Understanding California Wildfires: A Machine Learning Approach

Pedro Galarza, Jack Epstein, Sara Price, and Kieran Sim

Abstract

Wildfires have become an increasingly normal part of life for Californians causing environmental, economic and emotional damage throughout the state. Given that widespread devastation from massive wildfires will only increase with climate change, an understanding of location and potential severity of fires across the state is critical. Preemptive action against spontaneous fires could improve fire management and outcomes. In this paper, we use a two-stage classification model to identify areas at risk for wildfires of different magnitudes in the next month. The proposed model is driven primarily by data capturing weather conditions and historical fire risk. This predictive mechanism could enable more efficient allocation of limited resources by the California state parks and other forest fire fighting organizations.

I. Subject Understanding

California's natural climate makes it prone to wildfires with an expected hot and dry 'fire season' ranging from May through November. In fact, regular and modest-sized wildfires are an important mechanism for wildfire prevention; smaller burns prevent vegetation build up that can lead to larger and uncontrolled burns. However, largely due to climate change bringing longer droughts and fire seasons, wildfires in California are becoming a more common and increasingly dangerous threat to communities throughout the state. Forest fires cause extreme damage to property and the environment and pose high risks to human health and life.

The acreage burned by wildfires in the United States has steadily grown over time from an average of 3 million acres a year in 1980 to over to 8 million acres a year in 2018 as shown in *Figure 1* [1]. Within the United States, California has the largest number of wildfires [2], and the 2020 California wildfire season has been the largest in recorded history: wildfires have burned over 4 million acres or over 4% of the state's total land [3].

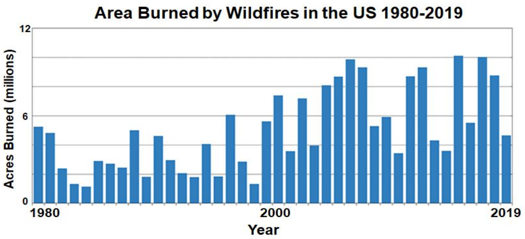


Figure 1

The California fire emergency fund has grown by a factor of 16 over the last 30 years [4] to provide requisite resources in managing these disasters. As a result, policy makers and local governments have demonstrated enthusiasm for machine learning solutions developed in both industry and academia to help constrain, predict and recognize wildfires [5, 6]. Current state of the art machine learning tools for wildfire control tend to focus on two specific applications of fire prediction: physical-spread and early detection. Spread models apply the laws of physics to predict direction and spread of active fires in real time while early detection models often deploy image recognition techniques on satellite images to recognize ignition sites [7]. While not as prevalent, there has also been work to predict wildfire spread given ignition site and on-the ground conditions [8].

Here we propose a model that functions more as a risk assessment tool for communities and local fire prevention authorities. By leveraging historical and on-the-ground data, this model predicts not only which areas of the state are at highest wildfire risk in the next month but also, for those at-risk regions, what the likely magnitude of the predicted fire will be. At a minimum, this model would provide state and local authorities a mechanism for preparing and informing their communities.

Ultimately, handling fires comes down to prevention and containment. This model could also help address the former by allowing state and local authorities more efficiently allocate limited resources to mitigate and prevent spontaneous fires from reaching devastating magnitudes.

II. Data Understanding

A. *California Wildfire Data*

The primary source for historic spontaneous (i.e. non-prescribed) wildfire data is California's Fire and Resource Assessment Program (FRAP) [9], which contains GIS wildfire perimeter data from 1879-2019. From this data, we extract details including cause, fire start and end dates and GIS geometries indicating location and size. For our modeling dataset, we limit to data from 1990-2019 due to degradation in quality and accuracy of data from earlier timeframes.

B. *Feature Data*

Analysis of the wildfire data revealed that most common historical causes of wildfires fall into two categories: environmental triggers and human activity, which include equipment use, arson, campfires, etc. We therefore collect historical data on weather, topography, demography, infrastructure, and arson crimes to cover or create proxies for these known common causes.

We hypothesize that current and past weather conditions are a significant predictor of future fires. Therefore, we leverage the ERA5-Land data set from the Copernicus Climate Data Store [10] which uses a combination of climate observations and physical modeling to build accurate descriptions of hourly meteorological characteristics at a 9km naive spatial resolution. This data ensures high-resolution weather data for all regions of the state. In order to generalize across seasonal trends while capturing time of day variability, we access monthly average measurements at four evenly spaced points in the day for temperature, wind speeds, precipitation, surface pressure, humidity and vegetation.

Given the diversity of California's landscape, which ranges from mountains to coastlines to deserts, we also hypothesize that topography is influential in modeling wildfire spread and size. We include elevation data from the USGS [11], which has a map of contour lines across the whole state.

Finally, we source demographic data including population density from the US Census [12], infrastructure data, such as presence of roads and powerlines, which serve as a proxy for human activity from the California Energy Commission [13,21], and arson crime data from the California DOJ [14] to supplement our model.

III. Data Exploration and Preparation

A. Grid Specification

A key consideration when modeling GIS data is spatial and temporal segmentation. We initially considered predicting wildfire risk at the county level but realized this would impose challenges and result in less useful recommendations given the wide disparity in county size across the state. Therefore, we implement a grid segmentation strategy that allows us to model equal-sized areas across the state with inevitable variation at the state's borders (see *Figure 1* which shows historical wildfires overlaid on the generated grid). This resulted in 133 separate sections each covering roughly 1900 mi^2.

For temporal segmentation, we predict at the month level because when considering model deployment, shorter time frames would not provide enough time for agencies and communities to effectively use model outputs for preparation and prevention. The final dataset contains approximately 48k unique observations representing each grid section (which we also refer to as grid ID) and month combination between 1990-2019.

