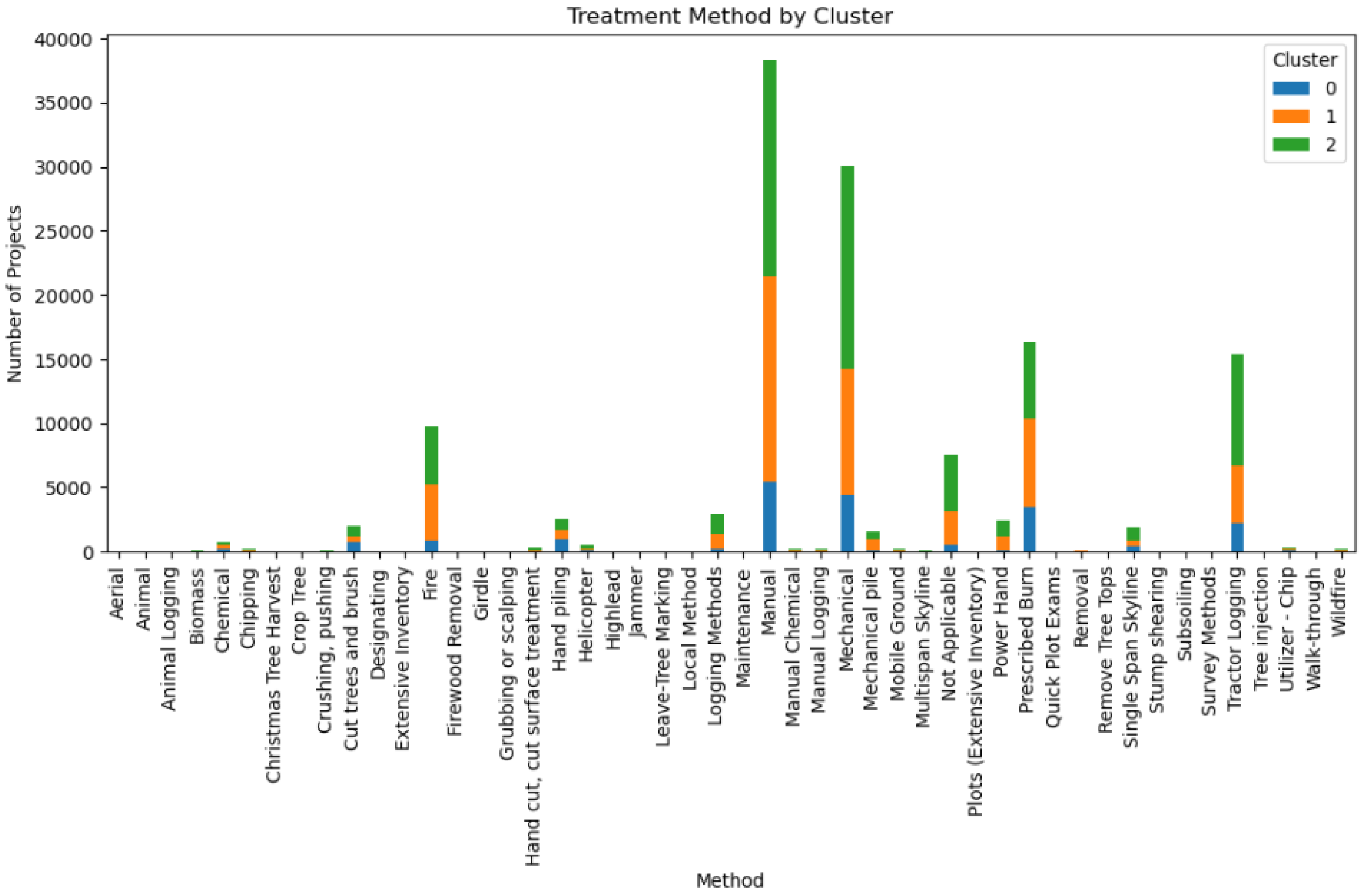
Treatment Type Distribution by Cluster

After assigning cluster labels to each record, a stacked bar chart was created to compare how treatment types are distributed across the three clusters. The chart shows clear differences in how treatment types are grouped. For example, “Thinning” and “Biomass Removal” dominate Cluster 2, suggesting larger or more resource-intensive operations. In contrast, Cluster 0 and Cluster 1 show more balanced distributions among treatments like “Machine Pile,” “Lop and Scatter,” and “Broadcast Burn.”



