Introduction

Amid the destruction of wildfires in California, widespread data collection and analysis presents the opportunity to identify patterns in damage to anticipate vulnerabilities in future catastrophes. Using a large dataset containing the information of over 100,000 properties across the state of California from June 2020 to December 2024, I constructed two models aiming to predict which homes are most likely to be severely damaged in wildfires. Of those models, the Naïve Bayes and the k-nearest neighbor (kNN) classifiers, the kNN model produced the superior 90% accuracy level using data concerning properties' physical and geographic characteristics and their outcomes after wildfires, ranging from unaffected properties to over 50% structurally damaged buildings. As wildfires continue to rage in California, stakeholders like homeowners, insurers, and public officials can use predictive models like this one in tandem with continually expanding datasets to identify high-risk properties and alleviate vulnerabilities before wildfires occur.

Data

The dataset contains 100,230 observations in comma-separated value (.csv) format. Using the programming language Python, I loaded the data and several packages like pandas, numpy, and others that allowed me to view, clean, manipulate, and analyze the entire dataset. The data contains detailed physical characteristics of properties in California in fire-prone areas, including the type of roof, siding, year of construction, real estate value, and geographic coordinates. Each property also has information on a wildfire-related incident between June 2020 and December 2024, corresponding to its level of damage. Damage was classified as no damage, affected (1-9%), minor (10-25%), major (26-50%), or destroyed (>50%). I generated a bar graph to display the distribution of damage classifications (Table 1).

Before building a predictive model, I took a few steps to clean the data and prepare it for analysis. I adjusted the names of the columns to remove erroneous asterisks and spaces. I removed rows that contained missing values, instead imputing them with the median value for numerical columns and the most common value for categorical, or non-numerical values. Using a method called one-hot encoding, I converted those non-numerical variables into binary variables, represented by a 0 or 1. Since the target variable of the predictive model would be damage, I also created a binary variable to represent whether a property had sustained severe damage (>50%) as 1, and 0 otherwise. I split the dataset into two buckets: I dedicated 70% of it to train the model and left the other 30% to test its accuracy, such that the model would learn from the existing data and evaluate itself with previously unseen data.

Methods

The first model, the kNN classifier, is a machine learning technique that works by comparing each observation to the most similar other observation in the dataset – hence, its nearest neighbors – to

predict the target variable. The kNN classifier is simple to understand and does not require complex assumptions about the data. However, it has some limitations: It can struggle with large datasets, as it needs to compare every new property to all others in the dataset. The model's performance depends heavily on how the data is prepared, especially on scaling numerical features and choosing the right number of neighbors. In this case, the model compared properties with similar characteristics, values, and locations to predict whether the property would suffer severe damage in the event of a wildfire. Using a scikit-learn feature called KNeighborsClassifier, I scaled the variables to make sure that no one variable dominated the similarity calculation. Then, I filtered the dataset to only include columns with important observable characteristics that could affect the property's fire vulnerability and the target variable: damage. I calculated the predicted values using the training bucket I separated earlier and compared it to the testing bucket. I generated the output, signifying how many of the predictions of severe damage were correct, in a formatted classification table (Table 2). The accuracy was about 90%.

The second model, the Naïve Bayes classifier, is a similar machine learning technique that assumes predictors are independent of each other – as such, it's based on Bayes theorem of conditional probability. This classifier is simple and predictors are easy to estimate, but it suffers from the strong assumption of independence. Once again – this time using another scikit-learn feature called GaussianNB – I separated the data into training and testing buckets, generated predictions, calculated the accuracy, and formatted the results into another formatted table (Table 3). The accuracy was about 54.6%.

Results and Analysis

Of the two models, the kNN classifier was much more accurate, with an accuracy rate of 90% versus 54.6%. Both models were slightly more effective at predicting severely damaged homes than non-severely damaged homes. I ran several cross-validation tests (Table 4) that evaluated the precision of each model – notably, I tested different k values, or number of neighbors, for the kNN classifier. The accuracy was generally the same across k values, with a marginally higher accuracy around k=7. These results suggest that the kNN classifier was highly effective at predicting whether properties would suffer severe damage in wildfires.

To visualize the data, I converted the damage categories into a single-digit value corresponding to the severity of the property's damage. I color-coded these values, with red signifying severe damage. Using each property's latitude and longitude, I plotted each property as a dot on an interactive heatmap of California. This web-based map (Table 5) reveals geographic patterns in fire damage around more fire-prone and heavily-wooded areas of the state, including neighborhoods of Los Angeles severely damaged in the Palisades fire in January 2024, along with other hotspots including Ventura County, north of Los Angeles, and Marin County, north of the San Francisco Bay area.

