UNIVERSITY OF CALIFORNIA, SAN DIEGO

Analyzing and Predicting Wildfires

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Sumedha Khatter

Committee in charge:

Ilkay Altintas, Chair
Arun Kumar
Julian McAuley

2017

The thesis of Sumedha Khatter is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

Chair

University of California, San Diego

2017

DEDICATION

To my beloved father Sh. Harish Chander, mother Mrs. Veena Rani and my brother Kapil, from whom I get continuous support.

EPIGRAPH

I sought wisdom from Him,
He gave me problems to solve.
–snippet from a Hindi Prayer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to Ilkay Altintas for providing me this opportunity. Thanks to Ilkay Altintas, Mai Nguyen, Daniel Crawl, Jessica Block and Tomàs Artés Vivancos for all help, support, weekly meetings and feedback. Special thanks to Ilkay Altintas, Mai Nguyen and Arun Kumar for providing an intensive review on the writing of thesis.

VITA

2012	B.Tech. in Information Technology, National Institute of Technology, Kurukshetra
2013	Awarded Gold Medal for being Rank 1 holder in B. Tech. in the presence of the President of India Sh. Pranab Mukherjee, Governor of Haryana State and many other dignitaries on Golden Jubilee celebration at NIT, Kurukshetra
2012-2015	Software Engineer, The Royal Bank of Scotland, R&D Gurgaon
2017	M.S. in Computer Science, University of California, San Diego

ABSTRACT OF THE THESIS

Analyzing and Predicting Wildfires

by

Sumedha Khatter
Master of Science in Computer Science

University of California San Diego, 2017

Ilkay Altintas, Chair

Wildfires can rip through any part of the world and cause a havoc. Building technologies to analyze and predicting the occurrence of the hazardous wildfires is of paramount importance. The project brings a new approach to build a system which analyzes and find areas more susceptible to wildfires using historical data of fire and landscape features. This is achieved by preparing dataset using raw data downloaded from various sources, feeding that into various machine learning models and then predicting and analyzing the results. The skewness faced by the dataset is resolved using different resampling techniques and gauged via different scores. Use of certain kind of scores for this kind of dataset and comparison of results obtained via various supervised learning models is discussed next in the project.

Chapter 1

Introduction

Wildfires are uncontrolled flames which burn hectares of land and destroy everything in their way and are charged by dry hot conditions, dry fuel or human factors.

As per National Geographic, an average of more than 100,000 wildfires clear 4 million to 5 million acres of land in United States every year. In recent years wildfires have burnt up to 9 million acres of land. A wildfire can move as fast as 14 miles per hour, consuming everything - trees bushes, homes, even humans- in its path (National Geographic- Wildfires [37]).

This is just the statistics of United States. The victims of wildfires are spanned all across the world. Last year in 2016, there were major forest fires in Uttrakhand (India), Fort McMurray Wildfires (Canada), mainland Portugal and Madeira archipelago (Portugal) and many more which not only prompted the evacuation of thousands of people, but also caused a lot of destruction consuming homes, time and most precious the life of beings.

Saying this, it can be understood that wildfires can cause potential losses and hence it is significant to build technologies around its prevention. A lot of systems have already been built (as discussed in Section 1.4) and each carries their own significance. A new system with a notion to predict the occurrence of the wildfires is built in this project which focuses majorly on landscape features. This system is fed with historical data and will require current data to predict wildfires at a particular pixel in an area. Apart from predicting the occurrence of Wildfires, the project also talks about the performance and accuracy of various resampling techniques and machine learning models which are used to build the system.

1.1 The Problem Statement

Knowing the gravity of the hazardous effects caused by Wildfires, it is unswervingly essential to build a system that can predict its occurrence. The project builds a model using a new approach to predict the occurrence of Wildfires by utilizing the historical data of Landscape features and Fire history. Also the dataset faces class imbalance problems, various techniques are applied to solve it and achieve better scores.

1.2 Significance of the Problem

The causes of wildfire varies throughout the world. The wildfires occur all across the world with different intensities. The system built in this project can help us in dealing with hazards of wildfires. The prediction made by the system would render information prior to the occurrence of fire and issue warnings where fires are most likely to occur.

The project is a big prospective to solve and predict the wildfires. Adding more features and historical data into the model can turn this into a more strong, reliable and robust system. This is just a start, there is a long journey which can be envisioned.

1.3 Contributions of the Project

The main contributions of this project are:

The first is preparation of dataset using raw data collected from two different sources. The raw data from these two sources is mapped in different geographical references. Hence first data from both the sources is projected in same reference and then a series of gdal (python tools to read and write geo-spatial data) utilities are applied to convert it to a dataset.

The second is why we cannot use Accuracy, F1 and Precision scores for the dataset used in the project.

The third contribution is the evaluation of various resampling techniques applied to solve the class imbalance problem faced by this dataset. The dataset used in the project is pretty big (4665108 * 6) and skewed (one class is 98.5% and the other is 1.49%). Thus various resampling techniques are applied to comprehend this situation. The scores are compared and gauged after applying various

techniques.

The fourth contribution of this project is the application of model on dataset which predicts the fire occurrence of a pixel. The model has a potential to determine the likelihood of wildfires in an area. The prediction of wildfires of a given grid can be calculated, and can be used to analyze the trends and factors influencing wildfires the most.

1.4 Related Work

Since forest fires are one of the havoc causing environmental issue, researched all over the world try one way or the other to do something on the lines to prevent the destruction. Paulo Cortez and Anibal Morais[40] explored a data mining approach to predict the burned areas of forest fires using various techniques and a feature setup of spatial, temporal, FWI components and weather attributes. Joe H. Scott and Robert E. Burgman[26] described the use of fire behavior fuel model in Rothermal[44] surface spread fire model. FlamMap [13] is a program under US Forest Service that analyzes and computes potential fire behavior characteristics such as spread rate, flame length, fireline intensity etc. Fire Area Simulator FARSITE [12] is another program by US Forest Service that computes growth and behavior of wildfire for long periods under heterogeneous conditions of fuel, weather, ground. Canadian Wildland Fire Information System (Canadian Wildland Fire Information System[8]) has a Canadian Forest Fire Weather Index (FWI) System that calculates its six components based on the observations of temperatures, relative humidity, wind speed, and 24 hour rainfall. These six components account for the effects of fuel moisture and wind on fire behavior.

Thus related work provided intuition behind how fire prevention, prediction or behavior systems could be built. The project is however altogether a new approach and new concept and utilizes new set of features and a synthetic feature to analyze and predict the wildfires.

Chapter 2

Understanding Geographic Information System (GIS)

Before deep diving into the project it is squarely important to understand the basic concepts used in it. The system is built using the historical data of Landscape and Fire Perimeters which are in different forms of geographic data. And to view or process this data, various Geographical Information System (GIS) tools and libraries in python are used. Though there are books and long manuals for all of these; here the concepts are discussed in brief so as to make rest of the thesis clear.

2.1 Introducing GIS

A Geographic Information System (GIS) is a computer system for capturing, storing, checking, and displaying data related to positions on Earth's surface. GIS can show many different kinds of data on one map. This enables people to more easily see, analyze, and understand patterns and relationships (National Geographic, Geographic Information System[36]).

A GIS consists of the following components (QGIS Documentation[42]):

1. Hardware: The resources on which GIS operates is the hardware.
2. Software: This provides tools to store, process and represent the geographic information. Eg- QGIS, ArcMap are the most widely used applications.
3. Data: This data contains geographic features, attribute information which is digitized into the application.

The GIS application can then be used to view the data, create the new spatial information and do spatial analysis on it.

2.1.1 GIS Software

GIS applications have rich user interface to view, create, customize and analyze the data. There are various tools provided to work on geometry, data management, geo-processing and conversion of data from one form to another. Not only this, there are plugins available to perform some unique functions. These plugins are embedded in the application. For this project, Count Overlapping plugin is used to calculate cumulative frequency of fire. The use of Count Overlapping plugin is discussed in more detail in Chapter 3.

A brief about the two of the most used GIS applications is written below:

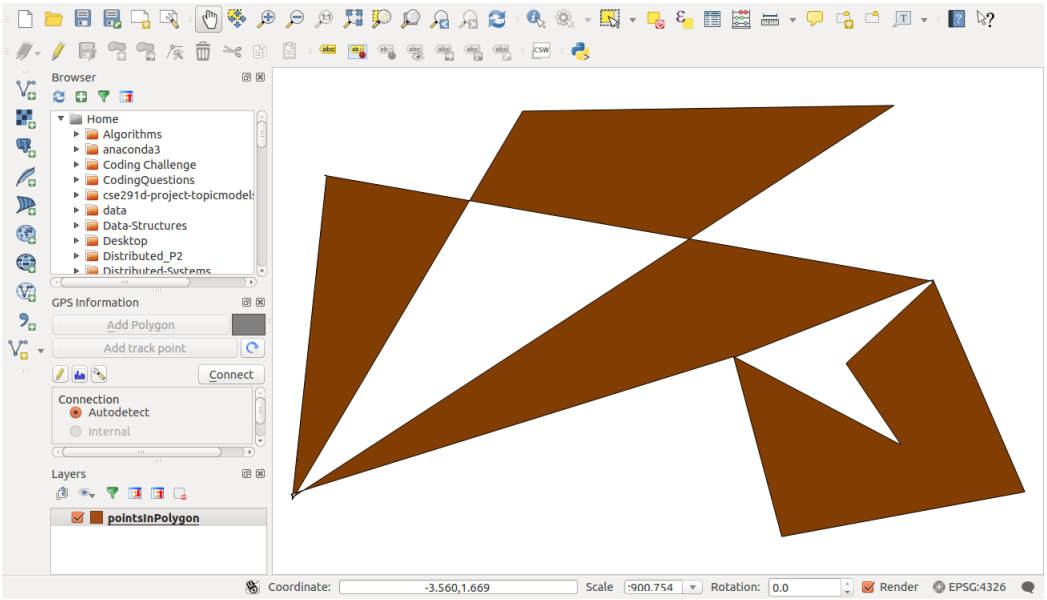

Figure 2.1: QGIS UI displaying a data layer in the View Panel.

1. QGIS: QGIS (previously known as Quantum GIS) is an application available for Windows, Mac, Linux, BSD and Android(beta version). It is free and is a part of open-source software community. Figure 2.1 captures the QGIS UI and a data layer displayed in the view panel.

2. ArcMap: It is a part of ArcGIS and is used mostly to view, edit and analyze the data. It is a commercial application and is available for various versions of Windows.

2.2 GIS Data

This data links a non-geographical information to a geographical information. A plain example can be an association of a particular ‘FuelType‘ at a specific

'Latitude' and 'Longitude' captured on a particular 'Date'. The Table 2.1 below shows this example.

Table 2.1: GIS Data instance of a Fuel type at a specific location on a particular date.

Latitude	Longitude	FuelType	Date
26.87043	-31.85	Shrub	07/28/1990

The geographical component is stored in 'Latitude' and 'Longitude' and the non-geographical content is stored in 'FuelType' and 'Date'.

GIS Systems work with different types of data. Most common are:

- Vector data
- Raster data

2.3 Vector Data

Vector data is used to represent real world entities. Consider an image represented in Figure 2.2. This image consists of a lot of entities such as houses, roads, water. These entities when represented in terms of vertices and paths, is called as Vector Data. The representation of entities can be in the form of points, lines or polygons. The features to describe the vector data are called as attributes. The vector data carries information about the spatial extent (maximum and minimum of X and Y coordinates) and the geographical reference system. The three different forms in which vector data can be represented are described below-

1. Point: It is used to describe discrete entities which are generally non-adjacent. Figure 2.3 represents a 'point' layer created to represent three houses of the landscape image (Figure 2.2). Hence there are three features in this layer. Figure 2.4 shows the point layer without landscape image and the attributes table of this layer. This layer has two attributes- 'id' and 'houseColor' and stores data of each feature using the two attributes.

2. Line: When the data cannot be merely represented by a single vertex and has different start and end points, 'line' form is used to represent the vector data. Figure 2.5 shows a vector layer of type 'line' created to represent the road of the landscape image. Figure 2.6 shows the vector layer solely to represent the road and its attribute table. The attribute 'RoadType' stores information about the type of road.

Figure 2.2: Real entities such as houses, roads, water bodies in this image can be represented in the form of vector data. This image was captured at Point Loma, San Diego.

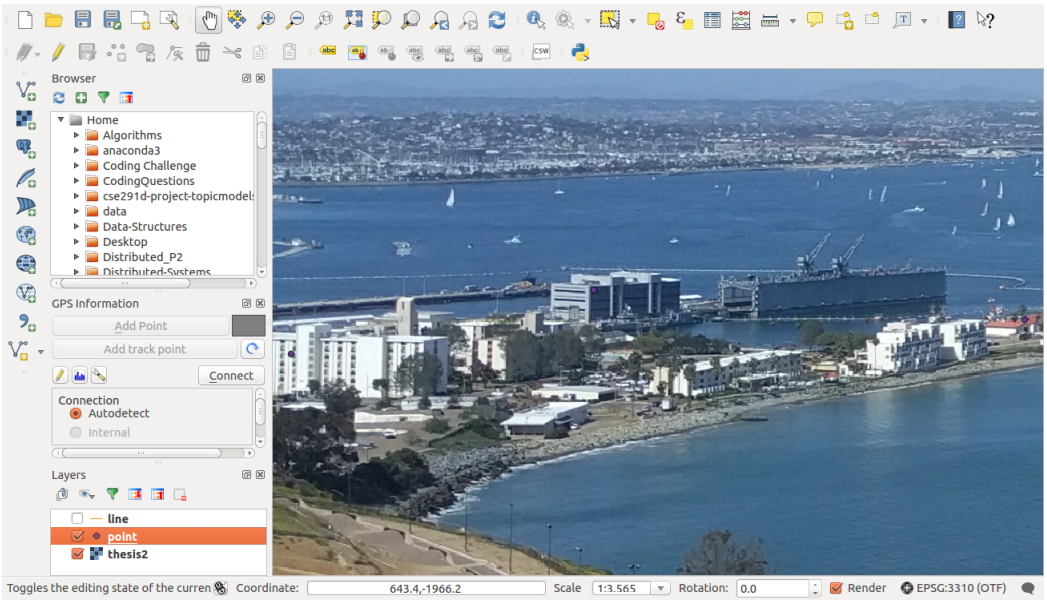

Figure 2.3: A vector layer of type 'point' created to represent the houses of the landscape image.

3. Polygon: When data to be represented is an enclosed shape, vector layer of type 'polygon' is used. The first and last vertex of this layer are same as it is an enclosed area. Figure 2.7 shows a vector layer of type 'polygon' created to represent the water body of the landscape image. Figure 2.8 shows the same vector layer solely with its attribute table. The attribute depth stores the information about the depth of this polygon.

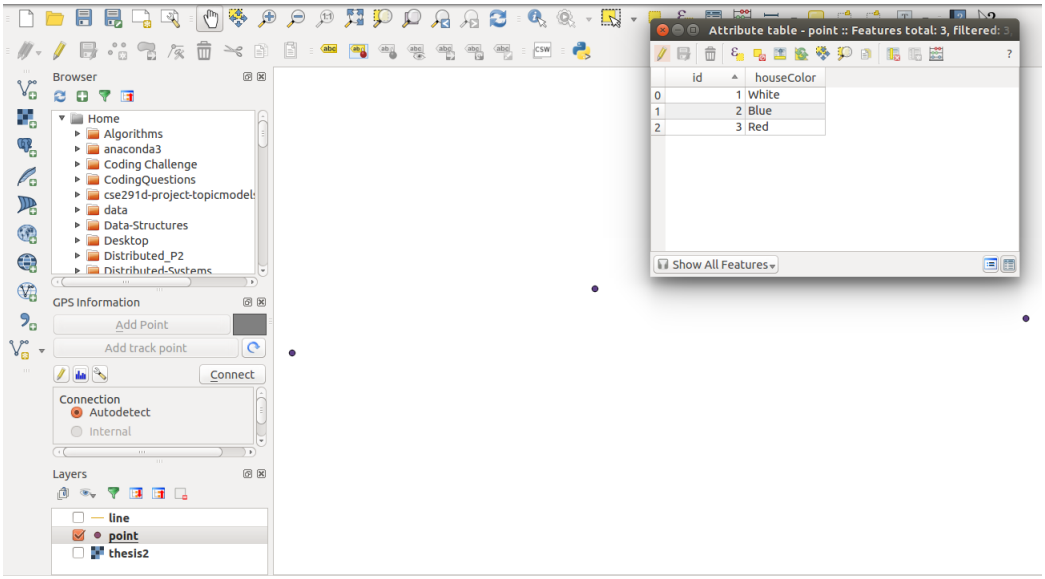

Figure 2.4: A vector layer of type 'point' with its attributes table to represent the houses of the landscape image.

