Final report

Authors:
Sarah Dahan sarah.dahan1@mail.huji.ac.il
Hsin-Chun Yin Hsin-Chu.Yin@mail.huji.ac.il
Shkitan Levy shkitan.levy@mail.huji.ac.il
Almog Mor almog.mor@mail.huji.ac.il

Introduction:

Our research process workflow contains 2 stages:

1. Preparation
2. Experimentation

What actually happens is that we iteratively try to add more features and try new models, and see how that affects the performance. Here we will explain what we did.

Instruction:

Running the notebook on a local computer takes about 15 mins. Feature encoding takes about 7 mins, and running each model takes about 3 mins. In order to run the notebook, please place data.csv and our provided data “storm_events.csv” under the same directory of the notebook. And create an empty folder called “tempFiles” under the same directory.

Preparation:

We divided the column spaces by the group members and each member of the teams explored a subset of the columns.

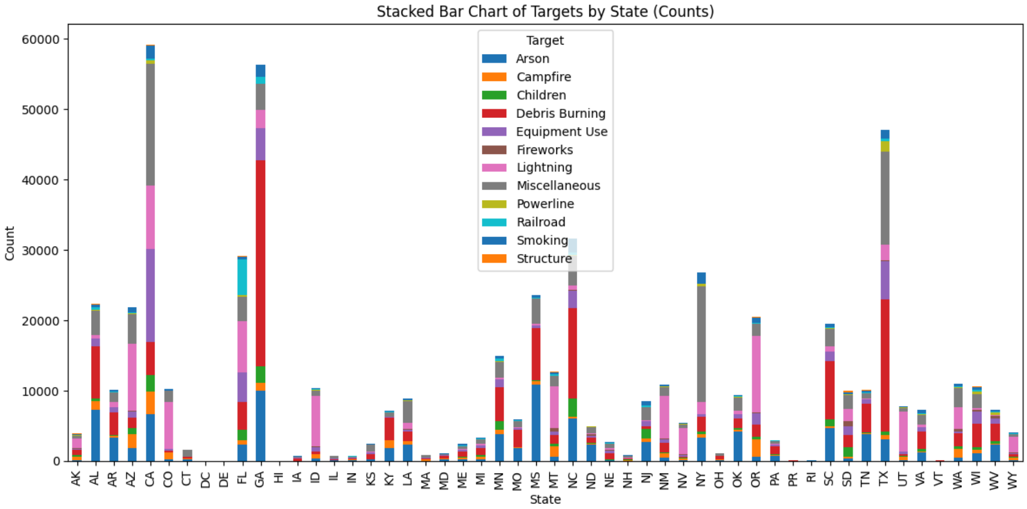
The given data is about wildfires. We were given a tabular dataset where each row represents a case of a wildfire and each column represents a feature that is related to the specific row’s case. One of the columns is the wildfire cause and our goal is to classify given the other columns for each specific row, what is the cause of the wildfire.

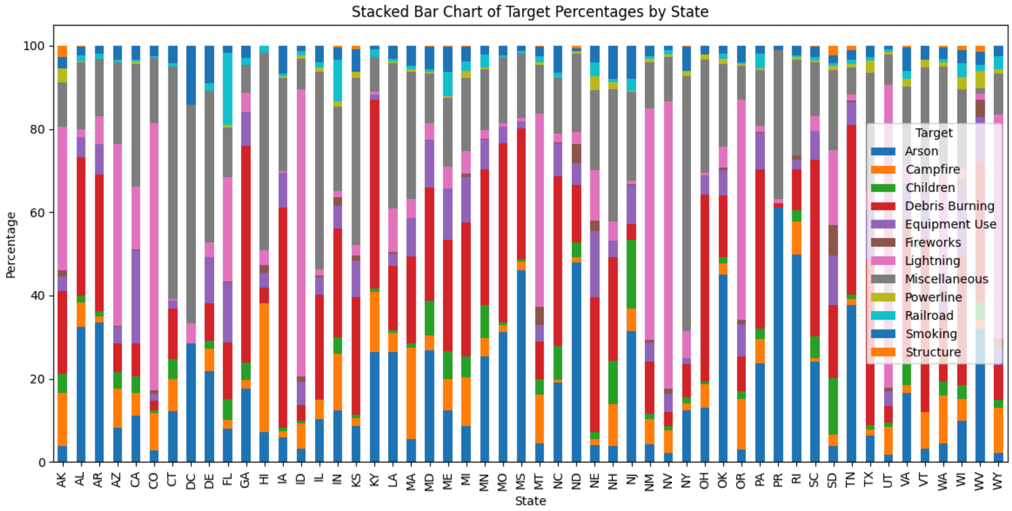
After understanding and scoping the problem, we reached out to understand the columns. The exploration of the data together with our intuition about the problem lead us into making data driven decisions at the exploration and model choosing part.

Geospatial Data

EDA

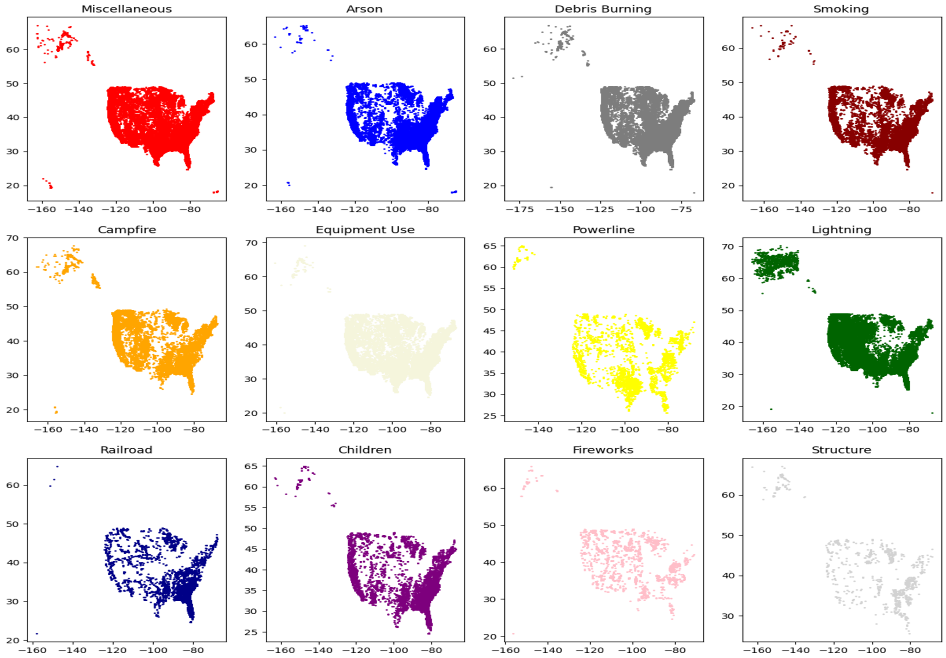
First we checked the distribution of the labels per state to see if we can detect some anomaly cases


Stacked Bar Chart of Targets by State (Counts)
Count
0
10000
20000
30000
40000
50000
60000
State
AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA PR RI SC SD TN TX UT VA VT WA WI WV WY
Target
Arson
Campfire
Children
Debris Burning
Equipment Use
Fireworks
Lightning
Miscellaneous
Powerline
Railroad
Smoking
Structure


Stacked Bar Chart of Target Percentages by State
Percentage
0
20
40
60
80
100
State
AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA PR RI SC SD TN TX UT VA VT WA WI WV WY
Target
Arson
Campfire
Children
Debris Burning
Equipment Use
Fireworks
Lightning
Miscellaneous
Powerline
Railroad
Smoking
Structure

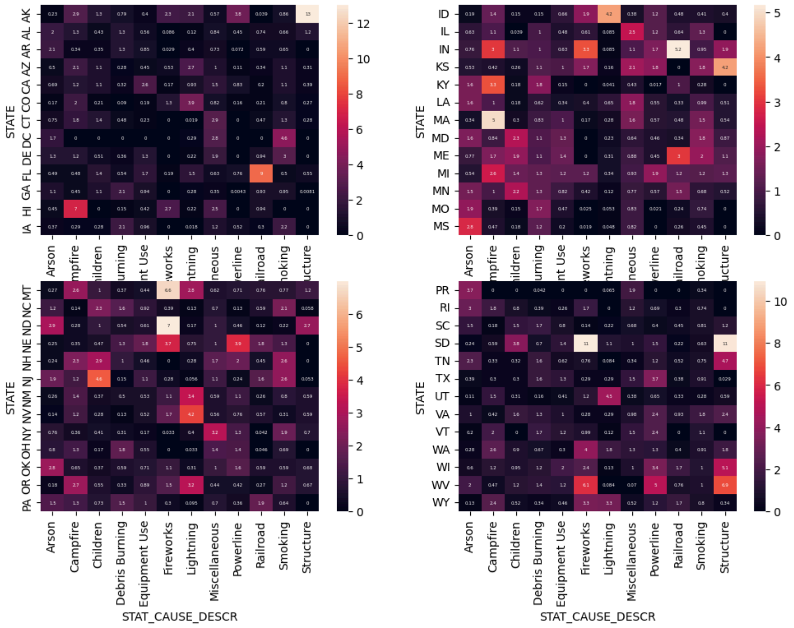
We can notice that some states have a high amount of incidents where most states have less than or equal to 10k. A big amount of the Debris Burnings happened in Georgia and Texas, states that are characterized with dry and hot climate.

We used geolocation to see it on the map.



Indeed we can see that the Debris Burning is dense at the South and East side of the map where Texas and Georgia are. We can also look at lightning and see a big dense circle at the top left where Alaska is.

We fix a state, and see how the distribution will change. We use the new distribution to divide by the original distribution to understand how much the fixing of the state influences the prediction of the target classes.



We can see that for each cause of fire, there are certain states that are higher in percentage in this cause of fire. So this will be a good indication.