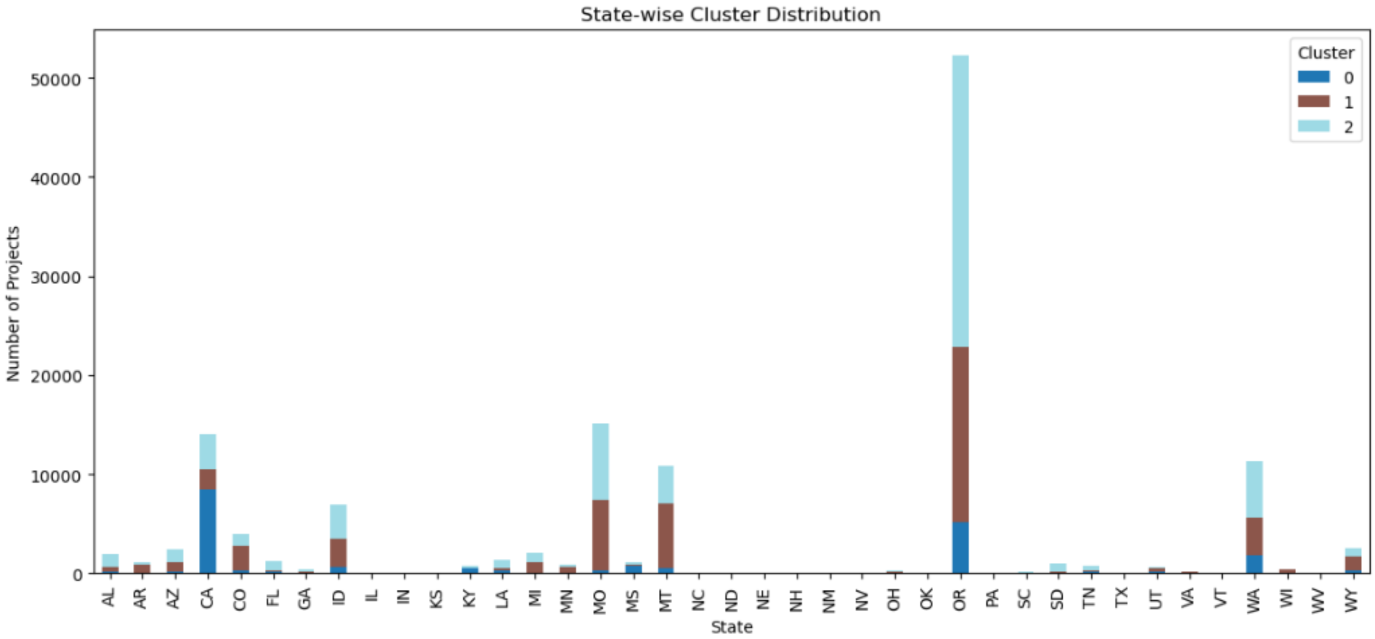
Treatment Method Distribution by Cluster

This stacked bar chart displays how different treatment methods are distributed across the three clusters. Some methods, like “Manual,” “Mechanical,” and “Logging Methods,” are spread across all clusters but appear most heavily in Cluster 2, which likely includes large-scale or high-intensity treatments. Other methods such as “Prescribed Burn,” “Chipping,” and “Maintenance” show a more even or moderate presence across clusters. The method “Fire Use” also stands out as highly concentrated in Cluster 2. This analysis shows that certain treatment methods are more closely tied to specific project types.



