Sonoma County Wildfire Risk Analysis

San Jose State University
Department of Applied Data Science
CMPE 257-Section 22
Spring 2020 Machine Learning

Afra Bijli 008340631 afra.bijli@sjsu.edu
(Ginny) Jeehee Choi 012565696 jeehee.choi@sjsu.edu
Ililta Gebrihiwet 013828048 Ililta.gebrihiwet@sjsu.edu
Sukriti Mishra 014580696 sukriti.mishra@sjsu.edu
Sunanda Das Suchi 013866684 sunandadas.suchi@sjsu.edu

Section 1 - Introduction

Motivation of Project

Wildfires are some of the most destructive, yet preventable forces of nature. Not only have wildfires cost billions of dollars in damage, but the high costs are also due to the resources and time spent that go into fighting these fires. A myriad of factors contribute to the increased risk of wildfire ignition. Natural factors include: hot and dry weather conditions, soil moisture, and presence of trees and shrubs. Human factors contribute to over 80% of wildfires in the US [1]. Human-caused fires include: campfires, housefires, gas leaks, arson, and falling power lines [2]. A good example of a human-caused fire was the Kincaide wildfire in Sonoma County, California back in October of 2019. This fire was caused by downed power lines that landed on the trees and sparked a blaze; the burned area was over 70,000 acres [3]. Although only 5% of wildfires are caused by power lines in California, these fires are much larger, especially when weather factors, such as wind speed, are part of the equation [2]. The probability of a powerline sparking a fire increases as the wind speed increases [2]. Human activity has also been the source of the global shift in weather patterns over time. Severe weather patterns such as lack of precipitation, extreme heat, droughts, and dry winds, along with natural vegetation are further increasing the risk of wildfire outbreaks. Consequently, wildfire seasons are becoming longer and more unpredictable in many parts of the world, such as the United States, in particular, California.

Highlight Major Differences Between Your Work and Existing Work

The existing work are (1) prediction of carbon dioxide impact on the wildfires with the development of Changed Climate Fire Modeling System (2) analysis of burning patterns using Geographic Information System(GIS), (3) Prediction of fine particulate matter (particles with aerodynamic diameter less than 2.5 micrometer) concentrations during Northern California wildfires by using different machine learning models. The major difference between our work and other existing works are (1) Collection and then classification of the remote sensing dataset (NASA satellite images dataset), of Sonoma county in Northern California, (2) Implementation of different machine learning models (Logistic Regression, Random Forest, KNN, and SVM) on the preprocessed dataset, to calculate the accuracy rate of different implemented models, (3) Cross validation and models comparison for the risk prediction of wildfires occurrence.

Project Organization and Structure

There are three purposes of this project: (1) Collection of datasets such as Fire History dataset, weather dataset, Terrain and surface dataset and human activities, (2) Selection of different ML models according to the attributes and its impacts on wildfires, and then implementation of selected models and AI algorithms on the collected datasets (3) Comparative studies of different models on the basis of accuracy percentage to determine which model is best for the risk prediction of wildfires occurrence.

Related Work

The term wildfires mean an uncontrolled fire caused by natural vegetation. There are both natural and human factors that can lead to wildfires. In North America, wildfires caused by natural vegetation known as ‘brushfires’ whereas in Australia it is known as ‘bushfires’. The bushfire occurrences from various climate data contain spatial and temporal information as well as climatic aspects of bushfires. There is an aim to derive the contextual information of a model to calculate accurate prediction for future bushfire hotspots. From weekly climatic surfaces, researchers developed an ensemble method on the basis of a two-layered machine learning model to establish a relationship between fire incidence and climatic data to provide highly accurate bushfire incidence hot spot estimation with a global accuracy of 91% [4].

In Sonoma county, human activities were responsible for the wildfires whereas in Australia, due to natural causes such as lowest rainfall, high temperature, and so on. If we consider both the events to create the model then it is highly likely that the model will not be accurate enough for similar conditions. Hence, we have focused on gathering the details on Sonoma County wildfire so that the model built will be applicable to test other wildfire occurrences that were caused due to human intervention.

In the US, fire danger is rated and fire behavior prediction models are used to aid in fire management decision-making. The US uses the National Fire Danger Rating System (NFDRS) as a tool to estimate the risk of seasonal fire occurrence on a given day for a given area. NFDRS is a “climatology-based system”[5] that uses “current, historical, and forecast weather data” [5]. Analysis of historical fire danger is important as it will aid in the accurate interpretation and application of wildfire indices [5]. Data on

historical instances of wildfires is closely tied with weather data as they are both location-based factors. Weather factors and site descriptions, for example "fuel model and slope class, are used to calculate fuel moisture values" [5], which are then used to calculate wildfire indices such as Burning Index (BI). BI measures the intensity of the wildfire and determines the probability of containment.

Section 2 - Project Team and Roles

Project work distribution among members as follows:

Responsibility	Team Member Contribution
Data Preparation	All
Data Preprocessing	All
Data Visualization	All
Random Forest Classifier (Fire history & Weather)	Afra Bijli
KNN, Logistic Regression (Fire history and weather dataset merged with NDVI)	Sunanda Das Suchi
Support Vector Machine (Fire history and weather dataset merged with NDVI)	Sukriti Mishra
Logistic Regression & Random Forest (Human factors)	Ililta Gebrihiwet
Logistic Regression (Fire history & Weather)	(Ginny) Jeehee Choi

Table 1: Task Distribution

Section 3 - Project Data Preparation

We have decided to focus on the Northern California county of Sonoma for our wildfire risk prediction. We are planning to obtain location-based data on the four main factors that contribute to wildfires: fire history, weather, NDVI and human factors. Our datasets will include historical data that covers a time period of 10 years.

1- Wildfire History:

We plan to use the ArcGIS tool to extract and map the wildfire history data.

A) Data Preparation

Fire history dataset is acquired from the Fire and Resource Assessment Program's fire perimeter GIS data. The dataset has burned area, location, ignition time, fire boundary, and cause variables with date ranges from January 1st 2010 to 2019 December 31st. Project coordination system used in ArcGIS is NAD 1983 California Teale Albers.

Each variable is identified as below:

Burned area: Burned_Area (km^2)

Location: Lat_Long (latitude, longitude), State

Ignition time: Ignition_Time (day)

Fire Boundary: Fire Boundary (km)

Cause: Cause

B) Data Preprocessing

Two fire history data shapefiles have all the above variables except severity. Data is preprocessed to increase the accuracy of the result. Excluded null values in year, replaced missing 'Year' values with year value in a 'Start_Date', and selected fire events occurred after January 1st 2010 and located within Sonoma county boundary. Created 'ignition time' variable using currently available variables 'start_date' and 'end_date'. Changed the burned area metric from acres to km^2. Created field for longitude, latitude, and calculated fire boundary (km) using fire shapefile perimeters in Calculate Geometry function. Unnecessary fields are dropped and checked null values.

C) Data Visualization

```
fire_train.head()
```

	OBJECTID	YEAR_	Ignition_Time	Fire_Boundary	Burned_Area	CAUSE	Lat_Long	ALARM_DATE	CONT_DATE	Latitude
0	2	2012	2	0.960190	0.023183	10	38.337645251,-122.739026393	2012-10-01	2012-10-02	38.337645
1	3	2012	1	2.221969	0.119592	14	38.288162914714 3,-122.854540249021	2012-08-21	2012-08-21	38.288163
2	4	2013	1	0.728971	0.032011	11	38.531183206647 7,-123.201231145777	2013-09-25	2013-09-25	38.531183
3	5	2013	2	3.764155	0.643220	14	38.343034621539 1,-122.652953225 3	2013-10-04	2013-10-05	38.343035
4	6	2013	1	1.424529	0.086790	9	38.743806623356 2,-122.985053386708	2013-05-08	2013-05-08	38.743807

Figure 1 : Fire history training set

```
fire_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 13 columns):
OBJECTID        25 non-null int64
YEAR_           25 non-null int64
Ignition_Time   25 non-null int64
Fire_Boundary   25 non-null float64
Burned_Area     25 non-null float64
CAUSE           25 non-null int64
Lat_Long        25 non-null object
ALARM_DATE      25 non-null datetime64[ns]
CONT_DATE       25 non-null datetime64[ns]
Latitude        25 non-null float64
Longitude       25 non-null float64
Shape_Length    25 non-null float64
Shape_Area      25 non-null float64
dtypes: datetime64[ns](2), float64(6), int64(4), object(1)
memory usage: 2.7+ KB
```