Feature Engineering

Longitude and latitude

We used DBSCAN to cluster the longitude and longitude points and then calculated the distance of each point to its nearest cluster (the centroid of the cluster). We applied these changes on the target column as well.

State

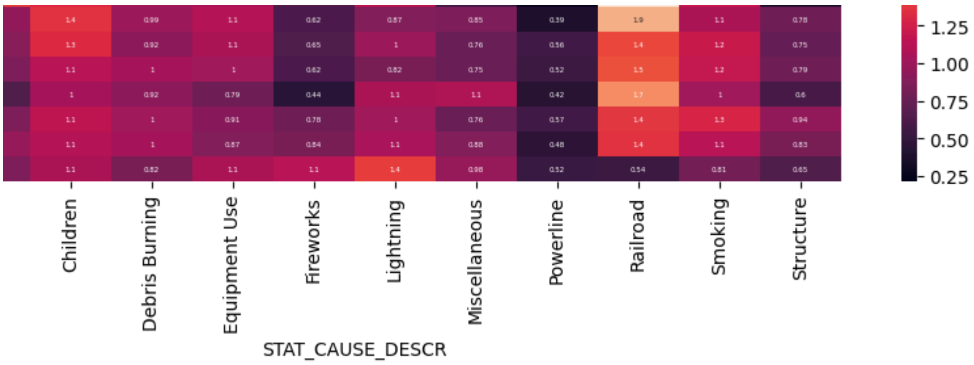
We encode the state by how much they contribute to predicting the target class. Basically we take the number in the heatmap above, and for each target class, we encode the sample by how much the state this sample is in affects the prediction of the label.

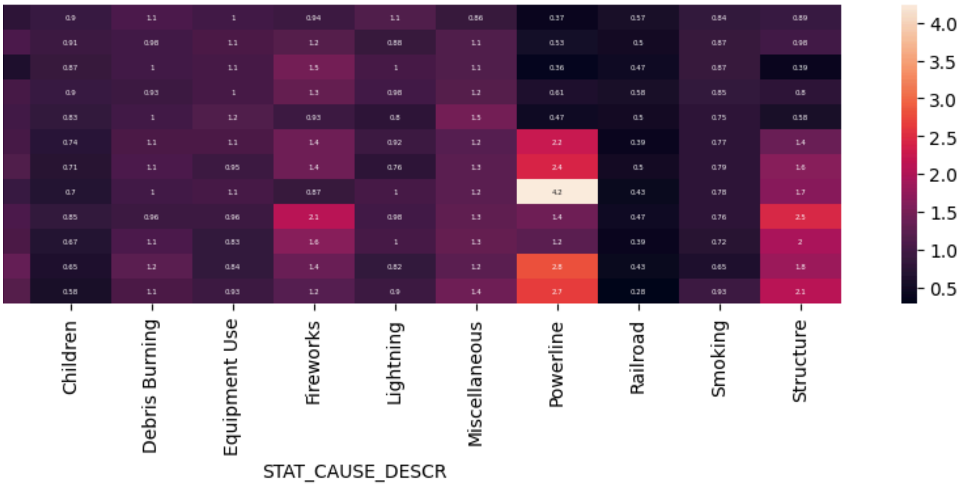
County

The county-fip_code-fip_name are being null together. County column being null has a negative influence on powerline and rail road. But encoding FIPS_code into frequency encoding turns out to be helpful in this case.
We impute the county by the state it is in, and it doesn't really help or harm the result, so we keep it.

Time Data

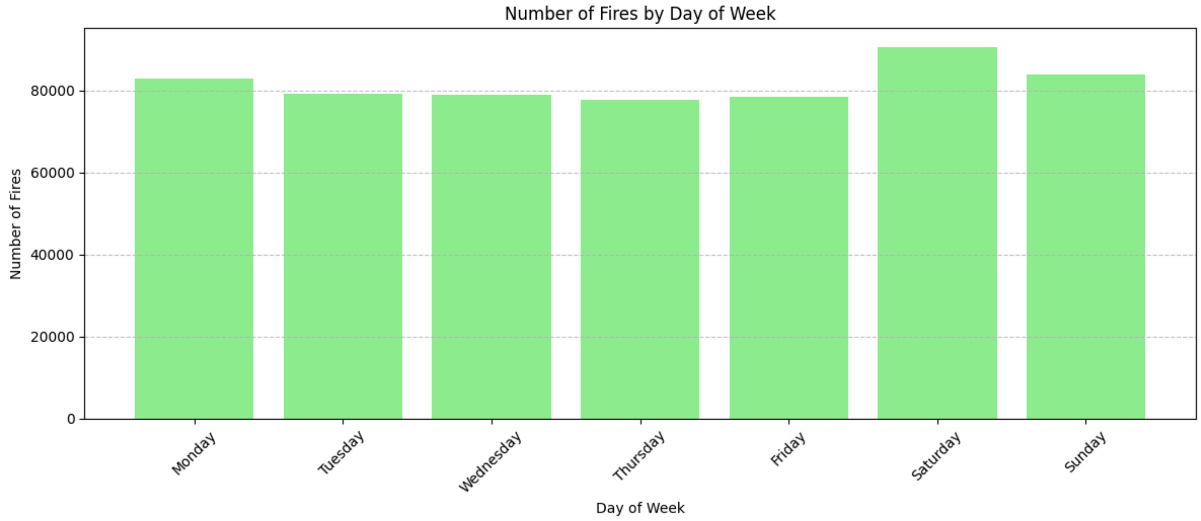
Let's see how a fire year affects the distribution of the target labels. We first calculate the distribution of the target classes when fixed a year then we use the distribution divided by the original distribution of the target classes.

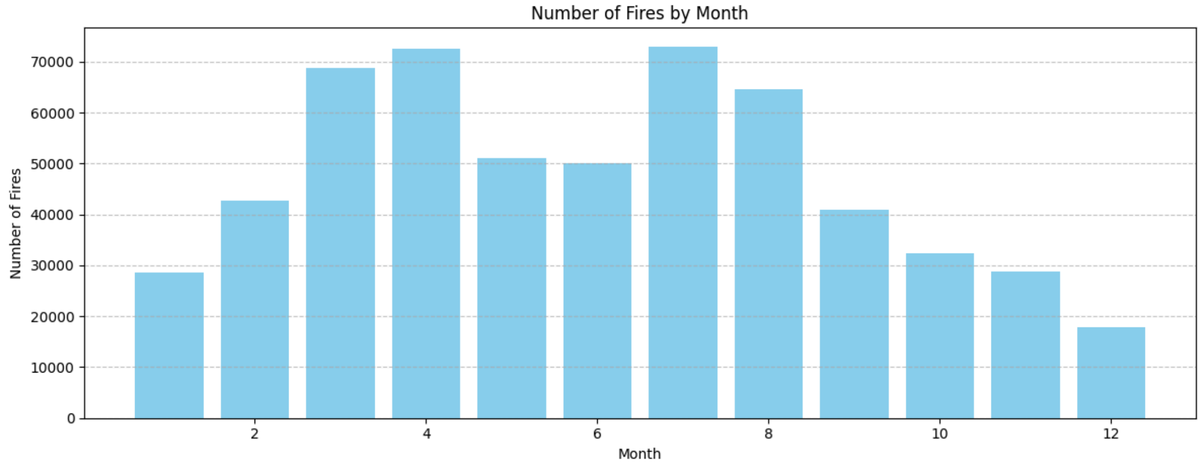




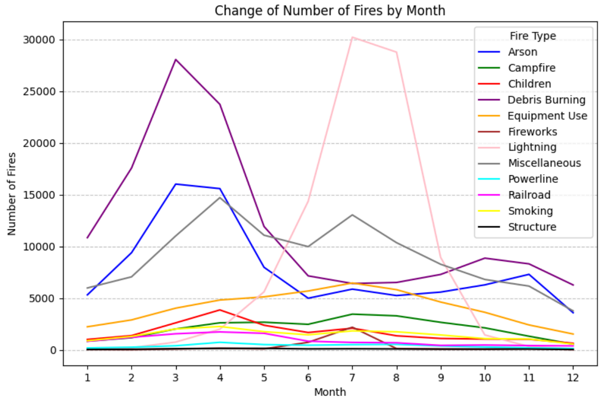
We can see that some target classes such as Arson and Railroad have higher percentage in the distribution in earlier years, and some other target classes, such as Powerline, have higher percentage in the distribution in recent years. This seems to be reasonable.

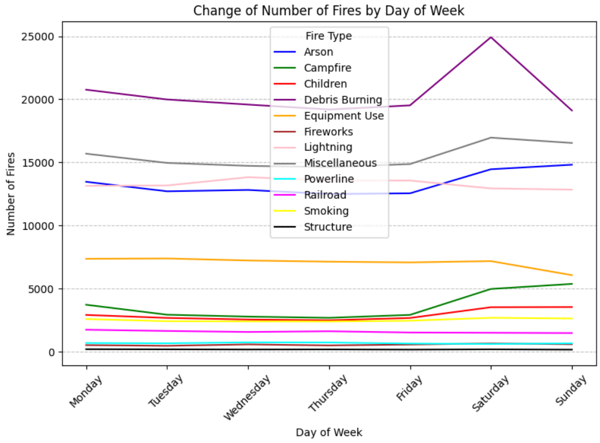
Now we explore how the day of week and month would affect the number of fires.



