# California Wildfire Prediction

## Team name - Placeholder

## Team members - Chinmayi Sunku, Ujwala Mote, Nihal Kaul, Suma Nagral

### Roles

Every member of the team contributed to each of the homework, documentation, and papers which resulted in this project submission, a culmination of all our work.

Link to Colab: https://drive.google.com/file/d/1kEiHOG6qq3LT3Xi5OeKxxWbz-ah7ENPJ/view?usp=sharing

# Project Description

- Wildfires in California pose a significant threat to lives, property, and the environment. Our primary goal is to utilize machine learning to offer investors in California guidance and insights related to wildfires for making informed decisions when purchasing investment properties.
- In our business narrative, we aim to empower real estate professionals and investors in California by integrating advanced ML models, like clustering, classification, and regression to assess wildfire risk accurately.
- In the data narrative, we emphasize the need for comprehensive datasets that encompass historical wildfire data, whether in various California counties, etc. factors to enable data-driven decision-making.

### Experiments

a. Predict wildfires based on weather conditions and a few other factors.
b. Which California counties are more prone to wildfires?
c. Fire Danger Index and Fuel Moisture and how these values affect the predictions
d. Temperature and precipitation yearly patterns in California

# Data

Our data narrative draws from two primary sources: **Wikipedia and the National Oceanic and Atmospheric Administration (NOAA)**. Wikipedia serves as a comprehensive knowledge base, providing background information and historical context related to wildfires in California. It offers insights into the various aspects of wildfires, including their causes, impacts, and trends. On the other hand, NOAA is a trusted source for weather-related data, offering a wealth of meteorological information crucial for understanding wildfire ignition and spread. This includes real-time data on temperature, humidity, wind speed, precipitation, and other relevant variables that play a vital role in wildfire prediction and risk assessment.

## Data Set 1

Wiki sources link (we are scraping the data from links like this for each year): https://en.wikipedia.org/wiki/2020_California_wildfires

## Data set 2

Temperature: https://drive.google.com/file/d/1uMivJmKg5MrlDIM49vWcHF9nk_jP3Bk-/view?usp=sharing

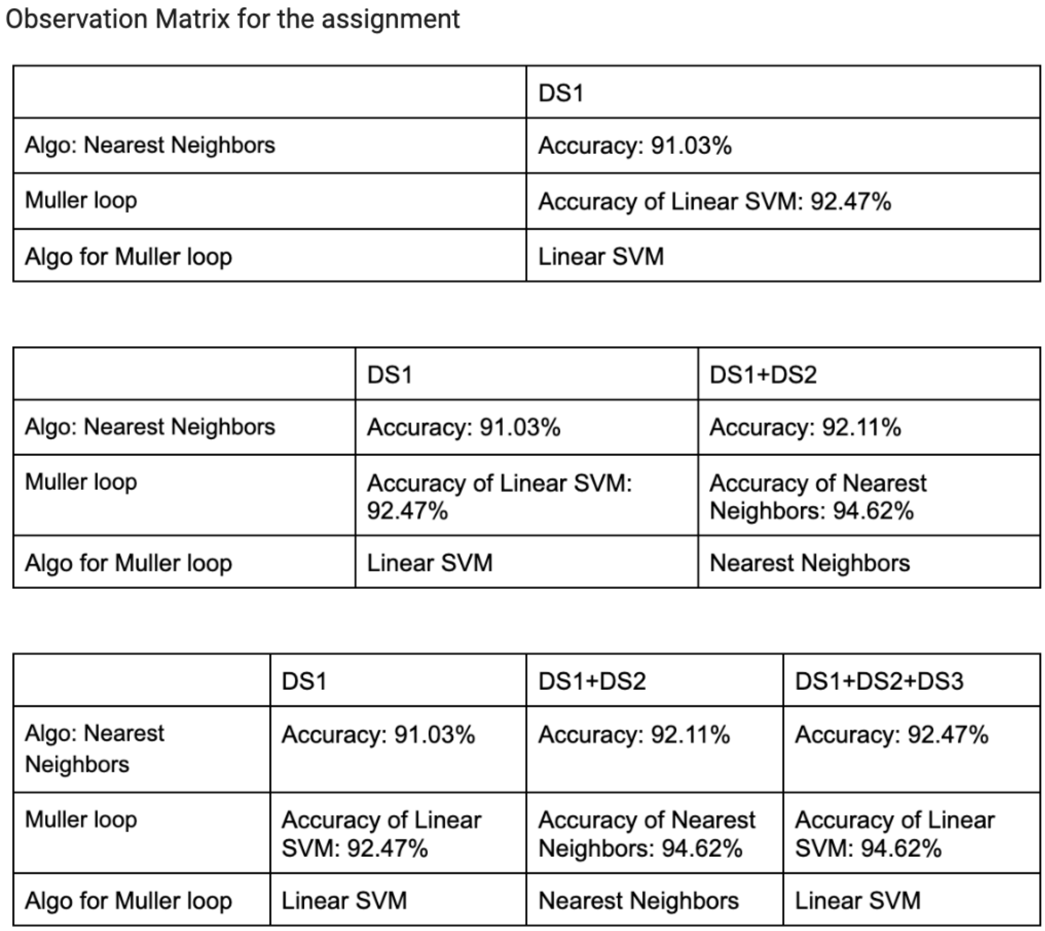
## Data set 3

Precipitation: https://drive.google.com/file/d/1XztOWTvoaHI883WIRHBaxg8V1HAUk0MO/view?usp=sharing

## California counties:

https://drive.google.com/file/d/1JiSZIjOPbE6SScNiFmyclIMtxn5MNBFf/view?usp=sharing

## Discussion on Amalgamation of the Datasets



Observation Matrix for the assignment

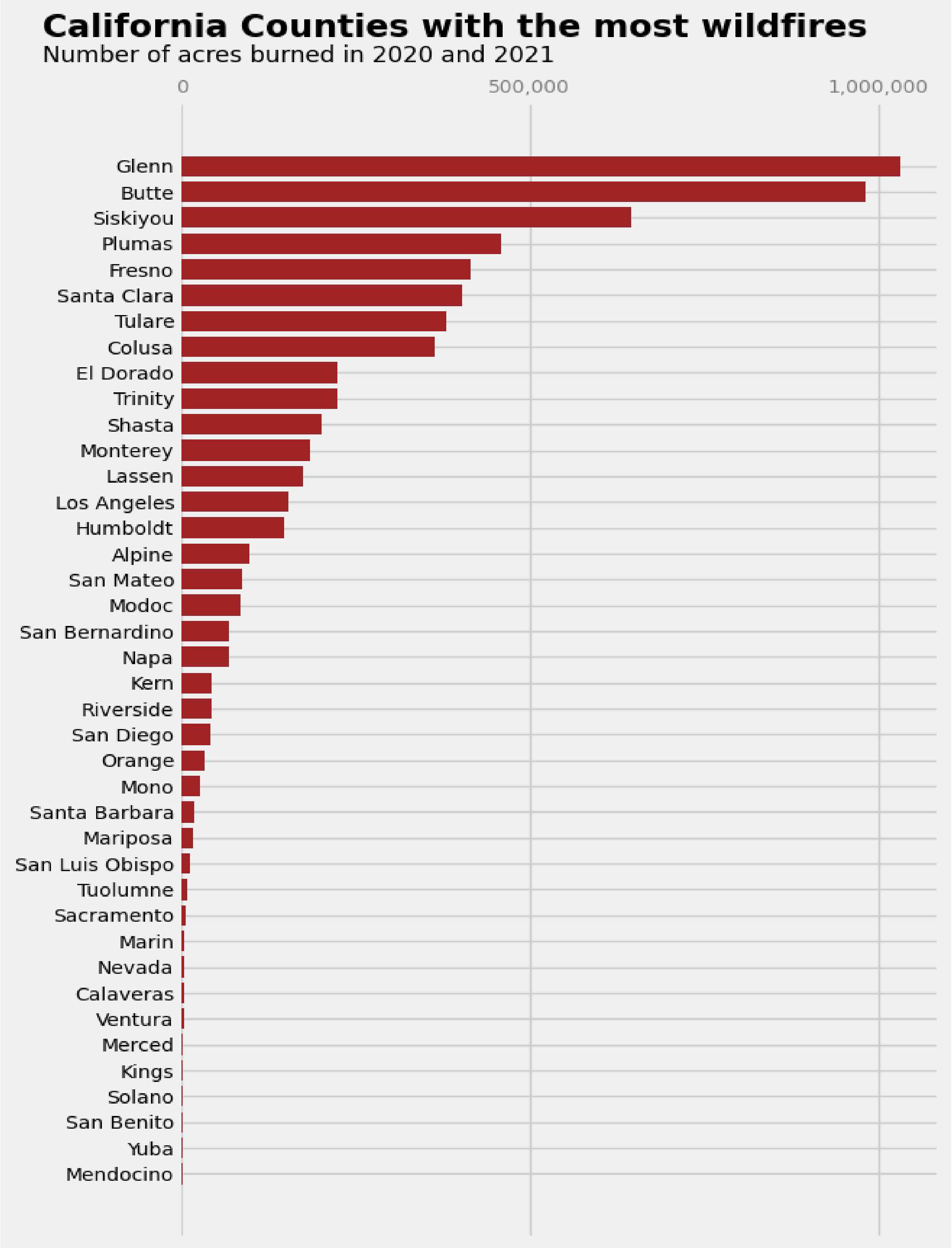
| | DS1 |
|---|---|
| Algo: Nearest Neighbors | Accuracy: 91.03% |
| Muller loop | Accuracy of Linear SVM: 92.47% |
| Algo for Muller loop | Linear SVM |

| | DS1 | DS1+DS2 |
|---|---|---|
| Algo: Nearest Neighbors | Accuracy: 91.03% | Accuracy: 92.11% |
| Muller loop | Accuracy of Linear SVM: 92.47% | Accuracy of Nearest Neighbors: 94.62% |
| Algo for Muller loop | Linear SVM | Nearest Neighbors |

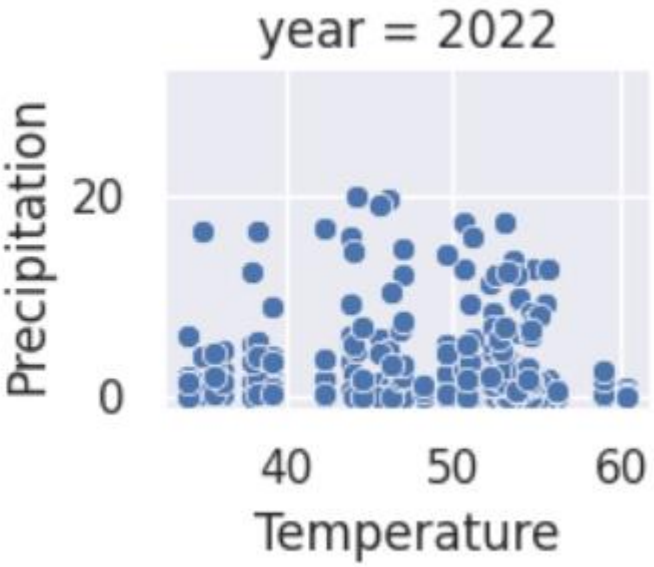
| | DS1 | DS1+DS2 | DS1+DS2+DS3 |
|---|---|---|---|
| Algo: Nearest Neighbors | Accuracy: 91.03% | Accuracy: 92.11% | Accuracy: 92.47% |
| Muller loop | Accuracy of Linear SVM: 92.47% | Accuracy of Nearest Neighbors: 94.62% | Accuracy of Linear SVM: 94.62% |
| Algo for Muller loop | Linear SVM | Nearest Neighbors | Linear SVM |

Based on the above analysis we can see that with just dataset 1 (wildfires) Linear SVM was a more suitable classification model with an accurate prediction of 92.47%. With dataset 1 and dataset 2 (wildfires + temperature), we can see that nearest neighbors are more accurate with 94.26%. With DS1 + DS2 + DS3 (wildfires + temperature + precipitation) we can see that the linear SVM accuracy increased to 94.62%. Which ensures that the factors we added/ amalgamated are adding to the prediction of wildfires.