Figure 2: *Fire history training set info*

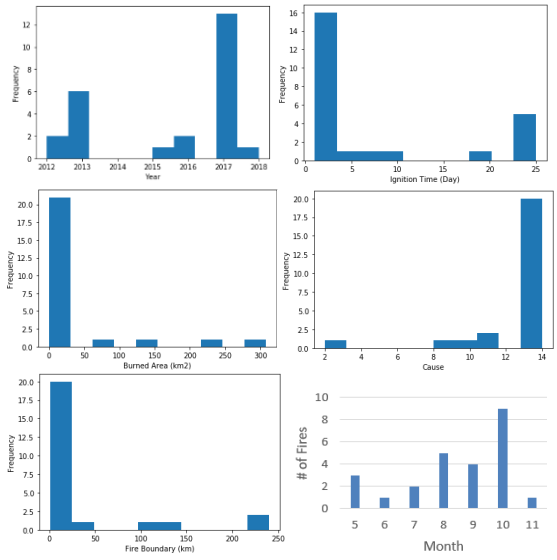


Figure 3: Trend plots of various fire history data features

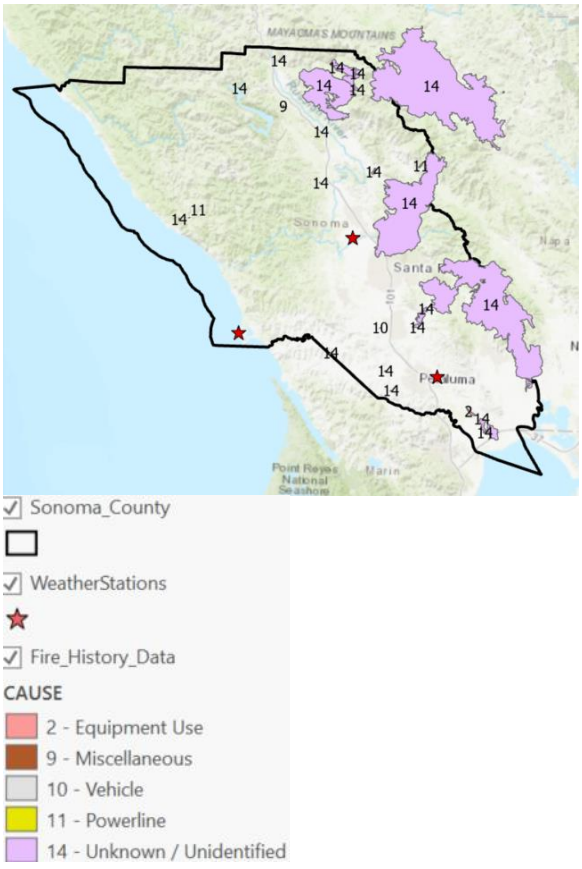


Figure 4: Sonoma County with Fire History and Weather Datasets on ArcGIS

The above map shows Sonoma county with fire history data. The different color and number indicates different causes of fire. The red star indicates the weather station.

While 19 types of fire causes are available in options, only 5 types are the cause in Sonoma County. The main cause of fire in Sonoma county is unknown or unidentified with 20 occurrences for the past 10 years. The average fire ignition time is 7.4 days and median is 2 days. The average burned area is 31 km^2 and median is 0.15 km^2. Fire most frequently occurred between August to October.

D) Training and Test Data

The fire history dataset is combined with the weather dataset to predict the fire risk depending on weather status. Training and testing datasets are split into 80:20 ratios using train_test_split function.

	DailyAvgDr	DailyAvgPr	DailyAvgRe	DailyAvgHo	LongLat_X	LongLat_Y	FireEvent	Year	Month	Day
0	52.33	0.01	98.90	0.92	-122.8102	38.5038	0	2010	1	1
1	48.65	0.00	95.12	3.90	-122.8102	38.5038	0	2010	1	2
2	41.05	0.00	96.71	1.22	-122.8102	38.5038	0	2010	1	3
3	43.59	0.00	96.48	1.93	-122.8102	38.5038	0	2010	1	4
4	43.04	0.00	96.88	0.95	-122.8102	38.5038	0	2010	1	5

Figure 5: An Example of combined dataset of fire history and weather datasets

E) Data Validation

Using the train_test_split function, training data is validated. The training dataset's Mean Absolute Error(MAE) is 0.013 while the validation dataset's MAE is 0.012.

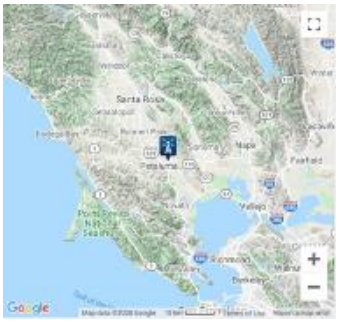
2- Weather:

Hourly, location-based weather data had been obtained from weather stations across Sonoma county in the format of a csv text file. This remote sensing data values will be cleaned and analyzed using Jupyter notebook and visualized using ArcGIS. The parameters of this weather dataset will include: WBAN, Station ID, temperature, precipitation, wind speed, and humidity.

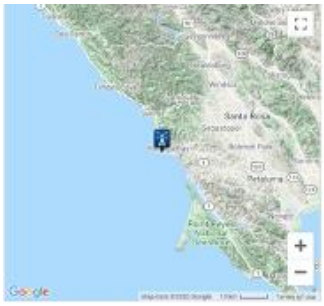
A) Data Preparation:

Hourly weather data was retrieved from 3 weather stations across Sonoma County from the NOAA, Climate Data Online, Climatology website: https://www.ncdc.noaa.gov/cdo-web/

SANTA ROSA SONOMA CO AIRPORT, CA US & WBAN: 23213 STATION ID- This hourly weather data was obtained for the time period of January 1st, 2010 - December 31st, 2019.



PETALUMA MUNICIPAL AIRPORT, CA US & WBAN: 00320 STATION ID- This hourly weather data was obtained for the time period of July 31st, 2014 – Dec 31st, 2019 (No data before July 31st, 2014).



BODEGA 6 WSW, CA US & WBAN: 93245 STATION ID- This hourly weather data was obtained for the time period of January 1st, 2010 - December 31st, 2019.

Figure 6: Weather station locations

B) Data Preprocessing

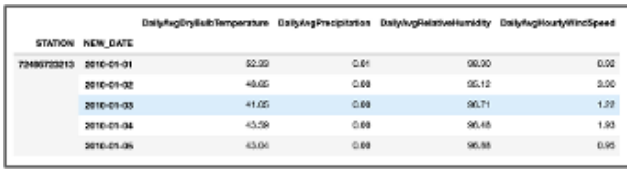
In order for our model to yield the best results, it is important that the data collected is pre-processed and clean of any inconsistencies, errors, and missing values as this can cause results to skew. Imputation is a process used to deal with missing values. Missing values are imputed based on the datatype of the values of a certain attribute. Another reason imputation is a good idea is because it can help avoid further shrinking the dataset. The less data we have to work with, the more likely it will impact important metrics, such as the accuracy of a model's prediction.

Feature engineering is another equally important process in the data cleansing phase. It is a fundamental technique that involves careful selection of relevant attributes of the dataset. Relevant attributes meaning, the attributes that are necessary for solving the problem, in this case, predicting the risk of wildfire occurrence. Feature engineering is a great way to drastically improve the performance of machine learning models. Essentially, the size and quality of a dataset are important in ensuring quality output from a model. We'd like to also know whether there is inherent class imbalance in the dataset so that we can

correct for it as this can cause bias leading to misleading results.

There were many irrelevant features in our weather datasets and so using feature engineering, 124 attributes were stripped down to only 6 important features related to fire risk. There were two weather datasets. The first weather dataset contained data from two weather stations combined: Station IDs 93245 & 23213. The second weather dataset contained data from the last weather station: Station ID 00320. Missing values were imputed to each of the two datasets separately. We had noticed there were many missing values , so we imputed hourly temperature columns with the mean temperature of the dataset. We selected the mean temperature of the dataset because it seemed to be normally distributed. The hourly precipitation, humidity, and wind speed columns were imputed with the median value as these distributions seemed skewed; hence, imputing with the mean value for these columns would not have been a good idea.

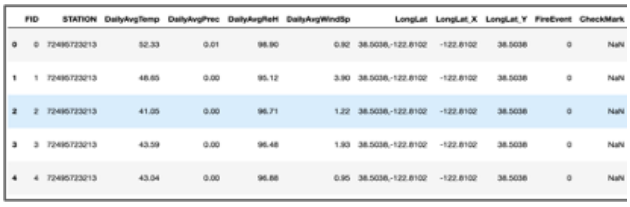
After the imputation process, the weather data from all three stations were then vertically merged together, which resulted in a total of 9,184 records. We then aggregated the hourly weather data from these three stations to get the daily average for each feature or weather factor by station ID and date. The reason the data was aggregated in this manner was because we needed to link the weather dataset to the fire history dataset, as it is also location-based data.



STATION	NEW_DATE	DailyAvgDryBulbTemperature	DailyAvgPrecipitation	DailyAvgRelativeHumidity	DailyAvgHourlyWindSpeed
72495723213	2010-01-01	52.33	0.01	98.90	0.92
	2010-01-02	48.65	0.00	95.12	3.90
	2010-01-03	41.05	0.00	96.71	1.22
	2010-01-04	43.59	0.00	96.48	1.93
	2010-01-05	43.04	0.00	96.88	0.95

Figure 7:. Resulting weather dataframe from vertical merging and aggregation of all three weather station data

Lastly, we merged both the pre-processed fire-history and weather datasets into one single dataset to obtain the final fire-history and weather dataset, which is the dataset that will be used to train the Random Forest Classifier (RFC) model with in order to predict wildfire risk in Sonoma county.



