MACHINE LEARNING FOR VIRGINIA FINAL REPORT: VIRGINIA WILDFIRES SIZE PREDICATION

Mike Cai (mc6gb), Yunlu Li (yl4df), Yilin Huang (yh3mw)

1 Abstract

We consider using machine learning algorithm to predict fire size in Virginia using multiple features to improve wildfire control and prevention in Virginia. We use a data set that contains the information of wildfires that occurred in the United States from 1992 to 2015 and extract wildfires data of Virginia from it. Our approach is to use classification on selected features to predict the fire size since some features are irrelevant. The most relevant feature in our experiment is $DURATION$, which indicates the duration of a wildfire incident and were created by subtracting discovery time from contained time. We chose Polynomial SVM and RBF SVM because SVM works well when there are a clear separation between classes. We also tried using Decision Tree, Random Forest as well as Artificial Neural Network. So far, the best result we have is from Random Forest, which has the accuracy of 0.6480.

2 Introduction

Wildfires have always been a huge issue across United States. Incidents of wildfires destroying houses often appear on news. With climate change rapidly increasing the danger and scales of wildfires, Virginia, a state that is 65% covered by forest and sometimes accompanied with high wind, is definitely vulnerable. Due to extended periods of below average rainfall, dry weather conditions, and record-high temperatures across our commonwealth, most of Virginia faces an increased risk of wildfires. During fire incidents, decisions regarding how to effectively deploy resources are made with limited time and high significance. In this project, we will apply machine learning algorithms on a data set of wildfire incidents to help predict the seriousness of potential wildfires that will take place at a specific time and location in Virginia in the future. The government and related departments can use the results of this project to better distribute wildfire prevention resources.

3 Method

The original data set is about 700 MB in SQLite form. For this project, we focused on the fires occurred in the state of Virginia, so we only kept the records about Virginia by SQL filtering queries and exported the csv file for as the main data set of this project. The filtered csv file includes 38 features. Since we wanted to predict wildfire size, some features were not relevant, such as name of the reporting agency and number or code that uniquely identifies an incident report. After that, we further cleaned up our data by removing records containing null values. Since the most of features are categorical data, then we applied Label Encoder to encode categorical features and also used Standard Scaler to transform the numerical data.

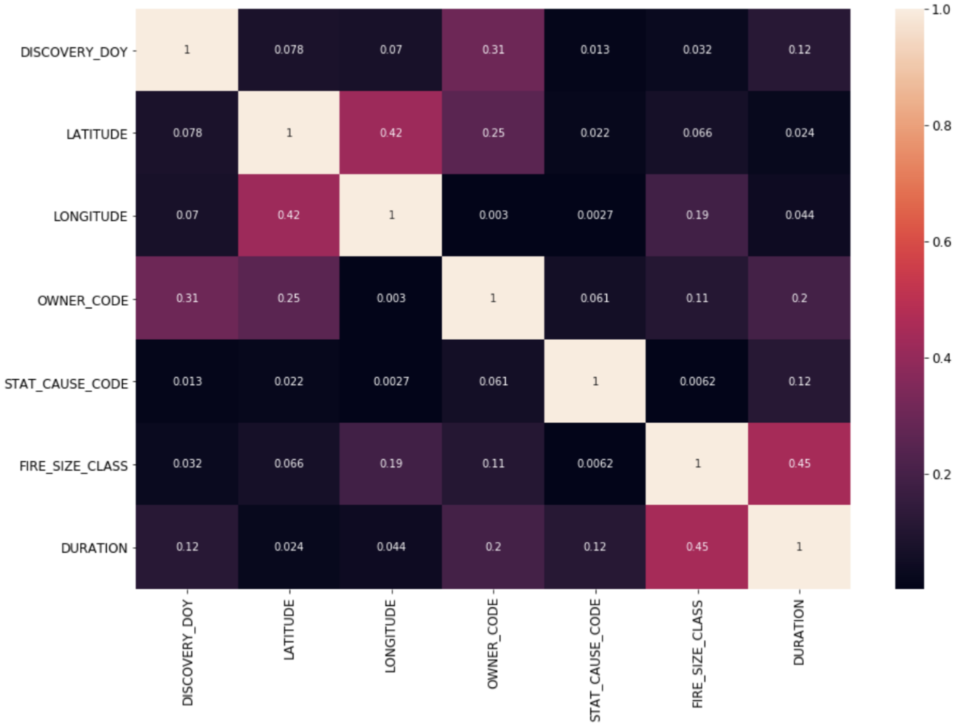
We first extracted the features we want, which are,

