# An Interactive Map for Forest Fire Zones Prediction in California

Nghi M. Nguyen

College of Professional and Global Education, San Jose State University

DATA 270: Data Analytics Process

Dr. Eduardo Chan

December 10, 2021

**Abstract**

In California, wildfires have become undeniable environmental threats that lead to catastrophic results on both human lives and the economy. In fact, most fires recorded in California are caused by humans. Therefore, this study aims to introduce a solution to predict future potential fire zones via a web-based interactive map visualization. The goal is to predict potential zones down to county levels that are potential threats to wildfire given necessary factors like human or natural causes, human-related activities, etc. The data used for this study is historical fire data from USDA. We explore Naive Bayes, k-nearest neighbors and one-vs-rest as machine learning approaches to solve this problem. Besides utilizing scikit-learn's libraries, we also implement Naive Bayes from scratch. The results show that the Naive Bayes model from scratch is the best approach with the highest average accuracy of 0.73 and macro F1 score of 0.35 and therefore will be used to set a starting baseline for accuracy score. In conclusion, with Naive Bayes model we have successfully set up a good accuracy foundation for this problem; with this application, we can not only predict the most potential fire zones with certain factors, we can potentially learn more about fire patterns that are caused by humans. Therefore we can expand this and integrate other alert technologies to notify authorities in order to make better decisions to prevent future incidents.

*Keywords:* wildfire prediction, predictive visualizations, naive bayes, knn, one-vs-rest, machine learning, multi-class classification.

# Chapter 1 - Introduction

## 1.1 - Project Background

### *Problems and Motivations*

Wildfires, nowadays, have become unquestionable environmental threats to our societies with disastrous impacts on climate,  human lives and economy. For instance, the August Complex fire destroyed 1,032,648 acres and 1,329 structures, the Dixie fire destroyed 960,335 acres and 935 structures, and so many more (Cal Fire, 2020). Plus, wildfires caused by humans have become an increasingly and catastrophically alarming trend. According to Prestemon and Butry (2010), human-caused fires are an alarming concern because they have contributed to the majority of wildfires and have turned into a trend in climatic factors for fire settings.

Furthermore, according to the USDA's California fire historical data from 1997 to 2018, there were a total of 142,551 fire incidents. Within them, 70.72% were caused by humans, 8.91% were classified as natural causes and 20.37% were unclassified. Furthermore, as we drill down into the data, in this case we can look into the top 100 largest wildfires each year, 68.06% of them were human-caused (Short, 2021). In other words, 70.72% of these incidents were preventable or, at least, they were reducible. With that being said, this project is motivated by the ideas of preventing potential fire incidents by relying on a machine learning model to make predictions of potential fire zones based on multiple factors such as nature or human causasions, human-related activities, fire seasons, and so on.

### *Expected Project Contributions and Applications*

This project aims to introduce an interactive visualization for predicting fire zones in California. The goal of this visualization is to present potential fire zones down to county levels. In addition, this map will allow users to see fire zone predictions based on a combination of features such as seasons, date ranges, time, and further human-related activities. Thus, by

predicting potential fire zones, we can have more understanding of fire patterns and therefore potentially prevent future incidents.

***Project Approaches and Methods***

Following machine learning models are considered as potential solutions for this project:

- Naive Bayesian (from scratch and scikit-learn's)
- K-nearest neighbors (KNN)
- Artificial neural networks (ANN)
- One-vs-rest

The goal of the model is to predict the most potential fire zones of each area given necessary features. The expected output of the model should reflect the predicted zones at county levels, FIPS codes, with the probability of fire likelihood. Then, we can map these outcomes onto a web-based interactive map.

**1.2 - Project Requirements**

***Functional Requirements***

Final product deliverable will be a web-based interactive map visualization with main capability is to present potential fire zones down to county levels according to prediction outcomes. Users will be able to modify their searches using filters to make predictions based on a combination of features. Filters should include following features:

- Season
- Time range of day: day, afternoon, evening
- Month
- Causation (human or nature)
- Human related causations such as arson, fireworks, etc.

***Machine Learning Model Requirements***

The desired machine learning model for this project will be a supervised multi-class classification machine learning model to predict multiple potential firezones. The model will

predict the most potential fire zones of each area or region down to county levels (represented as FIPS codes). The model will be looking at potential features from historical fire data such as discovery date, discovery season, discovery time ranges, human activities, etc. Table 1 describes the desired model's outcome outputs.

**Table 1**

*Expected Model Probability Outcomes*

| FIPS Codes (County) | Probability (%) |
|---|---|
| 06079 | 20 |
| 06085 | 18 |

*Note.* Expected outputs for the model.

***Data Requirements***

Dataset collected for this project need to satisfy following conditions:

- Dataset contains daily GIS historical fire data in California.
- Dataset contains fire causations either with human causes and natural causes.
- Dataset contains data for human activities related to the fires.

***Data Source***

The dataset used for this project will be the historical wildfire data for the United States dataset by USDA.

**1.3 - Project Deliverables**

Deliverables for this project will include the following:

- Individual research reports from chapter one to chapter four.
- Final research report.
- Final presentation.
- Project codes.
- A web-based interactive map with capability of fire zones prediction in California.

### 1.4 - Technology and Solution Survey

Four possible solutions that can be implemented for this problem, namely naive Bayes, artificial neural network, k-nearest neighbor and one-vs-rest.

#### *Naive Bayes*

**Approaches.** According to Kaviani & Dhotre (2017), Naive Bayes is a part of Bayesian theory, the model makes naive assumptions about independent features, hence the name Naive Bayes. Naive Bayes classifier is described as equation 1.

$$\Pr(c, x_1, x_2, \ldots, x_m) = \Pr(c) \cdot \prod_{i=1}^{m} \Pr(x_i \mid c) \quad (1)$$

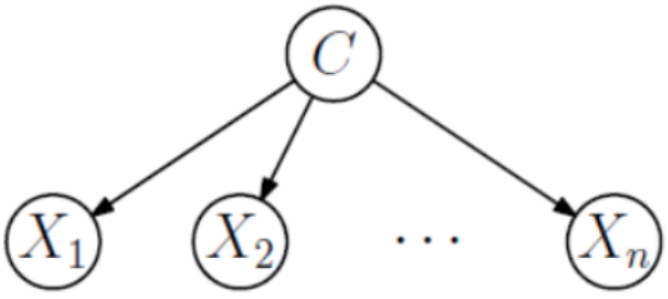
Where c is the class label, $x_i$ is a value of a attribute $X_i$ , Pr(c) is the class prior and $\Pr(x_i \mid c)$ is the conditional probability of attribute $x_i$ given c. Then, these parameters from data can be estimated using maximum likelihood. Once the naive Bayes model classifies data, new label of class c* can be chosen using maximum posterior probability. c* can be computed as equation 2

$$c^* = argmax_{\ c} \Pr(c \mid x_1, \ldots, x_m) \quad (2)$$

A naive Bayes classifier can also be presented as a graph-like structure as in Figure 1.

**Figure 1**

*Naive Bayes Classifier*




*Note.* C is a class label and $x_1, x_2, ..., x_n$ are attributes *(Kaviani & Dhotre, 2017)*.

**Advantages and Disadvantages.** Kaviani and Dhotre (2017) mentioned naive Bayes's advantages and disadvantages in their research.

***Advantages.***

- The Naive Bayes model is simple to understand and implement.
- The Naive Bayes model has better prediction performance than other classification models.
- The Naive Bayes model can be trained easily with small datasets.

***Disadvantages.***

- Naive Bayes model makes strong assumptions of independence features.

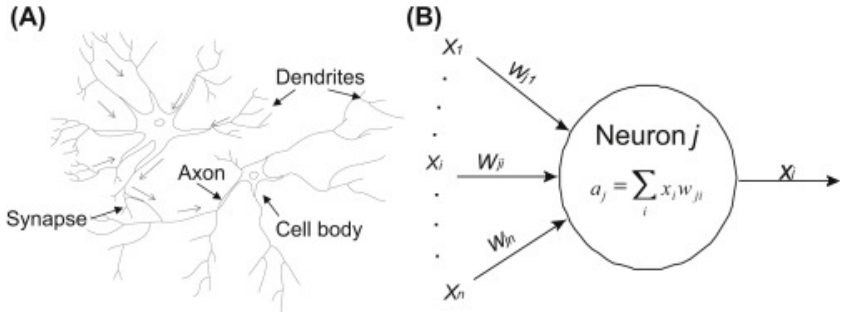
**Applications.** Naive Bayes has many popular applications for supervised learning. Some examples are text classification, spam classification, sentiment analysis, and recommendation systems (Kaviani & Dhotre, 2017).

***Artificial Neural Network (ANN)***

**Approaches.** In the book *Developments in Environmental Modelling*, Park and Lek (2016) presented ideas of artificial neural network concepts as depicted in Figure 2. The ANN concept was inspired by characteristics of the human brain in terms of how biological neural networks work. The two networks are similar in the way highly interconnected neurons control entire network functions. In the ANN model, inputs are represented as X = (x1, x2, …, xn), and each input has a relative weight W = (w1, w2, …, w3). Weights are coefficients that influence input signals. The output signal of a neuron is a summation of the multiplication of weights and input signal as shown in Figure 2B.

**Figure 2**

*Schematic Representation of Artificial Neural Network (Park and Lek, 2016)*




*Note.* Schematic representation of (A) a biological neural network and (B) artificial neural network.

One of the popular algorithms for ANN is Multi-layer Perceptron (MLP). MLP is a supervised learning algorithm that uses more than one hidden layer between input and output (Scikit Learn, n.d., 1.17. Neural network models (supervised)).

**Advantages and Disadvantages.** Following are some advantages and disadvantages of ANN (Ciaburro & Venkateswaran, 2017).

***Advantages.***

- ANN models are flexible and can be used for regression and classification problems.
- ANN models are a good model for non-linear data with large data points.
- Predictions are very fast once training is done.
- ANN models work for any number of layers and inputs.

***Disadvantages.***

- Training computations with traditional CPUs are expensive and time consuming.
- ANN training needs lots of data which can lead to overfitting and generalization problems.
- ANN models are back boxes which means we can't know the influences independent variables have on dependent variables.

**Applications.** ANN has quite a lot of applications in unsupervised learning such as speech recognition, computer vision, pattern recognition (Liu et al., 2017).

***K-nearest Neighbor (KNN)***

**Approaches.** According to the book *Data Algorithms,* Parsian (2015) explored the concepts of KNN, the KNN classifies n-dimensional objects to other n-dimensional objects based on their similarities. The idea is to build a classifier which makes no assumptions about the smooth function f(x), where x = ($x_1$, $x_2$, …, $x_n$), y = f(x). And given a new observation p = ($p_1$, $p_2$, …, $p_n$), nearest neighbors can be identified using the Euclidean distance which is defined as equation 3.

$$distance(X, P) = \sqrt{(x_1 - p_1)^2 + (x_2 - p_2)^2 + \ldots + (x_n - p_n)^2} \quad (3)$$

The KNN algorithm will calculate the Euclidean distance from an observation n-dimensional object to all training objects. Then, the observation object will be assigned a label which most k closest training data has.

Besides Euclidean distance, other distance calculation methods can be used for KNN such as Manhattan and Minkowski.

**Advantages and Disadvantages.** Sun & Shi (2018), Imandoust & Bolandraftar (2013) examined some advantages and disadvantages of KNN in their research.

***Advantages.***

- KNN models are more robust to noisy data and effective for large datasets.
- KNN models have more competitive performance for classification for many areas.
- KNN models are simple, effective and intuitive.
- KNN models are easy to understand and implement with no need to estimate parameters.
- KNN models are suitable for multi classification problems.

***Disadvantages.***

- KNN models can have poor runtime performance and memory overhead with large datasets.
- KNN models are very sensitive to irrelevant features.
- Distance based learning is unclear of the type of distance and features to use for best results.
- Computation cost is expensive due to distance computations of each instance to all training samples.

**Applications.** KNN is known to be rich in applications in both supervised and unsupervised learning. Some popular applications of KNN are recognized in pattern recognition, computer vision, data compression, DNA sequencing, spell check suggestion, plagiarism investigation, and many more (Sun & Shi, 2018).

***One-vs-rest***

**Approaches.** One-vs-rest or also known as one-vs-all is a multi-class prediction strategy that fits one classifier per class. Let K is the number of classes in the target feature, one-vs rest can create K binary SVM classifiers to distinguish one class from the other K-1 classes. This method is considered the most commonly used strategy and a good default choice (Aly, 2015; Scikit Learn, n.d, OneVsRestClassifier).

**Advantages and Disadvantages.** According to Aly (2015) and Scikit Learn (n.d, OneVsRestClassifier), there are some advantages and disadvantages for the one-vs-rest model.

***Advantages.***

- Easy to interpret since each class only has one classifier.

***Disadvantages.***

- Memory demand is high for large datasets.

**Applications.** This model is a good default choice for multi-class classification problems.

***Technology Comparison***

Table 2 and Table 3 show comparisons of three models in terms of learning applications, algorithms, advantages and disadvantages.

**Table 2**

*Machine Learning Models Comparison in Learning Applications and Algorithms*

| Models | Learning Application | Algorithms |
|---|---|---|
| ANN | Unsupervised | Input signals to each neuron are values of features and their influenced weights.<br>Output signals of each neuron are summation of the multiplication of weights and input signal.<br>Multi-layer perceptron MLP is a popular ANN algorithm that has one or more hidden layers between inputs and outputs. |
| Naive Bayes | Supervised | The model makes naive assumptions about independent features.<br>Classifier is described as the product of class prior probability and the product of conditional probability of each attribute given class information.<br>New labels can be chosen using maximum posterior probability. |
| KNN | Supervised and unsupervised | Classify a n-dimensional object to other n-dimensional objects based on similarities using k closet training data.<br>Utilize shortest distance calculation to compute closest neighbors such as Euclidean, Manhattan and Minkowski. |
| One-vs-rest | Supervised and unsupervised | One-vs rest can create K binary SVM classifiers to distinguish one class from the other K-1 classes |

*Note.* Technology survey comparison between learning application and algorithm.

**Table 3**

*Machine Learning Models Comparison in Advantages and Disadvantages.*

| Models | Advantages | Disadvantages |
|---|---|---|
| ANN | Flexible models for regression and classification.<br>Good for non-linear data.<br>Predictions are fast once training is done.<br>Models work for any number of layers and inputs. | Expensive computation cost and time consuming for traditional CPUs.<br>Prone to overfitting and generalization for large datasets.<br>We can't know the influences independent variables have on dependent variables. |
| Naive Bayes | Simple to understand and implement.<br>Better performance than other classification models.<br>Can be trained easily with small datasets. | Make strong assumptions of independent features. |
| KNN | Robust to noisy data.<br>Effective for large datasets.<br>Simple model to understand and implement.<br>More suitable for multi classification problems. | Poor runtime performance and memory overhead with large datasets.<br>Expensive computation cost due to distance computations.<br>Sensitive to irrelevant features. |
| One-vs-rest | Easy to interpret. | Memory performance for large datasets. |

*Note.* Technology survey comparison between advantages and disadvantages.

### 1.5 - Literature Survey of Existing Research

#### *Prediction of Forest fires with Artificial Neural Network*

**Introduction.** In the research *Prediction of Forest Fires Using Artificial Neural Networks*, Safi and Bouroumi (2013) explored prediction of forest fires using an artificial neural networks model, a multilayer perceptrons (MLP) architecture in particular. The input data used in the research is a set of twelve features from historical forest fire data, and the output signal is a single number representing the area of burned area in each instance.

**Data Collection and Training.** The data chosen for this research was a Canadian fire system dataset for rating fire danger. The researchers included the following twelve features for

model training: coordinates, month, day, fine fuel moisture code, duff moisture code, drought code, initial spread index, outside temperature, outside relative humidity, outside wind speed, outside rain, and total burned area. The study area was the Portuguese Montesinho natural park with the observation time period from January 2000 to December 2003.

For numbers of hidden layers, researchers didn't have a fixed number, but instead they tried several topologies and used total error rate (ER) to compare. The topology A had two hidden layers with twelve and six neurons respectively, and the topology B had one hidden layer with 36 neutrons. The results showed that the performance of the neural networks didn't necessarily increase with the number of hidden layers.

**Results.** The final results show that the topology B achieved better performance with a ER of 5% after 10,000 iterations of the learning process.

***Performance Evaluation of Machine Learning Method for Forest Fire Prediction***

**Introduction.** In the paper *Performance Evaluation of Machine Learning Methods for Forest Fire Modeling and Prediction*, Pham et al. (2020) explored and compared four models Bayes network, Naive Bayes, decision tree, and multivariate logistic regression for predicting forest fires.

**Data Collection and Training.** The features used in this research included historical fire data from Pu Mat national park, Nghe An province, Viet Nam and environmental features of slope degree, elevation, average annual temperature, aspect, drought index, land cover, river density, distance from roads and residential areas.

**Results.** These four models were validated with the receiver operating characteristic curve (ROC-AUC). The outcomes showed that these four had high accuracy scores in predicting fire vulnerability with AUC > 0.9. The Bayes model had the highest AUC = 0.96, the decision tree model came second with AUC = 0.91, the naive Bayes came third with AUC = 0.939 and the multivariate logistic regression came last with AUC = 0.937. Also, the research

showed that regions with lots of human activities were associated with moderate to high levels of fire vulnerability.

