**Data Mining Final Project Writeup**

**Ian Mabon and Ben Eber**

**5/3/25**

**Introduction (Ben)**

**Overview:**

The increasing intensity and frequency of wildfires in California has exposed the need for more data driven/a machine learning approach risk analysis in the residential housing market. Structural vulnerability, particularly in the materials and construction features of a home, plays a key role in determining the extent of fire damage. Despite this, current fire risk assessments rarely leverage detailed building-level data.

**Goals:**This project aims to:

* Discover patterns in home construction that correlate with fire damage severity.
* Segment homes into meaningful clusters based on construction material features.
* Visualize these segments to inform practical policy and insurance decisions.

**Decision-Makers:**

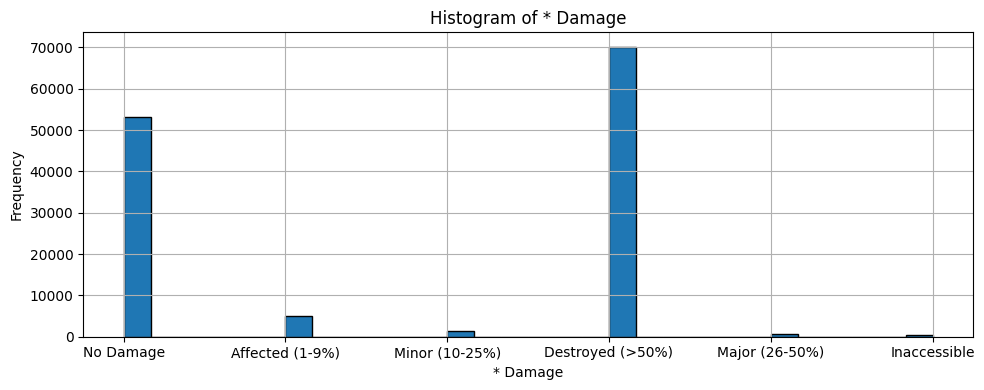
* Insurance underwriters assessing property-level fire risk.
* Real estate investors evaluating long-term asset vulnerability.
* Urban planners and building-code officials seeking to reduce fire impact through regulation.

**Potential Benefits:**

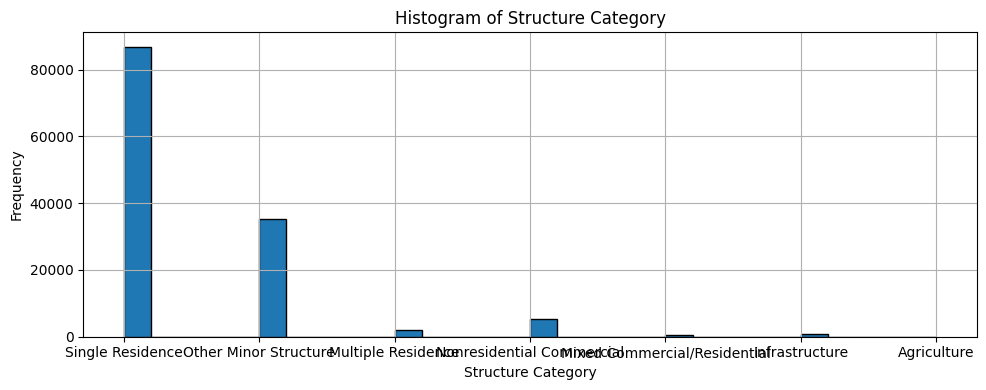
* Improve premium pricing accuracy by understanding risk differentials across home types.
* Target prevention or reinforcement programs to the most vulnerable homes.
* Use data insights to update local fire codes and construction standards.

**Variable Distributions (Ian)**

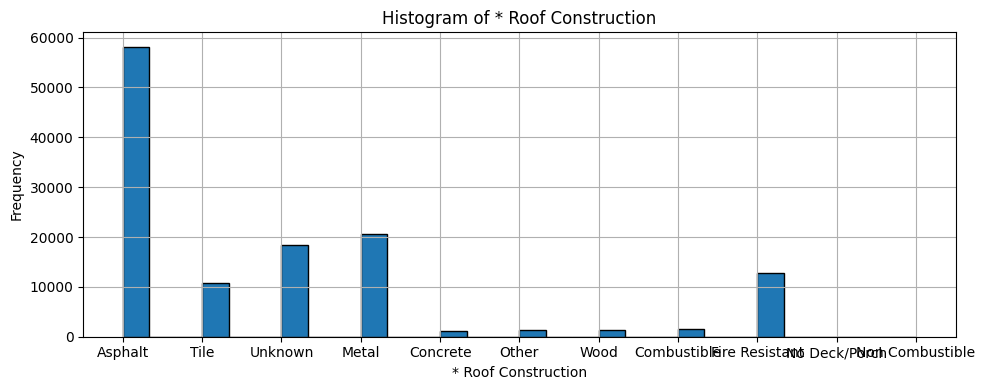
These are the distributions of each variable used in both models

****

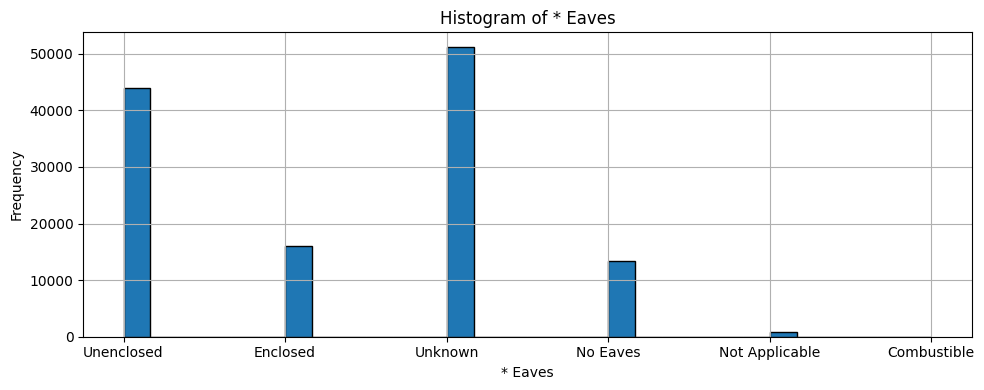
Very high class imbalance in the target variable with most rows being classified as either having no damage or are over 50% destroyed.



