Predicting Risk of Fire in Atlanta Area

Prepared by: Beenapreet Kaur

Problem Statement: To Predict High Fire risk spots of Atlanta Area based on various Built Environment Variables using various Machine Learning - Classification Algorithms

Summary: This report explores various classification algorithms such as Logistic Regression, SVM and Random Forest to get the best model for prediction. The report summarizes the results and findings of the analysis which comprised of Data Cleaning and Manipulation techniques, Exploratory Data Analysis, Feature Selection and finally Model building.

Raw Data Cleaning and Manipulation

The Data cleaning and Manipulation process comprised of the following:

1) Dealing with Missing Data – The following steps were taken:

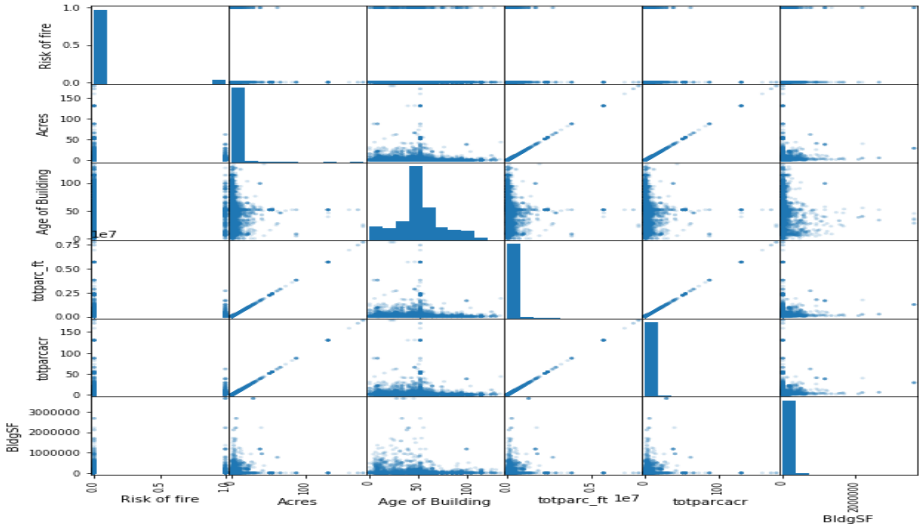
    a. Variables that had more than 50% of missing values were excluded from the data set
    b. Most of the numerical data which had fewer than 15% missing values such as Age of Building, Tax_Value, Last_Sale_Value etc were imputed with the mean of the respective variables
    c. Most of the categorical variables such as Year the building was built, Tax_Year,Class, Construction type was imputed with the mode of the data - For example – MASONRY comstruction type was the most common building type.
    d. Some of the categorical variables such as Lot_Size and Building_sq_ft, floor size etc, depend on the the type of Property(Retail, Industrial, Educational, Residential etc), hence they were imputed with the mean or mode of their property type
    e. Some of the data that had properties built after 2014 and were excluded from the model.
    f. The data built between 2011 and 2013, was flagged as to later consider only in the test data- which predicted risk of fire for the years after 2013.

2) New Variables were created using one or more variables:

    a. Age of Building using Year Built and Current Year
    b. Number of Features and Number of Amenities columns were created by the Features and Amenities column

Exploratory Data Analysis

Exploratory Data Analysis shows us some high multi correlation between variables as shown in the plot



above.

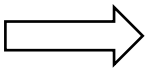
- Variables such as Total_parc_ft is highly correlated with No_of_acres
- Number of Bedrooms present in the building are highly correlated with the number of bathrooms
- Age of the building is Negatively correlated with the Number of Storeys in the building

Hence Feature Selection is important to reduce some of the multi-collinearity

Feature Selection

Before the modelling process the categorical variables such as Construction type, Lot Condition, structure_condition, sidewalks etc were converted to numerical indicator type numeric variables for example:

Index	Structure_Condition
1	Good
2	Good
3	missing
4	Fair
5	Fair



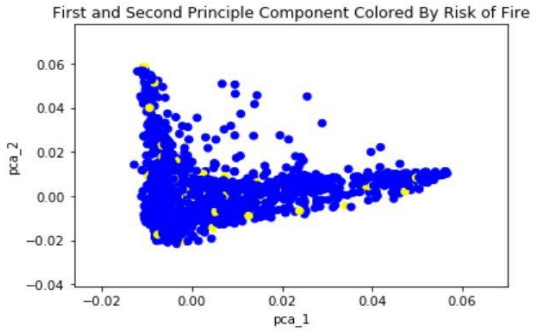
Index	Good	Fair	Missing
1	1	0	0
2	1	0	0
3	0	0	1
4	0	1	0
5	0	1	0

This expanded the number of variables from 65 to 109 variables, hence feature selection is important as:

1) EDA clearly shows that there is a high correlation among the predictor variables - which could lead to redundant variables and hence overfitting and high bias in the model.
2) May introduce the effect of curse of dimensionality – where at high dimensions, distances begin to lose their value) , especially for models that use the distance metric such as SVM
3) To reduce time of modelling, it is imperative to reduce the number of variables as Cross validation and hyper-tuning can become very long procedures for data with large number of variables

The following techniques using Feature selection are explored

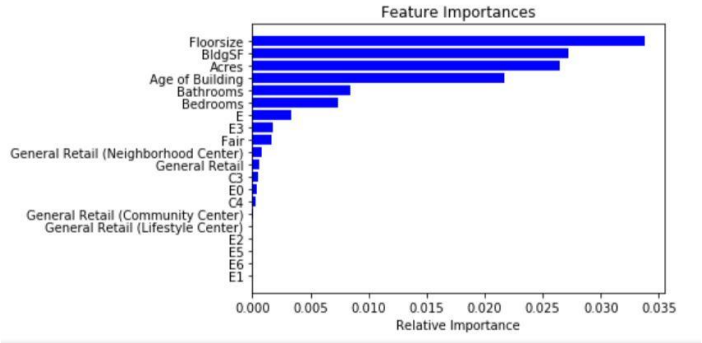
1) Principal Component Analysis:
    a. In this question since the most important part of the problem is prediction and not exactly exploration of variables I chose Principal Component Analysis to get the best result from the Model.
    b. Secondly, by using principal components, I can make the best use of all my variables and not lose out on the variance contributed to the model by a certain variable.
    c. This method also takes care of the multi-correlations between the various variables and captures the maximum variability explained by the model



- Before Principle Component Analysis standardization of data was done as projection of the original data into maximum variance directions would not yield the best result and would project the most in variables with higher variance.
- The first 2 Principal Components only explain 10% of the variation in the model, still, in the figure above we can see some distinction between the yellow (high risk) and blue (low risk) data points
- From the 109 variables I was able to come down to 65 variables. The first 65 principle components were able to explain 85% of the variance in the data.

2) Feature Importance using Random Forest

This step was performed after fitting the Random Forest model, it gave a good overview of which features are important for the model. Using Information Gain, I evaluated how much each variable decreases the weighted impurity in the tree. Thus across 100 trees the impurity gain from each feature is averaged and then ranked. In this model Floor size contributes to the lowest impurity gain and the highest information gain, thus splitting the tree on floor-size variable gives us the highest explanatory variance in the model.



Classification Model

Before starting the modelling, the data was normalized, to mean=0 and std dev=1 as SVM tries to maximize the distance between the separating plane and the support vectors. If one feature (i.e. one dimension in this space) has very large values, it will dominate the other features when calculating the distance. If you rescale all features (e.g. to [0, 1]), they all have the same influence on the distance metric.

The data was then split into 2 parts Training and Testing using the 70-30% rule. The training set would later be used in Cross Validation to evaluate different models and choose the best one.