Appendix

Table 1: Bar graph showing the distribution of damage classifications

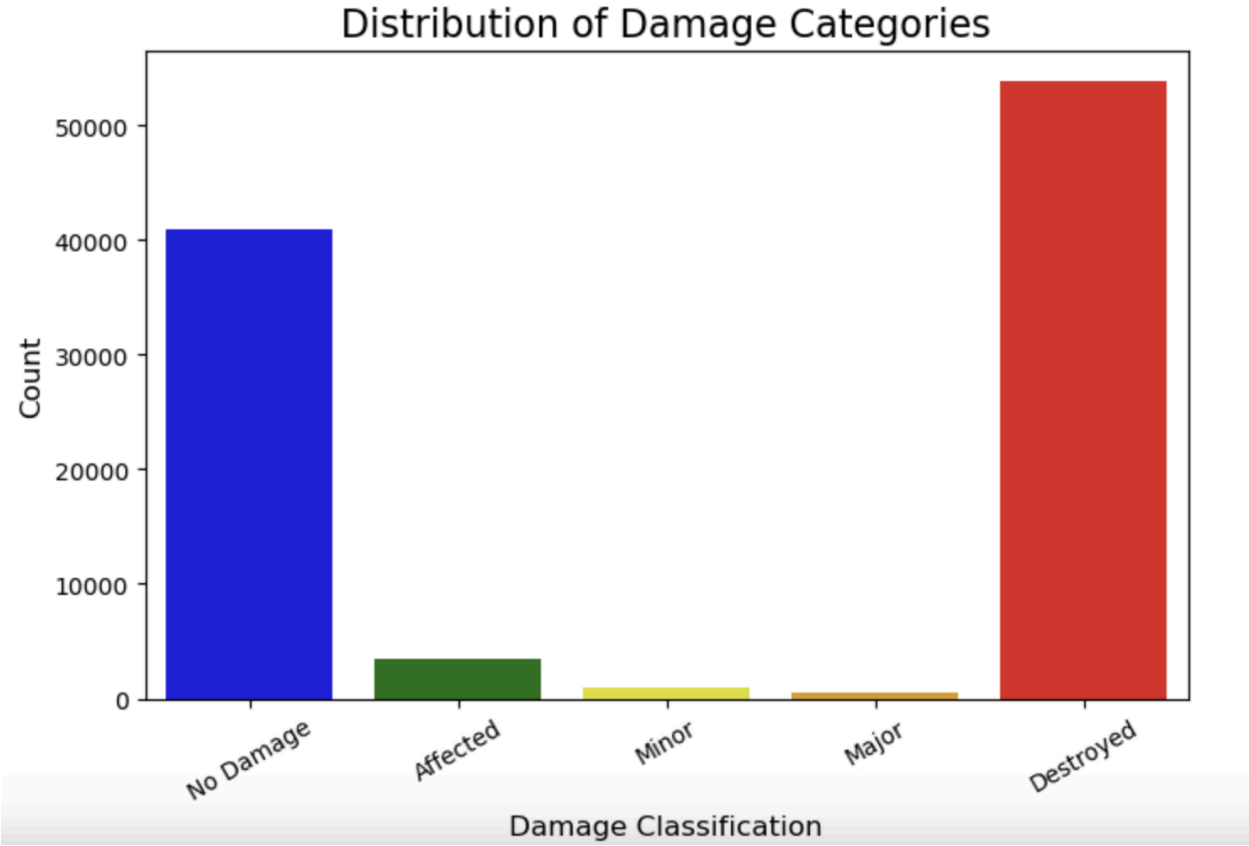

Table 2: k-nearest neighbor classification report

Model Accuracy: 0.9001962153713127

Classification Report:

	precision	recall	f1-score	support
0	0.893022	0.890903	0.891961	13905.000000
1	0.906341	0.908191	0.907265	16164.000000
accuracy	0.900196	0.900196	0.900196	0.900196
macro avg	0.899681	0.899547	0.899613	30069.000000
weighted avg	0.900182	0.900196	0.900188	30069.000000

Table 3: Naïve Bayes classification report

Naïve Bayes Model Accuracy: 0.546276896471449

Classification Report:

	precision	recall	f1-score	support
0	0.541351	0.123337	0.200902	13905.000000
1	0.546857	0.910109	0.683200	16164.000000
accuracy	0.546277	0.546277	0.546277	0.546277
macro avg	0.544104	0.516723	0.442051	30069.000000
weighted avg	0.544311	0.546277	0.460168	30069.000000

Table 4: Cross-validation table

```
Name: count, dtype: int64
KNN Cross-validation scores: [0.50478899 0.54694203 0.53526888 0.48328844 0.53407164]
Mean cross-validation score: 0.5208719944128505
Accuracy with k=1: 0.8934450763244538
Accuracy with k=2: 0.8894209983704147
Accuracy with k=3: 0.899863646945359
Accuracy with k=4: 0.8972696132229206
Accuracy with k=5: 0.9001962153713127
Accuracy with k=6: 0.8997638764175729
Accuracy with k=7: 0.9004955269546709
Accuracy with k=8: 0.8979347500748279
Accuracy with k=9: 0.8977684658618511
Naïve Bayes Mean Accuracy: 0.5306594831886661
```

Table 5: Heatmap showing fire damage in California and selected hotspots including Ventura and Marin Counties, with red indicating more severe damage