DS-GA 1001 Term Project: Predicting the Severity of Wildfires in the United States

Yuhan Liu, Yuhan Gao, Zhifan Nan, Long Chen
Center for Data Science
New York University
New York City, USA
{yl7576, yg2417, zn2041, lc3424}@nyu.edu

***Abstract*—Wildfire has become a big issue in the United States, causing casualties and substantial economic losses every year. In this project, we aim to predict the fire size of wildfires using machine learning algorithms. The dataset we used contains about 190,000 cases of wildfires in California, Alaska, New Mexico, Nevada, and Texas from 1992 to 2009. We select important features such as the coordinate of wildfire, elevation, land's owner, cause, and fire month by calculating multicollinearity, mutual information, and learning performance curve. We then choose Random Forest as the baseline model and compare performances among other model candidates, including XGBoost, KNN, Multilayer Perceptron, and stacking ensemble model with XGBoost as the meta-model. The stacking ensemble model yields the highest accuracy of 64.51%. Such model can be deployed by fire departments as reference for more efficient allocation of wildfire prevention resources once a wildfire took place.**

***Index Terms*—Wildfire, Classification, Hyperparameter Tuning, Stacking Ensemble Model, EDA, Feature Selection, Feature Engineering**

I. Introduction

Wildfire has become a huge issue around the United States as it has destroyed millions of houses and families every year. In 2019, wildfires in California alone scorched nearly half a million acres, caused economic losses of an estimated \$80 billion[1], according to AccuWeather. With rapid climate change, severe thunderstorms, as well as many human factors, the scale, intensity, and the frequency of wildfires in many states including California, Nevada, and Arizona in the Western United States have grown more alarming in 2020. To solve the challenge, many governors pay lots of attention to forest management and implement lots of strategies and policies to prevent wildfires, as President Trump said, "forest management is the single solution to combating fires out West" [2].

However, we will focus on another aspect of wildfire control. While natural factors that cause wildfires are difficult to control and forest management is a long and arduous undertaking, we want to predict the possible severity of a wildfire once it takes place, as Thompson et al. [7] has suggested, "...effective response to fire is the key to contain wildfire." Correct decision making regarding how to effectively deploy resources and control the development of wildfires

[1] https://www.accuweather.com/en/weather-news/california-wildfires-will-cost-tens-of-billions-accuweather-estimates/612548

[2] https://www.washingtonpost.com/podcasts/can-he-do-that/is-the-federal-government-to-blame-for-wildfires-gone-out-of-control/

should be made within limited time and with high priority. By predicting the possible fire size class under some specific situations, governments and fire departments can take our model result as a reference combining their domain knowledge and experiences to better manage priorities of fire fighting tasks and allocate wildfire prevention resources, so that casualties and property loss can be avoided as much as possible. In this project, we will formulate this problem as a multi-classification task. With the information collected when a wildfire takes place, our model can help to predict the expected level of the wildfire size and assign probabilities to each level.

II. Data Understanding

The Kaggle dataset [6] we used contain a spatial database of wildfires that occurred in the United States from 1992 to 2009, which is acquired from the reporting systems of federal, state, and local fire organizations. The original dataset is in SQLite form with 38 columns and 1,048,575 rows. Each row represents an incident of wildfire and related information including identifiers of the unit preparing the report and the local fire management organization, location information such as state, county, and coordinate, and fire information such as discovery date and cause. Most importantly, we will use fire size class, which is based on the number of acres within the final fire perimeter expenditures, to measure the severity of wildfires. There are 7 classes ranging from A to G such that A is less than 0.25 acre and G is greater than 5,000 acres.

A. Exploratory Data Analysis

First of all, we want to understand the geographic distribution of wildfires around the United States over the years. By plotting every incident of wildfire with its coordinate as shown in Fig. 1, we found that wildfires spread all over the United States and the fire size is larger in the Western region. To be more specific, California, Texas, Alaska, New Mexico and Nevada are the five states that suffer the most from wildfire (see Appendix Fig. A.1a), and our analysis will focus on these five states.

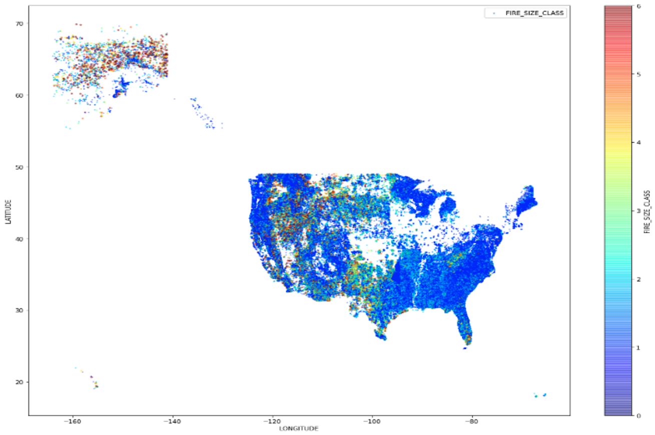


Fig. 1: Geographical distribution of wildfires in the United States from 1992 to 2009. Fire size is represented with colors in spectrum.