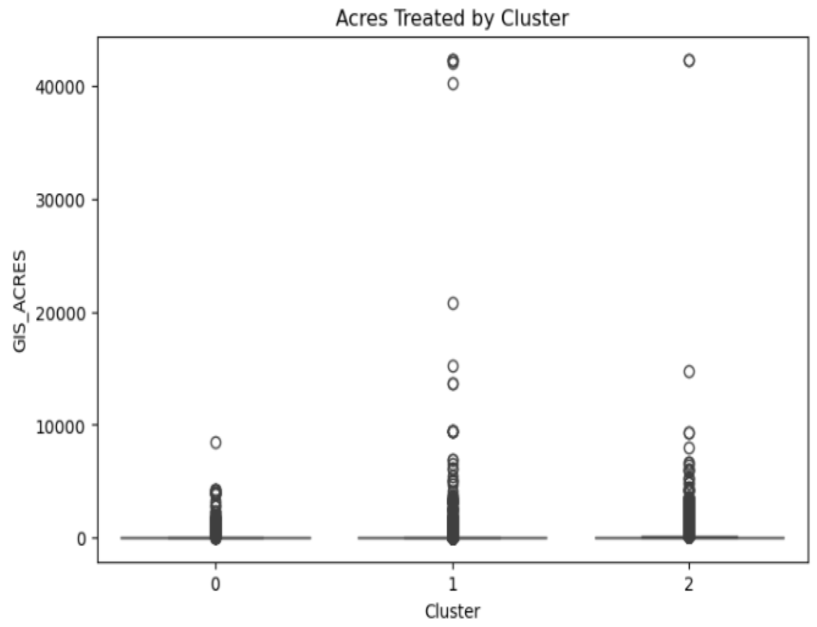
State-wise Cluster Distribution

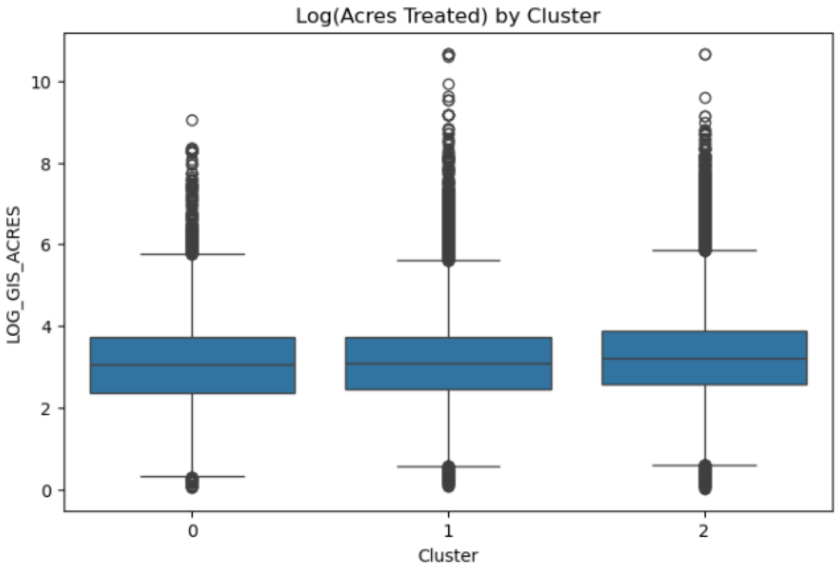
This stacked bar chart shows how fuel treatment projects are distributed across clusters in each state. States like Oregon (OR), Missouri (MO), and California (CA) have the highest number of projects, with Oregon having a strong dominance of Cluster 2. This suggests that larger or more resource-intensive projects are more frequent in these states. Other states show more balanced distributions or have smaller overall project counts. This geographic breakdown helps identify where different types of treatments are most common and can support region-specific planning or policy decisions.



Acres Treated by Cluster

Two boxplots were used to compare treatment area (GIS_ACRES) across clusters, using both raw and log-transformed values. The plots show that Cluster 2 has a slightly higher range and more large-scale treatments, while Cluster 0 contains generally smaller projects. Despite some overlap, the distribution of project sizes varies by cluster, with visible differences in median and spread. The log-transformed version provides a clearer view by reducing the influence of extreme outliers, making it easier to compare the central tendency and spread across the three clusters.



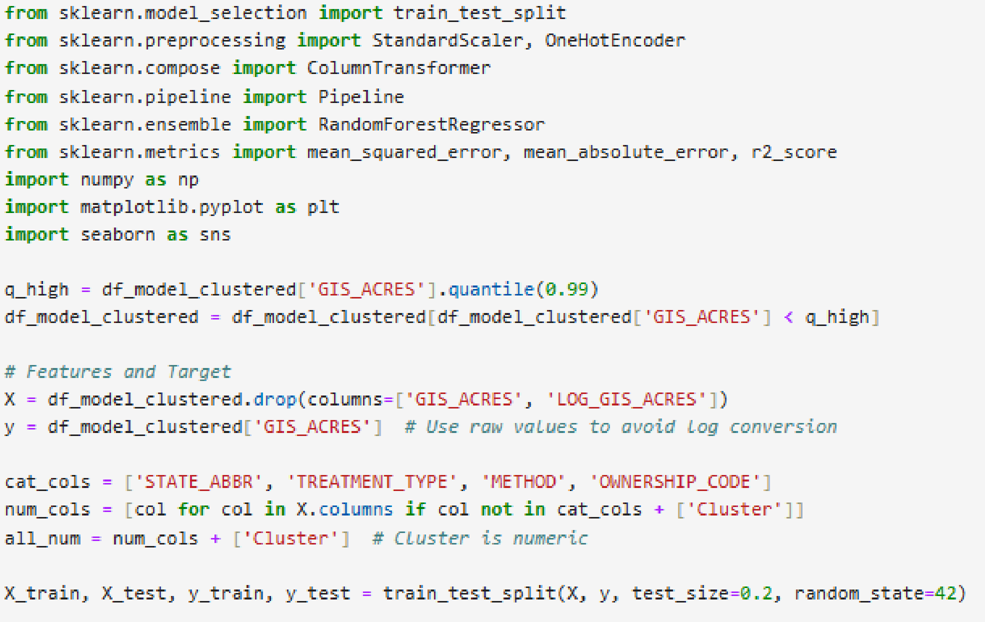


Model Building and Evaluation

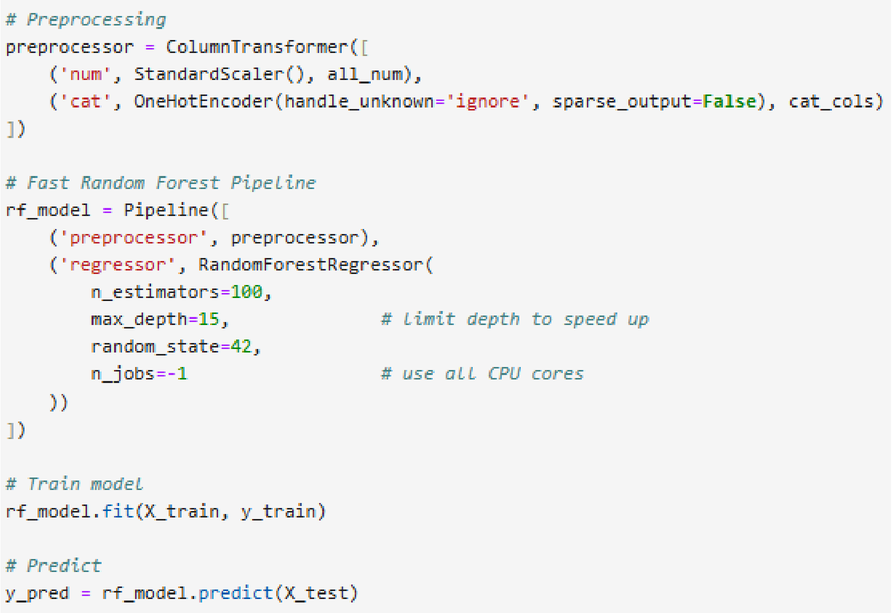
Random Forest

To predict the treated area (GIS_ACRES), a Random Forest regression model was built. Records in the top 1% of acreage were removed to reduce the effect of extreme outliers. The target variable was kept in its raw form, and the feature set included both numeric and categorical variables such as treatment type, method, state, ownership, and cluster. The data was split into training and testing sets using an 80/20 split.