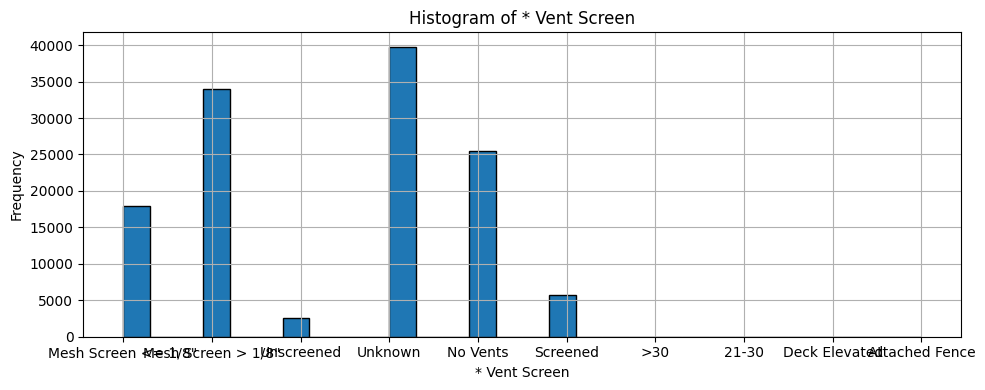
Most structures fall in the single residence category with the second most being classified as “minor structures.”

****

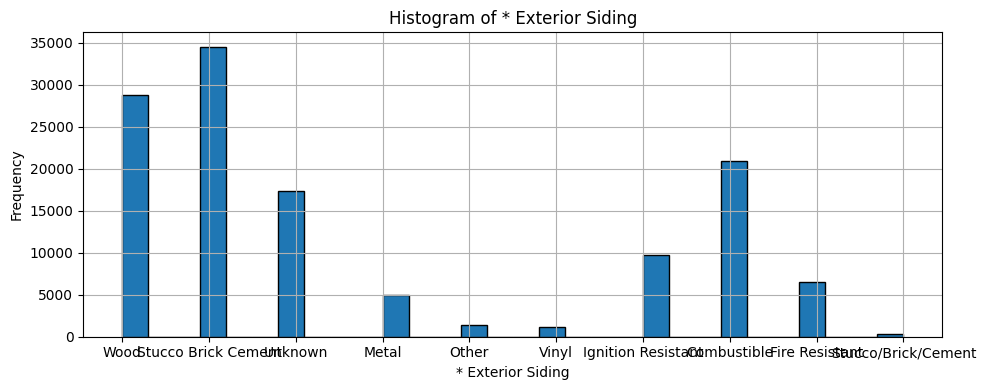
Roofs are mostly asphalt but some are metal with a significant amount of unknowns.



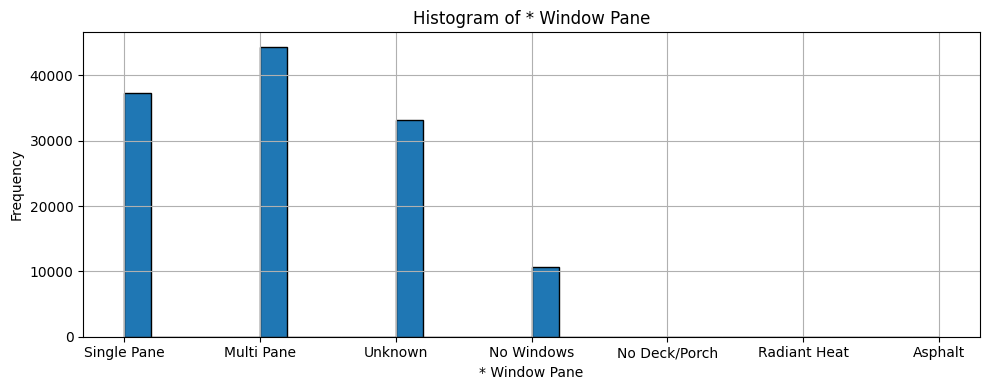
Most of the eaves (the lower edges of a roof that overhang the exterior walls of a building) are classified as unknown with the second most being unenclosed.



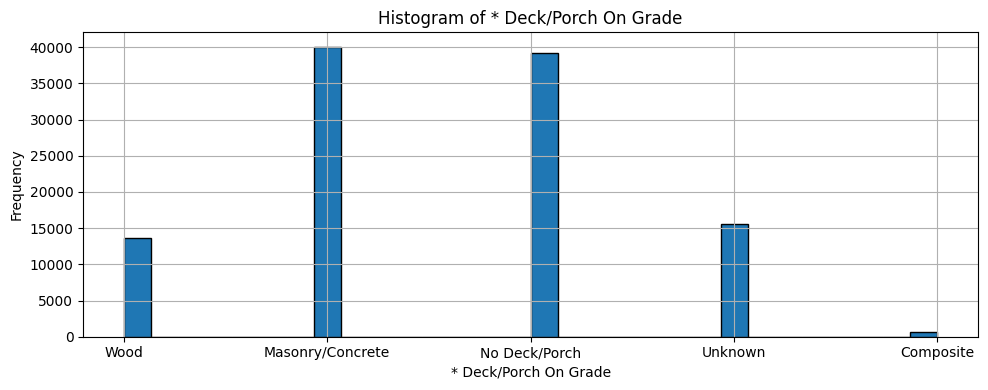
Trend that the majority of many of the categorical variables fall in the unknown category.