B. Target Variable Assignment

To create viable target variables and identify which fires occurred in which grid IDs and for how long, we project wildfire geometries onto the grid. We begin with a base data frame indexed on unique combinations of grid ID and date between 1990-2019. Leveraging GeoPandas spatial

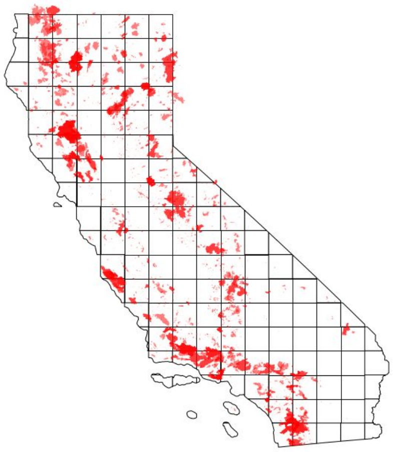


Figure 2: Historic CA Wildfires

operations, we identify areas of intersection for each wildfire and grid section. Then for each wildfire that occurred in a given grid section, we add features associated with that fire to the base dataframe for all days between the fire start and end dates. These features include area of fire and grid section overlap and total wildfire area. Using total wildfire area, we define a new categorical target variable called fire size class, which buckets wildfires by acres burned using the National Wildfire Coordinating Group (NWCG) Fire Size Class Code [15] (see *Appendix A* for detailed class size definitions).

We ultimately aggregate the daily fire data by month and grid ID to create 3 target variable options in *Table 1.*

Target	**Modeling Strategy**
1. Binary: 1 if a new fire starts or spreads into a grid section in given month	Binary Classification
2. Multi-class: max fire size class of new fires starting or spreading in grid section in a month	Multi-class Classification
3. Continuous: proportion of a grid section experiencing fires in a given month	Regression

Table 1: Target Variable Options

We use final fire areas to create options 2) and 3) given the raw fire data only contained total area burned. In order to avoid leakage, target variables only represent new fires in a given month for each grid ID because we are not interested in predicting if an existing fire will continue into the current month but instead how likely a fire is to begin in the current month.

Fire	Observations	Representation in Data
0	44,213	91.65%
1	4,027	8.35%

Table 2: Binary Target Variable Distribution

Fire Size Class	Observations	Representation in Data
0	44,213	91.65%
1	1,838	3.81%
2	1,103	2.29%
3	1,086	2.25%

Table 3: Multi-class Target Variable Distribution

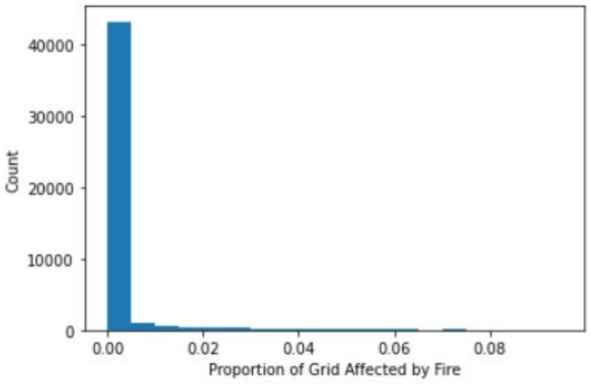


Figure 3: Regression Target Variable Distribution

C. *Feature Transformation*

Temporally, we aggregate weather and infrastructure features by month and demographic, topographic, and arson data by year. Most of the data was already aggregated in its raw form. Topographic data was not, so from the full contour map, we calculate mean, median, standard deviation and range of elevations in each grid ID. The standard deviation and range of elevations in a grid ID serve as proxies for some element of steepness, which our research indicated could impact how fires spread [16].

When aligning features with targets, we time-shift monthly features back one month and the annual features back one year to avoid target leakage. In the case of weather data, we recognize that this may cause some loss in predictive power for fires occurring at the end of the month. However, we hypothesize that aggregate weather conditions are more deterministic for fires than daily or weekly fluctuations. For instance, one month without rain is more indicative of drought-like conditions conducive to wildfires than one week or day without rain.

Geographically, demographic, infrastructure, and arson data came at the county level and so we assigned each grid the same value as that of the county it was in.

Many features are on different scales, so standard scaling before using linear models was critical.

D. Feature Engineering

After observing baseline model performance (see *Section IV.* A. *Baseline Models*), we note that many weather features rank high on entropy and gini feature importance. Therefore, we create features capturing average historical weather conditions in the previous year, five years and ten years to gain additional leverage from these features. In the original dataset we have 24 weather features, so creating these historical features added 72 new features.

We also create 12 new features to capture the increased likelihood of fire due to the historic presence of fire over different time periods or due to spread of fire from adjacent grids. See *Appendix B* for detailed description of historical fire features. Since these features are derived from historic versions of our target variables, we are especially careful to avoid target leakage by only counting historic fires that have finished burning.

E. Test-train Split

Our modeling dataset is longitudinal, so we cannot use a traditional random test-train split on the data from 1990-2019. Therefore, for the majority of modeling, we simply allocate data from 1990-2015 for training and data from 2016-2019 for testing. In order to ensure robustness from the model selection and hyperparameter tuning process, we do use a time-series specific form of cross validation (see *Cross Validation*).

Due to our temporal based train-test split, we acknowledge the potential of concept drift due to factors such as climate change or new policies introduced in recent years. We make a fundamental assumption that the concept drift plays a minimal role in the features and importance of features in predicting new fires.

IV. Model and Evaluation

A. Baseline Models

We fit baseline models for each of the three potential target variables (see *Table 1*) to identify the most promising approach for further iteration. *Table 4* details performance for models fit without hyperparameter tuning, feature selection, or addition of engineered features discussed in *Feature Engineering*. We do scale features for linear models and use balanced class_weight for sklearn's Logistic Regression to prevent the classifier from predicting only 0s due to high class imbalance.

Model Class	Model Type	Performance
Binary Classifier	Log. Regression	AUC : 0.81 Recall : 0.75
	Random Forest	AUC : 0.83 Recall : 0.22
Multi-Class Classifier	Log. Regression	Mean class AUC: 0.79 Large fire recall: 0.43
	Random Forest	Mean class AUC: 0.5 Large fire recall: 0.07
Regression	Linear Regression	Adjusted R^2: -9.84e+19

Table 4: Baseline Model Performance

Classifiers clearly far out-perform regression with the logistic regression binary classifier showing the most promising results on both recall and AUC.

For the regression model, despite iteration attempts performing feature selection, testing different transformations of the target variable, and using other types of models including random forest regression and ridge regression, performance was consistently poor. A potential reason for this is because of the extreme left-skew (see *Figure 3*) of our continuous target variable which persisted even after ensembling and variable transformations. Because of this level of performance, we focus the remainder of this report on classification.

