A Machine Learning Approach:

Classification and Clustering of Wildfires in the USA

HEC Lausanne Spring 2022, Machine learning in Business Analytics

Group G

Noman Bashir
Rudrapriya Bose
Chung Hsiang Cheng
Yinyu Kao
Emile Paris

Table of Contents

Introduction

Motivation

Wildfires are some of the most economically devastating disasters to occur on a regular basis. They cause billions of dollars worth of damage in the US every year. According to estimates from the NOAA, the total cost of wildfires in 2017 and 2018 was more than $40billion.[1] The US spends an annual $1.6 billion fighting wildfires.[2] Due to climate change and erratic weather patterns, the frequency of large wildfires has increased in recent years. Unfortunately, the US Forest Service does not have the means to cover all the wildfires that break out across the country every year. In 2021, a total of approximately 60 thousand fires occurred burning about 7 million acres of land.[3] During the summer months, multiple fires can break out at the same time leaving the US Forest Service short-handed and unable to deal with all of the fires. However, not all fires are of serious concern. A large proportion of fires break out around campsites and do not require intervention to be put out once reported. Thus, there exists a need to have a system which can predict whether a fire that has been discovered has the possibility of large-scale damage. According to research[4], 85% of wildfires are caused by humans, however the size of the fire depends a lot upon the weather conditions that exist prior to the breaking out of the fire.

In this report, we endeavour to test out whether machine learning approaches can be used to predict which fires have the possibility to become damaging. We utilize data from the US Forest Service, combined with weather data of the location of the fire in order to build and train our models.

Data Sources and Description

The dataset utilized here is a sub-sample of 1.88 Million US fires. This subsampled dataset contains data about 50 thousand wildfires occurring in the US along with geospatial coordinates, weather data over the month preceding the fire, the remoteness of the location and vegetation data. Details regarding the dataset can be found in **Appendix 1.**

Data Wrangling and Exploration

Initial Data Cleaning

The first step of the predictive process was to transform the data into a usable format. There were several fires which had no accompanying weather and vegetation data and thus had to be removed from the dataset. Dates were transformed into R's POSIXct format for easier wrangling. Putout times were in a non-usable format and were recreated using date formats. Not all fires had putout times as some did not require intervention and extinguished themselves.

The dataset also had several extraneous columns that would not be useful in predictive situations and only existed as reference points; notably data regarding weather station identifiers and location of the weather files.

Lastly, there were several columns that had to be converted into factors from character: fire_size_class, stat_cause_descr, state, Vegetation and discovery_month.

Exploratory Analysis

After the basic wrangling was complete, a basic exploration was conducted on the dataset. **Fig 1.1** and **Fig 1.2** show the distribution of fires across the different classes and the leading causes of most fires.

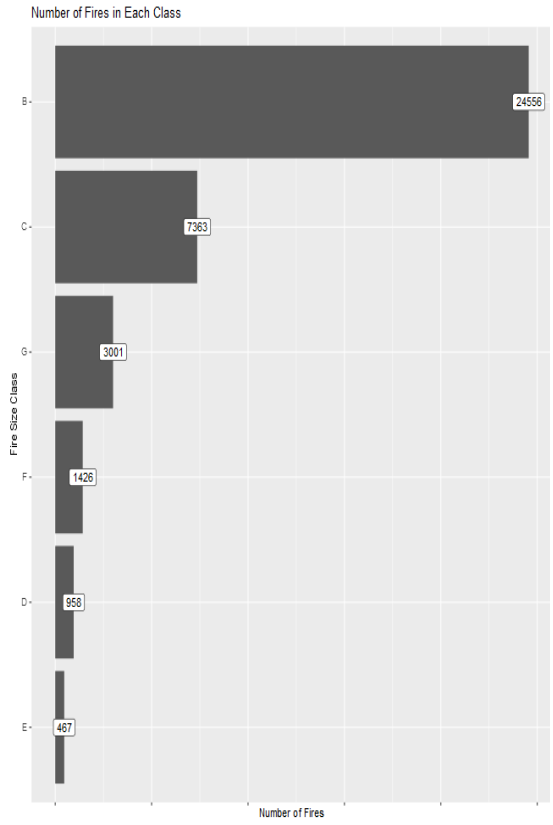


Fig 1.1 – Distribution of Classes

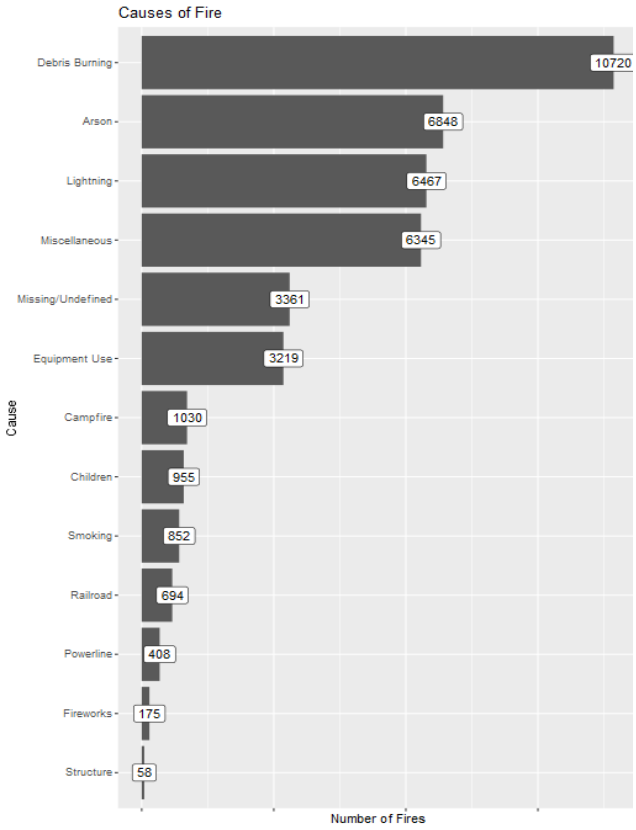


Fig 1.2 – Causes of Fires

The dataset is severely unbalanced. This is understandable as there are far more smaller fires of "Class B" than there are larger fires. Most fires do not grow beyond a certain size. The fire causes present a more interesting viewpoint. Most fires are caused by debris burning. The fire causes also reveal another anomaly – a lot of the fire causes are unknown i.e. Missing/Undefined. As we have factorised the "Causes", missing/undefined causes will only obfuscate the link between fire size class and the cause, thus they were removed. Another notable observation is that "Fireworks" and "Structure" represent only 175 and 58 instances. These instances are too few to keep separate. However, there is a possibility that they could be over-represented in one fire class which would make their merging with another cause counter-productive. **Fig 1.3** shows how different fire size classes vary by cause.

There is no clear link between a specific fire class and a cause. However, it can be noted that Lightning seems to be represented to a large degree in larger fires. Most of "Class G" fires are caused by lightning, for example. Moreover, we can note that structure and fireworks are not really overrepresented in any particular class, they only make up small percentages of the causes for each class and thus can be joined with the miscellaneous category.

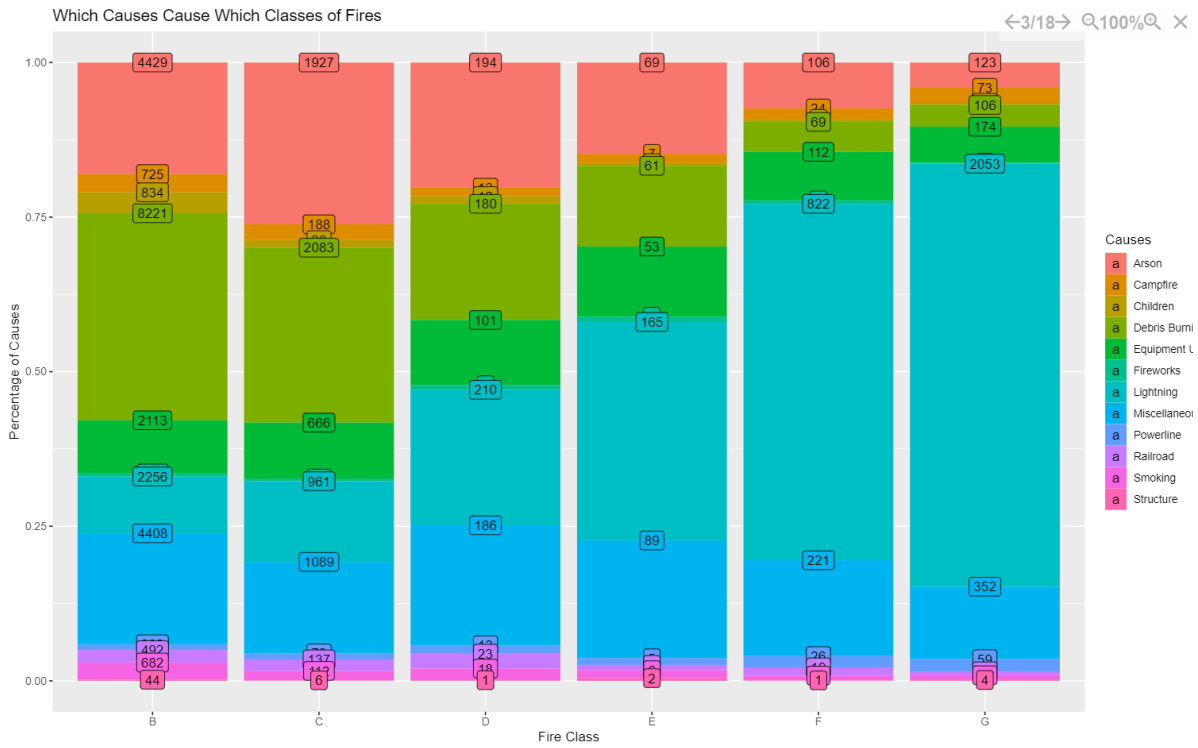


Fig 1.3 – Comparison of Causes by Fire Size Class

Next, an analysis was done on time to see the number of fires per month and also the average fire size of month. **Fig 1.4** displays the number of fires per month in our dataset. As this is a subsampled dataset, this may not be fully accurate of the actual number of fires that break out in a month. This is a limitation of working with subsampled sets. As we can see, March and April seem to have the most fires in the year.

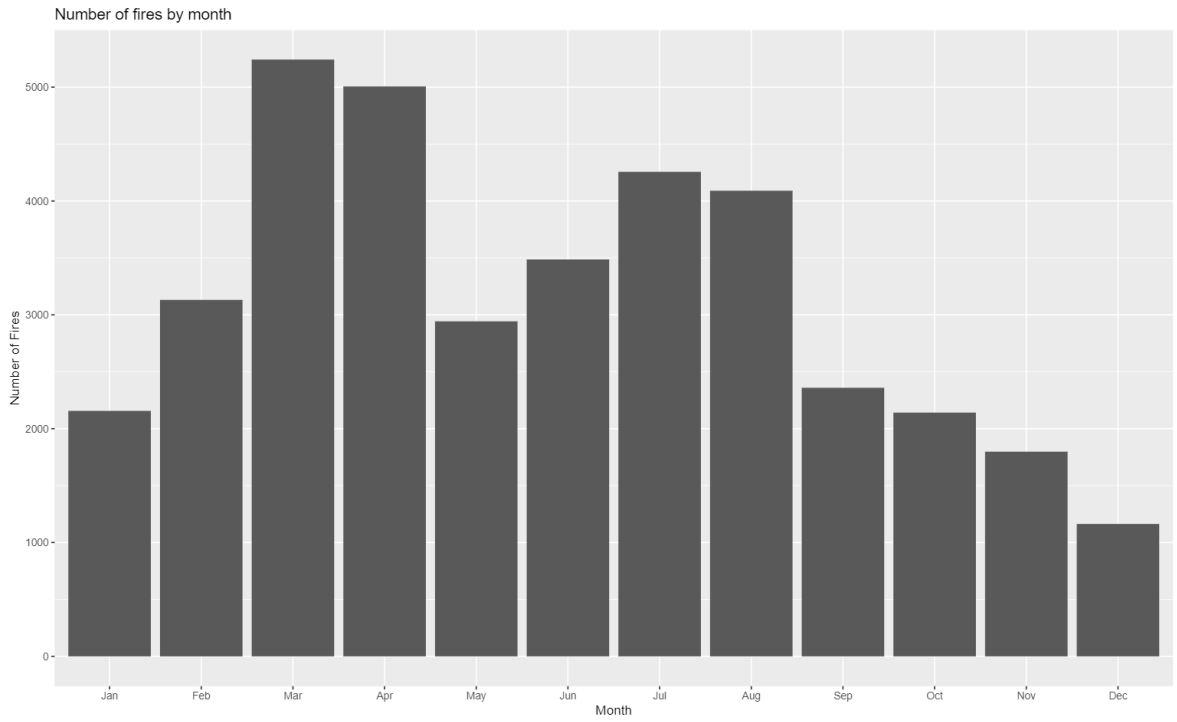


Fig 1.4 – Number of Fires by Month

The average fire sizes tell a different story. Predictably, it's the summer months that show the largest fires. **Fig 1.5** showcases this relationship.