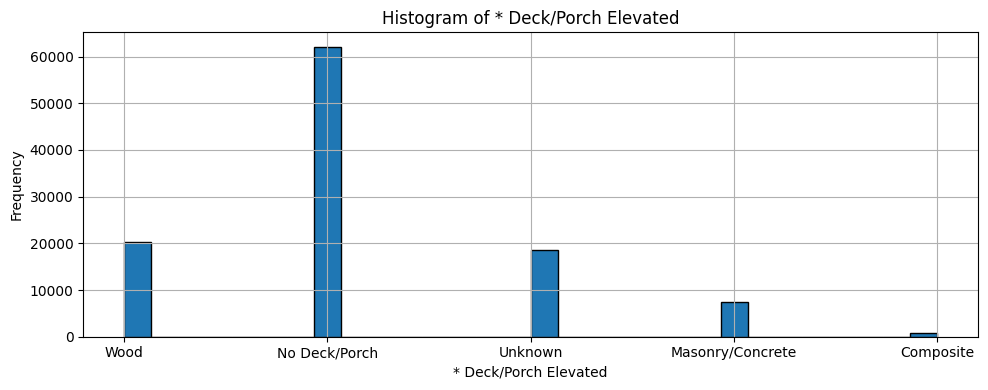
Most of the exterior siding of buildings are brick and wood, but it is tough to interpret the difference between combustible and wood exteriors.



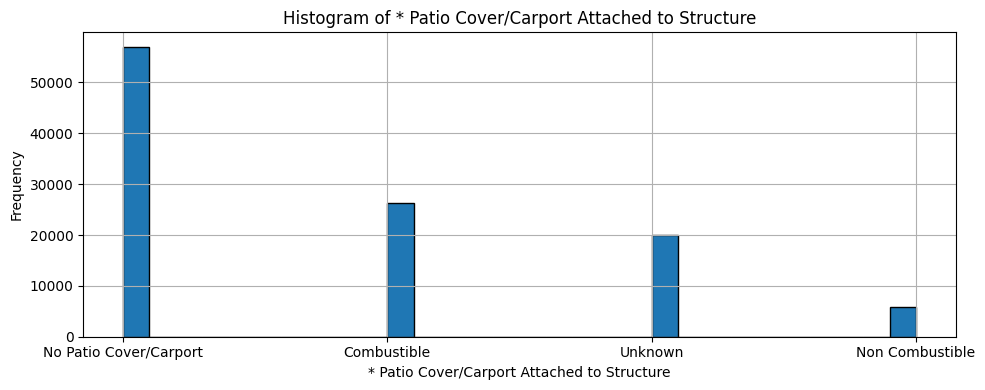
Might be some leakage that included radiant heat and asphalt window panes.

****

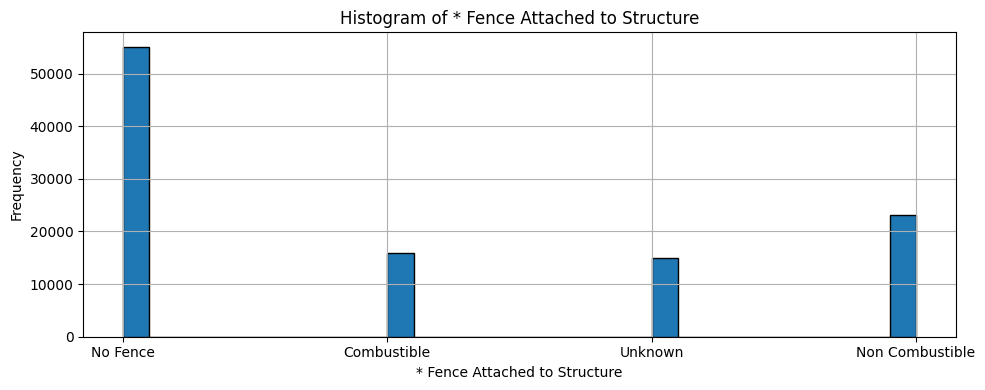
No deck or porch grade does not mean unknown.



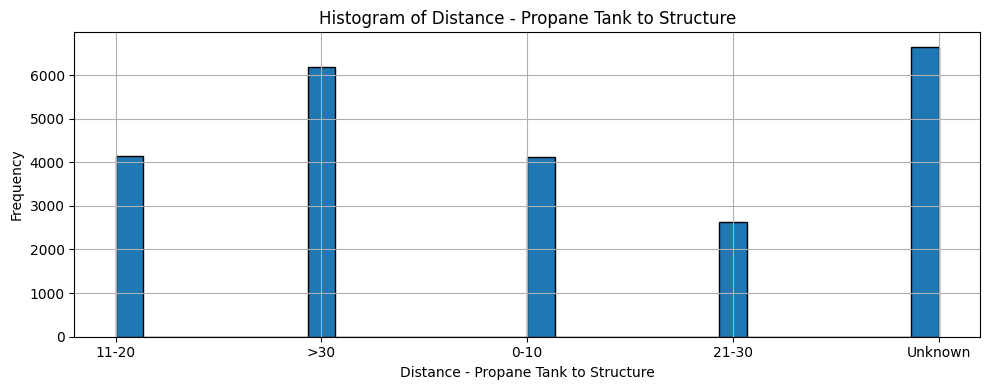
Same goes for the elevated decks/porches.