B. Evaluation Metrics

For both the binary and multi-class problems, we assume that a false positive is less harmful than a false negative. False positives could result in potential wasted resources or unnecessary evacuation orders while false negatives, particularly in the case of class 2 or 3 fires, could result in massive unnecessary destruction and loss of human life. Further, for the multi-class problem, we need to consider relative costs of incorrectly predicting wildfires of different sizes. We make the assumption that costs associated with a false negative for a class increase proportionally to the size of fires in that class. For instance, predicting 0 when the true wildfire class 3 would be more harmful than if the true wildfire class is only 1.

Therefore, while we use traditional evaluation metrics including AUC and recall, we also develop cost-sensitive evaluation frameworks for the binary and multi-class problems. *Appendix C* provides detailed methodology descriptions for the binary and multiclass cost matrices. We use these cost matrices to calculate the following expected value metric:

<u>Binary Expected Value</u>

$$EV=P(1)[TPR*V_{TP}+(1 - TPR)*C_{FN}]+ \\ P(0)[FPR* C_{FN}+(1 - FPR)*V_{TN})$$

where V_{TP} and V_{TN} are values for true positive and true negative and C_{FP} and C_{FN} are costs for false positive and negative

We extend the above equation to a multi-class setting by multiplying the rows of a normalized confusion matrix, which correspond to predictions for each class, by the corresponding rows of the multi-class cost matrix in *Appendix C.*

We only assign negative cost values to the matrices and no positive value to correct predictions. Correctly predicting and hopefully preventing a fire does not generate explicit value beyond the status quo; it merely helps prevent damage. Even then, there is no guarantee that correctly predicted fires will result in successful prevention. The cost of incorrectly predicting that damage is already captured in the high false negative costs.

C. Logistic Regression and Linear SVM

For the baseline models, logistic regression clearly outperforms random forest particularly on recall, so we hypothesize that linear classification models like logistic regression and linear SVC will best capture the target-feature relationship. We still iterate on random forest by testing different numbers of trees, balancing class weight and using traditional down-sampling on the negative class and also test other non-linear models (see section *Nonlinear Modeling Approaches*). We were unable to significantly improve fire recall using random forest (see *Appendix E* for final outcome). However, feature importance rankings from random forest models were critical in feature selection for linear classification models.

Binary classifiers demonstrate better performance than multi-class classifiers in initial model development; however, we acknowledge that utility from binary fire predictions is lower than those from a multi-class model. Therefore, we fit multi-class versions of most attempted binary models to assess performance.

Linear classification models have relatively few hyperparameters to tune, so feature selection was a central focus for model improvement.

i. Feature Selection

Including engineered features, our full modeling dataset contains 172 features, many of which are weather features, which can be highly correlated. This creates concern for multicollinearity and high variance estimates in linear models, so we propose an iterative feature selection approach to maximize predictiveness with the target variable and minimize correlation between features. The below algorithm takes a sorted feature importance list output from a random forest model, K (number of features to select), and ρ (maximum allowable correlation between features) as inputs:

```
1) Initialization
      Create empty list L for storing selected features
2) First pass
      Select feature with highest importance from input list $X_1$
3) Iterative feature selection
   While length(L) < k:
```

a) From remaining features in input list, select feature with highest importance X_i
b) Compute pairwise correlation between X_i and features in L
c) If all pairwise correlations for $X_i < \rho$ then add X_i to L

We used a "bake-off" framework to test logistic regression and SVM model performance on different subsets of features selected using the above algorithm with Ks ranging from 20-45. We also fit random forest models using entropy and gini as splitting criteria to generate different feature importance lists to pass the feature selection algorithm. Top features remained relatively constant regardless of the specifications of the random forest model as shown in *Appendix D,* which compares the top 10 features output from the feature selection algorithm using gini and entropy..

ii. PCA for Latent Factors

From the feature importance lists, weather and historical fire features appear most predictive of wildfires. We hypothesize there could be latent factors in weather and environmental conditions that are not captured by the features alone but could be through dimensionality reduction using PCA. Thus we also test PCA as a feature selection method for logistic regression and linear SVC. We fit models on different numbers of principal components ranging from [5-40]. We include best performance for these PCA tests on binary and multi-class classifiers in *Appendices E and F.*

For logistic regression and linearSVC models, PCA increased recall for binary modes by roughly 0.05 while expected value was slightly worse. This improvement over non-PCA feature selection was too small to justify losing the benefit of the interpretability of logistic regression.

D. Cross Validation for Hyperparameter Tuning

After identifying optimal feature selection for logistic regression and linear SVC, we use cross-validation as a robust approach to tune the regularization hyperparameter for both models. We use a time-series specific algorithm called sliding-window cross validation that allows us to sample subsets from the training data while respecting the longitudinal nature of the data [17]. Our method evaluates model performance on six-year subsets of the training data against a 3-year validation set that follows the training. *Figure 4* depicts performance on the binary expected value metric using cross validation tested across C

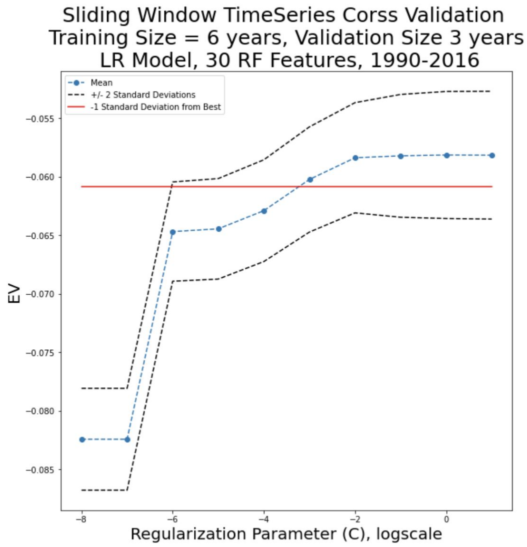


Figure 4:Regularization Tuning w/ Cross Validation

values from [$1x10^{-4}$, 100] for logistic regression. Per standard practice, we choose the highest regularization that is within one standard error of the max expected value. *Appendix I* contains similar plots using AUC and recall to assess if hyperparameter selection changes based on the evaluation metric. Ultimately it did not. Results for the best binary logistic regression and linear SVC models after feature selection and hyperparameter tuning are in *Appendix E*.

E. Non-linear Classification Approaches

As previously noted, we also test decision trees, gradient boosted decision trees and k-nearest neighbors (KNN).