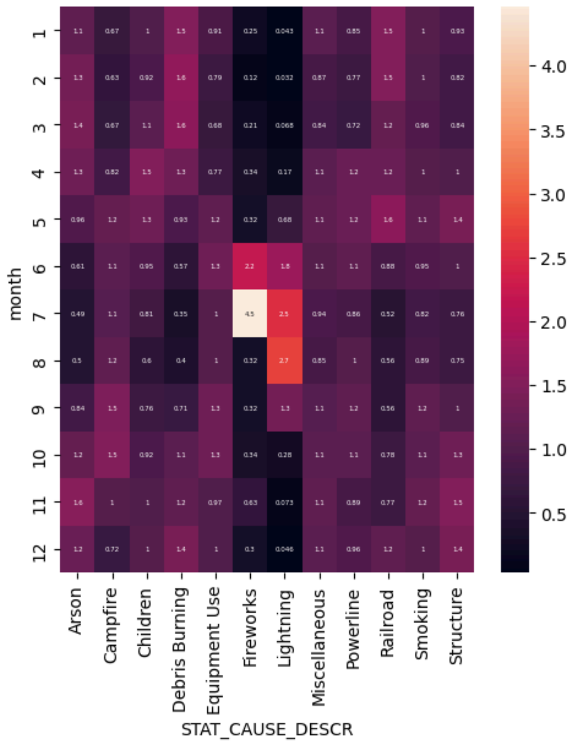


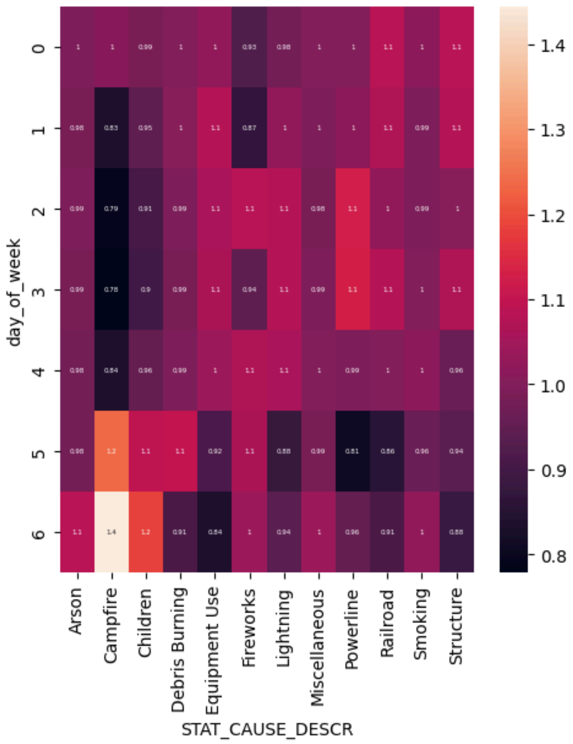
We can see that on Saturdays and Sundays, the number of fires are slightly higher. And in March and April, and also in July and August. So let’s see why:
We plot out for each target class, how the number fluctuate in the year and in the week.



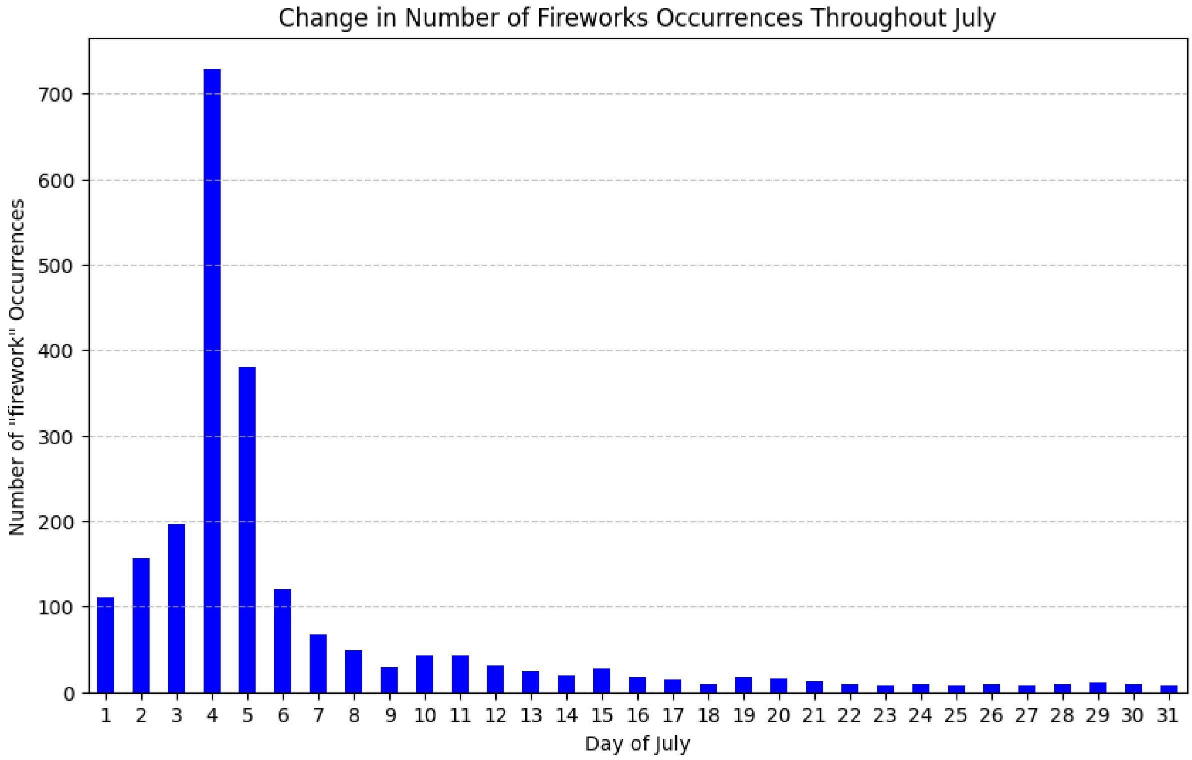


Then just like above, we will see how fixing a month and fixing a day of time would affect the distribution of the target classes.

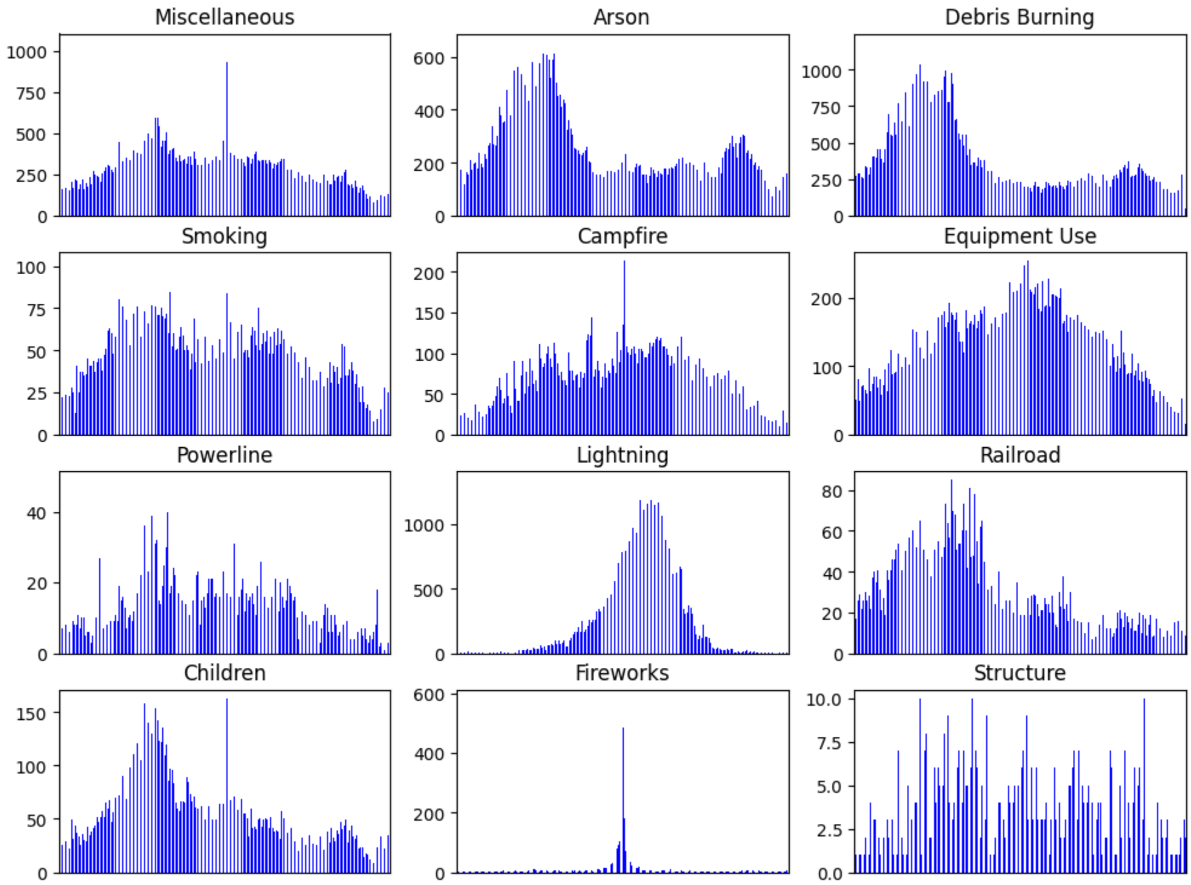




We can see that Fireworks are abnormally high in July, and Campfires are very high on Saturday and Sunday. Both make sense. So in order to further verify our hypothesis, let's look at firework's number during July.

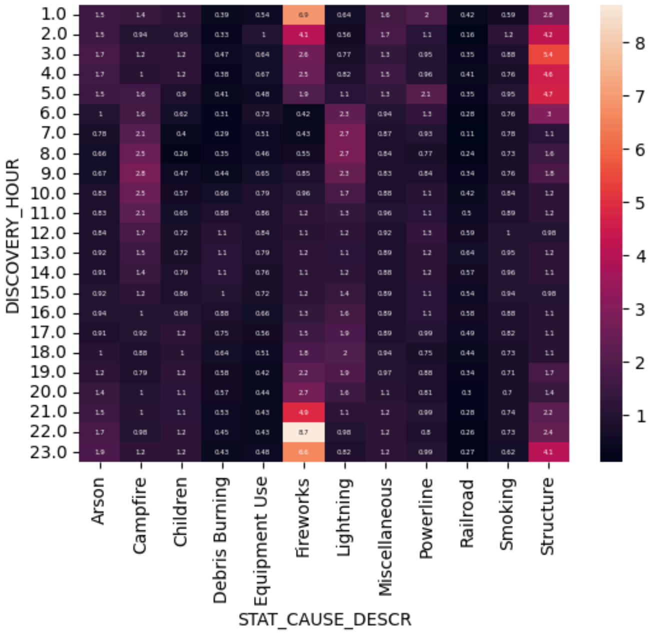


We can see that the firework number increment is indeed because of July the 4th.
Also we want to look at how does each target class fluctuates during the year every day, so we plotted:



We can see that some target classes are more common in the summers and some are more common in the Spring, this explains why we see that the number of fires are higher in spring and summer. From here we also see that miscellaneous fireworks have overlaps Fireworks on July the fourth, and it also overlaps with many other classes. In order to make better predictions, maybe we should separate this class.