By analyzing the number of fires over the 18 years (see Appendix Fig. A.1c), we could not see an increasing or decreasing trend over time, which means wildfires highly depend on specific situations each year. In 2006, there were more than 90,000 wildfires, while in 2009, there were only about 30,000 cases in the U.S.

What's more, when we looked at the distribution of fire size class, we found out that about 85% of fires are classified as A and B, 12% belongs to class C, and only 3% are large fires (Class D, E, F, G), which means our dataset is extremely imbalanced (see Appendix Fig. A.1d). That's why we should not only consider states that have large numbers of fires, but more importantly, consider states that have more serious wildfires, such as the five states we selected to include more large fire instances in our dataset so that the label will not be that biased. Finally, the cause of wildfires is also an important factor (see Appendix Fig. A.1e). Among all 13 causes, lightning, miscellaneous, and arson are the top three and lightning even caused more than 100,000 cases of wildfires, indicating that natural factors such as weather and location are significant predictors for estimating the severity of wildfires.

III. Data Preparation

In this section, we explain methodologies used to prepare the dataset, as well as our feature selection and engineering procedures.

A. Data Cleaning and Integration

In the original dataset, there are too many columns related to identifiers for wildfires and columns containing more than 600,000 null values, which are useless for our prediction. Therefore, to produce the format for data mining, we only kept variables such as discovery date, coordinate, state, cause code, and fire size class, and etc. Also, as claimed previously, we only focused on California, Alaska, New Mexico, Nevada, and Texas. Then, to make the discovery date of wildfire clearer, we integrated fire year and discovery date of year into date format and extracted month from the date to create a new column because wildfire is likely to vary with months, which may be important to our model.

B. Target Variable

Our target variable is the fire size class. As it is reported by the responsible fire management organization and measures the expansion of wildfires after fires were under control, it is able to represent the severity of wildfires in our model. Similarly, we also used the label encoder to transform class A to G to numbers 0 to 6 in the model.

C. External Data – Elevation

As claimed by Hayes, "When the mountain tops and slopes are still under snow and only the valley bottoms are dry enough to burn, then the fire control force needed and the cost of that force can and obviously should be held to a minimum." [5] Therefore, it is reasonable to assume that the expansion of a wildfire varies with the altitude because of the effects of terrains and temperatures. Thus, we introduced external data, elevation, into our datasets by inputting the longitude and latitude information to the USGS API [3].

To better understand elevation, we first visualized its distribution (see Appendix Fig. A.1f). Elevations in the five states range from -83.12 to 3,985.29 meters and the average is 1,504.32 meters. Then, more specifically, we compared distributions of elevations in different states (see Appendix Fig. A.3). For most terrains, Alaska and Texas are relatively low, ranging from 200 to 400 meters, but Texas has some Plateau regions. Most regions in California are about 500 to 2,000 meters high. New Mexico and Nevada have approximately normal distributions centered around 2,000 meters. Furthermore, we also looked at the distribution of elevations for each fire size class (see Appendix Fig. A.2). It shows that when fire size class is low, the distribution reaches its peak in high elevations around 2,000 meters, while many larger fires occur in lower elevations below 1,000 meters. Therefore, elevation is potentially an important factor for predicting the fire size class.

To sum up, before any feature engineering, features in our dataset are fire year, discovery date of year, control date of year, discovery month, latitude, longitude, state, elevation, land's owner code (the agency or private owner who administers the land), and fire size class.

D. Feature Selection

In this part, we plotted the correlation matrix to understand the multicollinearity between our features and further selected the most informative ones using the forward stepwise selection

[3] https://www.usgs.gov/core-science-systems/national-geospatial-program/national-map

approach based on the mutual information between each feature and the target variable.

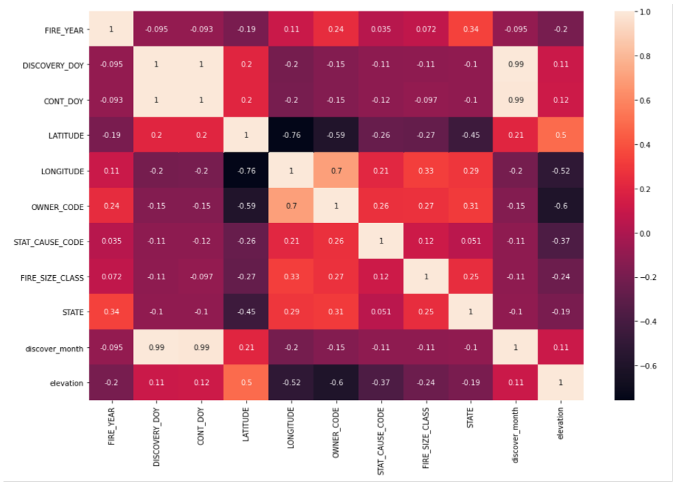


Fig. 2: Correlation matrix with selected features.