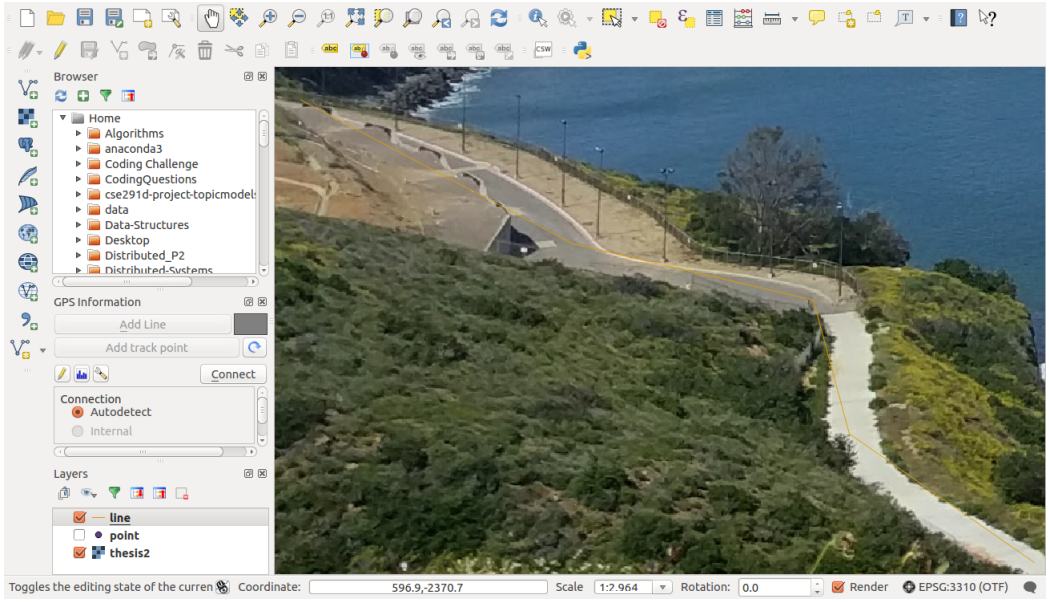

Figure 2.5: A vector layer of type 'line' created to represent the road of the landscape image.

2.4 Storage of Vector Data

2.4.1 Shapefile

The vector data contains the geometry as well as the attributes of entities. It is mostly stored in the form of a 'Shapefile'. The Shapefile further stores data in six different files having different formats. Whenever a shapefile is used, all the files are required. One without the other does not work. Table 2.2 shows the extensions

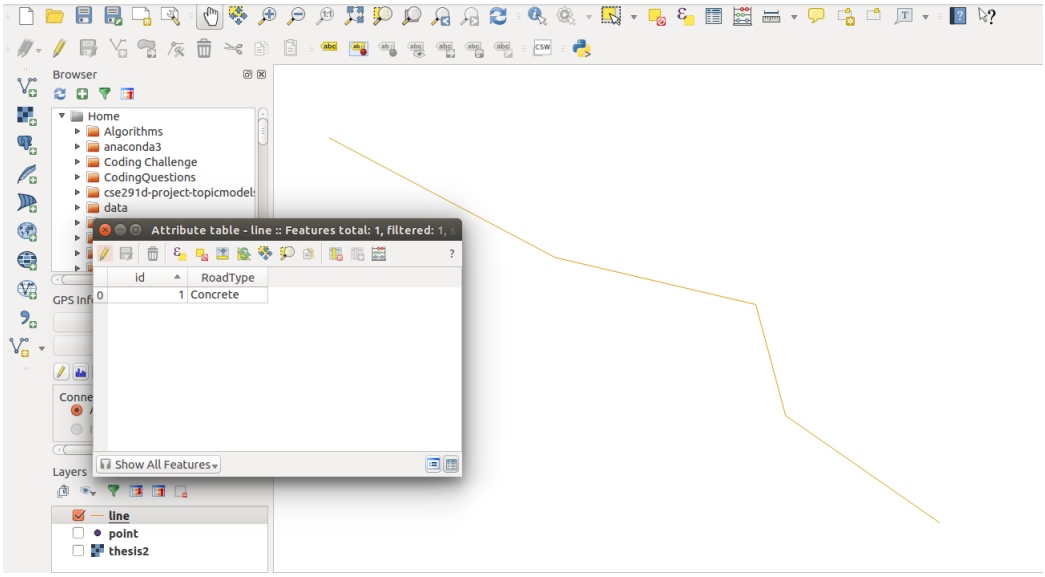

Figure 2.6: The image shows the vector layer of type 'point' and its attribute table to represent the road of the landscape image.

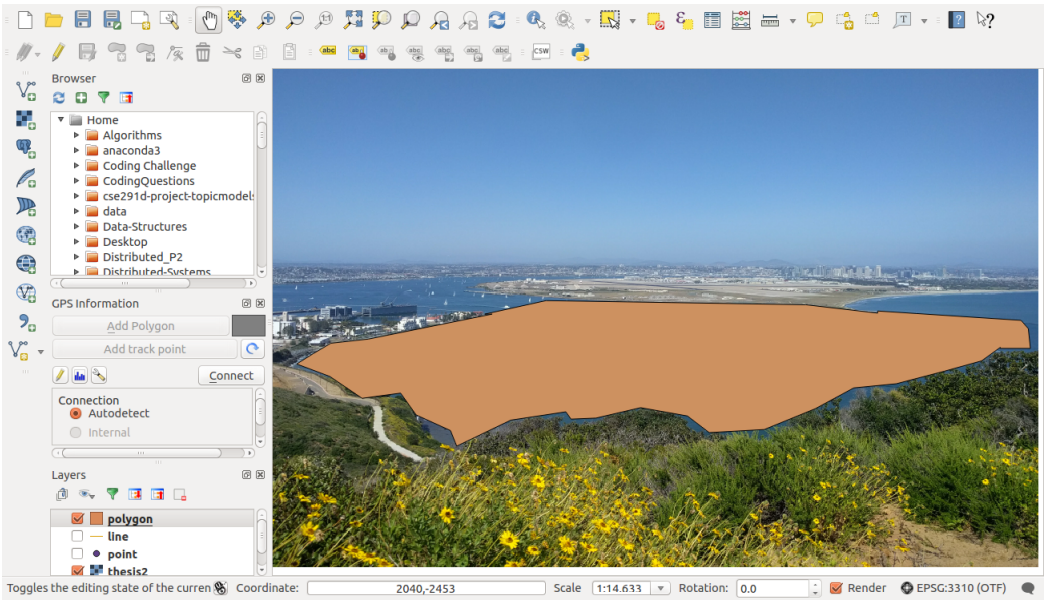

Figure 2.7: A vector layer of type 'polygon' created to represent the water body of the landscape image.

of files mapped to their descriptions.

Table 2.2: Table showing different extensions of a shapefile

File Extension	Description
.shp	Stores the geometry of vector data.
.dbf	Stores the attributes of vector data.
.shx	Stores index to find features easily.
.cpg	Stores the code page to identify the character set.
.prj	Stores the projection information.
.qpj	Stores the project created in QGIS.

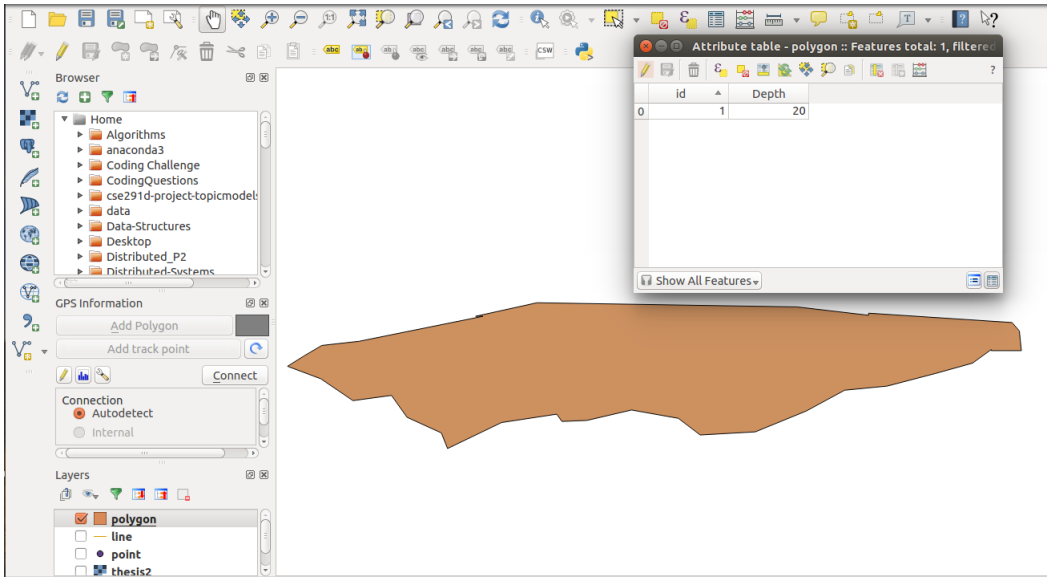

Figure 2.8: A vector layer of type 'polygon' with its attribute table.

2.4.2 File Geodatabase

The geodatabase is a container to hold a collection of dataset. One of geodatabase is File Geodatabase (ArcGis Geodatabases[3]). A file geodatabase is a collection of files in the folder on the disk that can store, query and manage both spatial and non-spatial data. The folder is stored with extension .gdb (ArcGis - File geodatabase[2]). Also, the database folder can store multiple vector files. One of the data downloaded from sources used in this project is in the form of file geodatabase.

2.5 Raster Data

A raster consists of data arranged in a grid (or matrix) of rows and columns. The grid contains pixels (or cells) which stores information about the geographical area (ArcGis- Raster Data [4]).

Why is raster data required when vector data can store the geographical data? Vector data stores the homogeneous data. For example, consider the case of 'polygon' type vector data described above. The polygon stores information about the depth of water body. But the problem is, there is a single value (20) of the depth for whole water body. What if there are different values of depth at different points in water body and not a single one? This problem is solved using raster data which can keep different values of depth in a grid per pixel. The explanation above is similar to what is explained in QGIS Raster Data documentation[43].

Figure 2.9 displays a part of grid how raster data can be used to store the data more accurately by keeping heterogeneous values per cell.

Spatial resolution of a raster image is defined using its cell size. Thus if a cell size is 5 * 5, the resolution is 5 meters. The smaller cell size gives more detail about the area. The dataset prepared in this project uses a cell size of 30 * 30.

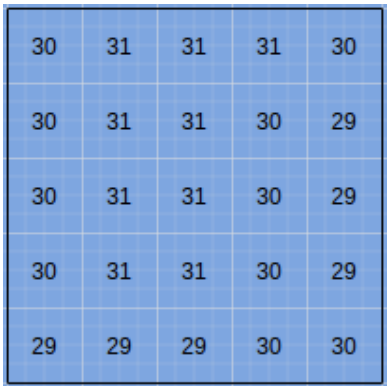

Figure 2.9: A part of grid displaying heterogeneous data.

The raster image can be an aggregation of various bands, hence can be multi-band. Various bands can store information about different features of the image. Figure 2.10 displays a raster image and its three bands. The various bands could be Elevation, Slope, Aspect, Fuel Model Number etc.

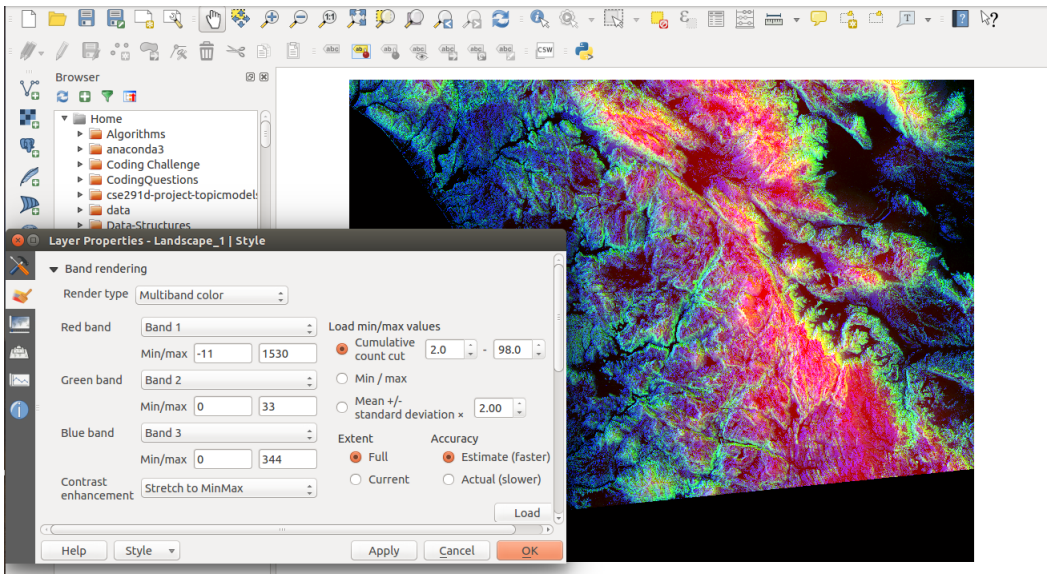

Figure 2.10: A raster image displaying three bands.

One of the raw data collected for this project is a Landscape file(.lcp) file which is a multiband raster consisting of eight bands. The eight bands describe eight different features of the landscape.

2.5.1 Geo-referencing

Geo-referencing is the method to plot the image exactly at same location from where it was captured (QGIS- Raster Data[43]). The information required to geo-reference an image is stored in the meta-data of the image. Thus if two images overlap in same coordinate reference system, that signifies they are pointing to same location.

The metadata to geo-reference an image consists of top left coordinate, size of pixel in both X and Y dimensions and the extent by which the image is rotated (QGIS- Raster Data [43]).

2.6 Map Projection

A map projection is a systematic transformation of the latitudes and longitudes of locations from the surface of a sphere or an ellipsoid into locations on a plane (Sydner, J.P.(1989)[50]). The projections always distort the spherical surface of Earth in a way or another. But they are required as maps can accommodate big scales of data for an intensive study. Thus it is best for large scale map study of an area.

2.7 Coordinate Reference System (CRS)

Coordinate Reference System (CRS) is a system used to locate objects using coordinates in the directions of the reference system. There are different codes defined for different CRSs such as European Petroleum Search Group (EPSG). The one used in the project is EPSG26911 (More details can be found here: `https://epsg.io/26911`). The coordinate reference system can be either geographic or projected.

2.7.1 Geographic Coordinate System

This system uses latitudes, longitudes or sometimes a height to describe a location (QGIS- Coordinate Reference System[41]). The most common is WGS84 (also known as WGS 1984, EPSG:4326) (National Geospatial-Intelligence Agency[38]). The coordinate origin of WGS84 is located at the Earth's center of mass, and the error is believed to be less than 2 cm (The EGM96 Geoid Undulation- NASA[55]).

Latitudes and Longitudes are imaginary lines cutting Earth horizontally and vertically respectively. The reference line for the latitudes is Equator and for longitudes is Prime Meridian. The distance from Equator to poles is assigned a value between 0° to 90° for Northern Hemisphere or 0° to -90° for Southern Hemisphere in degrees. The longitudes are assigned values from 0° to 180° East or West of the Prime Meridian (QGIS- Coordinate Reference System[41]).

2.7.2 Projected Coordinate Reference Systems

Instead of having Y-origin coordinate on Prime Meridian, this reference system has origin at some specified longitude on Equator.

2.7.3 Universal Transverse Mercator (UTM)

The Universal Transverse Mercator (UTM) identifies locations on the surface of Earth independently of vertical position (NOAA History[39]; Buchroithner, Manfred[33]). The UTM system divides the Earth between 80°S and 84°N into 60 zones, each 6° of longitude in width. Each 60 zones uses a traverse Mercator projection that can map a region of large North-South extent with low distortion.

2.7.4 North American Datum (NAD)

The North American Datum (NAD) is used to define the geodetic network in North America (Shalowitz[46];Craig, Cindy[10]). A geodetic datum defines the shape of Earth along with an anchor point for coordinate system. The two most common NAD are NAD of 1927 and 1983 having slight differences in the measurements.

North American Datum of 1983

The North American Datum of 1983 (NAD83) is the most current datum being used in North America. It provides the location information using the reference ellipsoid GRS80. This datum forms a basis of coordinates of all horizontal positions for Canada and United States. It is a successor to NAD27 and corrects some of the distortions from NAD27 over distance by using a set of positions from terrestrial and Doppler satellite data. The horizontal datums like this are improved and used today continuously.(GIS Geography: Geodetic Datum [22])

2.8 Geospatial Data Abstraction Library (GDAL)

The Geospatial Data Abstraction Library is a translator library for raster and vector geospatial data formats. It also comes with a variety of useful command line utilities for data translation and processing (GDAL Documentation[15]).

Table 2.3 shows the commandline gdal utilities used in this project and their functions. Table 2.4 shows the options which can be used with the utilities to customize the functioning.

