California Wildfire Analysis

Aiden, Ethan, Sarju, Ty

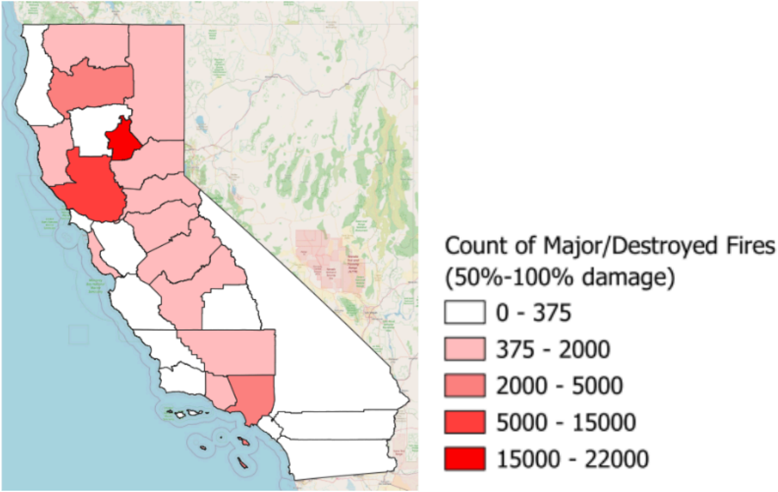
2025-03-09

Slide 1: Introduction

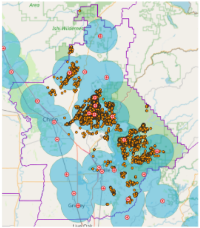
Slide 2: Research Questions

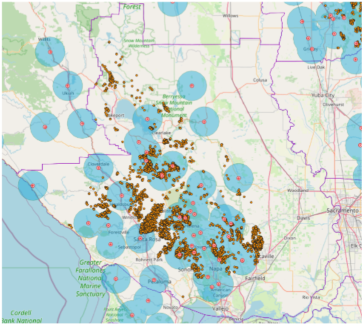
- What variables are most impactful in fire damage levels?
- Are fire damage and occurrence predictable based on weather variables?

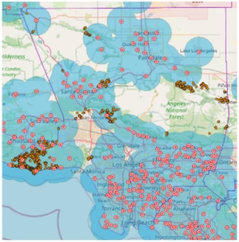
Slide 3: Choropleth Map



Slide 4: Buffer



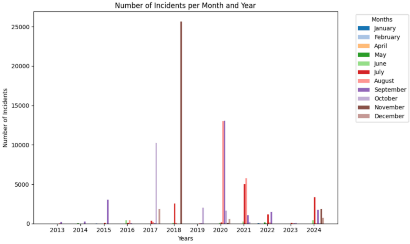


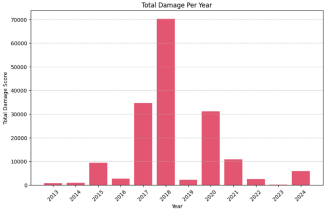


Slide 5: Summary

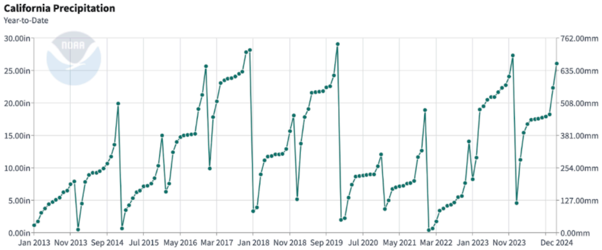
	Butte County Unit	Lake County Unit	Los Angeles County Unit
Number of Majorly Damaged Buildings Within 5-Mile Buffer of Fire Station	21561	9789	1836
Total Number of Majorly Damaged Buildings in Cal Fire Unit Jurisdiction	21964	14305	2065
Percent of Majorly Damaged Buildings within 5-Mile Buffer of Fire Station	98.17%	68.43%	88.91%