As shown in Fig. 2, there are several noticeable multicollinearities between our variables. First, the correlation between discovery date of year and control date of year is 1, because in most of the cases, the fire was controlled within one day, so the date of discovery and date of fire under control is the same. Also, as we want to create a predictive model, including the control date will lead to the data leakage problem because it's an indicator of the fire size, and data leakage is definitely something we try to avoid, so we dropped this feature from our feature set. Besides, the correlation between discovery month and discovery date of year is 0.99 because the month column was generated from date of year, so we only included the discovery month feature because of its better generalization ability. With regards to the target fire size class, longitude, owner code, state, and cause code are the four features with high positive correlations with each other, which makes sense since all of them encode the geographic location information, and it can be significant when predicting the fire size class. Latitude and elevation have the lowest negative correlations with fire size. One reasonable hypothesis is that as the latitude and elevation decrease, temperature goes up and wildfire is more likely to happen and tends to be more severe under higher temperature. So, until now, we have our initial feature candidates including the latitude, longitude and the elevation of the fire's site, the land's owner code, the root cause of the fire, the state and the discovery month of the fire.

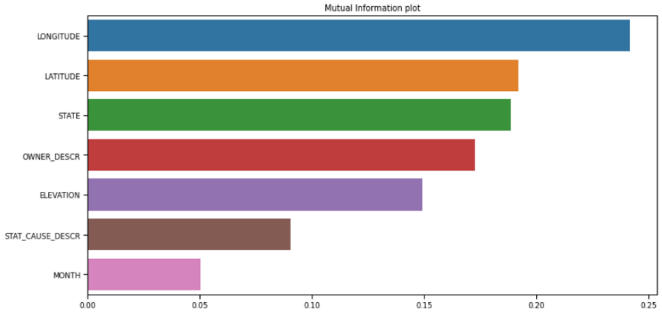


Fig. 3: Normalized mutual information between feature and target.

Next, we would like to further select the most informative features from our preliminary feature candidates and potentially drop those that didn't add any new information to our model using the forward stepwise selection approach based on the mutual information between feature and the target variable. Fig. 3 shows the mutual information between each feature and the fire size, and we can see that longitude, latitude and state are the three features that have the highest mutual information with the fire size. We used two baseline models, Random Forest and Logistic Regression, to accomplish the forward stepwise selection process. The reason we chose these two models is that we wanted to compare their performance to understand the level of non-linearity in our data so that later we can better prepare our model candidates in the modelling part. Forward stepwise selection [2] is a feature selection process that begins with a model with no feature, and starts adding the most significant features one after the other until all the features are included in the model. In our case, we used mutual information as the proxy for significance, so we started from the feature with the highest mutual information, which is longitude, and kept adding features and tracking the accuracy of both models on the validation set. Fig. 4 shows the learning curve of both models.

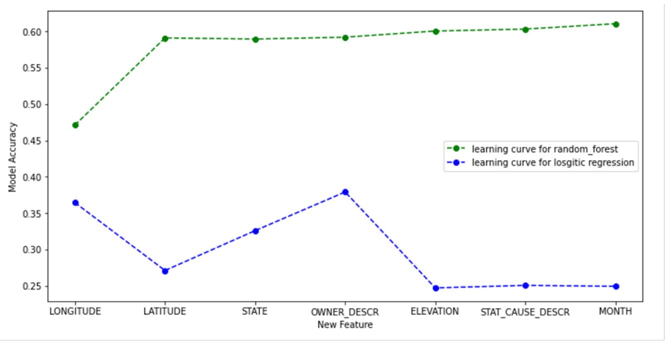


Fig. 4: Learning performance curves for Random Forest and Logistic Regression classifiers.

From these two learning curves, it's easy to see that logistic regression performs poorly compared to the random forest, so there must be some non-linearity in our data. By observing the random forest's learning curve, we can see there's a huge boost to our model's performance by adding the first two features, Longitude and Latitude. However, although State has the third largest mutual information with the fire size, it does not improve our model's performance because its information is mostly encoded by the combination of longitude and latitude, so we decided to remove it from our feature set. The last four features provide marginal boosts to our model's accuracy, so we can keep them. The final feature set we decided to feed into our model is Latitude, Longitude, Owner, Elevation, Cause and Month.

IV. Modelling

From the feature selection part, we noticed the non-linearity in our data, so Logistic Regression is not a feasible choice in our case. Considering the scale of our data, we have over 190,000 instances in our dataset, which could be an obstacle for running algorithms like SVM, whose computational cost is assumed to be $O(n^3)$ [3]. What's more, since we have continuous features, Naive Bayes, which is designed for discretized variables, would not fit for our task.