Table 2.3: GDAL Utilities

Utility Program	Function
gdalinfo	Report information about a file[16]
gdalwarp	Warp an image to a new coordinate system[20]
gdal_rasterize	Rasterize vector data into Raster data[19]
gdal_translate	Copy a raster file, with control output format[17]

Table 2.4: Options used in GDAL Utilities

Option	Function
burn <Int>	Burn the integer value in the raster band[19]
tr <xres> <yres>	Represents the target resolution[19],[20]
cutline <datasource>	Enable the use of a blend of cutline from the OGR supported datasource[20]
crop_to_cutline	Crop the extent of the target dataset to the extent of the cutline[20]
of <format>	Select the output format[20]
t_srs	Target spatial reference set[20]
b	Select an input band for output file[17]

Chapter 3

Data Preparation

This chapter explains the preparation of dataset which is used by the machine learning models to analyze the trends and relationships. The project utilizes fire history data and landscape features of a part of San Diego County. And these data are collected from two different sources.

3.1 Data Sources

The landscape data are collected from Landfire.gov[29]. Landfire is a program that provides over 20 national geo-spatial layers such as vegetation, fuel, disturbance, databases, and ecological models that are available to the public for the US and insular areas (Landfire[31]). The Landscape data is retrieved for San Diego area in the form of LCP format (Raster data) having 8 bands in CRS (EPSG: 5070, NAD83 Conus Albers).

The fire history data are collected from The Fire and Resource Assessment Program(FRAP). The FRAP assesses the amount and extent of California's forests and rangelands and identifies alternatives and management and policy guidelines (FRAP- AboutUs[14]). Fire history data of complete California State collected is in the form of OpenFileGDB(Vector data) format in Polygon geometry in CRS(EPSG:3310, NAD83 California Albers).

Thus, it is clear that the raw data collected from both these sources are in different Coordinate Systems. Hence the first step to prepare the dataset is to align data in same Coordinate System with same cell size and same origin. The data are then cropped for a particular region which is used as dataset in the model. Any projection which is suitable for this location could be used. For this project, all the data are projected in CRS (EPSG26911 UTM11 N' NAD83) with a cell size

30*30.

3.2 Sources of Features and Label of Dataset

The model used in the project is trained with labeled data, that is for each data sample having features, there is a label against it. Table 3.1 shows the features and the details of their sources. Table 3.2 shows the label and the details of its source. The dataset can be seen as toal of feature set(X), which is a set of five features(x_1, x_2, x_3, x_4, x_5), and a label(Y). The details about the features and their significance is discussed in Chapter 4.

Table 3.1: Features and its sources details

Feature(X)	Source
Elevation(x_1)	Landscape file for the year 2012
Slope(x_2)	Landscape file for the year 2012
Aspect(x_3)	Landscape file for the year 2012
Fuel Model Number(x_4)	Landscape file for the year 2012
Fire Frequency(x_5)	Cumulative frequency of fires from 1992-2012

Table 3.2: Label and its source details

Label(Y)	Source
Fire Occurrence in 2014	Fire Perimeters from FRAP

The data are prepared for a particular region of San Diego County. The motivation to have a particular area of San Diego is to prepare a small dataset which can run easily on the hardware without any memory errors. The small part was chosen so as to include areas having fires from zero to maximum number of fires during past twenty years. And this is done by viewing the overlaps of polygons using Count Overlapping Polygon tool in ArcMap (The use of this tool is discussed in more detail in Section 3.4.2). Thus area which showed maximum to zero overlap of polygons in the San Diego was chosen. In the Figure 3.2, the green area represents the total San Diego county, while the brown colored area represents the area selected for the dataset. Carefully looking into Figure 3.1 can make it clear how this area was chosen. The brown colored area is made using a shapefile of type 'polygon' and is used as a reference file to clip the data from the raw data.

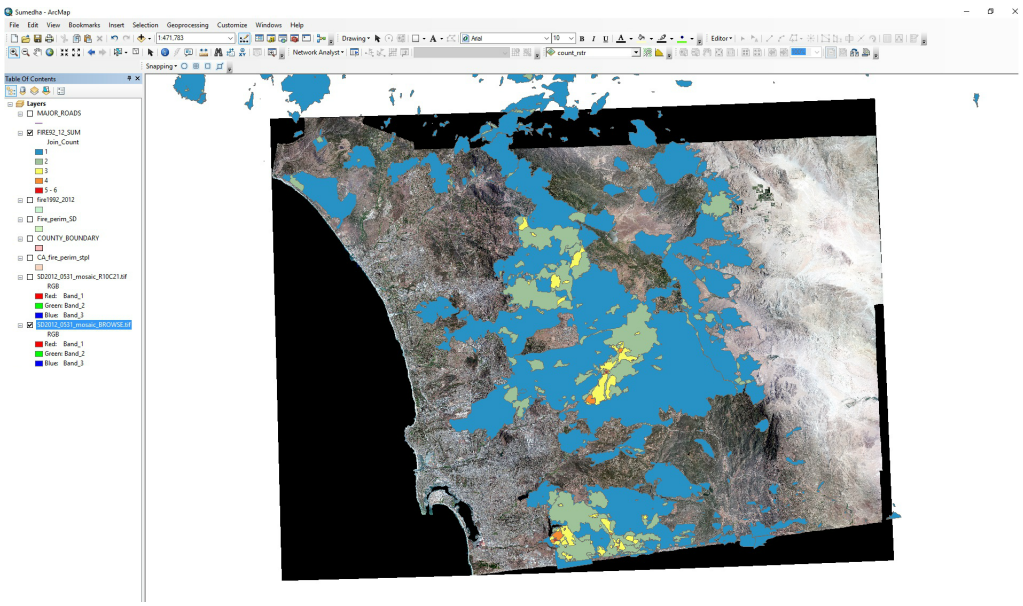

Figure 3.1: ArcMap: The tool gives the count of overlapping polygons of the shape file. The left pane shows the frequencies from 0-5

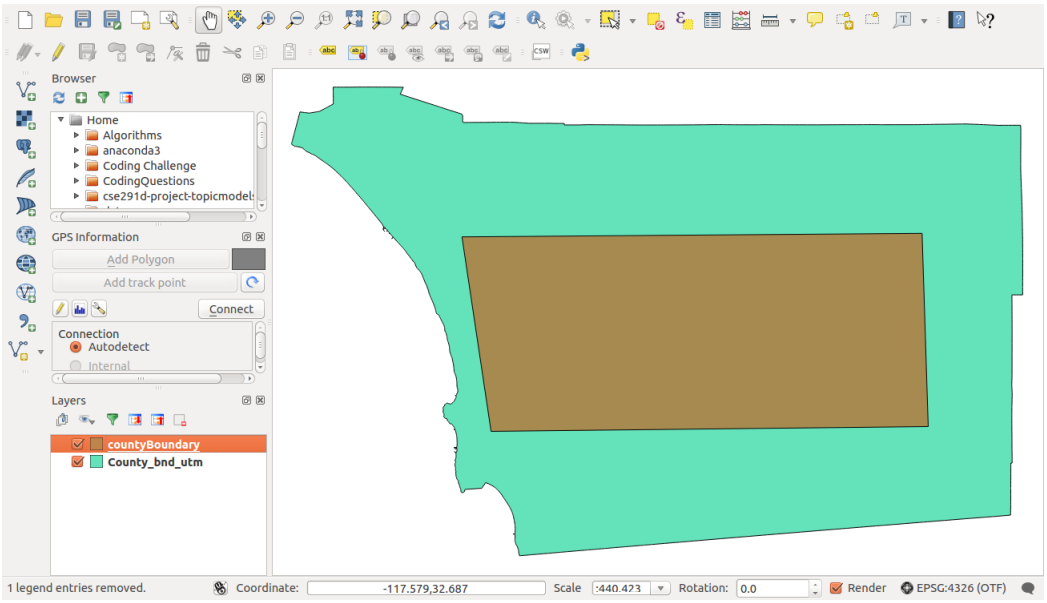

Figure 3.2: QGIS: The green area represents the total San Diego county and the brown area is the dataset which is used in the project

3.3 Preparing Data from Fire Perimeters

As discussed in Section 3.1, the data for fire history is collected from FRAP. The file used is Fire Perimeters 15_1. This data provides a reasonable view of the spatial distribution of past large fires for whole California state. Figure 3.3 shows the original Fire Perimeters data downloaded from FRAP of California state.

This fire perimeter data is used to prepare one feature (Fire Frequency(x_5)) and labels(Y) of the dataset. The label is a binary number which is true(1) if the

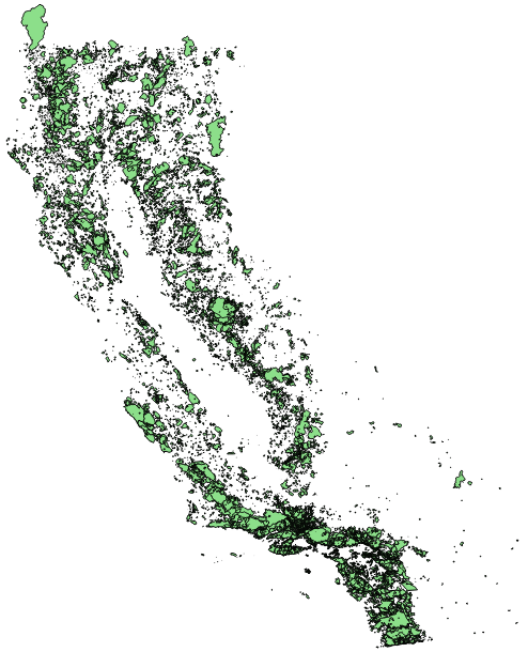

Figure 3.3: Original Fire Perimeters downloaded from FRAP.

fire happened in year 2014 and false(0) otherwise. The feature Fire Frequency(x_5) is the cumulative frequencies of fires that happened from 1992 - 2012. The subsection 3.3.1 explains the preparation of Label(Y) and consequent Subsection 3.3.2 explains the Fire Frequency as the latter utilizes the former.

3.3.1 Preparing Labels of the Dataset

The steps to prepare the Labels(Y) of the dataset from Fire Perimeters file are written below:

1. Saving file in .shp format
   The original file downloaded from Fire Perimeters is a file geodatabase file (.gdb format). Open this Fire Perimeter File in QGIS/ArcMap and save it in .shp file format. The projection of this file should be NAD83 EPSG 26911 UTM 11N‘.

2. Selecting rows for the year 2013-14
   Turn on ”Toggle Edit Mode” and open the attributes table of the new file. Select ”Select features using Expression”. This feature selects the rows which satisfies this expression and write the query in the expression panel. Figure 3.4 shows screenshot of the dialog box where the expression is written in QGIS.
   ”YEAR_” >= 2013 AND ”YEAR_” <= 2014

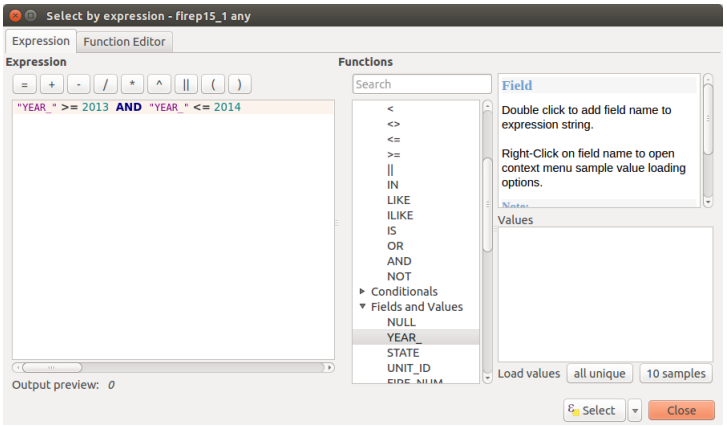

Figure 3.4: QGIS: Window representing the expression query to select rows satisfying the query criteria.

The query selects the rows which had fires in the year greater than or equal to 2013 and less than or equal to 2014. Invert selection and delete the inverted selected rows. The remaining rows are thus the ones which are required. Save the file.

3. Rasterization

   The file obtained above is a vector data. It is rasterized to convert it into Raster data. This is done using the utility gdal_rasterize. This program utility burns vector geometries (points, lines, polygons) into the raster band of a raster image (GDAL- Rasterize[19]). The command used is-

   gdal_rasterize -burn 1 -tr 30 30 fire2014.shp fireRaster2014.tif

   Running the above command gives a Raster file in the .tif format. The value '1' is burned where polygons exist or where fire happened. The cell size in both the X and Y directions is kept 30. Also, it is worth noting that first rasterization of complete file is done and then we do the clipping of the required area from the total area as the reverse does not give the adequate results (data gets lost).

   The above command takes 'fire2014.shp' as input file and outputs 'fireRaster2014.tif' as output file.

4. Clipping

   The raster file obtained in the previous step is clipped using the utility gdalwarp. The gdalwarp utility is an image mosaicing, reprojection and warping utility (GDAL- Warp[20]).

gdalwarp -q -cutline countyBoundary.shp -crop_to_ cutline -of GTiff -tr 30 30 fireRaster2014.tif fire2014.tif

This command clips an area of 'countyBoundary.shp' from 'fireRaster2014.tif' file and outputs 'fire2014.tif'. The cell size option is used again, as not using it changes the cell size.

3.3.2 Preparing Cumulative Fire Frequency Feature of the dataset

As discussed previously, one of the feature is cumulative sum of the fire frequencies of past twenty years from 1992- 2012. The cumulative sum is done using a plugin -Count Overlapping Polygons in the ArcMap. This tool is not implicitly available in ArcMap. It is downloaded from the following url- `https://www.arcgis.com/home/item.html?id=1dd4a6832b3d40b494dbf8521cc5134c` and then installed in ArcMap. The tool counts the sum of overlapping polygons at a particular point and returns the count. Thus when an input file of Fire Perimeters containing fire history from 1992-2012 (prepared using the first step of Section 3.4.1) is fed to this tool, it returns the sum count of fires from 1992 to 2012. Figure 3.5 shows the Count Overlapping tool window when used in ArcMap.

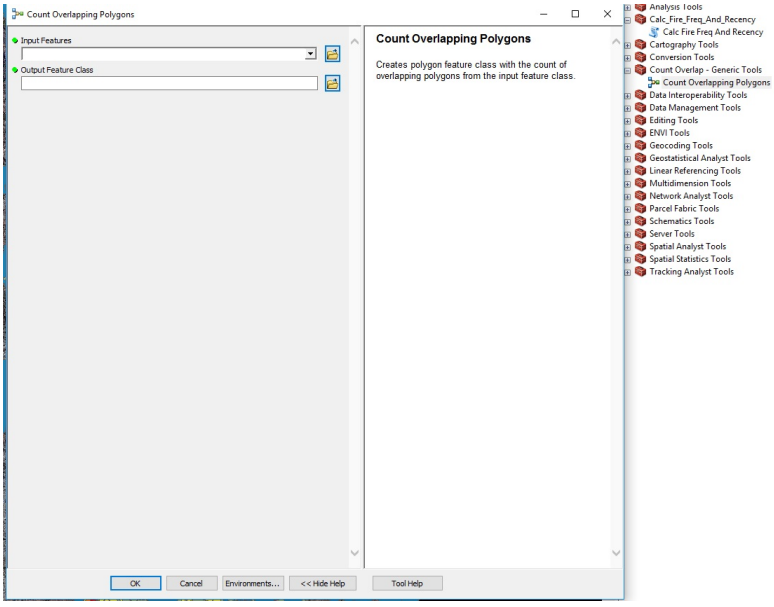

Figure 3.5: Count Overlapping tool window in ArcMap

1. Selecting Rows

   Follow the first step of Section 3.4.1 to make a shapefile and then use the expression to select rows from 1992 - 2012.

"YEAR_" >= 1992 AND "YEAR_" <= 2012

Thus a shapefile having polygons from 1992 - 2012 is obtained following this step.

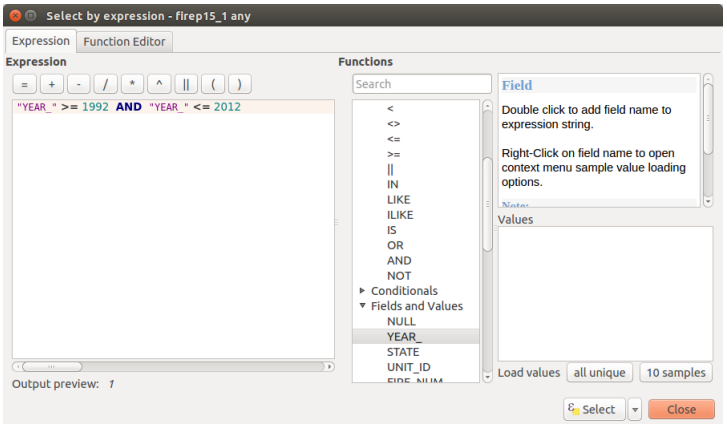

Figure 3.6: QGIS: Window representing the expression

2. Use Count Overlapping Polygons Plugin

   The shapefile created above is then used as 'Input Features' in the Count Overlapping tool to get the summation of overlapping polygons. The output is a new feature class with the removal of overlaps and addition of new 'Join_Count' field, which contains the number of overlapping polygons.