### *Mapping Forest Fires with K-nearest Neighbor and Random Forest Classifiers*

**Introduction.** In the research *Assessment of k-Nearest Neighbor and Random Forest classifiers for mapping forest fire areas in central Portugal using Landsat-8, Sentinel-2, and Terra Imagery*, Pacheco et al. (2021) discussed using K-nearest neighbor and random forest techniques for mapping burned fire areas. The research focused on mapping burned fire areas with remote sensing techniques. The approaches were to use supervised and unsupervised classifications down to pixel level utilizing satellite technologies such as Landsat-8, Sentinel-2, and Terra Imagery.

**Data Collection and Training.** The study area was in the districts of Santarem and Castelo Branco, central Portugal where there was a 93.4 $km^2$ fire on July 20, 2019. Data used for this research were multisensor satellite images and spectral bands from Landsat-8, Sentinel-2, and Terra Imagery.

For k-nearest neighbors classification, the model classified features based on the closet k training samples. In this research, values of k from 5 to 20 are selected, and the lowest Root Mean Square Error (RMSE) was used to select the best k. In this case, k = 5 was chosen due to the lowest of RMSE.

For random forest classification, the models classified features based on a combination of decision trees to archive more precise and stable predictions. In the research, the model was evaluated for 10 to 400 trees for sets of images from each sensor, and Out of Bag (OOB) error was used to validate the model. In this case, the best number of trees, which was 400 trees, was chosen using the lowest of OOB. The results also showed that the classification error didn’t change significantly within the same tree, but the error decreased with the increase of trees.

**Results.** The final results showed that both algorithms, in case of supervised learning, could classify approximately 90% of burned areas according to the ROC curve with AUC varying between 0.88 and 0.94 and Overall Accuracy (OA) > 89% and Dice Coefficient (DC) > 0.8.

### *Research Comparisons*

Table 4 represents the comparison of above three researches in terms of data types, tuning methods, and validation methods.

**Table 4**

*Comparison of Existing Researches*

| Research | Data | Models | Tuning | Validation |
|---|---|---|---|---|
| 1 | Historical fires and environmental data | ANN | Numbers of hidden layers and neurons | Total error rate |
| 2 | Historical fires and environmental data | Naive Bayes<br>Bayes network<br>Decision Tree<br>Multivariate Logistic Regression | | Receiver operating characteristic curve (ROC-AUC) |
| 3 | Satellite images and spectral bands | KNN | Number of k neighbors | Root mean square error<br>Overall accuracy<br>Dice coefficient |
| | | Random Forest | Number of trees | Out of bag error<br>Overall accuracy<br>Dice coefficient |

*Note.* In research 2, authors didn't mention tuning methods.

Research 1 is *Prediction of Forest Fires Using Artificial Neural Networks* by Safi and Bouroumi (2013).

Research 2 is *Performance Evaluation of Machine Learning Methods for Forest Fire Modeling and Prediction* by Pham et al. (2020).

Research 3 is *Assessment of k-Nearest Neighbor and Random Forest classifiers for mapping forest fire areas in central Portugal using Landsat-8, Sentinel-2, and Terra Imagery* by Pacheco et al. (2021).

# Chapter 2 - Data & Project Management

## 2.1 - Data Management Plan

This data management plan is based on DMP Tool's write data section, we will cover data collection, documentation and metadata, ethics and legal compliance, responsibilities and resources, storage and backup, selection and preservation, and data sharing (DMPTool, n.d).

### *Data Collection*

The dataset used for this project will be the historical fire data from the US Department of Agriculture (USDA). This data is publicly shared via USDA's research data archive with multiple data formats and metadata.

According to Short (2021), the data includes historical fires that took place in the United States from 1992 to 2018, and the data was originally collected via the national Fire Program Analysis (FPA) system. The data included key elements like discovery date, final fire size, and geo locations. The data also conforms to the standard of the National Wildfire Coordinating Group (NWCG) to indicate wildfire causations. Basic error-checking was also performed to reduce redundant data. The final outcome data includes 2.17 million geospatial historical fire records during a period of 27 years.

From the original USDA dataset, we will generate a specific dataset for California historical fires only. This data will cover all incidents that took place in the state of California and were associated with a FIPS code.

In addition, outcome prediction data with specific timestamps along with filters used will also be collected for further analysis.

Initially, original data can be downloaded in SQLITE format. Then, we export the data to CSV format and store original data in Google Cloud Platform (GCP) Storage. Next, this data will be cleaned and prepared with Tableau prep builder. Once cleaning is completed, cleaned data will be exported to CSV format and again stored in GCP Storage, and corresponding tables will be created or updated in GCP BigQuery for each data stored in GCP Storage.

***Documentation and Metadata***

The original metadata documentation provided by USDA will be included for this research. All original features' definitions, data formats, reference information, distribution information, and credits can be found in this document.

Additionally, metadata documentation for all formatted data stored in GCP and used for model training will also be available. This documentation includes metadata for all tables, all features' definitions, access controls, backup information, etc.

***Ethics and Legal Compliance***

The data is publicly published by USDA and can be used without permission or fee (Short, 2021). Therefore, USDA data source will be formally and properly cited in this research and on the final web application product.

***Responsibilities and Resources***

Since I am the only member in this research, only I can manage data access in Github repository and GCP IAM access.

In terms of data management resources, this project will utilize GCP Storage, GCP BigQuery and Github repository to manage, backup and host the data.

***Storage and Backup***

Original data, formatted data, and generated prediction data will be stored in GCP Storage in CSV format and corresponding tables for that data will be created in GCP BigQuery. Beside GCP, all CSV data will be backed up and stored in this research private Github repository.

Data access will be controlled in two levels: Github access and GCP IAM data access. Only I can only access these resources.

### *Selection and Preservation*

Since this research mainly focuses on the California region, only California historical fire data should be retained and preserved for future studies. This data will be obtained from the original dataset using only records with the state of California. Also, outcome prediction data will also be generated and updated after each prediction call to use for future studies and evaluations. After the project is finished, the original dataset can be removed from GCP storage and Bigquery.

### *Data Sharing*

We will not share any data to public users. However, the original USDA data source will be clearly stated to public users.

## 2.2 - Project Development Methodology

This research applies CRISP-DM methodology combined with AGILE approach for project management. We will divide this 4-month project into 5 sprints, each sprint will have an average of 2-week duration.

### *Sprint 1*

In this sprint, CRISP-DM's business understanding, data understanding and data preparation phases will be carried out.

### *Sprint 2*

In this sprint, we will focus on developing a machine learning model that will align with the research goals. Activities in this sprint include identifying model assumptions, designing tests, extracting data insights, and implementing the model.

### *Sprint 3*

After successfully implementing the model from sprint 2, we will move on to model evaluation. In this phase, various multi-class evaluation metrics will be applied to evaluate the

performance of the model. Then, we can adjust or fine tune the model to make performances better.

***Sprint 4***

In this sprint, product development will be carried out. The goal of this phase is to have the web-based interactive prediction map to be fully implemented and integrated with the prediction capability from the model evaluated in the previous sprint. The activities in this phase include identifying the user requirements, implementing UI/UX interfaces, and implementing back-end layers.

***Sprint 5***

In the last sprint, CRISP-DM's deployment phase will be implemented. After the web-based visualization successfully implemented in sprint 4, the next activity will be to deploy the web visualization to production using GCP Cloud Run.

**2.3 - Project Organization Plan**

Figure 3 depicts the detailed work breakdown structure for each sprint and phase.

***Project Understanding***

In this phase, we focus on understanding the problems and the dataset, survey available technologies and potential solutions.

***Data Collection***

Original data is collected in this phase, and initial data exploration is also carried out with multiple EDA tools to identify data statistics and data issues.

***Data Preparation***

In this stage, California data is isolated and extra derived data is also added to the dataset. Plus, all data issues are handled and data transformation is carried out to prepare for the modeling phase.

### *Modeling*

Initial modeling and tuning models are carried out here with various machine learning models.

### *Model Evaluation*

All models are evaluated using various multi-class classification evaluation metrics. The best models will be picked based on their performances for implementation.

### *Product Development*

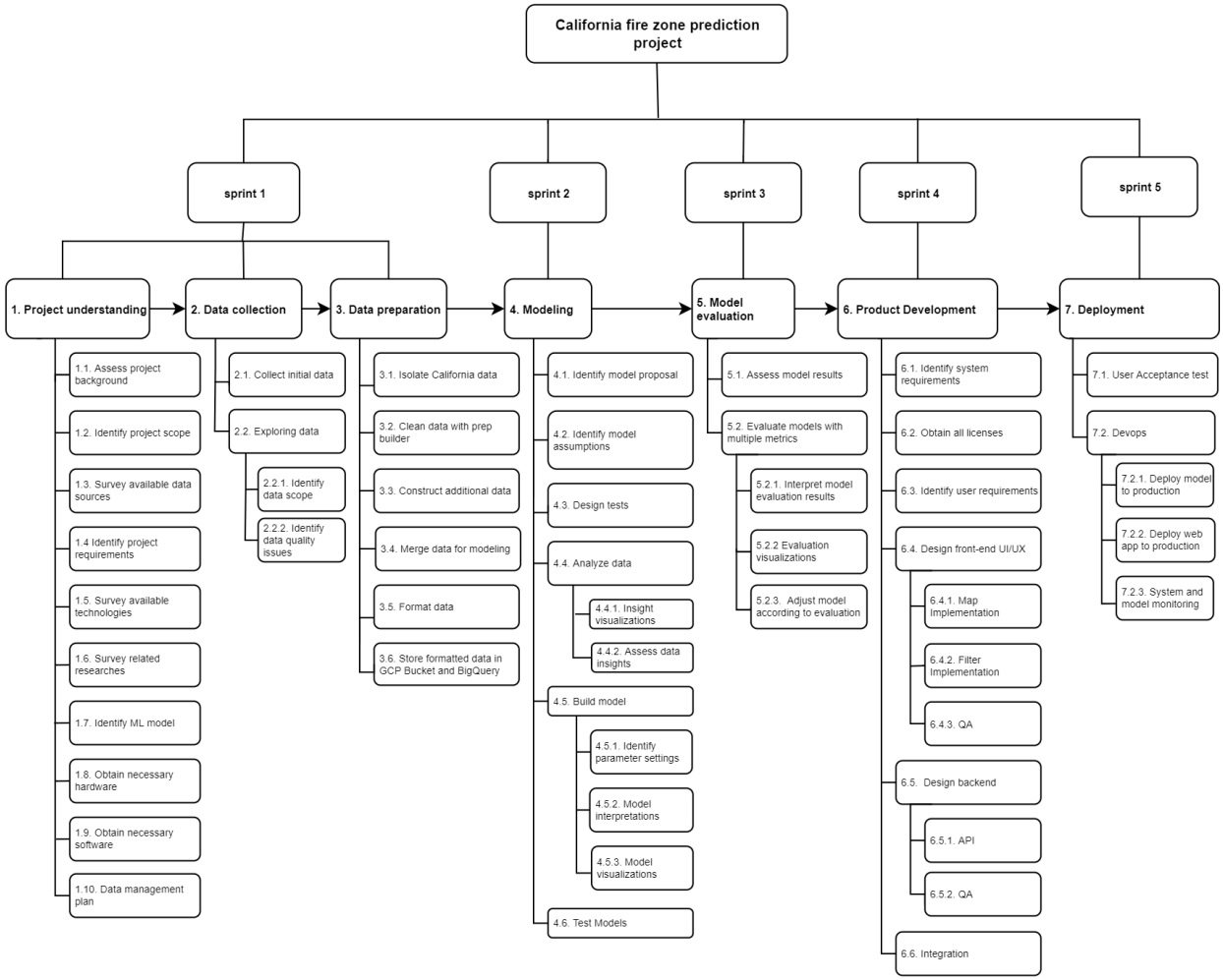
After the best model is picked from the previous stage, this stage focuses on implementing the web-based map visualization to display model predictions.

### *Deployment*

The web visualization is deployed in this stage.

**Figure 3**

*Work Breakdown Structure*




*Note.* Project understanding and data collection can take place in parallel, and the rest of stages need to follow a serial order.

## 2.4 - Project Resource Requirements and Plan

### *Tableau creator individual*

**Specifications.** Both Tableau desktop and Tableau prep builder are needed for data visualizations and data preparation.

**Justification.** Tableau creator individual includes access to both tableau desktop and tableau prep builder. Tableau desktop is great to extract initial data insights with visualizations.

Tableau prep builder is great for data cleaning and data preparation.

**Cost.** $70 per user per month (Tableau, n.d).

***Anaconda software package***

**Specification.** Jupyter Notebook, Python environment and PyCharm IDE.

**Justification.** Anaconda package includes all necessary tools for machine learning projects including building and evaluating machine learning models, and python environment for web development.

**Cost.** Free.

***DB Browser for SQLite***

**Specification.** Latest version.

**Justification.** This is a free tool to view sqlite databases and export data to CSV format.

**Cost.** Free.

***GCP Storage***

**Specification.** 2 GB of data storage, US muti-regions location.

**Justification.** GCP Storage provides great features for any kind of data workload such as high availability, high scalability, object versioning, pub/sub notifications, multiple redundancy options, etc. (Google Cloud, n.d, Cloud Storage).

**Cost.** $0.026 per GB per month (Google Cloud, n.d, Cloud Storage pricing).

***GCP Bigquery storage and queries***

**Specification.** Active storage for 2 GB of data and on-demand queries.

**Justification.** GCP Bigquery offers good features for a machine learning project such as serverless data warehouse, built-in machine learning and AI integration, real-time analytics, standard SQL, etc. (Google Cloud, n.d, BigQuery).

**Cost.** BigQuery storage costs $0.020 per GB per month and BigQuery on-demand queries cost $5.00 per TB (Google Cloud, n.d, BigQuery).

### *GCP Cloud Run*

**Specification.** Deploy estimations based on Cloud Run calculator for hosting this project include 2 CPU, 4 GiB Ram, CPU is only allocated during request processing, 200 requests per month, 2 minimum of instances, 90,000 ms execution time per request, and 20 concurrent requests per container instance.

**Justification.** Cloud Run provides a devop environment and tools to deploy any scalable containerized applications on a serverless platform. It also contains lots of great features such as supporting any language and any library, fast auto scaling, processing web traffic, portability, etc. (Google Cloud, n.d, Cloud Run).

**Cost.** CPU costs $0.00002400 per vCPU-second, memory costs $0.00000250 per GiB-Second, and requests costs $0.40 per million requests (Google Cloud, n.d, Cloud Run).

In addition, utilizing GCP BigQuery, Cloud Run and Storage make managing tasks easier since everything including access control can be easily accessed via the GCP console.

### *Total Cost Estimation*

Finally, Table 5 shows the estimated total cost needed for this research.

**Table 5**

*Total Cost Estimation*

| Softwares/Tools | Duration | Total Cost |
|---|---|---|
| Tableau creator individual | 4 months | $280 |
| Anaconda software package | 4 months | free |
| DB Browser for SQLite | 4 months | free |
| GCP Storage | 4 months | $0.208 |
| GCP Bigquery storage | 4 months | $0.16 |
| GCP BigQuery queries | 4 months | $5.00 |
| GCP Cloud Run | 1 month | Estimate for $120 |
| Total | | $406.00 |

*Note.* These costs are based on initial estimations, and are subject to change. Estimations already include expected total durations for the entire project.

**2.5 - Project Schedule**

***Task Breakdown Summary***

Figure 4 and Figure 5 show detailed task breakdown in each CRISP-DM phase based on the work breakdown structure in Figure 3.