For decision trees and gradient boosted decision trees, we aim to see if results are somewhat consistent with our initial random forest testing. For decision trees, we test on an exponential range of both minimum leaf lizes [2, 1024] and min_split sizes [16, 2048] to cover under and overfitting scenarios. We also test using all features vs the top 15 features based on both gini and entropy feature importance. In gradient boosting tests, we test different numbers of trees [100,200,1000] and also test both the SKLearn and XGBoost packages. We use the out of bag hyperparameters on individual trees. As shown in *Appendices E, F*, results are similar to random forest testing and do not justify further fine-tuning relative to the more successful linear approaches.

We test KNN under the hypothesis that while rare, positive instances may exhibit similarities. We test k's from 1 to 20 with Euclidean distance and generally observed declining recall with increased k. Similar to the tree-based models, KNN struggles with recall given the limited positive instances. Thus, increasing k, while improving AUC, only exacerbates the recall issue and allows more negative instances to influence all decisions on the testing data. As an attempt to mitigate the class imbalance, we test only on data from May-November, California's fire season, but see limited increases in recall and lower expected values (see *Appendix E*).

Thus, we conclude the logistic regression classifier is the optimal binary classifier with AUC = 0.88, recall = 0.84, and expected value = -0.05.

F. Multi-class Ensemble

With this strong binary classifier, we revisit the problem of size classification to see if a phased approach can improve on results in *Appendix F.* We hypothesize that binary predictions from a "first-stage" classifier could be used as a feature in a "second-stage" multi-class classifier.

To avoid target leakage, we split the data into three time ranges: 1) 1990-2005 as training data for a stage 1 binary model; 2) 2006-2015 as training data for a stage 2 multi-class model; 3) 2016-2019 as the test set for the full ensemble. We train the best binary classifier, logistic regression with regularization C and 30 selected features, on stage 1 data. We then use this model to score phase 2 data and add the resulting prediction as a new feature in the stage 2 dataset. Using this augmented stage 2 data, we train and test logistic regression and linear SVC models for multi-class classification given these are the top performing multi-class classifiers on the full training data (*see Appendix F).*

We also test a different two-stage approach that uses binary predictions to filter the data used in training the stage 2 model. This was less successful than including predictions as a feature (see *Appendix G*).

We did have concerns around losing predictive power when using a smaller stage 1 training population. However, we are able to demonstrate in *Figure 5,* that models trained on sliding 3-year windows from 1990-2016 have fairly robust performance compared to models trained on the full 1990-2016 population.

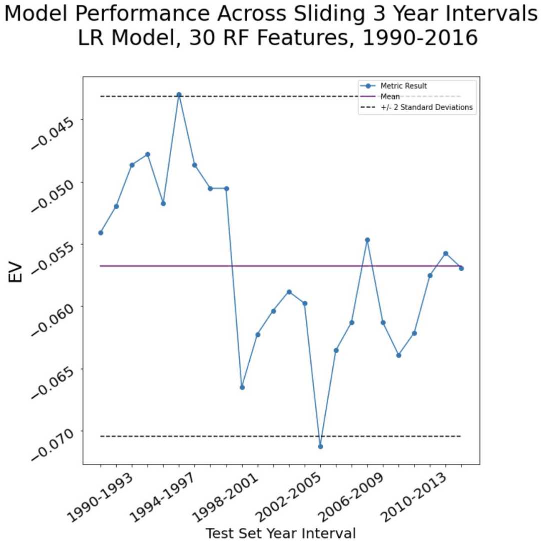


Figure 5: Model Variability Across Training Subsets

Furthermore there is no significant model improvement as we increase the amount of years in the training set (see appendix K). Ultimately we decide the performance difference is small, and is well worth the net benefit of improved multi-class predictions.

As shown in *Appendix G*, the winning model for the multi-stage approach is the stage 1 binary logistic regression paired with stage 2 multi-class logistic regression, which leads to a 11% increase in expected value over the best single-stage multi-class model. We acknowledge that recall for fire size 3 is only 50%, so this is an area for improvement in future development.

V. Deployment

A. Implementation

Our model makes predictions at the monthly level, so in practice, it would be run at the end of each month to predict areas of highest fire risk in the upcoming month. Many features including demographics, infrastructure, topography and crime, are pseudo-static and only update once a year so these features can be prepared in advance. At serving time, we will need to pipe in weather forecasts for the future month from Copernicus and calculate the latest engineered features from the previous month. Once fire predictions are made, they will be passed onto the state, specifically the California Department of

Forestry and Fire Prevention [18], who would determine resource deployment.

B. *Model Retraining*

To avoid concept drift and ensure our model leverages new research on wildfire causes and risks, we would want to retrain the model at regular intervals. We recommend retraining the model annually to align with when annual feature datasets are refreshed. We can also actively monitor model performance month to month and proactively retrain our model off cycle if our evaluation metrics fall below a certain threshold.

C. *Ethical Considerations*

Our biggest ethical consideration revolves around evaluation metrics. We deliberately choose a utility instead of dollars-based framework when calculating costs and expected values. We feel that wildfire prediction is not exclusively an economic issue. There are many non-monetary costs to uncontrolled wildfires such as increased potential health impacts from degradations in air quality or loss of biodiversity that we wanted to capture in our cost framework. We acknowledge that while it is easier to put a value on property damage, we wanted to ensure we do not have a system that prioritizes resources for the richest areas. Not only would this be morally wrong, it would also be impractical given certain wealthier areas are already using additional private resources to help fight fires. [19]

To determine the practical effectiveness of our model, we calculate a misclassification cost for each grid using predictions for 2016-2019. We estimate the misclassification cost, C, using the following formula:

$$C \ = EV \ \ \cdot log(popdens)$$

EV is our multi-class expected value defined in *Section IV. B.*
popdens is the average population density in count of people per km^2 of the county containing the grid

We use population density as a proxy for the potential damage a fire in the area could do, and apply a log transform to temper the weight of this feature (see *Appendix J* for cost map unweighted by population density).