Lastly, let’s look at how the hour of discovery affects each label.



We can see that some columns like Campfire are more likely to happen during day time, and some other columns such as fireworks are more likely to happen during night time, and they all make sense.

FEATURE ENGINEERING

From EDA we see that there are many things that will meaningfully affect the distribution of the prediction. So we will encode them. The first things we encode are 'independence_day', 'MONTH', 'DAY_OF_WEEK', 'FIRE_YEAR', 'DISCOVERY_HOUR'. And we found out that encoding these columns by frequency encoding has better results than just passing them in as is.

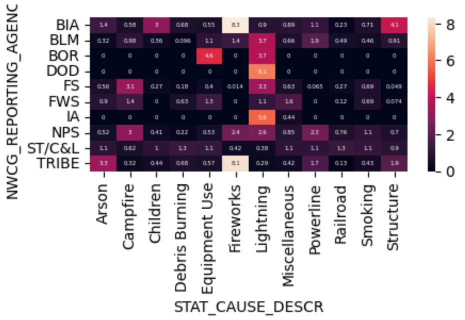
Then we calculate the time to contain the fire. Since CONT_TIME has null values, we fill in all the null values of DEALT_TIME by the mean grouped by each FIRE_SIZE_CLASS.

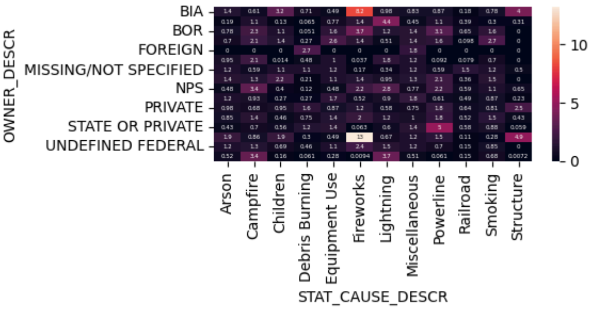
Then we performed Cyclic transformation on the day of year, month, day of week and hour of discovery. But it doesn't really help or harm the performance.

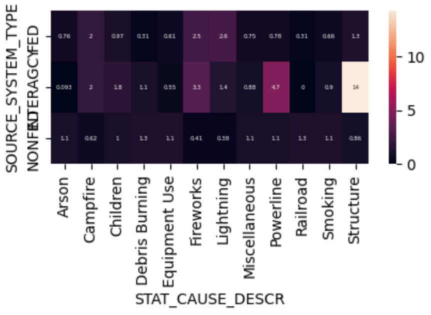
Categorical

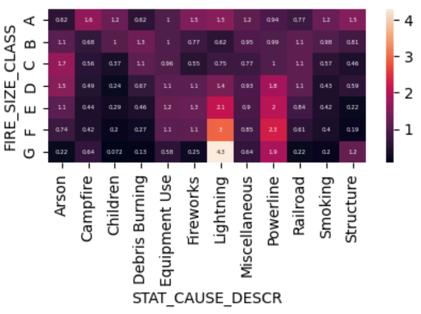
EDA

First we analyze how each categorical class affects the target classes, in order to determine if they are worth encoding. For example, the analisis we do for NWCG_REPORTING_AGENCY , OWNER_DESCR, SOURCE_SYSTEM_TYPE, FIRE_SIZE_CLASS









We divided the categorical feature into two groups: HC_features, which encompasses numerous possible value options, and LC_features, which comprises fewer value options.

Feature Engineering - Frequency Encoding

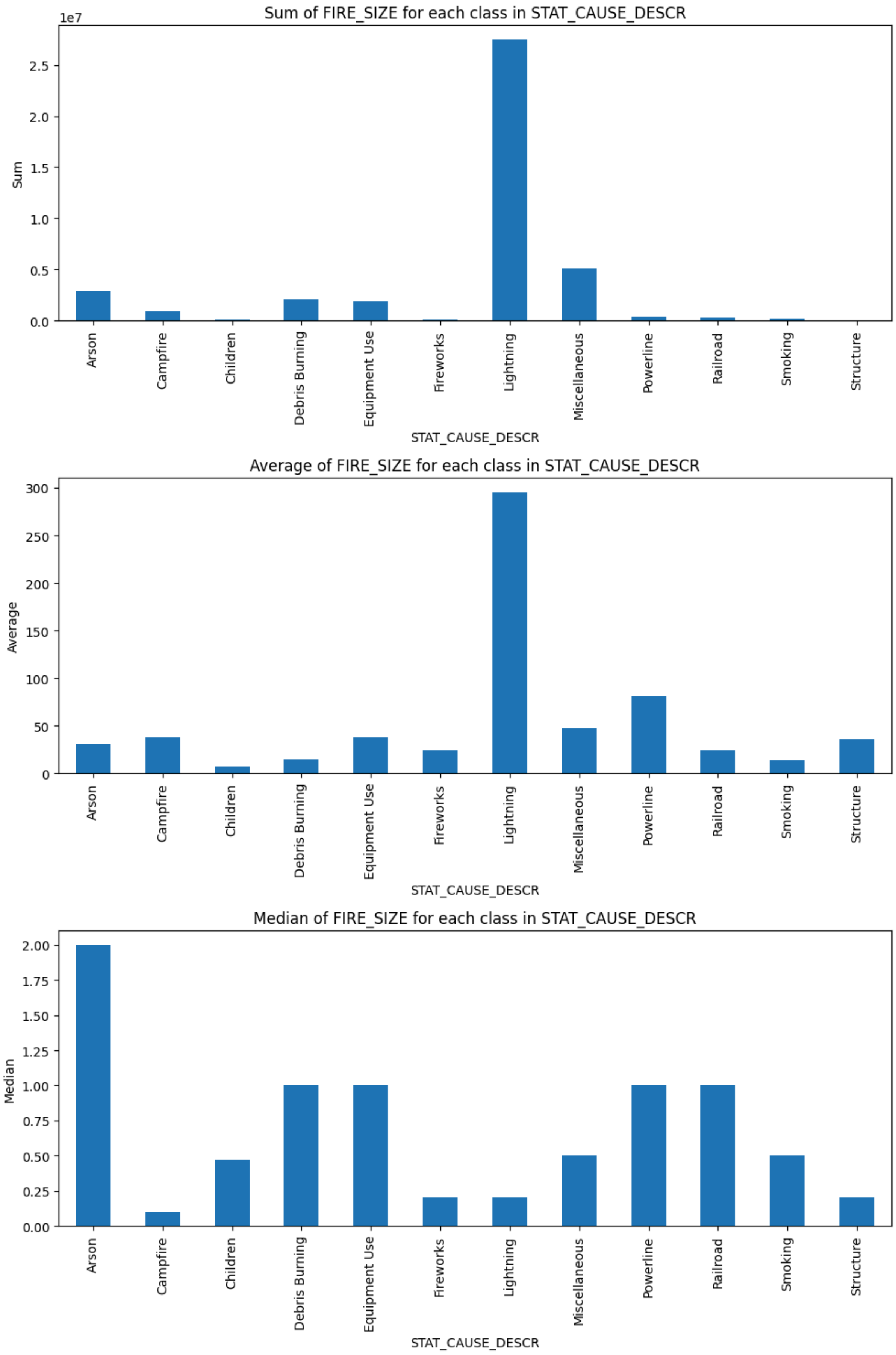
We take those categorical features, catagorical_time, catagorical_geospacial and create new features which represent the frequency_encoding

In conclusion, for these categorical features we notice the model performs better with only the new frequency_encoding features so we continue working only with them.

Numerical

EDA

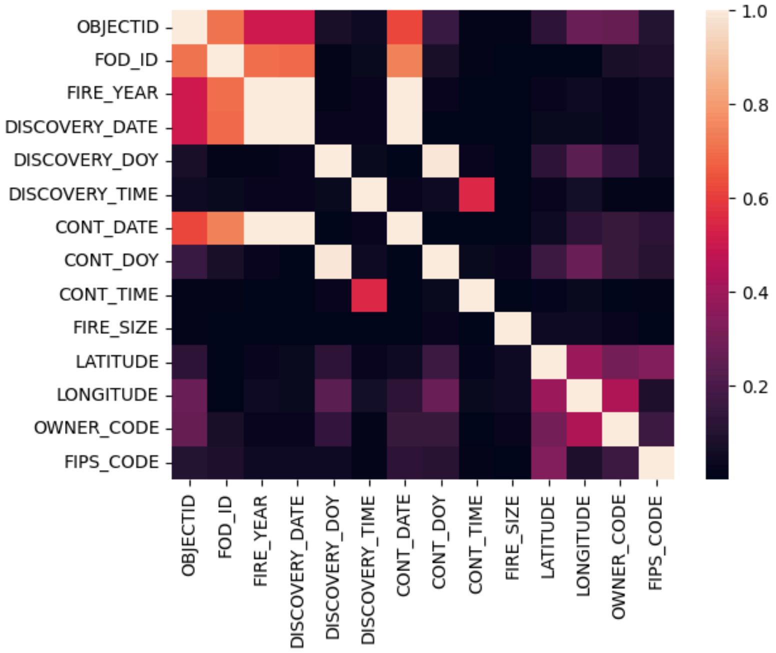
Now we analyze the numerical features:
First we check how the fire size is distributed among the classes


Sum of FIRE_SIZE for each class in STAT_CAUSE_DESCR
1e7
Sum
0.0
0.5
1.0
1.5
2.0
2.5
Arson
Campfire
Children
Debris Burning
Equipment Use
Fireworks
Lightning
Miscellaneous
Powerline
Railroad
Smoking
Structure
STAT_CAUSE_DESCR
Average of FIRE_SIZE for each class in STAT_CAUSE_DESCR
Average
0
50
100
150
200
250
300
Arson
Campfire
Children
Debris Burning
Equipment Use
Fireworks
Lightning
Miscellaneous
Powerline
Railroad
Smoking
Structure
STAT_CAUSE_DESCR
Median of FIRE_SIZE for each class in STAT_CAUSE_DESCR
Median
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Arson
Campfire
Children
Debris Burning
Equipment Use
Fireworks
Lightning
Miscellaneous
Powerline
Railroad
Smoking
Structure
STAT_CAUSE_DESCR

we can see that there is a correlation between the fire size and the target classes. So we can understand that this feature is important for our model.