	FID	STATION	DailyAvgTemp	DailyAvgPrec	DailyAvgReH	DailyAvgWindSp	LongLat	LongLat_X	LongLat_Y	FireEvent	CheckMark
0	0	72495723213	52.33	0.01	98.90	0.92	38.5038,-122.8102	-122.8102	38.5038	0	NaN
1	1	72495723213	48.65	0.00	95.12	3.90	38.5038,-122.8102	-122.8102	38.5038	0	NaN
2	2	72495723213	41.05	0.00	96.71	1.22	38.5038,-122.8102	-122.8102	38.5038	0	NaN
3	3	72495723213	43.59	0.00	96.48	1.93	38.5038,-122.8102	-122.8102	38.5038	0	NaN
4	4	72495723213	43.04	0.00	96.88	0.95	38.5038,-122.8102	-122.8102	38.5038	0	NaN

Figure 8: Sample snippet of the final fire-history weather dataset after completing all data cleansing steps and how it is organized

C) Data Visualization

```
df1 = pd.read_csv('Weather_FireHistory_TrainData.csv')
```

```
df1.head()
```

	FID	STATION	DailyAvgDr	DailyAvgPr	DailyAvgRe	DailyAvgHo	LongLat	LongLat_X	LongLat_Y	FireEvent	CheckMark	W_Date1
0	0	72495723213	52.33	0.01	98.90	0.92	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-01 0:00:00
1	1	72495723213	48.65	0.00	95.12	3.90	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-02 0:00:00
2	2	72495723213	41.05	0.00	96.71	1.22	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-03 0:00:00
3	3	72495723213	43.59	0.00	96.48	1.93	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-04 0:00:00
4	4	72495723213	43.04	0.00	96.88	0.95	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-05 0:00:00

Figure 9: Fire-History Weather Dataset

```
# Let's rename the columns so it's easier to read
df1.rename(columns={'DailyAvgDr':'DailyAvgTemp','DailyAvgPr':'DailyAvgPrec',
                    'DailyAvgRe':'DailyAvgReH','DailyAvgHo':'DailyAvgWindSp'}, inplace=True)
```

```
df1.head()
```

	FID	STATION	DailyAvgTemp	DailyAvgPrec	DailyAvgReH	DailyAvgWindSp	LongLat	LongLat_X	LongLat_Y	FireEvent	CheckMark	W_Date1
0	0	72495723213	52.33	0.01	98.90	0.92	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-01 0:00:00
1	1	72495723213	48.65	0.00	95.12	3.90	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-02 0:00:00
2	2	72495723213	41.05	0.00	96.71	1.22	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-03 0:00:00
3	3	72495723213	43.59	0.00	96.48	1.93	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-04 0:00:00
4	4	72495723213	43.04	0.00	96.88	0.95	38.5038,-122.8102	-122.8102	38.5038	0	NaN	2010-01-05 0:00:00

Figure 10: Fire-History Weather Dataset after renaming columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9184 entries, 0 to 9183
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   FID             9184 non-null   int64
 1   STATION         9184 non-null   object
 2   DailyAvgTemp    9184 non-null   float64
 3   DailyAvgPrec    9184 non-null   float64
 4   DailyAvgReH     9184 non-null   float64
 5   DailyAvgWindSp  9184 non-null   float64
 6   LongLat         9184 non-null   object
 7   LongLat_X       9184 non-null   float64
 8   LongLat_Y       9184 non-null   float64
 9   FireEvent       9184 non-null   int64
 10  CheckMark       3742 non-null   object
 11  W_Date1         9184 non-null   object
dtypes: float64(6), int64(2), object(4)
memory usage: 861.1+ KB
Data Shape: (9184, 12)
```

Figure 11: Fire-History Weather Dataset for Sonoma County information

D) Training and Test Data

The fire-history weather dataset is used to predict the fire risk depending on the weather conditions. Training and testing datasets are split into 67:33 ratios using train_test_split function.

E) Data Validation

We had calculated the cross validation score using 10 folds and scoring type as roc_auc in order to test the performance of the random forest classifier model.

3- Terrain and Land cover: For the third factor i.e Land cover and terrain surface, we calculated NDVI for Sonoma County. For human and terrain land

surface dataset:For wildfire prediction project, land surface dataset plays a significant role in the prediction. We can calculate wildfire risk indices such as NDVI - Normalized Vegetation Index,EVI - Enhanced Vegetation Index,MSAVI - Modified Soil Adjusted vegetation Index,BAI - Burn Area Index and NBR - Normalized Burn Index. For this project we calculated NDVI. The Normalized Difference Vegetation Index (NDVI) is a numerical indicator that uses the visible and near-infrared bands of the electromagnetic spectrum and is adopted to analyze remote sensing measurements and assess whether the target being observed contains live green vegetation or not.he formula for NDVI is:

NDVI= (NIR-RED) / (NIR+RED)

where NIR = Near Infra-red light and Red = Red light.NDVI values always range from -1 to +1. For example, when we have negative values, it means that there is a higher possibility that it's water. On the other hand, if you have a NDVI value close to +1, there's a high chance that it's dense green leaves.But when NDVI is close to zero, there are not green leaves and it could even be an urbanized area.

A) Data Preparation:

For this project, we faced so many challenges for the data collection section. First, we tried the NASA Earth Data remote sensing data for Sonoma county, california. We got the files in the HDF format which we found really very difficult to convert into binary files.Second, we got some csv files but contain land cover parameters only for the land and terrain surface dataset. Third, We emailed the Customer Services of Earth Resources Observation & Science Center (EROS) to place the order for the Landsat-8 dataset. Finally, we selected the **Landsat Analysis Ready Dataset(ARD)-7 Data. Under the Landsat, we selected the Landsat 7 ETM+(Enhanced Thematic Mapper (ETM +) sensor and downloaded all the required Geotiff data product files for which login required.**Each zip file contains the tif band images and text file. We downloaded all the zip files(80GB) from 1/1/2010 to 12/31/2019. Required band images(tif) accessed with the help of **QGIS tool and calculated the NDVI statistics with the help of a raster calculator.**

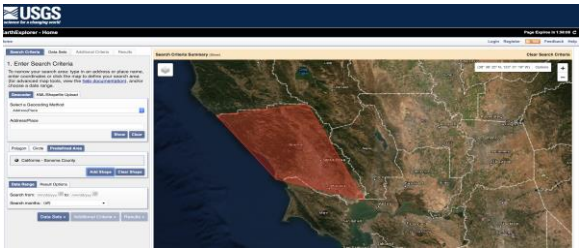


Figure 12: USGS Earth Explorer for Sonoma county

B) Data Preprocessing :

For data preprocessing, we calculated the NDVI formula for Landsat 7 remote sensing data. The formula is(band 4- band 3)/(band 3+ band 4) In qGIS tool, first we uploaded our band 3 and band 4 images. Then we calculated the NDVI statistics with the help of a raster calculator dated 1/1/2010 to 12/31/2019 for those images where the sensor completely covered the our selected location i.e Sonoma county. There are around 400+ NDVI images after calculation. We collected the statistics for those NDVIs and merged the data with the Sonoma county fire history and weather dataset. So, our final dataset contains the 7 independent variables(date, NDVI mean, temperature, precipitation, wind speed, relative humidity, and burned area) and 1 target variable(0=no fire, 1=fire). As from the final dataset, it is very clear that it is a classification problem where we tried to calculate the risk assessment for wildfires. The NDVI images before and after fire are shown below:

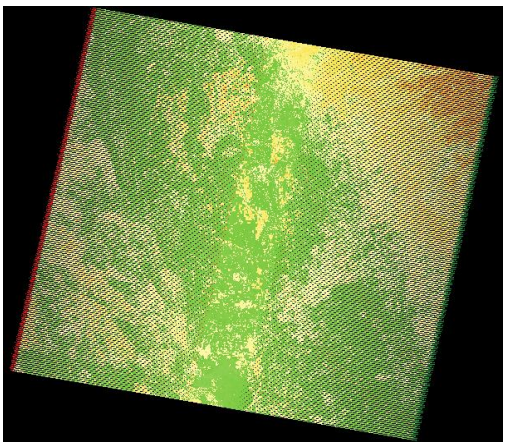


Figure 13: No Fire

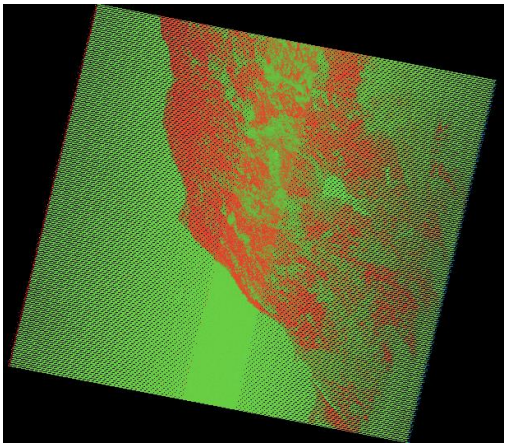


Figure 14: Fire

C) Data Visualization:

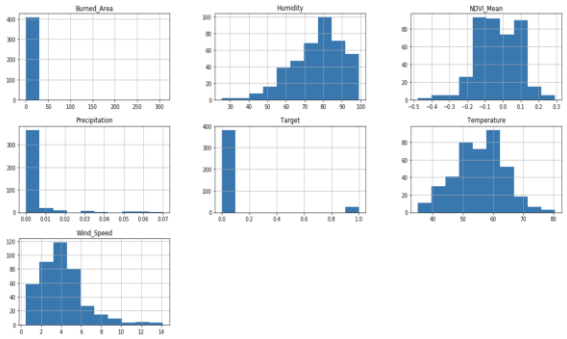


Figure 15: Histograms for NDVI