**Figure 4**

*Task Breakdown Structure for Project Understanding, Data Collection, Data Preparation and Modeling Phases*

| All | WBS | Task Name | Duration | Planned Start Date | Planned Finish Date | Linked From | Complete |
|---|---|---|---|---|---|---|---|
| 1 | 1 | project understanding | 2.6 weeks | 8/30/2021 | 9/15/2021 | | ☑ |
| 2 | 1.1 | project background | 1 day | 8/30/2021 | 8/30/2021 | | ☑ |
| 3 | 1.2 | project scope | 1 day | 8/31/2021 | 8/31/2021 | 2 | ☑ |
| 4 | 1.3 | survey available datasource | 2 days | 9/1/2021 | 9/2/2021 | 2 | ☑ |
| 5 | 1.4 | project requirement | 1 day | 9/3/2021 | 9/3/2021 | 3, 4 | ☑ |
| 6 | 1.5 | survey available technologies | 1 day | 9/6/2021 | 9/6/2021 | 5 | ☑ |
| 7 | 1.6 | survey related researches | 3 days | 9/6/2021 | 9/8/2021 | 5 | ☑ |
| 8 | 1.7 | identify ML model | 1 day | 9/9/2021 | 9/9/2021 | 6, 7 | ☑ |
| 9 | 1.8 | obtain hardware | 1 day | 9/13/2021 | 9/13/2021 | 5 | ☑ |
| 10 | 1.9 | obtain software | 1 day | 9/13/2021 | 9/13/2021 | 5 | ☑ |
| 11 | 1.10 | data management plan | 2 days | 9/14/2021 | 9/15/2021 | 4 | ☑ |
| 12 | 2 | **data collection** | **4 days** | **9/16/2021** | **9/21/2021** | | ☑ |
| 13 | 2.1 | collect intial data | 2 days | 9/16/2021 | 9/17/2021 | 9, 10, 4, 11 | ☑ |
| 14 | 2.2 | **explore data** | **2 days** | **9/20/2021** | **9/21/2021** | | ☑ |
| 15 | 2.2.1 | data scope | 2 days | 9/20/2021 | 9/21/2021 | 13 | ☑ |
| 16 | 2.2.2 | data quality report | 2 days | 9/20/2021 | 9/21/2021 | 13 | ☑ |
| 17 | 3 | **data preparation** | **5 days** | **9/22/2021** | **9/28/2021** | | ☑ |
| 18 | 3.1 | isolate California data | 1 day | 9/22/2021 | 9/22/2021 | 12 | ☑ |
| 19 | 3.2 | clean data with prep builder | 1 day | 9/23/2021 | 9/23/2021 | 18 | ☑ |
| 20 | 3.3 | construct addtional data | 1 day | 9/23/2021 | 9/23/2021 | 18 | ☑ |
| 21 | 3.4 | merge data for modeling | 1 day | 9/24/2021 | 9/24/2021 | 20, 19 | ☑ |
| 22 | 3.5 | format data | 1 day | 9/27/2021 | 9/27/2021 | 21 | ☑ |
| 23 | 3.6 | store format data in Big query | 1 day | 9/28/2021 | 9/28/2021 | 22 | ☑ |
| 24 | 4 | **modeling** | **18 days** | **10/11/2021** | **11/3/2021** | | ☐ |
| 25 | 4.1 | identify model propasal | 1 day | 10/11/2021 | 10/11/2021 | 8, 17 | ☑ |
| 26 | 4.2 | identify model assumption | 1 day | 10/12/2021 | 10/12/2021 | 25 | ☐ |
| 27 | 4.3 | design test | 2 days | 10/12/2021 | 10/13/2021 | 25 | ☐ |
| 28 | 4.4 | **analyze data** | **3 days** | **10/18/2021** | **10/20/2021** | | ☐ |
| 29 | 4.4.1 | insight visualizations | 2 days | 10/18/2021 | 10/19/2021 | 23 | ☐ |
| 30 | 4.4.2 | assess data insight | 1 day | 10/20/2021 | 10/20/2021 | 29 | ☐ |
| 31 | 4.5 | **build model** | **8 days** | **10/21/2021** | **11/1/2021** | | ☐ |
| 32 | 4.5.1 | identify parameter settings | 3 days | 10/21/2021 | 10/25/2021 | 26, 30 | ☐ |
| 33 | 4.5.2 | model interpretation | 3 days | 10/26/2021 | 10/28/2021 | 32 | ☐ |
| 34 | 4.5.3 | model visualization | 2 days | 10/29/2021 | 11/1/2021 | 33 | ☐ |
| 35 | 4.6 | test models | 2 days | 11/2/2021 | 11/3/2021 | 27, 31 | ☐ |

*Note.* "Linked From" column refers to the "All" column to show dependencies in a task.

**Figure 5**

*Task Breakdown Structure for Model Evaluation, Product Development and Deployment Phases*

| All | WBS | Task Name | Duration | Planned Start Date | Planned Finish Date | Linked From | Complete |
|---|---|---|---|---|---|---|---|
| **36** | **5** | **model evaluation** | **9 days** | **11/4/2021** | **11/16/2021** | | |
| **37** | 5.1 | assess model result | 3 days | 11/4/2021 | 11/8/2021 | 35 | |
| **38** | **5.2** | **evaluate model w/ multiple metr…** | **6 days** | **11/9/2021** | **11/16/2021** | | |
| **39** | 5.2.1 | interpret eval results | 2 days | 11/9/2021 | 11/10/2021 | 37 | |
| **40** | 5.2.2 | evaluation visualizations | 2 days | 11/11/2021 | 11/12/2021 | 39 | |
| **41** | 5.2.3 | adjust model accordingly | 2 days | 11/15/2021 | 11/16/2021 | 40 | |
| **42** | **6** | **product development** | **15 days** | **11/17/2021** | **12/7/2021** | | |
| **43** | 6.1 | identify system requirements | 1 day | 11/17/2021 | 11/17/2021 | 36 | |
| **44** | 6.2 | obtain all licenses | 1 day | 11/18/2021 | 11/18/2021 | 43 | |
| **45** | 6.3 | identify user requirements | 2 days | 11/19/2021 | 11/22/2021 | 44 | |
| **46** | **6.4** | **design front-end UI/UX** | **8 days** | **11/23/2021** | **12/2/2021** | | |
| **47** | 6.4.1 | map implementation | 5 days | 11/23/2021 | 11/29/2021 | 45 | |
| **48** | 6.4.2 | filter implementation | 5 days | 11/23/2021 | 11/29/2021 | 45 | |
| **49** | 6.4.3 | QA | 3 days | 11/30/2021 | 12/2/2021 | 48, 47 | |
| **50** | **6.5** | **design backend** | **8 days** | **11/23/2021** | **12/2/2021** | | |
| **51** | 6.5.1 | API | 5 days | 11/23/2021 | 11/29/2021 | 45 | |
| **52** | 6.5.2 | QA | 3 days | 11/30/2021 | 12/2/2021 | 51 | |
| **53** | 6.6 | Integration | 3 days | 12/3/2021 | 12/7/2021 | 46, 50 | |
| **54** | **7** | **deployment** | **10 days** | **12/8/2021** | **12/21/2021** | | |
| **55** | 7.1 | user acceptance test | 2 days | 12/8/2021 | 12/9/2021 | 42 | |
| **56** | **7.2** | **Dev ops** | **8 days** | **12/10/2021** | **12/21/2021** | | |
| **57** | 7.2.1 | deploy model to production | 3 days | 12/10/2021 | 12/14/2021 | 36 | |
| **58** | 7.2.2 | deploy web app to production | 3 days | 12/15/2021 | 12/17/2021 | 57 | |
| **59** | 7.2.3 | system and model monitor | 2 days | 12/20/2021 | 12/21/2021 | 58 | |

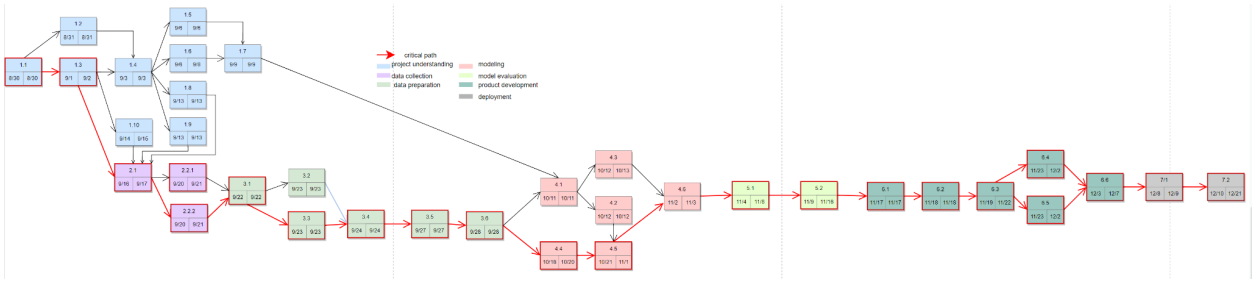
*Note.* This figure is a continuation from Figure 4.

***PERT chart***

This PERT chart in Figure 6 mainly shows dependencies and main timeline between tasks, and for sub tasks that need to do sequentially only their parent task is shown. For example, task 7.2.1, 7.2.2, and 7.2.3 are sequential; hence, only parent task 7.2 is shown.

**Figure 6**

*Pert Chart from Work Breakdown Structure*



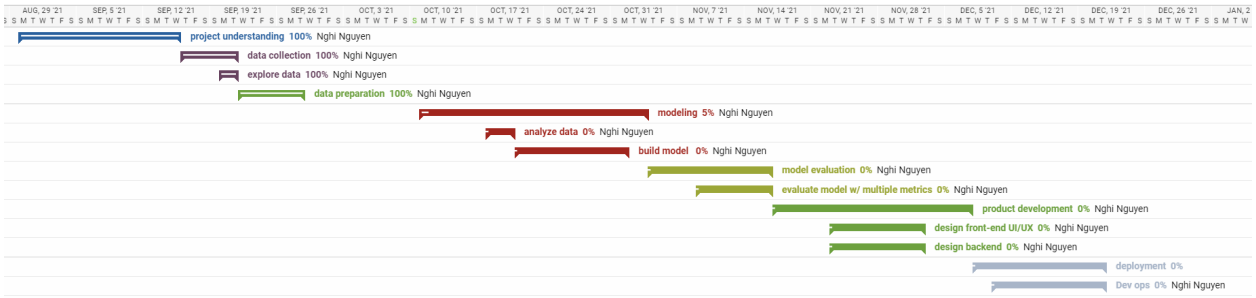

*Note. Refer to Figure 4 and Figure 5 for detailed tasks.*

***Gantt Chart Summary***

The summary Gantt chart in Figure 7 shows only main tasks in bold in Figure 4 and Figure 5 without dependencies.

**Figure 7**

*Gantt Chart Summary View*



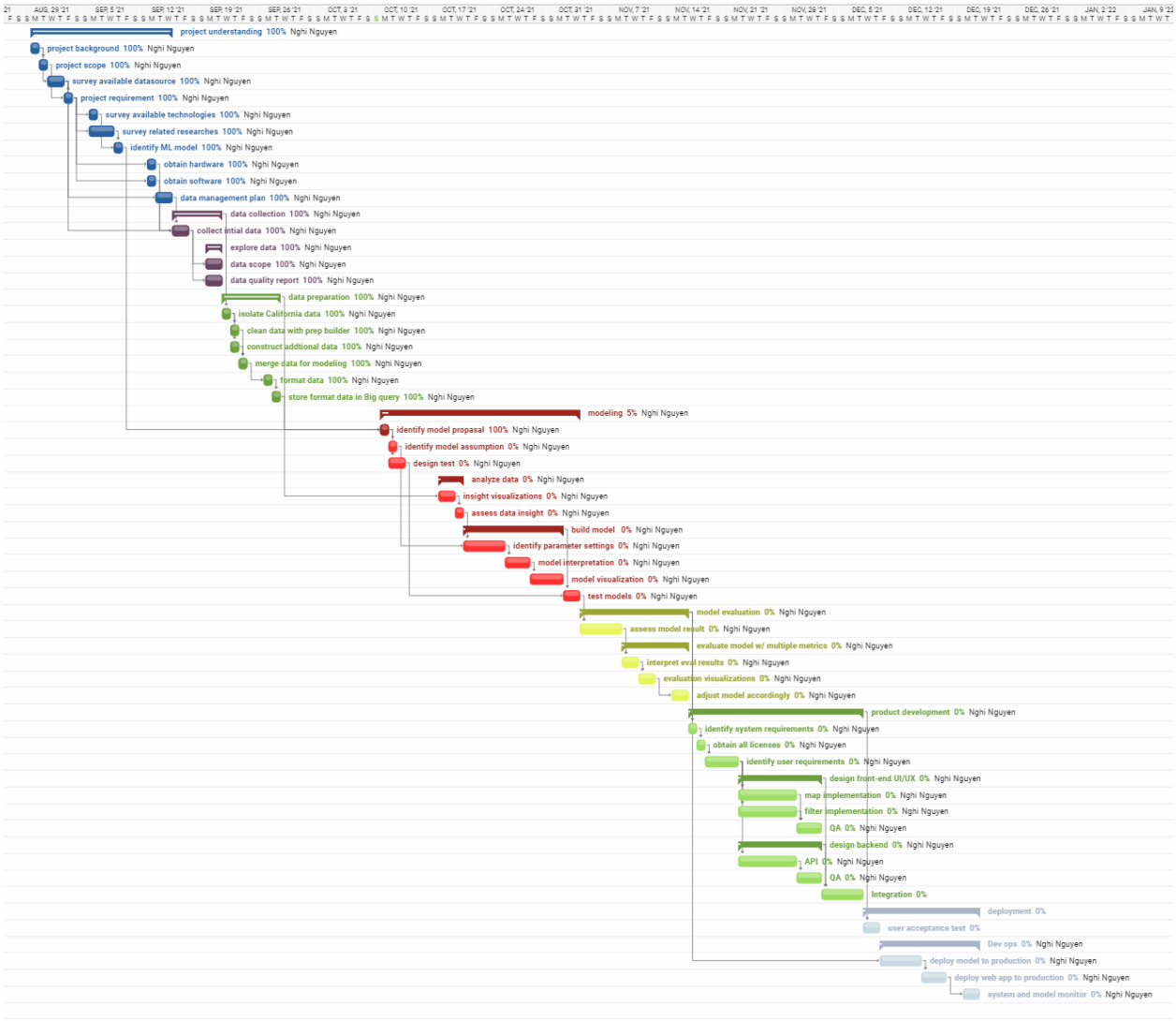

*Note.* This Gantt chart shows a summary view without dependencies of main parent tasks in Figure 4 and Figure 5.

### *Detailed Gantt Chart with Dependencies*

The detailed Gantt chart in Figure 8 shows all tasks in Figure 4 and Figure 5 with detailed dependencies.

**Figure 8**

*Gantt Chart Detailed View with Dependencies*




*Note.* This Gantt chart shows entire tasks including dependencies as mentioned in Figure 4 and Figure 5.

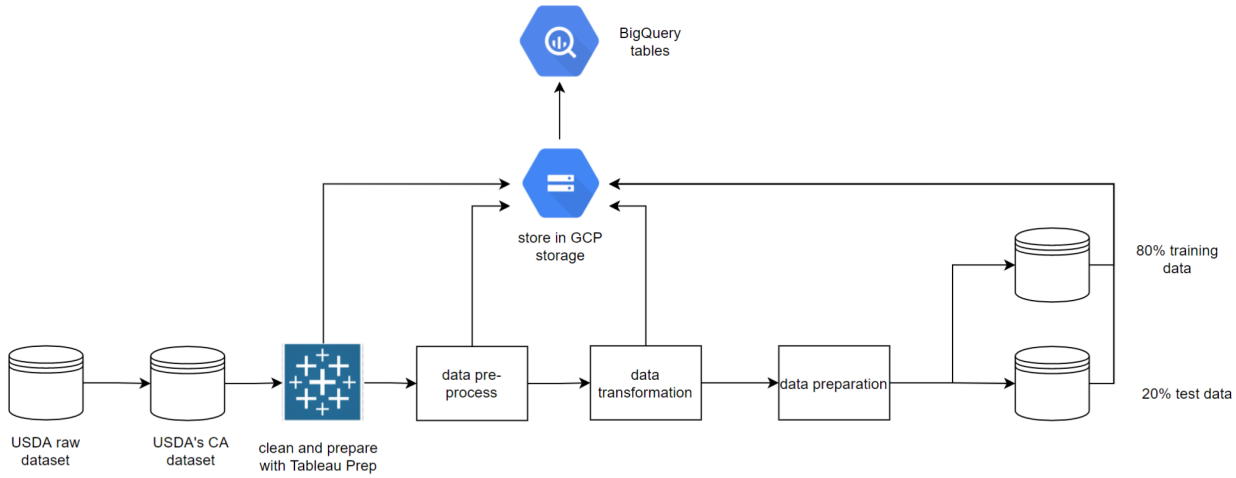
# Chapter 3 - Data Engineering

## 3.1 - Data Process

Firstly, the raw dataset is collected in SQLite format from the U.S. Department of Agriculture (USDA)'s Research Data Archive. Then, since we only focus on California (CA) for this project, only CA data with non-null FIPS codes is isolated from the raw dataset. Next, both raw data and derived CA sub-dataset are stored on GCP storage for storing and backing up purposes, and the corresponding CA fire data table is created on GCP BigQuery. After that, we can use BigQuery data for next steps namely data pre-processing, data transformation, and data preparation. Figure 9 depicts the overall process in this project.

**Figure 9**

*Overall Data Process*




*Note.* Overview of the data process for the data engineering and model development for this project. Data outputs of each stage are stored in GCP storage and BigQuery.

In the data pre-processing phase, we first clean and add additional derived data using Tableau Prep Builder; then, the data can be further processed and handled for data issues and feature selection. The data output in this stage is again stored in GCP storage and corresponding tables created in GCP BigQuery.

Next, in the data transformation stage, pre-processed data from the previous stage is transformed into a suitable format for modeling. For continuous features, techniques such as normalization can be carried out; and for categorical features, techniques such as ordinal encoding can be applied. Then, transformed data is again stored in GCP storage and GCP BigQuery.

Finally, during the data preparation phase, transformed data for each region will be splitted into two disjoint sets of training and testing. 80% will be used for the training set, 20% will be used for the testing set. Plus, for the purpose of model tuning validation, we further apply stratified K-fold on training sets to create validation sets.

### 3.2 - Data Collection

#### *Data Sources and Quantity*

According to Short (2021), the raw dataset is historical fire incidents in the United States from 1992 to 2018. The data collected from the archive system of federal, state and local fire organizations. The data also conforms to the National Wildfire Coordinating Group (NWCG) to include wildfire causasions, and redundant records were already performed. The original dataset has 2.17 million records representing 165 million acres burned from 1992 to 2018 in the United States.

However, for the purpose of this project, we isolate only CA data with non-null FIPS codes. Plus, some columns are also dropped due to having no purpose of solving this problem. After this process, we have an exclusive CA historical fire dataset with incidents from 1997 to 2018 and a total of 142,551 records.

#### *Key Features*

In the original dataset, there are a total of 37 features; Figure 10, 11, 12 and 13 show all the features along with data type and data definitions. However, for the purposes of the project, following key parameters are considered for potential feature selection: DISCOVERY_DATE,

DISCOVERY_TIME, NWCG_GENERAL_CAUSE, NWCG_CAUSE_CLASSIFICATION, NWCG_CAUSE_AGE_CATEGORY, FIRE_SIZE_CLASS, OWNER_DESCR, FIPS_CODE.