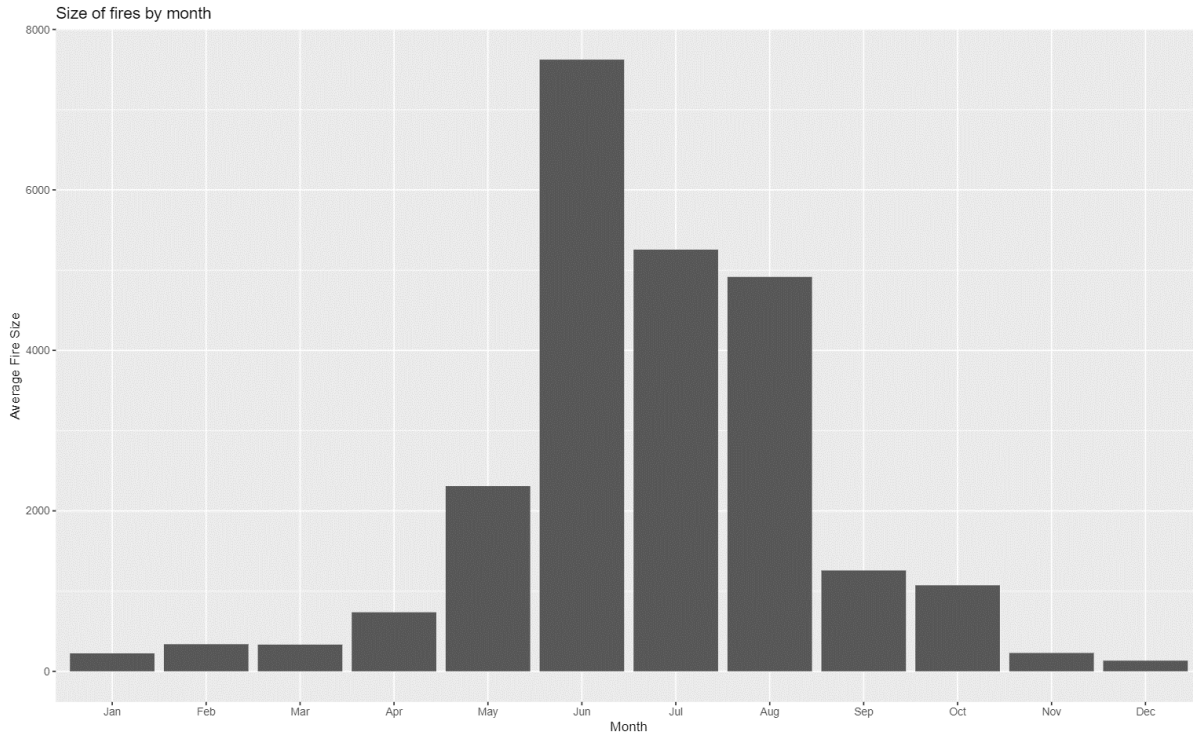


Fig 1.5 – Size of Fires by Month

Fig 1.6 delineates the relationship between causes and area burnt. While there is a large variability across all causes, lightning stands out as the leading cause in larger fires. This confirms our earlier analysis where we noted that lightning was overrepresented in larger classes.

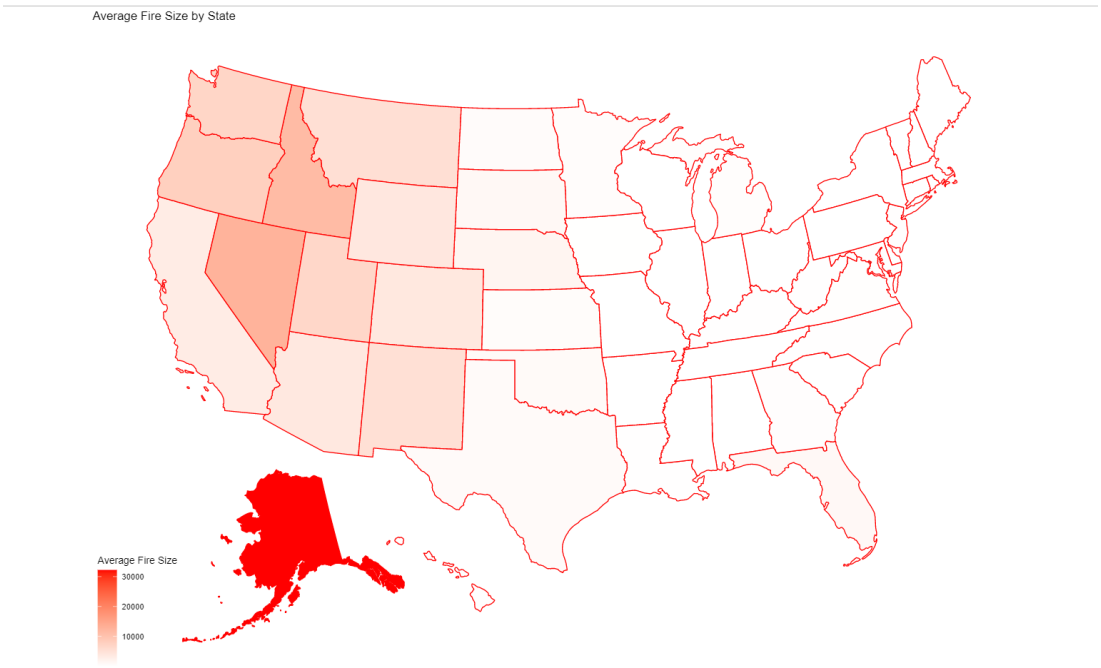


Fig 1.7 – Average Fire Size by State

In terms of distributions across states, **Fig 1.7** shows that the western states tend to have larger fires. The largest fires happen in Alaska which comes as no surprise given that it is mostly woodland forest. Any fire that breaks out in Alaska is prone to spreading quickly. The Western United states also suffers from droughts more than the east and therefore, the likelihood of a larger fire is greater. At the same time, this information can be contrasted by the number of fires that break out per state that can be seen in **Fig 1.8**. The density of the points shows that there are more fires in the east than the west of the country. However, since the east does not have the same kind of vegetation Alaska or the Western US has, the fires tend to be smaller. In short, the state in which the fire breaks out should serve as a good predictor for the size of the eventual fire.

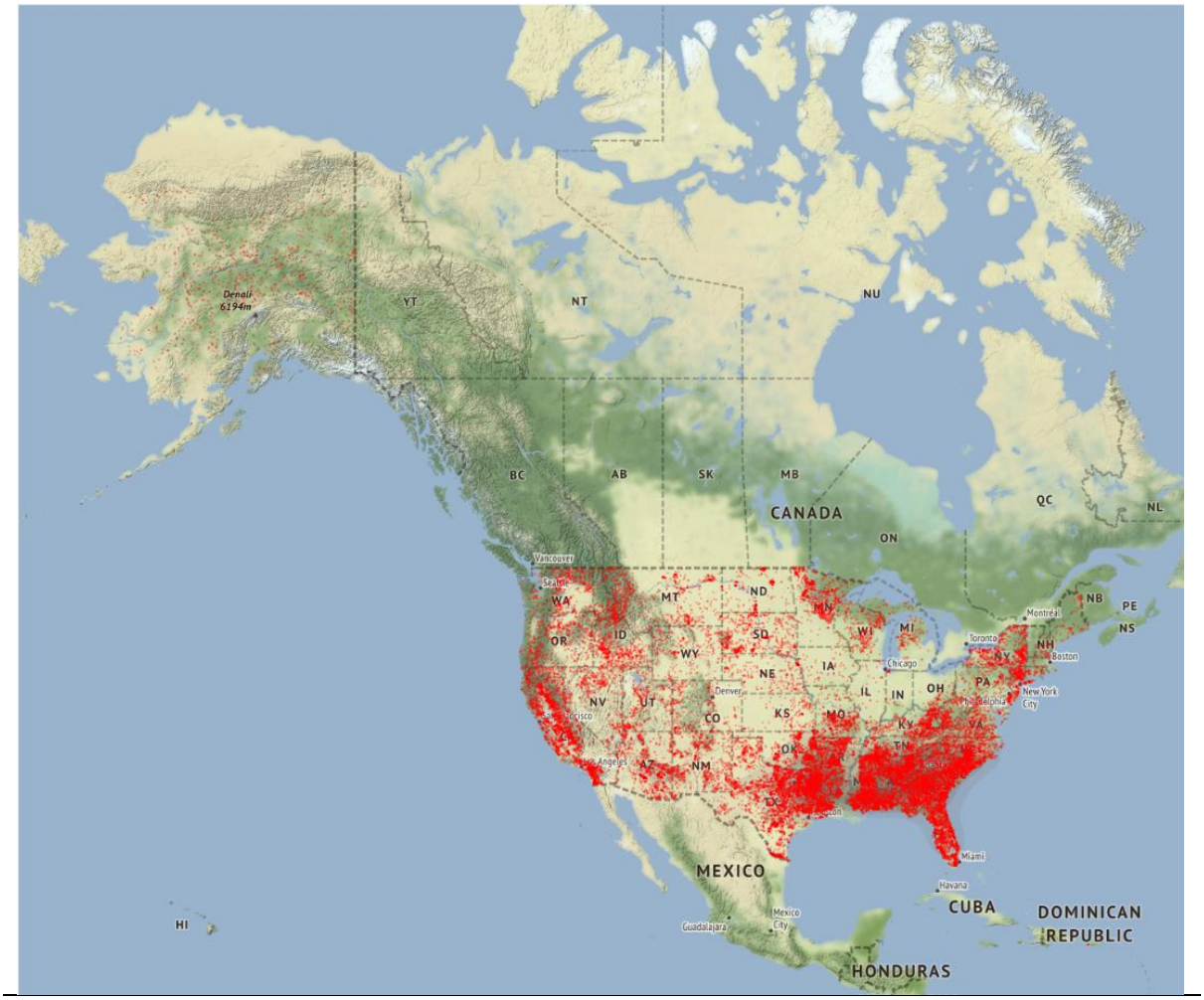


Fig 1.8 – Geospatial Layout of Fires Across the US

Fig 1.9 shows the variation in fire size (log-scale) and the cause. Lightning stands out as the major cause of large-scale wildfires with a median higher than all other causes. There are numerous outliers in every category, however.