```
sns.countplot(x='Target', data=data)
data.loc[:,'Target'].value_counts()
```

```
0    381
1     26
Name: Target, dtype: int64
```

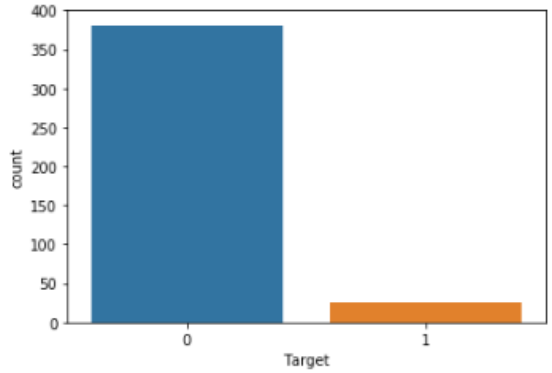


Figure 16 : Number of target variables for NDVI

D) Training and Test Data:

```
data.describe()
```

	NDVI_Mean	Target	Temperature	Precipitation	Humidity	Wind_Speed	Burned_Area
count	407.000000	407.000000	407.000000	407.000000	407.000000	407.000000	407.000000
mean	-0.029196	0.063882	55.523931	0.002482	76.482408	4.132629	0.813626
std	0.119732	0.244844	8.120177	0.009204	13.738028	2.247313	15.278814
min	-0.477279	0.000000	35.140000	0.000000	25.600000	0.480000	0.000000
25%	-0.114239	0.000000	50.215000	0.000000	67.550000	2.550000	0.000000
50%	-0.027559	0.000000	55.900000	0.000000	79.180000	3.910000	0.000000
75%	0.068728	0.000000	61.015000	0.000000	86.110000	5.095000	0.000000
max	0.292164	1.000000	80.500000	0.070000	99.020000	14.160000	307.904696

Figure 17: NDVI dataset for sonoma county

```
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 407 entries, 0 to 406
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   NDVI_Mean      407 non-null    float64
 1   Target         407 non-null    int64
 2   Temperature    407 non-null    float64
 3   Precipitation  407 non-null    float64
 4   Humidity       407 non-null    float64
 5   Wind_Speed     407 non-null    float64
 6   Burned_Area    407 non-null    float64
dtypes: float64(6), int64(1)
memory usage: 22.4 KB
```

Figure 18: NDVI data types for sonoma county

4- Human Causes

Human factors data is a supporting data used in fire prediction. One of the factors that is known for fire ignition is powerline. Hence it's important to model and see how the human factors contribute to causing fire. We will be focusing on powerline, highway, railroad and structures as predictors. The fire history data will be used to check fire occurrence.

Features:Latitude, Longitude, Highway, Railway, ,Powerline, Structure density.

A) Data Preparation

For the human factors dataset, we have collected shapefiles for the Electric transmission line from the California Energy Commission website. Shapefiles for Highway was obtained from Data.gov website, Railroad from Caltrans GIS data, Structures and Sonoma county map from Sonoma County GIS website. The modelling approach is to create a gridded dataset from the shapefiles mentioned above.

To create the dataset, a grid resolution of 2km x 2km was selected. The grid for Sonoma county was created using Qgis, a tool used for analysing geospatial data, using the coordinates of the extent of the squared grid around Sonoma County. Grid points of Sonoma

County were selected by finding the intersection of the squared grid with the shapefile of Sonoma County. The total number of grid points is 1414 data points. The shapefiles for powerline, highway, railroad,structures and fire history were mapped to the grid of Sonoma county. To create the data points for powerline, highway and railroad, we calculated the length of these variables per grid point, the line mile. For the structures, we counted the number of structures per grid point to measure the density of the structures.

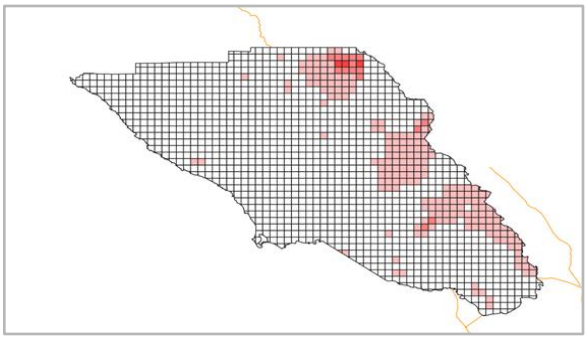


Figure 19: It shows wildfire events in Sonoma County. Sonoma County shapefile was obtained from the Sonoma County website, and square grids in 2km x 2km scale.

Figure 20: Powerline shapefile mapped to Sonoma County Grid

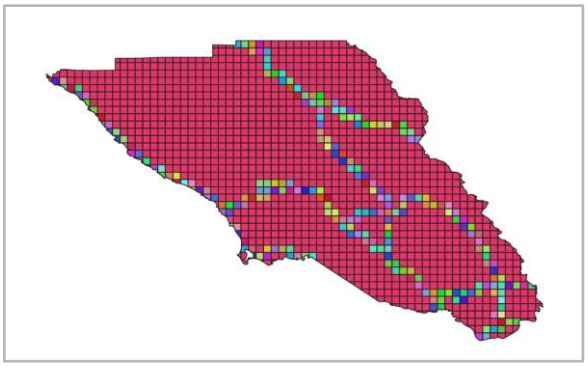


Figure 21 : Highway shapefile mapped to Sonoma County Grid

Figure 22 : Railway shapefile mapped to Sonoma County Grid

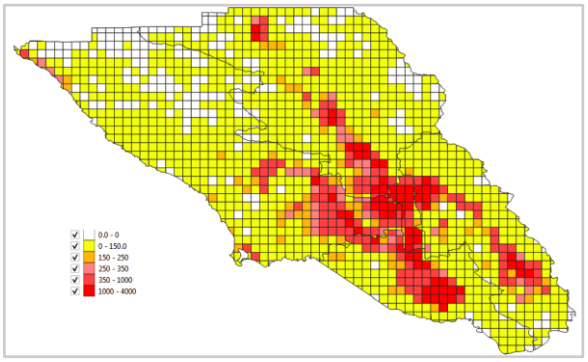


Figure 23 : Structures shapefile mapped to Sonoma County Grid

B) Data Preprocessing

Data preprocessing is an important step before feeding any data to the machine learning models. Data cleaning and feature selection can have an impact in the accuracy of the machine learning models.For cleaning, we selected the relevant features and dropped the unnecessary ones. The features have different ranges of values, hence these features were normalized to constrain the values between 0 & 1.

C) Data Visualization

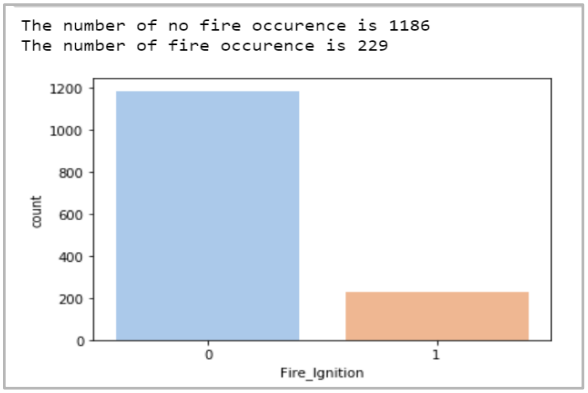


Figure 24: Number of for target variable