**Figure 10**

*Raw Dataset Schema with Definitions - Part I*

| Field name | Type | Mode | Policy Tags | Description |
|---|---|---|---|---|
| FOD_ID | INTEGER | NULLABLE | | Unique numeric record identifier. |
| FPA_ID | STRING | NULLABLE | | Unique identifier that contains information necessary to track back to the original record in the source dataset. |
| SOURCE_SYSTEM_TYPE | STRING | NULLABLE | | Type of source database or system that the record was drawn from (federal, nonfederal, or interagency). |
| SOURCE_SYSTEM | STRING | NULLABLE | | Name of or other identifier for source database or system that the record was drawn from. |
| NWCG_REPORTING_AGENCY | STRING | NULLABLE | | Active National Wildlife Coordinating Group (NWCG) Unit Identifier for the agency preparing the fire report. |
| NWCG_REPORTING_UNIT_ID | STRING | NULLABLE | | Active NWCG Unit Identifier for the unit preparing the fire report. |
| NWCG_REPORTING_UNIT_NAME | STRING | NULLABLE | | Active NWCG Unit Name for the unit preparing the fire report. |
| SOURCE_REPORTING_UNIT | INTEGER | NULLABLE | | Code for the agency unit preparing the fire report, based on code/name in the source dataset. |
| SOURCE_REPORTING_UNIT_NAME | STRING | NULLABLE | | Name of reporting agency unit preparing the fire report, based on code/name in the source dataset. |
| LOCAL_FIRE_REPORT_ID | INTEGER | NULLABLE | | Number or code that uniquely identifies an incident report for a particular reporting unit and a particular calendar year. |
| LOCAL_INCIDENT_ID | STRING | NULLABLE | | Number or code that uniquely identifies an incident for a particular local fire management organization within a particular calendar year. |

*Note.* Data definitions provided by USDA's metadata (Short, 2021).

**Figure 11**

*Raw Dataset Schema with Definitions - Part II*

| | | | |
|---|---|---|---|
| FIRE_CODE | STRING | NULLABLE | Code used within the interagency wildland fire community to track and compile cost information for emergency fire suppression (https://www.firecode.gov/). |
| FIRE_NAME | STRING | NULLABLE | Name of the incident, from the fire report (primary) or ICS-209 report (secondary). |
| ICS_209_PLUS_INCIDENT_JOIN_ID | STRING | NULLABLE | Primary identifier needed to join into operational situation reporting data for the incident in the ICS-209-PLUS dataset. |
| ICS_209_PLUS_COMPLEX_JOIN_ID | STRING | NULLABLE | If part of a complex, secondary identifier potentially needed to join to operational situation reporting data for the incident in the ICS-209-PLUS dataset (2014 and later only). |
| MTBS_ID | STRING | NULLABLE | Incident identifier, from the MTBS perimeter dataset. |
| MTBS_FIRE_NAME | STRING | NULLABLE | Name of the incident, from the MTBS perimeter dataset. |
| COMPLEX_NAME | STRING | NULLABLE | Name of the complex under which the fire was ultimately managed, when discernible. |
| FIRE_YEAR | INTEGER | NULLABLE | Calendar year in which the fire was discovered or confirmed to exist. |
| DISCOVERY_DATE | DATE | NULLABLE | Date on which the fire was discovered or confirmed to exist. |
| DISCOVERY_DOY | INTEGER | NULLABLE | Day of year on which the fire was discovered or confirmed to exist. |

*Note.* Data definitions provided by USDA's metadata (Short, 2021).

**Figure 12**

*Raw Dataset Schema with Definitions - Part III*

| | | | |
|---|---|---|---|
| DISCOVERY_TIME | INTEGER | NULLABLE | Time of day that the fire was discovered or confirmed to exist. |
| NWCG_CAUSE_CLASSIFICATION | STRING | NULLABLE | Broad classification of the reason the fire occurred (Human, Natural, Missing data/not specified/undetermined). |
| NWCG_GENERAL_CAUSE | STRING | NULLABLE | Event or circumstance that started a fire or set the stage for its occurrence. |
| NWCG_CAUSE_AGE_CATEGORY | STRING | NULLABLE | If cause attributed to children (ages 0-12) or adolescents (13-17), the value for this data element is set to Minor; otherwise null. |
| CONT_DATE | DATE | NULLABLE | Date on which the fire was declared contained or otherwise controlled (mm/dd/yyyy where mm=month, dd=day, and yyyy=year). |
| CONT_DOY | INTEGER | NULLABLE | Day of year on which the fire was declared contained or otherwise controlled. |
| CONT_TIME | INTEGER | NULLABLE | Time of day that the fire was declared contained or otherwise controlled (hhmm where hh=hour, mm=minutes). |
| FIRE_SIZE | FLOAT | NULLABLE | The estimate of acres within the final perimeter of the fire. |
| FIRE_SIZE_CLASS | STRING | NULLABLE | Code for fire size based on the number of acres within the final fire perimeter (A<=to 0.25 acres, B=0.26-9.9 acres, C=10.0-99.9 acres, D=100-299 acres, E=300 to 999 acres, F=1000 to 4999 acres, and G=5000+ acres). |
| LATITUDE | FLOAT | NULLABLE | Latitude (NAD83) for point location of the fire (decimal degrees). |
| LONGITUDE | FLOAT | NULLABLE | Longitude (NAD83) for point location of the fire (decimal degrees). |
| OWNER_DESCR | STRING | NULLABLE | Name of primary owner or entity responsible for managing the land at the point of origin of the fire at the time of the incident. |

*Note.* Data definitions provided by USDA's metadata (Short, 2021).

**Figure 13**

*Raw Dataset Schema with Definitions - Part IV*

| | | | |
|---|---|---|---|
| CONT_TIME | INTEGER | NULLABLE | Time of day that the fire was declared contained or otherwise controlled (hhmm where hh=hour, mm=minutes). |
| FIRE_SIZE | FLOAT | NULLABLE | The estimate of acres within the final perimeter of the fire. |
| FIRE_SIZE_CLASS | STRING | NULLABLE | Code for fire size based on the number of acres within the final fire perimeter (A<=to 0.25 acres, B=0.26-9.9 acres, C=10.0-99.9 acres, D=100-299 acres, E=300 to 999 acres, F=1000 to 4999 acres, and G=5000+ acres). |
| LATITUDE | FLOAT | NULLABLE | Latitude (NAD83) for point location of the fire (decimal degrees). |
| LONGITUDE | FLOAT | NULLABLE | Longitude (NAD83) for point location of the fire (decimal degrees). |
| OWNER_DESCR | STRING | NULLABLE | Name of primary owner or entity responsible for managing the land at the point of origin of the fire at the time of the incident. |
| STATE | STRING | NULLABLE | Two-letter alphabetic code for the state in which the fire burned (or originated), based on the nominal designation in the fire report. |
| COUNTY | INTEGER | NULLABLE | County, or equivalent, in which the fire burned (or originated), based on nominal designation in the fire report |
| FIPS_CODE | INTEGER | NULLABLE | Five-digit code from the Federal Information Process Standards (FIPS) publication 6-4 for representation of counties and equivalent entities, based on the nominal designation in the fire report. |
| FIPS_NAME | STRING | NULLABLE | County name from the FIPS publication 6-4 for representation of counties and equivalent entities, based on the nominal designation in the fire report. |

*Note.* Data definitions provided by USDA's metadata (Short, 2021).

Table 6 and Table 7 summarizes the key points of these key features.

**Table 6**

*Key Features Summary - Part I*

| Variable title | DISCOVERY_DATE | DISCOVERY_TIME | NWCG_GENERAL_CAUSE | NWCG_CAUSE_CLASSIFICATION |
|---|---|---|---|---|
| Input (X) or output (Y) | X | X | X | X |
| Unit of measurement | date | Hour & minute | N/A | N/A |
| Data type | date | integer | string | string |
| Derived Data | Season | Part of Date Discovery hours | N/A | N/A |
| Continuous or Categorical | categorical | continuous | categorical | categorical |
| Levels (#) | N/A | N/A | 13 | 3 |
| Constraints | State = CA Non-null FIPS | State = CA Non-null FIPS | State = CA Non-null FIPS | State = CA Non-null FIPS |
| From | 1997 | 1997 | 1997 | 1997 |
| End | 2018 | 2018 | 2018 | 2018 |

*Note.* Derived data describes which data can be derived from a feature. Levels (#) describes the cardinality for a feature.

**Table 7**

*Key Features Summary - Part II*

| **Variable title** | **NWCG_CAUSE_AGE_CATEGORY** | **FIRE_SIZE_CLASS** | **OWNER_DESCR** | **FIPS_CODE** |
|---|---|---|---|---|
| Input (X) or output (Y) | X | X | X | Y |
| Unit of measurement | N/A | acre | N/A | N/A |
| Data type | string | string | string | integer |
| Derived Data | N/A | N/A | N/A | N/A |
| Continuous or Categorical | categorical | categorical | categorical | categorical |
| Levels (#) | 1 | 7 | 14 | 58 |
| Constraints | State = CA<br>Non-null FIPS | State = CA<br>Non-null FIPS | State = CA<br>Non-null FIPS | State = CA<br>Non-null FIPS |
| From | 1997 | 1997 | 1997 | 1997 |
| End | 2018 | 2018 | 2018 | 2018 |

*Note.* Derived data describes which data can be derived from a feature. Levels (#) describes the cardinality for a feature.

### *Raw Dataset Sample*

Figure 14, 15 and 16 shows samples of raw California dataset with the schema described in Figure 10, 11, 12 and 13.

**Figure 14**

*Sample Raw Dataset Part I*

| Row | FOD_ID | FPA_ID | SOURCE_SYSTEM_TYPE | SOURCE_SYSTEM | NWCG_REPORTING_AGENCY | NWCG_REPORTING_UNIT_ID | NWCG_REPORTING_UNIT_NAME | SOURCE_REPORTING_UNIT | SOURCE_REPORTING_UNIT_NAME | LOCAL_FIRE_REPORT_ID | LOCAL_INCIDENT_ID | FIRE_CODE | FIRE_NAME |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1371900 | SFO-2000CACDFBDU005619 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 005619 | null | LINE |
| 2 | 1371945 | SFO-2000CACDFBDU008393 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 008393 | null | WYE |
| 3 | 1371842 | SFO-2000CACDFBDU001969 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 001969 | null | COLLINS |
| 4 | 1371956 | SFO-2000CACDFBDU009129 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 009129 | null | HOT 2 |
| 5 | 1372048 | SFO-2001CACDFBDU003333 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 003333 | null | WEST |
| 6 | 1372021 | SFO-2002CACDFBDU003180 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 003180 | null | BARTELL |
| 7 | 1372144 | SFO-2002CACDFBDU006776 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 006776 | null | CANAL |
| 8 | 1372200 | SFO-2003CACDFBDU006261 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 006261 | null | AIRPORT |
| 9 | 1372242 | SFO-2003CACDFBDU009032 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 009032 | null | BLACK ROCK |
| 10 | 1372339 | SFO-2004CACDFBDU002653 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 002653 | null | WILLIAMS |
| 11 | 1372564 | SFO-2005CACDFBDU012489 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 012489 | null | WYE |
| 12 | 1372852 | SFO-2008CACDFBDU000095 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 000095 | null | BIG PINE BIG_PINE |
| 13 | 1372871 | SFO-2008CACDFBDU004843 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 004843 | null | OLANCHA_AREA |
| 14 | 1372908 | SFO-2007CACDFBDU009442 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 009442 | null | Q3 3 |
| 15 | 1372858 | SFO-2008CACDFBDU002924 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 002924 | null | BISHOP BISHOP 2 |
| 16 | 300309265 | SFO-2009-CACDFBDU008242 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | 1515 | 008242 | null | SUNSET RD /S BARLOW LN |
| 17 | 300334299 | SFO-2009-CACDFBDU013298 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | 29794 | 013298 | null | LINE |
| 18 | 1373037 | SFO-2009-CACDFBDU001287 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 001287 | null | FORT |
| 19 | 1372771 | SFO-2007CACDFBDU003820 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 003820 | null | CANAL |
| 20 | 1372044 | SFO-2001CACDFBDU002941 | NONFED | ST-CACDF | ST/C&L | USCABDU | San Bernardino Unit | null | CDF - San Bernardino Unit | null | 002941 | null | RESERVIOR |

*Note.* First part of the sample raw data set includes features: fod_id, fpa_id, source_system_type, source_system, nwcg_reporting_agency, nwcg_reporting_unit_id, nwcg_reporting_unit_name, source_reporting_unit, source_reporting_unit_name local_fire_report_id, local_incident_id, fire_code, fire_name.

**Figure 15**

*Sample Raw Dataset Part II*

| ICS_209_PLUS_INCIDENT_JOIN_ID | ICS_209_PLUS_COMPLEX_JOIN_ID | MTBS_ID | MTBS_FIRE_NAME | COMPLEX_NAME | FIRE_YEAR | DISCOVERY_DATE | DISCOVERY_DOY | DISCOVERY_TIME | NWCG_CAUSE_CLASSIFICATION | NWCG_GENERAL_CAUSE | NWCG_CAUSE_AGE_CATEGORY |
|---|---|---|---|---|---|---|---|---|---|---|---|
| null | null | null | null | null | 2000 | 2000-07-18 | 200 | 2034 | Human | Recreation and ceremony | null |
| null | null | null | null | null | 2000 | 2000-10-09 | 283 | 1040 | Human | Recreation and ceremony | null |
| null | null | null | null | null | 2000 | 2000-03-15 | 75 | 1425 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2000 | 2000-11-11 | 316 | 2009 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2001 | 2001-04-27 | 117 | 1436 | Human | Missing data/not specified/undetermined | null |
| 2002_CA-BDU-003180_BARTELL | null | null | null | null | 2002 | 2002-04-15 | 105 | 1900 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2002 | 2002-07-22 | 203 | 2038 | Human | Missing data/not specified/undetermined | null |
| 2003_CA-BDU-006261_AIRPORT | null | null | null | null | 2003 | 2003-06-29 | 180 | 1300 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2003 | 2003-08-29 | 241 | 1920 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2004 | 2004-03-18 | 78 | 1220 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2005 | 2005-11-19 | 323 | 1343 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2008 | 2008-01-02 | 2 | 147 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2008 | 2008-05-05 | 126 | 1430 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2007 | 2007-08-28 | 240 | 633 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2008 | 2008-03-14 | 74 | 2246 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2009 | 2009-07-30 | 211 | 1910 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2009 | 2009-11-26 | 330 | 1256 | Human | Missing data/not specified/undetermined | null |
| 2009_CA-BDU-001287_FORT | null | CA-BDU-001287-20090205 | FORT | null | 2009 | 2009-02-05 | 36 | 1141 | Human | Debris and open burning | null |
| 2007_CA-BDU-003820_CANAL | null | null | null | null | 2007 | 2007-04-09 | 99 | 1535 | Human | Missing data/not specified/undetermined | null |
| null | null | null | null | null | 2001 | 2001-04-13 | 103 | 1400 | Human | Missing data/not specified/undetermined | null |

*Note.* Second part of the sample raw data set includes features: ics_209_plus_incident_join_id, ics_209_plus_complex_join_id, mtbs_id, mtbs_fire_name, complex_name, fire_year, discovery_date, discovery_doy, discovery_time, nwcg_cause_classification, nwcg_general_cause, nwcg_cause_age_category.

**Figure 16**

*Sample Raw Dataset Part III*

| CONT_DATE | CONT_DOY | CONT_TIME | FIRE_SIZE | FIRE_SIZE_CLASS | LATITUDE | LONGITUDE | OWNER_DESCR | STATE | COUNTY | FIPS_CODE | FIPS_NAME |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2000-07-19 | 201 | 10 | 6.0 | B | 37.36888888 | -118.3769444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2000-10-09 | 283 | 1045 | 0.3 | B | 37.38305555 | -118.3780556 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2000-03-15 | 75 | 1510 | 1.0 | B | 37.29694444 | -118.3038889 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2000-11-11 | 316 | 2045 | 1.0 | B | 37.36888888 | -118.3411111 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2001-04-27 | 117 | 1530 | 0.3 | B | 37.35388888 | -118.3961111 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2002-04-15 | 105 | 2200 | 4.0 | B | 37.16694444 | -118.2619444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2002-07-22 | 203 | 2113 | 1.0 | B | 37.16694444 | -118.2619444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2003-06-29 | 180 | 1800 | 2.0 | B | 37.36888888 | -118.3769444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2003-08-29 | 241 | 2002 | 0.3 | B | 36.90611111 | -118.2269444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2004-03-18 | 78 | 1300 | 0.5 | B | 37.38305555 | -118.3961111 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2005-11-19 | 323 | 1800 | 0.8 | B | 37.36888888 | -118.3769444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| null | null | null | 0.5 | B | 37.165 | -118.2888889 | UNDEFINED FEDERAL | CA | null | 6027 | Inyo County |
| null | null | null | 1.0 | B | 36.16694444 | -117.795 | UNDEFINED FEDERAL | CA | null | 6027 | Inyo County |
| null | null | null | 0.3 | B | 36.68611111 | -118.1088889 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| null | null | null | 0.3 | B | 37.33305555 | -118.3561111 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| null | null | null | 1.0 | B | 37.352042 | -118.423209 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |
| 2009-11-26 | 330 | 1319 | 0.75 | B | 37.361504 | -118.368422 | STATE OR PRIVATE | CA | null | 6027 | Inyo County |
| null | null | null | 945.0 | E | 36.89111111 | -118.1911111 | UNDEFINED FEDERAL | CA | null | 6027 | Inyo County |
| null | null | null | 25.0 | C | 37.36194444 | -118.3680556 | MISSING/NOT SPECIFIED | CA | null | 6027 | Inyo County |
| 2001-04-13 | 103 | 1800 | 10.0 | C | 37.41194444 | -118.4319444 | MUNICIPAL/LOCAL | CA | null | 6027 | Inyo County |

*Note.* Third part of the sample raw data set includes features: cont_date, cont_doy, cont_time, fire_size, fire_size_class, latitude, longitude, owner_descr, state, county, fips_code, fips_name.