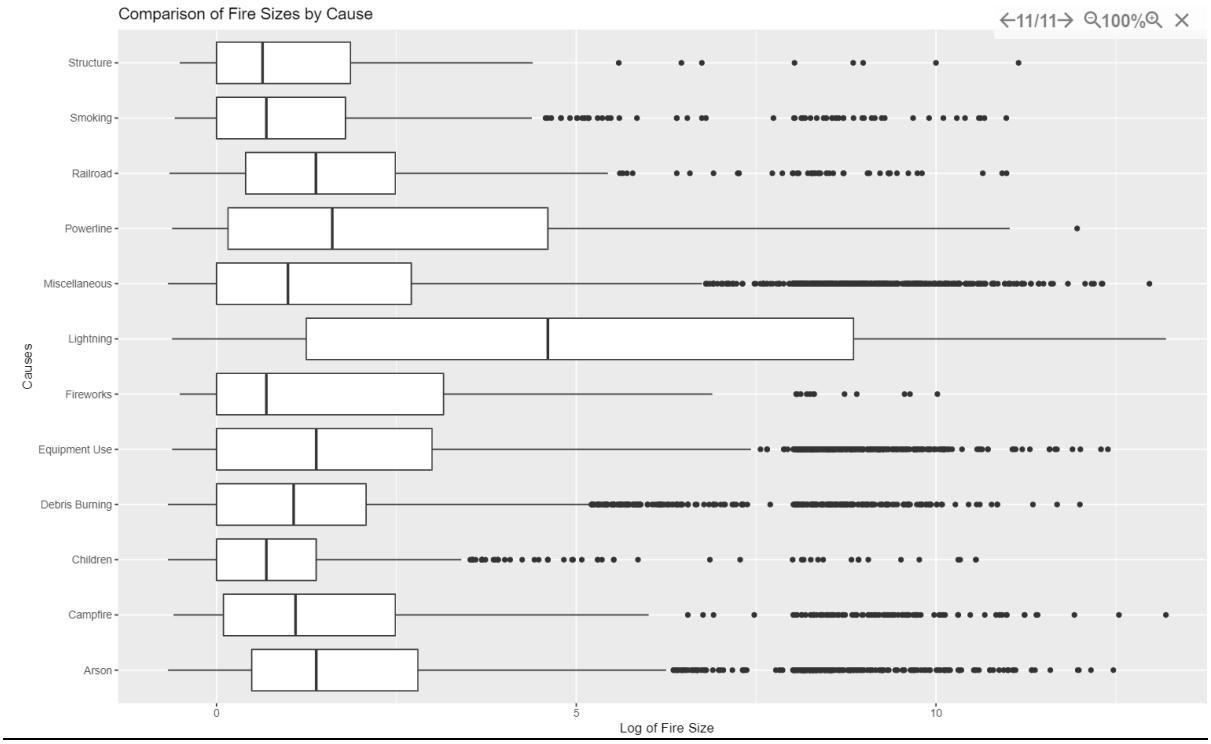


Fig 1.9 – Size of Fires by Cause

In terms of vegetation, types 4 and 14 show a greater variability and tend to have larger fires within their interquartile region. These two are *Temperate Evergreen* and *Desert* land covers. It makes intuitive sense why the areas around deserts tend to have larger fires, as the land is very arid and shrubbery is dry and catches fire very easily. However, there does not seem to be a clear connection.

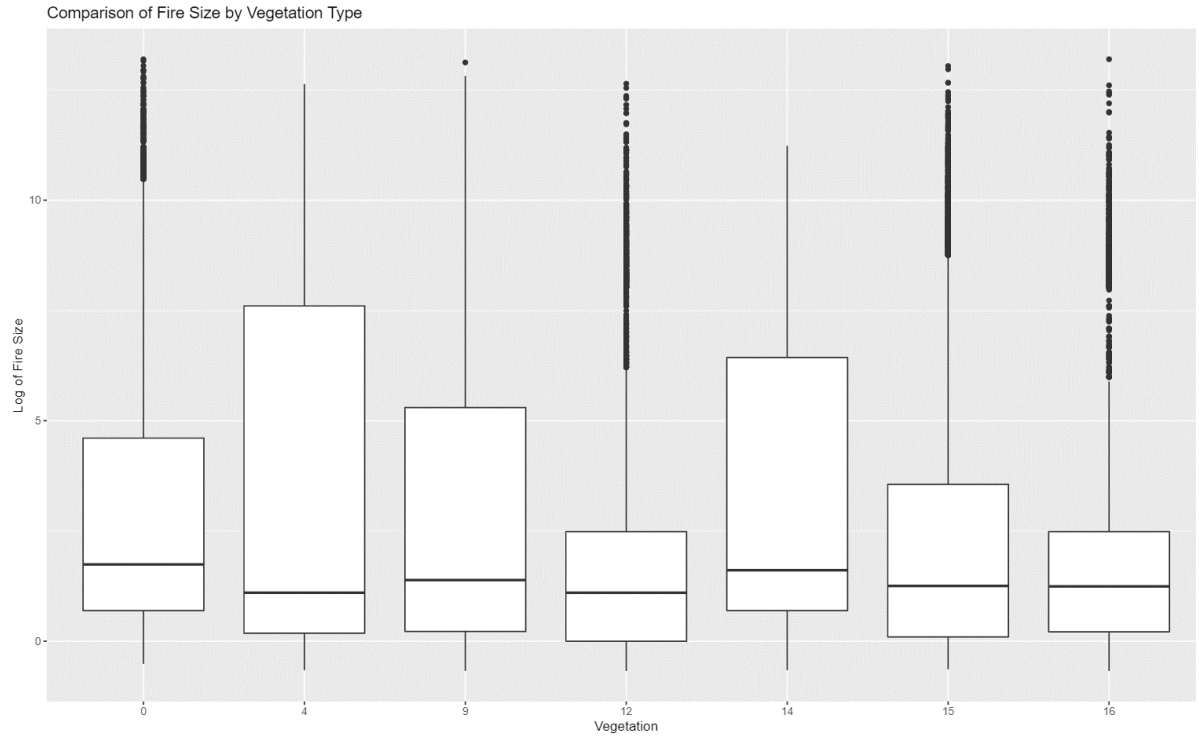


Fig 1.10 – Fire Size by Vegetation

Lastly, we can see the distribution of fire sizes and remoteness in **Fig 1.11.** There is an interesting trend as the fire sizes tend to be larger when there are a lot of people (i.e. remoteness is low) and when the break-out location is distant. Once again, this makes intuitive sense as areas closer to people are more prone to have accidents starting fires whilst areas that are further away are harder to put out. At the same time, this makes the classification task more difficult as remoteness is not a linear predictor.

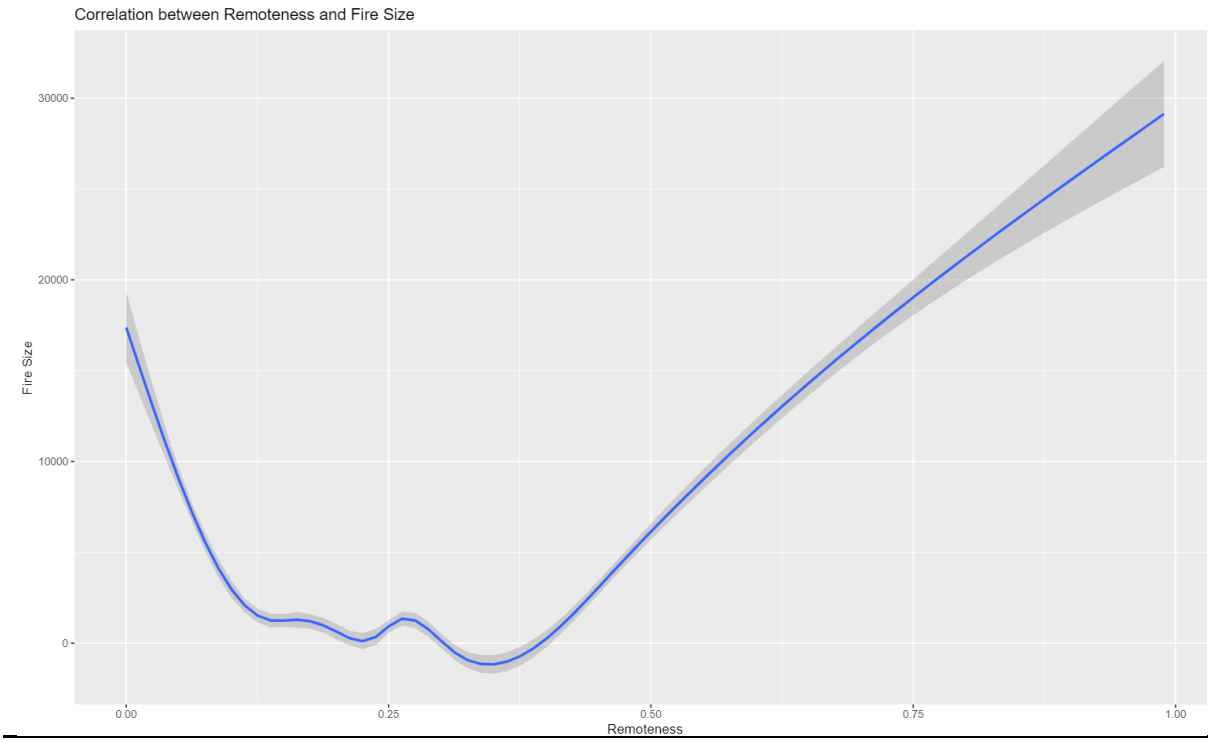


Fig 1.11 – Size of Fires by Remoteness

Further details regarding distribution of fire sizes, put-out times and variations of the number of fires per month can be found in **Appendix 2.**

Data Wrangling

Having considered all the facets of the data and cleaned it up, there are a few final steps required to make the data ready for use. First, there are several weather metrics that share similar data – *temp_pre_cont*, *temp_pre_30* et al cover the same metric (see **Appendix 2**) but over differing time periods. For now, we will simply use *temp_pre_30* i.e. the average of the temperature, weather, humidity and precipitation for the last 30 days and remove the other features. Next, we will scale all the numeric features to deal with any outliers that may skew our models.

Subsampling

As noted in **Fig 1.1**, the data is heavily unbalanced with most of the observations coming from "Class B" i.e. smaller fires. In order to not allow this unbalanced dataset to influence our models, we will utilize subsampling to create a subset of the data with balanced classes. Subsampling was utilized instead of resampling as the dataset is already quite large and subsampling will not remove too many instances.

The dataset was first split into training and test sets using an 80-20 split as the test set is not supposed to be balanced. Then, using just the training set, the class with the fewest instances was calculated. This turned out to be "Class E". Next, the same number of rows were sampled from all other classes without replacement. Finally, this new dataset was binded together.

However, there is cause for concern here as subsampling means our test data is larger than our test data. However, we still choose to subsample as there are enough representative instances for each class (387 each) making up a total of 2322 instances while the test data has 8000 instances. Computational constraints meant that resampling was not tenable as even the simplest of models would take over half an hour to run. Another solution considered was changing the split to 95-5, however, this would still mean that the lowest class would be no higher than 500 instances. There did not seem enough of a difference between 387 and 500 instances to warrant running all the models again and thus, the split remained as described.

Modelling

Choice of Models

There are a variety of models available for testing for a classification problem. However, the unique characteristics of the dataset made some of them more difficult to use. Naïve Bayes could not be used as it assumes that the features are independent of each other. However, this dataset contains weather data and a lot of the metrics – wind and precipitation, for instance, may be linked. As such, the assumptions of naïve bayes would not hold and thus it was not considered. Logistic regression was also similarly ignored as it is mostly used for binary classification problems. However, the problem under consideration is a multi-class problem. While workarounds exist for using logistic regression for multi-class problems i.e. running several models for each class separately and amalgamating them, this would be a resource intensive endeavour and thus was not considered. SVM was not considered because of computational constraints, experiments on even reduced problems would crash R.

The final selection of models to be tested:

1) K-NN
2) Neural Nets
3) Random Forest

These models would be compared on accuracy, balanced accuracy and the Cohen's Kappa.

K-NN

The KNN model was trained on a 10-fold CV with tuning of the hyperparameter *k*. The metric under consideration was "Accuracy". Values from 1 to 10 were considered for *k*. The training optimized a value of *9* for *k*. Following this, the finalised model was run on the test data with this tuned hyperparameter.

The results of the model are summarized in **Fig 2.1**. The overall accuracy of the model is quite low at 35%. If we look at the specificity, we can see that the model is competent at predicting when an instance *does not* belong to a class but performs rather poorly when sensitivity is under consideration. Surprisingly, despite there being fewer instances of Class G

in the overall dataset, the best sensitivity/specificity trade-off occurs in Class G as the model both predicts a large proportion of actual positive cases and actual negative cases.

The model however does outperform a random model so is not completely useless. The Cohen's Kappa is **0.1392** which indicates that it is only marginally better than a random model, however. The accuracy of a random model would be around 17% and the KNN model has double the accuracy. However, the training accuracy was much higher than the test accuracy – 49% compared to 35% - indicating that the model may be overfitting with the value of k. A lower value of *k* was selected to compare the difference. However, lower values would give even a bigger difference between training and test accuracy (54% compared to 33% with *k* = 5). Thus the best-tuned value of k was kept.