```
HF_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1415 entries, 0 to 1414
Data columns (total 7 columns):
Latitude              1415 non-null float64
Longitude             1415 non-null float64
Highway               1415 non-null float64
Railway               1415 non-null float64
Powerline             1415 non-null float64
Structures_Density    1415 non-null int64
Fire_Ignition         1415 non-null int64
dtypes: float64(5), int64(2)
memory usage: 77.5 KB
```

Figure 25: Human Factors training set info

D) Training and Test Data

The human factors data is used to predict fire occurrence. The data is split into 70:30 ratios into training, testing and validation set respectively.

	Latitude	Longitude	Highway	Railway	Powerline	Structures_Density	Fire_Ignition
1390	38.256	-122.331	0.000	0.000	0.000	0	0
1391	38.238	-122.331	4945.542	0.000	2459.345	2	0
1392	38.220	-122.331	0.000	0.000	0.000	0	0
1393	38.400	-122.313	0.000	0.000	0.000	1	1
1394	38.382	-122.313	0.000	0.000	0.000	13	1
1395	38.364	-122.313	0.000	0.000	0.000	3	0
1396	38.346	-122.313	0.000	0.000	1570.322	15	1
1397	38.328	-122.313	5063.439	0.000	1638.613	14	1
1398	38.310	-122.313	0.000	1603.976	0.000	15	1
1399	38.292	-122.313	0.000	124.826	0.000	1	0

Figure 26: Human Factors training dataset

E) Data Validation

We've calculated the cross validation score using 5 folds to test the performance of the tuned logistic Regression model.

Section 4- Machine Learning Models

According to the particular dataset, each member has selected one machine learning model to predict fire ignition and to find out the impacts of that factor on Sonoma county wildfires. In wildfire risk analysis, from those four selected factors, we will try to figure out which factor is most responsible for the event based on the performance of our integrated model. We will use 5 machine learning models on the four factors. We will be using Random Forest and Logistic Regression on fire history data and weather data; K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression on fire NDVI merged with the fire and weather history data; and Logistic Regression on human factors dataset.

A) Logistic Regression (LR)

Description:

Logistic regression is a classification algorithm used to predict the probability of a dependent target variable, which is binary in nature and there would be only two possible classes.

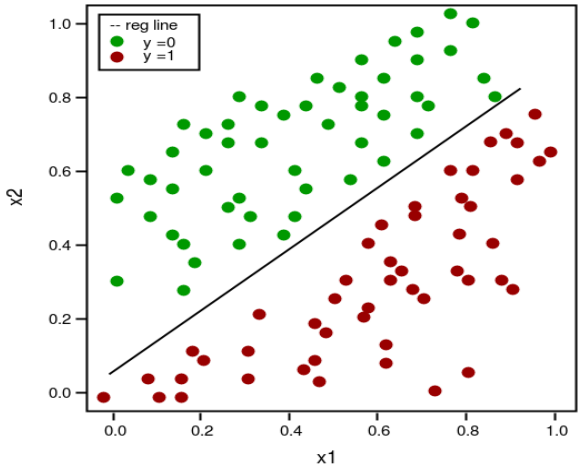


Figure 27: Binary Logistic Regression

Justification:

The goal of this project is to predict fire '1' or no fire '0'. Logistic Regression performs well when the dataset is linearly separable and it is easier to implement, interpret and very efficient to train. The dependent variables are categorical, numerical, and explanatory.

B) Random Forest (RF)

Description: Random Forest constructs a collection of decision trees. This model uses the bagging method for making this group. Random Forest finds out the foremost attribute within the subset of attributes randomly.

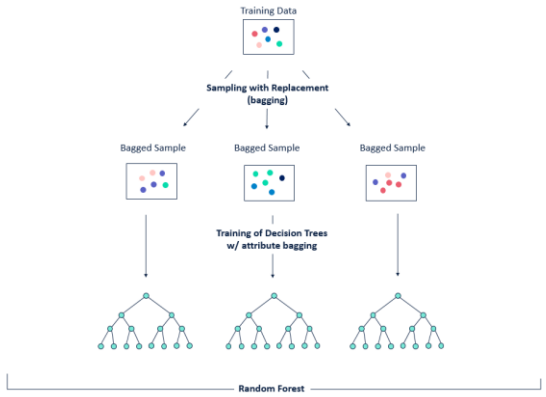


Figure 28: Random Forest Classifier Algorithm Process [22]

This model has a high tolerance for over-fitting problems if the number of trees increases. Random Forest selects pseudo-random variables for each node. As this model creates different decision trees, it does not return the same result. It returns the value which has better accuracy through the committee system. The Random Forest (RF) model integrates important variables that are used to display the effects of wildfire directly. RF is originated from bootstrap samples of training data by using random feature selection. This model calculates the prediction by combining the predictions of the ensembles. The advantage of using RF is its runtime is fast and it handles missing and imbalanced data.

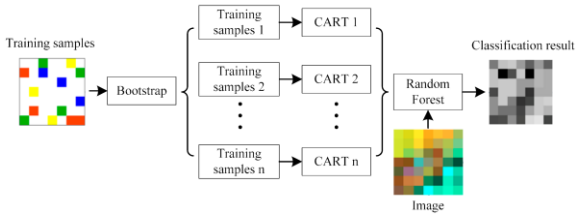


Figure 29: Schematic diagram of Random Forest [23]

Justification: The goal is to build a classifier that can accurately predict the probability of a fire based on various weather factors and the fire history. The output of the model is the binary classification of the results (1= fire event or 0 = no fire event). In this project, the Random Forest Classifier model will be used to do the wildfire risk analysis of Sonoma County, California using the final weather and fire-history dataset. The Random Forest Classifier model is the best model to do the prediction of wildfire as it is an ensemble learning method based on classification . It aggregates the votes from different decision trees to decide the final class of test variable and hence is referred to bootstrap aggregation or bagging technique. Bagging is a technique used to reduce variance of a model. It is a technique used to reduce the complexity of the models that overfit the training data. Random Forest has been applied to remote sensing data and has "demonstrated good predictive ability" for meteorological data [6]. In 2001, Breiman [6] presented the Random Forest algorithm for handling nonlinear and non-Gaussian data. Results from this study include: the AUC values for the training sample ranged from 0.740 to 0.807, which "indicates a moderate model fit based on meteorological factors" and the model accuracy of around 80% [6].

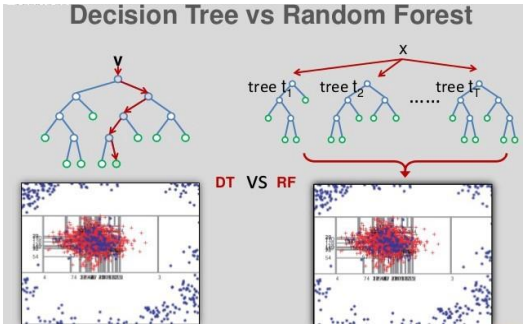


Figure 30:: Decision Tree vs. Random Forest Models [8]

C) Support Vector machine(SVM)

Support Vector Machine(SVM):

Support vector machines are a great example of supervised machine learning algorithms mostly used for regression and classification events. In SVM, we create a number of hyperplanes first and then select that hyperplane having maximum extremity from the support vector and separates data into two classes. Hyperplanes help in the classification of the data points "These planes are used for data classification, regressions and so on. It uses kernel functions to classify a data point and separates the data into its two classes with a maximum margin" [9]

"LIBSVM is an open source machine learning library where the Sequential minimal optimization (SMO) algorithm is implemented for kernelized support vector machines (SVMs) and supports classification and regression" [9]

Support Vector Machine: "SVM is a supervised machine learning model that analyzes the large amount of data. It uses classification and regression analysis to identify the patterns in the sample. In this method, we aim to find the best hyperplane that separates the dataset into two classes, as shown in Figure SVM consists of a datapoint that is a p-dimensional vector, a list of p numbers, and a (p − 1) dimensional hyperplane that separates the data points. We can derive many hyperplanes from the dataset that can classify the data. However, we have to find the hyperplane that represents the largest separation, or margin, between the two classes. This means that the hyperplane that has the maximum distance from it to the nearest data point on each side of the dataset. If such a hyperplane exists then it is known as the maximum-margin hyperplane and the linear classifier" as shown in Figure

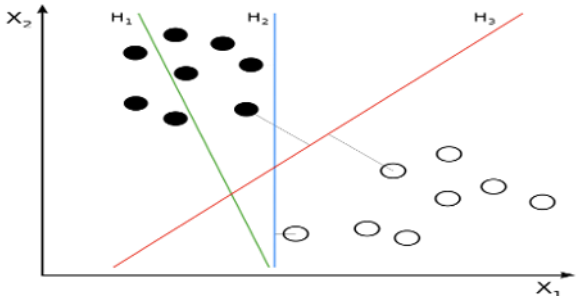


Figure 31: "H_1 does not divide the classes, H_2 does, but only with a small margin, H_3 separates them with the maximum margin. So, H_3 will be considered as the best hyperplane with maximum margin".

Justification: In this project, we use Support vector Machine to do the wildfire risk analysis, In the case of regression, margin tolerance is set in the SVM according to our problem, to minimize error, individualizing the hyperplane which has maximum number of points. SVM classifier has the best stability and accuracy compared to most of the other approaches for wildfire detection with more than 90% accuracy rate.

K-Nearest Neighbors (KNN): KNN is a supervised algorithm. This model is used for both regression and classification problems. This model calculates the interval between the vectors. The classification is done by discovering the highest repeated class from the k number of closest samples. The experiment is performed by varying the number of k (k = 3,5,7)

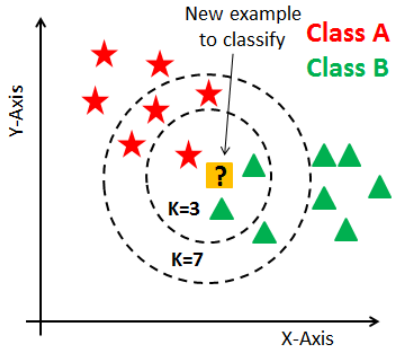


Figure32: KNN classification

Euclidean distance formula is used to calculate the distance [25]. It is also called L2 distance.

$$d(p,q) = \sqrt{\sum_{i-1}^{N} (q_i - p_i)^2}$$

Justification:
KNN is one of the very popular models for binary classification. It is faster than other models for example : support vector machine(SVM), linear regression. It is very easy to implement as it classifies the new data point by reading the whole dataset. New data can be added seamlessly which will not affect the accuracy. This model gives the flexibility to choose the distance like: euclidean distance, hamming distance, manhattan distance, minkowski distance[26].

Section 5- Machine Learning Results

Different machine learning algorithms are used per different combined datasets. This section covers the detailed graphics and results for each model. Examples of testing data and results are explained.

A) Logistic Regression (LR) on Fire history & Weather Dataset

To visualize the Receiver-Operator-Curve, the function roc_curve is used. The method returns the true positive rate (recall) and the false positive rate (probability for a false alarm) for different thresholds. This curve shows the tradeoff between recall (detect fire) and false alarm probability.

Results with Graphics:

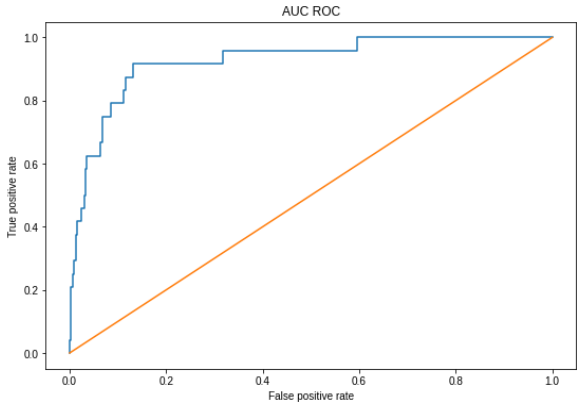


Figure 33: AUC ROC Curve

This model classifies all weather data attributes with a fire probability of 50% as fire. The training data's accuracy rate is 98.6% with AUC (Area under the ROC(Receiver-Operator-Curve))) rate 0.91. The test data's accuracy is 98.5% with AUC rate 0.92. The f1 score is 0.97.

Testing Data and Results:

```
y_train_hat = model1.predict(X_train)
y_train_hat_probs = model1.predict_proba(X_train)[:,1]
train_accuracy = accuracy_score(y_train, y_train_hat)*100
train_auc_roc = roc_auc_score(y_train, y_train_hat_probs)*100
print('Confusion matrix:\n', confusion_matrix(y_train, y_train_hat))
print('Training accuracy: %.4f %%' % train_accuracy)
print('Training AUC: %.4f %%' % train_auc_roc)
```

```
Confusion matrix:
 [[6001    0]
 [  80    0]]
Training accuracy: 98.6844 %
Training AUC: 91.4493 %
```

```
y_test_hat = model1.predict(X_test)
y_test_hat_probs = model1.predict_proba(X_test)[:,1]
test_accuracy = accuracy_score(y_test, y_test_hat)*100
test_auc_roc = roc_auc_score(y_test, y_test_hat_probs)*100
print('Confusion matrix:\n', confusion_matrix(y_test, y_test_hat))
print('Testing accuracy: %.4f %%' % test_accuracy)
print('Testing AUC: %.4f %%' % test_auc_roc)
```

```
Confusion matrix:
 [[1598    0]
 [  24    0]]
Testing accuracy: 98.5203 %
Testing AUC: 92.6210 %
```

```
print(classification_report(y_test, y_test_hat, digits=6))
```

```
              precision    recall  f1-score   support

           0   0.985203  1.000000  0.992547      1598
           1   0.000000  0.000000  0.000000        24

    accuracy                       0.985203      1622
   macro avg   0.492602  0.500000  0.496273      1622
weighted avg   0.970626  0.985203  0.977860      1622
```

Figure 34: Training and Testing Data Accuracy and AUC

When the threshold is set to 90%, the recall is 0% and the false positive rate is the same (figure 35). When the threshold is down to 10%, around 30.3% of all fire cases are detected but false positive rate doubles with 23 false alarms (figure 36).

```
y_hat_90 = (y_test_hat_probs > 0.90 )*1
print('Confusion matrix:\n', confusion_matrix(y_test, y_hat_90))
print(classification_report(y_test, y_hat_90, digits=6))
```

```
Confusion matrix:
 [[1598    0]
 [  24    0]]
              precision    recall  f1-score   support

           0   0.985203  1.000000  0.992547      1598
           1   0.000000  0.000000  0.000000        24

    accuracy                       0.985203      1622
   macro avg   0.492602  0.500000  0.496273      1622
weighted avg   0.970626  0.985203  0.977860      1622
```

Figure 35: Accuracy, recall, and f1-score with threshold 90%

```
y_hat_10 = (y_test_hat_probs > 0.10)*1
print('Confusion matrix:\n', confusion_matrix(y_test, y_hat_10))
print(classification_report(y_test, y_hat_10, digits=4))
```

```
Confusion matrix:
 [[1575   23]
 [  14   10]]
              precision    recall  f1-score   support

           0     0.9912    0.9856    0.9884      1598
           1     0.3030    0.4167    0.3509        24

    accuracy                         0.9772      1622
   macro avg     0.6471    0.7011    0.6696      1622
weighted avg     0.9810    0.9772    0.9790      1622
```

Figure 36: Accuracy, recall, and f1-score with threshold 10%

B) Random Forest (RF) on Fire history & Weather Dataset

Two random forest classifier models had been created. One is the baseline model and another is a tuned version of the baseline model. To view the performance of the model, we calculated the accuracy, AUC scores, and also generated a classification report and confusion matrix. To visualize the Receiver-Operator-Curve, the function roc_curve is used. This curve shows the tradeoff between recall (detect fire) and false alarm probability.

1) RFC Model 1: Baseline

We first fit the baseline model,which is basically a random forest classifier model with given default parameters, and ran the model to see what scores we would get. The mean AUC score was 0.698 and the accuracy score was about 99%.

```
rf = RandomForestClassifier(random_state = 42)
rf.fit(X_train, y_train)
```

```
=== Confusion Matrix ===
[[2999    1]
 [  29    2]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      3000
           1       0.67      0.06      0.12        31

    accuracy                           0.99      3031
   macro avg       0.83      0.53      0.56      3031
weighted avg       0.99      0.99      0.99      3031


=== Accuracy Score ===
0.9901022764764105


=== All AUC Scores ===
[0.80671806 0.62960553 0.60187225 0.62565078 0.83375551 0.84124469
 0.68832599 0.68667451 0.58838104 0.67930088]


=== Mean AUC Score ===
Mean AUC Score - Random Forest:  0.6982128554265118
=== Standard Dev AUC Score ===
Standard Dev - Random Forest:  0.09082132466646084
```

Figure 37: Performance Metrics for base model

We then plotted the ROC curve for this baseline model.

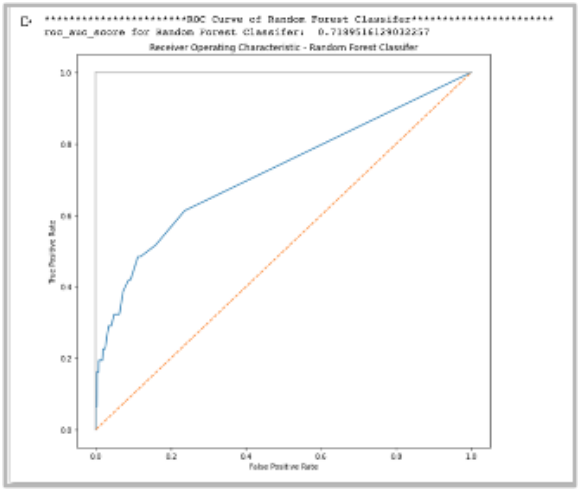


Figure 38: ROC-AUC curve for base model

We also visualized the baseline random forest classifier model.

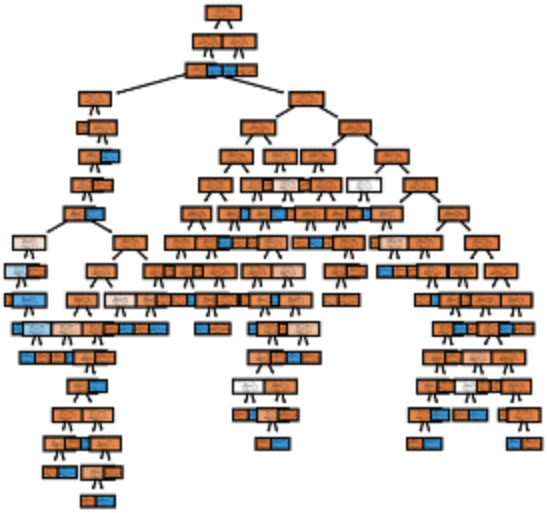


Figure 39: RFC Model 1- Baseline

Next, we did a randomized CV search for parameters that would yield better AUC scores. The best parameters suggested by the search include, n_estimators = 1600 and max_depth of forest = 10.