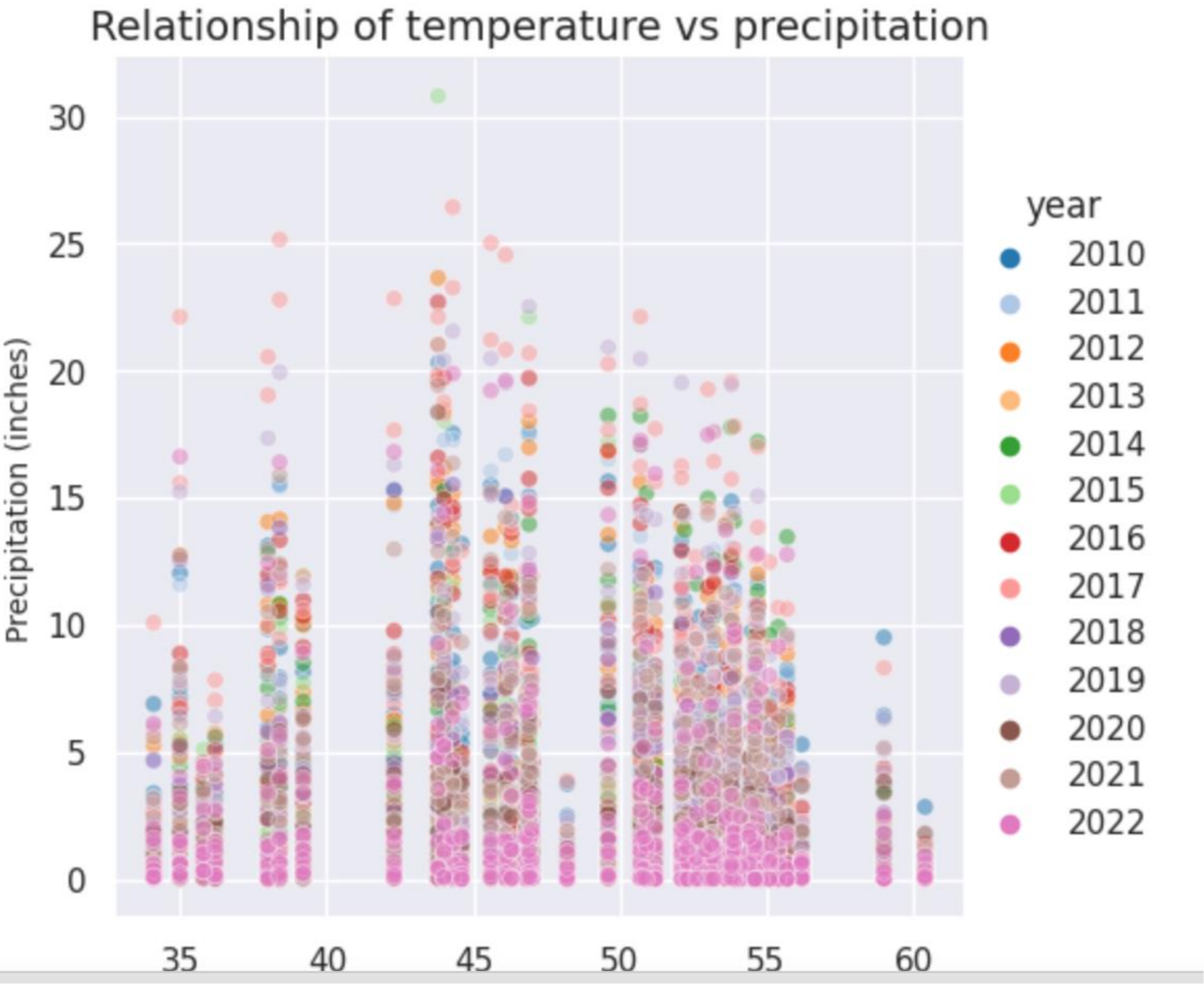
Exploratory Data Analysis and Visualization



Temperature and Precipitation Yearly Pattern

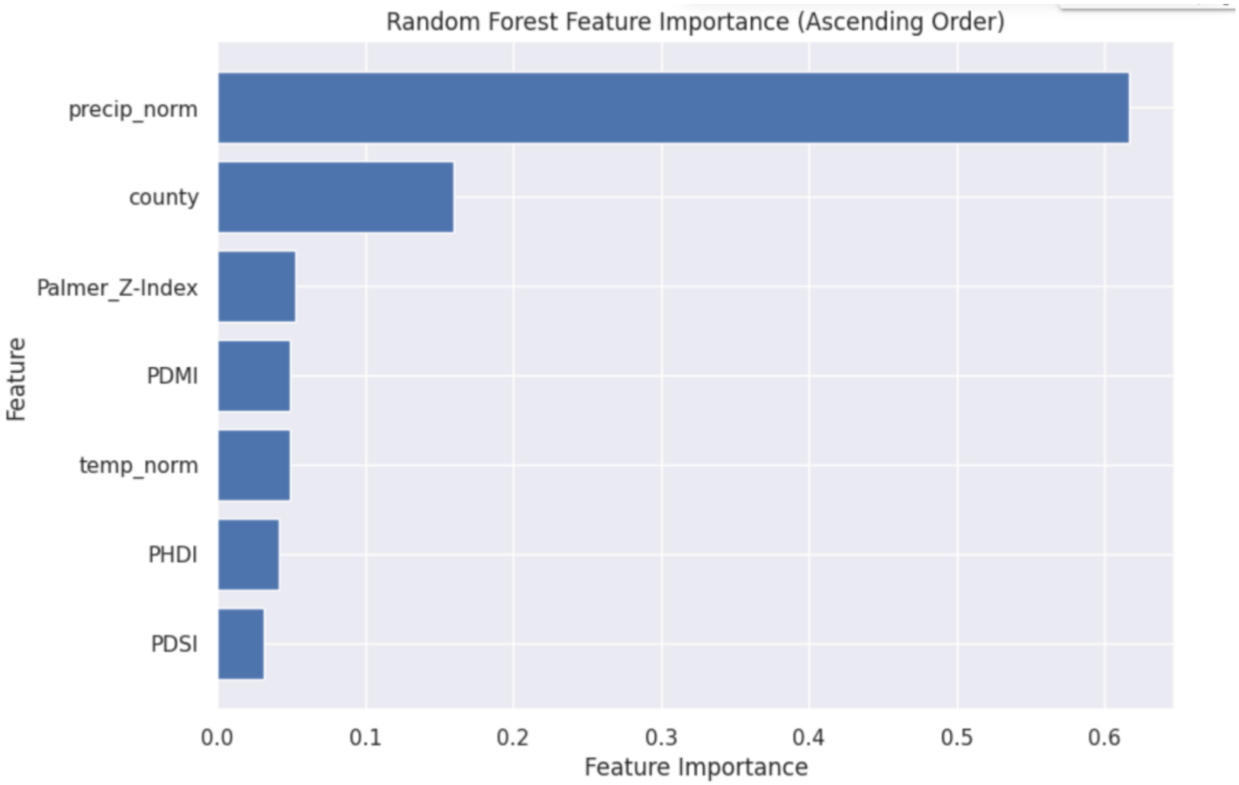


## Relationship of Temperature vs Precipitation

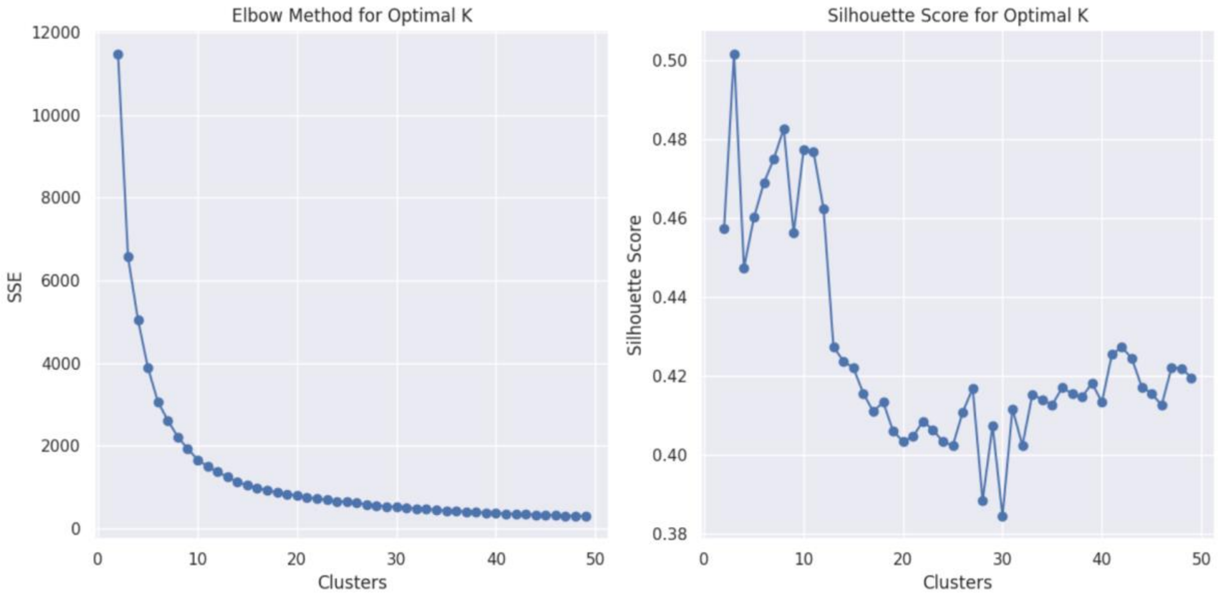


From the graph it can be inferred that as the temperature increases precipitation decreases.