3. The shape file is then rasterized and clipped as was done for Fire 2014 data.

3.4 Preparing Data from Landscape File

The dataset uses the landscape features of year 2012. The data is collected for the year 2012 under heading 'us_130 LCP 13 Fire Behavior Fuel Models - Anderson' found at the following url:`https://landfire.cr.usgs.gov`.

The data downloaded is in the form of LCP format. LCP is a multiband raster format used by wildland fire behavior and fire effect simulation models and store data describing terrain, tree copy and surface fuel (GDAL- LCP format[18]). The dataset prepared for this project uses first four bands of the file namely Elevation, Slope, Aspect and Fuel Model Number.

Steps to prepare dataset features using this file are written below:

1. Warping

   The orginal file is reprojected to CRS (UTM 11N EPSG26911 NAD83). This

is done using the gdal_warp commandline utility (GDAL- Warp[20]). The command used is:

gdalwarp -tr 30 30 -t_srs '+proj=utm +zone=11 +datum=NAD83' Landscape_1.lcp landscape.lcp

Running this command gives an output file 'landscape.lcp' having desired projection with cell size 30*30.

2. Translating

   Next step is to make individual band files out of multiband raster obtained in the previous step using gdal_translate (GDAL- Translate[17]). Since the dataset requires first four bands (Elevation, Aspect, Slope and Fuel Model Number), the command is ran four times. The command used is:

   gdal_translate -b 1 landscape.lcp band_1.tif
   gdal_translate -b 2 landscape.lcp band_2.tif
   gdal_translate -b 3 landscape.lcp band_3.tif
   gdal_translate -b 4 landscape.lcp band_4.tif

   The gdal_translate retrieves the required (1,2,3 or 4) bands from the multi-band (landscape.lcp) file and gives the output file(band_<x>.tif file.

3. Clipping

   The individual band files are clipped using the following command (exactly similar to the step 4 of Section 3.5.1):

   gdalwarp -q -cutline countyBoundary.shp -crop_to_cutline -of GTiff -tr 30 30 band_1.tif band1.tif

After following the procedure above, a total of six files are created. Label (Y) file using Section 3.3.1. First four features(x_1, x_2, x_3, x_4) of the feature set using Section 3.4 and feature(x_5) using Section 3.3.2. The origin of each of these files is (477422.18, 3674534.79) and the pixel size of (30.00,-30.00). Thus, our dataset is ready to be used by the machine learning model.

Chapter 4

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an approach for data analysis so as to increase the understanding of the dataset before applying the models on the dataset. The techniques generally used to explore data are graphs, histograms.

4.1 Shape of the original Dataset

As already discussed in Section 3.3, the dataset has 5 features. The shape of the dataset is described in Table 4.1.

Table 4.1: Shape of the original Dataset

Class	Number of Samples	Percentage
Total	4665108	100
Non-Burnt(0)	4519046	98.51
Burnt(1)	68230	1.49

From Table 4.1, it is clear that class 'Non-Burnt' is in high majority and the class 'Burnt' is in acute minority. The dataset thus faces class imbalance problem, as the classes are not represented equally.

4.2 Features of the Dataset

The file obtained for each feature is of size 1554 * 3002 with cell size 30 * 30 m each. Thus total area becomes $(1554 * 3002 * 30 * 30)/(1000 * 1000) = 4198.59$ sq.km. Total area of San Diego county is 11720 sq.km as per Wikipedia[56]. The area chosen for the preparation of the dataset is 35.82% (also refer Figure 3.2- the green colored area is 35.82 of the brown) of total area of San Diego county.

The consequent subsections explain the five features of the dataset in detail using histograms and statistics.

4.2.1 Elevation

Elevation represents land height, in meters, above mean sea level (Landfire: Topographic[32]). Figure 4.1 represents the Elevation band of the dataset when viewed in ArcMap. The dark shade represents the higher values of Elevation; lighter shade represents the lower values of Elevation.

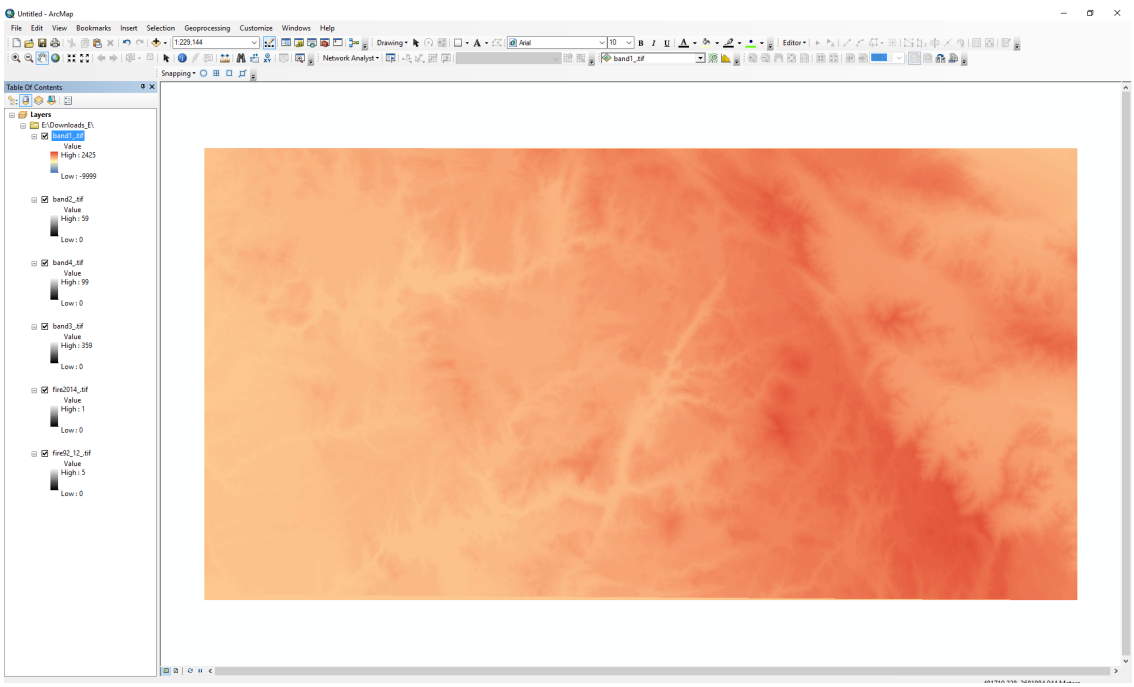

Figure 4.1: Elevation band view in ArcMap

Table 4.2: Statistics of Elevation

Statistics	Value
Maximum	1977
Minimum	0
Mean	663.61
Median	578.0
Std. Deviation	454.39
Variance	206466.67

From the table we can see that, the area has an average of 663.61m above sea level. The maximum value goes to 1977. (The area chosen for the dataset is a part of San Diego county, not complete. The wikipedia (Wikipedia: San Diego County[56]) says that maximum Elevation of San Diego County is 1992(m). Thus values are good to go.)

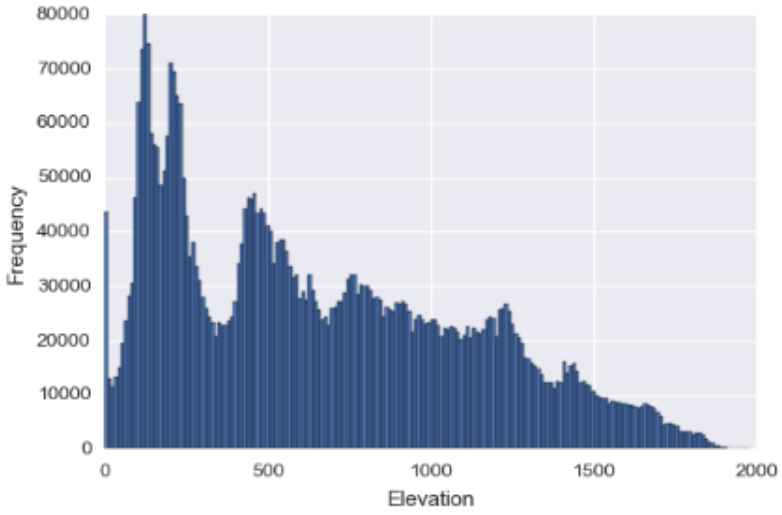

Figure 4.2: Histogram describing the statistics of Elevation

Figure 4.2 shows the histogram describing the statistics of Elevation. It might appear that 0m is occupying a lot of area, but it is only 0.92%(43000/4665108 = 0.0092) of the total area. Again the wikipedia (Wikipedia: San Diego[56]) says the water is 830 sq km and total is 11720 sq km, which is 0.0708 of the total. Thus the area chosen for the dataset has less percentage of water as compared to the percentage of water in total San Diego County.

4.2.2 Slope

Slope is the relationship of vertical rise to a horizontal run, expressed as a percentage change of Elevation over a specific area. It is expressed in degrees (Landfire- Topographic[32]). Figure 4.3 represents the Slope band of the dataset when viewed in ArcMap. The dark dodger blue represents the higher values of Slope as compared to the lighter shades which represents lower values.

Table 4.3: Statistics of Slope

Statistics	Value
Maximum	66
Minimum	0
Mean	11.62
Median	10.0
Std. Deviation	9.175
Variance	84.185

According to definition, if the value of Slope is 26.6, then tan(26.6) = 50, which means there are 100 units of vertical units for 200 units of horizontal distance. The mean of the area is 11.62 which gives tan(11.62) = 20.56 percent.

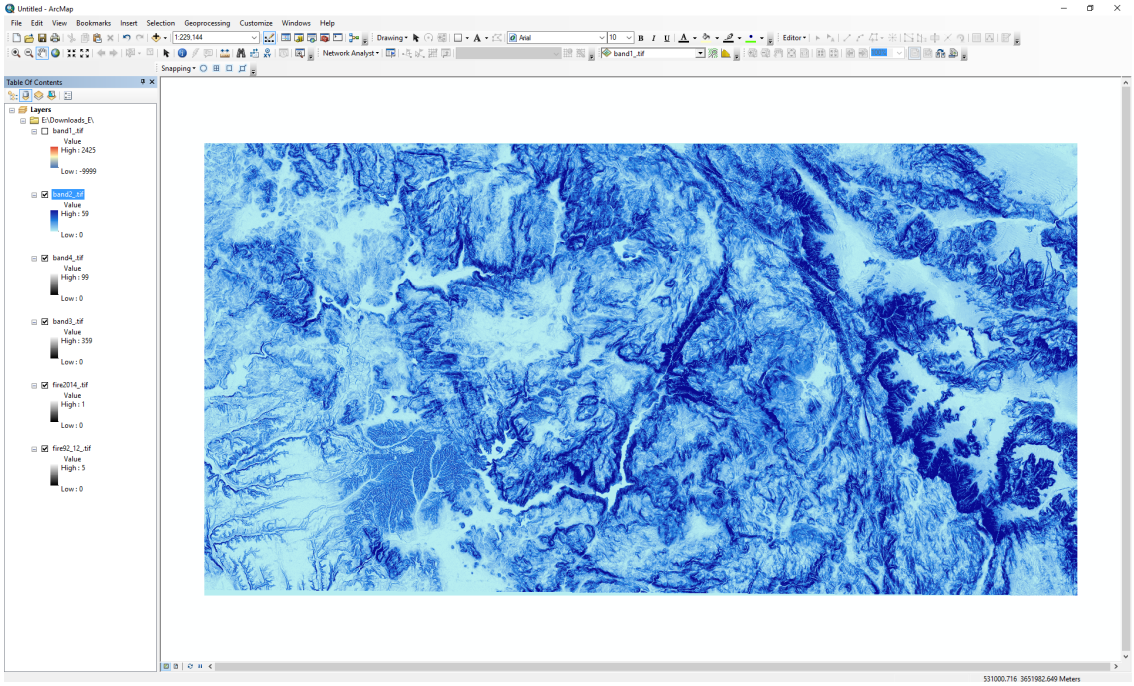

Figure 4.3: Slope band view in ArcMap

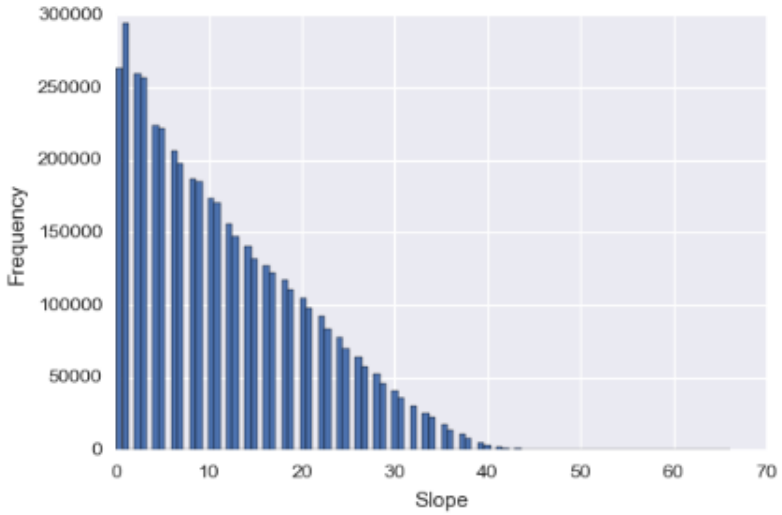

Figure 4.4: Histogram describing the statistics of Slope

As per geographyfieldwork.com(Geography Field Work- Slope Steepness Index[21]), the Slope Terminology is defined in the Table 4.4:

Figure 4.4 represents the histogram describing the statistics of Slope band. Looking at the histogram, it can be inferred that the higher frequencies of Slope lie in the level, gentle and moderate scales. The median value is 10 and mean is 11.62. The higher values from 11.62 to maximum (66) are distributed in rest half of the area of dataset.

Table 4.4: Slope Terminology

Slope(%)	Degrees	Terminology
0 - 2	0 - 1.1	Level Slope
2 - 9	1.1 - 5	Gentle Slope
9 - 15	5 - 8.5	Moderate
15 - 45	8.5 - 24	Strong Slope
45 - 70	24 - 35	Extreme Slope
>70	>35	Steep Slope

4.2.3 Aspect

Aspect identifies the downslope direction of the maximum rate of change in value from each cell to its neighbors. It can be considered as Slope direction. The values from 0 (North) to 360 (Again North) clockwise indicate the compass direction. The values in the dataset gives the information about the cell's Slope faces (ArcGIS - Aspect[5]). Figure 4.5 represents the Aspect band of dataset when viewed in ArcMap. The bluish shade represents the higher values of Aspect and the pinkish shade represent the lower values of Aspect.

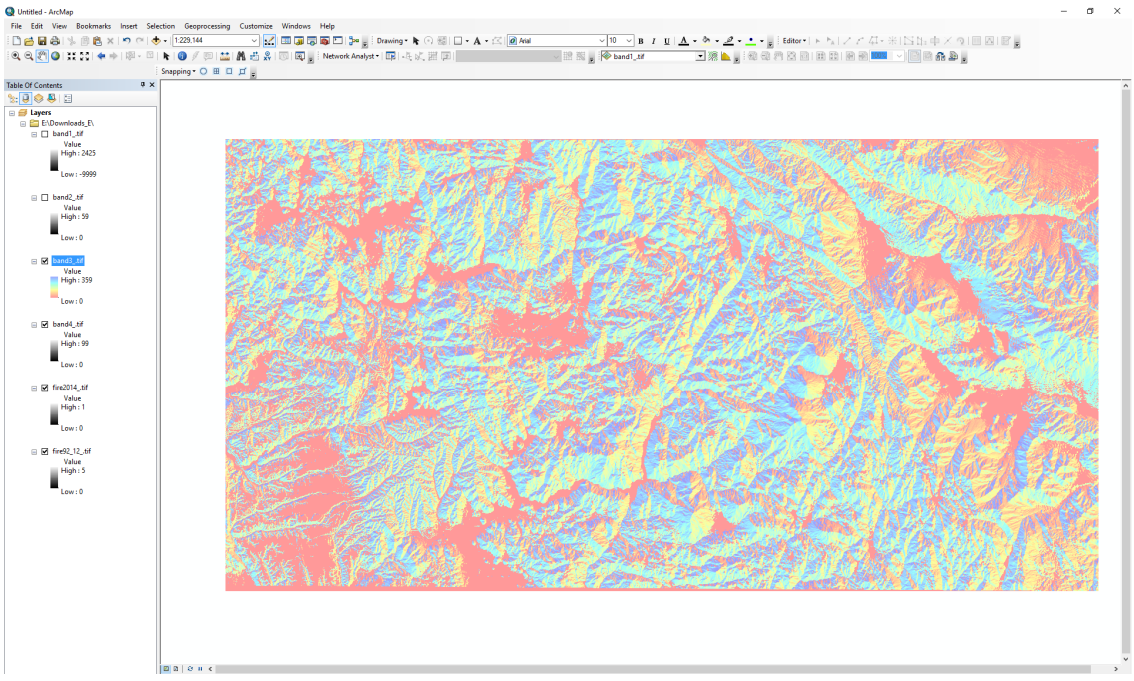

Figure 4.5: Aspect band viewed in ArcMap

The influence of Aspect with respect to Sun is most important in mid-latitudes, and San Diego lies in this region. Hence it is important to consider this as a feature for the study.