After excluding a few algorithms, we narrowed down our choices. First of all, to handle the non-linearity in our data, tree-based models are the go-to choices, so Random Forest and XGBoost should be the top candidates. Both of them can work well with our large size dataset, handle the imbalance in our data, and are less prone to overfitting. K Nearest Neighbors can also be helpful in our case because it has no assumption about the data, and our decent number of features can help it avoid the curse of dimensionality. But it doesn't work pretty well with imbalanced data, which could be something to keep in mind. A Multi-Layer Perceptron (MLP) with the activation function in the hidden layer can also deal with the non-linearity in our data, and the large scale of data is more than enough to train the classifier to a great extent, but one drawback is its low interpretability. We also included an ensemble model by stacking these four models together and added another XGBoost as our second stage meta-model to further improve our model's performance. This stacking model will generally have better overall performance compared to our individual models, but it will have the highest computational cost and the least interpretability.

A. *Baseline Model and Evaluation Metrics*

Even though our dataset is extracted from Kaggle communities, there is not any predetermined performance evaluation metric for us to compare with the leaderboard. To underline, the objective is also not specified in the Kaggle community, and the task is framed by ourselves after understanding the data and evaluating the business interests. Therefore, no relevant projects are available for us to compare with.

Since we are dealing with multi-class labels, we decided to use accuracy as the evaluation metric, which is the most

TABLE I: Model used in study with hyperparameter candidates. Optimal hyperparameter candidates are shown in bold.

Model	Hyperparameter	Candidates
Random Forest	n_estimators	100, 150, 200, 250, **300**, 350, 400
XGBoost	max_depth	5, 7, **9**, 11, 13
K-Nearest Neighbor	n_neighbors	10, 30, 50, 70, 90, 110, **130**, 150, 170, 190, 210
Multi-layer Perceptron	hidden_dim	5, **10**, 15, 20, 25

straightforward and informative evaluation metric for the multi-class classification task. From the previous forward stepwise selection section, we saw Random Forest with the default settings ($n\_estimator = 100$) did a decent job in making the classification with an accuracy of 60.91%, thus used as our baseline model.

B. Model Selection

The model candidates we selected are: a) Random Forest, b) XGBoost, c) KNN, d) Multilayer Perceptron with one hidden layer, and e) Stacking ensemble model with XGBoost as the meta-model. First, for each of the four individual models, we tuned one of the most important hyperparameters and chose the optimal one according to their performance on the k-fold cross validation sets with $k = 5$. The hyperparameter candidates we choose to tune for each model are shown in Table. I.

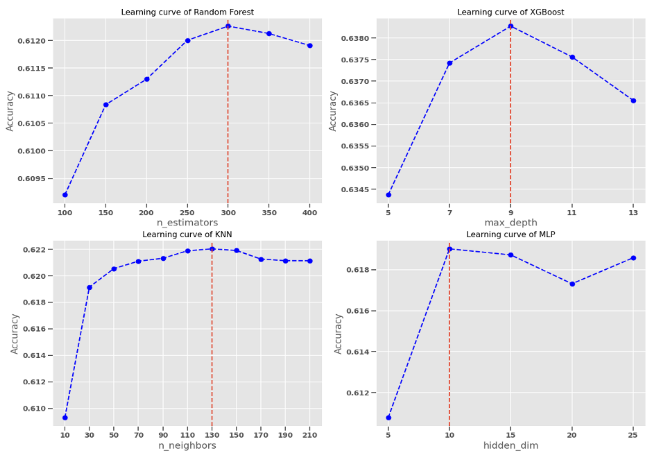


Fig. 5: Learning curve for each classifier. The red dotted lines indicate optimal hyperparameter values associated with highest average accuracy.

Fig. 5 shows the learning curve for each model and the red line indicates the optimal hyperparameter associated with the highest average accuracy on the 5-fold cross validation sets. The optimal hyperparameters for each individual model are highlighted in Table. I.

Next, we created our stacking ensemble by combining these four optimized individual models and using XGBoost as the second stage meta-model to see whether it can further boost the accuracy. Still, we used average accuracy on the 5-fold cross validation sets to evaluate the performance of our ensemble. Table. II shows the performance of all models on the k-fold cross validation sets. It's obvious to see that Ensemble has the best overall performance on the validation set. It improves the baseline performance by 3.6% in terms of accuracy and outperforms all the other individual models.

TABLE II: Average accuracy of models on 5-fold cross validation sets.

Model	Average Accuracy
Baseline	60.91%
Random Forest	61.25%
XGBoost	62.82%
KNN	62.2%
MLP	61.9%
Ensemble	**64.51%**

C. Evaluation

In our training phase, we identified that the stacking ensemble is the optimal model that has the best out-of-sample accuracy, we will now check how it performs on the testing set in terms of predicting the 7 fire size classes using a confusion matrix.