Unfortunately, the KNN model is not interpretable as it simply uses the highest vote when using to classify. In this case, there are a lot of ties. A subsample of the predicted probabilities for the KNN model can be found in **Appendix 3.1**. In short, the model does not perform adequately for the triage purpose. It uses a large value for k and is prone to overfitting.

```
Confusion Matrix and Statistics

          Reference
Prediction    B    C    D    E    F    G
         B 1687  389   29    9    8   17
         C 1270  413   41   15   14   15
         D  854  300   52   21   22   21
         E  670  239   34   28   37   40
         F  315  101   23   14   46   81
         G  115   30   12    6  158  426

Overall Statistics

               Accuracy : 0.3512
                 95% CI : (0.3404, 0.3621)
    No Information Rate : 0.6503
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1389

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.3435  0.28057 0.272251 0.301075 0.161404  0.71000
Specificity            0.8289  0.77714 0.834533 0.863252 0.926517  0.95383
Pos Pred Value         0.7887  0.23360 0.040945 0.026718 0.079310  0.57028
Neg Pred Value         0.4044  0.81691 0.977873 0.990006 0.965720  0.97443
Prevalence             0.6503  0.19492 0.025291 0.012315 0.037738  0.07945
Detection Rate         0.2234  0.05469 0.006886 0.003708 0.006091  0.05641
Detection Prevalence   0.2832  0.23411 0.168167 0.138771 0.076801  0.09891
Balanced Accuracy      0.5862  0.52885 0.553392 0.582164 0.543960  0.83191
```

Fig 2.1 – Confusion Matrix for KNN

The figure below shows the variable importance for K-NN classifier. Over 10 iterations, remoteness comes out as the most important variable. The rest of the variables are not as important. The surprise in this model is *Vegetation*. The expectation is that the drier the vegetation is, the easier it is to burn. However, the different types of vegetations seem to have negligible impact in terms of classifying wildfires in the K-NN model. Another surprise inclusion is dstation_m. The variable is simply the distance between where the weather was recorded and the breakout of the fire location, closer locations should theoretically have better predictability of weather and thus fires. However, it does not have a direct link to fires breaking out. Nonetheless, it features highly here. Causes were investigated in the EDA portion earlier on and there appeared to be a direct link between cause and fire class. However, here they do not feature highly.

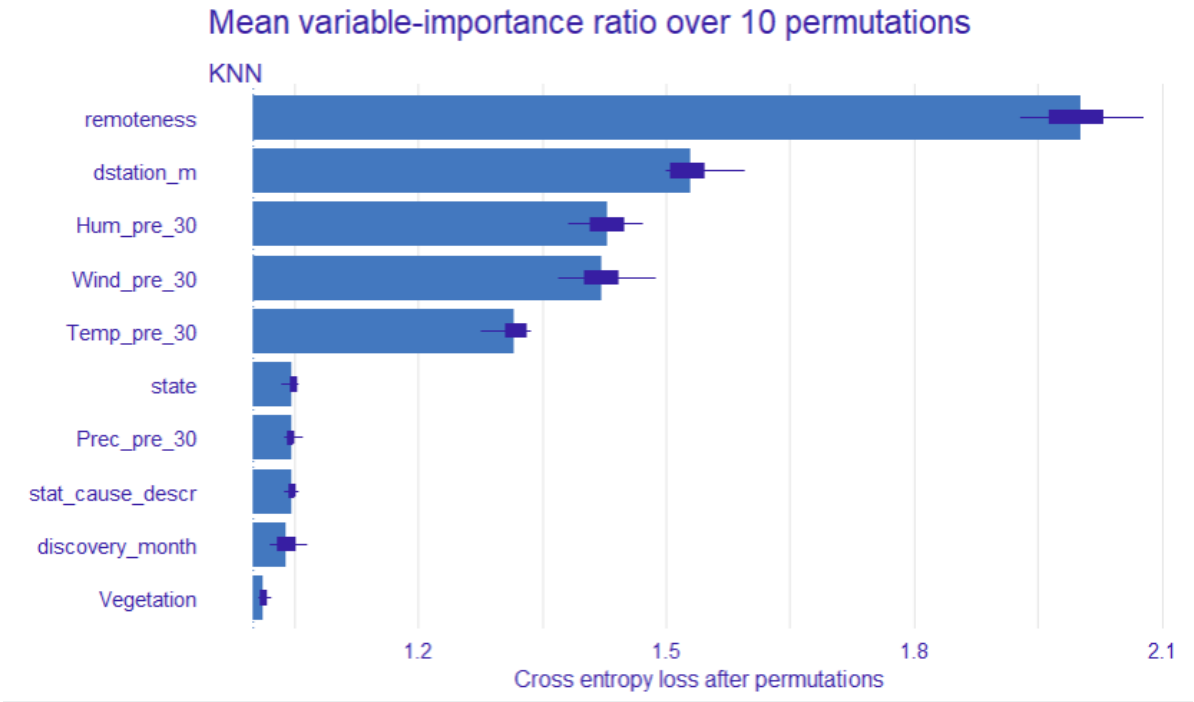


Fig 2.11 – Variable Importance

Neural-Net

The neural net model was similarly trained on a 10-fold CV with the tuning of hyperparameters *size* and *decay*. For this dataset, the *nnet* package was used with only one hidden layer in order to reduce computational complexity. For the *size* hyperparameter, values from *1 to* 10 were considered. For the *decay* parameter, values from *0.1* to *0.*5. The training using CV selected values of 7 for size and 0.4 for decay. The final model was 85-7-6 model i.e. 85 parameters with 7 neurons in the one hidden layer and 6 outputs. The total number of weights selected were 650. The results of the model are summarized in **Fig 2.2.**

```
          Reference
Prediction    B    C    D    E    F    G
         B 2325  503   51   16   13   10
         C 1073  386   27    9    1    3
         D  515  207   29   19    6    7
         E  689  271   55   33   42   36
         F  258   90   23   14   72  109
         G   51   15    6    2  151  435

Overall Statistics

               Accuracy : 0.4343
                 95% CI : (0.4231, 0.4456)
    No Information Rate : 0.6503
    P-Value [Acc > NIR] : 1

                  Kappa : 0.187

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.4734  0.26223  0.15183  0.35484 0.252632  0.72500
Specificity            0.7755  0.81694  0.89757  0.85347 0.932021  0.96764
Pos Pred Value         0.7968  0.25751  0.03704  0.02931 0.127208  0.65909
Neg Pred Value         0.4420  0.82058  0.97607  0.99066 0.969510  0.97606
Prevalence             0.6503  0.19492  0.02529  0.01231 0.037738  0.07945
Detection Rate         0.3079  0.05111  0.00384  0.00437 0.009534  0.05760
Detection Prevalence   0.3864  0.19849  0.10368  0.14910 0.074947  0.08739
Balanced Accuracy      0.6244  0.53958  0.52470  0.60415 0.592327  0.84632
```

Fig 2.2 – Confusion Matrix for NN

The neural net clearly has superior performance to the K-NN model. However, the difference between the two models is not large. Like the K-NN model, the neural net model is adept at predicting Class G. However, it suffers from all the other classes. The Balanced Accuracies are higher for all classes indicating a better sensitivity and specificity fit. Nonetheless, the Cohen's Kappa is still below 0.2 indicating that the model does not outperform a random model by a large amount. Moreover, with 650 weights and 85-7-6 neural net structure, the model has almost no interpretability. The *summary* function lists out all the coefficients and using the sign of the coefficient, an evaluation is possible. However, with the large number of weights and the intrinsic stochastic nature of nnet, interpretability is not possible. The nnet model is not overfitted like the K-NN. The training accuracy is 47% compared to the test accuracy of 43%.

Variable importance for the neural net model will be addressed in the final neural net model.

Random Forest

Neither the neural net nor the K-NN model performed supremely well. Therefore, an ensemble model (Random Forest) was used as the last model evaluated. The random forest

model was similarly trained on a 10-fold CV with the tuning of the hyperparameter *mtry*. The training selected a value of 9. The results of the model are summarized in **Fig 2.3**

```
          Reference
Prediction    B    C    D    E    F    G
         B 2131  427   29   11    3    5
         C 1192  413   38    6    9    9
         D  678  282   48   22   14    9
         E  625  257   48   35   33   36
         F  248   84   24   14   70  120
         G   37    9    4    5  156  421

Overall Statistics

               Accuracy : 0.4129
                 95% CI : (0.4017, 0.4241)
    No Information Rate : 0.6503
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1822

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: B Class: C Class: D Class: E Class: F Class: G
Sensitivity            0.4339  0.28057 0.251309 0.376344 0.245614  0.70167
Specificity            0.8201  0.79375 0.863470 0.866068 0.932572  0.96965
Pos Pred Value         0.8177  0.24775 0.045584 0.033849 0.125000  0.66614
Neg Pred Value         0.4379  0.82005 0.977997 0.991102 0.969251  0.97413
Prevalence             0.6503  0.19492 0.025291 0.012315 0.037738  0.07945
Detection Rate         0.2822  0.05469 0.006356 0.004635 0.009269  0.05575
Detection Prevalence   0.3451  0.22074 0.139433 0.136917 0.074153  0.08369
Balanced Accuracy      0.6270  0.53716 0.557389 0.621206 0.589093  0.83566
```

Fig 2.3 – Confusion Matrix for Random Forest

The random forest model shows a similar performance to the NN model. However, the NN model still proves superior. The overall results are in a similar ballpark to the NN model and the K-NN model with the same classes showing the same level of predictability.

The advantage of the randomforest model is that it allows for interpretability. An explainer was created using the final tuned randomforest model and **Fig 2.4** and **Fig 2.5** show the output of the two attempts at variable importance.