Leakage

Leakage is information that we are not supposed to know during inference time. From the data set description, we can determine that some columns are either leakage or useless. Thus we should drop these columns.

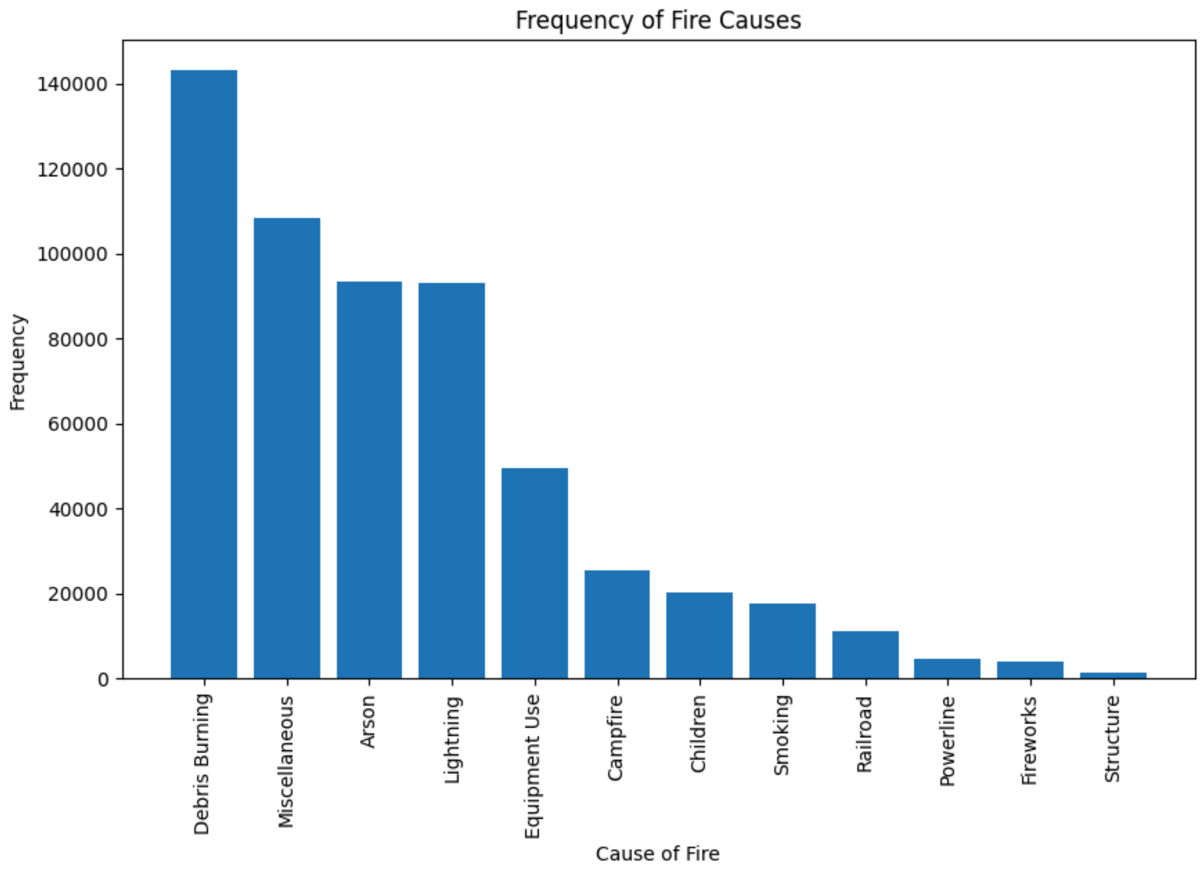


We will not use FIRE_NAME and FIRE_CODE since from the description of these two columns are leakage.

Additionally, there are three ID columns: 'OBJECTID', 'FOD_ID', and 'FPA_ID', which we have observed do not contribute to improving the model. We are concerned that even if they were to help, they could potentially lead to data leakage.

Labels distribution

To get a sense of the data we plotted the distribution of the different labels (‘Frequency of Fire Causes’). We noticed that the distribution is not close to unified and the second most frequent cause is ‘Miscellaneous’ which might be hard later on because it aggregates different unknown labels.



Experiment

Model Experiment 1 - (Baseline)

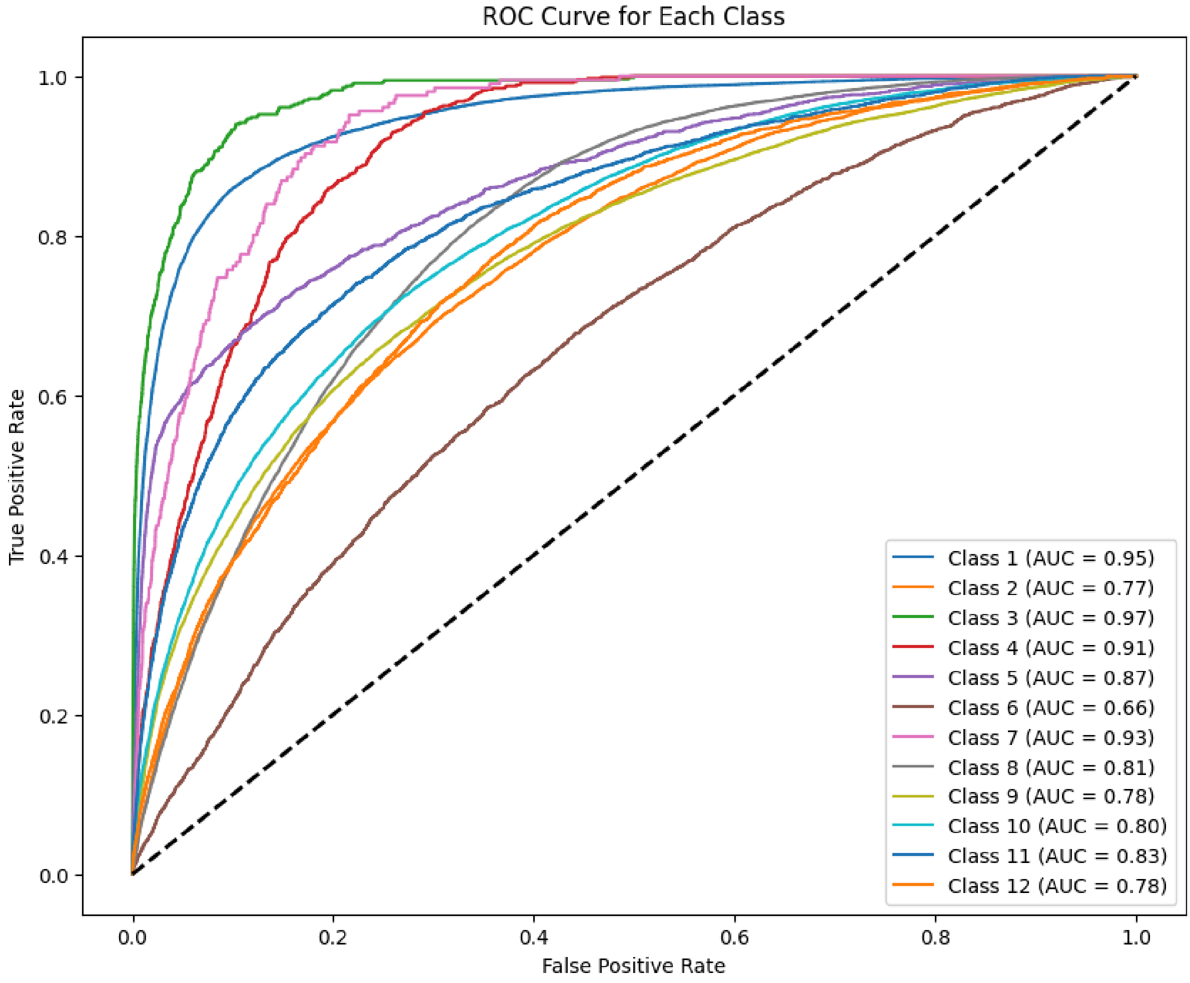
We conducted a comprehensive exploration of four different machine learning models:

1. Decision Tree classifier
2. Cat Boost classifier
3. Random Forest classifier
4. XGBoost

After preprocessing the data and splitting it into training, validation, and test sets, we extracted the target column and trained each model on the numeric features of the dataset. Upon evaluation, XGBoost emerged as the standout performer, surpassing the 80% accuracy threshold, a notable achievement compared to the other models. The Decision Tree classifier achieved an accuracy of 0.4798, the CatBoost classifier scored 0.5107, and the Random Forest classifier reached an accuracy of 0.5700. However, XGBoost notably outperformed the others with its accuracy exceeding 0.82.