From the confusion matrix in Figure 6, we can see the model did a great job in predicting fire size A and B, but it works

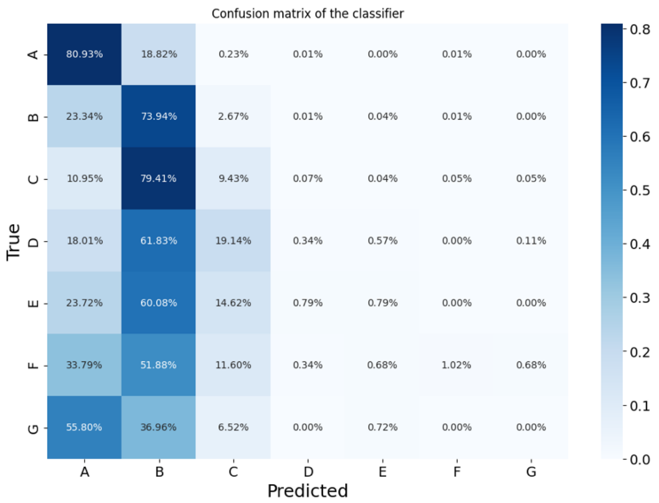


Fig. 6: Confusion Matrix of the stacking ensemble model.

poorly for those fires size over C. The recall is over 80% for Class A and over 73% for Class B, which are the two majority classes, but it goes down drastically to less than 10% for Class C and almost 0 for the last four classes. This phenomenon is hugely attributed to the extreme imbalanced dataset we have. Recall is actually one important metric in our case because the cost of false negative (type 2 error) for predicting large fires is extremely high. Giving a large fire a prediction of class A would result in some serious consequence because the lack of resources deployment can barely impede the development of fire, which will lead to huge casualties and property damage. The way we deal with this is that instead of directly predicting the class label with the highest probability, we assigned a unique threshold for each class, and if the model's predicted probability for that class is greater than the given threshold, the output will be that class. Fig. 7 shows one example of how we choose the threshold for class G. The underlying logic is that we first fitted our model to the training set, and then we split our training set into two groups, one group was instances with label G, and the other one was instances with label A-F. By observing the logarithm of model's output probability for Class G of these two groups and plotting their distributions, we created the cut-off threshold that can best help us separate these two distributions, which is indicated by the red line in the graph. If the model's predicted probability on one class is greater than the threshold, the instance will be classified as the class, otherwise it will be further compared to the thresholds of other classes and be classified accordingly. The algorithm can be described as such:

```
for each $prediction$ in the $output$ do
    if $prediction[G] \geq threshold[G]$ then
        $class \leftarrow G$
    else if $prediction[F] \geq threshold[F]$ then
        $class \leftarrow F$
        ... ...
    else
        $class \leftarrow A$
    end if
end for
```

Using this approach, we can see our recall for those large fires significantly improved without sacrificing too much accuracy, as shown in Table. III and as the confusion matrix reflects in Fig. 8.

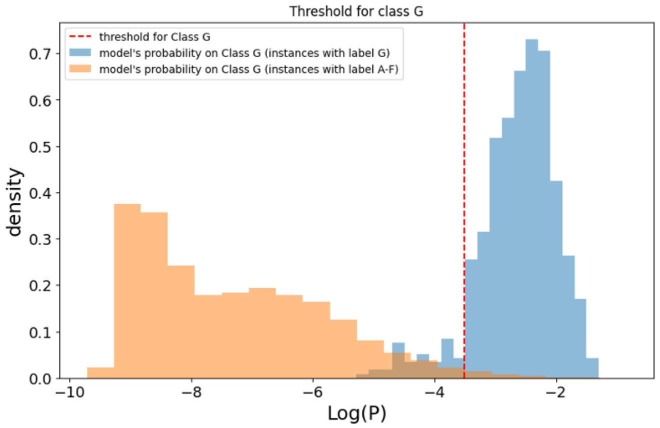


Fig. 7: Distribution of model probability on Class G with correct and false predicted label, in logarithm scale. The distribution is used to calculate optimal threshold for the class.

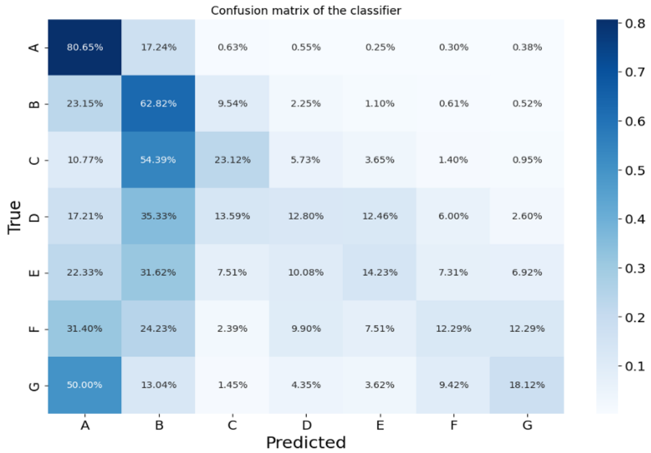


Fig. 8: Confusion Matrix of the stacking ensemble model, with modified threshold. In comparison with the confusion matrix without threshold tuning, minority classes (Class C to G) have significant improvement in true predictions.