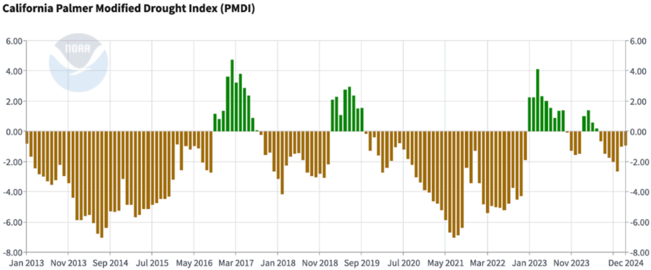
Slide 6: Data Analysis





Slide 7: Data Analysis





Slide 8: Structure Variables

	Category
0	# Units in Structure (if multi unit)
1	# of Damaged Outbuildings LT 120 SQFT
2	# of Non Damaged Outbuildings LT 120 SQFT
3	* Deck/Porch Elevated
4	* Deck/Porch On Grade
5	* Eaves
6	* Exterior Siding
7	* Fence Attached to Structure
8	* Patio Cover/Carport Attached to Structure
9	* Roof Construction
10	* Structure Type
11	* Vent Screen
12	* Window Pane
13	Distance - Propane Tank to Structure
14	Distance - Residence to Utility/Misc Structure...
15	Structure Category
16	Structure Defense Actions Taken

Slide 9: XG Boost Model

```
CAFire["* Damage"] = CAFire["* Damage"].astype("category").cat.codes

#Does not create column for null values
X = pd.get_dummies(CAFire.loc[:, slice('Structure Defense Actions Taken', 'Distance - Residence to Utility/Misc Structure &gt; 120 SQFT')])
y = CAFire["* Damage"]

#Clean for proper usage of XGBoost
X.columns = X.columns.str.replace('[', '').str.replace(']', '').str.replace('<', 'LT').str.replace('>', 'GT')
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
# Computationally expensive (takes a long time)

# Tested max_depth from 5-15 (Best: 10)
# Tested learning_rate from 0.01 - 0.1 (Best: 0.05)
# Tested n_estimators from 150 - 250 (Best: 200)

param_grid = {
    'max_depth': [5, 10, 15],
    'learning_rate': [0.01, 0.05, 0.1],
    'n_estimators': [100, 150, 200]
}

grid_search = GridSearchCV(estimator=xgb.XGBRegressor(), param_grid=param_grid, scoring='neg_mean_squared_error', cv=3)
grid_search.fit(X_train, y_train)

print("Best parameters:", grid_search.best_params_)
```

```
Best parameters: {'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 150}
```

Slide 10: XG Boost Model Cont.

```
#Fit the model with the optimized parameters

start_time = time.time()

model = xgb.XGBClassifier(n_estimators=150, learning_rate=0.1, max_depth=10)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
# Create dataframe that shows features to importance
df_features = pd.DataFrame({"Features": X.columns, "Importance": model.feature_importances_})


# Ai prompt: How to sort dataframe
df_features = df_features.sort_values(by="Importance", ascending=False)
```

```
# Find level of importance by columns in original dataframe
df_features['Category'] = df_features['Features'].str.split('_').str[0]

category_importance = df_features.groupby('Category')['Importance'].sum().reset_index()

category_importance = category_importance.sort_values(by='Importance', ascending=False)

print(category_importance)
```

Slide 11: Structure Variables By Importance

	Category	Importance
6	* Exterior Siding	0.349417
9	* Roof Construction	0.188227
5	* Eaves	0.105213
16	Structure Defense Actions Taken	0.074437
15	Structure Category	0.073792
10	* Structure Type	0.055231
7	* Fence Attached to Structure	0.033774
13	Distance - Propane Tank to Structure	0.018481
11	* Vent Screen	0.017387
3	* Deck/Porch Elevated	0.015708
12	* Window Pane	0.015121
14	Distance - Residence to Utility/Misc Structure...	0.014398
4	* Deck/Porch On Grade	0.011427
1	# of Damaged Outbuildings LT 120 SQFT	0.008693
8	* Patio Cover/Carport Attached to Structure	0.008445
2	# of Non Damaged Outbuildings LT 120 SQFT	0.006937
0	# Units in Structure (if multi unit)	0.003313

Slide 12: Model Evaluation

Linear Regression
- R-squared: 0.1814
- Mean Squared Error: 0.0494

Random Forest
- Accuracy: 0.8388

XGBoost
- Accuracy: 0.8556
- Runtime: 5.5 sec.

XGBoost Optimized
- Accuracy: 0.83
- Runtime: 1.9 sec.

Slide 13: Conclusion

- Wildfire likelihood and coverage by county
- Precipitation $>$ Temperature
- Real life application