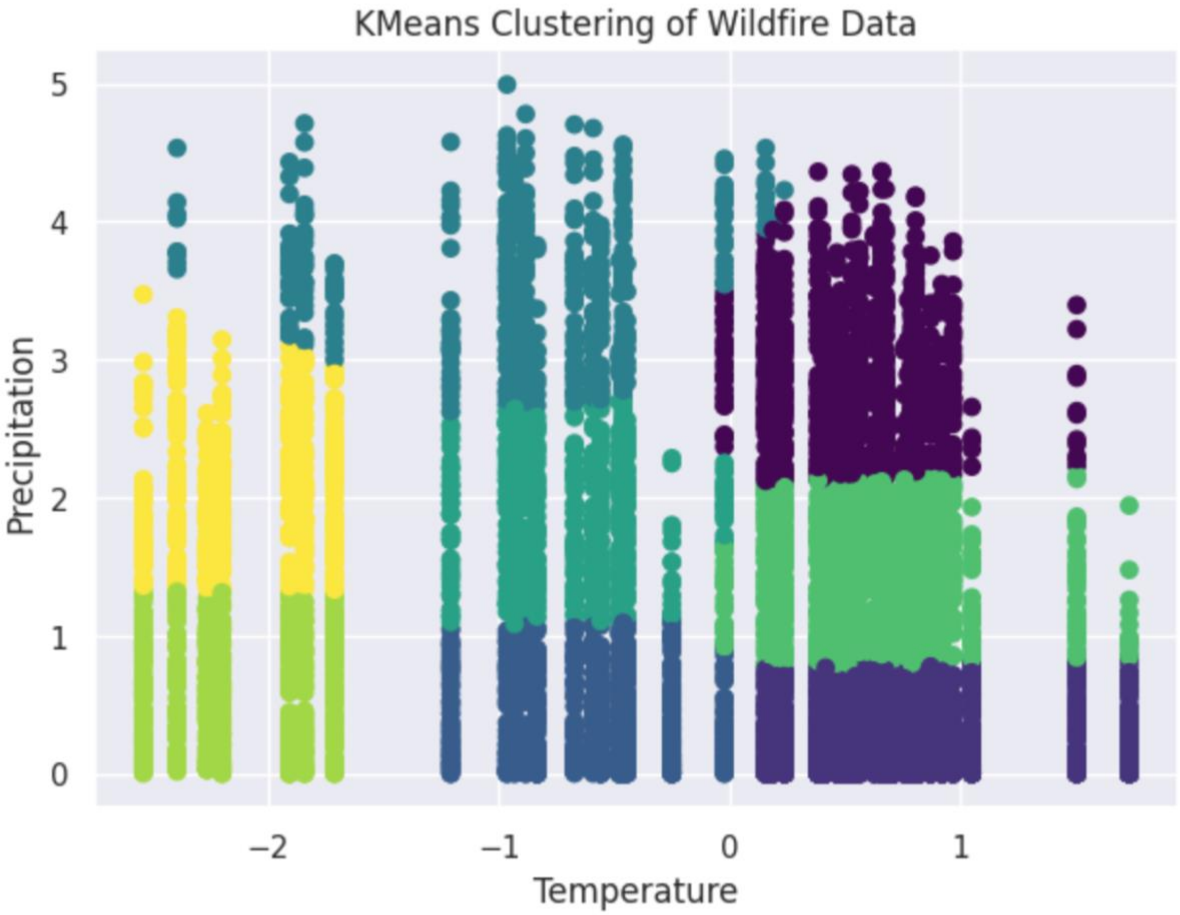
## Feature Importance



# Clustering

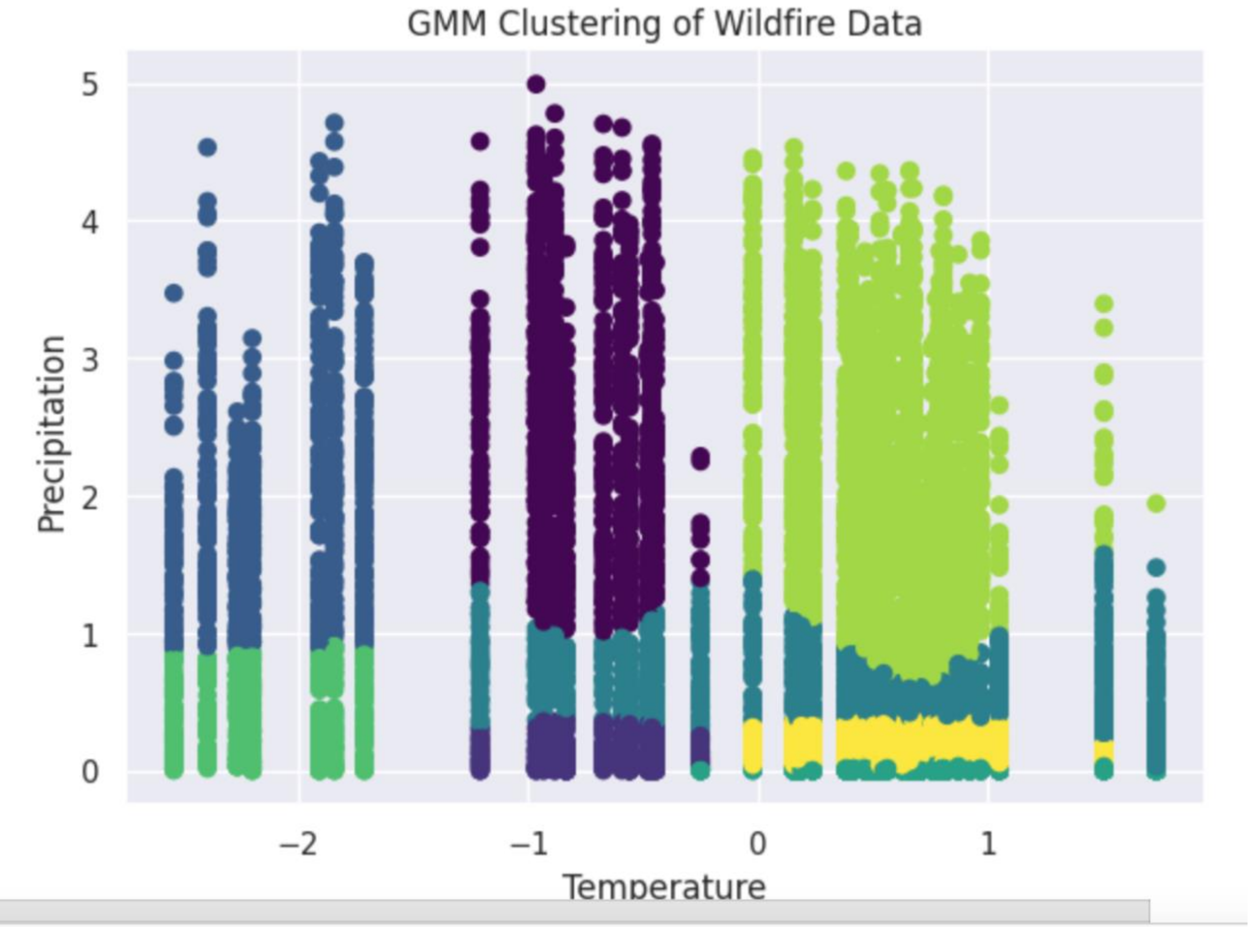


From the above SEE and Silhouette graph k = 8 seems like a good choice. There is a drastic change after k=12 in the Silhouette graph and an elbow at 8 in the elbow curve.



GMM clustering

It is a probability distribution that is symmetric around its mean. It is characterized by a specific probability density function that gives the likelihood of observing a particular value in a continuous dataset.



If we observe both Kmeans and GMM cluster graphs look similar. K-means and Gaussian Mixture Model (GMM) clusters look similar because both algorithms are deterministic in nature and both the models were tested using the same dataset and hyper parameter

## Golden Cluster

Based on the above analysis we can conclude that cluster 0 is the 'Golden Cluster' since it has the most number of wildfires. We can utilize this information in real estate to identify the regions that are more prone to wildfires, this helps us to make an informed decision and assess the risks of properties realtors would like to invest in.

**What is the objective function?**

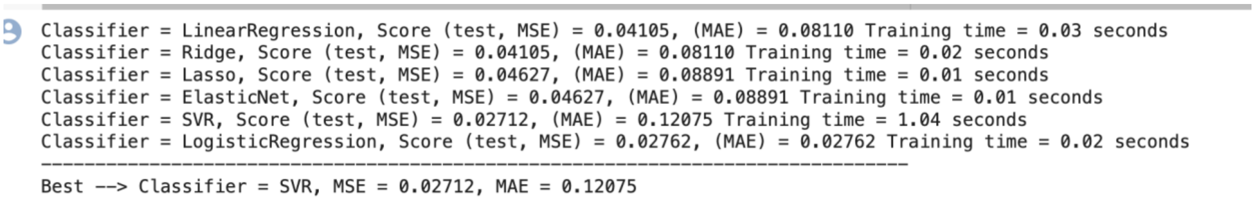
# Latent Variables and Manifolds

Fire Suppression Capability: This latent variable could represent the effectiveness of fire suppression efforts in a region. It might include factors like the availability of firefighting resources, response times, and firefighting strategies.

Environmental Conditions: Latent variables related to weather and environmental conditions can capture factors like temperature, humidity, wind speed, and precipitation patterns, which play a crucial role in fire behavior.

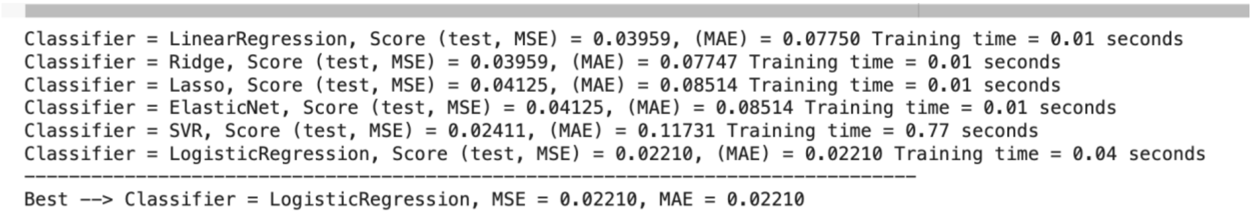
- FDI: A composite measure that integrates weather-related variables (temperature, precipitation, drought indices like PDSI) to estimate the potential fire risk.
  - Manifest variables: temperature, precipitation and PDSI

  Muller loop result:



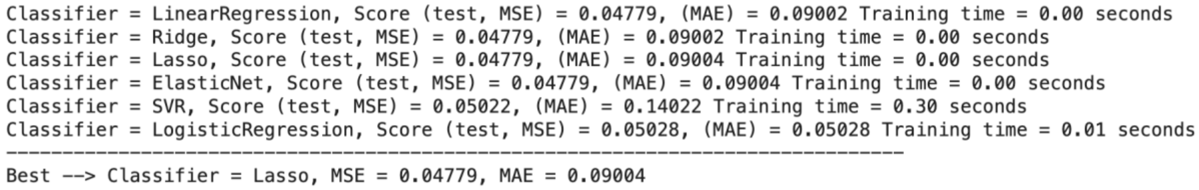
```
Classifier = LinearRegression, Score (test, MSE) = 0.04105, (MAE) = 0.08110 Training time = 0.03 seconds
Classifier = Ridge, Score (test, MSE) = 0.04105, (MAE) = 0.08110 Training time = 0.02 seconds
Classifier = Lasso, Score (test, MSE) = 0.04627, (MAE) = 0.08891 Training time = 0.01 seconds
Classifier = ElasticNet, Score (test, MSE) = 0.04627, (MAE) = 0.08891 Training time = 0.01 seconds
Classifier = SVR, Score (test, MSE) = 0.02712, (MAE) = 0.12075 Training time = 1.04 seconds
Classifier = LogisticRegression, Score (test, MSE) = 0.02762, (MAE) = 0.02762 Training time = 0.02 seconds
--------------------------------------------------------------------------------
Best --> Classifier = SVR, MSE = 0.02712, MAE = 0.12075
```

- Fuel Moisture: The moisture content of vegetation and fuel sources in a given area. It affects the ease with which fires can ignite and spread.
  - Manifest variables: PDSI