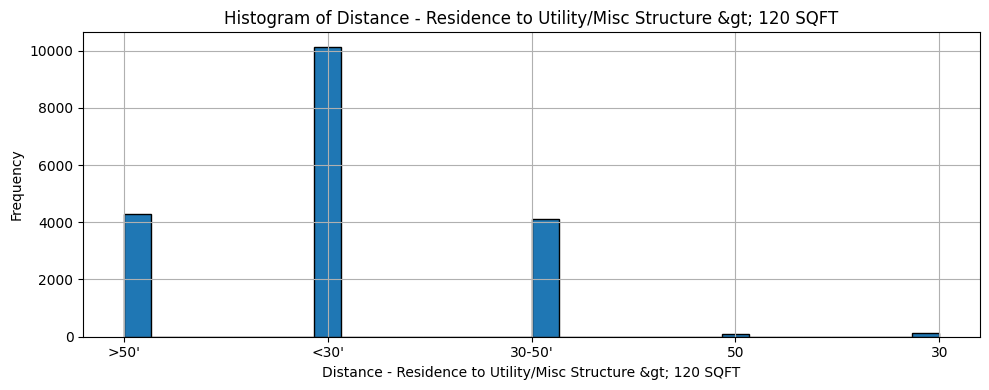
Not many patio covers and the difference between combustible and non combustible covers are difficult to interpret without more domain knowledge.



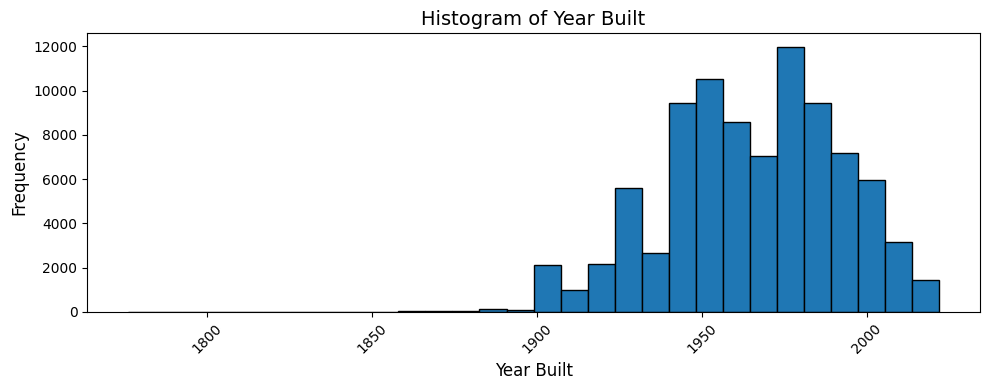
Also the same goes for the above graph.



Distance is in square feet and most are unknown.



Unclear what a utility or miscellaneous structure is but most were within 30 square feet of the structure.

****

Pretty even distribution of when each building was built, but the majority of buildings were built between the 1950s and the 2000s.

**Clustering Preprocessing (Ben)**

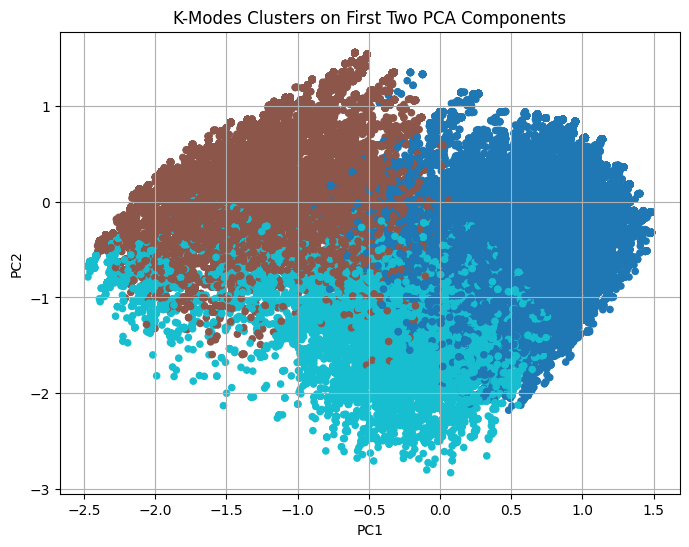
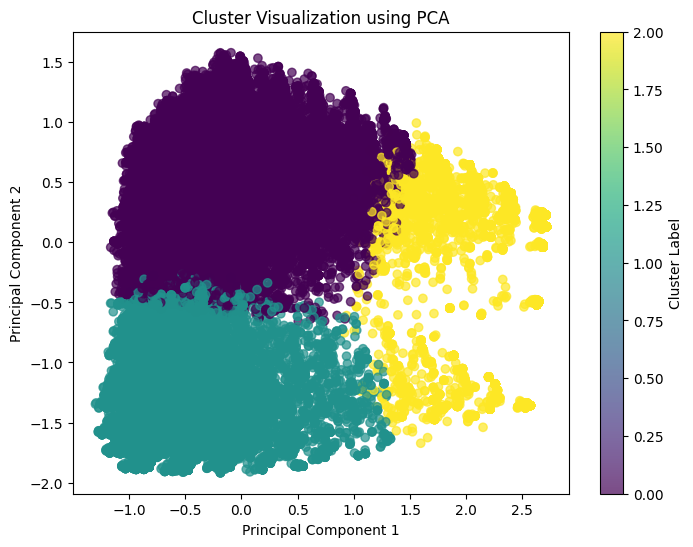
Preprocessing this dataset for clustering proved challenging largely because nearly all of our features are categorical and high‐cardinality. First, we had to identify “material” and “construction” fields, then convert them to a uniform string type and impute missing values with a “Missing” token. One‐hot encoding this full set created a very sparse, high‐dimensional matrix, so for hierarchical clustering we sampled a subset of 2,000 observations to keep the number of calculations feasible. For k-modes (purely categorical) and k-means (with all categorical encoded to dummy variables) each required careful choice of initialization, number of clusters (elbow/dendrogram methods), and hyperparameters to avoid trivial solutions. We also ran PCA on the one-hot matrix to visualize cluster separation, and aggregated cluster modes (the most frequent category per variable) rather than raw means, but preparing the data for each algorithm, in particular, filling, encoding, sampling, and scaling numeric inputs was both time-consuming and prone to the kind of sparsity and imbalance artifacts that can skew unsupervised learning as well as frequent runtime issues in Google Colab because of the number of levels in each categorical variable.