A ColumnTransformer was used for preprocessing: numeric features were scaled using StandardScaler, and categorical features were encoded using OneHotEncoder. These steps were bundled into a pipeline with a RandomForestRegressor that used 100 trees and a maximum depth of 15 for efficiency. The model was trained on the training set and used to predict treatment area values on the test set.

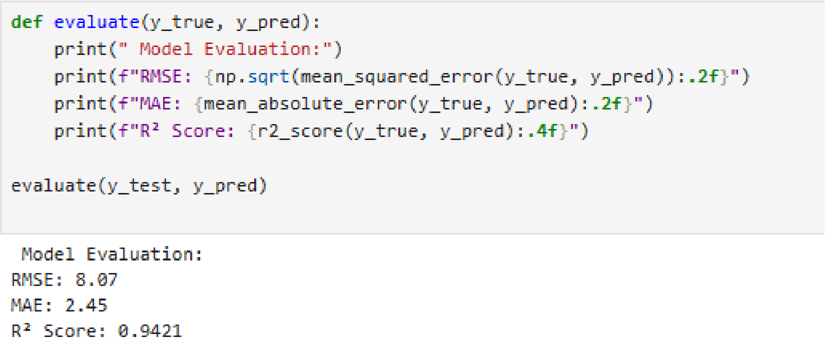


```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

q_high = df_model_clustered['GIS_ACRES'].quantile(0.99)
df_model_clustered = df_model_clustered[df_model_clustered['GIS_ACRES'] < q_high]

# Features and Target
X = df_model_clustered.drop(columns=['GIS_ACRES', 'LOG_GIS_ACRES'])
y = df_model_clustered['GIS_ACRES']  # Use raw values to avoid log conversion

cat_cols = ['STATE_ABBR', 'TREATMENT_TYPE', 'METHOD', 'OWNERSHIP_CODE']
num_cols = [col for col in X.columns if col not in cat_cols + ['Cluster']]
all_num = num_cols + ['Cluster']  # Cluster is numeric

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```
# Preprocessing
preprocessor = ColumnTransformer([
    ('num', StandardScaler(), all_num),
    ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), cat_cols)
])

# Fast Random Forest Pipeline
rf_model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(
        n_estimators=100,
        max_depth=15,              # limit depth to speed up
        random_state=42,
        n_jobs=-1                  # use all CPU cores
    ))
])

# Train model
rf_model.fit(X_train, y_train)

# Predict
y_pred = rf_model.predict(X_test)
```

The performance of the Random Forest model was evaluated using three metrics: RMSE, MAE, and R^2 Score. The Root Mean Squared Error (RMSE) was **8.07**, and the Mean Absolute Error (MAE) was **2.45**, both of which indicate low average prediction errors. The R^2 Score was **0.9421**, showing that the model explains over 94% of the variance in the treated acreage. These results suggest that the model performs very well and makes accurate predictions on the test data.



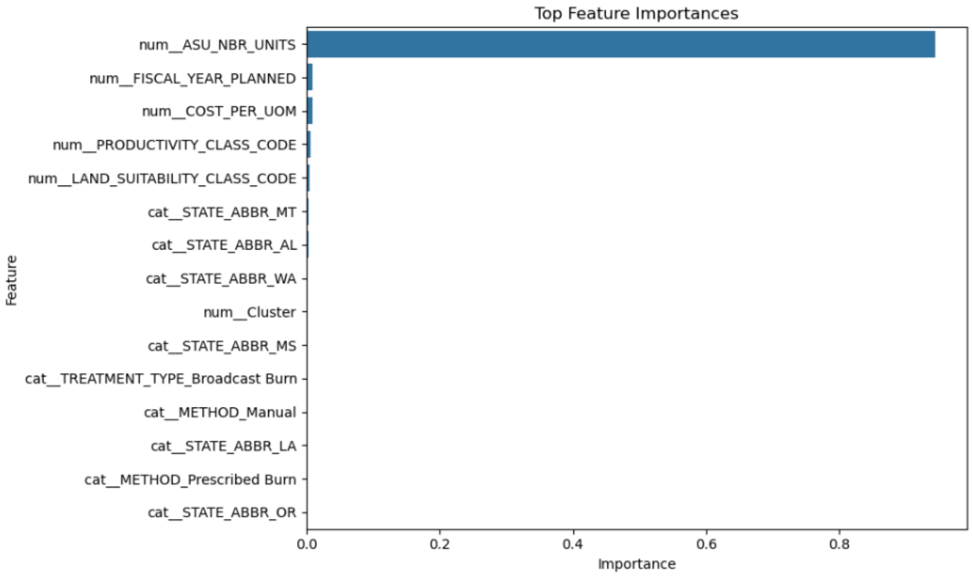
```
def evaluate(y_true, y_pred):
    print(" Model Evaluation:")
    print(f"RMSE: {np.sqrt(mean_squared_error(y_true, y_pred)):.2f}")
    print(f"MAE: {mean_absolute_error(y_true, y_pred):.2f}")
    print(f"R² Score: {r2_score(y_true, y_pred):.4f}")

evaluate(y_test, y_pred)
```

```
 Model Evaluation:
RMSE: 8.07
MAE: 2.45
R² Score: 0.9421
```