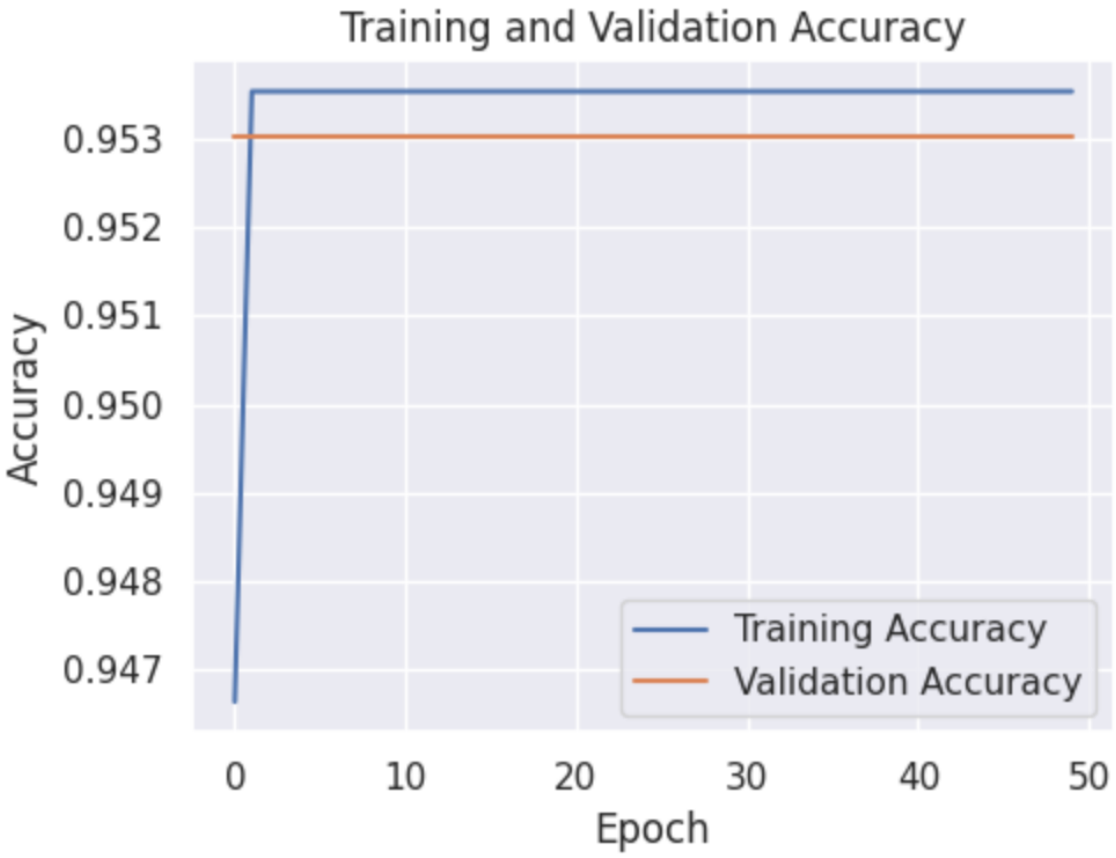
```
Classifier = LinearRegression, Score (test, MSE) = 0.03959, (MAE) = 0.07750 Training time = 0.01 seconds
Classifier = Ridge, Score (test, MSE) = 0.03959, (MAE) = 0.07747 Training time = 0.01 seconds
Classifier = Lasso, Score (test, MSE) = 0.04125, (MAE) = 0.08514 Training time = 0.01 seconds
Classifier = ElasticNet, Score (test, MSE) = 0.04125, (MAE) = 0.08514 Training time = 0.01 seconds
Classifier = SVR, Score (test, MSE) = 0.02411, (MAE) = 0.11731 Training time = 0.77 seconds
Classifier = LogisticRegression, Score (test, MSE) = 0.02210, (MAE) = 0.02210 Training time = 0.04 seconds
--------------------------------------------------------------------------------
Best --> Classifier = LogisticRegression, MSE = 0.02210, MAE = 0.02210
```

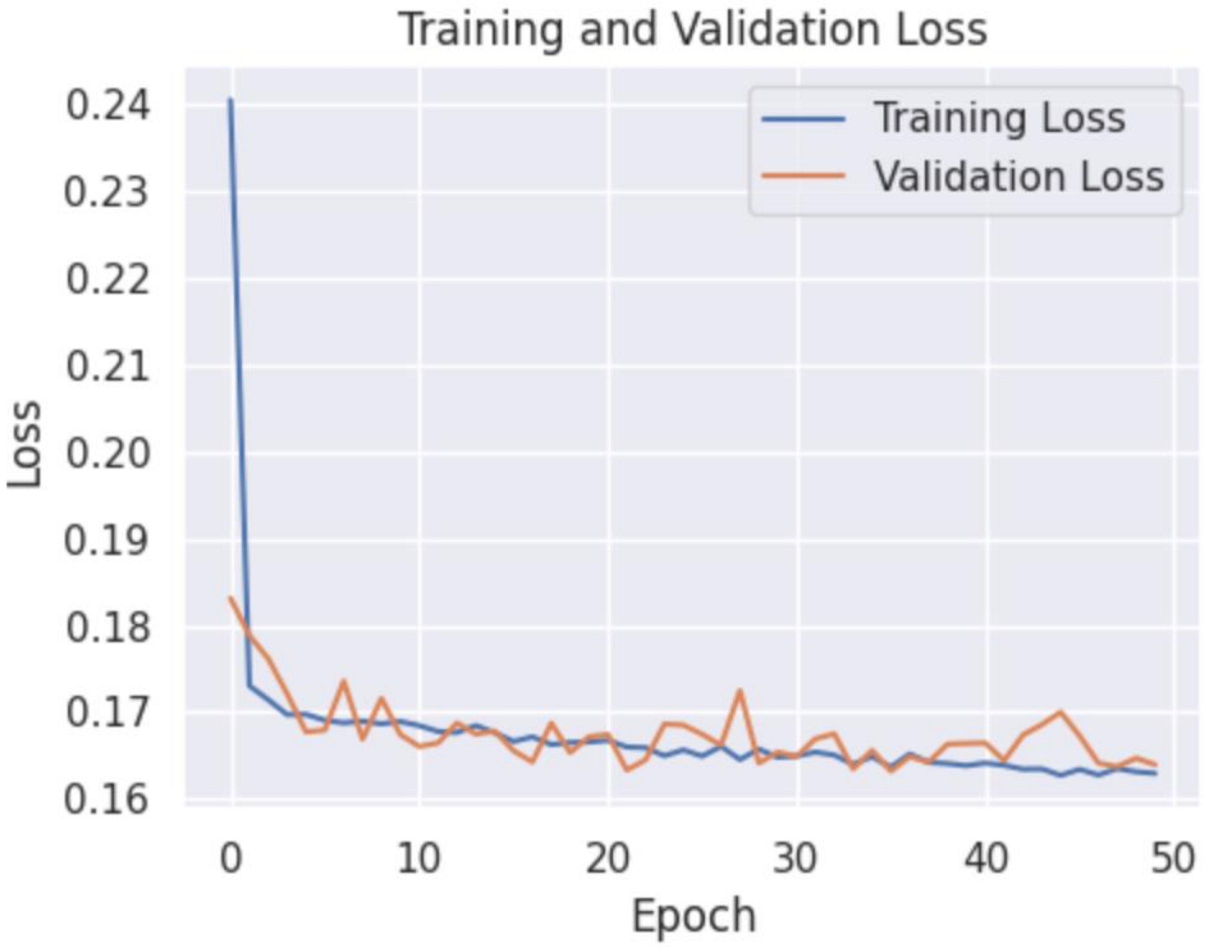
- Weather Moisture: Temperature: The temperature at the time of the wildfire.Relative Humidity is referred to the humidity level in the area.
    - Manifest variables: temperature and precipitation



```
Classifier = LinearRegression, Score (test, MSE) = 0.04779, (MAE) = 0.09002 Training time = 0.00 seconds
Classifier = Ridge, Score (test, MSE) = 0.04779, (MAE) = 0.09002 Training time = 0.00 seconds
Classifier = Lasso, Score (test, MSE) = 0.04779, (MAE) = 0.09004 Training time = 0.00 seconds
Classifier = ElasticNet, Score (test, MSE) = 0.04779, (MAE) = 0.09004 Training time = 0.00 seconds
Classifier = SVR, Score (test, MSE) = 0.05022, (MAE) = 0.14022 Training time = 0.30 seconds
Classifier = LogisticRegression, Score (test, MSE) = 0.05028, (MAE) = 0.05028 Training time = 0.01 seconds
--------------------------------------------------------------------------------
Best --> Classifier = Lasso, MSE = 0.04779, MAE = 0.09004
```

Multimodel MLP :





Model is trained, and its performance is evaluated using Keras Algorithm. The evaluation visualized is in the form of subplots for accuracy and loss during training. It will help us visualize how your model's performance changes over the training epochs.

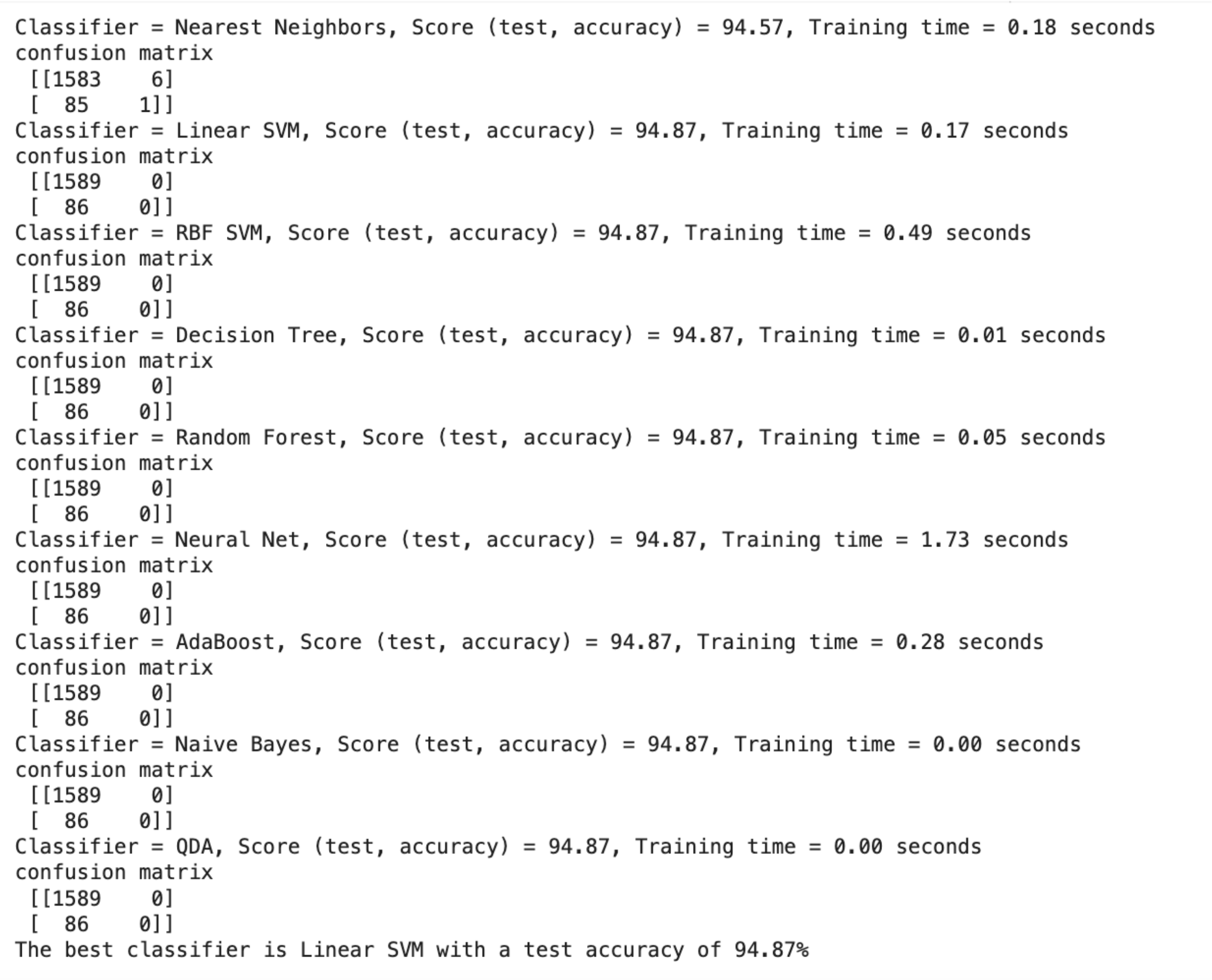
**What metrics are you using?**

To predict the best possible results we are utilizing metrics such as precision, recall, f1-score and support.