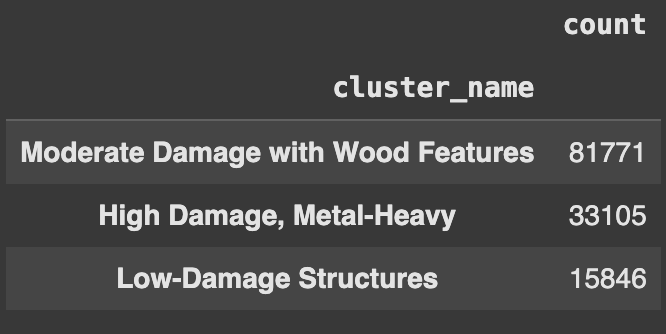
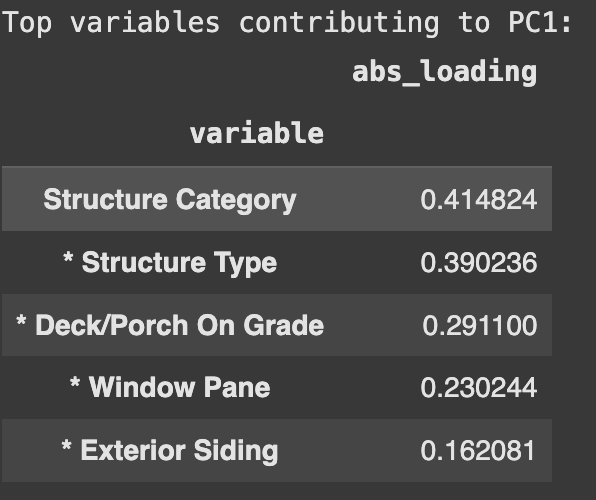
Interpreting the resulting clusters remains difficult. Because clusters are defined by combinations of discrete levels rather than continuous centroids, it’s hard to distill them into simple, actionable profiles, one cluster might be driven by “\* Window Pane\_single\_pane” and “Deck/Porch Elevated\_Masonry/Concrete,” another by “Fence Attached to Structure\_No Fence” but these modal descriptions often overlap or contradict each other.

PCA loadings tended to be dominated by a single variable (e.g. “Structure Actions Taken”), so the lower-variance components that might capture more subtle patterns aren’t easily mapped back to meaningful material distinctions. Moreover, k-modes and k-means with dummy variables can produce different assignments, highlighting that cluster membership can be highly sensitive to algorithm choice and to how we encoded and sampled the data. In practice, without clear ground truth or external labels, these clusters serve more as exploratory groupings than definitive categories, and translating them into business‐relevant insights requires further validation and domain expertise.

**Cluster Results (Ben)**

Left: K-means with dummy variables, Right: K-modes with “unknown” levels imputed

****

****

Left: Top variables resulting from pca for visualization, Right: Somewhat interpretable labels for K-means (with dummy variables) clusters

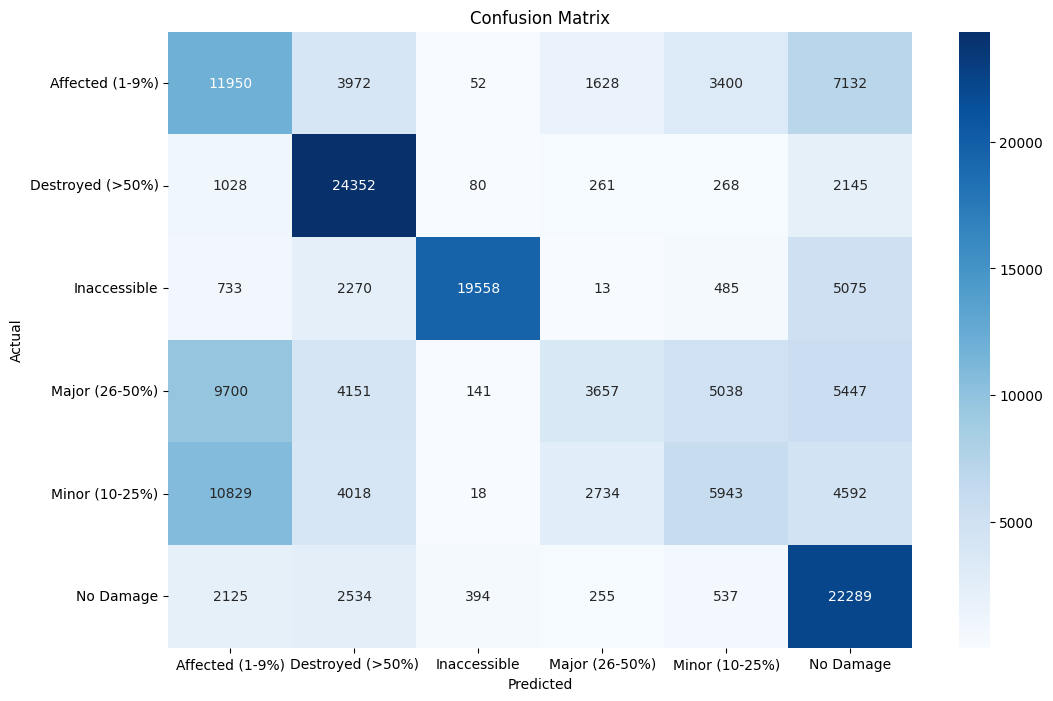
**Classification Preprocessing (Ian)**

The first step in preprocessing for the classification models was to remove any extraneous variables like ‘OBJECTID’ and ‘HAZARD TYPE’ since they would not meaningfully contribute to the model. Since these models' focuses were on the building type rather than specific fires or battalion numbers, all building related variables were kept. The value of the building designated as “Assessed Improved Value” was also dropped due to the concern of data leakage, since the lower the value of a property the more likely it was to be destroyed. Now realizing that dummies were not necessary to build decision trees, this introduced more sparsity and dimensionality for each classification model, making the models more complex without the need to do so. There was an intention of making a logistic regression model, but that was never done and the preprocessing was left the same. If done again, categorical variables would have remained the same to reduce the column size and possibly get a better model.