TABLE III: Recall for each class, with and without threshold tuning. Recall performances on Class C to G improved significantly, while that on Class B decreases. Overall accuracy went down slightly.

	Recall w/o threshold	**Recall w/ threshold**
Class A	0.81	**0.81**
Class B	**0.74**	0.63
Class C	0.09	**0.23**
Class D	0.00	**0.13**
Class E	0.01	**0.14**
Class F	0.01	**0.12**
Class G	0.00	**0.18**
Overall Acc.	**0.63**	0.61

Although there's a significant boost after adjusting the prediction mechanism by involving the threshold, the overall recall for large fires is still pretty low. Therefore, it's not reliable to entirely depend on our model to predict the size of wildfire because there are far more other random factors that determine how the wildfire spreads, like the weather conditions and human interference. However, our model can serve as a valuable forecasting tool to reference when users are making their decisions. Experts can apply their practical expertise and domain knowledge, combined with the model's prediction to make the optimal judgement and assign appropriate resources and workforce to impede the spread of wildfire.

V. Deployment and Future work

The model works as a forecast tool for stakeholders like the US Forest Service, and Fire Department to optimize the resources and response level given different levels of wildfire. The prediction will be made when a wildfire is reported by inputting the feature information about the wildfire. A list of probabilities for each level of wildfire will be available for the stakeholders. To illustrate, if a higher probability of large fire is returned, more resources should be prepared and distributed, and emergency responses should be more immediate. Furthermore, to evaluate the model in real practice, it should be highlighted that wildfire is often developed from small fire to large fire, so a caution of large wildfire should not be simply considered wrong when it turns out to be medium or small because it could result from the immediate response from the fire departments. Alternatively, an A/B test can be used to evaluate the model in actual use to see if places with deployment of the model have fewer cases of large wildfire. However, this is very unethical to conduct such an experiment in reality because of the unfair risks to people and property at places without the model, which contrasts with the objective of the model. Therefore, a proper method to evaluate the model in practice is to test it with new data and update the models periodically.

The model gives a good prediction for every wildfire, but it is not absolutely reliable because there are other random factors that are not under consideration that can affect the level of the wildfire, for example, humidity, wind level, and precipitation. Users should combine the knowledge about the situation and the predictions of the model to make the best decision. Also, our model is trained for states with high potential of large wildfire, so its deployment should be restricted to certain states preventing from overreactions.

Given the limitations and potentials of the model, for future works, more features like humidity, precipitation, wind level, and forest areas [1] can be included in the model to improve the performance of the model. At the same time, more states can be introduced into the training data. States like South Carolina, Florida, and Georgia also suffer a lot from the wildfire but most of them are small wildfires, but our model can still work as a caution for the fire department to prepare for the worst and optimize the resources.

VI. Team Contributions

Contributions of team members on the project are describes as following:

- **Long Chen:** exploratory data analysis, data preprocessing, feature engineering, and evaluation
- **Yuhan Gao:** feature selection, model training, model deployment and future works
- **Yuhan Liu:** data understanding, exploratory data analysis, data preparation and feature selection
- **Zhifan Nan:** feature engineering, model selection, hyperparameter tuning and evaluation

References

[1] Stuart AJ Anderson, Jonathan J Doherty, and H Grant Pearce. “Wildfires in New Zealand from 1991 to 2007”. In: *New Zealand Journal of Forestry* 53.3 (2008), pp. 19–22.

[2] George Choueiry. *Understand Forward and Backward Stepwise Regression*. URL: https://quantifyinghealth.com/stepwise-selection/.

[3] *Computational complexity of machine learning algorithms*. Apr. 2018. URL: https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/.