In northern hemisphere, south facing slopes receive more heat as compared to north facing slopes which receive very little heat from Sun. Thus there would be a

Table 4.5: Aspect and direction relationship

Value	Direction
0 - 22.5	North
22.5 - 67.5	Northeast
67.5 - 112.5	East
112.5 - 157.5	Southeast
157.5 - 202.5	South
202.5 - 247.5	Southwest
247.5 - 292.5	West
292.5 - 337.5	Northwest
337.5 - 360	North

Table 4.6: Statistics of Aspect

Statistics	Value
Maximum	359
Minimum	0
Mean	151.59
Median	150
Std. Deviation	116.32
Variance	13532.353

different climatic conditions in North as compared to South. North will be cooler, more shadier, moist and would have different kind of vegetation as compared to the South facing Slope. The Southern side will be warmer and drier. Species will be hard , vegetation would include hardwoods, maples etc (National Avalanche Center- [35]).

East facing Slopes are colder than west facing Slopes as the East side receive the sunlight during the morning hours while the west side receive the sunlight during the noon hours.(National Avalanche Center- [35]).

As per the distribution of Aspect as shown in Figure 4.6, it can be seen that 0.186 (approx) of the total data is flat. The remaining 0.813 (approx) has values from 1-360. The value of median and mean is bent towards the first half due to high frequency of zero. Also, the values between 150-359 have higher frequencies as compared to the values from 1-150. Thus it can be inferred that data is more from South East - South - South West - West - North West as compared to the North - North East - East.

4.2.4 Fuel Model Number

Fuels are classified into four groups- grasses, brush, timber and slash. The difference in fire behavior among these groups are basically related to the fuel load

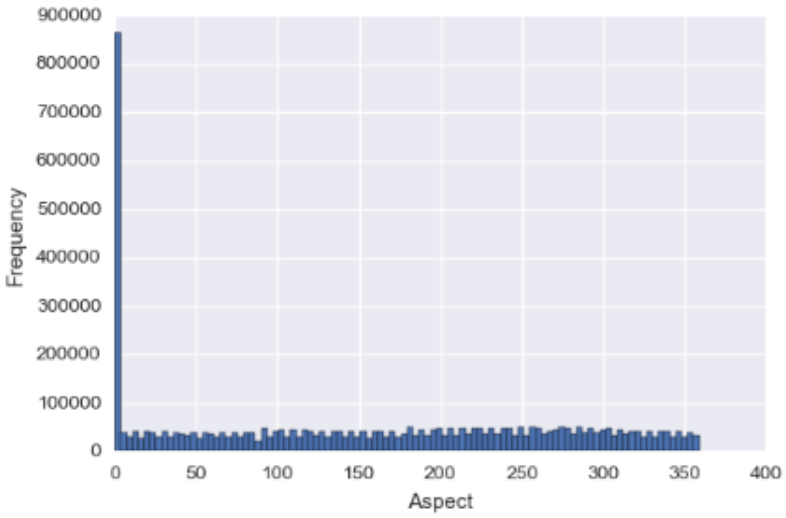

Figure 4.6: Histogram describing the statistics of Aspect

and its distribution (Anderson- Aids to Determining Fuel Models For Estimating Fire Behavior[1]). Figure 4.7 represents the distribution of Fuel Model Number band in ArcMap. The dark red values represent the non-burnable fuel. The lighter values represent the Fuel Model Number corresponding to burnable category. On mapping Figure 4.7 with Figure 4.9, it can be said that the lighter shade of red (burnable fuel) almost maps to the past twenty years of fire frequency layer. Thus area which lies is dark red, is pretty impervious to wildfires and represents the characteristic features of non-burnable class.

The brief of Fire Behavior Fuel Model (FBFM) Numbers is written here. The description of each Fuel Model Number is added in the Appendix A (Landfire-Data Dictionary[30], [1]).

- 0 (No data)

Grass and grass-dominated

- Short Grass (FBFM1)
- Timber Grass (FBFM2)
- Tall Grass (FBFM3)

Chaparral and shrub fields

- Chaparral (FBFM4)
- Brush (FBFM5)
- Dormant Brush, hardwood slash (FBFM6)

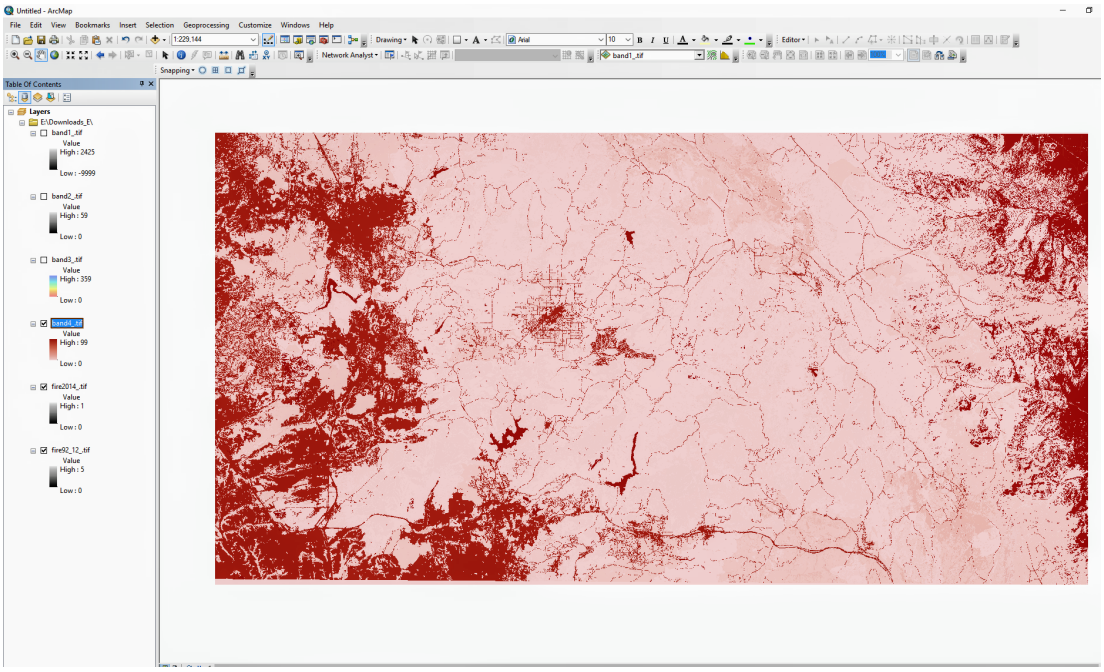

Figure 4.7: Fuel Model Number band view in ArcMap

- Southern rough (FBFM7)

Timber litter

- Closed timber litter (FBFM8)
- Hardwood litter (FBFM9)
- Timber (litter and understory) (FBFM10)

Slash

- Light logging slash (FBFM11)
- Medium logging slash (FBFM12)
- Heavy logging slash (FBFM13)

Non-burnable

- Urban (91)
- Snow/Ice (92)
- Agriculture (93)
- Water (98)
- Barren (99)

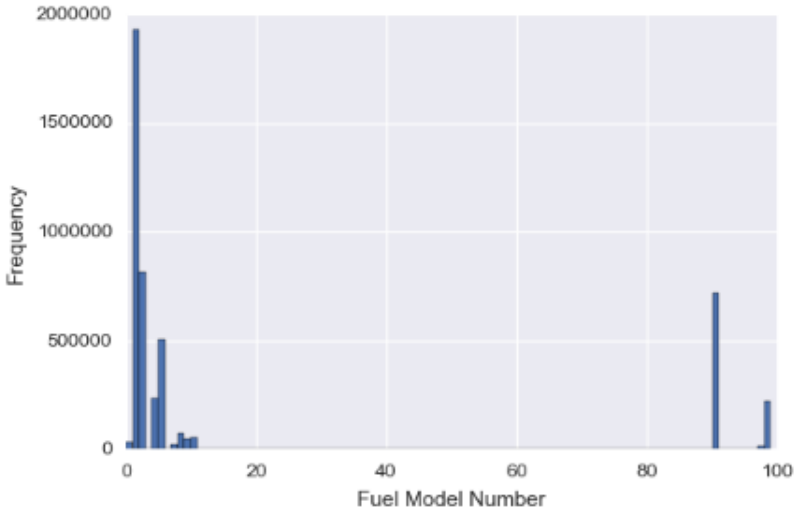

Figure 4.8: Histogram describing the statistics of Fuel Model Number

Figure 4.8 represents the histogram of Fuel Model Number distribution. The values which are visible as '0' can be counted as NoData or the error in the file as there is no Fuel Model Number corresponding to '0'. It can be seen that Short Grass occupies 0.41 of the total, Grass with Timber/ Shrub Overstory occupies 0.17 of the total and then Young Brush. All these fuels have different fire behavior properties.

4.2.5 Cumulative Fire Frequency

This feature is made using special tool in ArcMap. This feature can help in understanding, how are fire frequencies related and answer questions like are there areas which are more susceptible to get burn again and again? The number represented by this feature is the aggregation of frequencies from 1992-2012. 0 represents the pixel never got burnt, 1 represents the pixel got burnt once, 2 means twice and so on. The maximum value is 5 which signifies that a pixel got burnt 5 times in past twenty years.

Figure 4.9 represents this layer when viewed in QGIS. The black area represents the area which never got burnt in past twenty years. The lighter shade represents the higher values of fire frequency. The lightest shade represents the value five.

From the histogram shown in Figure 4.10, it can be deduced that around 50% of the area never got burned in the past 20 years. 40.23% got burnt only once, 7.8% burnt only twice, 1.31% thrice, 0.12% four times, 0.004 % five times. Table 4.7 shows the data dictionary of cumulative fire frequency, by mapping the value (number of times a pixel got burnt) and its occurrence in number of samples.

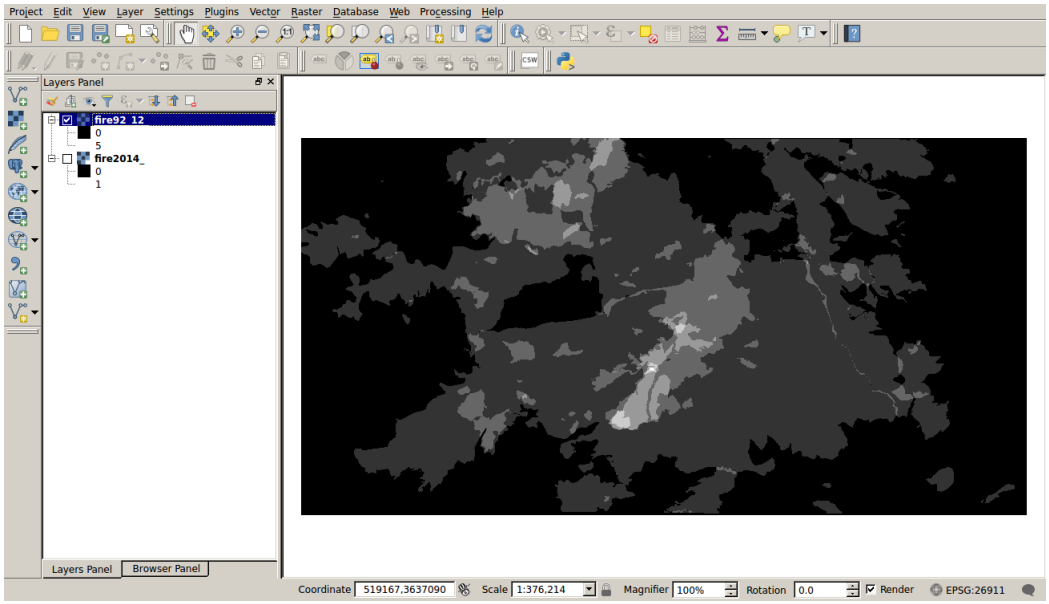

Figure 4.9: Cumulative fire frequency layer view in QGIS

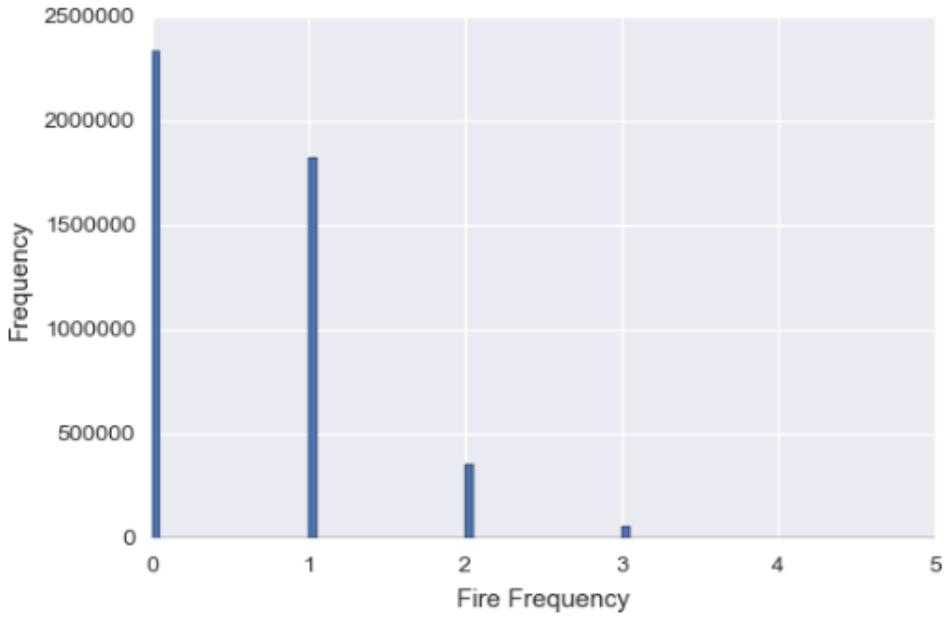

Figure 4.10: Histogram describing the statistics of cumulative fire frequency

4.3 Target of the Dataset

The target of dataset represents 1 if fire occurred in the the year 2014 and 0 if not. Figure 4.11 represents this layer when viewed in QGIS. The white area represents the area which had fires in the year 2014 and black, the opposite. From this image, it is evident that the dataset is imbalanced. The burnable class represents very less percentage of the total area.

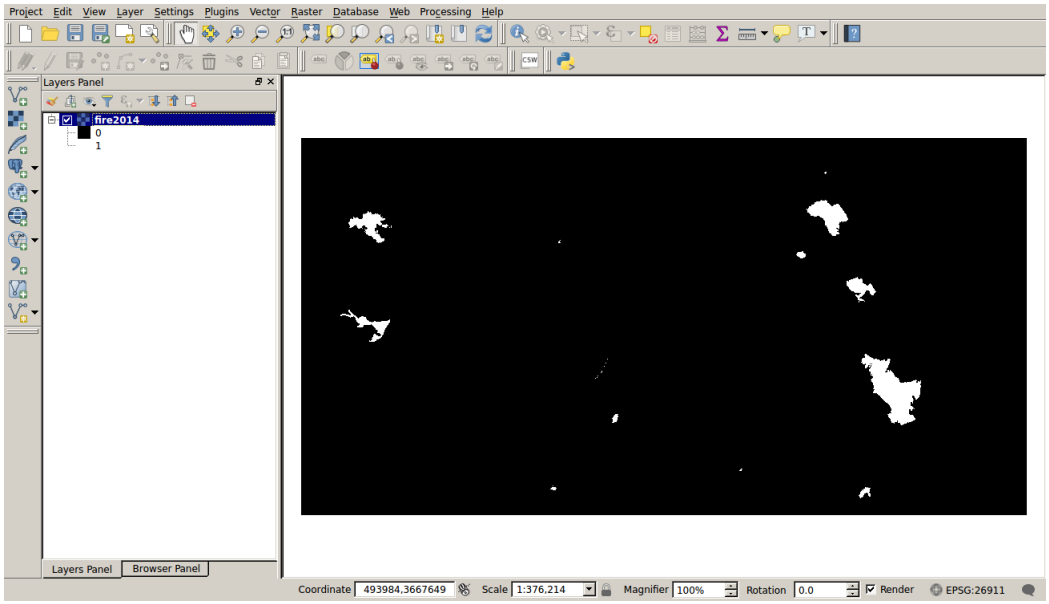

Figure 4.11: Target layer(fire occurrence in 2014) view in QGIS

4.4 Correlation Between Variables of the Dataset

A correlation matrix is a table showing correlation coefficients between sets of variables. Each variable X_i in the table is correlated with each other values in the table (X_j). This helps in understanding which pairs have highest, lowest or no correlation. And obviously, the diagonal elements are all 1 as it is a relationship reflecting how a variable is related to itself.