Since there was a lot of missing data in the dataset, missing values were imputed with means and modes depending on the column’s data type to be able to use more of the dataset. Categorical variables with less than 50 values were also dropped since those columns wouldn’t contribute to the quality of the overall models. In creating the X and y variables, SMOTE was used to oversample the rarer cases of the ‘Damage’ variable since the ‘No Damage’ and ‘Destroyed’ values dominated that column. The test size of the test-train split was 40%, meaning 60% of the data was used to train each model. A larger test size was used to ensure accurate evaluation on unseen data and avoid incorrect performance metrics, especially given the complexity and unpredictability of wildfire damage patterns. This decision was made because larger training sets generally help prevent overfitting, and in this case, the emphasis was placed on how well the model generalizes to the rare and complex conditions in the dataset.

**Classification Results (Ian)**

After all of the preprocessing was completed, each model was built on X\_train and y\_train (60% of the data) and assessed through the accuracy, precision, recall, and f1 score metrics using X\_valid and y\_valid (40% of the data). In addition to this, confusion matrices were plotted to show the distribution of predictions in each class and specific variable importance was determined for each model. First, a single decision tree classifier was built to get a good base estimate of how well these methods would work with the data. No max depth or minimum sample splits were given to allow Sci-Kit Learn to be given the most leeway to see what stands out given no hyperparameters (except random state which was 1). Here are the metrics generated:

****

Performance Metrics (Single Tree):

|  |  |  |  |
| --- | --- | --- | --- |
| Single Decision Tree | Precision | Recall | F1-Score |
| Affected (1-9%) | 0.33 | 0.42 | 0.37 |
| Destroyed (>50%) | 0.59 | 0.87 | 0.70 |
| Inaccessible | 0.97 | 0.70 | 0.81 |
| Major (26-50%) | 0.43 | 0.13 | 0.20 |
| Minor (10-25%) | 0.38 | 0.21 | 0.27 |
| No Damage | 0.48 | 0.79 | 0.60 |

**Accuracy:** 0.5198

**Top 5 Important Variables:**

\* Eaves\_Unknown (0.047577),

\* Patio Cover/Carport Attached to Structure\_Unknown (0.030856),

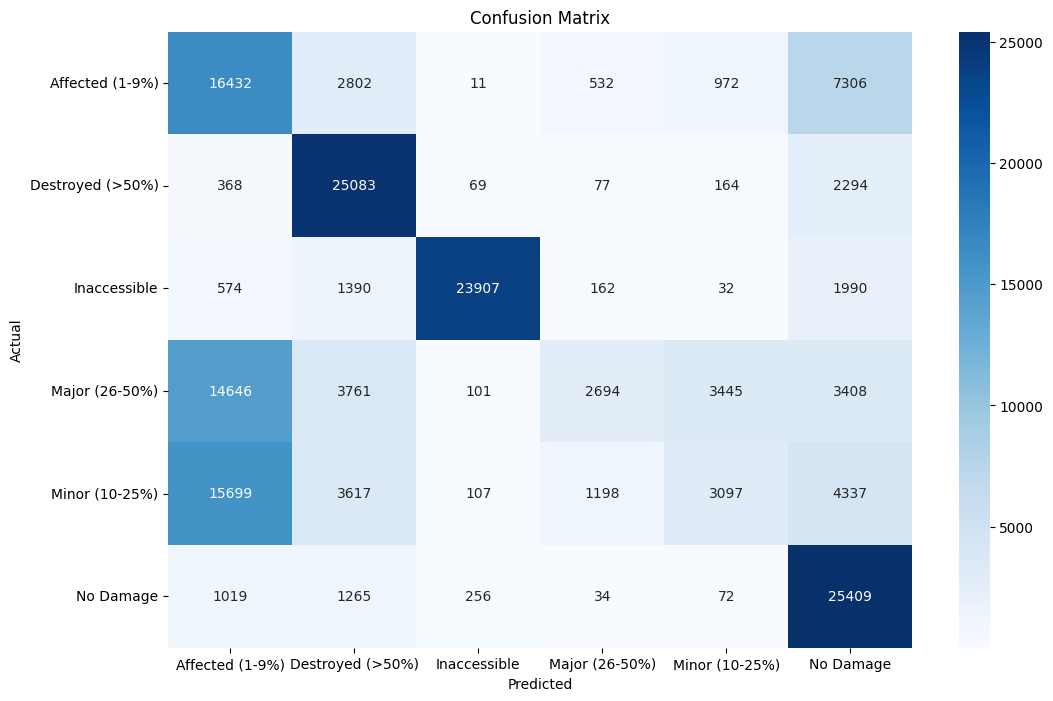
\* Fence Attached to Structure\_Unknown (0.028372),

# of Non Damaged Outbuildings < 120 SQFT (0.026717),

Distance - Propane Tank to Structure\_Unknown (0.026394)

Based on these results, the model is very good at predicting inaccessible properties, as each metric is the highest among the other classes. There are very few false positives through the precision metric, finds 70% of the true inaccessible cases, and has a decent combined f1 score. The destroyed and no damage classes are slightly worse, with the model capturing most of the total of these cases (high recall) but having low precision meaning many false positives paired with average f1 scores. The worst performers were the affected, minor, and major classes with affected being the best but not by much. The overall accuracy of this model is 52%, so this model is not reliable as a whole but some classes like inaccessible, destroyed, and no damage performed better than the others. For the variable importances, 4 of the top 5 variables were classified as “Unknown,” meaning that the categorical variable is truly unknown or does not fit into any of the existing categories. More domain knowledge is needed to confirm this is the case, but this is the best interpretation for now.