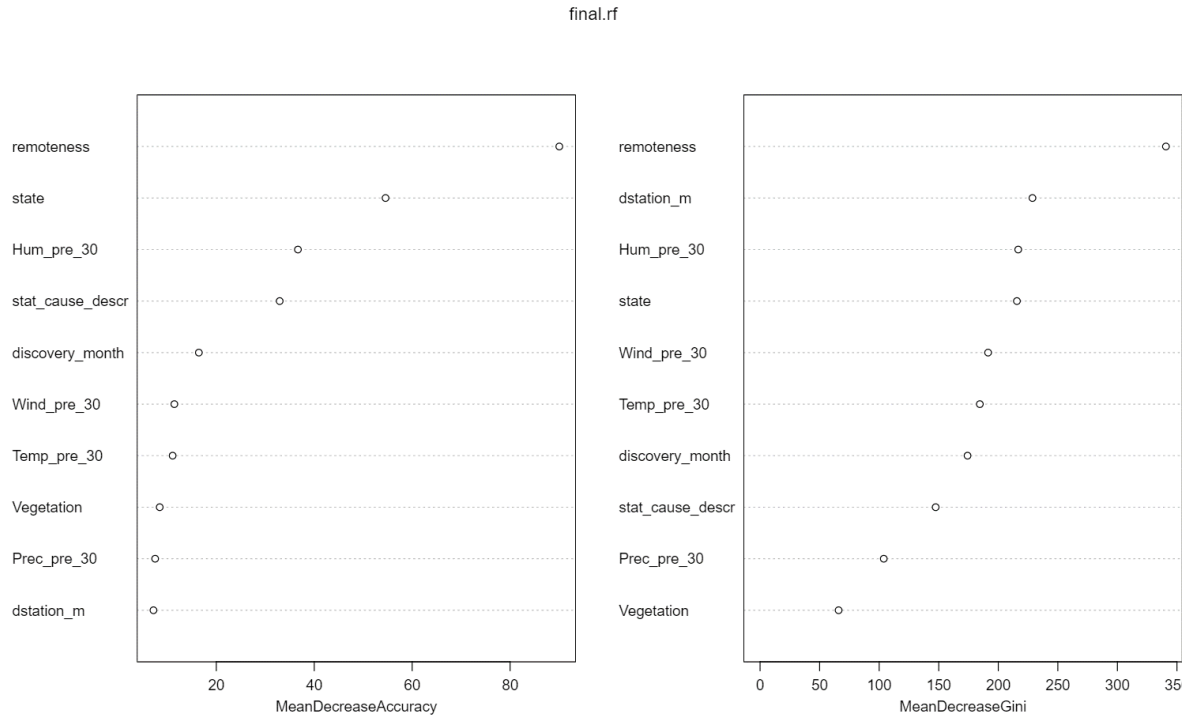


Fig 2.4 – Variable Importance for RandomForest

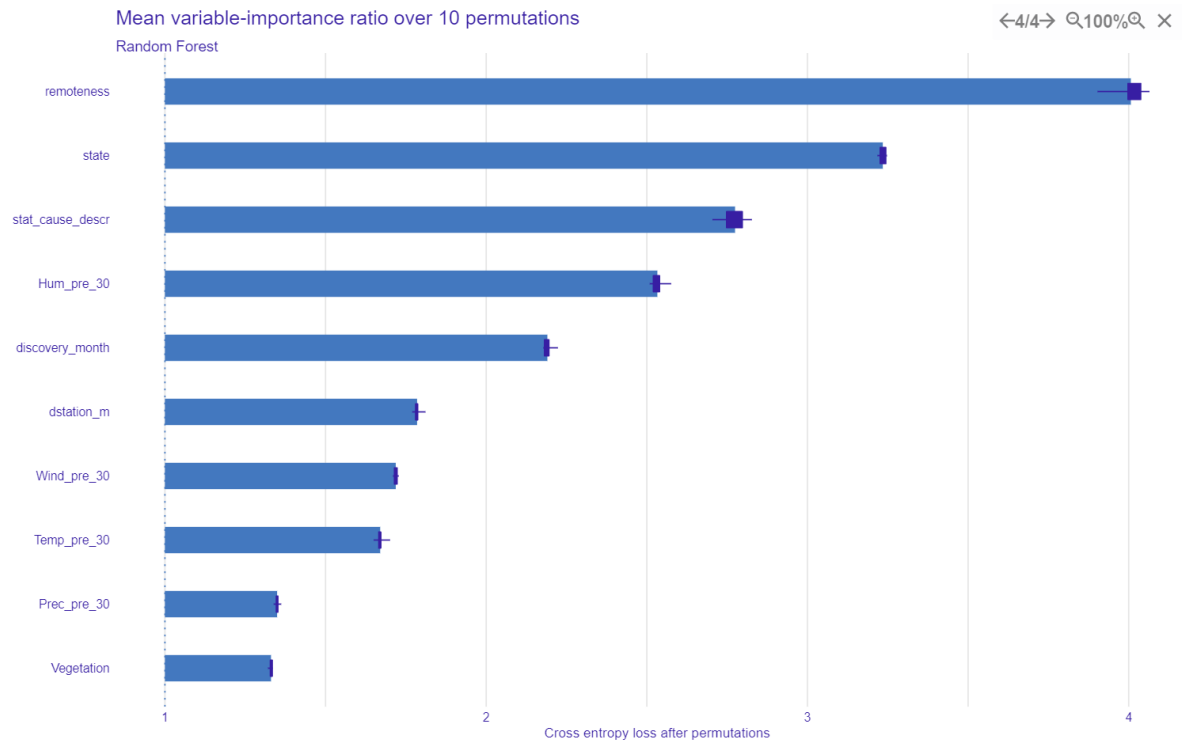


Fig 2.5 – Variable Importance for RandomForest

As we saw in the EDA, remoteness seemed correlated with fire size and that result is mimicked here. For the *randomforest* model, remoteness is one of the most important variables alongside state, humidity and cause. Unsurprisingly, the distance of the weather station from the fire is not relevant for the classification. What is more surprising is that variables like wind, temperature and vegetation have a much lower impact on fire class while

humidity has a very high importance. Intuitively, a windy environment should cause wildfires to spread more quickly and the greater the heat, the greater the fire. However, this hypothesis is not borne out by the results. The result for humidity makes sense as high humidity results in more water vapour in the air making it difficult for wood to catch fire easily.

Dimension Reduction Trained Neural-Net

During the data-wrangling, a lot of variables regarding weather were removed as they were closely correlated to each other. However, the performance of the previous models shows that there are some additional features necessary to make the models operate better. It is possible that by removing some of the earlier features, some important signal within the data was lost. The image below shows the correlations between all of the features. It is clear that some are highly correlated so should not be used together.

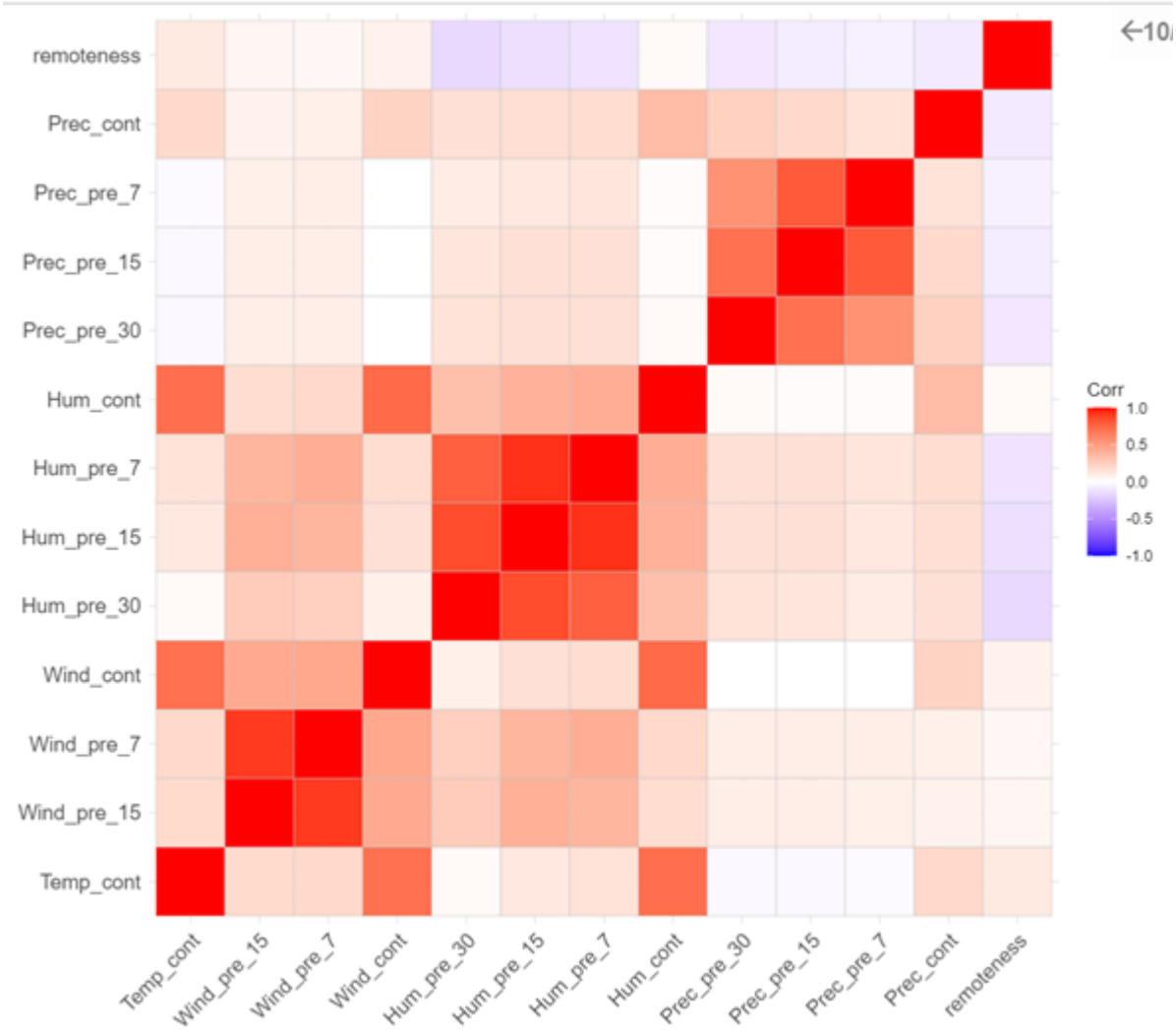


Fig 2.6 – Correlation Matrix

Therefore, the next step is to re-add all the removed features. This will increase the number of features to 23. However, in order to fit the best model, PCA will be utilized in order to take care of the dimensionality problem. The features will be projected into smaller dimensions and the new features from PCA that explain at least 75% of the variation in the data will be utilized. The numeric features were scaled and used to create a PCA plot.

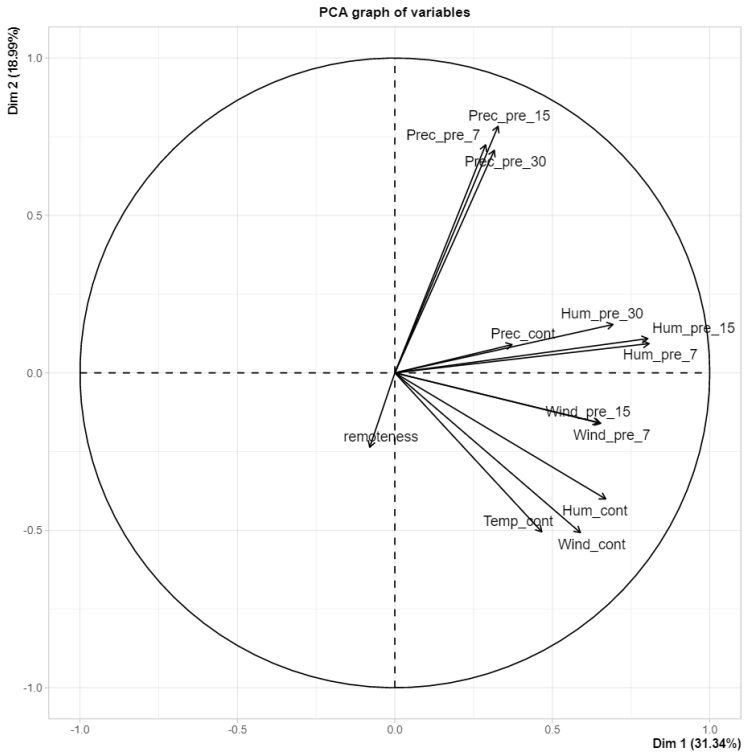


Fig 2.7 – PCA

From the plot, we can see that all precipitation features are positively correlated with the second dimension and to a large degree. The average precipitation in the last 15 days (*prec_pre_15*) displays the strongest correlation. Curiously, *prec_cont* i.e. precipitation up until the day of the fire being put-out is correlated more with Dimension 1 rather than Dimension 2, unlike the rest of the precipitation features. Meanwhile, humidity is positively and strongly correlated with Dimesnion1. The only temperature metric that shows up here is *temp_cont*. It is negatively correlated with Dimension 2 and positively with Dimension 1. From the variable importance analysis of *Random Forest*, we saw that remoteness held important predictive power. However, remoteness is not strongly correlated with either dimension in the PCA. A biplot with classes can be found in **Appendix 3.2**.

A summary of the cumulative variance explained by the features can be seen below:

Eigenvalues

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7	Dim.8	Dim.9	Dim.10
Variance	4.074	2.469	1.910	1.474	0.938	0.771	0.437	0.270	0.213	0.166
% of var.	31.338	18.992	14.689	11.339	7.217	5.933	3.359	2.078	1.636	1.279
Cumulative % of var.	31.338	50.330	65.020	76.358	83.575	89.508	92.867	94.946	96.582	97.861

The first four features explain 76% of the variability in the data. Therefore, these features will be extracted, bound together with *fire_size_class* and the categorical variables and then predicted. Since the NN model was the best performer in the last round, only that model will be trained and the accuracy compared with the earlier models.

The same process was repeated to select the hyperparameter *size* and *decay*. The training selected a best tune of size = 3 and decay = 0.3. The results of the final dimensionally reduced model are below in **Fig 2.8.** Unfortunately, despite reducing the dimensionality of

the model, it still did not perform better than previously expected. The Accuracy of 43% is not much different from the earlier accuracy of the full model which had a similar accuracy.