The features are represented as Ff(Fire Frequency), E(Elevation), S(Slope), A(Aspect), FM(Fuel Model Number). Positive values represent positive relationship and vice versa. Figure 4.12 shows the correlation matrix of features and target variable. From the figure, it can be seen that the relationship between Aspect and Slope is profound. And so is the relationship between Aspect and Fire Frequency. Higher the Aspect value, more is the fire frequency. Thus it is in tune what is expected. As the value of Aspect increases or it points to the Southern facing Slopes; warmth and temperatures increase and hence these areas are more prone to fires.

Though the exact relationship between Fuel Model Number and Fire Frequency cannot be gauged correctly by looking at the statistics and graph as these are some discrete values of Fuel Model Number. But it can be seen that the relationship is negative. Thus, higher values have opposite trends of Fire Frequency, which is again what is expected. The bigger values of Fuel Model Number (90) comes under Non-burnable fuels and hence justifies the relation.

	Ff	E	S	A	FM	Class
Ff	1.000000	-0.021765	0.246805	0.152234	-0.346138	0.018798
E	-0.021765	1.000000	-0.004875	-0.007552	0.002271	0.007642
S	0.246805	-0.004875	1.000000	0.310003	-0.240521	0.071563
A	0.152234	-0.007552	0.310003	1.000000	-0.162514	-0.021614
FM	-0.346138	0.002271	-0.240521	-0.162514	1.000000	-0.046330
Class	0.018798	0.007642	0.071563	-0.021614	-0.046330	1.000000

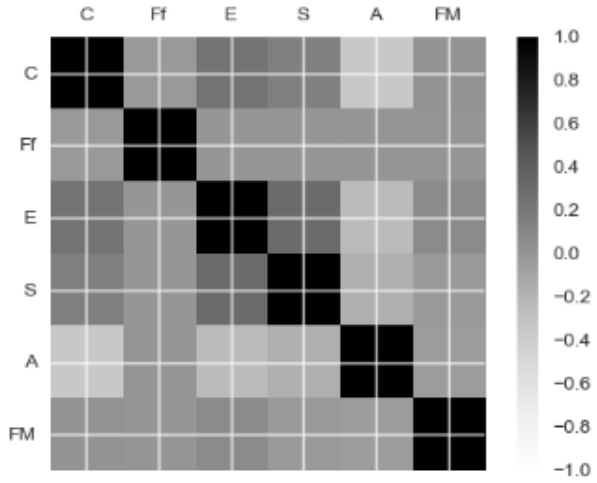

Figure 4.12: Correlation Matrix of different features and the target variable

4.5 Scatter Plot

Figure 4.13 is a scatter plot, made by randomly choosing 40,000 samples from the total dataset. The scatter plot also shows an over-abundance of Non-Burnable class. The burnable points lie in the regions having high Aspect values, low Fuel Model Numbers, high Elevation values. A combination of higher Elevation and higher Aspect shows more burnable points.

It can be noticed that burnable class curve matches most with frequency two of non-burnable class. Thus, the area which had burnt twice in the past 20 years are most likely to burn. Also graph between Elevation and Burnable class shows two peaks. Thus fires happen more in either very low Elevations or very higher Elevations. Similar trend can be observed with Aspect and Slope. Graph with Fuel Model Number clearly indicates that Burnable class lies for Fuel Model Numbers which have less values and negligible for higher Fuel Model Numbers (Non-burnable Anderson fuels).

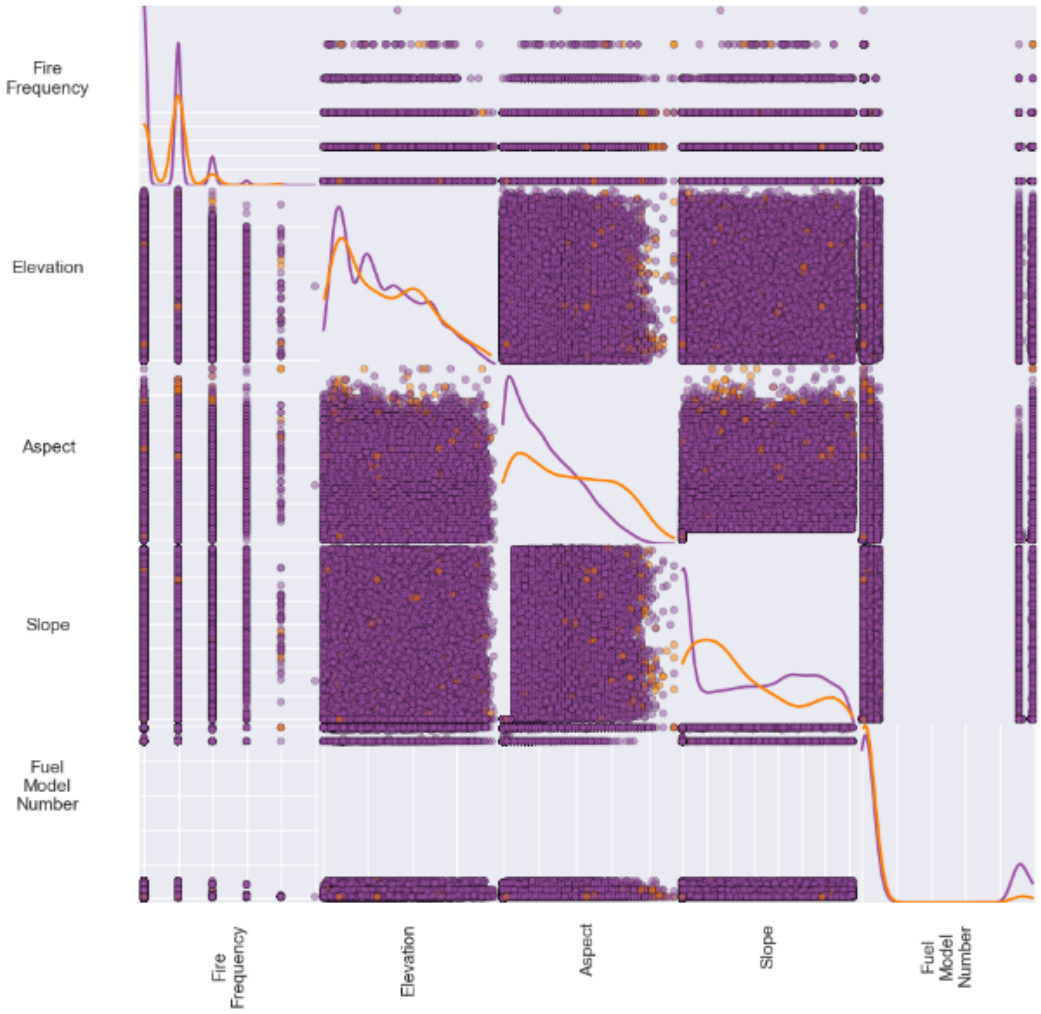

Figure 4.13: Scatter plot made using randomly chosen 40K samples from the total dataset. Purple-Non-Burnable; Orange-Burnable

Chapter 5

Data Modeling

Chapter 4 explained the EDA and this chapter illustrates the next step of Data Analysis i.e. Data Modeling. From the previous chapter, it is clear that dataset is too big and skewed(imbalanced), so this chapter starts with scores that can be used to evaluate the models, the re techniques which are applied to training set, and then selecting the best supervised learning model using the best re technique which gives favorable scores.

5.1 Splitting of Dataset

Table 5.1 shows how the dataset is split and used for the modeling. The training set is 80% of the total dataset and the remaining 20% is the test set. The data is split using a stratified fashion using the class labels (Sklearn- train, test split[49]). This is because, when subpopulations within an overall population vary, it is advantageous to sample each subpopulation or stratum independently (Hunt, Neville; Tyrell, Sidney(2001)). From the Table 5.1, it is also clear that the percentage of Non-burnt samples and burnt samples ratio is same in all three sets, 98.51% non-burnt samples and 1.49% burnt samples.

Table 5.1: Splitting of dataset

Set	Non-burnt(0)	Burnt(1)
Original(100%)	4596878(98.53%)	68230(1.47%)
Training(80%)	3677502(98.54%)	54584(1.46%)
Test(20%)	919376(98.53%)	13646(1.47%)

5.2 Scores to Evaluate the Classifier

There are various scores via which a model can be analyzed. The dataset used in this project is particularly big and skewed, so before using the scores, let us check if these scores stand valid for this dataset. Let us consider the most commonly used scores one by one-

Confusion Matrix is used to evaluate the quality of the output of a classifier. The diagonal elements represent the number of points for which the predicted labels is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier (Kohavi and Provost, 1998 [28]; Stehman, Stephen V[51]). Table 5.2 shows the confusion matrix.

Table 5.2: Confusion Matrix

Class	Prediction (0)	Prediction (1)
Actual(0)	True Negative (TN)	False Positive (FP)
Actual(1)	False Negative (FN)	True Positive (TP)

Accuracy Score is given by sum of total true values of both the classes by total values (Scikit- Accuracy Score). It can be written as

$$AccuracyScore = (TP + TN)/(TN + FP + FN + TP)$$

For this dataset, which is too big and too skewed the number of samples that belong to (FN + TP) is 1.5% of the total dataset and the number of samples that belong to (TN + FP) is 98.5% of the total dataset. Therefore, the following can be written:

$$(TP + TN) \approx TN$$

$$(TN + FP + FN + TP) \approx (TN + FP)$$

$$AccuracyScore \approx (TN)/(TN + FP)$$

From the above equation, it is plain that Accuracy Score does not reflect the values of FN or TP, which represents the minority class. Hence it is not a good evaluator for this dataset.

Precision also known as positive predictive value is the fraction of relevant instances among the retrieved instances (Kent, Berry, Luehrs Perry, 1955[27]). It is given as:

$$Precision = TP/(TP + FP)$$

For this dataset,

$$FP >> TP$$

hence

$$TP + FP \approx FP$$

Thus precision cannot be used as an evaluator for this dataset. And therefore, precision and recall curve also cannot be used to gauge the model.

F1- Score for the positive class(Van Rijsbergen- Information Retreival[53]), is given by the equations below:

$$F1(positive\_class) = (2 * Precision * Recall)/(Precision + Recall)$$

or

$$F1(positive\_class) = (2 * TP)/(2 * TP + FN + FP)$$

For this dataset, whatever be the value of False Positive cases, it will be still much greater than (TP + FN) due to the size of the dataset, hence

$$FP >> TP$$

$$FP >> FN$$

$$FP >> (TP + FN)$$

thus,

$$2 * TP << (2 * TP + FN + FP))$$

or

$$F1(positive\_class) \approx 0$$

Thus, F1 score for positive class will be close to zero. Similarly, for negative class, F1 score is given by:

$$F1(negative\_class) = (2 * TN)/(2 * TN + FN + FP)$$

For this dataset,

$$TN >> FN$$

$$FP >> FN$$

, hence

$$F1(negative\_class) \approx (2 * TN)/(2 * TN + FP)$$

And if the model is trained well,

$$2 * TN >> FP$$

hence,

$$(2 * TN) \approx (2 * TN + FP)$$

and therefore,

$$F1(negative\_class) \approx 1$$

Thus F1 score for negative class (majority) hides the effect of values from False Negative (FN) cases. The value will be close to one due to bigness of the dataset. Thus, overall F1 score cannot be used as an evaluator for either majority or minority class.

So, what could be the scores to evaluate this dataset? The scores which are not affected by the bigness of the dataset. Or the scores, where the minority class effect is not hidden due to majority class. These scores could be the one which are solely built for their class. Consider two operating characteristics- true positive rate (TPR) and false positive rate (FPR). TPR (Haiying Wang, Huiru Zheng[23]) is given by the following equation:

$$TPR = TP/(TP + FN)$$

and FPR(Burke, Donald; Brungade, John; Redfield, Robert(1998)[7]) is given by the following equation:

$$FPR = FP/(FP + TN)$$

Both TPR and FPR are rates for their corresponding class and hence the would not be affected by the bigness of the dataset. Also the Receiver operating characteristic, ROC curve (Metz, 1978[34]; Zweig Campbell, 1993[57]) which is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various settings, would not be affected by the bigness and skewness of the dataset. Hence Confusion Matrix, TPR, FPR and ROC curve can be used as evaluators for this dataset.

For the confusion matrix, higher values of the diagonal elements are desirable, indicating many correct predictions. TPR and (1- FPR) should be close to one

for a better model. Accuracy using ROC curve is measured using area under ROC curve. An area of 1 is a perfect value, 0.5 represents a worthless test. A value greater than 0.8 is considered as a good score(UNMC- Area Under the ROC Curve[52]).

5.3 Resampling Techniques to Combat the Imbalance of Dataset

This section illustrates the use of various resampling techniques on a decision tree classifier to solve the imbalance problem of the dataset. The various techniques are- Oversampling, Over-undersampling and Undersampling techniques. But before applying these techniques to a plain decision tree classifier, the results are also found using plain decision tree without any resampling technique.

5.3.1 Plain Decision Tree

Decision Tree is a divides the dataset in a tree like structure, where each internal node represents a "test" on an attribute, a branch represents the result of test and each leaf represents a class label (J. R. Quinlan[25]). The tree learns by splitting the dataset based on attribute value test and keeps on partitioning the data recursively to reach the leaves (Wei-Yin Loh[54]).

Results

The model is fitted with 80% of the data and tested on test set, which is 20% of the data. Table 5.3 shows the TPR and FPR using plain decision tree. The FPR scores are good, indicating that negative class (majority class) is classified better. But the TPR scores are worst, indicating that minority class is poorly classified.

Table 5.3: Scores using Decision Tree Classifier

Score	Value
TPR	0.14
FPR	0.012

Table 5.4 shows the actual confusion matrix of the data samples and the Table 5.5 shows the normalized confusion matrix. This matrix also shows that minority class is poorly classified.

The better classification of majority class is due to the fact that tree has learned exceedingly high with majority class that it cannot recognize the minority class.

Table 5.4: Confusion Matrix using plain Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	908039	11337
Burnt	11770	1876

Table 5.5: Normalized Confusion Matrix using Plain Decision Tree Classifier

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.99	0.01
Burnt	0.86	0.14

The tree only knows the majority class and very less about the minority class. Thus, each sample is classified as a majority class sample. The normalized value 0.86 of False Negative (FN) in Table 5.5 indicates this. It is thus required for a tree to learn about the minority class as well.

For a tree to learn better with minority class, either the samples of minority class should be increased to the samples of majority class (oversampling), or the samples of minority class be made equal to majority class and then reduced to increase the quality of minority class samples (over-undersampling), or the samples of majority class should be reduced to the number of minority class samples (undersampling). All these techniques will in one way or the other make the samples of both classes nearly equal to the other and hence hold the potential to combat the imbalance problem.

5.3.2 Oversampling using Synthetic Minority Oversampling

Synthetic Minority Oversampling with Replacement (SMOTE) is an oversampling technique where extra training data is created by performing certain operations on the real data. As per the authors of the paper, operations like rotation and skew are natural ways to tweak the original data. Synthetic samples of minority class are generated along the line segments joining any or all of the k minority class nearest neighbors (Chawla, Bowyer, Hall, Kegelmeyer- 2002[9]).

If the oversampling to be achieved is 300%, then three nearest neighbors from the five are chosen, and a synthetic sample is generated along each direction. The samples are generated as- taking the difference between the feature vector(sample) under consideration and its nearest neighbor and multiplying that with a randomly generated real number. Thus the synthetic sample lies between two specific features (Chawla, Bowyer, Hall, Kegelmeyer- 2002[9]).

Results

The SMOTE technique is applied on the complete training set. The shape of the training set after SMOTE is applied is shown below. Table 5.6 shows the shape of dataset after applying SMOTE on the training set.

Table 5.6: Shape of training set before and after applying SMOTE

Case	Non-burnt(0)	Burnt(1)
Before	3677502(98.54%)	54584(1.46%)
After	3677502(50%)	3677502(50%)

It is clear that SMOTE produce new synthetic samples of minority class equal to the difference between the number of majority class samples and minority class samples to make samples of both classes equal. And therefore the dataset size is also increased after applying SMOTE.

Table 5.7: Scores using SMOTE on Decision Tree Classifier

Score	Value
TPR	0.16
FPR	0.02

Table 5.8: Confusion Matrix using SMOTE on Decision Tree Classifier

Class	Non-burnt(0)	Burnt(1)
Non-burnt	900278	19098
Burnt	11433	2213

Table 5.9: Normalized confusion matrix using SMOTE on Decision Tree Classifier

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.98	0.02
Burnt	0.84	0.16

Table 5.8 shows the confusion matrix of data samples using SMOTE on decision tree classifier and Table 5.9 shows the normalized confusion matrix using SMOTE on decision tree classifier.The results using SMOTE are pretty much similar to what is obtained using plain decision tree classifier in the Section 5.3.1.