The next model created was the Random Forest Classifier using balanced class weights to further fix the class imbalance issues with the dataset:



|  |  |  |  |
| --- | --- | --- | --- |
| Random Forest | Precision | Recall | F1-Score |
| Affected (1-9%) | 0.34 | 0.59 | 0.43 |
| Destroyed (>50%) | 0.66 | 0.89 | 0.76 |
| Inaccessible | 0.98 | 0.85 | 0.91 |
| Major (26-50%) | 0.57 | 0.10 | 0.16 |
| Minor (10-25%) | 0.40 | 0.11 | 0.17 |
| No Damage | 0.57 | 0.91 | 0.70 |

**Accuracy: 0.5740**

**Top 5 Important Variables:**

Year Built (parcel) (0.085170)

\* Eaves\_Unknown (0.047577)

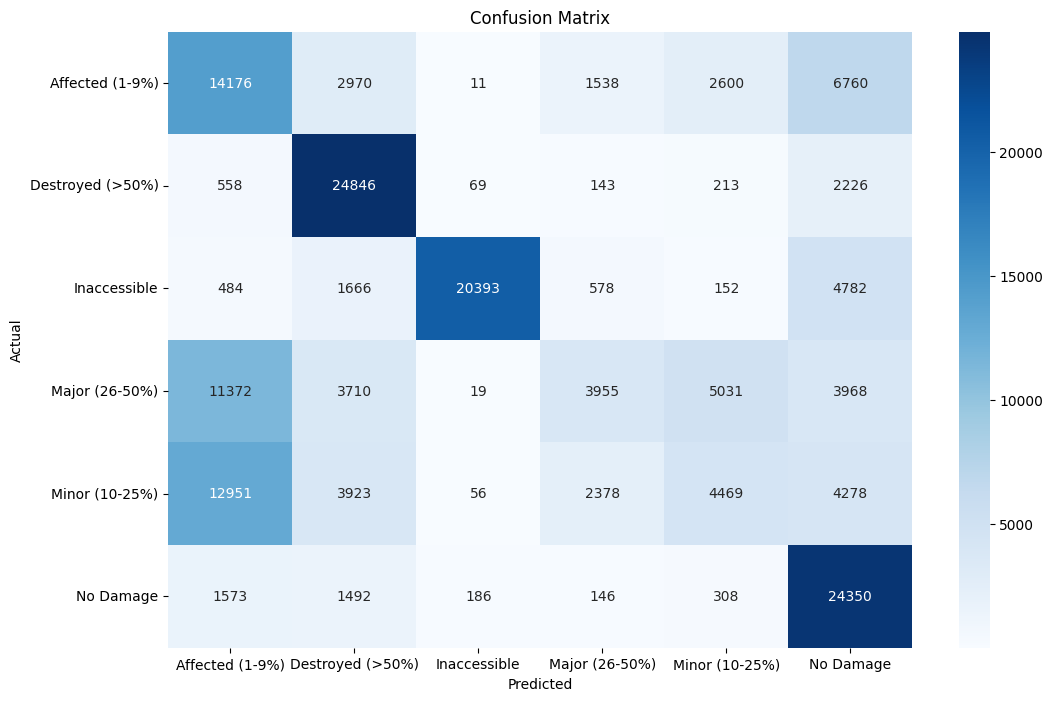
\* Patio Cover/Carport Attached to Structure\_Unknown (0.030856)

\* Fence Attached to Structure\_Unknown (0.028372)

# of Non Damaged Outbuildings < 120 SQFT (0.026717)

This model did well with an overall accuracy 57% and higher f1 scores for every class except major and minor. These classes consistently underperform since they are rare cases and it is possible the classification metric done by the persons who created this dataset could be subjective. The other classes are easily identified as a building with no damage is easily distinguishable from a completely destroyed or inaccessible building. What is very interesting is that the most important variable is the year the building was built, which makes sense since newer builds should be built with more modern materials as opposed to historical buildings composed of mainly flammable material like wood. 3 of the other important variables are unknown values, but the number of non-damaged outbuildings also seems to be contributing to the model as well. Non-damaged buildings attached to the property may include some data leakage, since if the buildings surrounding the target building aren’t damaged, it is likely the target isn’t damaged as well. Overall this is a good model, but more work and domain knowledge is needed to determine its effectiveness in the real world.

Next, here are the metrics for a Bootstrap Aggregation model with 100 base learners:



|  |  |  |  |
| --- | --- | --- | --- |
| Bootstrap Aggregation | Precision | Recall | F1-Score |
| Affected (1-9%) | 0.34 | 0.51 | 0.41 |
| Destroyed (>50%) | 0.64 | 0.89 | 0.75 |
| Inaccessible | 0.98 | 0.73 | 0.84 |
| Major (26-50%) | 0.45 | 0.14 | 0.21 |
| Minor (10-25%) | 0.35 | 0.16 | 0.22 |
| No Damage | 0.53 | 0.87 | 0.65 |

**Accuracy: 0.5477**

**Top 5 Important Variables:**

\* Deck/Porch On Grade\_Unknown (0.139699)

Year Built (parcel) (0.106726)

\* Eaves\_Unknown (0.067153)