Top Feature Importance

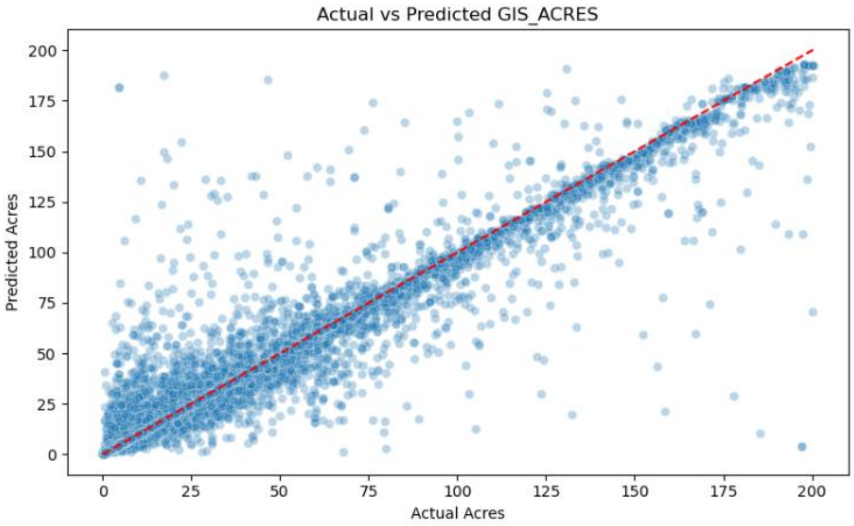
The top contributing features to the Random Forest model were identified using the model's built-in importance scores. The most influential feature by far was ASU_NBR_UNITS, which represents the number of units treated and had the highest impact on predicting GIS_ACRES. Other important features included FISCAL_YEAR_PLANNED, COST_PER_UOM, and land or productivity classification codes.

Several state identifiers and treatment attributes like method and type had lower but still noticeable influence. This analysis confirms that operational scale and planning context are key drivers of treatment size predictions.



Actual vs Predicted GIS_ACRES

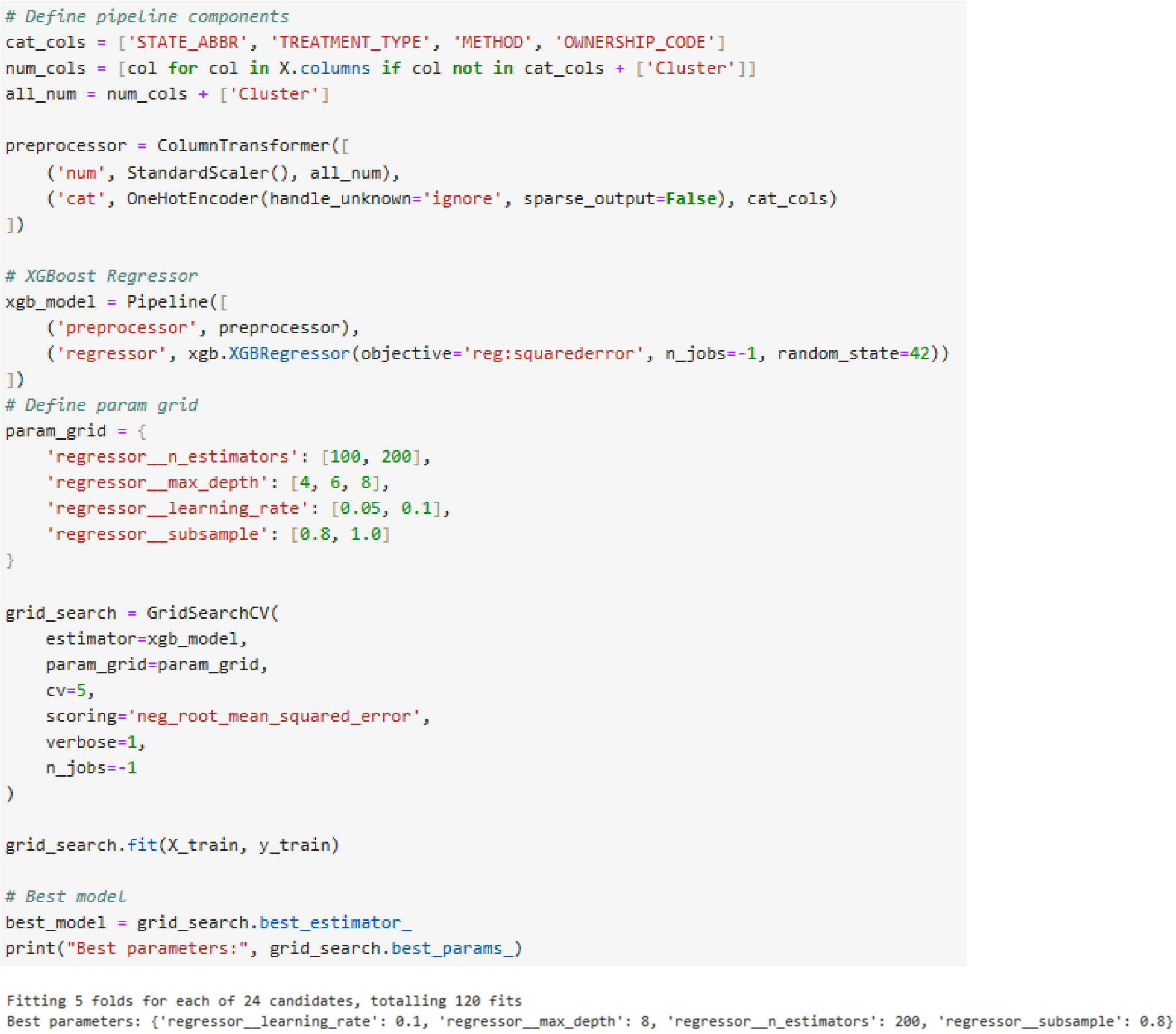
This scatter plot compares the actual versus predicted values of GIS_ACRES from the test set. Most points fall close to the diagonal red line, which represents perfect predictions. This indicates that the model's predictions align well with the true values, especially for lower and mid-range acreages. While a few predictions for higher acreages deviate slightly, the overall pattern confirms that the model captures the trend accurately.