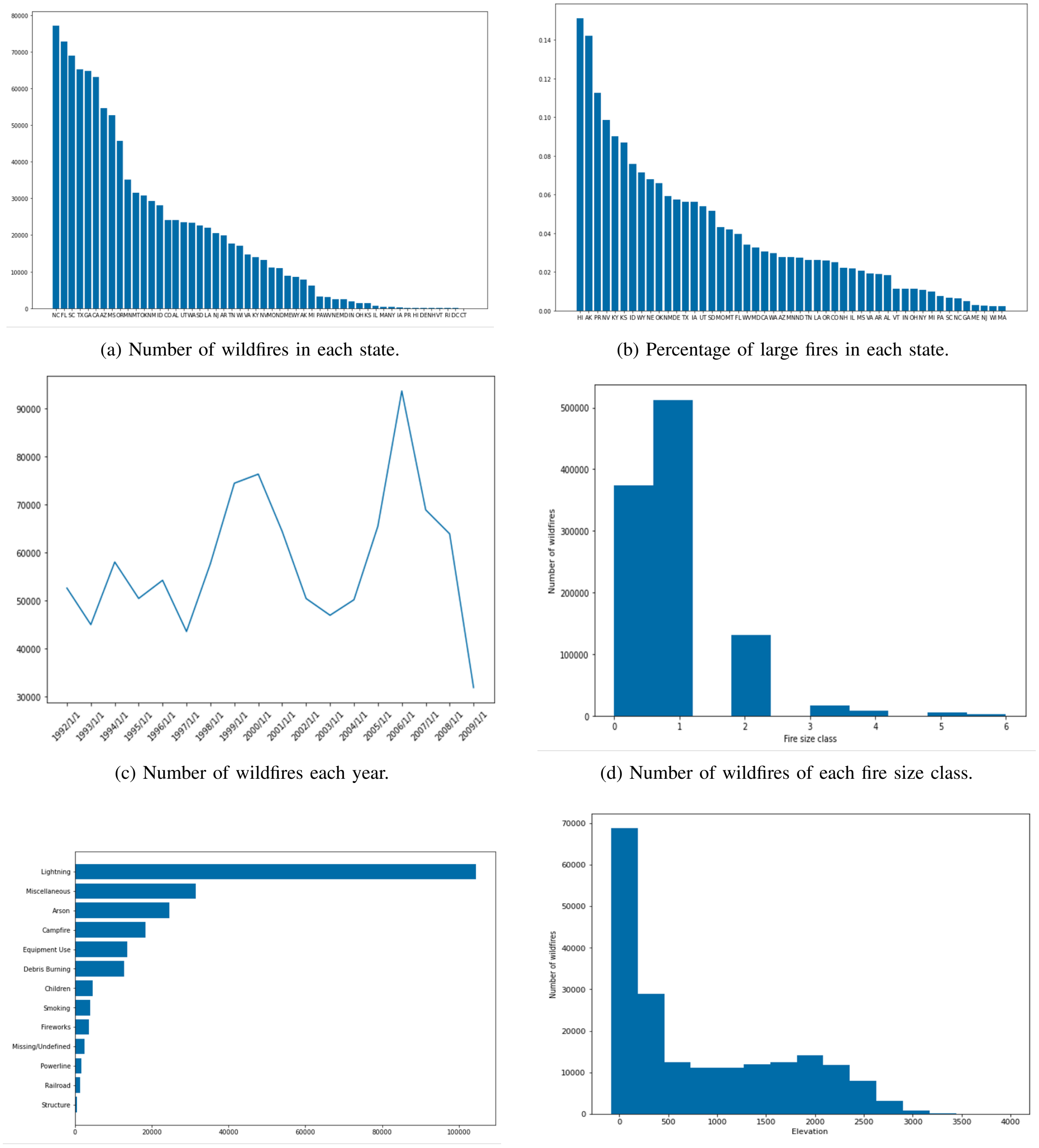
[4] Shailaja Gupta. *Pros and cons of various Classification ML algorithms*. June 2020. URL: https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6.

[5] G Lloyd Hayes. “Differences in fire danger with altitude, aspect, and time of day”. In: *Journal of Forestry* 40.4 (1942), pp. 318–323.

[6] Kaggle. *1.88 Million US Wildfires*. 2020. URL: https://www.kaggle.com/rtatman/188-million-us-wildfires.

[7] Matthew P Thompson et al. “Application of wildfire risk assessment results to wildfire response planning in the southern Sierra Nevada, California, USA”. In: *Forests* 7.3 (2016), p. 64.

Appendix



(a) Number of wildfires in each state.

(b) Percentage of large fires in each state.

(c) Number of wildfires each year.

(d) Number of wildfires of each fire size class.

(e) Number of wildfires for each fire cause.

(f) Distribution of elevation.

Fig. A.1: Supplemental figures for Exploratory Data Analysis.