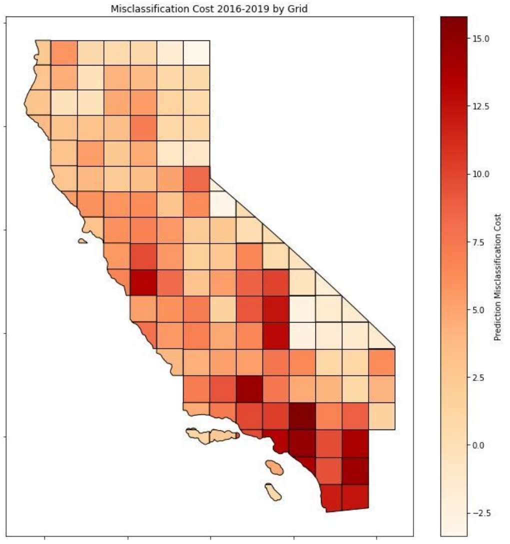

Misclassification Cost 2016-2019 by Grid
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
Prediction Misclassification Cost

Figure 6: Misclassification Cost Projection

D. *Future Work*

If we had more time, we would aim to continue to improve recall on bigger fires given that the two-stage approach still only achieves 50% recall on the largest fires. Possible next steps would include segmenting at a more granular temporal level than by month. We acknowledge that aggregating monthly could miss potential influential weather patterns particularly for features like wind that are less cumulative in nature.

Additionally, the fire data we used was the most complete historical source for spontaneous wildfires; however, it did not include features like point of ignition or speed of spread that could be influential in predicting fire size. However, features like these would likely require satellite data and possibly different model formulations altogether to incorporate.

.

VI. References

[1] National Interagency Coordination Center. "Total Wildland Fires and Acres." *National Interagency Fire Center*, 2019, www.nifc.gov/fireInfo/fireInfo_stats_totalFires.html.

[2] Insurance Information Institute. "Facts + Statistics: Wildfires." *III*, 2020, www.iii.org/fact-statistic/facts-statistics-wildfires.

[3] California Department of Forestry and Fire Protection (CAL FIRE). "Stats and Events." *Cal Fire Department of Forestry and Fire Protection*, 2020, www.fire.ca.gov/stats-events/.

[4] Armstrong, Martin, and Felix Richter. "Infographic: The Spiralling Cost of California's Wildfires." *Statista Infographics*, 11 Sept. 2020, www.statista.com/chart/19807/california-wildfire-emergency-fund-expenditure/.

[5] Said, Carolyn. "Artificial Intelligence Is Helping to Spot California Wildfires." *Government Technology State & Local Articles - E.Republic*, San Francisco Chronicle, 1 Sept. 2020, www.govtech.com/products/Artificial-Intelligence-Is-Helping-to-Spot-California-Wildfires.html.

[6] Holley, Peter. "California Has 33 Million Acres of Forest. This Company Is Training Artificial Intelligence to Scour It All for Wildfire." *The Washington Post*, WP Company, 18 Nov. 2019, www.washingtonpost.com/technology/2019/11/06/california-has-million-acres-forest-this-company-is-training-artificial-intelligence-scour-it-all-wildfire/.

[7] Radke, David. "FireCast: Leveraging Deep Learning to Predict Wildfire Spread." *International Journal of Wildland Fire*, 2019, doi:10.1071.

[8] "GOES Early Fire Detection (GOES-EFD) System." *CSTARS D3Science*, 2020, www.cstarsd3s.ucdavis.edu/systems/goes-efd/.

[9] California Department of Forestry and Fire Protection (CAL FIRE). "Fire Perimeters." *Cal Fire Department of Forestry and Fire Protection*, 2020, frap.fire.ca.gov/frap-projects/fire-perimeters/.

[10] "ERA5-Land Hourly Data from 1981 to Present." *Copernicus*, 2020, cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-land?tab=overview.

[11] "Coastal Changes and Impacts." *Contour & Preliminary Contour Data*, 2020, www.usgs.gov/core-science-systems/eros/coastal-changes-and-impacts/contour-preliminary-contour-data.

[12] Bureau, US Census. "County Population by Characteristics: 2010-2019." *The United States Census Bureau*, 22 June 2020, www.census.gov/data/tables/time-series/demo/popest/2010s-counties-detail.html.

[13] "California Electric Transmission Lines." *California Energy Commission*, 2020, cecgis-caenergy.opendata.arcgis.com/datasets/260b4513acdb4a3a8e4d64e69fc84fee_0/data.

[14] DOJ, Open Justice. "State of California Department of Justice." *Crimal Justice Data*, 2020, openjustice.doj.ca.gov/data.

[15] National Wildfire Coordinating Group. *NWCG Data Standard Fire Size Class Code*. 14 Nov. 2009, www.nwcg.gov/sites/default/files/data-standards/pdf/values.pdf.

[16] National Wildfire Coordinating Group. "8.7 Slope Effect on ROS." *NWCG*, www.nwcg.gov/course/ffm/fire-behavior/87-slope-effect-on-ros.

[17] Samuel, Pradip. Medium, 14 Jan. 2020, medium.com/@pradip.samuel/cross-validation-in-time-series-model-b07fbba65db7.

[18] California Department of Forestry and Fire Protection (CAL FIRE). "About Us." *Cal Fire Department of Forestry and Fire Protection*, 2020, www.fire.ca.gov/about-us/.

[19] Varian, Ethan. "While California Fires Rage, the Rich Hire Private Firefighters." *New York Times*, 26 Oct. 2019.

[20] "Wildfire Management vs. Fire Suppression Benefits Forest and Watershed." *Berkeley News*, 24 Oct. 2016.

[21] "USA Road Density" *Environmental Systems Research Institute,* Updated 2019. https://landscape3.arcgis.com/arcgis/rest/services/USA_Roads/ImageServer

VII. Contributions

All four of us contributed to aspects of data collection, clean up, modeling and writing. More specifically

Sara Price: Fire data gathering and cleaning; grid and target variable generation and specification; historical weather and fire feature engineering; creation of feature selection algorithm; baseline, binary and multi-class linear model testing and training; creation of cost matrix and expected value framework

Pedro Galarza: Business understanding/Domain Research; weather data gathering and processing and mapping; Model selection and validation (cross-validation, model variance, learning curves).