Given XGBoost's superior performance we made the decision to adopt it as the primary model for predicting wildfire causes. Consequently, we discontinued the use of the other models, recognizing XGBoost as the most suitable choice for our predictive modeling task.

We plotted an ROC curve for each class in order to understand the performance distribution of the classes.



Model Experiment 2 - add more features from new data

At this experiment we thought about new features that could correlate to some of the labels in the target and add heuristics for them.

The Experiment involved downloading the *NOAA Storm Events* Database dataset and preparing it for analysis. The preparation included converting date columns to datetime objects and filtering the dataset to include only records from the years 1992 to 2015, aligning with the timeframe of the wildfire dataset.

To facilitate matching states between the two datasets, a dictionary of state abbreviations was created. This dictionary was then used to replace full state names with their corresponding abbreviations in the storm events dataset.

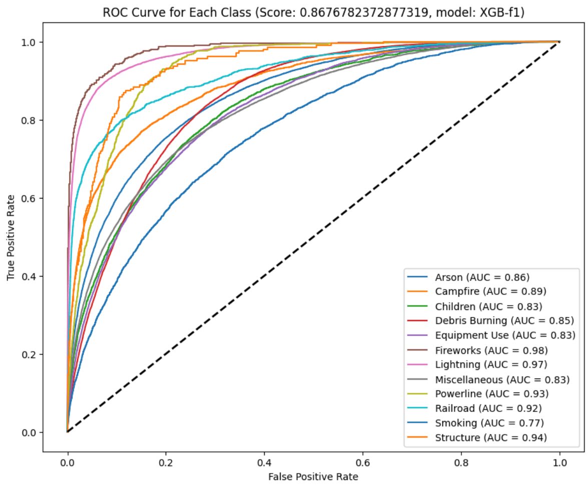
Subsequently, the focus shifted to the wildfire dataset. I introduced a new column, 'STORM', which served as a boolean indicator to signify whether a wildfire occurred concurrently with a storm in the same state. This determination was made by checking if the combination of the wildfire's discovery date and state matched any entries in the storm events dataset.

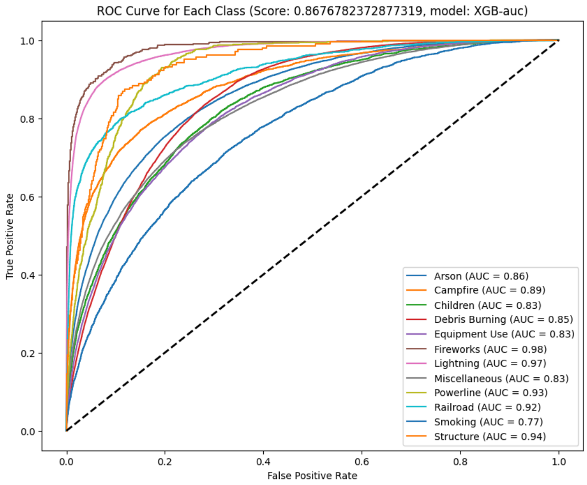
In summary, by incorporating storm event data into our analysis we improved the model performance.

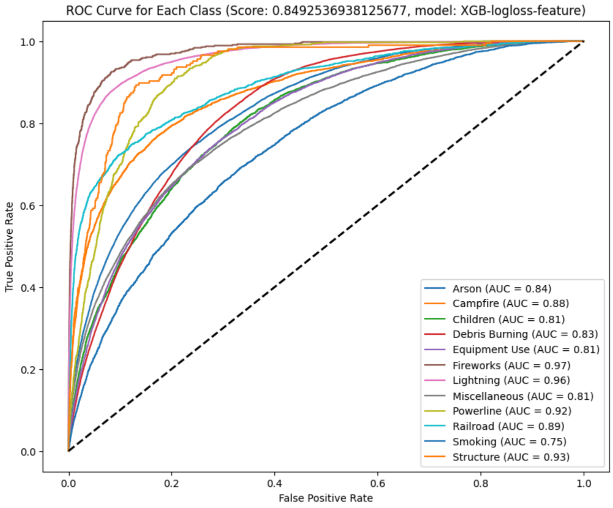
Model Experiment 3 - Exploring Model Optimization with Alternative Loss Metrics

To further enhance the model's performance, experimentation was conducted with alternative loss metrics.

Several loss metrics, including **f1**, **auc**, **logloss-feature** were tested. Despite these efforts, the results indicated that the choice of loss metric did not significantly impact the model's performance. The alternative loss functions failed to yield discernible improvements in predictive accuracy or reliability compared to the original approach.

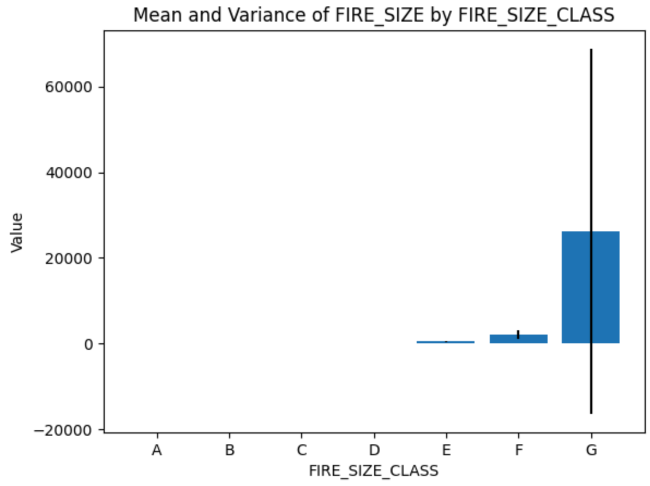


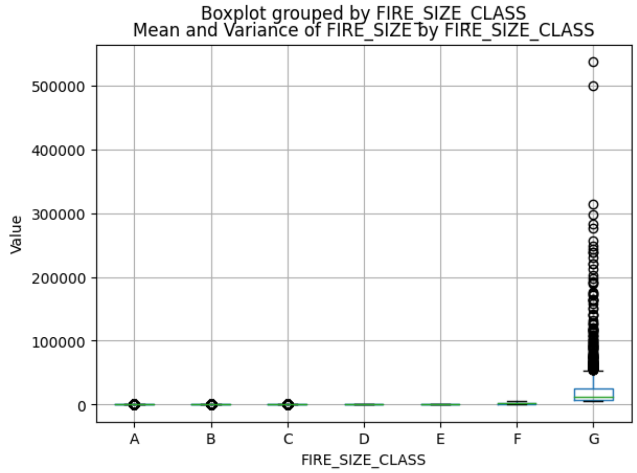




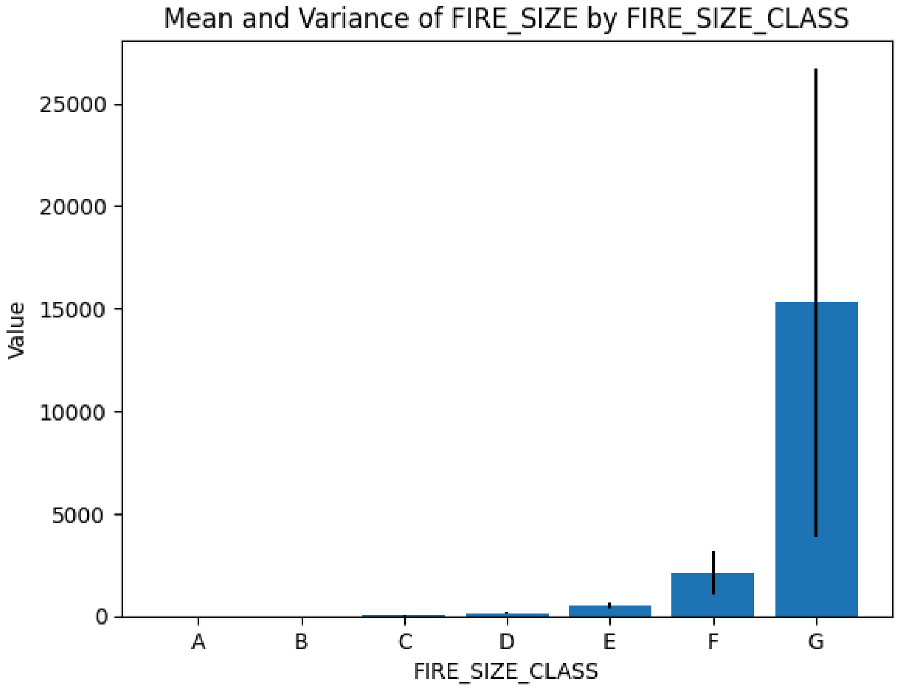
Model Experiment 4 - Anomaly detection (failed attempt)

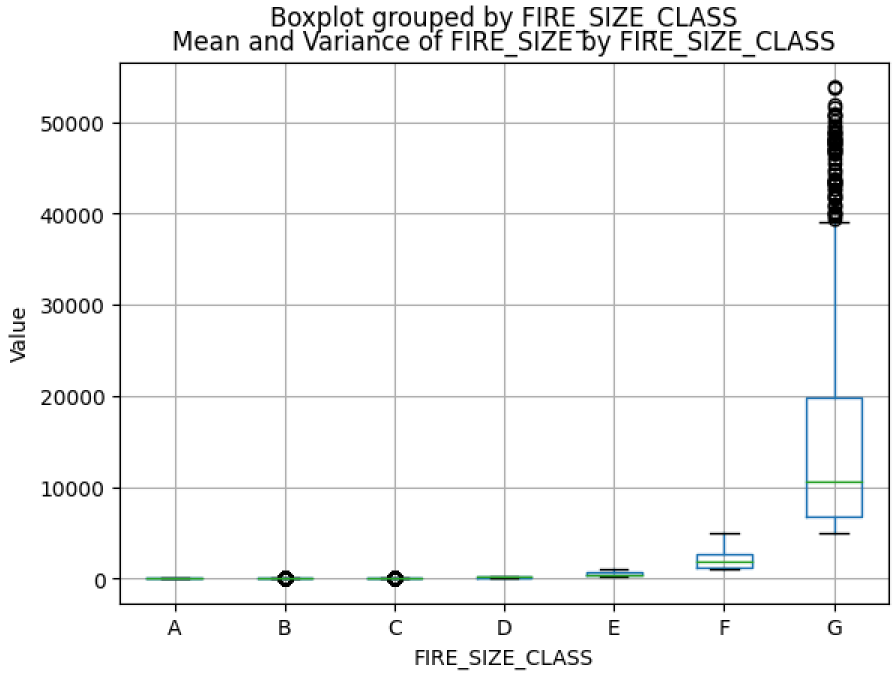
We'll try to use anomaly detection of fire size class features. We calculated the mean and variance of the fire size

