Forest Fires: A Predictive Analysis

By: Nadav Levanoni and Johnathan Ly

# Abstract:

In 2018, 97 civilians and 6 firefighters died from forest fires in California. The predictive analysis that was conducted used data on forest fires and weather conditions from 2012-2015 to train machine learning models, specifically a linear regression model and a neural network model, to predict the size (in acres) of a fire. The objective behind running this predictive analysis is to use the information gained by the analysis to better prepare civilians and firefighters for future fires. Our machine learning models turned out to be less accurate than desired and expected. A comparison of the visualizations of the results of the two models showed that the linear regression model was more insightful than the neural network model.

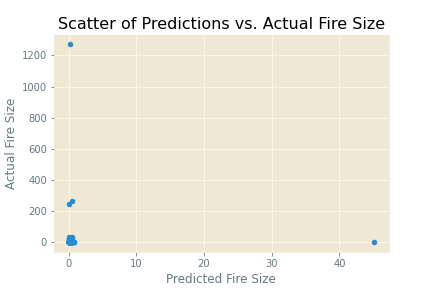
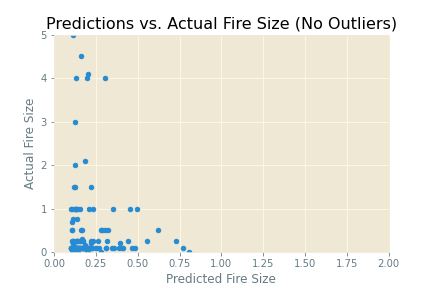
# The Data:

Data was pulled from two Kaggle datasets and one from data.gov: The first Kaggle dataset is about U.S. fires from 1992-2015 in a sqlite format. The second is about hourly weather conditions for major cities from 2012-2017 via multiple csv files per weather condition. Lastly, the data.gov file is about monthly temperature for California counties from 1895-2020 in csv format. The data from these three sources were subset to data from the following three greater areas of California: Los Angeles, San Francisco, and San Diego for 2012-2015. Hourly weather conditions were grouped more broadly into daily weather conditions, and weather descriptions were converted into boolean values pertaining to whether or not it had rained on that day. The data from all three sources were merged into the final dataset, which consisted of the following columns: Size (size of the fire in acres), Month\_Temp (average temperature for the month in which the fire broke out), datetime (the date on which the fire broke out), City (the city/area in which the fire broke out), Humidity (average humidity throughout the day in which the fire broke out), Pressure (average pressure throughout the day in which the fire broke out), Wind Speed (average wind speed throughout the day in which the fire broke out), hasRained (0 for did not rain on the day the fire broke out, 1 for did rain on that day).

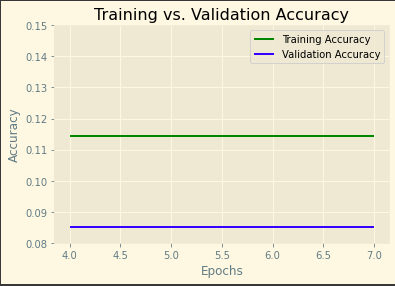
# Analysis:

The indices of the data were shuffled and the data was split into training (80% of the data), verification (10% of the data), and testing (10% of the data) sets. Correlations between each numerical input attribute and the Size column were computed. It was revealed that all of the correlations had extremely small magnitudes, and so it was decided that all of the numerical input attributes would be used as inputs in the machine learning models.

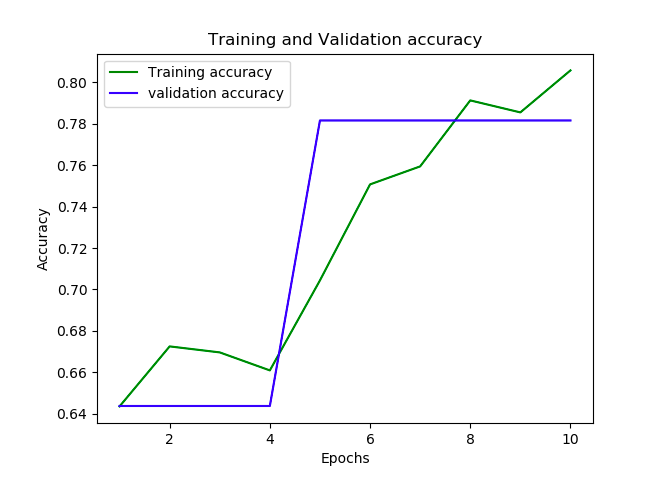
The linear regression model was fit using the training set and after running predictions on the verification set, it was decided that the model would have a polynomial bias of 2 and that the predictions would be scaled such that they were closer to the actual target values. When using the model to make predictions on the testing set, it was computed that the average difference in predicted Size and actual Size was 13.4 acres. Scatterplots of the predicted Sizes vs. the actual Sizes for the testing set are shown below, with the second scatterplot being a “zoomed in” version of the first:

The neural network model had three layers: the input layer and hidden layer had eight perceptrons and used the activation function of “relu,” while the output layer had one perceptron and used the activation function of “softmax.” The neural network model was compiled using the optimizer function “rmsprop” and the loss function “mean\_squared\_logarithmic\_error.” The model was fit using the training and verification data, using four epochs and a batch size of sixteen. The accuracy of the model when used to make predictions on the testing set was always around 0.11, regardless of how many epochs were used or how large the batch size was determined to be. The following visualization display how training and validation accuracy change with how many epochs are used show the oddity of how the accuracy remained consistent no matter how the model was compiled:



A more ideal plot of how training and validation accuracy should change as the number of epochs used changes looks more or less like the following (taken from the internet):



# Conclusion:

Both of the machine learning models turned out to be less accurate in predicting the size of a fire than what was expected. The visualizations provide some helpful insight however, they show that (1) the linear regression model was more accurate in predicting smaller fires than the largest of fires, or outliers and (2) the neural network model was likely not built or compiled correctly, since the accuracies should change as the number of epochs used change. This analysis allowed us to familiarize ourselves with the process of gathering and cleaning data to be used in a machine learning model, implementing machine learning models, and visualizing the results of running analysis with those models.