Kieran Sim: Feature engineering; demographic and crime data gathering and cleaning; regression model testing; general paper outlining

Jack Epstein: Topography and infrastructure data gathering and cleaning; modeling aggregate dataframe set up; ensemble multi-class and non-linear modeling

Appendix

A. Fire Size Class Definitions

fire size class Specification	
Fire Class	**Total Acres Burned**
0	No fire
1	Greater than 0 and less than or equal to 100
2	Greater than 100 and less than or equal to 1,000
3	Greater than 1,000 and less than or equal to 5,000

B. Descriptions of engineered features capturing historical and local wildfire risk

Engineered Features	
Feature Description	**Data Type**
Was there a fire that ended in the previous month?	Binary (0,1)
Was there a fire in the previous year? 5 years? 10 years? (3 separate features)	Binary (0,1)
Maximum size of a fire that ended in the previous month (using the fire size class definition in Appendix A)	Categorical (0, 1, 2, 3)
Maximum size of fire occurring in the previous year; 5 years; 10 years (3 separate features)	Categorical (0, 1, 2, 3)
Proportion of the area of a given grid section that experienced a fire in the previous month	Continuous (bounded 0,1)
Proportion of the area of a given grid section that experienced a fire in the previous year	Continuous (bounded 0,1)
Was there a fire in one of the 8 adjacent grids last month that was not also in the central grid last month?	Binary (0,1)
Count of fires in any of the 8 adjacent grids last month that were not also in the central grid last month	Discrete

C. Cost Matrices for Multi-class and Binary Classification

Given we are not subject matter experts on costs of wildfires, the actual values in these cost matrices are somewhat arbitrary. However, our methodology described below intends to assign costs roughly proportional to the magnitude of damage for each false positive or false negative prediction.

Binary Classification Cost Matrix

		Predictions	
		No fire (0)	**Fire (1)**
Actuals	**No fire (0)**	0	-0.1875
	Fire (1)	-0.75	0

For the binary classification problem we impose a higher cost for false negatives than false positives and use a heuristic that costs from a false negative are roughly four times as high as those for a false positive.

Multi-Class Classification (fire size class)

		Predictions (fire size class)			
		0	**1**	**2**	**3**
Actuals (fire size class)	**0**	0	-0.25	-2.5	-12.5
	1	-1	0	-0.225	-0.245
	2	-10	-0.9	0	-0.05
	3	-50	-0.98	-0.2	0

Costs for predicting no fire when there is a fire

We assign baseline costs for incorrectly classifying different fire classes as 0 proportional to the minimum acres burned for each fire class (see *Appendix A* for class size definitions).

Costs for predicting a fire when there is no fire:

We use the same heuristic here as in the binary classification case : the cost of a false positive for a given class is assumed to be 1/4th as detrimental as a false negative. So the cost for a false positive with class 1 is -1*0.25.

Costs for predicting a fire but underpredicting size:

We assume predicting a wildfire correctly but class size incorrectly is less costly than predicting no fire at all. We assume a fire that is 100 acres will receive roughly 1/10 of the resources of a fire that is 1000 acres. So we use the following formula when the model predicts a fire is size i when it is really size j and j > i:

$$Cost \ = \ (1 \ - \frac{C_i}{C_j} \ )$$

where C_i= cost of predicting no fire for a fire of size i and C_j = cost of predicting no fire for a fire of size j

Costs for predicting a fire but overpredicting size:

We use the same heuristic of a false positive having a cost 25% of that of a false negative. Thus we use the formula laid out for underpredicting size and multiply that outcome by 0.25. So we use the following formula when the model predicts a fire is size j when it is really size i and j > i:

$$Cost \ = \ 0.25 \ * (1 \ - \frac{C_i}{C_j} \ )$$

where C_i= cost of predicting no fire for a fire of size i and C_j = cost of predicting no fire for a fire of size j

D. Top Feature Importance Comparison

We tested different random forest models as generators of feature importance rankings, which are used as inputs for our feature selection algorithm. Below is a comparison of the top 10 features selected using the selection algorithm using different initial random forest feature importance lists. While the ranking is somewhat different, many of the variables are common between the two random forest feature importance ranking approaches.

Top 10 Features Selected using Class-Balanced Random Forest Model using Gini	Top 10 Features Selected using Class-Balanced Random Forest Model using Entropy
Proportion grid section that experienced a fire in the past year	Average temperature over the past 10 years
Proportion of grid section that experience a fire in the past month	Proportion grid section that experienced a fire in the past year
Average temperature over the past 10 years	Count of fires in the adjacent grids in the previous month
Count of fires in the adjacent grids in the previous month	Proportion of grid section that experience a fire in the past 10 years
Average precipitation over the past 5 years	Average precipitation over the past 10 years
Maximum fire size occurring in a grid section in the past year	Maximum fire size occurring in a grid section in the past year
Average proportion of grid section that experienced a fire in the past 5 years	Average precipitation in the past month
Maximum fire size occurring in a grid section in the previous month	Maximum fire size occurring in a grid section in the past five years
Average precipitation in the previous month	Average precipitation over the past year
Average wind in the previous month	Proportion of grid section that experience a fire in the past month

E. Binary Model Performance

Below are the best performances for the different models we trained and tested after iterating to find optimal hyperparameters and model specifications (winning model based on expected value is bolded)

Model	Feature Selection	Hyper-parameter(s)	AUC	Recall	Expected Value
Logistic Regression	**Top 30 Random forest entropy-determined feature importance**	**Class _weight = 'balanced'** **C = 0.001 (determined used CV)**	**0.88**	**0.84**	**-0.050**
Linear SVC	Top 30 Random forest entropy-determined feature importance	Dual = False Class_weight = 'balanced' C = 0.001 (determined used CV)	0.88	0.86	-0.052
Logistic Regression	First 15 principal components	Class_weight = 'balanced' C = 0.001 (determined used CV)	0.87	0.88	-0.058
Linear SVC	First 25 principal components	Dual = False Class_weight = 'balanced' C = 0.0001 (determined used CV)	0.87	0.91	-0.059
Random Forest	All Features	Criterion = 'entropy'	0.86	0.21	-0.075
Gradient Boosting DT	All Features	Package: SKLearn, n_estimators = 1,000	0.62	0.29	-0.073
Decision Tree	All Features	min_leaf_size =2, min_split_size =32	0.61	0.32	-0.071
KNN	Top 25 from DT feature list	k=1, distance='euclidean'	0.58	0.28	-0.076
KNN - Fire Season Only	Top 25 from RF feature list	k=1, distance='euclidean'	0.60	0.30	-0.114

F. Multi-class Classification Model Performance

Below are the best performances for the different types of multi-class models that we trained and tested on multi-class data. While we note in the report that binary classification became our modeling emphasis, we still did iterate on optimal hyperparameters for the linear multi-class classifiers.