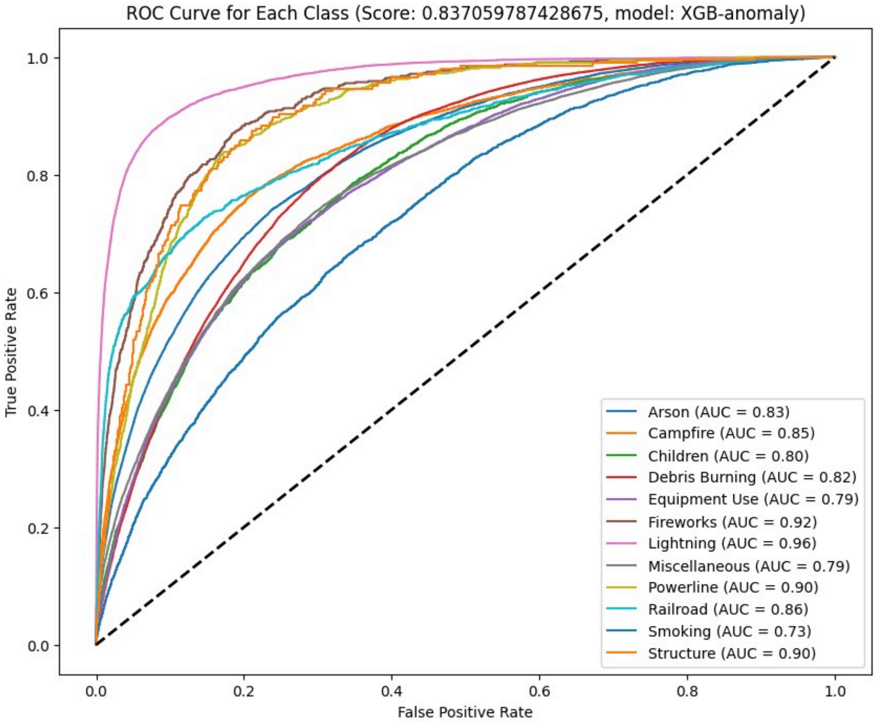




We have a pretty big variance, and there are many outliers. We tried to get rid of these outliers. We looked at the firesize afterwards.


Mean and Variance of FIRE_SIZE by FIRE_SIZE_CLASS
Value
25000
20000
15000
10000
5000
0
A
B
C
D
E
F
G
FIRE_SIZE_CLASS

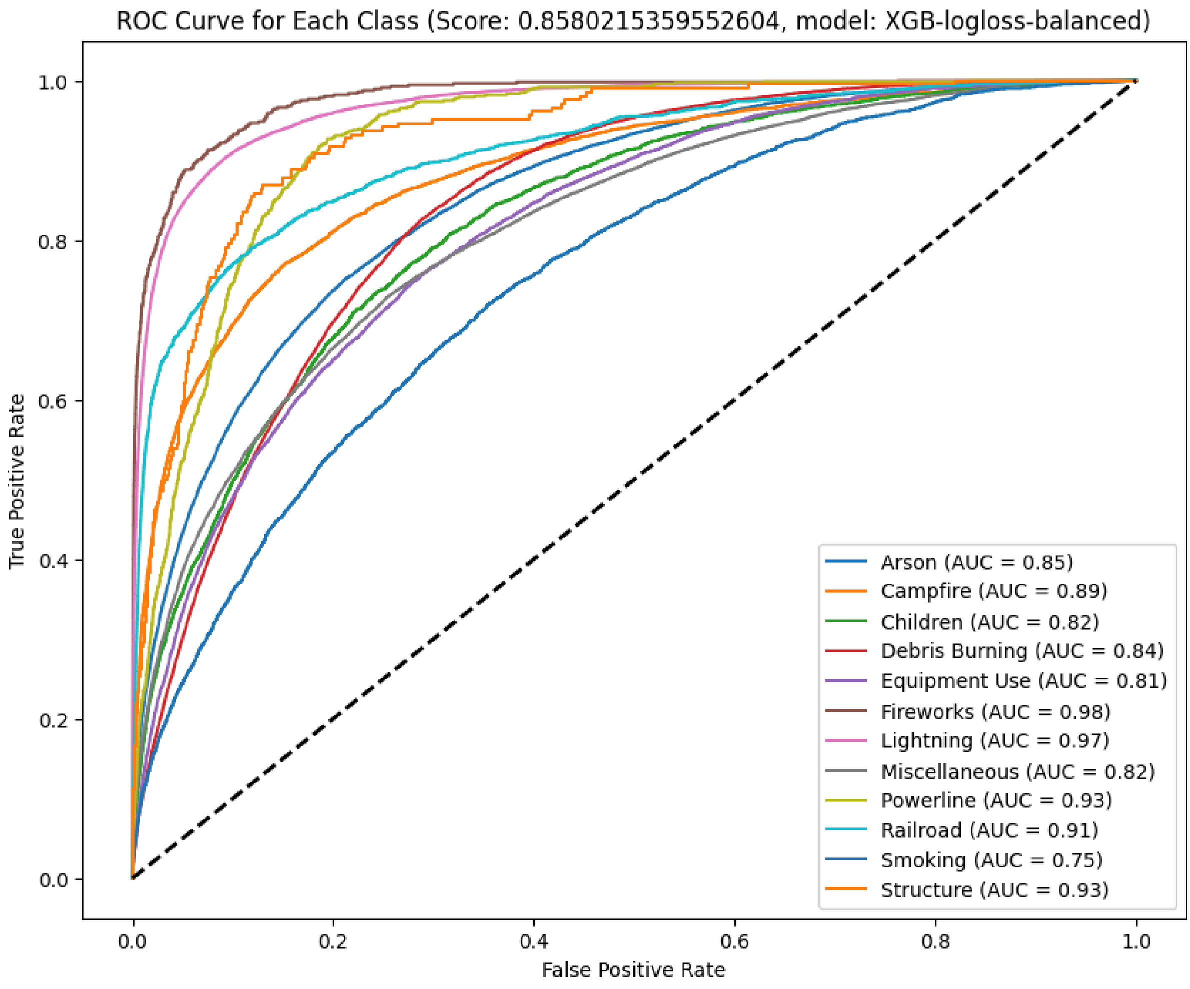




Getting rid of the outliers didn't increase the performance so we didn't use it eventually.

Model Experiment 5 - Balanced

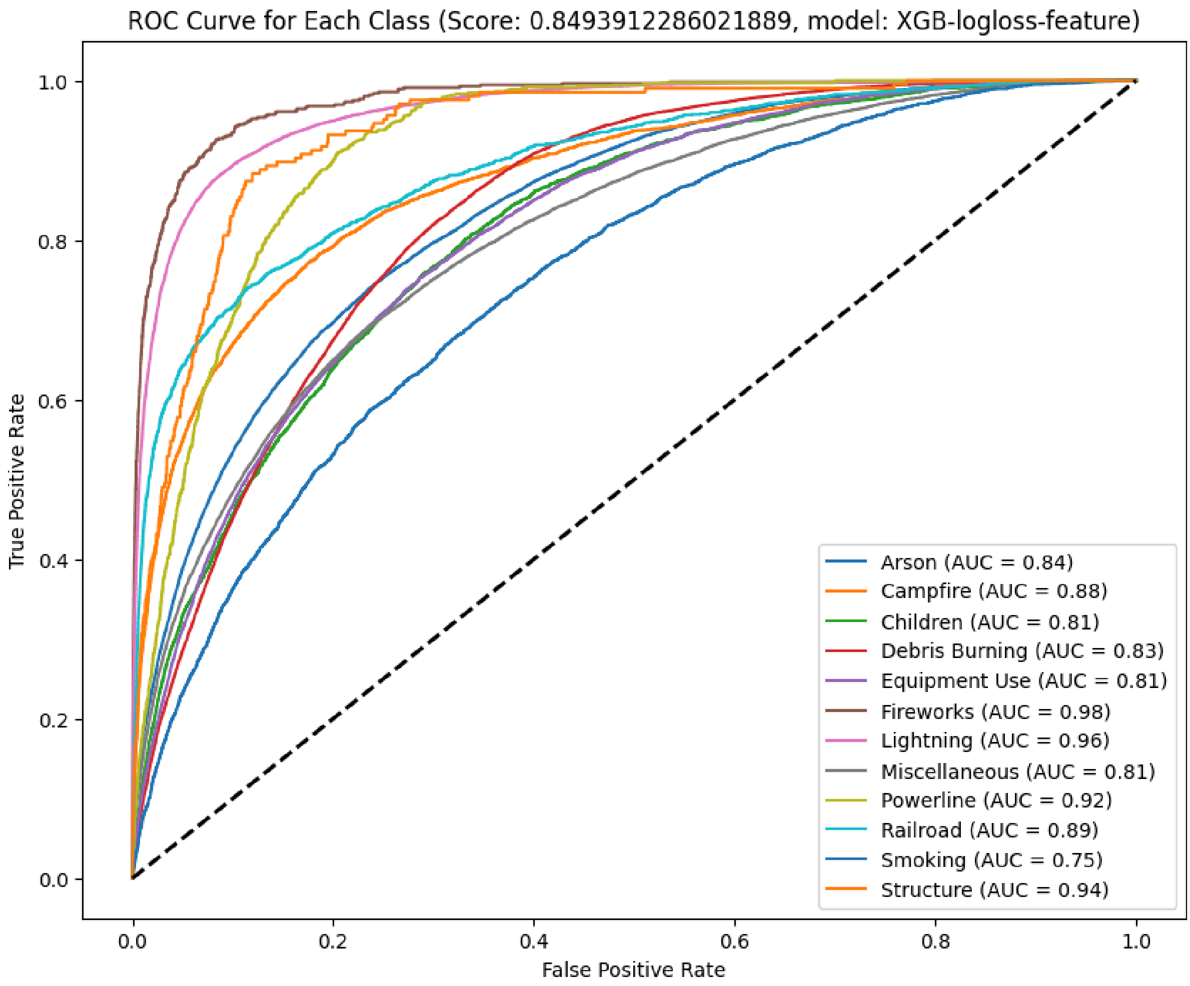
We see that the model doesn't perform so well, so we suspect that it might be an imbalanced problem. We tried to balance the data by giving sample weights that punish more for being wrong on minor classes. This is the performance:



We saw that it performs worse than before. We went on and checked the number of samples in each class. It turns out that the target class “Smoking” has much more samples than average, so it has less than 1 sample weight and would make the prediction worse. Eventually we decided to give up on this thread.