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5    6
         1 2407  479   42   13   16   31
         2  967  393   39   15    3    5
         3  522  236   39   15   24   17
         4  546  176   37   21   51   79
         5  249   88   18   12   60  125
         6  220  100   16   17  131  343

Overall Statistics

               Accuracy : 0.4321
                 95% CI : (0.4209, 0.4433)
    No Information Rate : 0.6503
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1772

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
Sensitivity            0.4901  0.26698 0.204188 0.225806 0.210526  0.57167
Specificity            0.7800  0.83076 0.889417 0.880815 0.932297  0.93038
Pos Pred Value         0.8056  0.27637 0.045721 0.023077 0.108696  0.41475
Neg Pred Value         0.4514  0.82398 0.977310 0.989160 0.967857  0.96178
Prevalence             0.6503  0.19492 0.025291 0.012315 0.037738  0.07945
Detection Rate         0.3187  0.05204 0.005164 0.002781 0.007945  0.04542
Detection Prevalence   0.3957  0.18829 0.112950 0.120498 0.073093  0.10951
Balanced Accuracy      0.6351  0.54887 0.546803 0.553311 0.571411  0.75102
```

Fig 2.8 – Dimension Reduction Neural Net

Reduction of Classes

All models have thus far failed to provide a substantial benefit over a random model. However, an analysis of the average fire size between classes reveals that there is not much of a difference between Classes B and C, Classes D and E, and Classes G and F. This analysis is shown in **Fig 2.9**

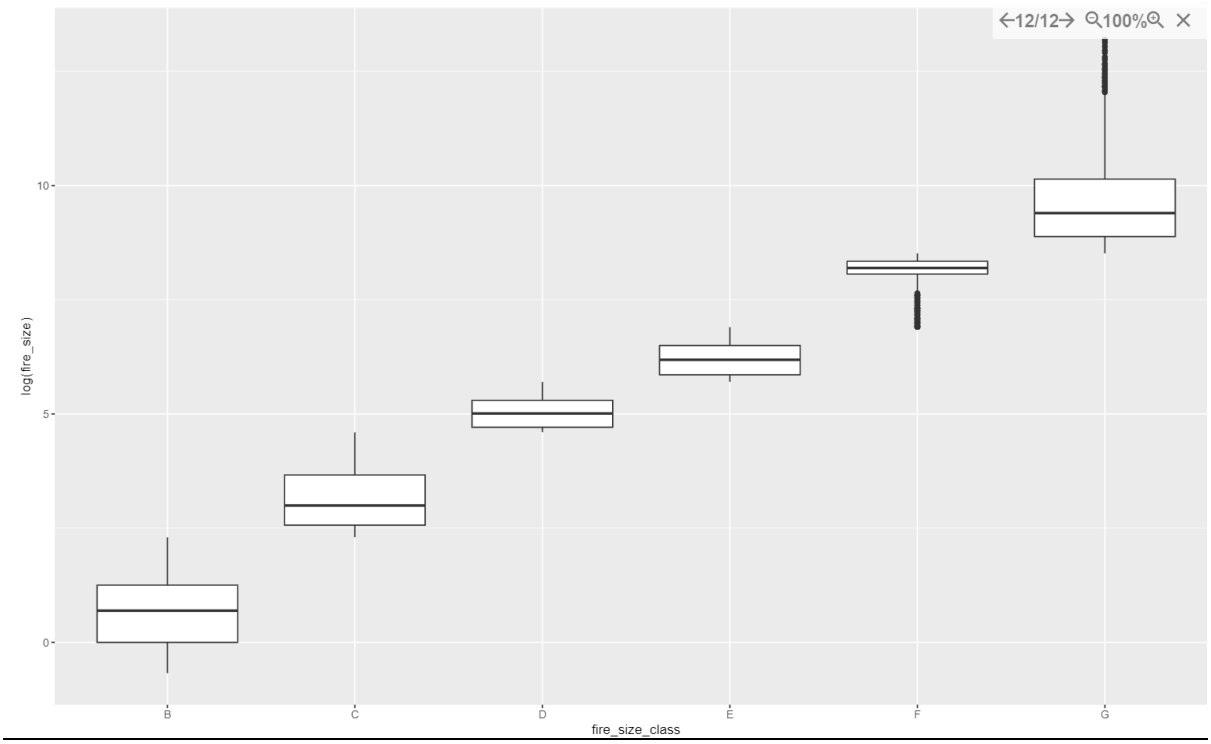


Fig 2.9 – Comparison of Fire Classes and Fire Size

As seen above, the fire classes can be divided into three subgroups:

1. Log(Fire Size) < 5
2. 5 < Log(Fire_Size) < 7.5
3. Log(Fire_Size) > 7.5

Therefore, it may be useful to combine these classes together and see if the predictability of the model increases. The classes were refactorized into three new levels according to the rule of thumb aforementioned:

- Class BC (combined class of B & C)
- Class DE (combined class of D & E)
- Class FG (combined class of F & G)

As these new classes were created, the entire subsampling process must be repeated. The same steps were followed in order to make sure that the three classes were balanced.

Lastly, as the best performer thus far has been Neural Nets, that is the only model that will be taken into consideration.

Final Neural Net Model

This model was trained in the same as the earlier neural net model. The hyperparameters were once again trained on a 10-fold CV. The training selected hyperparameters of *size* = 4 and *decay* = 0.4. The model is a 96-5-3 network with 503 weights; this is not too dissimilar from the non-reduced neural net model. The results of the model can be viewed in **Table 2.5**.

This model far outperforms every other model thus far with an accuracy of approximately 70%. However, this model is prone to overfitting as the training accuracy is 75% compared to

the test accuracy of 70%. Nonetheless, these values are close enough that the results can be considered valid.

```
          Reference
Prediction   BC   DE   FG
        BC 4245   86   20
        DE 2061  183   74
        FG   77   15  791

Overall Statistics

               Accuracy : 0.6911
                 95% CI : (0.6805, 0.7015)
    No Information Rate : 0.8452
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3667