### 3.3 - Data Pre-processing

#### *Cleaning and Prepare Derived Data with Tableau Prep*

In this phase, we utilize Tableau Prep Builder to clean and prepare extra derived data.

Following data are added to the dataset based on discovery_date feature:

- SEASON: identify spring, summer, fall, and winter based on month of DISCOVERY_DATE.

Following data are added to the dataset based on DISCOVERY_TIME feature:

- DISCOVERY_HOUR: hours range from 0 to 23 on which a fire was reported.
- DISCOVERY_PART_OF_DATE: identify morning, afternoon, evening and night based on DISCOVERY_HOUR.

Following cleaning steps are carried out:

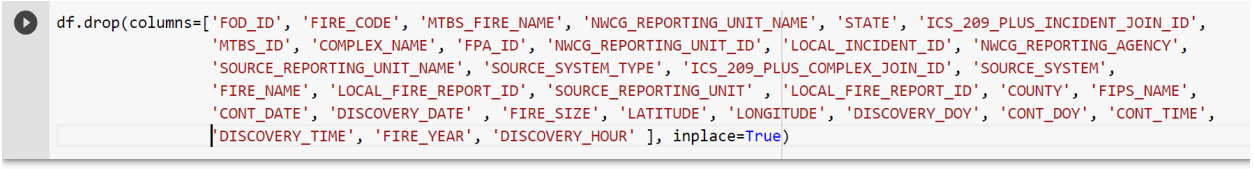
- Replace “Missing data/not specified/undetermined” by null values for NWCG_CAUSE_AGE_CATEGORY, NWCG_CAUSE_CLASSIFICATION, NWCG_GENERAL_CAUSE.

#### *Dropping Irrelevant Features*

Now recall the problem we defined in chapter 1, the problem we try to solve is detecting the probability of FIPS codes where likely to result in wildfires using users’ queries about season, part of day, type of properties, causations, burned sizes. Therefore, we can drop irrelevant features as depicted in Figure 17 that have no impact in solving the problem such as key ID features, fire information, reporting units or organizations, etc.

**Figure 17**

*Dropping Irrelevant Columns*



```
df.drop(columns=['FOD_ID', 'FIRE_CODE', 'MTBS_FIRE_NAME', 'NWCG_REPORTING_UNIT_NAME', 'STATE', 'ICS_209_PLUS_INCIDENT_JOIN_ID',
                 'MTBS_ID', 'COMPLEX_NAME', 'FPA_ID', 'NWCG_REPORTING_UNIT_ID', 'LOCAL_INCIDENT_ID', 'NWCG_REPORTING_AGENCY',
                 'SOURCE_REPORTING_UNIT_NAME', 'SOURCE_SYSTEM_TYPE', 'ICS_209_PLUS_COMPLEX_JOIN_ID', 'SOURCE_SYSTEM',
                 'FIRE_NAME', 'LOCAL_FIRE_REPORT_ID', 'SOURCE_REPORTING_UNIT' , 'LOCAL_FIRE_REPORT_ID', 'COUNTY', 'FIPS_NAME',
                 'CONT_DATE', 'DISCOVERY_DATE' , 'FIRE_SIZE', 'LATITUDE', 'LONGITUDE', 'DISCOVERY_DOY', 'CONT_DOY', 'CONT_TIME',
                 'DISCOVERY_TIME', 'FIRE_YEAR', 'DISCOVERY_HOUR' ], inplace=True)
```

*Note*. Python code shows a list of columns being dropped due to no impact in solving the problem. The data used for this stage is the output of the cleaning process in Tableau Prep Builder.

***Identify and Handle Data issues***

At this stage, we are left with all categorical data namely DISCOVERY_PART_OF_DATE, SEASON, NWCG_CAUSE_CLASSIFICATION, NWCG_GENERAL_CAUSE, NWCG_CAUSE_AGE_CATEGORY, FIRE_SIZE_CLASS, OWNER_DESCR, and FIPS_CODE. To capture categorical data critical data quality, we use categorical data quality reports to capture information of total, count, missing percentage, cardinality, mode, mode frequency, mode percentage, second mode, second mode frequency, and second mode percentage.

Figure 18 shows the data quality report before cleaning. From the result, we can see that there are missing data issues in features of NWCG_CAUSE_CLASSIFICATION, OWNER_DESCR, NWCG_GENERAL_CAUSE and NWCG_CAUSE_AGE_CATEGORY. Following steps are carried out:

- Replacing NWCG_CAUSE_CLASSIFICATION, NWCG_GENERAL_CAUSE , OWNER_DESCRwith a non-null mode.
- Dropping the NWCG_CAUSE_AGE_CATEGORY feature because the missing data percentage is too high.
- Dropping duplicate rows.

**Figure 18**

*Data Quality Report Before Cleaning*

| | Categorical Feature | total | count | miss% | card | mode | mode freq | mode pct | 2nd mode | 2nd mode freq | 2nd mode pct |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | SEASON | 142551 | 142551 | 0.00000 | 4 | Summer | 74094 | 51.97719 | Fall | 32401 | 22.72941 |
| 1 | DISCOVERY_PART_OF_DATE | 142551 | 142551 | 0.00000 | 4 | Afternoon | 62317 | 43.71558 | Everving | 30287 | 21.24643 |
| 2 | NWCG_CAUSE_CLASSIFICATION | 142551 | 113511 | 20.37166 | 3 | Human | 100806 | 70.71574 | None | 29040 | 20.37166 |
| 3 | NWCG_GENERAL_CAUSE | 142551 | 78801 | 44.72084 | 13 | None | 63750 | 44.72084 | Equipment and vehicle use | 24692 | 17.32152 |
| 4 | NWCG_CAUSE_AGE_CATEGORY | 142551 | 5407 | 96.20697 | 2 | None | 137144 | 96.20697 | Minor | 5407 | 3.79303 |
| 5 | FIRE_SIZE_CLASS | 142551 | 142551 | 0.00000 | 7 | A | 78334 | 54.95156 | B | 55564 | 38.97833 |
| 6 | OWNER_DESCR | 142551 | 45302 | 68.22050 | 15 | None | 97249 | 68.22050 | USFS | 14475 | 10.15426 |
| 7 | FIPS_CODE | 142551 | 142551 | 0.00000 | 58 | 6065 | 14989 | 10.51483 | 6037 | 8755 | 6.14166 |

*Note.* miss% represents missing data, mode and 2nd mode represent most frequent and second most frequent values respectively.

After cleaning, as can be seen in Figure 19, we no longer have missing data and the pre-processed data is now shrinking down to 22,737 rows. The reason for this huge shrink is because we are down from 37 continuous and categorical features to seven categorical features during the removing duplicates step.

**Figure 19**

*Data Quality Report After Cleaning*

| | Categorical Feature | total | count | miss% | card | mode | mode freq | mode pct | 2nd mode | 2nd mode freq | 2nd mode pct |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | SEASON | 22737 | 22737 | 0.00000 | 4 | Summer | 9236 | 40.62101 | Fall | 6283 | 27.63337 |
| 1 | DISCOVERY_PART_OF_DATE | 22737 | 22737 | 0.00000 | 4 | Afternoon | 8190 | 36.02058 | Morning | 5330 | 23.44197 |
| 2 | NWCG_CAUSE_CLASSIFICATION | 22737 | 22737 | 0.00000 | 2 | Human | 20492 | 90.12623 | Natural | 2245 | 9.87377 |
| 3 | NWCG_GENERAL_CAUSE | 22737 | 22737 | 0.00000 | 12 | Equipment and vehicle use | 7836 | 34.46365 | Debris and open burning | 2986 | 13.13278 |
| 4 | FIRE_SIZE_CLASS | 22737 | 22737 | 0.00000 | 7 | A | 10291 | 45.26103 | B | 8435 | 37.09812 |
| 5 | OWNER_DESCR | 22737 | 22737 | 0.00000 | 14 | USFS | 10669 | 46.92352 | PRIVATE | 3503 | 15.40661 |
| 6 | FIPS_CODE | 22737 | 22737 | 0.00000 | 58 | 6065 | 1163 | 5.11501 | 6073 | 937 | 4.12104 |

*Note.* After cleaning, the dataset has no missing data problem. Number of rows reduced to 22,797.

***Sample Pre-processed Dataset After Cleaning***

After cleaning, we no longer have missing data, all missing data are replaced with data mode, irrelevant columns are removed, duplicated rows are removed. At this stage, we only have 7 columns left as shown in Figure 20.

**Figure 20**

*Sample Dataset After Cleaning*

| SEASON | DISCOVERY_PART_OF_DATE | NWCG_CAUSE_CLASSIFICATION | NWCG_GENERAL_CAUSE | FIRE_SIZE_CLASS | OWNER_DESCR | FIPS_CODE |
|---|---|---|---|---|---|---|
| Fall | Afternoon | Natural | Natural | A | STATE OR PRIVATE | 6001 |
| Fall | Everving | Natural | Natural | B | USFS | 6001 |
| Fall | Morning | Natural | Natural | C | USFS | 6001 |
| Fall | Everving | Natural | Natural | C | STATE | 6001 |
| Fall | Afternoon | Natural | Natural | A | USFS | 6003 |
| Fall | Morning | Natural | Natural | A | USFS | 6003 |
| Fall | Night | Natural | Natural | A | USFS | 6003 |
| Fall | Everving | Natural | Natural | A | USFS | 6003 |
| Fall | Afternoon | Natural | Natural | B | USFS | 6003 |
| Fall | Morning | Natural | Natural | B | USFS | 6003 |
| Fall | Afternoon | Natural | Natural | A | BLM | 6003 |
| Fall | Afternoon | Natural | Natural | A | STATE OR PRIVATE | 6003 |
| Fall | Night | Natural | Natural | A | PRIVATE | 6003 |
| Fall | Morning | Natural | Natural | A | PRIVATE | 6003 |
| Fall | Everving | Natural | Natural | A | USFS | 6005 |

*Note.* Only seven categorical features left after cleaning.

### 3.4 - Data Transformation

In this stage, ordinal encoding is applied to all categorical features for data transformation which means mapping each feature level with an integer. Figure 21 shows ordinal encoding using scikit-learn's OrdinalEncoder library. Then, the transformed data and their mapping lookups are stored in GCP BigQuery for the next phase, data preparation.

**Figure 21**

*Sklearn's Ordinal Encoder*

```
[ ]  from sklearn.preprocessing import OrdinalEncoder
     enc = OrdinalEncoder()
     enc.fit(df[df.columns])
     df[df.columns] = enc.transform(df[df.columns])
```

*Note.* Scikit-learn's OrdinalEncoder library for categorical data encoding.

Figure 22 shows the sample dataset after data transformation, and Figure 23, 24, 25, 26, 27, 28 and 29 show the mapping for each feature's distinct values.

**Figure 22**

*Sample Dataset after Transformation*

| | SEASON | DISCOVERY_PART_OF_DATE | NWCG_CAUSE_CLASSIFICATION | NWCG_GENERAL_CAUSE | FIRE_SIZE_CLASS | OWNER_DESCR | FIPS_CODE |
|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 1 | 6 | 0 | 10 | 0 |
| **1** | 0 | 1 | 1 | 6 | 1 | 13 | 0 |
| **2** | 0 | 2 | 1 | 6 | 2 | 13 | 0 |
| **3** | 0 | 1 | 1 | 6 | 2 | 9 | 0 |
| **4** | 0 | 0 | 1 | 6 | 0 | 13 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **22732** | 3 | 3 | 0 | 8 | 0 | 13 | 48 |
| **22733** | 3 | 2 | 0 | 8 | 0 | 10 | 52 |
| **22734** | 3 | 0 | 0 | 8 | 0 | 13 | 53 |
| **22735** | 3 | 0 | 0 | 8 | 1 | 13 | 54 |
| **22736** | 3 | 0 | 0 | 8 | 0 | 13 | 54 |

*Note*. Data transformed with ordinal encoder.

**Figure 23**

*DISCOVERY_PART_OF_DATE Mapping*

| val | mapping |
|---|---|
| Afternoon | 0 |
| Everving | 1 |
| Morning | 2 |
| Night | 3 |

*Note. The first column is DISCOVERY_PART_OF_DATE and the second column is the integer mapping.*

**Figure 24**

*FIPS_CODE Mapping*

| val | mapping |
|---|---|
| 6001 | 0 |
| 6003 | 1 |
| 6005 | 2 |
| 6007 | 3 |
| 6009 | 4 |
| 6011 | 5 |
| 6013 | 6 |
| 6015 | 7 |
| 6017 | 8 |
| 6019 | 9 |
| 6021 | 10 |
| 6023 | 11 |
| 6025 | 12 |
| 6027 | 13 |
| 6029 | 14 |
| 6031 | 15 |
| 6033 | 16 |
| 6035 | 17 |
| 6037 | 18 |
| 6039 | 19 |
| 6041 | 20 |
| 6043 | 21 |
| 6045 | 22 |
| 6047 | 23 |
| 6049 | 24 |
| 6051 | 25 |
| 6053 | 26 |
| 6055 | 27 |
| 6057 | 28 |
| 6059 | 29 |
| 6061 | 30 |
| 6063 | 31 |
| 6065 | 32 |
| 6067 | 33 |
| 6069 | 34 |
| 6071 | 35 |
| 6073 | 36 |
| 6075 | 37 |
| 6077 | 38 |
| 6079 | 39 |
| 6081 | 40 |
| 6083 | 41 |
| 6085 | 42 |
| 6087 | 43 |
| 6089 | 44 |
| 6091 | 45 |
| 6093 | 46 |
| 6095 | 47 |
| 6097 | 48 |
| 6099 | 49 |
| 6101 | 50 |
| 6103 | 51 |
| 6105 | 52 |
| 6107 | 53 |
| 6109 | 54 |
| 6111 | 55 |
| 6113 | 56 |
| 6115 | 57 |

*Note.* The first column is the FIPS_CODE and the second column is the integer mapping.

**Figure 25**

*FIRE_SIZE Mapping*

| val | mapping |
|---|---|
| A | 0 |
| B | 1 |
| C | 2 |
| D | 3 |
| E | 4 |
| F | 5 |
| G | 6 |

*Note.* The first column is the *FIRE_SIZE* and the second column is the integer mapping.

**Figure 26**

*NWCG_CAUSE_CLASSIFICATION Mapping*

| val | mapping |
|---|---|
| Human | 0 |
| Natural | 1 |

*Note.* The first column is the *NWCG_CAUSE_CLASSIFICATION* and the second column is the integer mapping.

**Figure 27**

*NWCG_GENERAL_CAUSE Mapping*

| val | mapping |
|---|---|
| Arson/incendiarism | 0 |
| Debris and open burning | 1 |
| Equipment and vehicle use | 2 |
| Firearms and explosives use | 3 |
| Fireworks | 4 |
| Misuse of fire by a minor | 5 |
| Natural | 6 |
| Other causes | 7 |
| Power generation/transmission/distribution | 8 |
| Railroad operations and maintenance | 9 |
| Recreation and ceremony | 10 |
| Smoking | 11 |

*Note.* The first column is the *NWCG_GENERAL_CAUSE* and the second column is the integer mapping.

**Figure 28**

*OWNER_DESCR Mapping*

| val | mapping |
|---|---|
| BIA | 0 |
| BLM | 1 |
| BOR | 2 |
| COUNTY | 3 |
| FWS | 4 |
| MUNICIPAL/LOCAL | 5 |
| NPS | 6 |
| OTHER FEDERAL | 7 |
| PRIVATE | 8 |
| STATE | 9 |
| STATE OR PRIVATE | 10 |
| TRIBAL | 11 |
| UNDEFINED FEDERAL | 12 |
| USFS | 13 |

*Note.* The first column is the OWNER_DESCR and the second column is the integer mapping.

**Figure 29**

SEASON *Mapping*

| val | mapping |
|---|---|
| Fall | 0 |
| Spring | 1 |
| Summer | 2 |
| Winter | 3 |

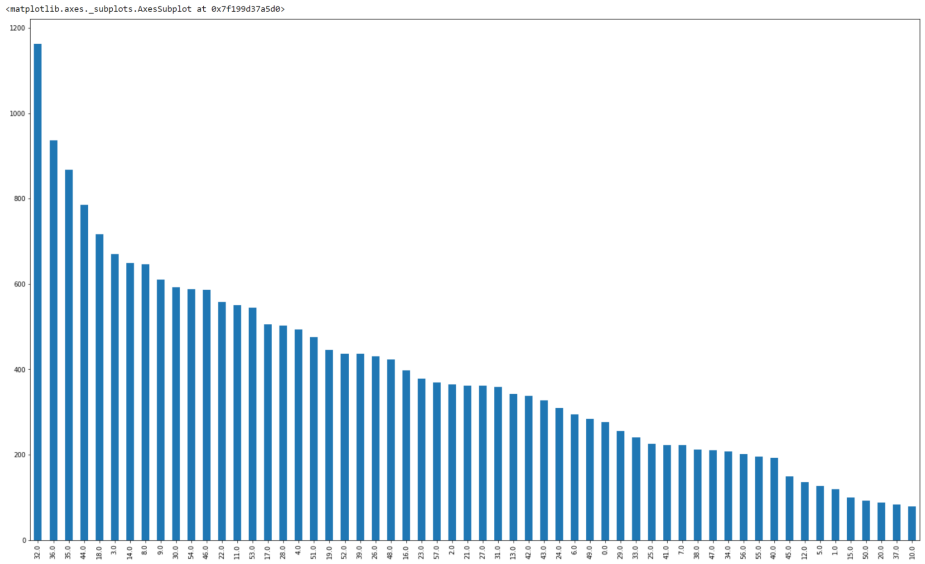
*Note.* The first column is the SEASON and the second column is the integer mapping.