Model Experiment 6 - Feature Selection

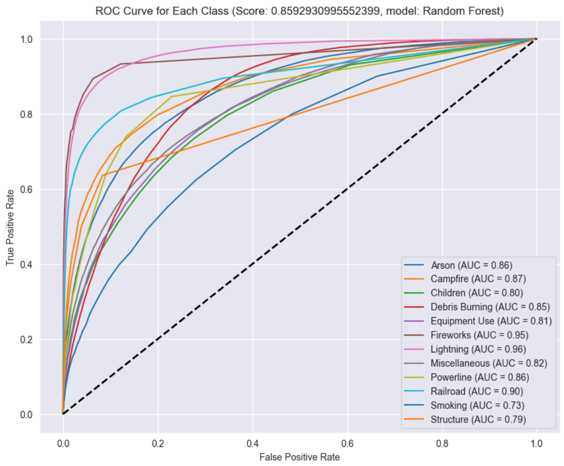
So after we had the model, we tried to find the best performing features. And We tried to run the model with the top 30 of the 75 features we have. However, it performs worse.

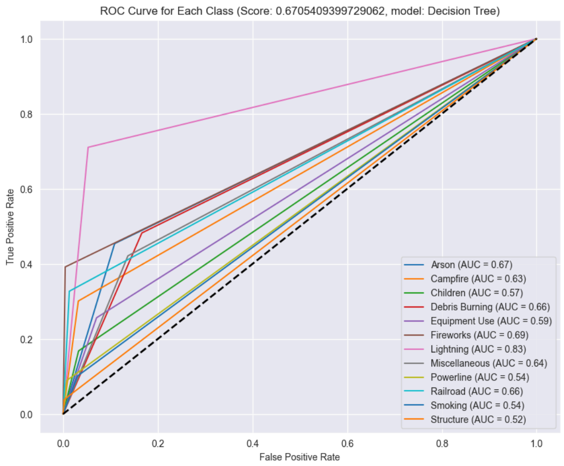


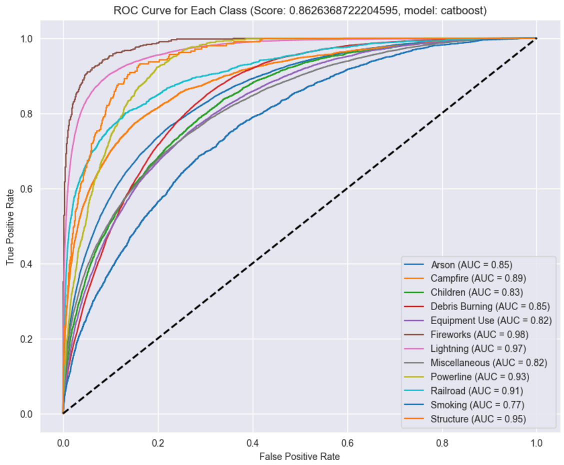
indicating that reducing model complexity does not help us. If we have more time, maybe what we should do is to increase model complexity by adding more encodings.

Model Experiment 7 - Model Selection

After we acquired the features, we proceeded to perform model selection. In order for our data to fit in the other models, we have to encode our categorical data. We have two categorical data: “MONTH” and “DAY_OF_WEEK”. We left it as categorical because there is no ordinal relation between them so it’s not reasonable to record them as numbers. So as what I did above, I calculate the distribution of target classes, and then I use KL-Divergence to calculate the distance of our target classes to the original target class, and use that as our encoding. This way we can sort month by the importance of providing information for predicting targets.
Then we plug in the data into three models: Random Forest, Decision Tree and CatBoost.
Then here are the results for each model:



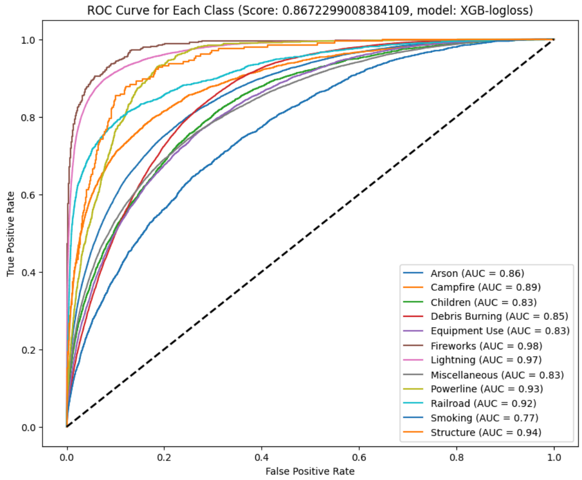


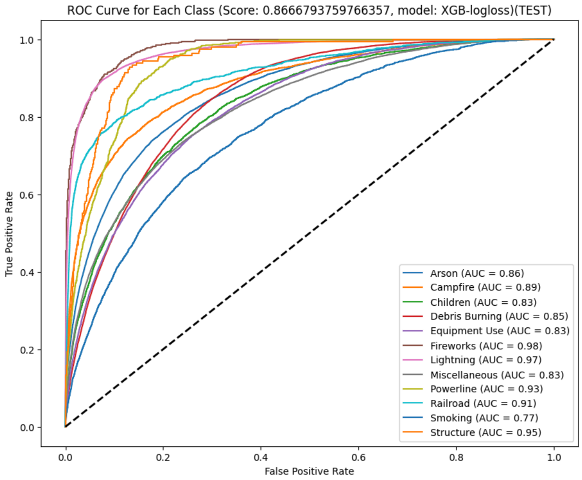


We can see that random forest and catboost performs almost as good as XGBoost. WIth 86% of **auc** score. I attribute this to the effectiveness of gradient boosting. Eventually, XGBoost still wins by less than 1%, so we will select it as our final model, but I think the difference is not significant.

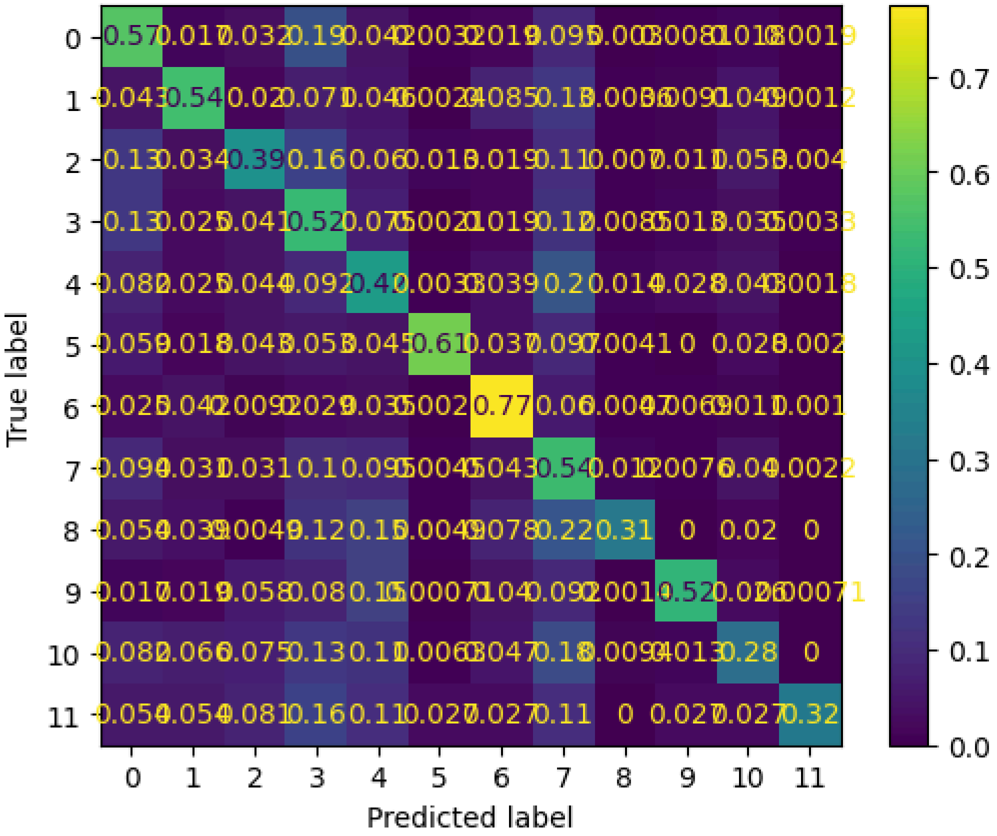
Model Experiment 8 - Final Results

So eventually, we take all the features we mentioned above, and run the model to get the best result. The best result is the following, with a **0.867 auc** score on validation set, and **0.866 auc** score on test set.





Also we will attach the confusion matrix of the model



We can see that this performance is quite acceptable.