XGBOOST

An XGBoost regression model was built using a pipeline similar to the earlier Random Forest approach. The pipeline included preprocessing with StandardScaler for numeric features and OneHotEncoder for categorical features. The model was tuned using GridSearchCV with 5-fold cross-validation to identify the best combination of hyperparameters.

The tuning process explored a range of values for number of estimators, max depth, learning rate, and subsample ratio. The objective function used was reg:squarederror, and model performance was evaluated using negative root mean squared error. After training, the best model and parameter combination were extracted for evaluation and comparison against the Random Forest results.



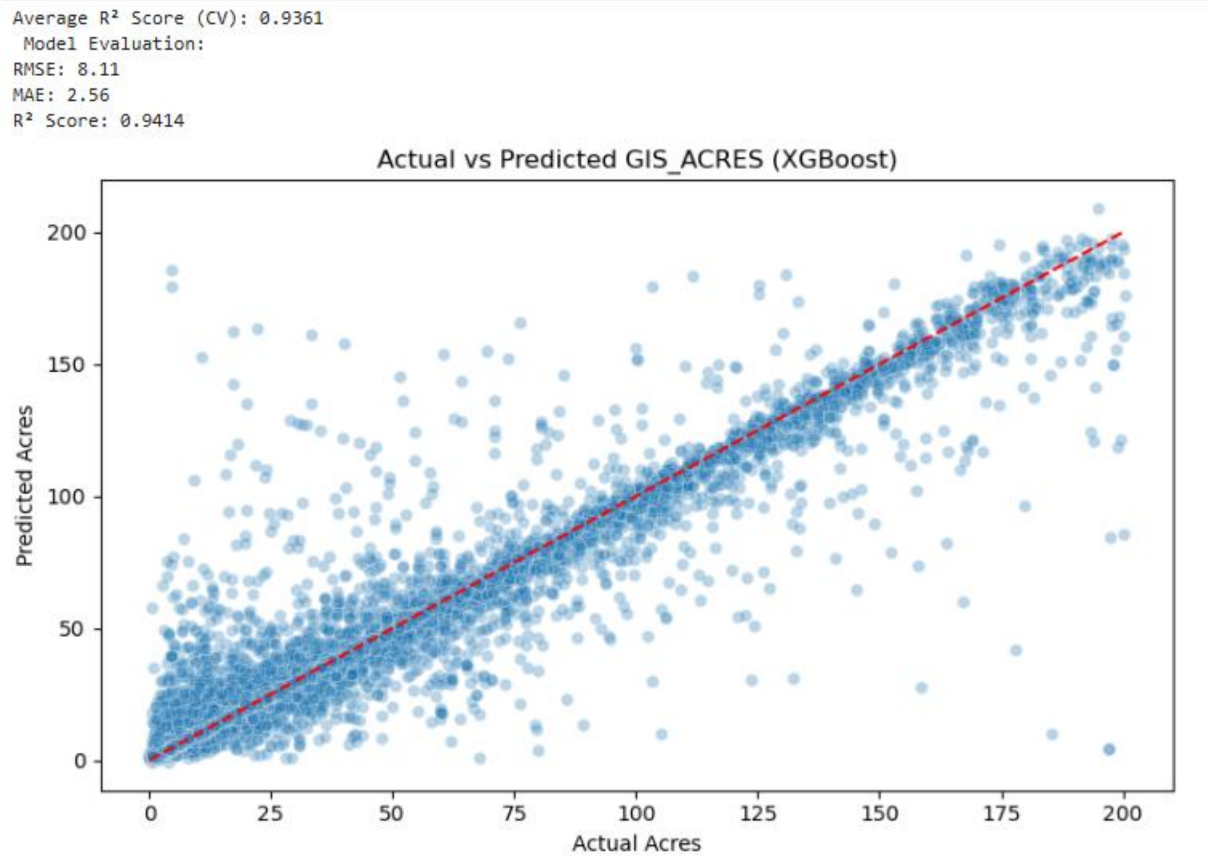
```
# Define pipeline components
cat_cols = ['STATE_ABBR', 'TREATMENT_TYPE', 'METHOD', 'OWNERSHIP_CODE']
num_cols = [col for col in X.columns if col not in cat_cols + ['Cluster']]
all_num = num_cols + ['Cluster']

preprocessor = ColumnTransformer([
    ('num', StandardScaler(), all_num),
    ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), cat_cols)
])

# XGBoost Regressor
xgb_model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', xgb.XGBRegressor(objective='reg:squarederror', n_jobs=-1, random_state=42))
])
# Define param grid
param_grid = {
    'regressor__n_estimators': [100, 200],
    'regressor__max_depth': [4, 6, 8],
    'regressor__learning_rate': [0.05, 0.1],
    'regressor__subsample': [0.8, 1.0]
}

grid_search = GridSearchCV(
    estimator=xgb_model,
    param_grid=param_grid,
    cv=5,
    scoring='neg_root_mean_squared_error',
    verbose=1,
    n_jobs=-1
)

grid_search.fit(X_train, y_train)

# Best model
best_model = grid_search.best_estimator_
print("Best parameters:", grid_search.best_params_)
```