Model	Feature Selection	Hyper-parameter(s)	Mean Class AUC	Size 3 Recall*	Expected Value
Logistic Regression	Top 25 features using RF gini importances	class _weight = ‘balanced’ C = 0.01	0.83	0.52	-0.99
Linear SVC	Top 35 features using RF gini importances	Dual = False Class_weight = ‘balanced’ C = 1	0.86	0.35	-6.38
Logistic Regression	First 35 principal components	Class_weight = ‘balanced’ C = 0.01	0.80	0.52	-0.92
Linear SVC	First 35 principal components	Dual = False Class_weight = ‘balanced’ C = 0.0001	0.85	0.42	-3.17
Random Forest	All Features	N_estimators = 100, criterion = ‘gini’	0.5	0.068	-11.80
Gradient Boosting RF	All Features	Package: SKLearn, n_estimators = 100	0.870	0.06	-11.24
Decision Tree	All Features	min_leaf_size =2, min_split_size =32	0.668	0.19	-9.81

**Note we emphasize fire size class 3 in our metrics since these are the most destructive and therefore most important to predict correctly.*

G. Multi-class Ensemble Model Performance

Models where Phase 1 Prediction is ‘Feature’ use the main methodology discussed in the body of the report. Models where Phase 1 Prediction is ‘Filter’ are trained only on instances in the stage 2 training population (2006-2015) that receive a positive prediction when scored using the stage 1 model (1990-2005).

Model	Phase 1 Prediction	Feature Selection	Hyperparameter(s)	Mean Class AUC	Size 3 Recall[1]	Expected Value
Logistic Regression	**Feature**	**Top 30 from RF feature list**	**class _weight = ‘balanced’** **C = 0.1**	**0.850**	**0.504**	**-0.883**
Logistic Regression	Filter	Top 30 from RF feature list	class _weight = ‘balanced’ C = 10	0.663	0.378	-3.313
LinearSVC	Feature	Top 30 from RF feature list	class _weight = ‘balanced’ C = 0.1	0.669	0.157	-6.452
LinearSVC[2]	Filter	Top 25 from RF feature list	class _weight = ‘balanced’ C = 0.0001	0.881	0.490	-4.380

[1] Note we emphasize fire size class 3 in our metrics since these are the most destructive and therefore most important to predict correctly.

[2] While this LinearSVC test performs relatively well in terms of both big fire recall and average class AUC, the expected value score was hurt by poor performance on class 1 and 2 fires.

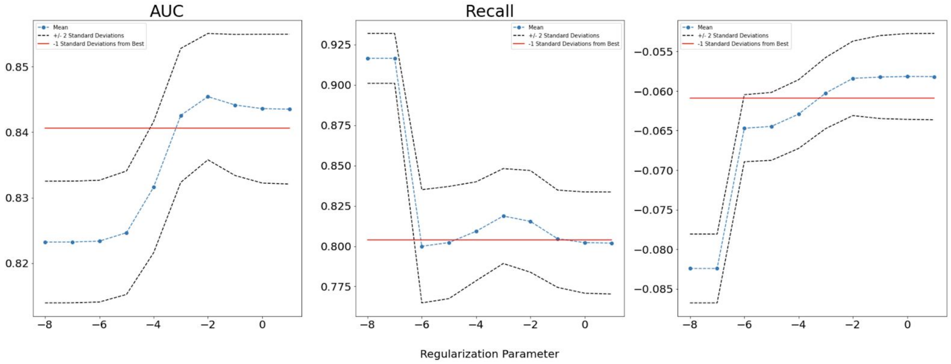
H. Regression Classification Model Performance

Below are the best performances for the different types of regression models that we trained and tested on stage II data. Stage II data was generated by leveraging our already identified optimal binary classifier, the logistic regression model, training it on 1990-2005 data, then passing only the predicted positive instances into our stage II model. On this stage II data set, we iterated through multiple potential regression models, their hyperparameters, and different Y transforms to try to smooth the skewed input.

Model	Y Transform	Hyper-parameter(s)	R2
Linear Regression	None	None	0.180
	log(Y+1)	None	0.197
	$\sqrt{Y}$	None	0.278
Random Forest Regressor	$\sqrt{Y}$	N-estimators = 100	0.258
		N-estimators = 500 Min leaf size = 30	0.290
Ridge Regression	$\sqrt{Y}$	None	0.161

I. Cross Validation Complete Results:

a. Sliding Window Cross Validation; Binary Prediction, LR, 30RF features



b. Sliding Window Cross Validation; Binary Prediction, SVM, 30RF features

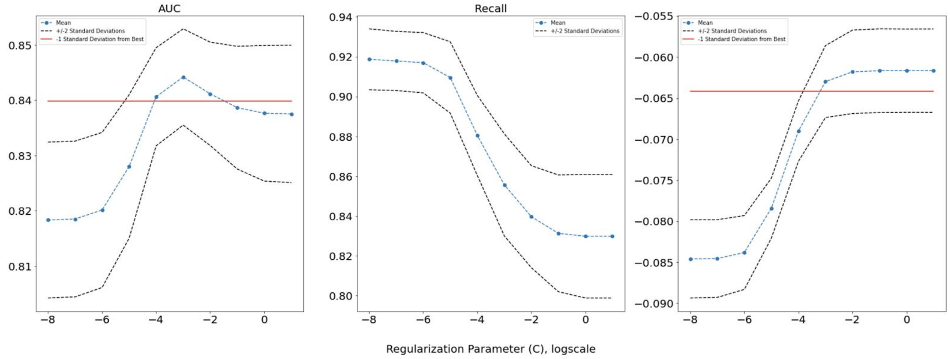
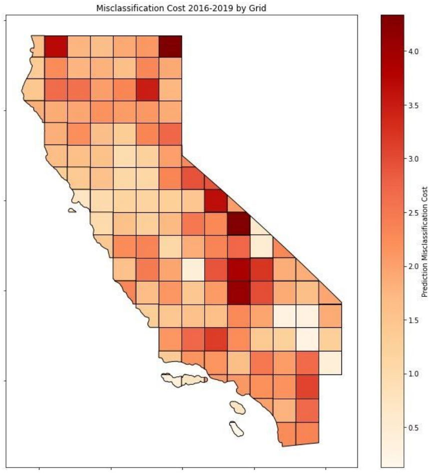


Figure I. Cross validation was performed using a time sensitive sliding window approach. We segment the training data into "folds" of size 9 years. The first six years are used to train the model and performance is validated on the last three. These folds are then slid along the complete training set at 3 year increments yielding 7 cross validation populations across 27 years. These populations are evaluated with regularization parameter values from [1x10-4, 100] (on a log scale). The means and standard errors for each parameter are charted above along with a red line signifying one standard error below the best performing model that is not one of the min/max parameter values. (a) Results from top LR model. Using EV as our deciding metric, we choose C= .001 since it is a lower parameter within one standard error of the maximum value. (b) Results from top SVM model. Using EV as our deciding metric, we choose C = .001 since it is a lower parameter within one standard error of the maximum value.