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: BC Class: DE Class: FG
Sensitivity             0.6650   0.64437    0.8938
Specificity             0.9093   0.70625    0.9862
Pos Pred Value          0.9756   0.07895    0.8958
Neg Pred Value          0.3321   0.98070    0.9859
Prevalence              0.8452   0.03761    0.1172
Detection Rate          0.5621   0.02423    0.1047
Detection Prevalence    0.5761   0.30694    0.1169
Balanced Accuracy       0.7872   0.67531    0.9400
```

Fig 2.10 – 3 Classes NN

Unfortunately, as mentioned prior, due to the complexity of the neural net structure and the number of weights, the model is not interpretable to a large degree. However, the encouraging signs for this model are that its balanced accuracy for the class FG i.e. the classes that cause the most damage is 0.94. Thus, the most damaging fires can be predicted with incredible accuracy using this model.

The DALEX:explain package was used to determine variable importance for this final neural net model based on three classes. The variable importance is depicted in **Fig 2.11**

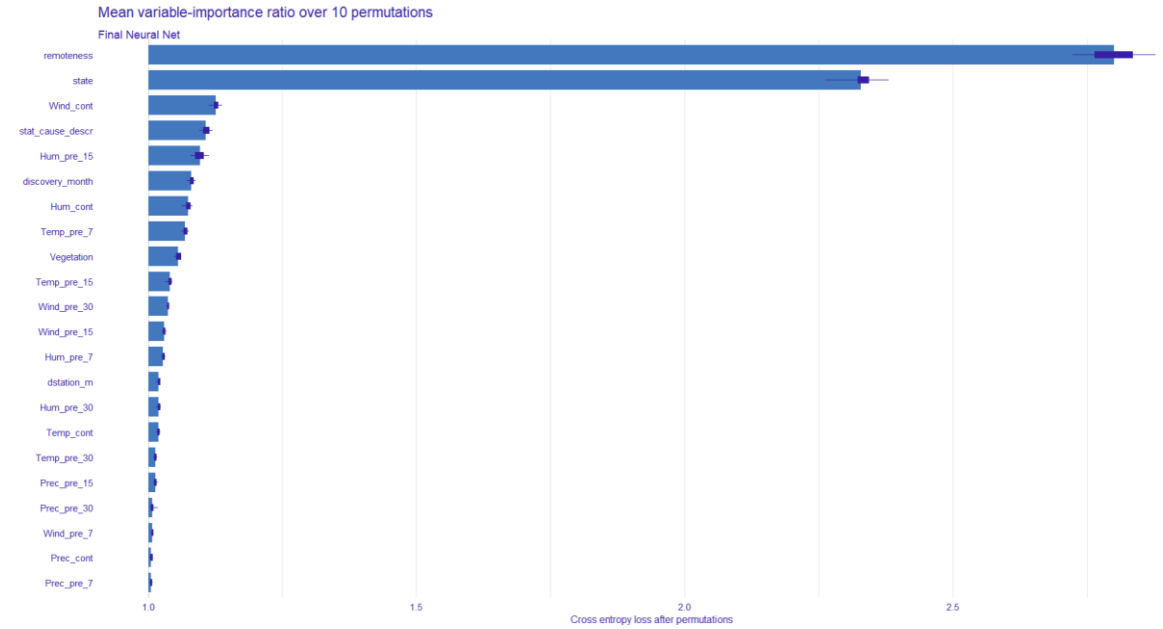


Fig 2.11 – Variable Importance for 3 Class NN

Like the *RandomForest* model, the neuralnet model also has overwhelming reliance on remoteness and state. However, the rest of the variables are not as important for this particular model. Whereas in the *RandomForest* model, humidity and cause of fire featured more extensively, they are limited in the neural net model. It comes as a surprise that most of the temperature metrics (especially precipitation) are not important. Research has shown that droughts are one of the best predictors of whether a fire will break out but they do not feature here much. In fact, precipitation variables are some of the least important variables in the model.

Conclusion

This report has detailed several machine learning and statistical techniques in order to derive a model that can predict wild fire classes based on several different factors. As the previous sections have shown, machine learning techniques can be used in conjunction with easily available data to predict wildfire classes. Weather data, vegetation and geospatial coordinates are easily accessible from the internet and can be accumulated by fire lookouts in order to train the machine learning models. These models can then provide up-to-date and instant feedback on whether a fire has the potential to be a damaging one. Using these techniques, the US Forest Service can triage the most relevant fires to address, especially during the summer months when multiple fires break out at the same time.

The analysis has shown that Neural Nets and Random Forests far outperform K-NN in this multi-class problem and should be considered during future research into wildfire triage. Recent research into wildfires prediction systems using Machine Learning has shown that RF, SVM and ANN models are commonly used – RF and SVM for their interpretability and Artificial Neural Networks for their greater predictive power. However, as indicated in the report, neural networks while being the better performers are complex and computationally expensive compared to RF models. They are also not as interpretable. In terms of variables, all models utilized showed that remoteness would be one of the most important factors in terms of what class a fire can achieve. This make intuitive sense as the further away a fire

location is from dense populous areas, the more difficult it is to divert resources towards extinguishing the fire. Moreover, more remote areas tend to have more vegetation which burns much faster than human-built infrastructure like roads which naturally acts as a breakpoint between fires spreading. States and fire cause also featured as important variables in well-performing models. However, precipitation was a surprise exclusion in terms of variable importance as it did not provide predictive power to the models. The exact relationship between precipitation, rather than drought, can be investigated in future research.

The purpose of this report was to test the hypothesis whether machine learning techniques can be used by the US Forest Service as a sort of triage mechanism when multiple fires break out at the same time. Whilst the accuracy of the basic models was not high, the final model displayed an accuracy of 70% and a Cohen's Kappa of 0.36 indicating greater predictability. However, the use case of the model is even more optimistic as the US Fire Service is probably more concerned with identifying and extinguishing potentially larger fires. The model is able to classify large fire classes with far greater accuracy than small fire classes. In this particular case, sensitivity is the most important thing as the desired model should be able to identify most positive cases as it's dangerous to let large fires develop, but it is not as damaging to not correctly identify small scale fires as they do not possess the same type of damage concerns. For "Class FG", the model was able to predict the fires with a 89% sensitivity rate. This is cause for optimism.

Limitations and Future Research

There are a number of key limitations in the model that need to be addressed. Firstly, the dataset is a subsample of a larger 1.8Million fire dataset. There is a possibility that it might not be representative enough as it only covers 2% of the population. As we have seen, certain months are overrepresented in terms of number of fires.