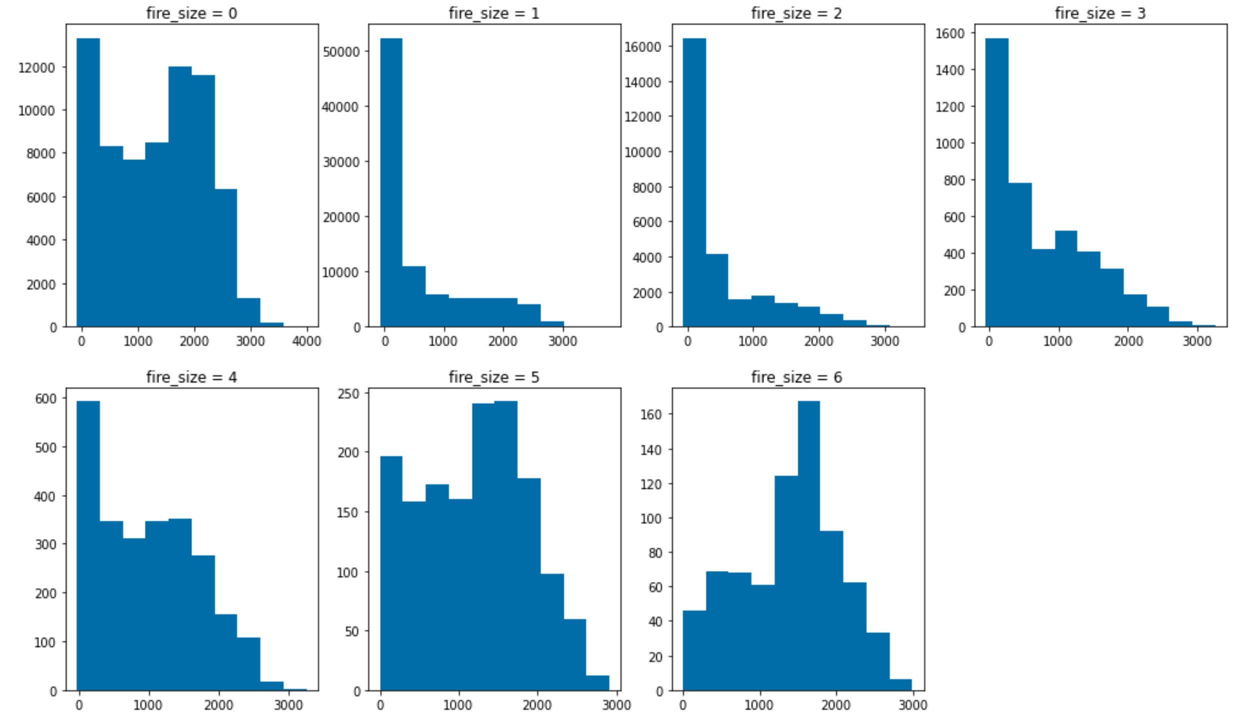


Fig. A.2: Distribution of elevation for each fire size class. Horizontal axes represent elevation of the location of the wildfire incidents. Vertical axes represent the number of wildfire incidents.

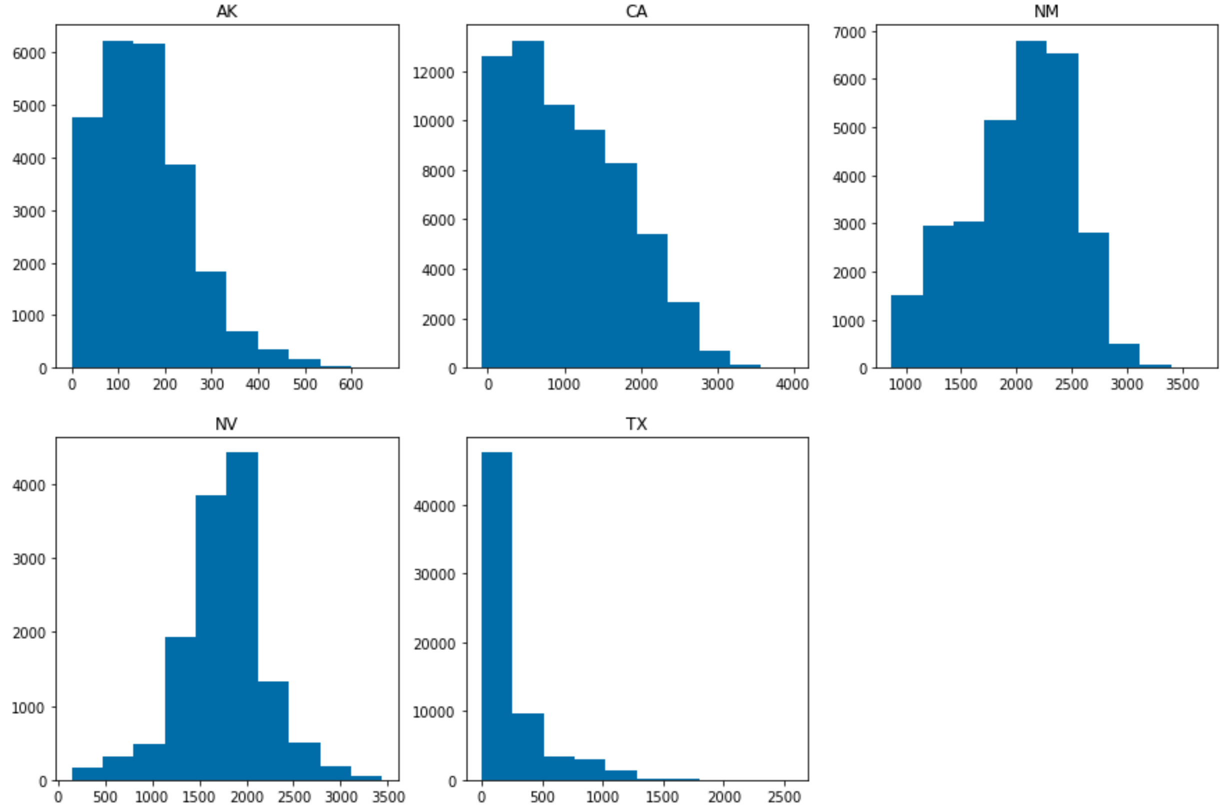


Fig. A.3: Distribution of elevation for each state. Horizontal axes represent elevation of the location of the wildfire incidents. Vertical axes represent the number of wildfire incidents.