# Classification

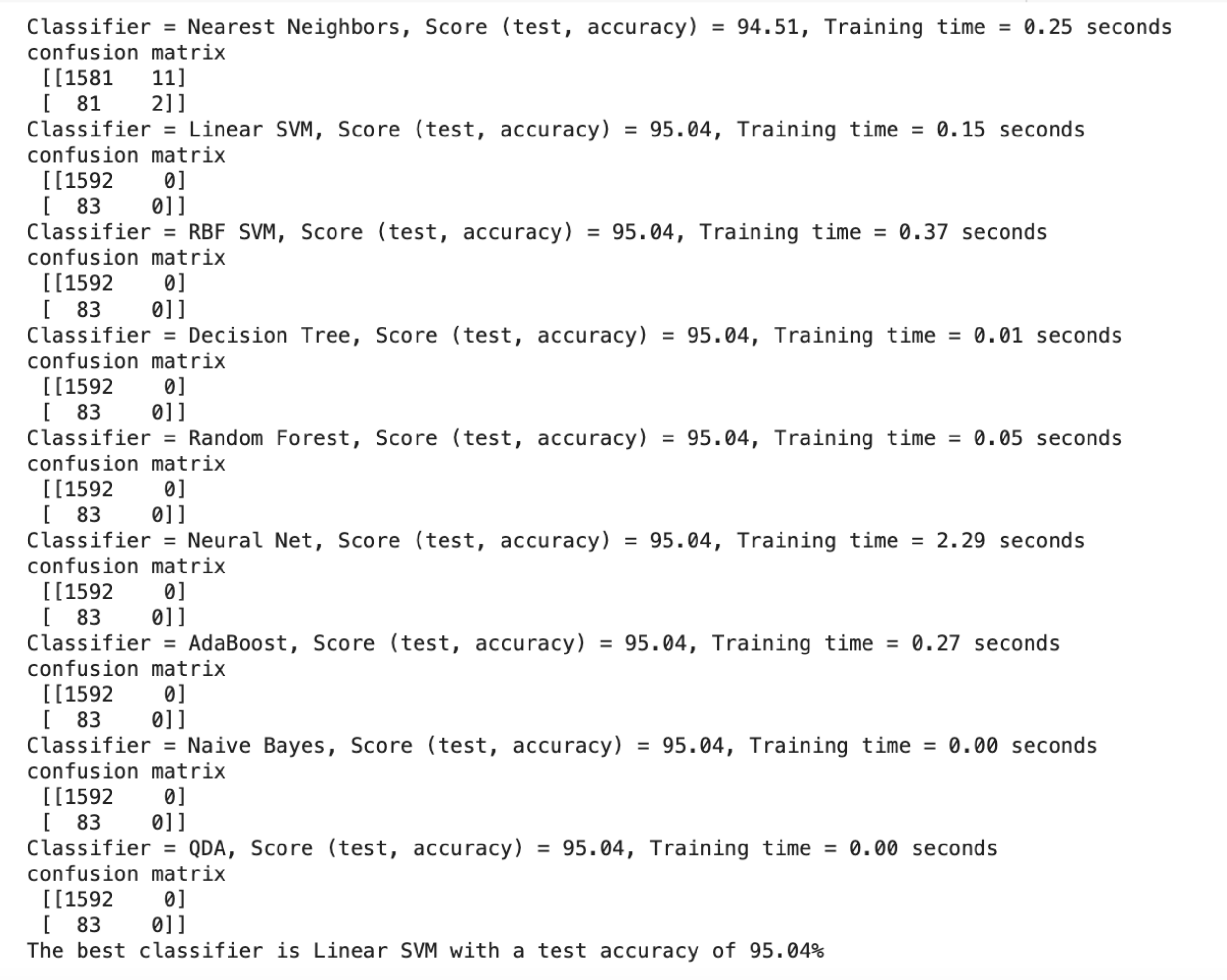
## Muller loop

### Muller Loop on Upsampled Data



```
Classifier = Nearest Neighbors, Score (test, accuracy) = 94.57, Training time = 0.18 seconds
confusion matrix
 [[1583    6]
 [  85    1]]
Classifier = Linear SVM, Score (test, accuracy) = 94.87, Training time = 0.17 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = RBF SVM, Score (test, accuracy) = 94.87, Training time = 0.49 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Decision Tree, Score (test, accuracy) = 94.87, Training time = 0.01 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Random Forest, Score (test, accuracy) = 94.87, Training time = 0.05 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Neural Net, Score (test, accuracy) = 94.87, Training time = 1.73 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = AdaBoost, Score (test, accuracy) = 94.87, Training time = 0.28 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Naive Bayes, Score (test, accuracy) = 94.87, Training time = 0.00 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = QDA, Score (test, accuracy) = 94.87, Training time = 0.00 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
The best classifier is Linear SVM with a test accuracy of 94.87%
```

### Muller Loop on Downsampled Data

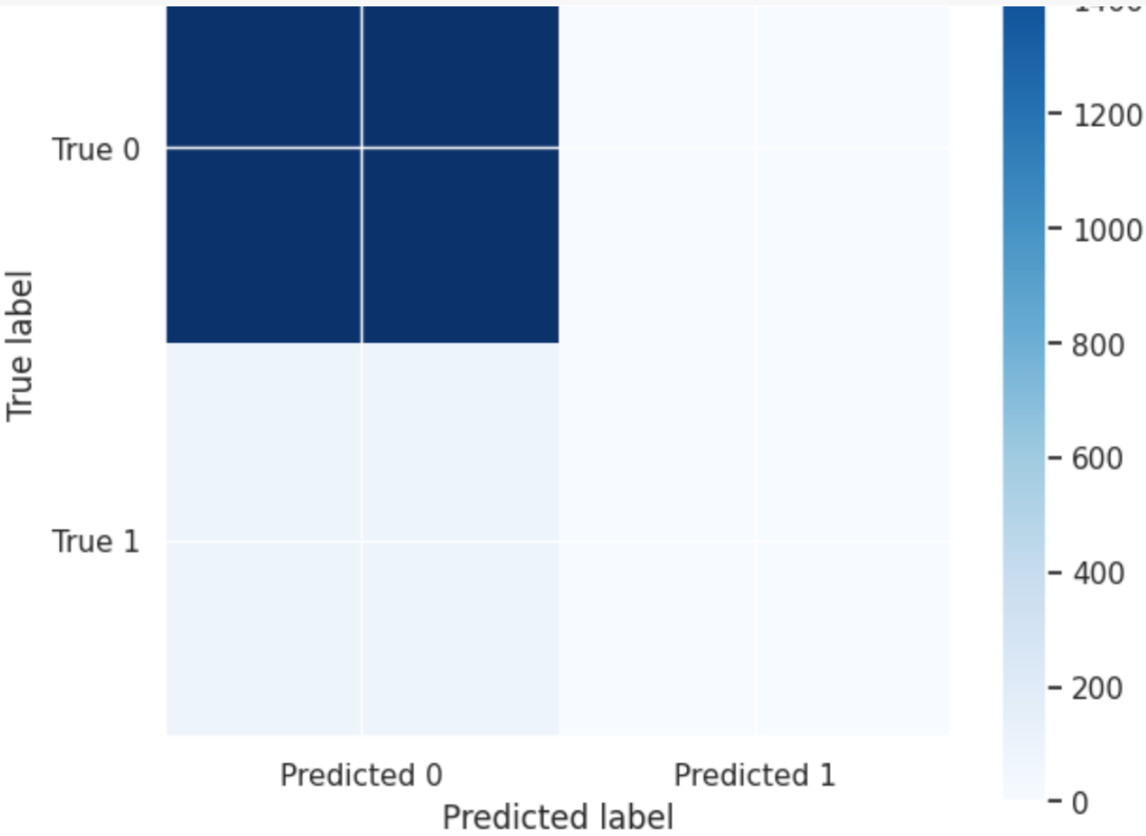


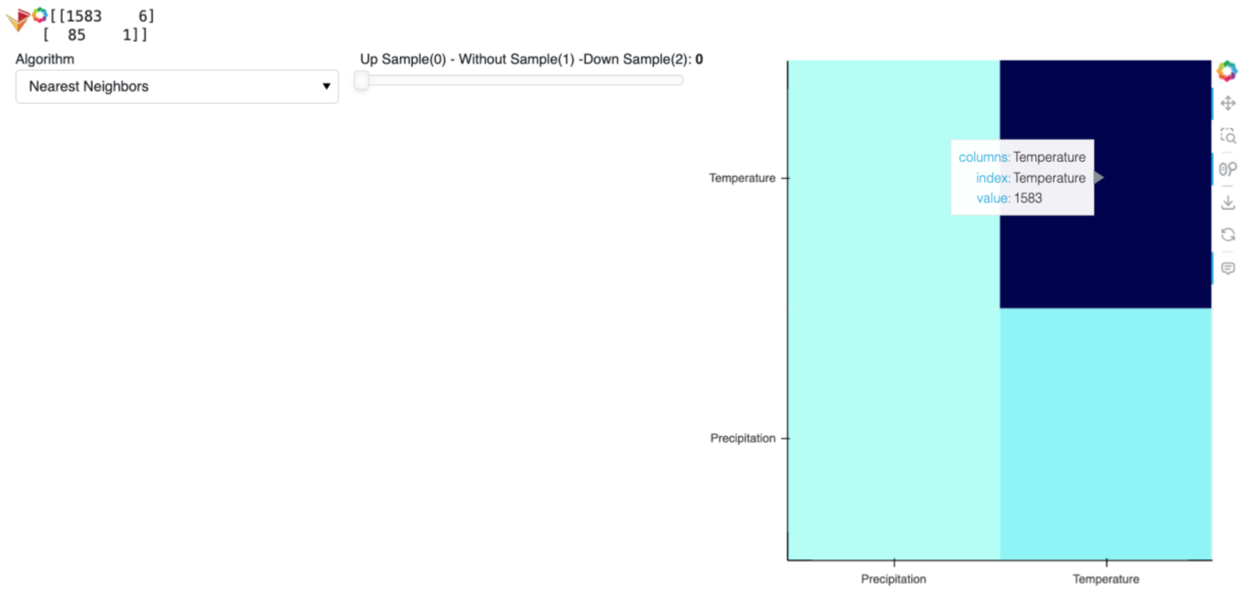
```
Classifier = Nearest Neighbors, Score (test, accuracy) = 94.51, Training time = 0.25 seconds
confusion matrix
 [[1581   11]
 [  81    2]]
Classifier = Linear SVM, Score (test, accuracy) = 95.04, Training time = 0.15 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = RBF SVM, Score (test, accuracy) = 95.04, Training time = 0.37 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Decision Tree, Score (test, accuracy) = 95.04, Training time = 0.01 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Random Forest, Score (test, accuracy) = 95.04, Training time = 0.05 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Neural Net, Score (test, accuracy) = 95.04, Training time = 2.29 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = AdaBoost, Score (test, accuracy) = 95.04, Training time = 0.27 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Naive Bayes, Score (test, accuracy) = 95.04, Training time = 0.00 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = QDA, Score (test, accuracy) = 95.04, Training time = 0.00 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
The best classifier is Linear SVM with a test accuracy of 95.04%
```