Pondering on this, it can be said that though the synthetic samples of minority class are produced, but they are generated along the already existing features and lie somewhere in between the original samples. Thus the samples are increased but the new samples do not possess any unique quality as they are generated at nearly same place of the original minority class samples. Thus the tree learns nearly same way without qualitative addition of minority class. On the other hand the majority

class has already a handful variability in its dataset. Thus again the classification of majority class turns out to be better without adding much to the classification accuracy of the minority class.

5.3.3 Over-undersampling using SMOTE+Tomek on Decision Tree

This is a method of applying Tomek links to the oversampled training set as a data cleaning method. Instead of removing only the majority class samples that form Tomek links, examples from both the classes are removed. First the original dataset is oversampled with SMOTE, then Tomek links are identified and removed, producing a well-defined class clusters (Batista, Prati, Monard[6]).

Tomek Link (T- Link)

If d(x,y) is the distance between two points x belonging to class A and y belonging to class B, then (x,y) is a Tomek link if for any point z, d(x,y) $<$ d(x,z) or d(x,y) $<$ d(y,z). And if any two examples are tomek links, then either one of them is noise or both of them could be located on the boundary classes (Batista, Prati, Monard[6]).

Results

Table 5.10: Shape of training set before and after applying SMOTETomek

Case	Non-burnt(0)	Burnt(1)
Before	3677502(98.54%)	54584 (1.46%)
After	3677502(50.1%)	3669695(49.9%)

From the Table 5.10, it is clear that SMOTE+Tomek produces synthetic samples nearly equal to the difference between number of majority class and minority class samples. This can be understood as first applying SMOTE and then do cleaning. The cleaning accounts for 0.2% difference in the majority and minority class after SMOTE+Tomek is applied.

Table 5.11: Scores using SMOTE+Tomek on Decision Tree Classifier

Score	Value
TPR	0.17
FPR	0.02

Table 5.12: Confusion Matrix using SMOTETomek on Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	900182	19194
Burnt	11380	2266

Table 5.13: Normalized confusion matrix using SMOTE+Tomek on Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.98	0.02
Burnt	0.83	0.17

Table 5.11 shows the TPR and FPR and Table 5.12 and Table 5.13 shows the confusion matrix scores. The results using this technique also resemble to the scores using SMOTE for the same reason that the synthetic samples add samples on the lines of already existing samples without adding much to the quality of the minority class and hence does not serve the purpose.

5.3.4 Over-undersampling using SMOTE+ENN on Decision Tree

The intuition behind this technique is similar to the above used technique SMOTE+TOMEK. Here instead of Tomek links, Edited Nearest Neighbor (ENN) removes any example that is misclassfied by its three nearest neighbors. (Batista, Prati, Monard[6]).

Table 5.14: Shape of training set before and after applying SMOTE+ENN

Case	Non-burnt(0)	Burnt(1)
Before	3677502(98.53%)	54584(1.48%)
After	3677502(51.82%)	3507629(48.81%)

Results

From the Table 5.14, we can see that SMOTE first over-samples the minority class and then ENN is applied for cleaning the misclassfied samples. Thus ENN in SMOTE+ENN accounts for 3.01% difference between the two classes. Also, we can see that ENN does more dense cleaning than Tomek as the difference between the 2 classes using SMOTE+ENN is more than SMOTE+Tomek.

Table 5.15 shows the TPR and FPR scores. TPR does not represent a good score and FPR represents a good score. Table 5.16 and Table 5.17 shows the confusion matrix scores, which again show a better classification for the majority class but not the minority class.

Table 5.15: Scores using SMOTE+ENN on Decision Tree

Score	Value
TPR	0.15
FPR	0.02

Table 5.16: Confusion Matrix using SMOTE+ENN on Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	904684	14692
Burnt	11651	1995

Table 5.17: Normalized Confusion Matrix using SMOTE+ENN on Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.98	0.02
Burnt	0.85	0.15

5.3.5 Undersampling using Random-undersampling on Decision Tree

This technique under-samples the majority class by randomly picking samples with or without replacement (Scikit- Random undersampling[45]).

Results

Table 5.18 shows the results after applying Random-undersampling to the training dataset. Thus the samples of the minority class and majority class become equal. Also, the size of the dataset is reduced to twice the size of the original minority class.

Table 5.18: Shape of training set before and after applying Random- undersampling

Case	Non-burnt(0)	Burnt(1)
Before	3615236(98.51%)	54584(1.48%)
After	54584(50%)	54584(50%)

Table 5.19 shows the scores of Random-undersampling on Decision Tree. The value of TPR is promising, unlike the values calculated using other techniques. Though FPR, has increased from the values calculates using the earlier techniques, but still it can be counted as a good score.

Table 5.19: Scores using Random-undersampling on Decision Tree

Score	Value
TPR	0.72
FPR	0.29

Table 5.20: Confusion Matrix using Random-undersampling on Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	642606	261204
Burnt	3820	9826

Table 5.21: Normalized Confusion Matrix using Random-undersampling on Decision Tree

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.71	0.29
Burnt	0.28	0.72

Table 5.20 and Table 5.21 shows the values confusion matrix on data samples and normalized confusion matrix obtained after applying Random-undersampling on Decision Tree Classifier. Using this technique, the number of samples of the majority class are considerably reduced and become equal to the the size of the minority class. Thus the variability in features of both the classes would be nearly similar as now the number of majority class samples are reduced to the number of minority class samples. From the tables, it is clear that diagonal elements are more than 0.7 (close to one) and off-diagonal elements are nearly 0.28 (close to zero). This technique works for the dataset and hence can be used in consequent sections in identifying the best supervised learning model.

5.4 Identifying Best Supervised Learning Model

This section will identify the best supervised learning model which can be used for the dataset after applying the Random-undersampling technique to the training set.

5.4.1 Logistic Regression with Random-undersampling

Logistic Regression is a binary logistic model which is used to estimate the probability of a binary response based on one or more predictor (independent) features. (David A. Freedman (2009)[11]).

Results

These results are obtained after applying Random-undersampling on the training set. Table 5.22 shows TPR and FPR values using this technique. The TPR is little more than half and (1 - FPR) is a good value.

Table 5.22: Scores using Logistic Regression

Score	Value
TPR	0.66
FPR	0.39

Table 5.23: Confusion Matrix using Logistic Regression

Class	Non-burnt(0)	Burnt(1)
Non-burnt	560244	359132
Burnt	4658	8988

Table 5.24: Normalized Confusion Matrix using Logistic Regression

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.61	0.39
Burnt	0.34	0.66

Table 5.23 and Table 5.24 show the confusion matrix scores on data samples and normalized confusion matrix respectively. The diagonal valus shows the true prediction. The minority class is classified 66% correctly and the majority class is 61% classified correctly. Thus looking to the scores, it seems that our problem is more than a linear dependence on its features and a complex classifier might give better scores.

5.4.2 SGDClassifier

SGDClassifier implements a plain stochastic gradient descent learning routine and supports different loss functions and penalties for the classification. The results shown below are using two types of SGDClassifier, one with 'hinge' loss and the other with 'log' loss. 'Hinge' loss gives a linear SVM. The 'log' loss gives a logistic regression, a probabilistic classifier (Scikit- SGDClassifier[48]).

Results using 'hinge' loss

Table 5.25 shows the TPR and FPR scores using SGDClassifier with 'hinge' loss after applying random-undersampling on the training set. The TPR score indicates a good value, but FPR score is far from a realistic value.

Table 5.25: Scores using SGDClassifier with 'hinge' loss

Score	Value
TPR	0.94
FPR	0.86

Table 5.26: Confusion Matrix using SGDClassifier with 'hinge' loss

Class	Non-burnt(0)	Burnt(1)
Non-burnt	127111	792265
Burnt	802	12844

Table 5.27: Normalized Confusion Matrix using SGDClassifier with 'hinge' loss

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.14	0.86
Burnt	0.06	0.94

Table 5.26 shows confusion matrix scores of data samples and Table 5.27 shows normalized confusion matrix scores. The minority class is classified with 94% accuracy but majority class samples are classified incorrectly. Thus this classifier is not apt for this dataset.

Results using 'Log' Loss

Table 5.28 displays the results of SGDClassifier with 'log' loss using random-undersampling on training set. The TPR value is good but the FPR values are dead poor.

Table 5.28: Scores using SGDClassifier with 'log' loss

Score	Value
TPR	0.94
FPR	0.78

Table 5.29: Confusion Matrix using SGDClassifier with 'log' loss

Class	Non-burnt(0)	Burnt(1)
Non-burnt	205540	713836
Burnt	884	12762

Table 5.30: Normalized Confusion Matrix using SGDClassifier with 'log' loss

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.22	0.78
Burnt	0.06	0.94

Table 5.29 shows the confusion matrix scores of data samples and Table 5.30 shows the normalized confusion matrix scores using SGDClassifier with 'log' loss. The diagonal value of the majority class is not acceptable, though for the minority class it has classified better. Thus the problem looks more than a linear problem.

5.4.3 Tuning the Decision Tree Classifier

Scores with Decision Tree using random-undersampling are already written in Section 5.3.5. This part does the tuning of the same decision tree so as to improve on the scores. For tuning the classifier, GridSearchCV (Sklearn- GridSearchCV[47]) is used over the various hyper-parameters.

Table 5.31 displays the set of values used for the various hyperparameters. This search runs over 1440(5*4*6*6*2) candidate parameter settings.

Table 5.31: Parameters grid to tune the Decision Tree Classifier

Parameters	Value
max_features	1,2,3,4,5
max_depth	1,5,10,20
min_samples_split	2, 10, 50, 100, 200, 500
min_samples_leaf	2, 10, 50, 100, 200, 500
criterion	gini, entropy

Table 5.32 shows the best values of hyperparameters obtained for the Decision Tree classifier.

Table 5.32: Decision Tree Model with Rank 1

Features	Value
Mean validation score	0.737
std	0.001
max_features	5
max_depth	20
min_samples_split	100
min_samples_leaf	50
criterion	entropy

Table 5.33: Confusion Matrix using Random-undersampling on Tuned Decision Tree Classifier

Class	Non-burnt	Burnt
Non-burnt	671325	248051
Burnt	3450	10196

Table 5.34: Normalized Confusion Matrix using Random-undersampling on Tuned Decision Tree Classifier

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.73	0.27
Burnt	0.25	0.75

Table 5.33 shows the confusion matrix of data samples obtained using the tuned decision classifier and Table 5.34 shows the normalized values of confusion matrix

using the tuned decision tree classifier. Comparing Table 5.34 with Table 5.21, it can be noticed that there is a change. The accuracy of minority class increased from 0.72 to 0.75 and the majority class from 0.71 to 0.73. Thus tuning a classifier appreciably increases the accuracy by 0.02 for the minority class and 0.03 for the majority class.

Figure 5.1 shows the ROC curve generated using the tuned decision tree classifier. The value AUC (Area Under Curve) is 0.83 which identifies the classifier as a good classifier (UNMC- The Area Under an ROC Curve[52]).

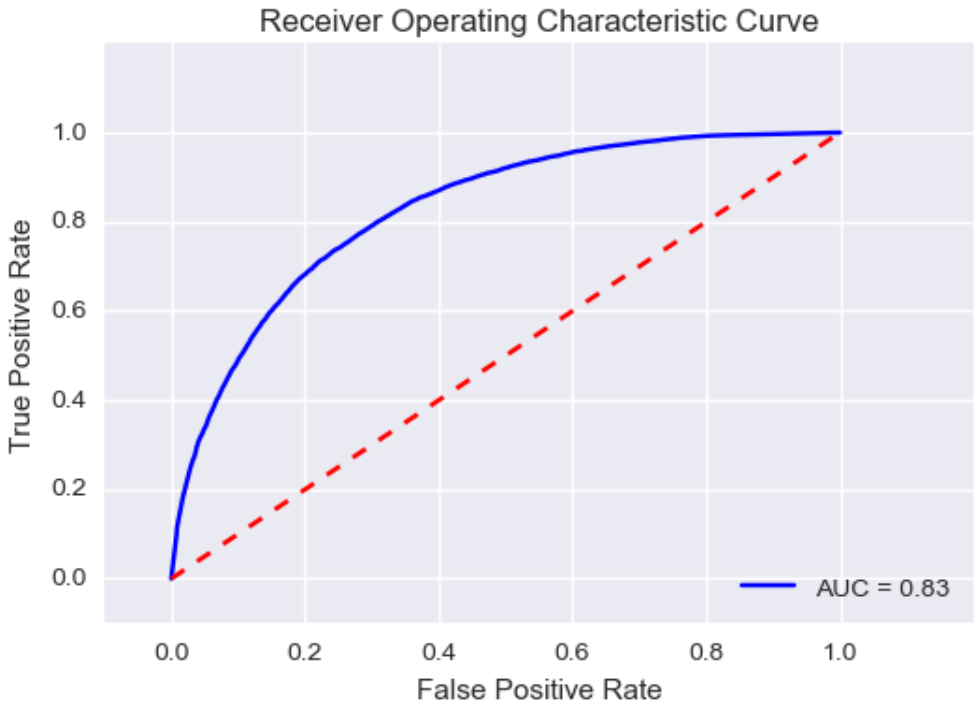

Figure 5.1: ROC curve using Random-undersampling on Decision Tree

5.4.4 Random Forest with Random-undersampling

Random Forest is an ensemble learning method for classification that uses a variable number of decision trees to train the data and outputs the class which is the mode of the classes of the individual trees (Ho, Tin Kam (1995)[24]).

Results

Table 5.35 shows the TPR and FPR scores generated using the Random Forest after applying Random-undersampling on training set. The scores are comparable with those obtained using a decision tree as shown in Table 5.19, with a little bit difference. Table 5.36 shows the confusion matrix scores of data samples and Table

5.37 shows the normalized confusion matrix scores using the random forest as a learner.

Table 5.35: Scores using Random-undersampling on Random Forest

Score	Value
TPR	0.72
FPR	0.24

Table 5.36: Confusion Matrix using Random-undersampling on Random Forest

Class	Non-burnt(0)	Burnt(1)
Non-burnt	695664	223712
Burnt	3870	9776

Table 5.37: Normalized Confusion Matrix using Random-undersampling on Random Forest

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.76	0.24
Burnt	0.28	0.72

The scores of the confusion matrix are bit better than Decision Tree Classifier scores (check Table 5.20 and Table 5.21). To obtain still better scores, the Random Forest is tuned using the hyper-parameters of the best Decision Tree Classifier, number of estimators and bootstrap settings. The Table 5.38 shows the parameter grid on which GridSearchCV is performed[47] to find the best random forest hyper-parameter.

Table 5.38: Parameters grid to tune the Random Forest

Parameters	Value
n_estimators	1, 10, 20, 30, 40, 50,60,70, 80
max_features	5
max_depth	20
min_samples_split	100
min_samples_leaf	50
criterion	entropy
bootstrap	True, False

The GridSearchCV thus runs over 18(9*1*1*1*1*1*2) candidate parameter settings. The best ranked Random Forest has the hyper-parameters shown in the Table 5.39:

Table 5.40 shows confusion matrix of data samples and Table 5.41 shows normalized confusion matrix scores on tuned random forest. There is a little bit

Table 5.39: Random Forest Model with Rank 1

Features	Value
mean validation score	0.738
std	0.001
n_estimators	70
max_features	5
max_depth	20
min_samples_split	100
min_samples_leaf	10
criterion	entropy
bootstrap	true

Table 5.40: Confusion Matrix using Random-undersampling on tuned Random Forest

Class	Non-burnt(0)	Burnt(1)
Non-burnt	673898	229912
Burnt	3002	10644

Table 5.41: Normalized Confusion Matrix using Random-undersampling on tuned Random Forest

Class	Non-burnt(0)	Burnt(1)
Non-burnt	0.75	0.25
Burnt	0.22	0.78

improvement on the scores, when compared with Table 5.36 and Table 5.37. Figure 5.2 shows the ROC curve using the tuned random forest. The Area Under the curve (AUC) is 0.85 which is better than the area obtained using the decision tree classifier (Area:0.82, Refer Figure 5.1).