1. $DISCOVERY\_DOY$ (day of year on which the fire was discovered or confirmed to exist)
2. $CONT\_DOY$ (day of year on which the fire was declared contained or otherwise controlled)
3. $LATITUDE$
4. $LONGITUDE$
5. $OWNER\_CODE$ (Name of primary owner or entity responsible for managing the land at the point of origin of the fire at the time of the incident)
6. $STATA\_CAUSE\_CODE$ (Code for the (statistical) cause of the fire)

7. $FIRE\_SIZE\_CLASS$ (Code for fire size based on the number of acres within the final fire perimeter expenditures (A=greater than 0 but less than or equal to 0.25 acres, B=0.26-9.9 acres, C=10.0-99.9 acres, D=100-299 acres, E=300 to 999 acres, F=1000 to 4999 acres, and G=5000+ acres)).

After looking at the correlation of the extracted features, see figure 2, we then created a new feature to predict wildfire size. We named the new feature $DURATION$ (unit is in days), which we got from $CONT\_DOY$ minus $DISCOVERY\_DOY$, representing the duration of the wildfire from discovered to controlled. We will use the features above to predict wildfire's size so we are dealing with a classification problem. $FIRE\_SIZE\_CLASS$ from the data set will be treated as our label.

Figure 1:



The current models we are using to train our data are the Polynomial SVM, RBF SVM, Decision Trees, Random Forest and Artificial Neural Network from the standard library. We did not use the Linear SVM because such simple model will have a high probability of under fitting our data set. We chose Polynomial SVM and RBF SVM because SVM works relatively well when there is a clear margin of separation between classes, which we do in this case since the samples are already grouped by fire size ($FIRE\_SIZE\_CLASS$). SVM cannot be used for large data sets because SVM needs to solve the quadratic programming problem (QP) in order to find a separation hyperplane, which causes an intensive computational complexity for large data sets. For the wildfire data set, we only have around 3000 samples so we are allowed to use SVM.

We also used Decision Trees because it can compared to other algorithms it requires less effort for data preparation during pre-processing, being able to handle both numerical and categorical data and does not require scaling of the data. We then wanted to try using random forest model because using a decision tree model on a given training data set the accuracy keeps improving with more and more splits but can easily lead to overfitting. For the Random Forest model, accuracy keeps increasing as you increase the number of trees, but becomes constant at certain point. Unlike decision tree, it won't create highly biased model and reduces the variance.

We also intend to use artificial neural network as our model for classification because ANN has the ability to learn and model non linear and complex relationships, which is suitable for us categorizing the 6 categories. The second reason is that artificial neural network can generalize. After learning from the initial inputs, it can infer unseen relationships on unseen data as well. The last reason is that ANN does not impose any restrictions on the input variables, performing well on modeling heteroskedasticity, having the ability to learn hidden relationships in the data.

4 Experiments & Results

Jupyter Notebook: `https://colab.research.google.com/drive/1_x6SzwzRstpXGiycn5-BVxkMipo1Napq#scrollTo=LLgZFQaKUizZ`

The first model that we tried is the Support Vector Machine with Gaussian RBF(Radial Basis Function) kernel. In the pipeline that we designed, we included Standard Scaler and One Versus Rest Classifier. We applied Standard Scaler because the SVM model is very sensitive to un-scaled data; we employed OVR strategy because we have 5 class in our classification problem. First of all, we fine-tuned the RBF model by using 3-fold Grid Search CV with accuracy as the evaluation metric. The parameters that we tuned are Gamma = {0.1, 0.01, 1} and C = {0.1, 1, 10}. The best estimator is Gamma = 1 and C = 1 with accuracy =0.55. Then we continue our fine-tuning by change the parameter ranges. The parameters of the second-round tuning are Gamma = {0.5, 1, 6} and C = {0.5, 1, 5} . The best estimator is Gamma = 6 and C = 0.5 with accuracy = 0.556. Then based on this group of parameters, we trained the model and had the accuracy around 0.61 for the test set.