# Regression

## Muller loop

### Confusion matrix



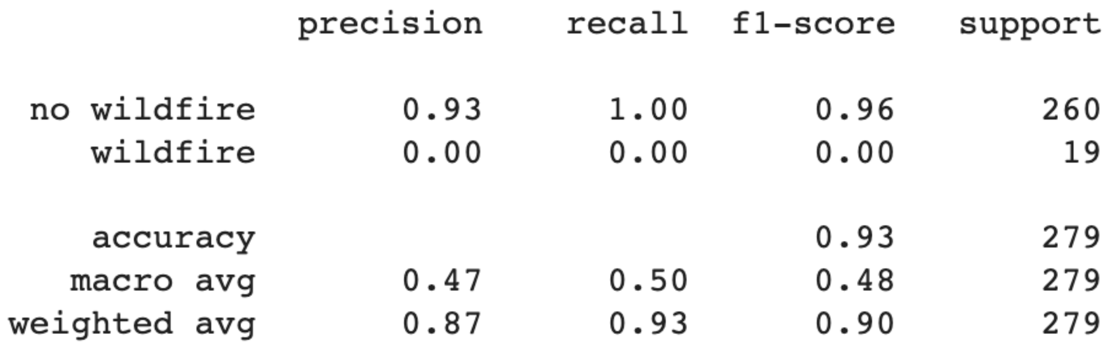


```
[[1583    6]
 [  85    1]]
```

Algorithm: Nearest Neighbors

Up Sample(0) - Without Sample(1) -Down Sample(2): **0**

## Calculate Metrics

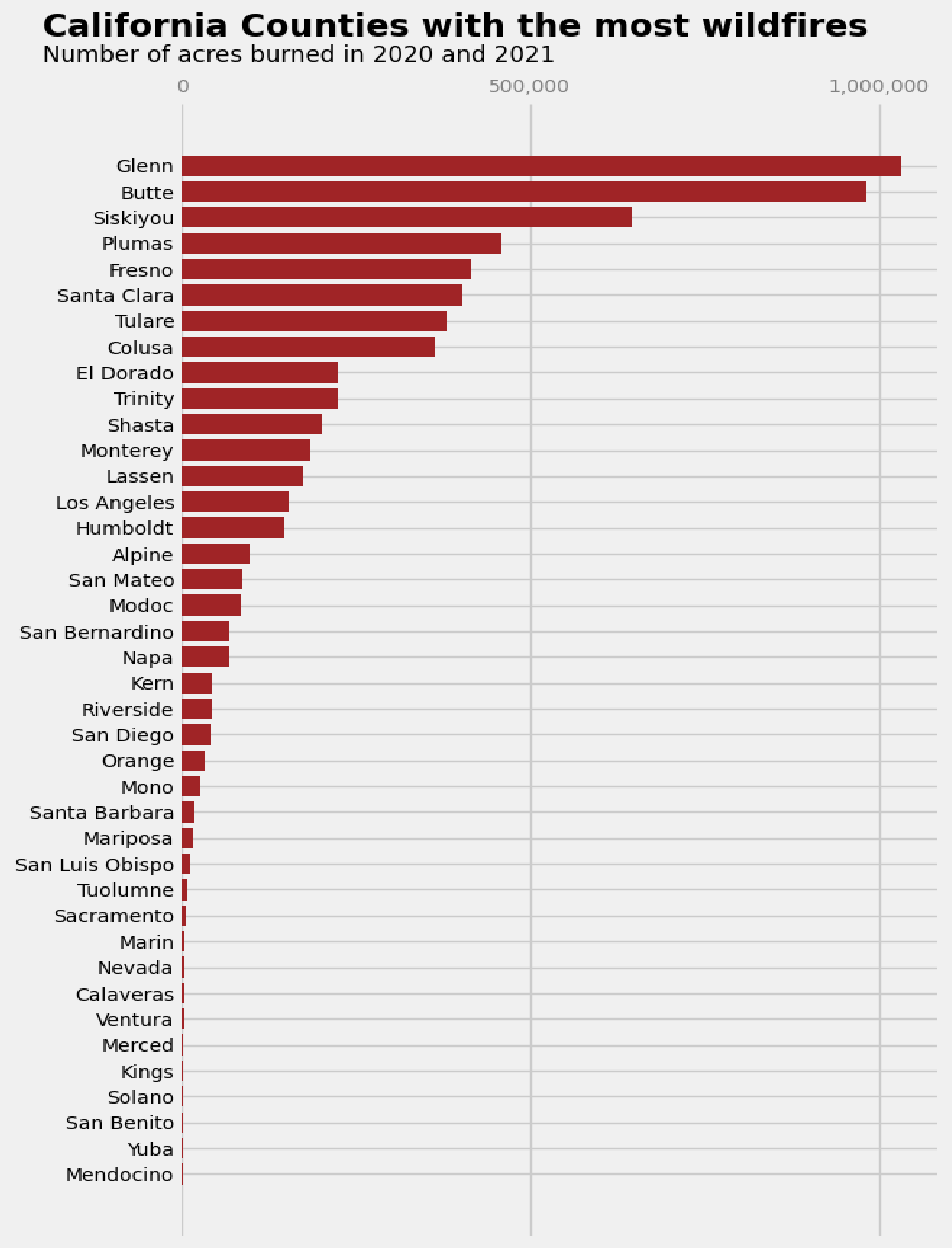


```
              precision    recall  f1-score   support

 no wildfire       0.93      1.00      0.96       260
    wildfire       0.00      0.00      0.00        19

    accuracy                           0.93       279
   macro avg       0.47      0.50      0.48       279
weighted avg       0.87      0.93      0.90       279
```

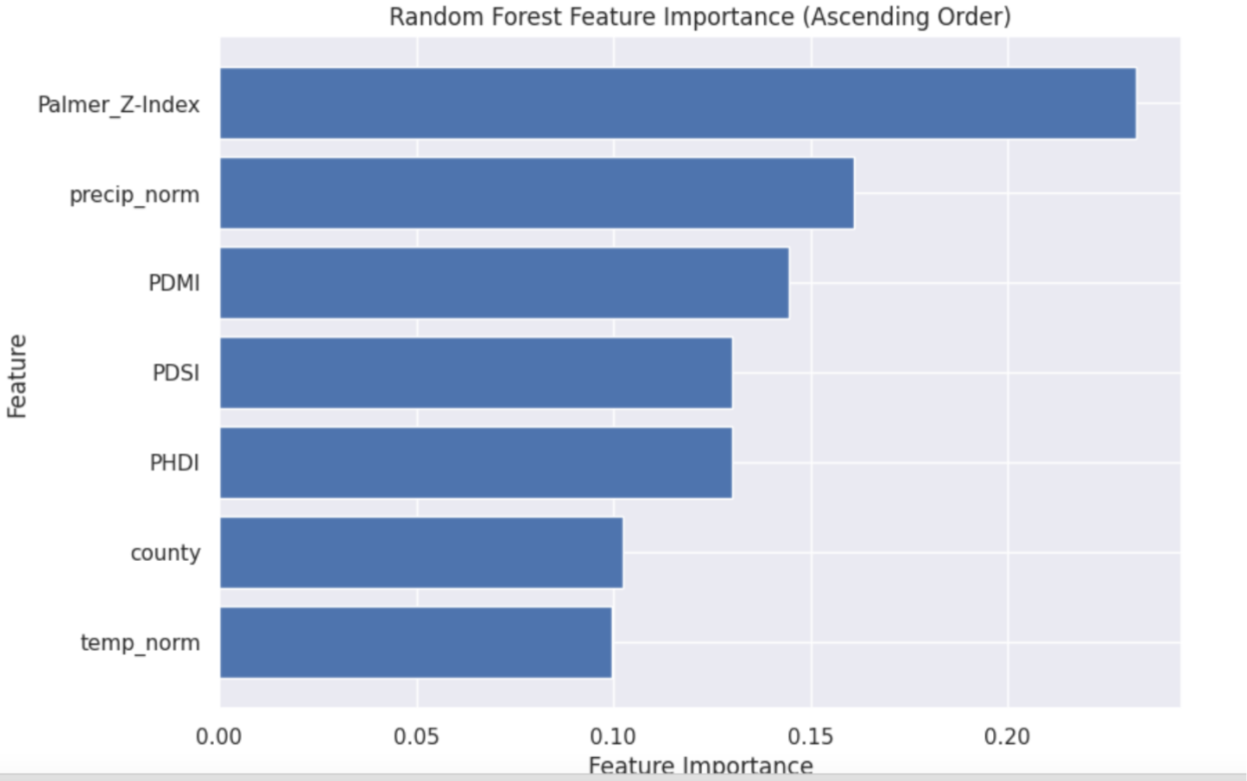
# Distributions of your Data



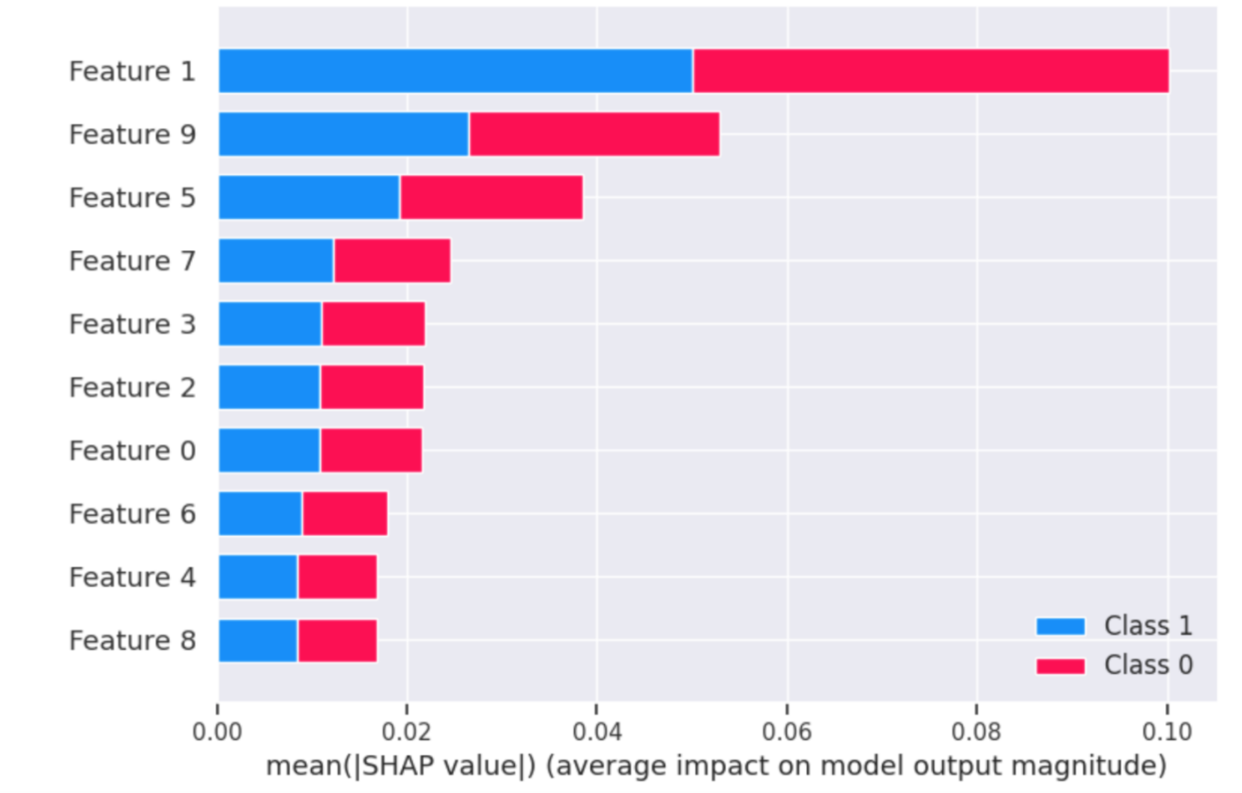
# Selected Features

Temperature and precipitation were selected as the top features.

- **Random Forest Feature Importance**



- **SHAP**



### Algorithms used to select features

We used the Random Forest Feature Importance and SHAP algorithm to select the best features for our use case.

## Changing Data Distributions

Oversampling our data gives us a better model

**What are the worst and best distributions of your datasets?**

Best: Oversampling

Worst: Undersampling

# Data Narrative

# Conclusions

[english language desc of the work done, conclusions]

Did you answer the questions that you set out to answer as part of your project description and set of experiments?

a) Which California counties are more prone to wildfires?

   The following counties are most prone to wildfires and have the most acres burnt.

   1) Amador County
   2) Butte County
   3) Lassen County
   4) Mono County
   5) Placer County
   6) Plumas County
   7) Sierra County

b) What properties should I invest in on the basis of how prone that region is to wildfires?

- Certainly, investing in properties in the above-mentioned counties would be riskier.

c) What is the relationship between temperature and precipitation levels that lead to wildfires?

- Higher temperatures and lower precipitation levels are more likely to cause wildfires.