By the results of the first three models, we can note that weather, vegetation and remoteness alone is not enough to successfully classify a fire into the six classes. There needs to be more data such as droughts over a longer period of time, past fires in the area, the population density et al that should be combined in order to have much better predictive capability. Simply weather data is insufficient in order to classify into six classes as there is a wide variety of fire sizes that have occurred at different levels. Moreover, climate change means wildfires are becoming more common and historical weather data may not be predictive and new variables need to be considered.

Secondly, computational complexity meant that other relevant models such as the radial SVM model could not be explored. It is possible that those models would perform superior to the models used in this report. The issue of subsampling the data because of unbalanced classes also means that the model does not make use of the full extent of the data in order to successfully classify the wildfires.

Lastly, a major obstacle for the adoption of ML models in wildfire prediction is the lack of interpretability as they are considered black-box models. Future research should develop models that have a greater interpretability in order to be adopted by the community. In closing, wildfire science is a diverse and multi-faceted discipline that requires a multipronged approach in order to make classifications accurately. The fact that this environment is changing rapidly means that the techniques utilized need to evolve and improve just as rapidly as well.

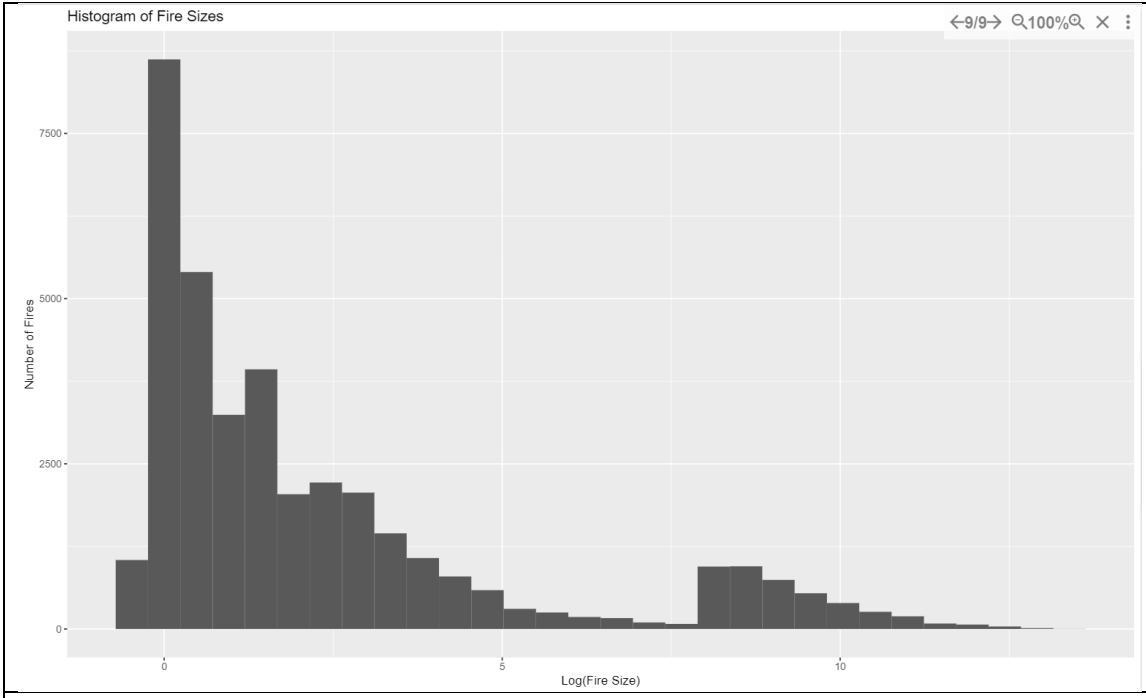
Appendices

Appendix 1

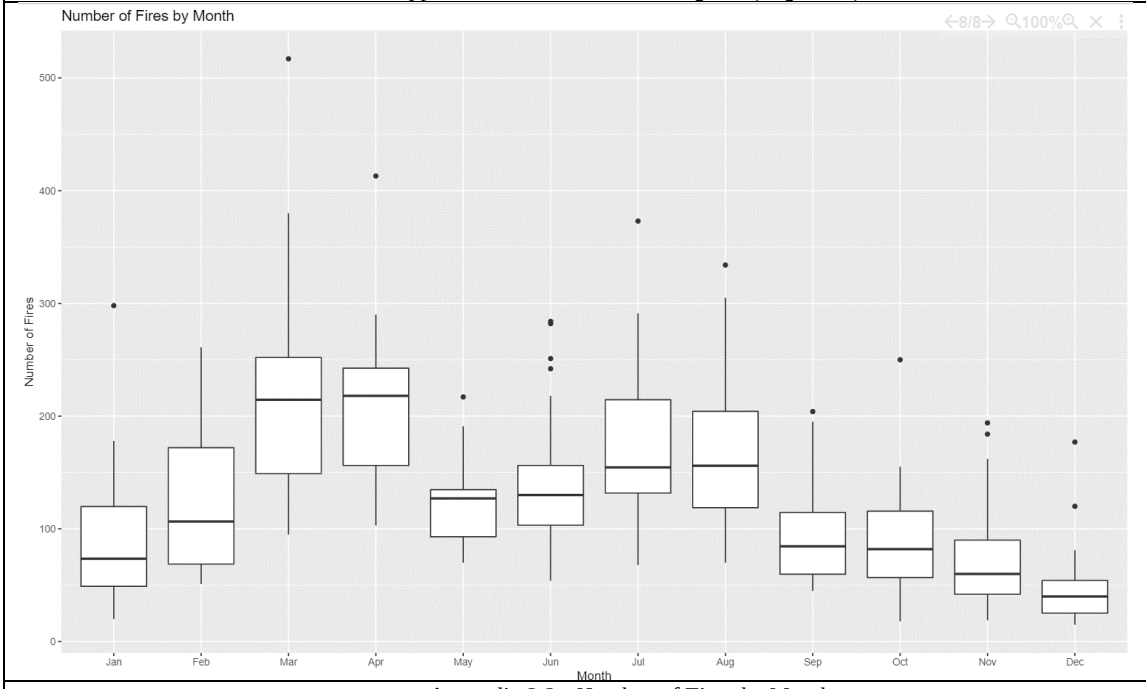
Data Source: Kaggle Data Source

Column Name	Description
Fire_name	Notable Fires are Named. Most are Emtpy
Fire_size	Size of fire in acres
Fire_size_class	Class of fire
Stat_cause_descr	Cause of fire
Latitude	Latitude of Fire Location
Longitude	Longitude of Fire Location
State	State in which fire broke out
Discovery_month	Month in which fire was discovered
Putout_time	Time taken to put out fire
Disc_pre_year	Year before discovery date
Vegetation	Most dominant vegetation type (1-14)
Fire_mag	Magnitude of fire (based on class fire)
Temp_pre_30	Temperature in C in the previous 30 days
Temp_pre_15	Temperature in C in the previous 15 days
Temp_pre_7	Temperature in C in the previous 7 days
Temp_cont	Temperature in C at the location of fire upto day the fire was contained
Wind_pre_30	Wind in m/s in the previous 30 days (average)
Wind_pre_15	Wind in m/s in the previous 15 days (average)
Wind_pre_7	Wind in m/s in the previous 7 days (average)
Wind_cont	Wind in m/s up to the day the fire was contained
Hum_pre_30	Humidity in % in the previous 30 days (average)
Hum_pre_15	Humidity in % in the previous 15 days (average)
Hum_pre_7	Humidity in % in the previous 7 days (average)
Hum_cont	Humidity in % up to the day the fire was contained
Prec_pre_30	Precipitation in mm in the previous 30 days (average)
Prec_pre_15	Precipitation in mm in the previous 15 days (average)
Prec_pre_7	Precipitation in mm in the previous 7 days (average)
Prec_cont	Precipitation in mm up to the day the fire was contained
Remoteness	Non-dimensional distance to closest city
Weather_file	Name of file used for whether scraping
Wstation_eyear	Last functional year of weather station
Wstation_byear	Year weather station was built
Wstation_wban	Global identifier of weather station
Dstation_m	Distance of the weather station to the fire
Wstation_usaf	US identifier of weather station
Disc_pre_month	30 days before fire was discovered (for weather scraping)
Disc_pre_year	Year before fire was discovered (for weather scraping)
Disc_pre_date	7 days before fire was discovered (for weather scraping)
Cont_date_final	(for weather scraping)
Disc_date_final	for weather scraping)
Cont_date_clean	for weather scraping)
Disc_date_clean	for weather scraping)

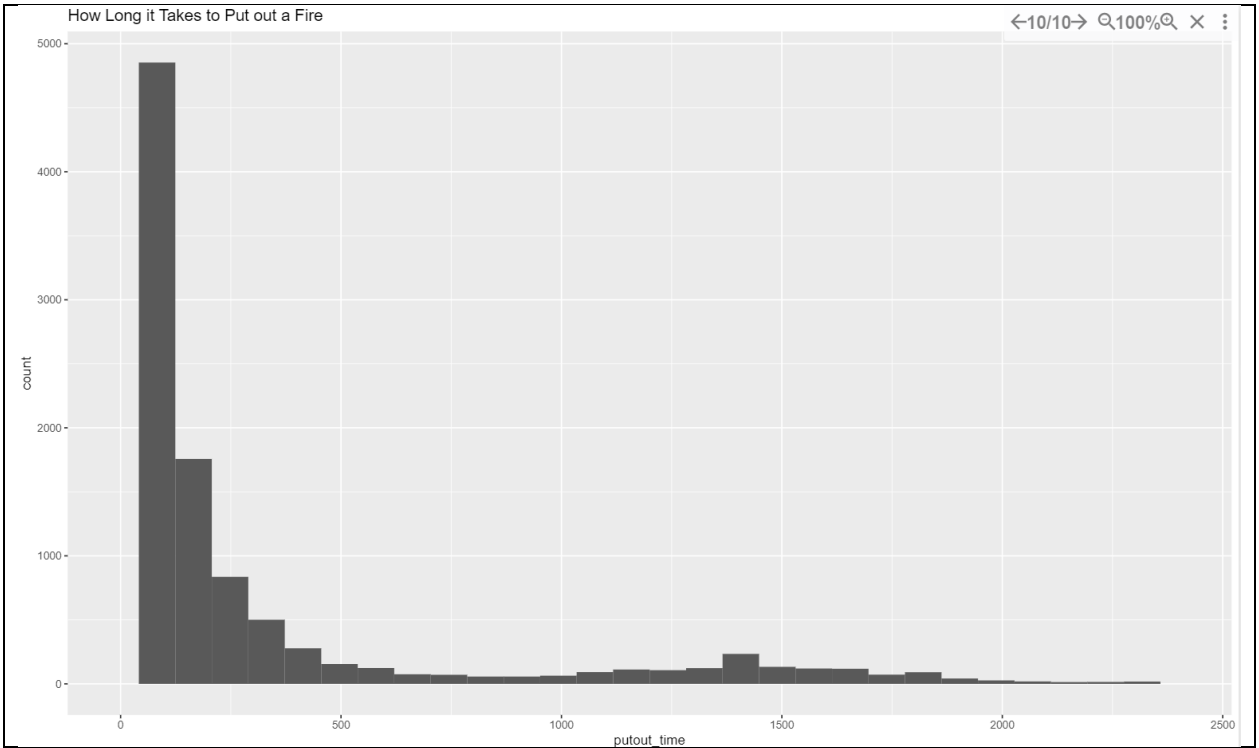
Appendix 2



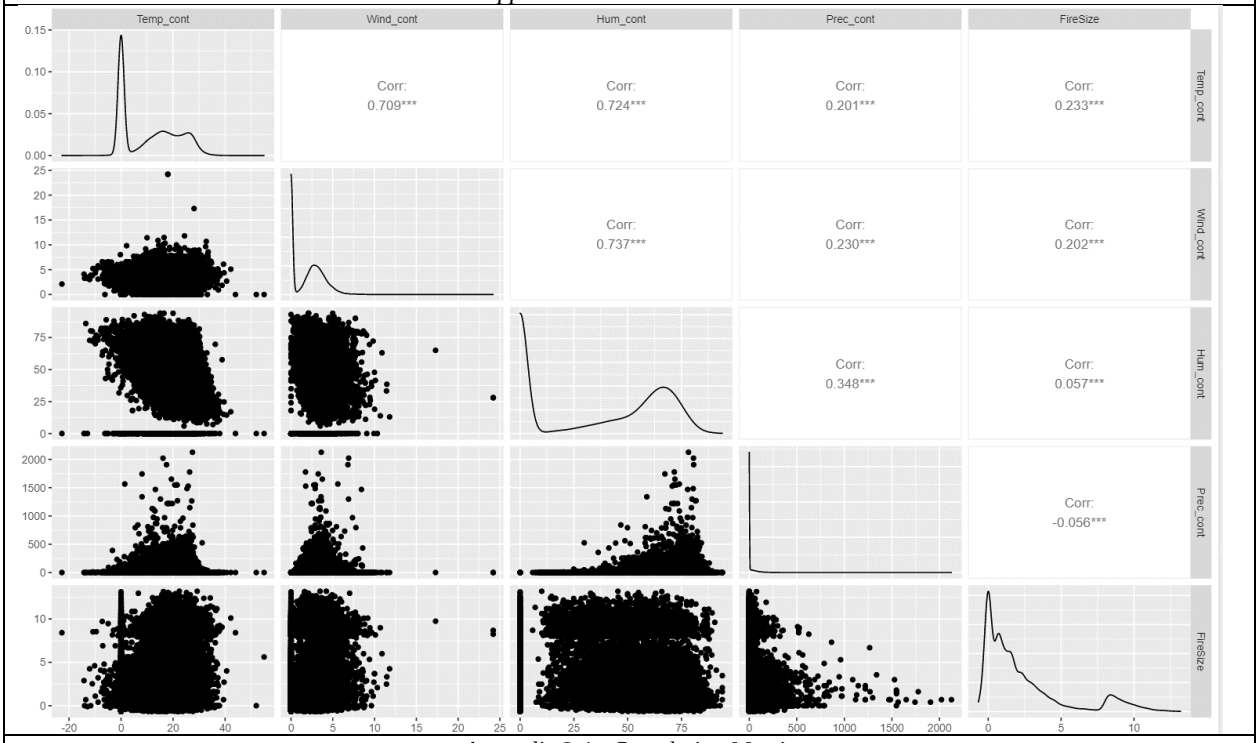
Appendix 2.1 - Fire Sizes Histogram (Log Scale)



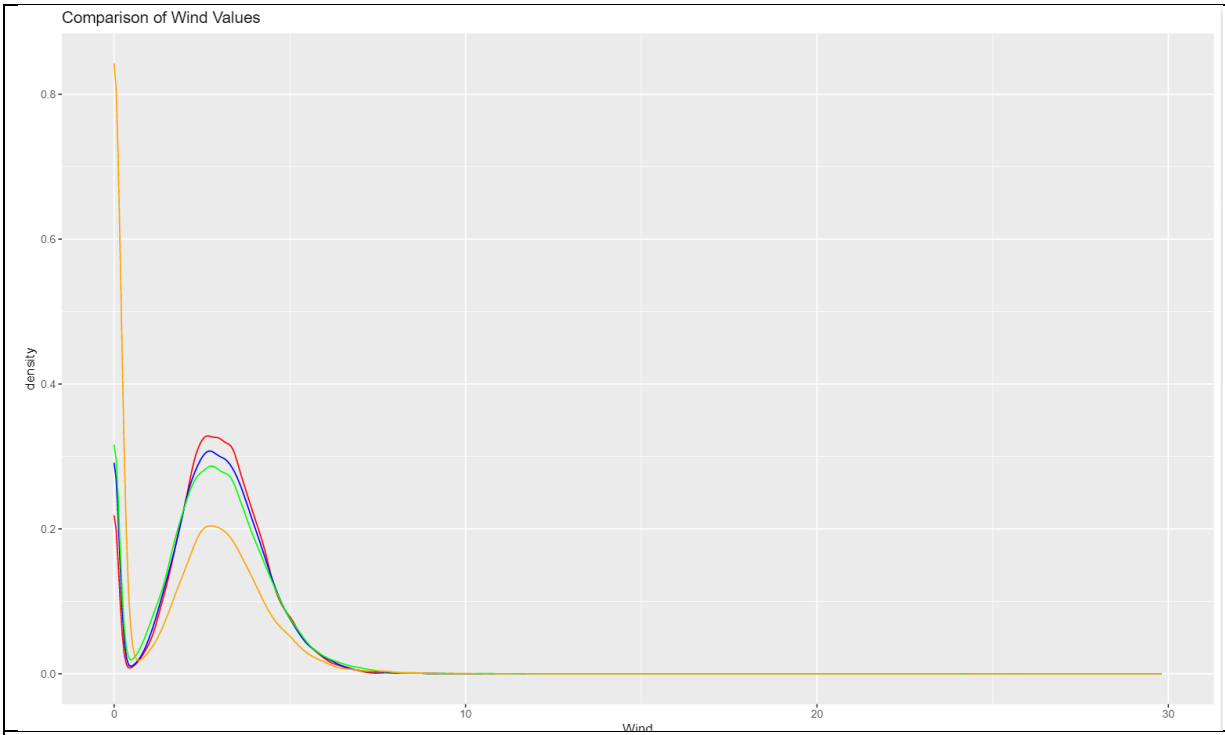
Appendix 2.2 - Number of Fires by Month



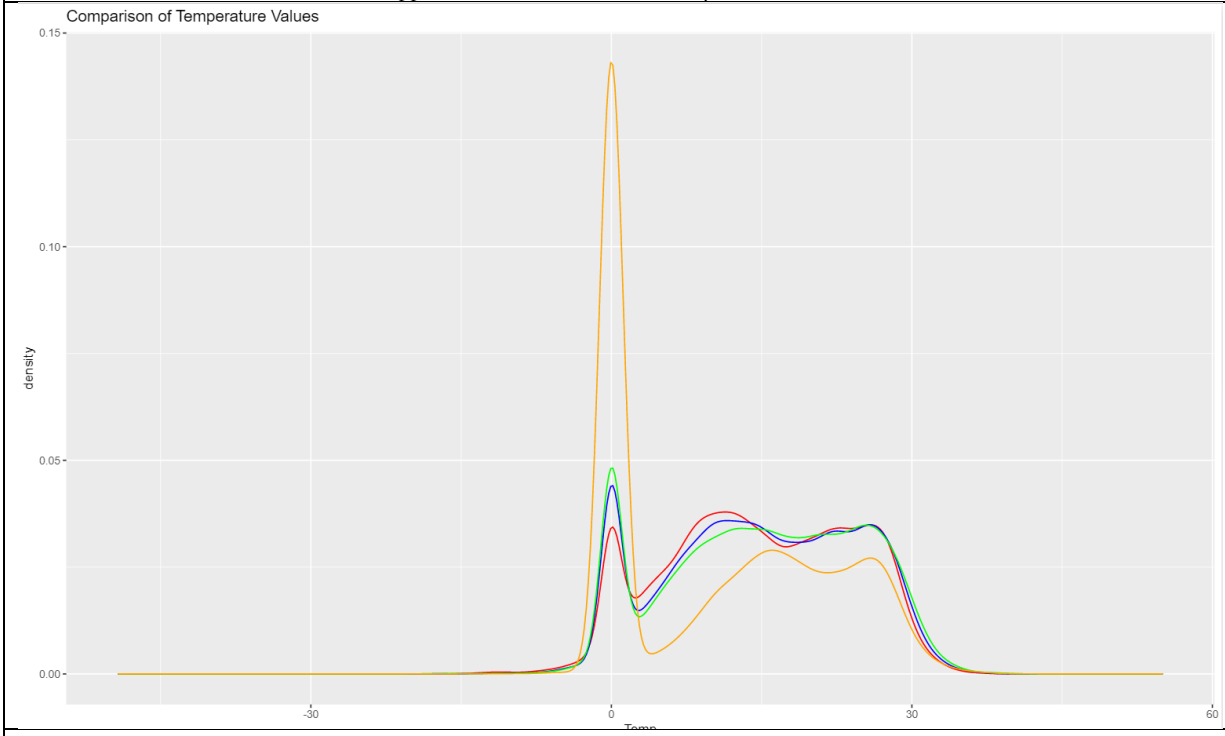
Appendix 2.3 - APut-Out Time



Appendix 2.4 - Correlation Matrix



Appendix 2.5 - Relation between four wind metrics

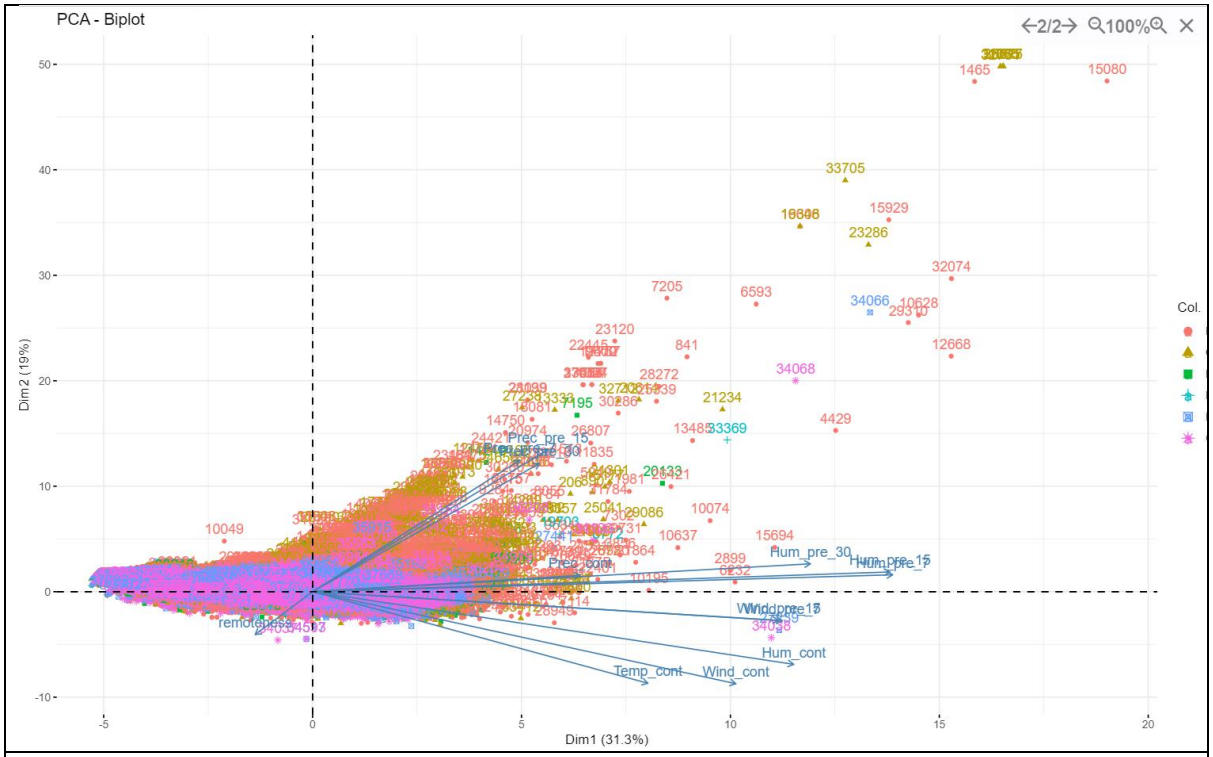


Appendix 2.6 - Relation between four temperature metrics

Appendix 3

B	C	D	E	F	G
0.625	0.25	0.125	0	0	0
0.375	0.625	0	0	0	0
0.125	0.375	0.25	0.25	0	0
0.125	0.5	0.25	0.125	0	0
0.25	0.5	0	0.25	0	0
0.25	0.125	0.125	0.5	0	0
0.25	0.625	0.125	0	0	0
0.25	0.25	0.25	0.25	0	0
0.375	0.125	0.5	0	0	0
0.25	0.25	0.125	0.25	0.125	0
0.125	0.25	0.375	0.125	0.125	0
0.125	0.25	0	0.125	0.375	0.125
0.5	0.25	0.25	0	0	0
0	0.375	0.25	0.375	0	0
0	0.25	0.375	0.25	0.125	0
0.5	0	0.25	0.125	0.125	0
0.125	0.375	0.25	0.25	0	0
0.25	0.25	0	0	0.25	0.25
0.125	0.375	0.375	0.125	0	0
0.625	0	0.375	0	0	0
0.625	0.125	0.125	0.125	0	0
0.375	0.375	0.125	0.125	0	0

Appendix 3.1 – Probabilities of KNN Model



Appendix 3.2 – Biplot of Dimension 1 and 2 with Classes

References

[1] Center for Climate Change and Energy Solutions: https://www.c2es.org/content/wildfires-and-climate-change/#:~:text=The%202018%20wildfire%20season%20went,damages%20in%20California%20and%20Alaska.

[2] United States Environmental Protection Agency. https://www.epa.gov/climate-indicators/climate-change-indicators-wildfires

[3] Disaster Philanthropy. https://disasterphilanthropy.org/disasters/2021-north-american-wildfire-season/#:~:text=United%20States,-As%20of%20Dec&text=31%2C%20the%20National%20Interagency%20Fire,fires%20burning%207.47%20million%20acres.

[4] National Park Service https://www.nps.gov/articles/wildfire-causes-and-evaluation.htm