### 3.5 - Data Preparation

After the data transformation phase, we end up with a very imbalanced dataset as depicted in Figure 30. So, we further use the undersampling technique to resolve this problem. Doing so will help with the problem of machine learning models being biased toward the majority of data. Consequently, we have a final balanced dataset as shown in Figure 31.

**Figure 30**

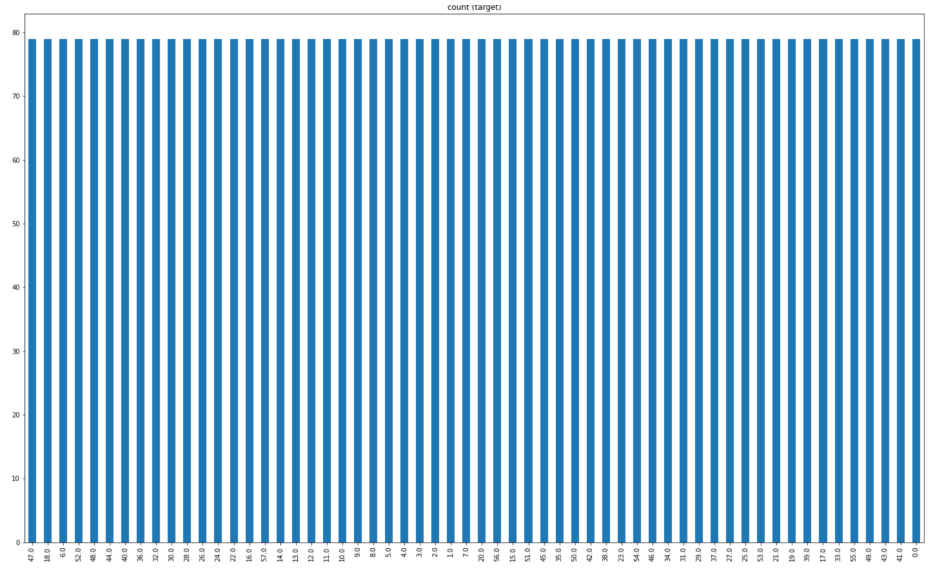
*Imbalance Dataset after Transforming*




*Note.* This is the count plot for FIPS_CODE, there are some FIPS_CODE that have very little incidents, whereas there are some FIP_CODE that have lots of incidents.

**Figure 31**

*Dataset after Undersampling*




*Note.* After undersampling, each FIPS_CODE has the same amount of data.

Then, during the modeling phase, we will further split the data into two disjoint sets of 80% training data and 20% testing data by random sampling for each region. Note that, also during the modeling phase, we further use stratified K-fold technique on the training set to generate validation sets for model validation. Figure 32 shows the Python randomly sampling code.

**Figure 32**

*Randomly Sampling Code*

```
traning_df = df.sample(frac = 0.8)
test_df = df.drop(traning_df.index)
```

*Note.* Transformed dataset is splitted into 2 disjoint sets, 80% for training and the rest 20% is for the test set.

Figure 33 and Figure 34 show samples of training and test set.

**Figure 33**

*Training Dataset*

| | SEASON | DISCOVERY_PART_OF_DATE | NWCG_CAUSE_CLASSIFICATION | NWCG_GENERAL_CAUSE | FIRE_SIZE_CLASS | OWNER_DESCR | FIPS_CODE |
|---|---|---|---|---|---|---|---|
| **19167** | 2 | 1 | 0 | 1 | 0 | 13 | 39 |
| **13393** | 3 | 2 | 0 | 10 | 1 | 13 | 57 |
| **17988** | 0 | 1 | 0 | 2 | 0 | 12 | 23 |
| **8799** | 0 | 2 | 0 | 2 | 0 | 1 | 52 |
| **13349** | 3 | 2 | 0 | 1 | 1 | 10 | 32 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **8978** | 1 | 2 | 0 | 11 | 0 | 10 | 18 |
| **9893** | 2 | 2 | 0 | 10 | 0 | 13 | 3 |
| **16910** | 2 | 3 | 0 | 0 | 1 | 8 | 28 |
| **9634** | 2 | 2 | 0 | 11 | 0 | 13 | 44 |
| **18544** | 1 | 1 | 0 | 2 | 0 | 5 | 13 |

*Note.* Sample of training set.

**Figure 34**

*Test Dataset*

| | SEASON | DISCOVERY_PART_OF_DATE | NWCG_CAUSE_CLASSIFICATION | NWCG_GENERAL_CAUSE | FIRE_SIZE_CLASS | OWNER_DESCR | FIPS_CODE |
|---|---|---|---|---|---|---|---|
| **3** | 0 | 0 | 1 | 6 | 0 | 10 | 1 |
| **11** | 0 | 0 | 1 | 6 | 0 | 10 | 8 |
| **15** | 0 | 0 | 1 | 6 | 0 | 13 | 10 |
| **16** | 0 | 0 | 1 | 6 | 0 | 13 | 11 |
| **30** | 0 | 0 | 1 | 6 | 0 | 8 | 18 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **22687** | 3 | 1 | 0 | 2 | 1 | 1 | 35 |
| **22689** | 3 | 1 | 0 | 2 | 1 | 12 | 36 |
| **22707** | 3 | 1 | 0 | 2 | 1 | 10 | 48 |
| **22713** | 3 | 1 | 0 | 2 | 1 | 10 | 55 |
| **22732** | 3 | 1 | 0 | 8 | 1 | 8 | 14 |

*Note.* Sample of test set.

### 3.6 - Data Statistics

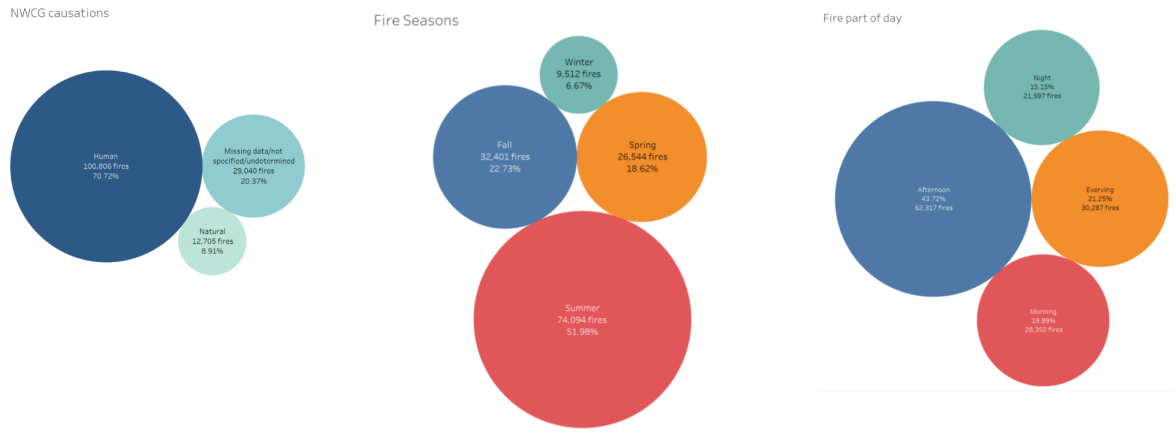
#### *Deriving Raw Dataset Summary and Overall Statistics*

Raw historical fire incidents dataset with data in the United States from 1992 to 2018 is obtained from USDA's Research Data Archive; then, we isolate only data for California with valid FIPS codes for the purposes of this research. Then, raw dataset is stored in GCP storage and GCP BigQuery for accessing and backing up purposes.

From the CA dataset, we can see that most fires reported are small to medium sizes and caused by humans. Figure  35, 36, 37, 38 and 39 show overall statistics for the CA dataset.

**Figure 35**

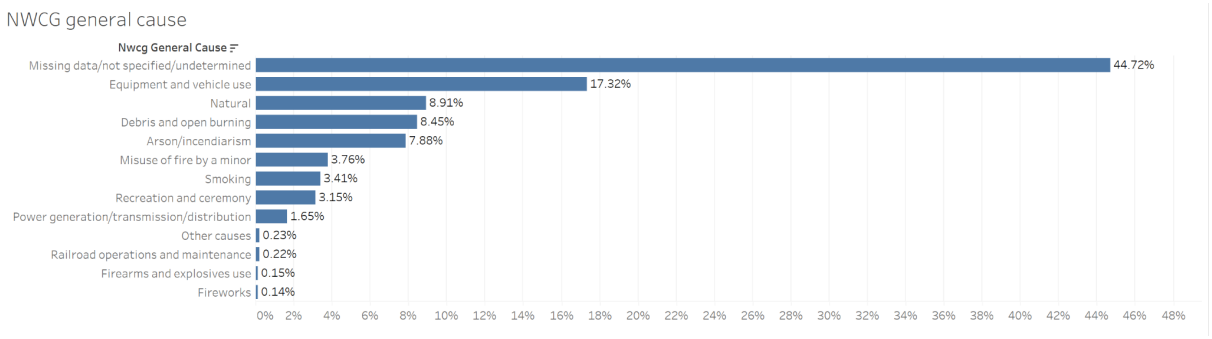
*CA Raw Data Statistics: NWCG Fire Causations, Fire Seasons, and Fire Part of Day*




*Note.* Most fires are caused by humans, summer is the peak season for wildfires, and most fires were reported in the afternoon.

**Figure 36**

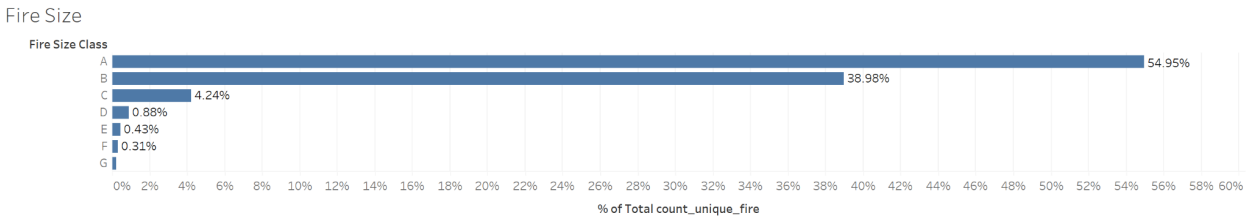
*NWCG General Causes Statistics*




*Note.* There are lots of missing or undetermined data for this feature. Besides that, fires caused by equipment and vehicle use make up the dominant.

**Figure 37**

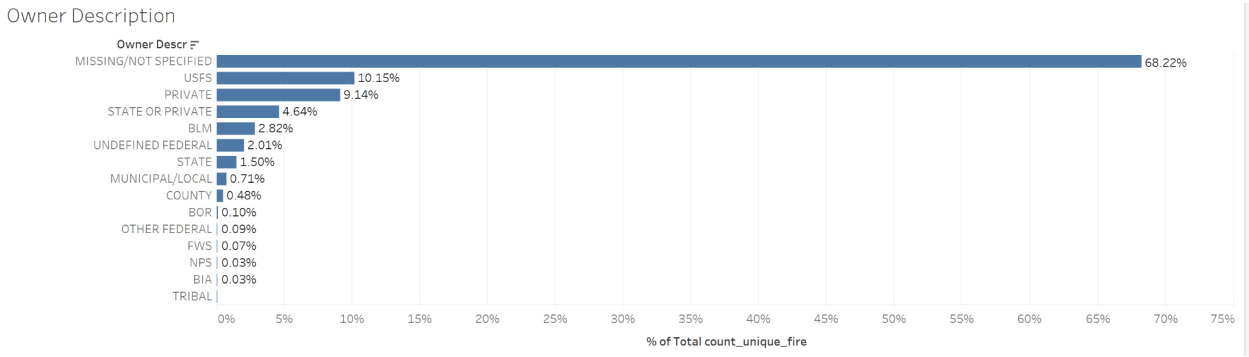
*Fire Size Statistics*




*Note.* Most fires are in class A and B which are from 0 to 99.9 acres (Short, 2021).

**Figure 38**

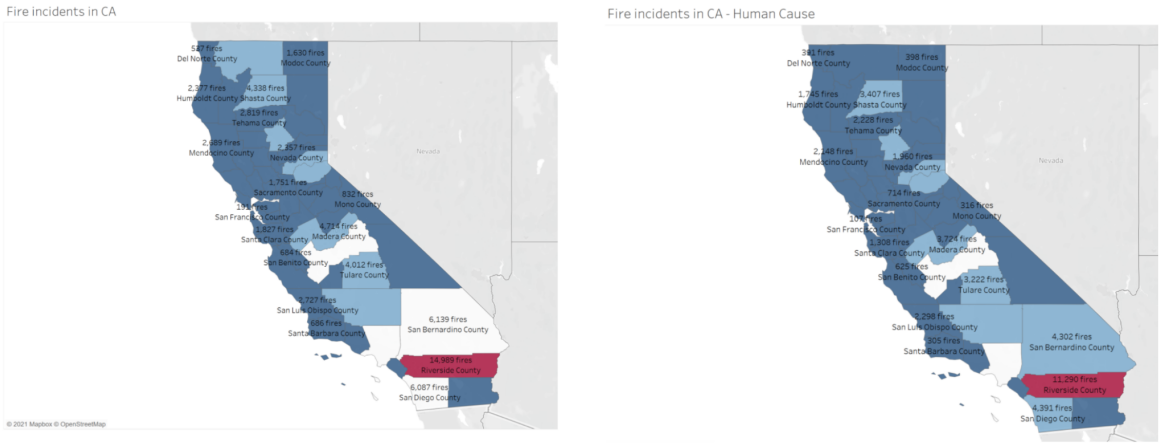
*Owner Description Statistics*




*Note.* Owner description describes the primary owner or organization who manages the property (Short, 2021). Most are missing or undetermined data in this feature. Besides that, United States Forest Service (USFS) and Private are most dominant.

**Figure 39**

*Fire Occurrences in CA by Counties*




*Note.* Most fires occurred in the forest areas, central CA, and southern CA. Riverside county has the most fires reported. Most fires are reportedly caused by humans.

***Pre-processing Summary***

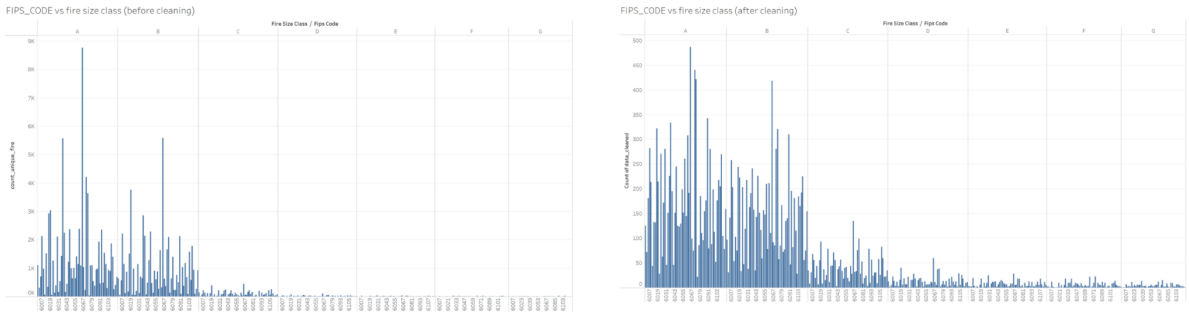
The problem with raw dataset is that there are lots of missing or undetermined data and irrelevant features. To solve this problem we replace missing categorical data with a non-null mode; and for a column with extremely high missing data like NWCG_CAUSE_AGE_CATEGORY, we drop the entire column. For irrelevant and redundant columns that have no contributions to solving the problem, we drop all of them. Also, duplicated rows resulting from dropping columns are also eliminated.

After preprocessing and cleaning, we reduce from 37 continuous and categorical features to 7 categorical features, and eliminate missing data issues. Figure 40, 41, 42, 43, 44 and 45 depict data distributions of the target feature fips_codes with other descriptive features before and after cleaning.

Then, cleaned data is also stored in GCP storage and GCP BigQuery.

**Figure 40**

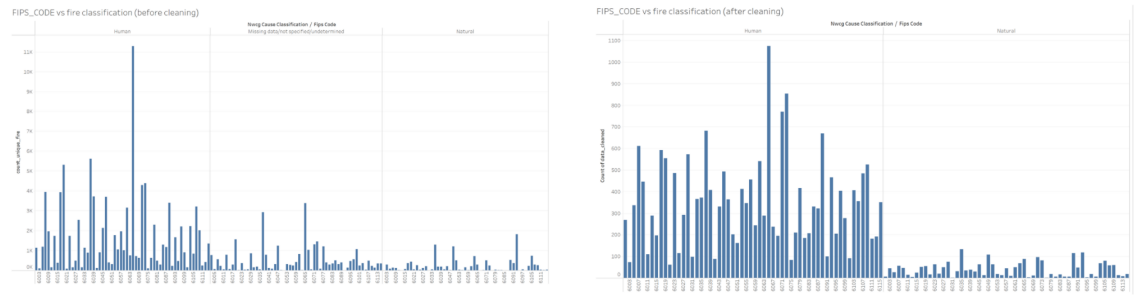
*Fips Codes and Fire Size Class before and after Cleaning*




*Note.* Most fires are still dominant in class A and B.