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best parameters: {'regressor__learning_rate': 0.1, 'regressor__max_depth': 8, 'regressor__n_estimators': 200, 'regressor__subsample': 0.8}
```

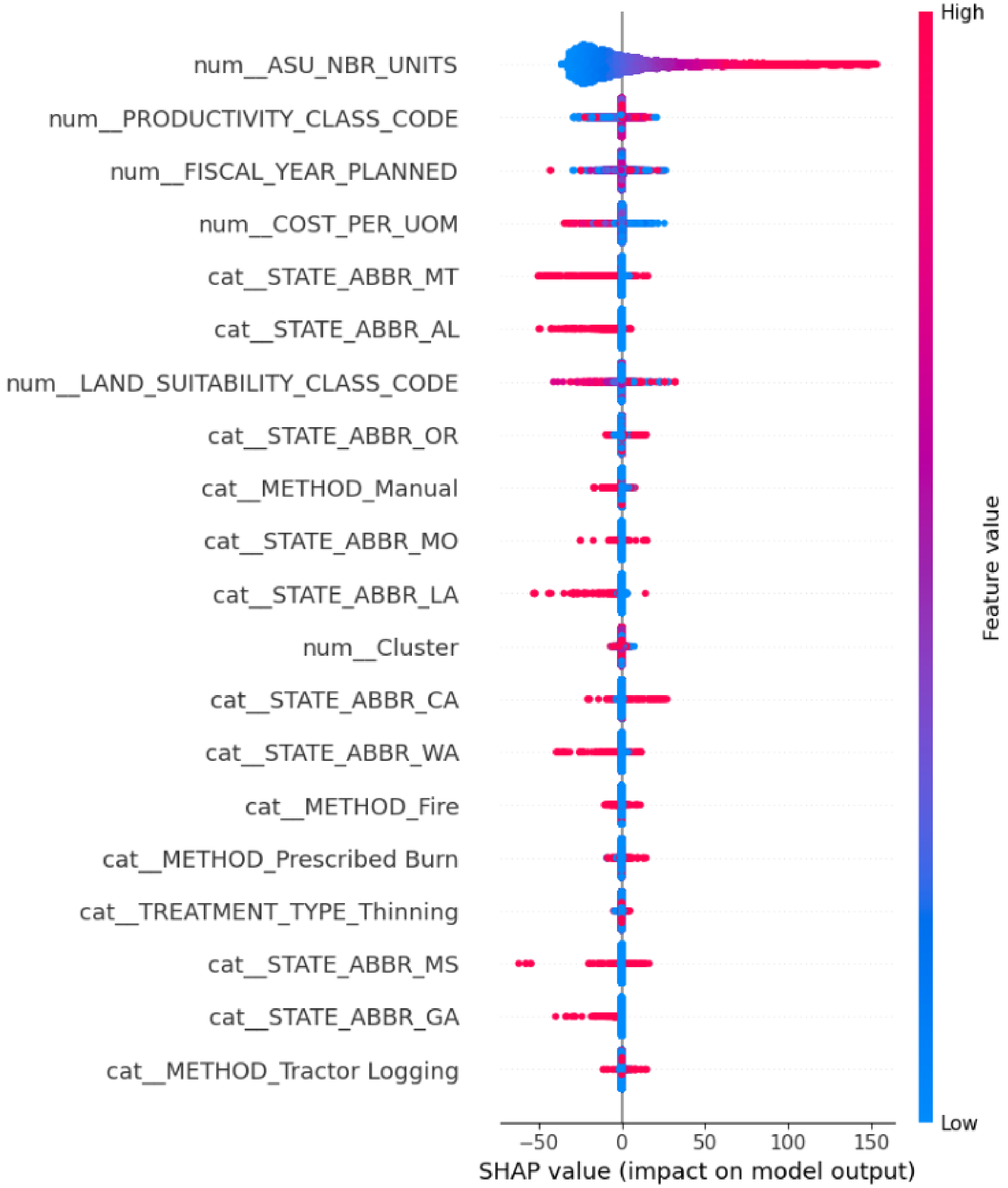
XGBoost Evaluation and SHAP Interpretation

The best XGBoost model, selected via grid search, was evaluated using cross-validation and test set performance. It achieved an average R^2 of **0.9361** from cross-validation and an R^2 of **0.9414** on the test set, with RMSE and MAE of **8.11** and **2.56**, respectively. The actual vs. predicted plot shows a tight alignment of points along the ideal prediction line, confirming strong predictive accuracy.

To understand the model's decisions, SHAP (SHapley Additive exPlanations) was used. The model was re-fit on the training set, and a SHAP summary plot was generated to highlight feature impact. This step supports transparency by explaining which features most influenced the predictions, making the model interpretable.

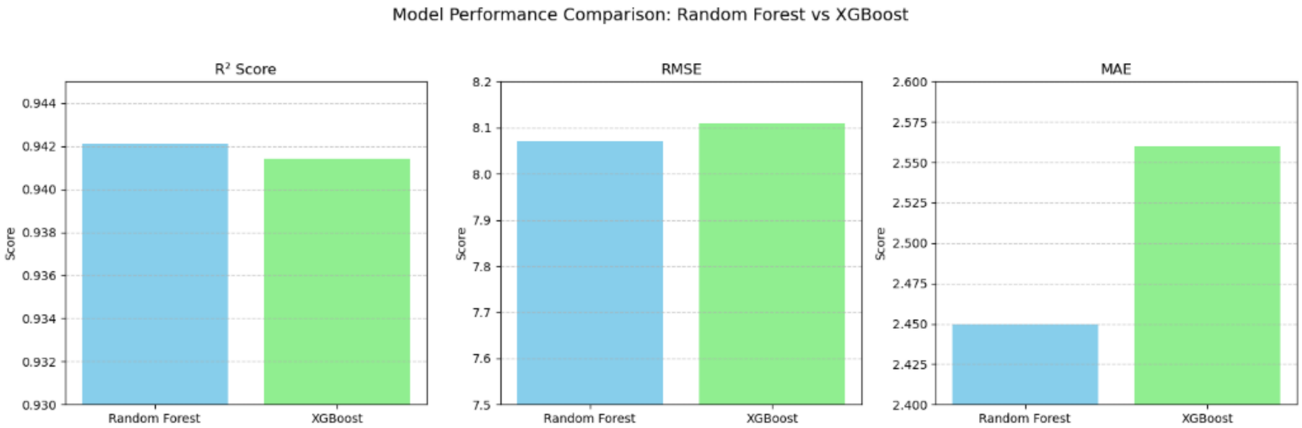