# Predicting Wildfires in California Counties

Ujwala Mote
ujwalabalbhim.mote@sjsu.edu

Nihal Kaul
nihal.kaul@sjsu.edu

Suma Nagral
sumasunil.nagral@sjsu.edu

Chinmayi Sunku
chinmayi.sunku@sjsu.edu

## 1. **Abstract**:

In the backdrop of escalating wildfire occurrences in California, this research endeavors to harness machine learning methodologies to formulate a predictive apparatus capable of anticipating both the likelihood and magnitude of such wildfires. Through the meticulous integration of geographical, meteorological, and historical wildfire datasets, the study seeks to utilize intricate machine learning algorithms to decode the multifaceted dynamics underpinning wildfires. Once subjected to rigorous training and validation processes, the resultant model promises to be an indispensable asset in early detection, efficient emergency mobilization, and strategic wildfire countermeasures, thereby aiming to attenuate the ramifications of these calamities.

## 2. **Introduction**:

California's wildfire problem has reached alarming proportions in recent years, prompting an urgent need for innovative solutions. The state's diverse landscapes, combined with the challenges posed by climate change, have made wildfires a recurring and devastating threat. This machine learning project is a proactive response to this issue, aiming to harness the potential of predictive modeling.

Our endeavor revolves around the development of a predictive model capable of foreseeing the occurrence and severity of wildfires in California. To construct this model, we have curated a comprehensive dataset, aggregating a wealth of variables that capture the intricate factors contributing to wildfire incidents. These include geographic attributes, weather conditions, historical wildfire records, and human-related activities.

The overarching objective is to equip stakeholders, policymakers, and emergency responders with a powerful tool for forecasting wildfires. By examining various machine learning algorithms, we seek to identify the most effective means of making accurate predictions. The implications of a successful wildfire prediction model are vast, encompassing early warnings for vulnerable communities, improved resource allocation, and informed land management strategies.

In this report, we will delve into our approach to data collection, data preparation, feature engineering, model selection, and the evaluation of our predictive models. By sharing our insights, we aspire to contribute to the ongoing discourse on disaster management and environmental stewardship, ultimately working toward a safer and more resilient California.

## 3. **Methodology:**

In our methodological approach to understanding California wildfires, we curated and synthesized multiple datasets to offer a comprehensive perspective. The foundational dataset was sourced from Wikipedia, encompassing historical wildfire data from 2010-2022, with the BeautifulSoup Python library streamlining data extraction for the initial phase (2010-2015). The subsequent dataset, procured from the National Oceanic and Atmospheric Administration (NOAA), presented temperature anomalies across California counties, serving as a lens to discern potential climatic impacts on wildfire patterns. Additionally, NOAA's precipitation dataset was assimilated to gauge its influence on wildfire dynamics. To enhance the spatial granularity of our analysis, a dataset delineating all California counties was integrated. Preliminary to any in-depth analysis, rigorous data preprocessing was undertaken, encompassing categorical encoding and the introduction of a 'No Wildfire' category for model balance. The datasets were subsequently amalgamated, yielding a composite repository juxtaposing wildfire occurrences against meteorological determinants. This meticulous data assembly laid the groundwork for subsequent Exploratory Data Analysis (EDA), latent variable extraction, classification modeling, and deep learning explorations. The ensuing sections detail the systematic unfolding of these methodological phases, encompassing algorithm selection, model evaluation, and predictive analytics, culminating in actionable insights on regions with elevated wildfire susceptibility.

### 3.1 Data Preparation:

#### 3.1.1 Data Set 1 :

In the precipitation dataset, column nomenclature was standardized to ensure consistency. The datasets corresponding to temperature and precipitation were integrated based on the shared attributes, specifically 'county' and 'date', using an inner join approach. The datatype of the 'date' column was transitioned to datetime for enhanced data handling capabilities. Subsequently, the year was segregated from this column to support granular, year-centric analysis.

The dataset under consideration offers an in-depth historical perspective on wildfires in California across two delineated intervals: 2010-2015 and 2016-2022. For the interval from 2010 to 2015, data extraction was executed employing web scraping methodologies. The BeautifulSoup library was utilized to transform unstructured data from Wikipedia into structured datasets. The acquired data encompasses attributes such as fire nomenclature, causative factors, geographic coordinates, inception and containment dates, area impacted in terms of acres, and a count of structures compromised.

In the pursuit of discerning the interrelation between temperature and precipitation:

- The distribution of temperature data was subjected to normalization via z-score computation.
- Given that the precipitation data adhered to a power-law distribution, a logarithmic transformation was employed for its scaling.
- Visualization tools, specifically Seaborn, facilitated the depiction of the interplay between temperature and precipitation via scatter plots. Such plots elucidated patterns in temperature vis-à-vis precipitation over varying years.

In a subsequent phase, the dataset representing wildfire occurrences (DS1) was amalgamated with meteorological data (DS2 + DS3). This resultant dataset encapsulates details regarding land area impacted by wildfires in conjunction with temperature and precipitation metrics. Superfluous columns were pruned and discrepancies arising from missing values were systematically addressed.

Conclusively, data tables sourced annually were consolidated into a singular dataframe, thereby providing a holistic view of the data spanning the aforementioned intervals.

**3.1.2 Data Set 2:** To analyze the potential effects of climate on wildfire trends in California, it becomes imperative to understand the temperature anomalies and fluctuations in different counties over the years. This brought us to the NOAA – the National Oceanic and Atmospheric Administration. This dataset provides rich insights into the temperature variations across different California counties. With data points like anomalies, it opens the door for some deep dives – understanding if there's a link between the rising temperatures and the frequency or intensity of wildfires.

**3.1.3 Data Set 3:** A comprehensive analysis of wildfires in California requires understanding not just temperature, but also precipitation patterns. Rainfall can be a critical factor in controlling fire spread. With this understanding, we turned to our reliable source – the NOAA – for the precipitation dataset.

1. Exploratory Data Analysis and Visualization(EDAV) :

4.1 Data Distribution : Below graph shows the number of acres burnt in counties due to wildfire.

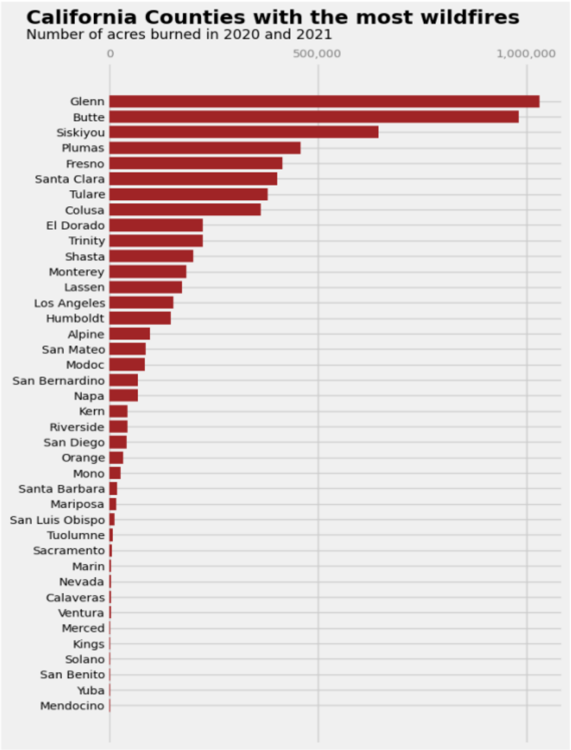


Fig 1: Number of acres burnt

4.2 . Feature importance :

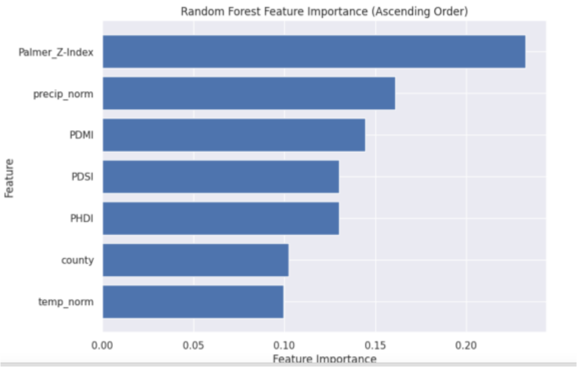


**Fig 2: feature importance**

The above graph shows the top 5 features of our dataset which include temperature, precipitation, county, Palmar_Z_index, and PDMI. We have focused mainly on two features from this i.e. temperature and precipitation for detecting the wildfire risk in a particular area.

**Gini Score:**

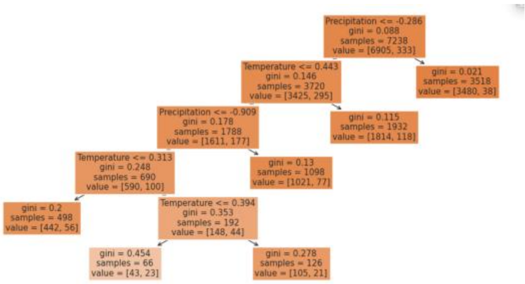


**Fig 3: gini score using decision tree**

Gini graphs show the importance of various features in predicting wildfires. It depicts the Gini impurity decreases as the decision tree incorporates different features like temperature, precipitation. Thus, temperature and precipitation are the two most important features for our model.

[feature transformation ;s

transform features, add new features to dataset via amalgamations (see below) , compare results with original

data distribution: plot and discuss

clean and normalize, use 2 of the 3 python libraries we discussed in class to analyze and visualize the data]

4.3. Relationship between Temperature and Precipitation

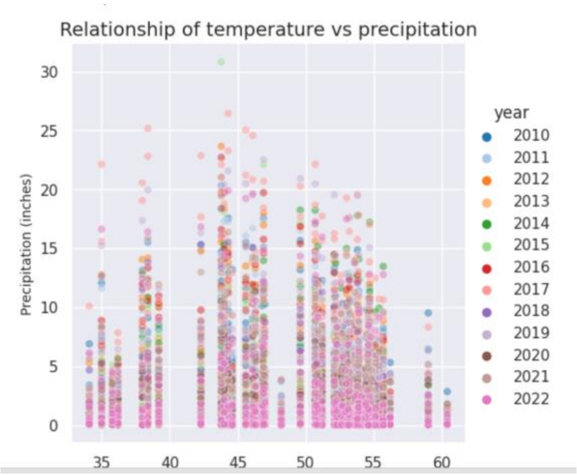


Fig 4: relationship between temperature and precipitation

From the graph it can be inferred that as the temperature increases precipitation decreases.

## 2. Clustering

Various clustering techniques were employed, including K-means, Gaussian Mixture Model (GMM), and Fractal clustering. These methods segmented counties based on features like temperature, rainfall, and wildfire occurrences.

### 5.1 K- Means Clustering

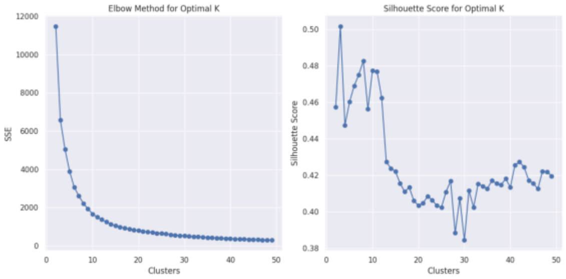


Fig 5: Elbow and Silhouette score

From the above SEE and Silhouette graph k = 8 seems like a good choice. There is a drastic change after k=12 in the Silhouette graph and an elbow at 8 in the elbow curve.

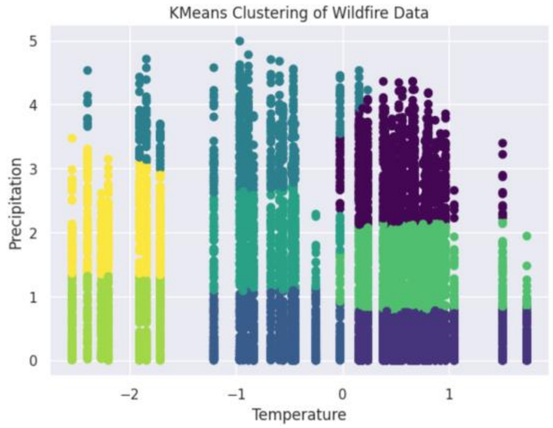


Fig 6: K-Means clustering

GMM clustering

It is a probability distribution that is symmetric around its mean. It is characterized by a specific probability density function that gives the likelihood of observing a particular value in a continuous dataset.