**Figure 41**

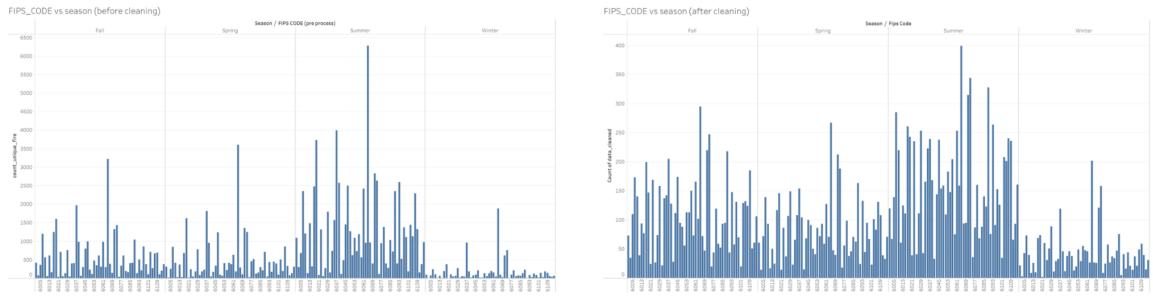
*Fips Codes and Fire Classification before and after Cleaning*




*Note.* Before cleaning, fire classification has missing data, after cleaning missing data is replaced by the mode which is human cause.

**Figure 42**

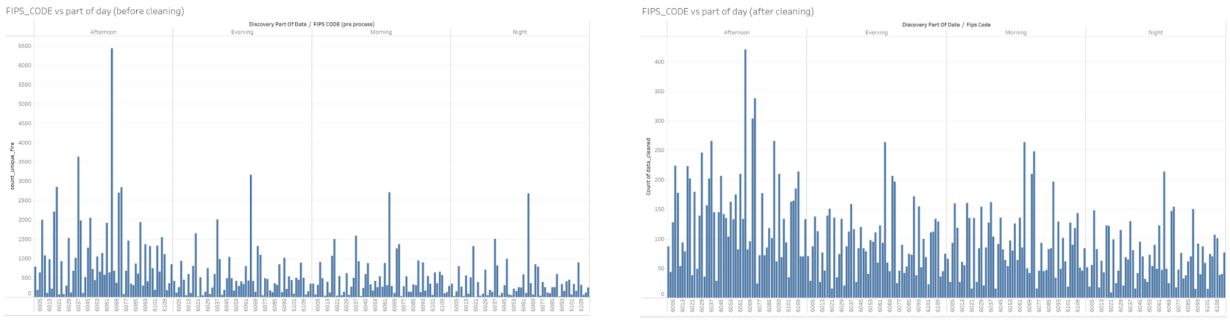
*Fips Codes and Fire Season before and after Cleaning*




*Note.* The data looks more spread out after cleaning.

**Figure 43**

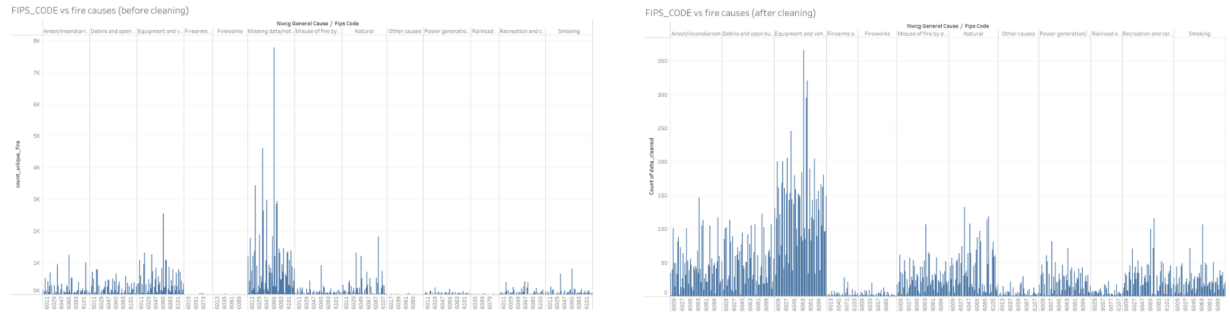
*Fips Codes and Fire Part of Day before and after Cleaning*




*Note.* The data looks more spread out after cleaning.

**Figure 44**

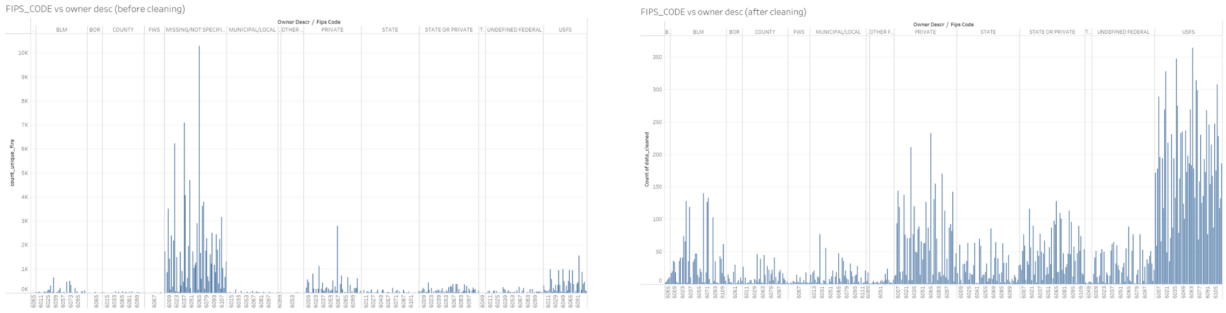
*Fips Codes and Fire General Causes before and after Cleaning*




*Note*. After cleaning, missing data are replaced with a non-null mode; so, equipment and vehicle use now is the most dominant.

**Figure 45**

*Fips Codes and Owner Description before and after Cleaning*



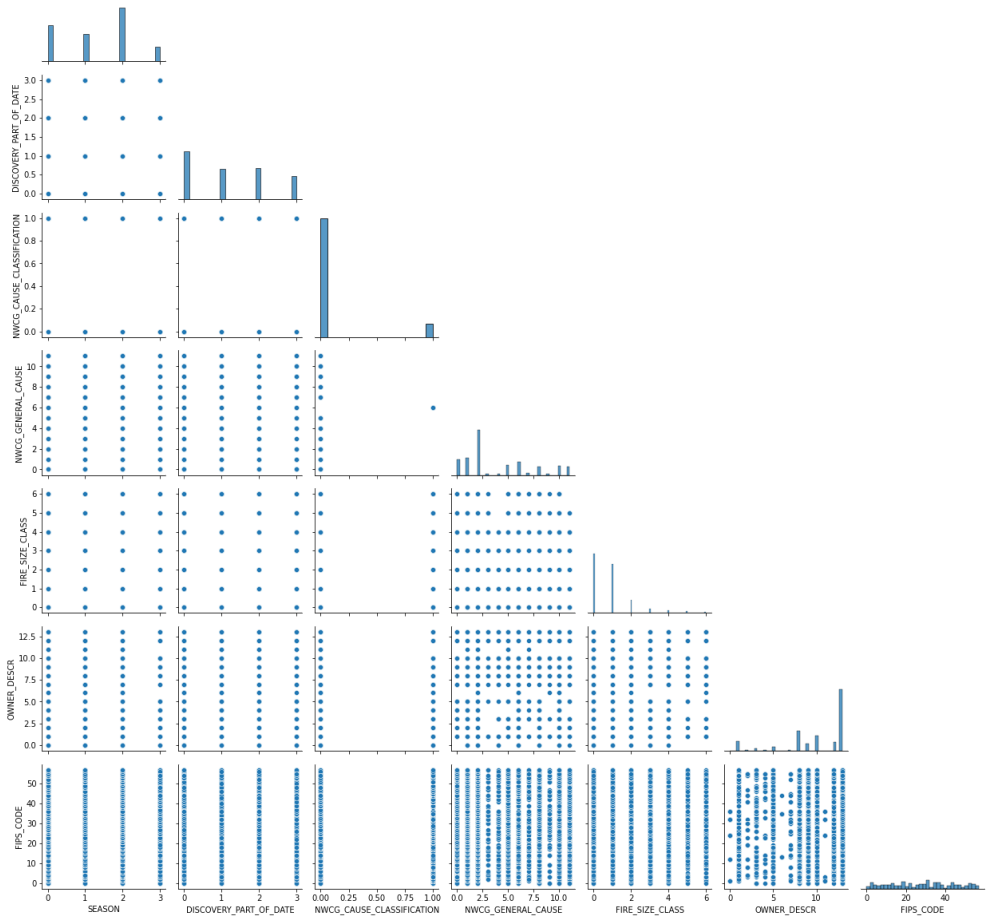

*Note.* After cleaning, missing data are replaced with a non-null mode; now, USPS and PRIVATE are the most dominant.

***Transformation Summary***

After the data is processed and cleaned, we transform the categorical data using ordinal encoding that is mapping each class level to an integer. Transformed data and mapping lookups are both stored in GCP storage and GCP BigQuery. Figure 46 shows the pairplot of transformed data implying there are no obvious patterns and relations between features. Therefore, Naive Bayes probably be the best choice for this problem since it assumes conditional independence between features.

**Figure 46**

*Pairplot of Transformed Data*




*Note.* Data looks all scattered around, no obvious patterns and relations.

***Data Preparation Summary***

After the data transformation phase, we will further use the undersampling technique to help with the imbalance dataset problem. After this process, we end up with a balanced dataset where each FIPS_CODE has the same amount of reported fire incidents as depicted in Figure 30. Then, during the modeling phase, for each selected region, we will split data into 80% of the training set and 20% of the test set. Also, for model validation purposes, we further use stratified K-fold technique to split training sets into folds of smaller training sets and validation sets.

# Chapter 4 - Model Development

## 4.1 - Model Proposals

### *Target Problem*

This problem is a multi-class categorical classification problem because we have 58 FIPS_CODE to predict given a user query. We will apply and compare Naive Bayes (from scratch and GaussianNB from scikit-learn), K-nearest neighbors (KNN), one vs rest machine learning models to predict the county which has the highest fire likelihood in an area given a user query.

A user can query about likelihood using a combination of factors of fire cause classifications (human or nature), fire season, fire reported part of date, fire general causes, and property type.

Expected output should include classified county codes (FIPS_CODE) and their probabilities. Table 8 shows an example of an expected output.

**Table 8**

*Expected Model Probability Outcomes*

| FIPS Codes (County) | Probability (%) |
|---|---|
| 06079 | 20 |
| 06085 | 18 |
| 06089 | 15 |
| 06099 | 14 |
| 06037 | 8 |

*Note.* This model will predict the most likely fire zones in each area.

### *Approaches*

We will group California FIPS codes into 15 regions and each of them consist of four FIPS codes; the order of grouping is county order on the map from top to bottom and left to right. For each region, a machine learning model will be used to predict the fips code that has the highest fire likelihood based on descriptive features. The final model will be an ensemble model from each region.

### *Concepts*

**Naive Bayes.** The Naive Bayes model according to Kelleher et al. (2015) returns a maximum a posteriori (MAP) prediction with conditional independence assumption. The Naive Bayes MAP is defined as equation 4:

$$\text{M(q)} = argmax_{l \,\epsilon\, levels(t)}((\prod_{i=1}^{m} P(q[i] \mid t = l)) \times P(t = l)) \tag{4}$$

where t is a label with a levels, levels(t), q is a query instance with features q[1], …, q[m].

The assumption of conditional independence of Naive Bayes is a simplifying assumption which is made regardless of correctness; however, it is still surprisingly found accurately in many domains. As a result of conditional independence assumption, the Naive Bayes is relatively good for data fragmentation and the curse of dimensionality problems.

Also, the Naive Bayes makes handling missing values easy since we just simply drop the conditional probabilities for events which use features taking values not in the data.

Additionally, the Naive Bayes is easy to train since we only need to calculate conditional probabilities for each feature given the target and the priors for each target. So, the Naive Bayes is fast and can represent large datasets.

One problem with Naive Bayes is that it can lead to zero probability issues for MAP prediction, and the solution for that is to use Laplace smoothing to distribute larger probabilities to smaller ones. Laplace smoothing is defined as equation 5:

$$P(f = l \mid t) = \frac{count(f = l \mid t) + k}{count(f \mid t) + (k \times |Domain(f)|)} \quad (5)$$

where count(f = l | t) is the count of event f = l happens in the subset of dataset given target level t, k is a hyperparameter, |Domain(f)| is the number of level of a feature, and count(f | t) is the count of the feature f given any level of target t. The larger the k, the more probability mass reduced from larger probabilities and given to smaller probabilities.

In this study, besides using Scikit learn's Gaussian Naive Bayes, we also use a Naive Bayes model designed from scratch. The algorithms for this model from cratch are described in Algorithm 1 and Algorithm 2.

**Algorithm 1**

*Naive Bayes Model Algorithm from Scratch*

```
Input: dataframe df, target, features, k_smoothing
Output:  model - dataframe of pre calculated conditional probabilities and prior probabilities

Initialize model with column post_p and p
For each i in df[target]'s level do
    model['post_p'] = string P(target=i)
    model['p'] =  calculate P(target=i)
    For each v in features do
        For each v1 in df[v]'s level do
            model['post_p'] = string P(v=v1 | target=i)
            Model['p'] = calculate  P(v=v1 | target=i) with laplace smoothing of k = k_smoothing
            as in equation 5
Return model
```

*Note.* Modeling process of Naive Bayes that build a detailed model based on prior probabilities and conditional probabilities with k smoothing.

**Algorithm 2**

*Naive Bayes Query Prediction Algorithm from Scratch*

```
Input: query, trained Naive Bayes model model, target, target level target_lv
Output:  y_pred, y_pred_p
```

```
For each i in target_lv do
    For each q in query do
        model  = filter for P(q's feature = q's value  | target = i) and P(target = i) in model
        Y_pred_p = Multiply all returned model['p'] as in equation 4
        Y_pred = i
y_pred, y_pred_p <- y_pred highest with highest y_pred_p
Return y_pred, y_pred_p
```

*Note.* Prediction process of Naive Bayes

**KNN.** According to Kelleher et al. (2015), the distance weighted KNN model is defined as equation 6:

$$M_k(q) = argmax_{l \epsilon levels(t)} \sum_{i=1}^{k} \frac{1}{dist(q, d_i)^2} \times \delta(t_i, l) \quad (6)$$

where $M_k(q)$ is the prediction of model M given query q and hyperparameter k; t is a label with a levels, levels(t) in target feature and l is an element in this set. The model iterate through $d_i$ in increasing distance from query q; $t_i$ is the target for $d_i$ instance. $\delta(t_i, l)$ is a Kronecker delta function that returns 1 if they are equals and 0 otherwise. The purpose of Kronecker delta function is to include only weights for instance whose target matches level $l$. The most commonly used distance measures for KNN can be Euclidean, Manhattan and Monkowski. Algorithm 3 depicts the pseudocode for KNN.

**Algorithm 3**

*KNN Pseudocode*

| Required: training data, a query instance |
| --- |
| Loop across the training data and find the nearest neighbor based on the shortest distance to the query instance.<br>Make a prediction for the query that is equal to the target's features value of the nearest neighbor. |

*Note.* KNN pseudocode describes the overall process to make predictions of a KNN algorithm (Kelleher et al., 2015).

**One vs Rest.** According to Aly (2015) and Scikit Learn (n.d, OneVsRestClassifier), one-vs-rest or also known as one-vs-all is a multi-class prediction strategy that fits one classifier per class. Let K is the number of classes in the target feature, one-vs rest can create K binary SVM classifiers to distinguish one class from the other K-1 classes. This method is considered the most commonly used strategy and a good default choice.

### 4.2 - Model Support

#### *Platform and Tools*

This project use following resources for model development:

- GCP storage and GCP bigquery.
- Google colab.
- Python machine learning libraries: scikit learn GaussianNB, KNeighborsClassifier, OneVsRestClassifier, SVC.
- Python StratifiedKFold library.

#### *Data Flow*

Data flow process can be described in sequential steps below:

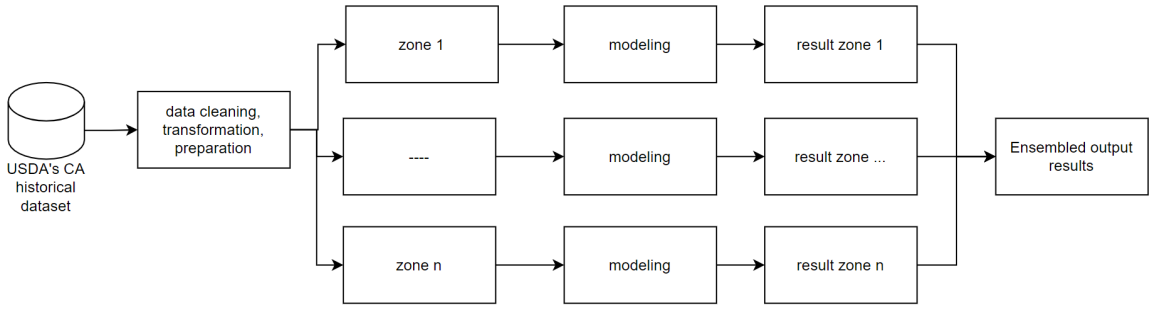
- Step 1: Extract California data from original USDA dataset
- Step 2: Perform data exploration, handle any found data issues, apply data transformation and perform data preparation for modeling

- Step 3: For each selected zone, split data into train set and test set with a ratio of 80:20
- Step 4: Train the model with the training data set.
- Step 5: Using stratified K-fold to validate the models for the best tuning hyperparameters.
- Step 6: Evaluate the performance with test set

The results from each zone prediction will be ensembled eventually for a final prediction as a tabular format depicted in Table 8. Figure 47 and Figure 48 show the overall data flow process and modeling data flow for each zone respectively.