The second model that we tried is the Support Vector Machine with Polynomial kernel. With the same reason as above, we included we included Standard Scaler and One Versus Rest Classifier in our pipeline. First of all, we fine-tuned the poly model by using 3-fold Grid Search CV with accuracy as the evaluation metric. The parameters that we tuned are Gamma = {0.1, 0.01, 0.001}, Degree = {3, 5, 6} and C = {1, 50}. The best estimator is Gamma = 0.1, Degree = 6 and C = 50 with accuracy =0.535. Then we continue our fine-tuning by change the parameter ranges. The parameters of the second-round tuning are Gamma = {0.05, 0.1, 0.15} and we keep Degree and C the same as last tuning. The best estimator is Gamma = 0.15 , Degree = 6 and C = 50 with accuracy =0.542. Then based on this group of parameters, we trained the model and had the accuracy around 0.60 for the test set.

The third model that we used is the decision tree model. With $max\_depth = 4$, $random\_state = 4$, and all the other parameters to be default, the decision tree classifier was put into a pipeline with a standard scaler. The accuracy we got was 0.5867. We tried tuning with the following parameters, $min\_samples\_split = [2, 3, 4]$, $max\_depth = list(range(1, 32))$, $max\_features = ['auto',' sqrt']$, $min\_samples\_split = [2, 5, 10]$, and $min\_samples\_leaf : [1, 2, 4]$ with cv=5. The best estimator are $min\_samples\_split = 2$,$max\_depth = 7$, $max\_features =' auto'$,$min\_samples\_split = 2$, and $min\_samples\_leaf = 1$. Then based on this group of parameters, we trained the model and had the accuracy around 0.5893 for the test set. Since the accuracy did not increase by a large amount after tuning, we used random forest as our next model for classification.

The fourth model we tried is the Random Forest Model. After the Random Forest classifier was put into a pipeline with a standard scaler, with $n\_estimator = 20$ and $random\_state = 0$, we got an accuracy of 0.5918. First of all, we fine-tuned the poly model by using 5-fold Grid Search CV with accuracy as the evaluation metric. The parameter that we are tuning is $bootstrap$, max_depth, $max\_features$, $min\_samples\_leaf$, $min\_samples\_split$, and $n\_estimators$. The best parameters we got is $bootstrap = True$, $max\_depth = 50$, $max\_features =' auto'$, $min\_samples\_leaf = 4$, $min\_samples\_split = 10$, and $n\_estimators = 200$. Base on this group of parameters, we trained the model and had the accuracy of 0.6480, which is the highest accuracy value we have so far

The fifth model that we tried is the artificial neural network models. The first model we implemented has one input layer, four hidden layers, and one output layer. The activation function for each layer we used was leaky ReLu and we have used batch normalization for each layer. The number of nodes in each hidden layer was 300, 100, 50, and 30. In the learning rate, we implemented exponential decay. The loss function we used was "$sparse\_categorical\_crossentropy$". The optimizer we used was "$optimizer.SGD$". After 30 epochs, the accuracy we have achieved on the test set was 0.6046. Since the accuracy was lower than the one from the random forest model, we decided to implement another artificial neural model and see if the accuracy can increase.

We thought that it might be that the fifth model contained too many hidden layers and not enough epochs for the model to train. So for the sixth model that we tried, we created a model with one input layer, two hidden layers, and one output layer. We used he normalization at each layer and the activation function that we used for each layer is PreLu with 100 epochs. With the learning rate as in exponential decay, loss function as "$sparse\_categorical\_crossentropy$", optimizer as "$optimizer.SGD$", after 100 epochs, the accuracy we achieved on the test set was 58.93$\%$. However, when we reached the 23rd epoch, the accuracy for the validation set has already reached 0.5636 and not increased in the rest of the 77 epochs, which means that increasing epochs is not enough to increase the accuracy of the sixth model.