```
Average R² Score (CV): 0.9361
 Model Evaluation:
RMSE: 8.11
MAE: 2.56
R² Score: 0.9414
```


High
Low
Feature value
num__ASU_NBR_UNITS
num__PRODUCTIVITY_CLASS_CODE
num__FISCAL_YEAR_PLANNED
num__COST_PER_UOM
cat__STATE_ABBR_MT
cat__STATE_ABBR_AL
num__LAND_SUITABILITY_CLASS_CODE
cat__STATE_ABBR_OR
cat__METHOD_Manual
cat__STATE_ABBR_MO
cat__STATE_ABBR_LA
num__Cluster
cat__STATE_ABBR_CA
cat__STATE_ABBR_WA
cat__METHOD_Fire
cat__METHOD_Prescribed Burn
cat__TREATMENT_TYPE_Thinning
cat__STATE_ABBR_MS
cat__STATE_ABBR_GA
cat__METHOD_Tractor Logging
−50
0
50
100
150
SHAP value (impact on model output)

Model Comparison: Random Forest vs XGBoost

A visual comparison of the Random Forest and XGBoost models was done using three key metrics: R^2 Score, RMSE, and MAE. Both models performed well, with **Random Forest slightly outperforming XGBoost** in all metrics. It achieved a marginally higher R^2 score (0.9421 vs. 0.9414), along with lower RMSE (8.07 vs. 8.11) and MAE (2.45 vs. 2.56). While the difference is small, Random Forest proved to be slightly more accurate and consistent in predicting the treated acreage, making it the preferred model for this task.



Conclusion

This project set out to answer a practical question: given what we know about a hazardous fuel treatment project — such as its location, method, plan, and estimated inputs — can we predict how much land will actually be treated? Using real-world data from the U.S. Forest Service, we explored this through a combination of data cleaning, visual analysis, clustering, and predictive modeling.

Throughout the process, it became clear that project outcomes are influenced by a complex mix of factors. Some — like the number of units planned, the year of execution, and cost — play a more dominant role. Others, such as treatment type or method, still matter but contribute more subtly. Using Random Forest and XGBoost models, we were able to build strong predictors, both achieving over 94% accuracy (R^2). While Random Forest slightly outperformed XGBoost in terms of precision, both models demonstrated reliable performance and practical value.

Beyond prediction, the model also offers visibility into the drivers of treatment scale. This insight is just as valuable — it tells us which inputs most affect acreage outcomes, giving forest managers levers they can adjust in future planning. With tools like SHAP, we made model decisions more interpretable, ensuring that these insights can be trusted and communicated clearly to stakeholders.

With continued refinement, this kind of data-driven approach could become an essential part of planning and executing land treatment programs across the country.