```
rf_random = RandomizedSearchCV(estimator=rf, param_distributions=random_grid,
                               n_iter = 100, scoring='neg_mean_absolute_error',
                               cv = 3, verbose=2, random_state=42, n_jobs=-1,
                               return_train_score=True)
```

```
rf_random.fit(X_train, y_train);

# Select best params out of all in search
rf_random.best_params_
```

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  37 tasks      | elapsed:  3.0min
[Parallel(n_jobs=-1)]: Done 158 tasks      | elapsed: 12.0min
[Parallel(n_jobs=-1)]: Done 300 out of 300 | elapsed: 22.1min finished
{'bootstrap': True,
 'max_depth': 10,
 'max_features': 'sqrt',
 'min_samples_leaf': 4,
 'min_samples_split': 2,
 'n_estimators': 1600}
```

Max_depth has been adjusted to 10; you can see results of RFC in diagram to the right

2) RFC Model 2: Randomized CV search

```
rf = RandomForestClassifier(n_estimators=1600, min_samples_split = 2, max_depth = 10,
                            min_samples_leaf = 4,  max_features =  'sqrt', random_state = 42)
rf.fit(X_train, y_train)
```

We tuned the baseline random forest classifier model with the suggested best parameters from the randomized CV search. We fit and ran this new, second model to see whether our model performance improved or not. The mean AUC score jumped to 0.790; this definitely shows an improvement in performance compared to the baseline score.

```
=== Confusion Matrix ===
[[3000    0]
 [  31    0]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      3000
           1       0.00      0.00      0.00        31

    accuracy                           0.99      3031
   macro avg       0.49      0.50      0.50      3031
weighted avg       0.98      0.99      0.98      3031


=== Accuracy Score ===
0.9897723523589574


=== All AUC Scores ===
[0.86223468 0.65448538 0.65398478 0.67671205 0.97213656 0.98403084
 0.91740088 0.90302863 0.63160793 0.64614537]


=== Mean AUC Score ===
Mean AUC Score - Random Forest:  0.7901767120544653
=== Standard Dev AUC Score ===
Standard Dev - Random Forest:  0.14160734384219184
```

Figure 40: Performance Metrics for randomized CV search model

We then plotted the ROC curve for this tune model.

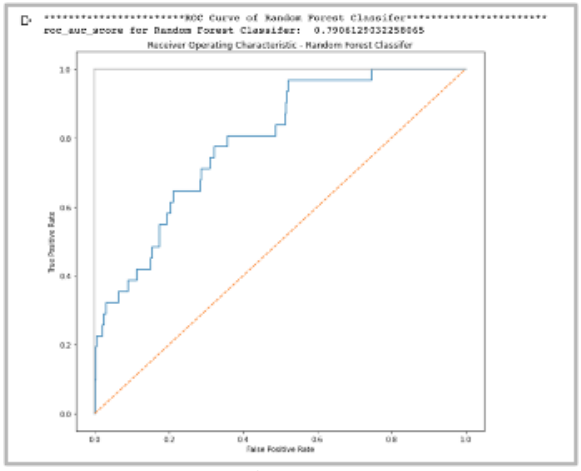


Figure 41: ROC-AUC curve for randomized CV search model

Lastly, we visualized the tuned random forest classifier model. Merely looking at figure 40, we can see the tuned random forest classifier model visualization looks less cluttered and much easier to interpret compared to the baseline random forest classifier model visualization shown in figure 41 above.

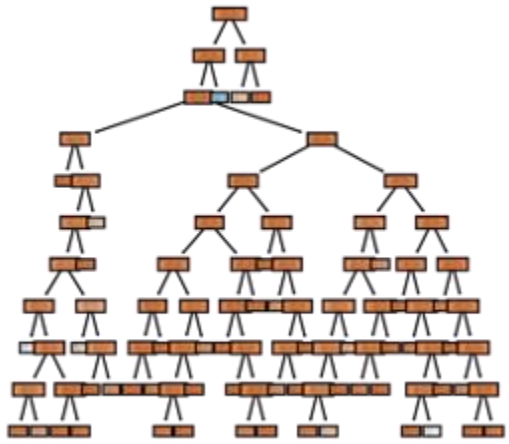


Figure 42: RFC Model 2- Randomized CV Search

Both randomized CV search and grid search were used to find optimized parameters that would provide the best results. The model performance as a result of using randomized CV search is only shown because it yielded a much better performance score than the model performance based on the grid search. The metric used in measuring the performance of the model was the mean AUC score instead of accuracy because accuracy can be a misleading metric when class imbalance is present in the dataset, which it is.

C) Support Vector Machine(SVM) on Fire history and weather dataset merged with NDVI Dataset:

Training and Testing Dataset

```
from sklearn.model_selection import train_test_split
x,y = data.loc[:,data.columns != 'Target'], data.loc[:,'Target']
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=0)
```

Support Vector Machine

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(x_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

```
y_pred = svclassifier.predict(x_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[80  0]
 [ 1  1]]
              precision    recall  f1-score   support

           0       0.99      1.00      0.99        80
           1       1.00      0.50      0.67         2

    accuracy                           0.99        82
   macro avg       0.99      0.75      0.83        82
weighted avg       0.99      0.99      0.99        82
```

Figure 42: Support Vector Machine

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='poly', degree=8)
svclassifier.fit(x_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=8, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

```
y_pred = svclassifier.predict(x_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[80  0]
 [ 2  0]]
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        80
           1       0.00      0.00      0.00         2

    accuracy                           0.98        82
   macro avg       0.49      0.50      0.49        82
weighted avg       0.95      0.98      0.96        82
```

```
/Users/sukritimishra/opt/anaconda3/envs/ml/lib/python3.7/site-packages/sklearn/m
```

Figure 43: SVM with non-linear(polynomial) activation unit.

D) K-Nearest Neighbor(KNN) on Fire history and weather data merged with NDVI Dataset:

```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
accuracy = accuracy_score(y_test,prediction)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.926829268292683
```

Figure 44: KNN model

E) Logistic Regression (LR) on Fire history & NDVI Dataset:

```
accuracy = accuracy_score(y_test,predictions)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.9512195121951219
```

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        77
           1       1.00      0.20      0.33         5

   micro avg       0.95      0.95      0.95        82
   macro avg       0.98      0.60      0.65        82
weighted avg       0.95      0.95      0.94        82
```

Figure 45: Logistic Regression for NDVI

F) Logistic Regression Human Factor Dataset

Logistic regression was the selected model for the human factors. LR model was first implemented then Grid Search CV was used to improve the model accuracy. We've calculated the accuracy to measure the model performance and also used classification reports to measure the quality of the prediction. We then plotted the Receiver Operator Curve (ROC) to show the tradeoff between true positive rate (recall) and false positive rate measured the Area Under the Curve (AUC).

1) Model 1- Logistic regression

We fit the logistic regression model and got accuracy of 84% and AUC of 0.85. Although this model performs well, we can see that the true negative is 0 from the confusion matrix, meaning the model is not detecting the no fire (0) correctly.

```
Confusion Matrix
[[356   1]
 [ 68   0]]

 Classification Report
              precision    recall  f1-score   support

           0       0.84      1.00      0.91       357
           1       0.00      0.00      0.00        68

    accuracy                           0.84       425
   macro avg       0.42      0.50      0.46       425
weighted avg       0.71      0.84      0.77       425
```

Figure 46: Performance Metrics for logistic regression

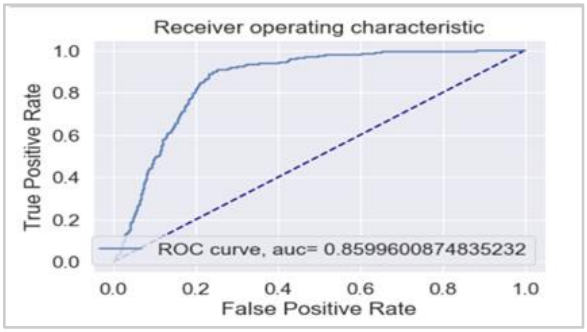


Figure 47: ROC & AUC for logistic regression

2)Model 2- GridSearchCV

We tried the GridSearchCV to select the best parameters to improve the performance of the LR model. The best parameters suggested were C=166.8 & penalty: L2

```
print("Best Logistic Regression Parameters: {}".format(logreg_cv.best_params_))
print(f'Best score/accuracy is {logreg_cv.best_score_:.2f}')

Best Logistic Regression Parameters: {'C': 166.81005372000593, 'penalty': 'l2'}
Best score/accuracy is 0.86

# feeding best parameters to the model
logreg2=LogisticRegression(C=best_params.get('C'),penalty=best_params.get('penalty'))
logreg2.fit(X_train,y_train)
print(f'Model with best params- accuracy score {logreg2.score(X_test,y_test):.2f}')

Model with best params- accuracy score 0.84
```

Figure 48: Best parameters for GridSearchCV

We then tuned the logistic regression model with the suggested parameters from the GridSearchCV. After we fit the model, we found out that the accuracy remained the same 84% with a slight increase in AUC, 0.86. We calculated the confusion matrix and noticed that the true negative has increased from 0 to 21. This shows an improvement in the classification.