5 Conclusion

In summary, our Random Forest model achieved the accuracy of 0.648 and we agree that this result is satisfactory given the availability and quality of current data. We would like to share our models and results with Virginia Department of Forestry. In the future, when people report the location, time and other relevant features of wildfire, Virginia Department of Forestry can apply our model to know the size of wildfire in advance and therefore help them to better decide how much rescue resources should be allocated. This will not only help to protect the forest in Virginia, but also reduce risks faced by fire fighters and Virginia residents when wildfire happens.

In addition to the implication, we would like to share some insights and thoughts about the models that we have currently ran and some future improvement and optimization that we can implement to increase the accuracy of the model, which we have not tried during this semester.

First, we hope to increase the size of data set. The current data set contains around 2500 records after we deleted all Nan values and extracted the data only related to Virginia. However, after we applied Artificial Neural Network to for the classification in our project, the small size of data set caused unsatisfactory accuracy score. We have tried different number of hidden layers, number of nodes, number of epochs, and different types of activation functions and there was not much of an improvement on accuracy, having 60.46 $\%$ as the highest accuracy. The reason might be that the current data set we have is not large enough to run a deep neural network training. Thus, we should try to increase the size of data set by acquiring more recent and complete data of the Virginia state, which will allow us to train the ANN and help to increase the accuracy score given by the ANN model.

Second, stratified sampling can be applied for our data. We realized that the number of wildfire in different class is not evenly distributed. The class of small wildfire contains more records than the class of serious wildfire. After the use of stratified sampling, our train and test data will be stratified, which is more evenly distributed, and thus increase the overall accuracy of the models that we tried.

Third, more features could be extracted from the currently available features. For example, from the feature named Date_Of_Year, we can extract more information such as Weekend_Or_Not, which denoted if the wildfire happened on a weekday or weekend. Besides that, we can create more features by combining the data set with weather data of each records. With the increased number of features, our models can then be more complex and help us better predict the classification of fire size. We believe that more features can help us make more accurate classifications and make our the accuracy of the classification higher.

After completing our original classification problem, we can explore our data set from another perspective. In the future, we are going to apply unsupervised machine learning algorithm for clustering, like K-Means and Hierarchical Clustering, after eliminating the current labels (fire_size_class). Based on the location, date, and other possibly useful features, clustering algorithm can group fires in similar groups. This helps us to decipher structure and patterns in the data set, which are not apparent to the human eye. As for the measurement and evaluation, we will use the Elbow curve to learn the optimal number of groups. Also we will calculate the correlation to make intra-clutser similarity high and inter-cluster similarity low.

6 Member Contribution

Mike Cai is responsible for Visualization of the data, implementation of the Random Forest. Mike Cai was in charge of editing the first half of the video. In terms of the report, Mike Cai contributed in the Abstract, Motivation, and Method section of document.

Yunlu Li is responsible for data cleaning and training of Polynomial SVM and RBF SVM model. As for the video presentation, Yunlu helped to write script and collect pictures. In terms of the report, Yunlu contributed in the Method, Preliminary Experiment, and Future Thoughts section.

Yilin Huang is responsible for implementing the Decision Tree and deep neural network model. Yilin Huang was in charge of editing the second half of the video. In terms of the report, Yilin Huang contributed in the Method and Preliminary section of the document.

7 References

1. Inside Nova. "Wildfire season begins in Virginia", https://www.insidenova.com/news/wildfire-season-begins-in-virginia-as-state-officials-warn-dry/article_d1bbc632-f007-11e9-b17e-9bc2d86e07c9.html

2. Tatman, Rachael. "1.88 Million US Wildfires", https://www.kaggle.com/rtatman/188-million-us-wildfires