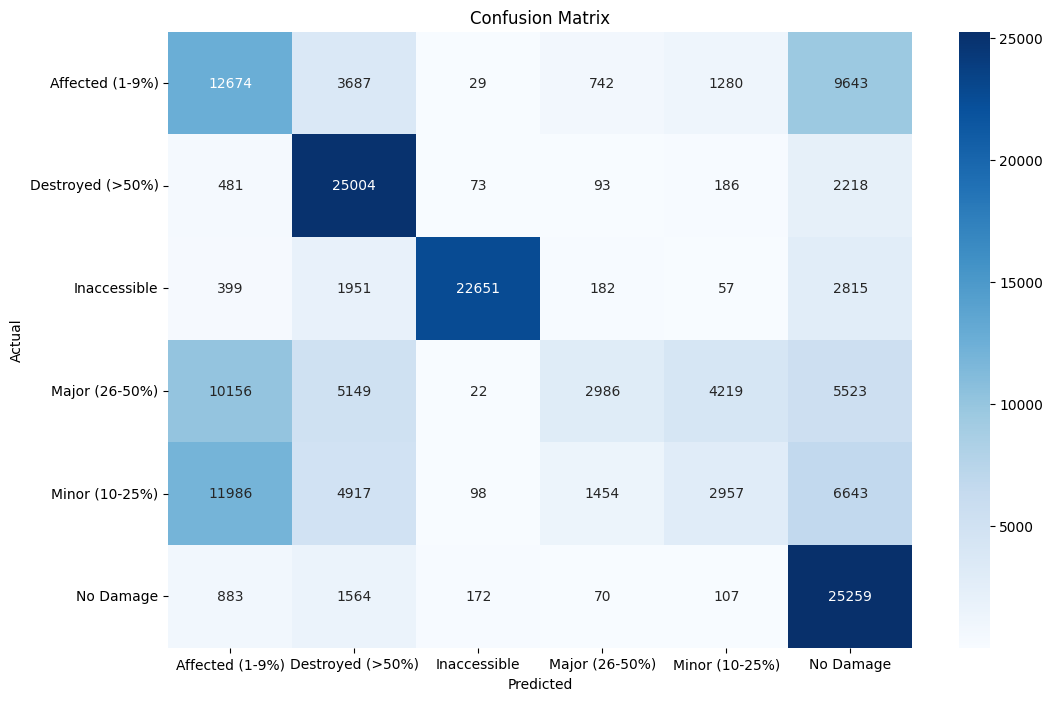
# of Non Damaged Outbuildings < 120 SQFT (0.033586)

Distance - Propane Tank to Structure\_Unknown (0.033197)

\* Eaves\_Unenclosed (0.023588)

The Bootstrap Aggregation model doesn’t improve as much as the random forest did. The high performing classes (destroyed, inaccessible, and no damage) still have high f1 scores, indicating a better balance of precision and recall. Affected does as well but Major and Minor underperformed again according to their f1 scores. Overall accuracy decreases a bit, but not a large departure from the random forest. Again, many of the important variables are unknown, but year built shines its way to the second most important value, similarly as to the random forest model. This indicates that the year built may be in fact the most important variable that can be easily interpreted.

Lastly, here are the metrics for a Gradient Boosting model with 100 base learners:

****

|  |  |  |  |
| --- | --- | --- | --- |
| Gradient Boosting | Precision | Recall | F1-Score |
| Affected (1-9%) | 0.35 | 0.45 | 0.39 |
| Destroyed (>50%) | 0.59 | 0.89 | 0.71 |
| Inaccessible | 0.98 | 0.81 | 0.89 |
| Major (26-50%) | 0.54 | 0.11 | 0.18 |
| Minor (10-25%) | 0.34 | 0.11 | 0.16 |
| No Damage | 0.48 | 0.90 | 0.63 |

**Accuracy: 0.5438**

**Top 5 Important Variables:**

\* Patio Cover/Carport Attached to Structure\_Unknown (0.326816)

\* Street Type (e.g. road, drive, lane, etc.)\_Drive (0.060367)

\* Deck/Porch On Grade\_Unknown (0.057211)

\* Eaves\_Enclosed (0.057025)

\* Structure Type\_Single Family Residence Single Story (0.052966)

This last model has similar strengths and weaknesses to the other ensemble model (bagging model). Again, the high performing classes, no damage, inaccessible, and destroyed do equally as well and underperformers get slightly worse with major and minor f1 scores dropping by a few points each. Accuracy stays about the same still at around 54%, but the important variables change dramatically, since gradient boosting uses random rules in its base learners to allow undervalued variables to shine through. While most of the variables are still in that unknown category, street type and family residence single story show up as important variables in this model. It was interesting to see these pop up since they were unseen in the other models, which could indicate their importance to the target variable.

**Conclusion (Ian)**

Across all four models, the most consistent findings were that the models were best at classifying the most extreme classes: Destroyed, Inaccessible, and No Damage, with F1 scores typically above 0.70. These outcomes are most likely due to the nature of the Damage variable and the subjectivity that comes with it when a human chooses to classify a structure’s damage. The more ambiguous classes like Affected, Minor, and Major damages are tough to classify using relative judgement, so these classes appeared less frequently and were unreliable to predict in each model. The Random Forest model performed best overall, achieving the highest accuracy (57%) and the best balance across class metrics. Notably, “Year Built” emerged as a consistently important and interpretable variable across multiple models, while many top predictors were marked as “Unknown,” suggesting either missing data or a characteristic that doesn’t fall into the predefined categories, which could be interpreted with more domain knowledge.

The choice of creating dummies for categorical variables likely introduced unnecessary dimensionality for the tree-based methods used and should be reconsidered in future model creations. Finally, the decision to use SMOTE and a large test set helped get rid of class imbalance and offered a more realistic assessment of model generalization, but further tuning and feature engineering would be necessary to use these models confidently in real-world wildfire damage prediction scenarios.