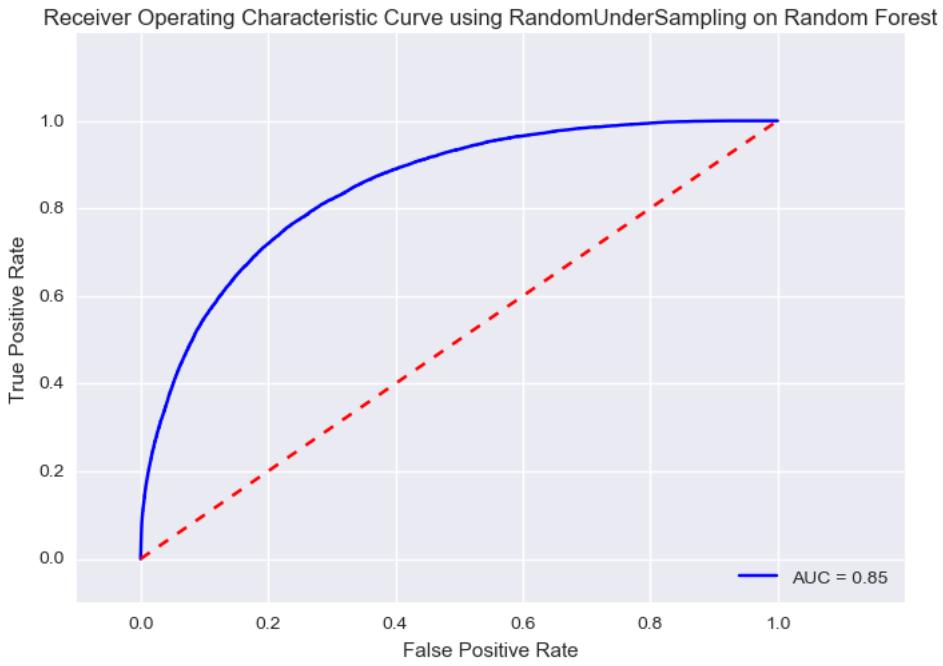

Figure 5.2: ROC curve using Random-undersampling on Random Forest

Chapter 6

Evaluation

The previous chapter built models and applied resampling techniques on the dataset. This chapter talks and evaluates those techniques. The dataset is too big and too skewed. Hence to solve the imbalancing problems, oversampling, over-undersampling and undersampling are used. And then out of the techniques, which give best results, a supervised learning model is applied using the best chosen resampling technique.

Oversampling and over-undersampling techniques do not give the desired results. The results are promising using the undersampling technique. The reason could be the synthetic production of samples is not that good to contain the features of the minority class properly and as a result the classification is again skewed. While using over-undersampling techniques, though cleaning is done using Tomek links and EEN for the misclassified samples produced during Oversampling, but it is still not sufficient. The cleaning only makes a difference of 0.1-2.76% between the majority and minority class. Thus, it can be further said that if the dataset is too big and the skewed like this, oversampling and over-undersampling techniques do not work well. Had the cleaning accounted for a considerable difference, qualitative samples would have been produced owing to better classification results, but this does not happen.

Undersampling using random-undersampling gives promising results. True positive rates increase and (1 -False Positive Rates) increase. Though there are various other undersampling techniques that would have applied engineering to reduce the samples of majority class in a qualitative way so as to cover all the facets of majority class, but again the dataset is too big and these techniques do not work out, keeping the system in a 'hanged' state and will require a long time to train via such techniques.

The results using random-undersampling with Decision Tree and Random Forest are almost comparable with slight difference and being better with Random Forests.

A classifier with a high True Positive Rate and high (1- False Positive Rate) is considered as a good classifier. Comparing rates using Decision Tree and Random Forest, we found that True Positive Rate with Decision Trees is 0.72 (Table 5.19) while with Random Forest is also 0.72 (Table 5.35). False Positive Rate with Decision Tree is 0.29 (Table 5.19), and with Random Forest is 0.24 (Table 5.35). Moreover the area under the curve of ROC using random forest (Figure 5.2) is slightly greater in random forest than decision tree (Figure 5.1). Thus Random Forest can be considered better.

On analyzing the samples which are wrongly classified, it is found that these have mostly less fire frequencies (0-1), Elevation ranging from medium to high, less Slopes from low to medium and Aspect values mostly very less or very high. Going further we can understand the trend of this outliers and add weightage to these outliers to make the machine more robust.

Chapter 7

Conclusion

The project solves a real world problem using real dataset. It is aimed to build a system which can analyze and predict the occurrence of wildfires. This was achieved with an accuracy of more than 70% which can predict fires potential of an area when its landscape features are provided. The system is built by training with a historical data of past twenty years (1992-2012) of fire history and landscape features of two years back (2012). The system predicts the output of fires for year 2014.

The data is collected from various sources. And the data from varied sources are in different projections. The data is geo-referenced, so that they can be mapped with each other. After the data is mapped, exploratory data analysis is done to get more insights into data. On exploring, it is determined that the dataset faces an imbalance problem. One class is represents 98.5% of the total data and the other just represents the 1.5% of total.

The imbalance problem is solved by using various resampling techniques, and it is the undersampling technique which works well. The project uses random undersampling which reduces the size of majority class by choosing random samples from the majority class to eventually make the size of both the classes same. The resampling technique is then applied to various supervised learning models. Due to the huge size and bias in the dataset, not all the scores can be used to evaluate the accuracy of models. The accuracy scores, Precision and Recall Curve, F1-score cannot be used as these scores will hide the affect of the minority class due to the substantial size of majority class. Confusion Matrix values, True Positive Rate (TPR), False Positive Rate (FPR) and area under the ROC curve are used as evaluators.

The decision trees and random forests provide an accuracy of more than 70%

for both the classes. The percentage of accuracy further increases when tuning is done by providing a grid of parameters. The area under the ROC curve is 0.82-0.85. Thus, a system is successfully built accomplishing the goals.

The system is a step towards predicting wildfires by feeding historical data and current data. Hence, the approach presented in this thesis can be thought as a potential solution to assess the risk of Wildfires in the coming future.

Chapter 8

Future Work

The project aims to predict and analyze the wildfires. There are multiple ways where it can be thought to improve and make the system even better and measures which could be taken after predicting the results.

The first could be including more features into the dataset. These features can be weather conditions- temperature, pressure, rainfall, humidity level etc. High temperatures, dry conditions, high pressure help in easing the happening of wildfires. Apart from this, an additional feature can be human vicinity, which is distance of human accessibility for a point. This will be even more deeper as fire initiates from a point and rip the complete forest, and these points can be the boundaries of the areas which got burnt or also the points which lie within an area, but have people accessibility.

Digging deeper into the project, and considering techniques used within the project. The dataset faces an imbalance problem. And this will be a perpetual problem, no matter how many additional features are added. The imbalance problem in the project is resolved using random undersampling, but there could be numerous smarter ways by which a majority class can be represented, given a better hardware, such as representing the class using the maximum frequency features. The accuracy can also be improved by switching to more complex supervised learners, such as Support Vector Machines, or using Neural Networks (multilayer perceptron).

This system is build using around 35.22% of San Diego county. The system could have wide application, when fed with more training data, data of varied places and topography. Also, system can be continuously updated by feeding in more data and adding more weightage to the outliers, if wildfires hold a significant reasoning for the cause of wildfires.

Appendix A

Fuel Model Descriptions

This appendix is prepared using Anderson- Aids to Determining Fuel Models[1].

Grass Group

1. Fire Behavior Fuel Model 1 (Short Grass)
   Very fine herbaceous fuels, little shrub or timber is present. Grassland and savanna are represented along with stubble, grass- tundra and grass-shrub combinations are found. The fire spread is caused majorly due to fine, porous herbaceous fuels.

2. Fire Behavior Fuel Model 2 (Timber)
   Fire spread is caused primarily due to herbaceous fuels, cured or dead.

3. Fire Behavior Fuel Model 3 (Tall grass)
   This model have most intense fires in the grass group. The fire is caused due to upper heights of grass and cross standing water.

4. Fire Behavior Fuel Model 4 (Chaparral)
   Fire intensity and fast spreading fires involve the foliage and live dead fine woody material in the crowns. Also, there is a dead woody material in the stand that also contributes to fire intensity.

5. Fire Behavior Fuel Model 5 (Brush)
   Fire is caused due to surface fuels that are made up of litter cast by the shrubs, and the grasses. The fires are generally not much intense.

6. Fire Behavior Fuel Model 6 (Dormant Brush)
   Fire carries through the shrub layer where foliage is more flammable than FBFM5, and requires moderate winds

7. Fire Behavior Fuel Model 7 (Southern rough)
   Fire burn through the surface and shrub and can occur at higher dead fuel moisture contents.

8. Fire Behavior Fuel Model 8 (Closed timber litter)
   Low flames with slow burning fires. This layer consists of needles, leaves and some twigs.

9. Fire Behavior Fuel Model 9 (Hardwood litter)
   Fire runs faster as compared to FBFM8. Fall fires in hardwoods are representative, but high winds will cause higher rates of spread.

10. Fire Behavior Fuel Model 10 (Timber(litter and understory))
    Fire run in the surface and ground fuels with great fire intensity than other timber models. Crowinng out, spotting is more common in this fuel.

11. Fire Behavior Model 11 (Light logging slash)
    Fire is active in slash and herbaceous fuel amalgamated in slash.

12. Fire Behavior Model 12 (Medium logging slash)
    When fire happen, it is not stopped until a fuel break. Reason of fire is mostly slash.

13. Fire Behavior Model 13 (Heavy logging slash)
    Fire happen due to presence of lime fuels and intensity builds up more slowly as large fuels start burning.

Bibliography

[1] *Anderson- Aids to Determining Fuel Models For Estimating Fire Behavior.* URL: https://training.nwcg.gov/pre-courses/s290/S-290%20Student%20CD/Aids%20to%20Determining%20Fuel%20Models.pdf.

[2] *ArcGis- File Geodatabase.* URL: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/administer-file-gdbs/file-geodatabases.htm.

[3] *ArcGis Geodatabases.* URL: http://desktop.arcgis.com/en/arcmap/10.3/manage-data/geodatabases/types-of-geodatabases.htm.

[4] *ArcGis- Raster Data.* URL: http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=What_is_raster_data?.

[5] *ArcMap- How Aspect Works.* URL: http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/how-aspect-works.htm.

[6] *Batista, Prati, Monard- A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data.* URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.7757&rep=rep1&type=pdf.

[7] *Burke, Donald; Brungade, John; Redfield, Robert(1998)- Measurement of the False Positive Rate in Screening Program for Human Immunodeficiency Virus Infections.*

[8] *Canadian Wildland Fire Information System- Natural Resources Canada.* URL: http://cwfis.cfs.nrcan.gc.ca/background/summary/fwi.

[9] *Chawla, Bowyer, Hall, Kegelmeyer-SMOTE.* URL: https://www.jair.org/media/953/live-953-2037-jair.pdf.

[10] *Craig, Cindy (October 2013) here Theory Meets Practice: United States Standard Datum.* URL: file:///home/sumedha/Downloads/cgs_afe_12.pdf.

[11] *David A. Freedman(2009)- Statistical Models: Theory and Practice. Cambridge University Press.*

[12] *FARSITE- US Forest Service.* URL: https://www.firelab.org/project/farsite.

[13] *FlamMap- US Forest Service.* URL: https://www.firelab.org/project/flammap.

[14] *FRAP- AboutUs.* URL: http://frap.fire.ca.gov/about/index.

[15] *GDAL- Documentation.* URL: http://www.gdal.org/.

[16] *GDAL- GDAL Information.* URL: http://www.gdal.org/gdalinfo.html.

[17] *GDAL- GDAL Translate.* URL: http://www.gdal.org/gdal_translate.html.

[18] *GDAL- LCP format.* URL: http://www.gdal.org/frmt_lcp.html.

[19] *GDAL- Rasterize.* URL: http://www.gdal.org/gdal_rasterize.html.

[20] *GDAL- Warp.* URL: http://www.gdal.org/gdalwarp.html.

[21] *Geography Field Work-Slope Steepness.* URL: http://geographyfieldwork.com/SlopeSteepnessIndex.htm.

[22] *GIS Geography- Geodetic Datum.* URL: http://gisgeography.com/geodetic-datums-nad27-nad83-wgs84/.

[23] *H Wang, H Zheng- Encyclopedia of Systems Biology, 2013 - Springer.*

[24] *Ho, Tin Kam(1995). Random Decision Forests. Procedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC.* URL: http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf.

[25] *J. R. Quinlan- Simplifying Decision Trees.* URL: http://www.sciencedirect.com/science/article/pii/S0020737387800536?via%5C%3Dihub.

[26] *Joe H. Scott, Robert E. Burgan- Standard Fire Behavior Fuel Models- A Comprehensive Set for Use with Rothermel's Surface Fire Spread Model.* URL: https://www.fs.fed.us/rm/pubs/rmrs_gtr153.pdf.

[27] *Kent, A.A, Berry, M.M., Luehrs, F.U. Jr Perry (1955). Operational criteria for designing informational retreival systems. American Documentation, 6(2), 1936-6108.*

[28] *Kohavi, R., and Provost, F. 1998. On Applied Research in Machine Learning. In Editorial for the Special Issueon Applications of Machine Learning and the Knowledge Discovery Process, Columbia University, New York, volume 30.*

[29] *Landfire- Data Downloads.* URL: https://landfire.gov/viewer/.

[30] *Landfire- Fire Behavior Fuel Model 13 Data Dictionary.* URL: https://landfire.gov/DataDictionary/f13.pdf.

[31] *Landfire- Main.* URL: https://landfire.gov/.

[32] *Landfire- Topogrpahic.* URL: https://landfire.gov/topographic.php.

[33] *Manfred F. Buchroithner- Geodetic grids in authoritative maps – new findings about the origin of the UTM Grid.* URL: http://www.tandfonline.com/doi/abs/10.1080/15230406.2015.1128851.

[34] *Metz CE(1978) Basic Principles of ROC Analysis.* URL: https://www.ncbi.nlm.nih.gov/pubmed/112681.

[35] *National Avalanche Center- Aspect.* URL: http://www.fsavalanche.org/aspect/.

[36] *National Geographic- GIS.* URL: https://www.nationalgeographic.org/encyclopedia/geographic-information-system-gis.

[37] *National Geographic- Wildfires.* URL: http://www.nationalgeographic.com/environment/natural-disasters/wildfires/.

[38] *National Geospatial-Intelligence Agency.* URL: https://web.archive.org/web/20120402143802/https://www1.nga.mil/ProductsServices/GeodesyandGeophysics/WorldGeodeticSystem/Pages/default.aspx.

[39] *NOAA History- Geodetic Surveying 1940-1990.* URL: http://www.history.noaa.gov/stories_tales/geod1.html.

[40] *Paulo Cortez and Anibal Morais- A Data Mining Approach to predict Forest Fires using Meteorological Data.* URL: http://www3.dsi.uminho.pt/pcortez/fires.pdf.

[41] *QGIS- Coordinate Reference System.* URL: https://docs.qgis.org/2.2/en/docs/gentle_gis_introduction/coordinate_reference_systems.html.

[42] *QGIS Documentation- Introducing GIS.* URL: https://docs.qgis.org/2.6/en/docs/gentle_gis_introduction/introducing_gis.html.

[43] *QGIS- Raster Data.* URL: http://docs.qgis.org/2.0/en/docs/gentle_gis_introduction/raster_data.html.

[44] *Richard Rothermal- A mathematical model for predicting fire spread in wildland fuels.* URL: https://www.fs.fed.us/rm/pubs_int/int_rp115.pdf.

[45] *Scikit- Imbalanced Learning, Random-under Sampling.* URL: http://contrib.scikit-learn.org/imbalanced-learn/generated/imblearn.under_sampling.RandomUnderSampler.html.

[46] *Shalowitz, USCGS Field Engineers Bulletin, December 1938.* URL: http://archives.profsurv.com/magazine/article.aspx?i=71429.

[47] *Sklearn- GridSearchCV.* URL: http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.

[48] *Sklearn- SGDClassifier.* URL: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

[49] *Sklearn- Train test split.* URL: http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[50] *Snyder, J.P. (1989). Album of Map Projections, United States Geological Survey Professional Paper. United States Government Printing Office. 1453.* URL: https://pubs.usgs.gov/pp/1453/report.pdf.

[51] *Stehman, Stephen- Selecting and interpreting measures of thematic classification accuracy.* URL: http://www.sciencedirect.com/science/article/pii/S0034425797000837?via%5C%3Dihub.

[52] *UNMC- The Area under ROC Curve.* URL: http://gim.unmc.edu/dxtests/roc3.htm.

[53] *Van Rijsbergen- F1 Score, C. J. Information Retreival 2nd Edition Butterworth (1979).*

[54] *Wei-Yin Loh- Classification and Regression Trees.* URL: http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf.

[55] *WGS84 NASA- The EGM96 Geoid Undulation with Respect to the WGS84 Ellipsoid NASA.* URL: https://cddis.nasa.gov/926/egm96/doc/S11.HTML.

[56] *Wikipedia- San Diego County.* URL: https://en.wikipedia.org/wiki/San_Diego_County,_California.

[57] *Zweig MH Campbell, Receiver Operating Characteristic(ROC) plots: a fundamental evaluation tool in clinical medicine.* URL: https://www.ncbi.nlm.nih.gov/pubmed/8472349.