```
Confusion Matrix
[[335  22]
 [ 47  21]]

 Classification Report
              precision    recall  f1-score   support

           0       0.88      0.94      0.91       357
           1       0.49      0.31      0.38        68

    accuracy                           0.84       425
   macro avg       0.68      0.62      0.64       425
weighted avg       0.81      0.84      0.82       425
```

Figure 49: Performance Metrics for GridSearchCV - LR

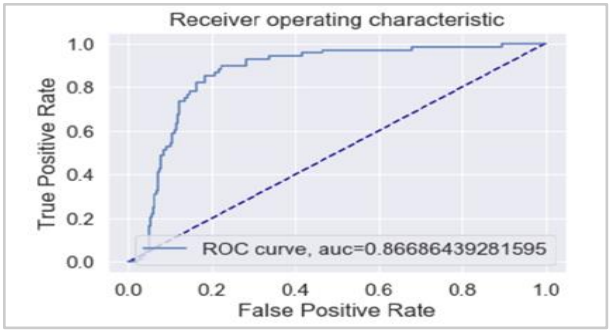


Figure 50: ROC & AUC for GridSearchCV - LR

We've also calculated the cross-entropy to evaluate the quality of the classification and the result was 0.31. Lower cross entropy indicates that the model is performing well.

```
y_pred = logreg2.predict_proba(X_test)[::, 1].ravel()
loss = log_loss(y_test, y_pred)
print('Log Loss / Cross Entropy = {:.4f}'.format(loss))

Log Loss / Cross Entropy = 0.3149
```

Figure 51: cross entropy

Section 6- A Comparison of Results Among Different Models

Models for Fire History and Weather	**Accuracy**	**F1-Score**
Logistic regression (Threshold 90%)	97.1%	97.8%
Logistic regression (Threshold 10%)	98.1%	97.9%

Table 2: Logistic Regression (LR) on Fire history & Weather Dataset

Model for Fire History and Weather	**Accuracy**	**Mean AUC**	**Standard Deviation AUC**
RFC Model 1 Baseline	99.01%	0.6982	0.0908
RFC Model 2: After Randomized Search CV	98.97%	0.7902	0.1416
RFC Model 3: After Grid Search	98.97%	0.7174	0.10417

Table 3: Random Forest Classifier (RFC) on Fire history & Weather Dataset

Models for NDVI	**Accuracy**	**F1-Score**
Logistic regression	95%	94%
Support Vector Machine (kernel: polynomial, Gaussian and sigmoid; degree=8)	98%	96%
KNN (K=3)	92%	90%

Table 4: Logistic Regression, Support Vector Machine, K-Nearest Neighbor on Fire history & NDVI Dataset

Models for Human Factors	**Accuracy**	**AUC**

Logistic regression	84%	0.85
LG Grid Search CV	84%	0.86

Table 5: Logistic Regression (LR) on Fire history & Human Factor Dataset

Section 7- Experience and Lessons Learned

Fire history dataset was easily found, preprocessed, and combined with a weather dataset using ArcGIS based on the date and the location of weather stations. Binary logistic regression was used for fire history and weather datasets. It gives above 98% accuracy rate on training/ testing data, and logistic regression threshold 90%. From this project, I learned that a good quality of datasets with properly feature- engineered machine learning algorithms gives a high accuracy rate.

The weather dataset required a lot of pre-processing steps such as handling missing values and merging and aggregating data from different weather stations across Sonoma county, and in the end, merging this weather dataset with the pre-processed fire-history dataset. The Random Forest Classifier model was the second model applied to this fire-history weather dataset. The randomized CV search parameters provided the best mean AUC score, a score of 0.790, and hence yielded the best performance results for the random forest classifier model when applied to the fire-history weather dataset. From this project, I learned that the data collection and preprocessing step, although takes the most time, is the most important step in obtaining good quality results. I also learned that hypertuning the model makes a rather big difference in the performance of the model and that the more research that goes into finding the optimal parameters, the higher chance we will find a model that will provide the most accurate results.

In wildfire risk prediction projects, the main difficult task is data collection. We spent more than 80% time on data collection and data preprocessing. We experienced that performance of any model depends on the quality of the dataset. Being a Data Analyst, we can say that for ML projects apart from data collection, the selection of ML algorithms is also a very important task. Every problem is not related to deep learning models. So, we have to choose our models very wisely. First we implemented linear and logistic regression models and then moved towards SVM, KNN and Random Forest. In the end, we would like to express our sincere thanks and gratitude to Prof. Jerry Gao who has supported us through with his patience and knowledge. We are also grateful to him for teaching us and playing a key role in generating interest in this field and providing us with guidance and motivation whenever needed. We believe we were able to fulfill most of our objectives and we did learn a lot during the process. Based on the conglomerate data, we would like to create and implement ensemble models in the future.

References:

[1] "Wildfires and Climate Change," Center for Climate and Energy Solutions, 07-Feb-2020. [Online]. Available: https://www.c2es.org/content/wildfires-and-climate-change/. [Accessed: 13-Mar-2020].

[2] C. Kousk, K. Greig, B. Lingle, and H. Kunreuther, "WILDFIRE COSTS IN CALIFORNIA: THE ROLE OF ELECTRIC UTILITIES1," Wharton Center for Risk Management and Disease Processes Center, Aug-2018. [Online]. Available: https://riskcenter.wharton.upenn.edu/wp-content/uploads/2018/08/Wildfire-Cost-in-CA-Role-of-Utilities-1.pdf. [Accessed: 11-Mar-2020].

[3] "Kincade Fire," Wikipedia, 11-Jan-2020. [Online]. Available: https://en.wikipedia.org/wiki/Kincade_Fire. [Accessed: 14-Mar-2020].

[4] R. Dutta, A. Das and J. Aryal, "Big data integration shows Australian bush-fire frequency is increasing significantly," Royal Society Open Science, vol. 3, no. 2, p. 150241, 2016.

[5] P. L. Andrews, "Fire danger rating and fire behavior prediction in the United States," In: Proceedings of Fifth NRIFD Symposium: International Symposium on Forest Fire Protection; 2005 November 30-December 2; Mitaka, Tokyo, Japan. Tokyo, Japan: National Research Institute of Fire and Disaster. p. 106-117., 01-Jan-1970. [Online]. Available: https://www.fs.usda.gov/treesearch/pubs/46553. [Accessed: 14-Mar-2020].

[6] L. Breiman, "Random Forests"Machine Learning 45 (1), 5–32, 2001. [Online]. Available: https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf. [Accessed: 14-Mar-2020].

[7] Y. O. Sayad, H. Mousannif, and H. A. Moatassime, "Predictive modeling of wildfires: A new dataset and machine learning approach," Fire Safety Journal, 24-Jan-2019. [Online]. Available:

https://www.sciencedirect.com/science/article/pii/S0379711218303941. [Accessed: 15-Mar-2020].

[8] P. Ma, "Galactic Structures and Random Forests," Medium, 28-Dec-2018. [Online]. Available: https://medium.com/datadriveninvestor/galactic-structures-and-random-forests-d16f6b61a1a7. [Accessed: 15-Mar-2020].

[9] S. Vigneshwaran , S. S. Shanthakumari and V. Ranganathan, "Fire Detection Using Support Vector Machines (SVM)," International Journal of Science and Research (IJSR), Vols. ISSN (Online): 2319-7064, p. 12, 2015.

[10] "CA Geographic Boundaries," California Open Data, 23-Oct-2019. [Online]. Available: https://data.ca.gov/dataset/ca-geographic-boundaries. [Accessed: 15-Mar-2020].

[11] Marcos Rodrigues, Juan de la Riva, An insight into machine-learning algorithms to model human-caused wildfire occurrence, Environmental Modelling & Software 57 (2014) 192-201

[12] Introduction to Boosted Trees. (2017, March 24). Retrieved March 14, 2020, from https://blog.bigml.com/2017/03/14/introduction-to-boosted-trees/

[13] "Gini Index For Decision Trees," QuantInsti, 05-Mar-2020. [Online]. Available: https://blog.quantinsti.com/gini-index/. [Accessed: 20-Mar-2020].

[14] A. Navlani, "Decision Tree Classification in Python," DataCamp Community, 28-Dec-2018. [Online]. Available: https://www.datacamp.com/community/tutorials/decision-tree-classification-python. [Accessed: 20-Mar-2020].

[15] "Seeing the Forest for the Trees: An Introduction to Random Forest" Alteryx Community, 08-Mar-2019. [Online]. Available: https://community.alteryx.com/t5/Alteryx-Designer-Knowledge-Base/Seeing-the-Forest-for-the-Trees-An-Introduction-to-Random-Forest/ta-p/158062. [Accessed: 20-Mar-2020]

[16]Feng, Q.; Liu, J.; Gong, J. "UAV Remote Sensing for Urban Vegetation Mapping Using Random Forest and Texture Analysis." 19-Jan-2015. [Online]. Available: https://www.mdpi.com/2072-4292/7/1/1074.
[Accessed: 20-Mar-2020]
*[17]*pyimagesearch."K-Nearest Neighbor Classification". RetrievedFrom: *https://gurus.pyimagesearch.com/lesson-sample-k-nearest-neighbor-classification/*

[18] Genesis (September 25, 2018). "*Pros and Cons of K-Nearest Neighbors*".
RetrievedFrom: *https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/*