Explanation for calculation of Response Variables Response Variables:

- For the Training Data the Response Variable was calculated in terms of the number of accidents in the past 3 years, if the Number of Fire Accidents were more than equal to 1, from 2011 to 2013 then the location was classified as High Risk or else no Risk. I used this methodology as for Fire Accidents, the cost of misclassification is extremely high, one has to make sure, that even the slightest possibility of fire, should be flagged, or if over looked it could cause a lot of damage in property and even life.
- For the Test Data I would be predicting the Risk of Fire – High or Low and evaluating that based on whether or not Fire occurred in the year 2014.

The following models were used:

1) Logistic Regression: To calculate the score of the Logistic Regression model accuracy, would not be appropriate as the labels are extremely disproportionate with around 73% of the data with no risk of fire and the remaining 27% with a High risk. Thus, I have used, the roc_auc ($Roc = \frac{TPR}{FPR}$) score which is a combination of both the Precision and Recall, However in our case as the cost of misclassification is high, The Recall ($Recall = \frac{True\ positives(TP)}{Total\ Actual\ Positives}$) the most important metric. That is we want to ensure that if an area is at a high Risk of fire, we flag it, and it does not go un-noticed.
2) Support Vector Machines: (Note:Before SVM standardization was done) For this model, there are 2 parameters C – which is the cost of classification and gamma – which is used for handling non linear Kernels. For this data, the best accuracy of the model is obtained by using Grid Search Hyper Tuning method, which is a greedy algorithm that selects the best permutations of all the Hyper parameters involved in the model. The best fit parameters are C=1 and kernel=Linear. For our model, we need a high C value in our case as the Cost Of misclassification is high. However, for this model we obtain very similar results for a linear Kernel using C values ranging from 0.001 to 1000.
3) Random Forest: For Random Forest Algorithm, I have used Hypertuning for 3 parameters:
    a. Number of Estimators - > This corresponds to the number of trees, for which the results should be averaged over.
    b. Maximum Depth-> This corresponds to the number of branches of the tree, Since the data has 109 variables I have used branching from 10 -30 in the grid search.

c. Maximum Features-> This corresponds to the top number of features that should be used in each branch for best classification

After running Hyper Tuning the best parameters obtained are:

Number of Estimators=100 ; Maximum Depth=10; Maximum Features=20

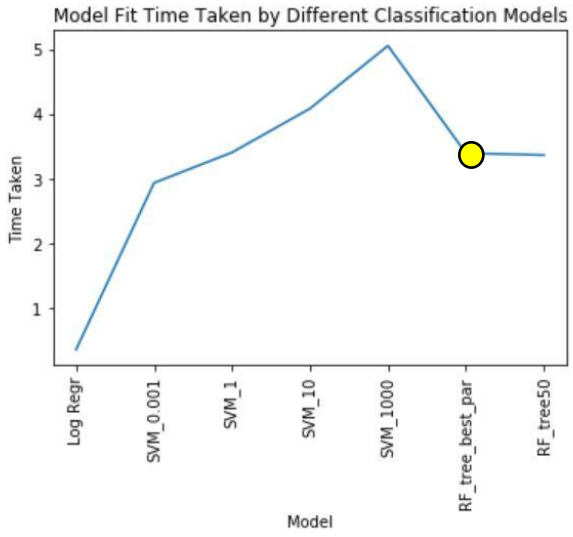
Results

After performing 10- fold cross validation on the same training set for each of the following models with varying parameters the results are:

Logistic Regr	SVM_C_0.01	SVM_C_1	SVM_C_10	SVM_C_1000	RF_tree_best_par	RF_tree_50
52.19%	94.86%	94.729%	94.729%	94.73%	97.16%	97.3%

Using the above results, I chose the RF_tree_100 model, as it gives a better averaged out performance across various trees. The 50 tree model may be fitting the noise and randomness of the training data too well, hence the higher accuracy.

The graph below shows the time taken to run each of the models:



Test Accuracy – After Running the data on Test Set using the RF_tree_best_par model (highlighted in bold above)– The accuracy obtained is 90.13%