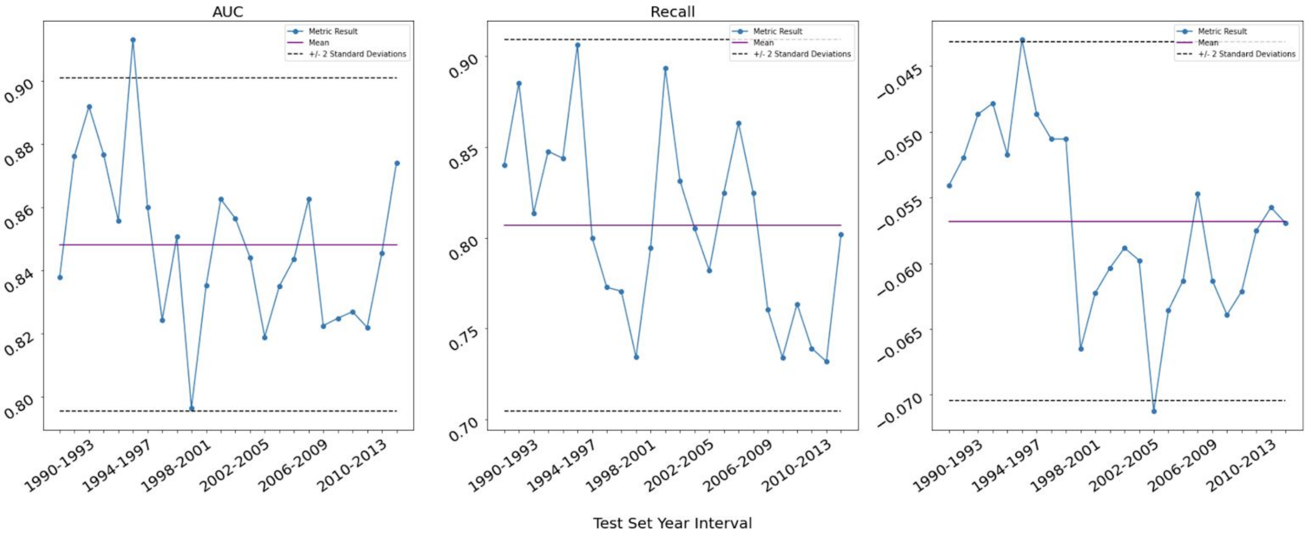
J. Unweighted Misclassification Cost (without Population Density)


Misclassification Cost 2016-2019 by Grid
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Prediction Misclassification Cost

Below is a misclassification cost graph highlighting the grid areas where our model does a weak job in predicting future fires, without weighting by population density. We can see that in this graph, we misclassify areas in the eastern desert region of California, however, this pattern does not appear on the graph in the main paper as we do not place a high weight on misclassifications in regions with low population densities such as this.

K. Learning Curves:

a. Model Performance Across Sliding 3 Year Intervals; LR Model, 30RF Features, 1990-2016



b. Cumulative Learning Curves (1 Year Increments); LR Model, 30RF Features, 1990-2016

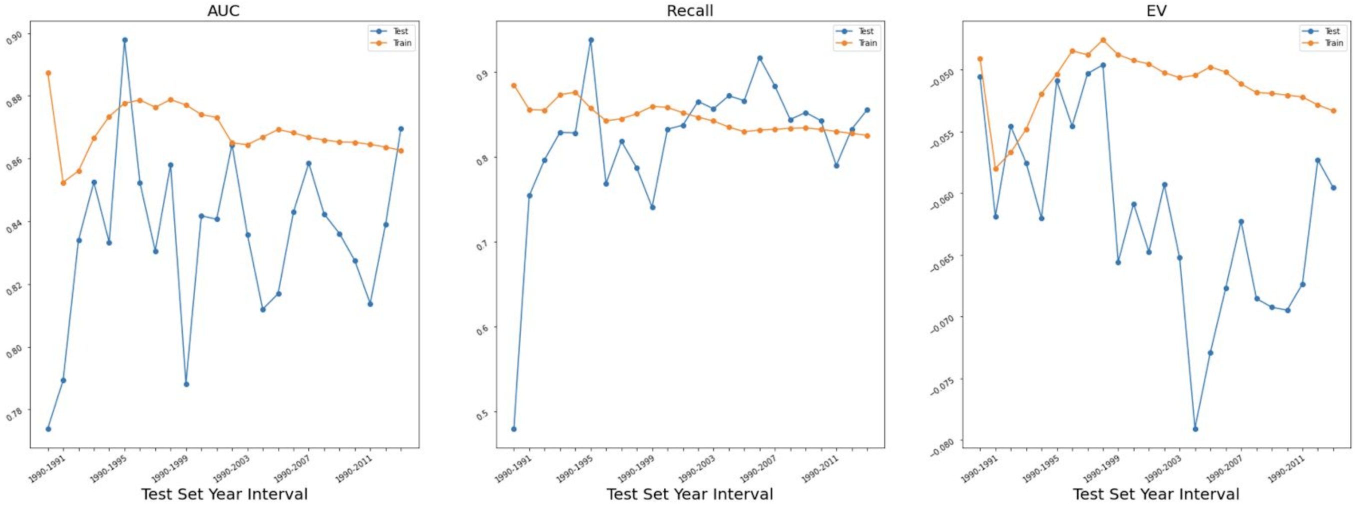


Figure G. We evaluated how training set size affects model performance using two perspectives.. (a) We first segment the training data in subsets of size 4 years and use a 1-year incremented sliding window approach across the entire training population. Within each subset, we evaluate model performance using the first three years as training data and then validating on the last. The result is a demonstration of model variance using a fixed smaller training population. (a) The second method uses a forward chaining approach to evaluate trends in model performance changes as more data is added to the training set. The smallest training set only includes 1990, we increment the size of the training subset by 1-year; our largest training subset ranges from 1990-2014. Each set is evaluated on the three years that follow the last year.