**Figure 47**

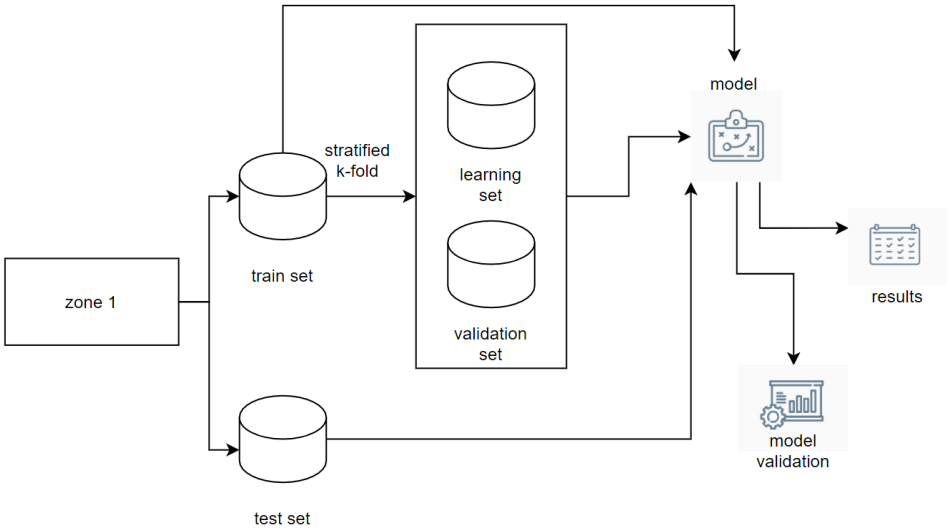
*Overall Data Modeling Data Flow*




*Note*. General data flow process. Final output result is an ensembled result from each zone prediction.

**Figure 48**

*Data Modeling Data Flow for Each Zone*




*Note.* Overall data flow during the modelling phase with example of zone 1.

### 4.3 - Model Comparison and Justification

#### *Similarities*

All three models are suitable for classification problems, but one-vs-rest are dedicated to multi-class classification problems. Both categorical and continuous data can be used for these three models.

#### *Advantages and Disadvantages*

Table 9 shows comparison between chosen models in terms of advantages and disadvantages.

**Table 9**

*Applied Model Comparisons*

| Model | Advantages | Disadvantages |
|---|---|---|
| Naive Bayes | Easy to implement and understand.<br>Fast to train.<br>Suitable for solving multi-class problems.<br>Suitable for categorical dataset while being robust to the curse of dimensionality. | Conditional independent assumption since such a dataset is hard to find in real life.<br>Susceptible to zero probability problem.<br>Eager learning: abstraction during training will go out of date, so retraining after the intervals is required and can be costly. |
| KNN | Easy to interpret.<br>Can handle both categorical and continuous features.<br>Lazy learner: can update without retraining.<br>Robust to concept drift (relationship between features and target change over time) | Sensitive to the curse of dimensionality.<br>Slow at making predictions (with large datasets). |
| One-vs-rest | Easy to interpret since each class only has one classifier. | Memory required is high that may raise problems with large datasets and memory.<br>Training sample can be imbalance due to the ratio of training samples to rest of the classes is 1: K-1 where K is total numbers of class |

*Note. Naive bayes, KNN comparisons are according to* Kelleher et al (2015), one-vs-rest comparison is according to Aly (2015) and Scikit Learn (n.d, OneVsRestClassifier).

***Justifications***

Table 10 shows justifications of chosen models for this project.

**Table 10**

*Model Justifications*

| Model | Justification |
|---|---|
| Naive Bayes | Easy to interpret method, and straightforwardly easy to calculate probabilities for each prediction. Therefore, the initial baseline accuracy would be easy to understand.<br>Suitable for categorical data (this dataset consists of all categorical features).<br>Naive Bayes helps with the curse of dimensionality (this dataset consists of seven features and the FIPS_CODE alone consists of 58 distincts levels). |
| KNN | Easy to interpret model.<br>This model will be beneficial for future deployment and maintenance because of lazy learner robustness of concept drift nature which means re-trainings are not required. |
| One-vs-rest | The model is a good default choice and dedicated for multi-class classification problems. |

*Note*. These justification are based on the nature of models, advantages and disadvantages.

**4.4 - Model Evaluation Methods**

We will look at muti-class classification evaluation metrics such as average accuray, error rate, macro precision, macro recall, and macro F1-score. Table 11 describes all evaluation metrics used for this problem.

**Table 11**

*Multi-class Evaluation Methods by* Sokolova and Lapalme (2009).

| Metric | Formula | Explanation |
|---|---|---|
| Average accuracy | $\frac{\sum_{i=1}^{k} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{k}$ | The average per-class accuracy of a classifier |
| Error rate | $\frac{\sum_{i=1}^{k} \frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i}}{k}$ | The average per-class classified error rate |
| Macro precision | $\frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FP_i}}{k}$ | The average per class agreement of true labels with classifier's ones |
| Macro recall | $\sum_{i=1}^{k} \frac{TP_i}{TP_i + FN_i}$ | The average per-class effectiveness of the classifier in identifying labels |
| Macro F1-score | $\frac{2 * macro\ \ precision * macro\ recall}{macro\ precision\ + macro\ recall}$ | The harmonic mean of macro recall and macro precision. |

*Note.* TP, TN, FP, FN stands for True Positive, True Negative, False Positive, False Negative respectively.

Two main metrics we focus on here are the average accuracy and the F1-score. For this particular problem, we want average accuracy and F1-score as high as possible because they indicate model performance. At this stage, we expect a baseline average accuracy score of 70%. Besides, we want error rate and macro recall to be as low as possible as they indicate wrongly classifications of models and macro precision as high as possible as it indicates true classifications of models.

### 4.5 - Model Validation and Evaluation

#### *Tuning Results*

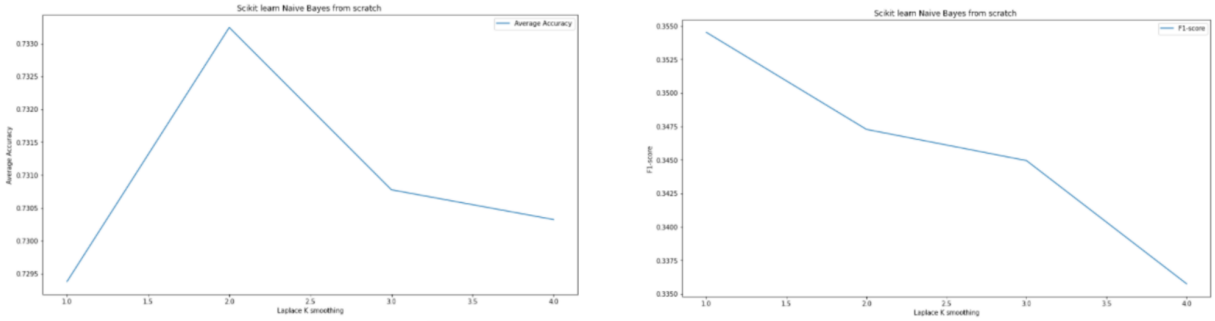
According to the tuning results in Figures 49, 50, 51 and 52, the final models used for testing evaluation will be set on these hyperparameters:

- Naive Bayes from scratch: Laplace k kmoothing = 2.
- Scikit Learn’s Naive Bayes GaussianNB: var_smoothing = 1.

- Scikit Learn’s KNN: n_neighbors = 25.
- Scikit Learn’s One-vs-rest: estimator= SVC().

**Figure 49**

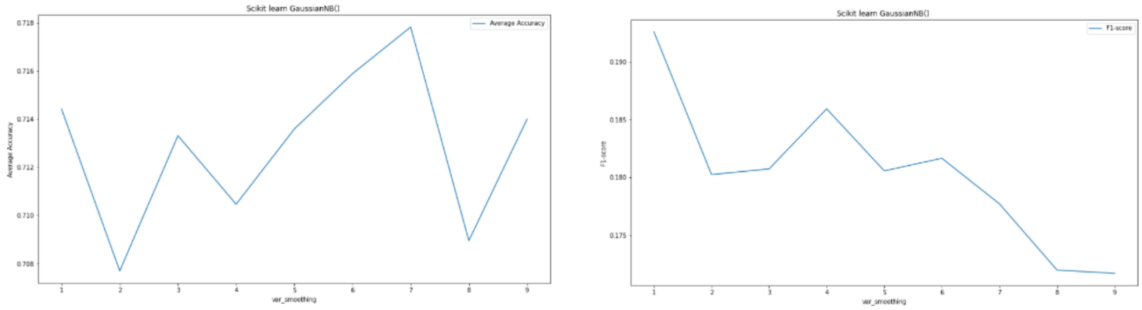
*Naive Bayes from Scratch Tuned on Laplace K Smoothing*




*Note.* In Naive Bayes from scratch, we pick laplace smoothing k = 2 due to the best achieved performance compared between overall accuracy and F1-score.

**Figure 50**

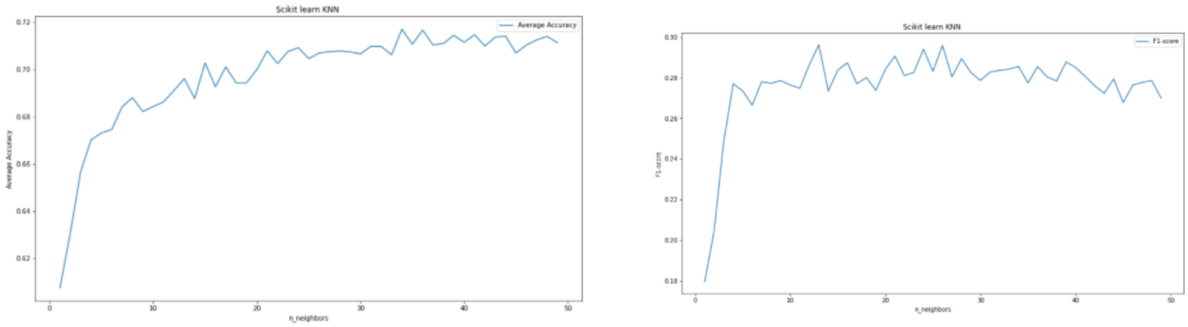
*Naive Bayes from Scikit Learn Tuned on var_smoothing*




*Note.* In a model using scikit-learn's GaussianNB() , we pick var_smoothing k = 1 due to the best achieved performance compared between overall accuracy and F1-score.

**Figure 51**

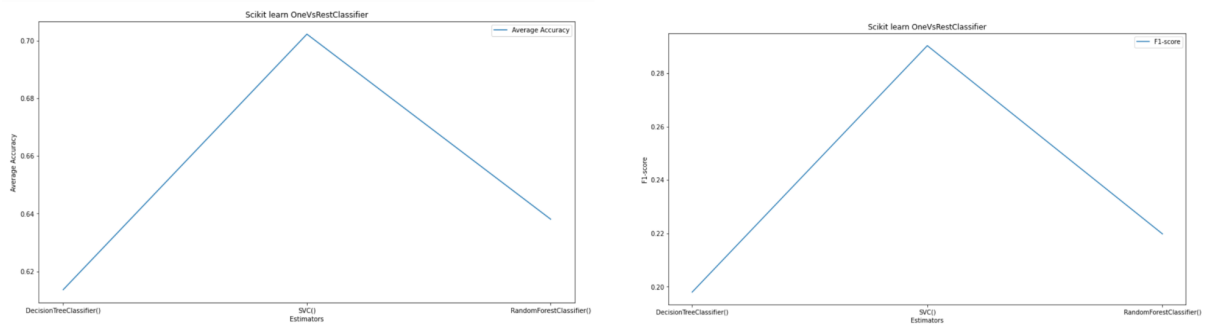
*KNN from Scikit Learn Tuned on n_neighbors*




*Note.* In a model using scikit-learn's KNN KNeighborsClassifier(), we pick n_neighbors = 25 due to the best achieved performance compared between overall accuracy and F1-score. At n_neighbors = 25, F1-score starts to drop.

**Figure 52**

*One-vs-rest from Scikit Learn Tuned on Estimator*




*Note.* In a model using scikit-learn's One-vs-rest OneVsRestClassifier() , we pick estimator = SVC() due to the best achieved performance compared between overall accuracy and F1-score.

### *Final Result*

Table 12 shows final evaluation results for measures mentioned in Table 11.

**Table 12**

*Final Evaluation Results*

| Models | Average Accuracy | Error Rate | Macro Precision | Macro Recall | F1 Score |
|---|---|---|---|---|---|
| Naive Bayes from scratch | 0.73 | 0.27 | 0.40 | 0.36 | 0.35 |
| Scikit learn's gaussianNB | 0.71 | 0.29 | 0.18 | 0.28 | 0.19 |
| Scikit learn's KNN | 0.70 | 0.30 | 0.34 | 0.30 | 0.28 |
| Scikit learn's One-vs-rest | 0.70 | 0.30 | 0.30 | 0.31 | 0.28 |

*Note.* Results from final model evaluation against test set. Models in this stage used hyperparameters tuned using stratified K-fold.

### *Model Development Conclusion*

From the evaluation results depicted in Table 12, we can conclude that the Naive Bayes model from scratch achieves the best performance with highest average accuracy, macro precision and F1 score; hence, we will apply this model as a baseline model to predict county fire zones for this project. However, the low F1-score indicates that the model is still suffering from underfitting problems. Therefore, the next modeling improvement version would be understanding and improving this issue.

**References**

Aly, M. (2005). Survey on multiclass classification methods. *Neural Netw*, *19*, 1-9.

Cal Fire. (2021). *Top 20 Largest California Wildfires*. **https://www.fire.ca.gov/media/4jandlhh/top20_acres.pdf**

Ciaburro, G., & Venkateswaran, B. (2017). *Neural Networks with R*.

DMPTool. (n.d). *Write Plan*. Retrieved October 8, 2021, from https://dmptool.org/plans

Google Cloud. (n.d). *BigQuery*. Retrieved October 8, 2021, from https://cloud.google.com/bigquery

Google Cloud. (n.d). *BigQuery pricing*. Retrieved October 8, 2021, from https://cloud.google.com/bigquery/pricing

Google Cloud. (n.d). *Cloud Run*. Retrieved October 8, 2021, from https://cloud.google.com/run

Google Cloud. (n.d). *Cloud Storage*. Retrieved October 8, 2021, from https://cloud.google.com/storage

Google Cloud. (n.d). *Cloud Storage pricing*. Retrieved October 8, 2021, from https://cloud.google.com/storage/pricing

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, *3*(5), 605-610.

Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

Kelleher, J., & Namee, B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics*. The MIT Press.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11-26.

Pacheco, A. D. P., Junior, J. A. D. S., Ruiz-Armenteros, A. M., & Henriques, R. F. F. (2021). Assessment of k-Nearest Neighbor and Random Forest classifiers for mapping forest fire areas in central Portugal using Landsat-8, Sentinel-2, and Terra Imagery. *Remote Sensing*, *13*(7), 1345.

Park, Y. S., & Lek, S. (2016). Artificial neural networks: multilayer perceptron for ecological modeling. In *Developments in environmental modelling* (Vol. 28, pp. 123-140). Elsevier.

Parsian, M. (2015). K-Nearest Neighbors. *Data Algorithms*. O'Reilly Media, Inc.

Pham, B.T., Jaafari, A., Avand, M., Al-Ansari, N., Dinh Du, T., Yen, H.P.H., Phong, T.V., Nguyen, D.H., Le, H.V., Mafi-Gholami, D., Prakash, I., Thi Thuy, H. & Tuyen, T.T.(2020). Performance Evaluation of Machine Learning Methods for Forest Fire Modeling and Prediction. *Symmetry*, *12*(6), 1022. doi:10.3390/sym12061022

Prestemon, J. P., & Butry, D. T. (2010). Wildland arson: a research assessment. *In: Pye, John M.; Rauscher, H. Michael; Sands, Yasmeen; Lee, Danny C.; Beatty, Jerome S., tech. eds. Advances in threat assessment and their application to forest and rangeland management. Gen. Tech. Rep. PNW-GTR-802. Portechtland, OR: US Department of Agriculture, Forest Service, Pacific Northwest and Southern Research Stations: 271-283*, *802*, 271-283.

Safi, Y., & Bouroumi, A. (2013). Prediction of forest fires using artificial neural networks. *Applied Mathematical Sciences*, *7*(6), 271-286.

Scikit Learn. (n.d.). *1.17. Neural network models (supervised)*. **https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron**

Scikit Learn. (n.d.). *OneVsRestClassifier*. **https://scikit-learn.org/stable/modules/multiclass.html#ovr-classification**

Short, Karen C. 2021. *Spatial wildfire occurrence data for the United States, 1992-2018 [FPA_FOD_20210617]. 5th Edition. Fort Collins, CO: Forest Service Research Data Archive*. **https://doi.org/10.2737/RDS-2013-0009.5**

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437.

Sun, J., Du, W., & Shi, N. (2018). A Survey of kNN Algorithm. *Information Engineering and Applied Computing*, *1*(1).

Tableau. (n.d). *Pricing for data people*. Retrieved October 8, 2021, from https://www.tableau.com/pricing/individual