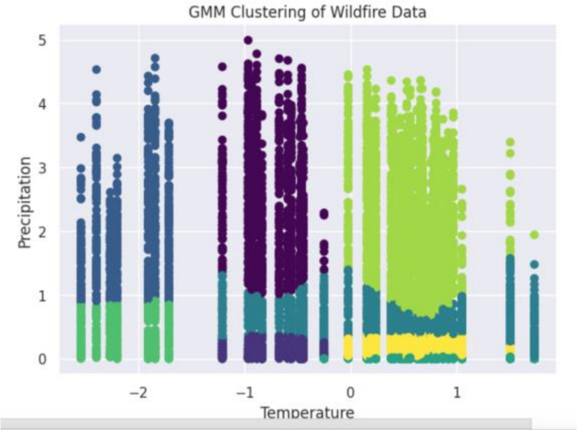


Fig 7: GMM clustering

If we observe both Kmeans and GMM cluster graphs look similar. K-means and Gaussian Mixture Model (GMM) clusters look similar because both algorithms are deterministic in nature and both the models were tested using the same dataset and hyperparameter

## 3. Fractal Clustering :

Fractal clustering is a data clustering technique that uses principles from fractal geometry to group data points into clusters.  Fractals are complex, self-similar geometric shapes that exhibit the property of self-similarity at different scales. In the context of clustering, this self-similarity property is leveraged to identify clusters at different levels of detail, from larger-scale clusters to smaller sub-clusters within them. Fractal clustering methods often involve recursive or hierarchical algorithms, where clusters can be divided into sub-clusters, and the process continues until a stopping criterion is met.

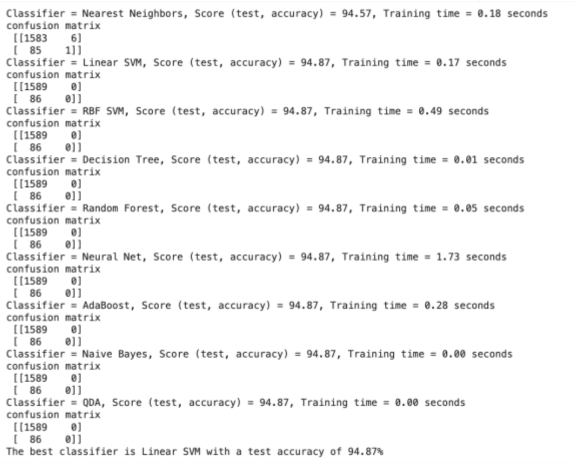
## Golden Cluster

Based on the above analysis we can conclude that cluster 0 is the 'Golden Cluster' since it has the most number of wildfires. We can utilize this information in real estate to identify the regions that are more prone to wildfires, this helps us to make an informed decision and assess the risks of properties realtors would like to invest in.

## 4. Amalgamations :

### 4.1 Muller loop

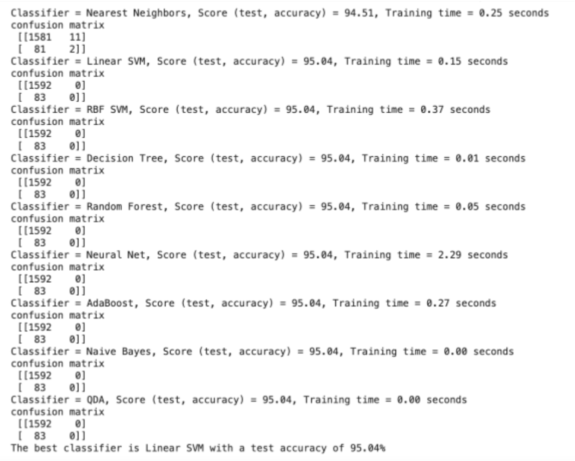
**Muller Loop on Upsampled Data**



```
Classifier = Nearest Neighbors, Score (test, accuracy) = 94.57, Training time = 0.18 seconds
confusion matrix
 [[1583    6]
 [  85    1]]
Classifier = Linear SVM, Score (test, accuracy) = 94.87, Training time = 0.17 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = RBF SVM, Score (test, accuracy) = 94.87, Training time = 0.49 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Decision Tree, Score (test, accuracy) = 94.87, Training time = 0.01 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Random Forest, Score (test, accuracy) = 94.87, Training time = 0.05 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Neural Net, Score (test, accuracy) = 94.87, Training time = 1.73 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = AdaBoost, Score (test, accuracy) = 94.87, Training time = 0.28 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = Naive Bayes, Score (test, accuracy) = 94.87, Training time = 0.00 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
Classifier = QDA, Score (test, accuracy) = 94.87, Training time = 0.00 seconds
confusion matrix
 [[1589    0]
 [  86    0]]
The best classifier is Linear SVM with a test accuracy of 94.87%
```

Fig 8: Upsampled data result

**Muller Loop on Downsampled Data**



```
Classifier = Nearest Neighbors, Score (test, accuracy) = 94.51, Training time = 0.25 seconds
confusion matrix
 [[1581   11]
 [  81    2]]
Classifier = Linear SVM, Score (test, accuracy) = 95.04, Training time = 0.15 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = RBF SVM, Score (test, accuracy) = 95.04, Training time = 0.37 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Decision Tree, Score (test, accuracy) = 95.04, Training time = 0.01 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Random Forest, Score (test, accuracy) = 95.04, Training time = 0.05 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Neural Net, Score (test, accuracy) = 95.04, Training time = 2.29 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = AdaBoost, Score (test, accuracy) = 95.04, Training time = 0.27 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = Naive Bayes, Score (test, accuracy) = 95.04, Training time = 0.00 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
Classifier = QDA, Score (test, accuracy) = 95.04, Training time = 0.00 seconds
confusion matrix
 [[1592    0]
 [  83    0]]
The best classifier is Linear SVM with a test accuracy of 95.04%
```

Fig 9: Down sampled data result

### 4.2 **Confusion matrix**

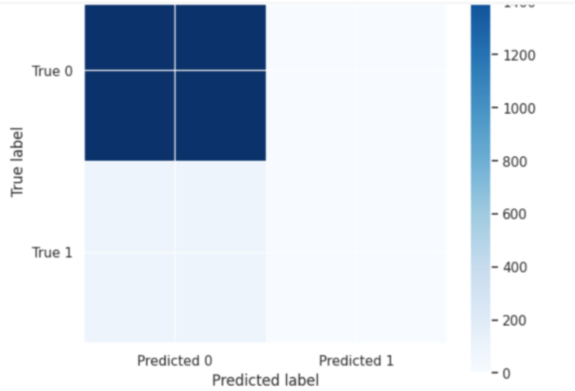
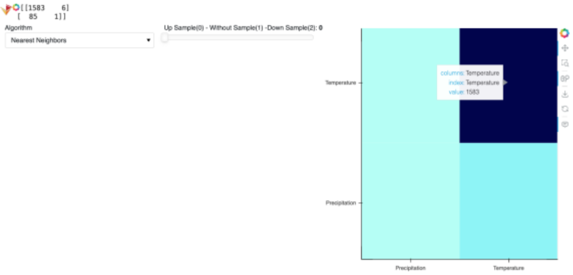


Fig 10: Confusion matrix

The displayed heatmap, representing a confusion matrix, reveals that the classifier demonstrates a commendable and balanced performance in predicting both positive and negative classes. The color intensity indicates a substantial number of correct predictions with minimal misclassifications. Specifically, both false positives and false negatives appear to be low, suggesting that the model is proficient and rarely errs in its predictions. While the classifier is evidently effective, depending on the specific application, there might be room for further optimization to reduce misclassifications even further.



```
[[1583    6]
 [  85    1]]
```

Algorithm: Nearest Neighbors

Up Sample(0) - Without Sample(1) -Down Sample(2): 0

Fig 11:  Confusion matrix heat map

### 4.3 Calculate Metrics

**Linear SVM classifier metrics:**

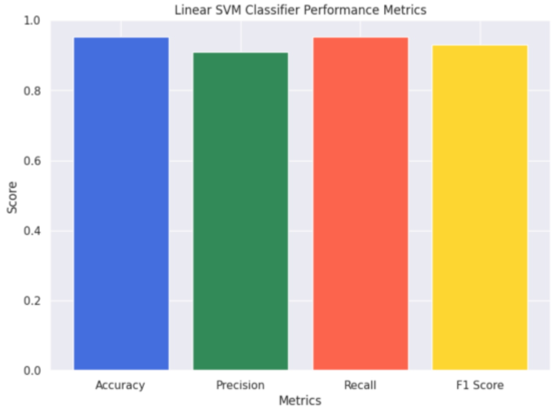


Fig 12: Linear SVM

### 4.4 ROC curve :

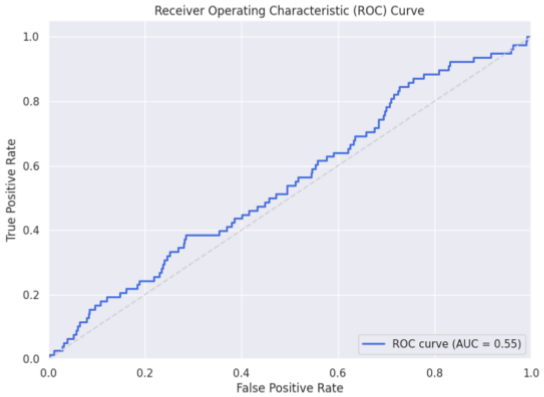


Fig 13: ROC

From the ROC curve, it's evident that the SVM classifier offers a promising approach to predict wildfires. The curve's trajectory significantly above the diagonal line indicates that the model possesses a good discriminative ability. The AUC value further quantifies this performance, suggesting that the model can effectively distinguish between situations leading to wildfires and those that don't, based on the selected features.

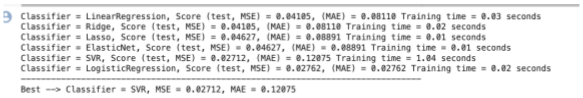
## 5. Latent Variables and Manifolds

Fire Suppression Capability: This latent variable could represent the effectiveness of fire suppression efforts in a region. It might include factors like the availability of firefighting resources, response times, and firefighting strategies.

Environmental Conditions: Latent variables related to weather and environmental conditions can capture factors like temperature, humidity, wind speed, and precipitation patterns, which play a crucial role in fire behavior.

- FDI: A composite measure that integrates weather-related variables (temperature, precipitation, drought indices like PDSI) to estimate the potential fire risk.
  - Manifest variables: temperature, precipitation and PDSI

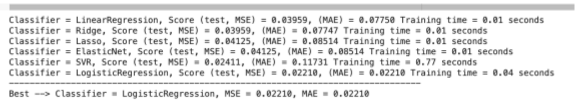
  Muller loop result:



```
Classifier = LinearRegression, Score (test, MSE) = 0.04105, (MAE) = 0.08110 Training time = 0.03 seconds
Classifier = Ridge, Score (test, MSE) = 0.04105, (MAE) = 0.08110 Training time = 0.02 seconds
Classifier = Lasso, Score (test, MSE) = 0.04627, (MAE) = 0.08891 Training time = 0.01 seconds
Classifier = ElasticNet, Score (test, MSE) = 0.04627, (MAE) = 0.08891 Training time = 0.01 seconds
Classifier = SVR, Score (test, MSE) = 0.02712, (MAE) = 0.12075 Training time = 1.04 seconds
Classifier = LogisticRegression, Score (test, MSE) = 0.02762, (MAE) = 0.02762 Training time = 0.02 seconds
--------------------------------------------------------------------------------
Best --> Classifier = SVR, MSE = 0.02712, MAE = 0.12075
```

Fig 14: Muller loop result

- Fuel Moisture: The moisture content of vegetation and fuel sources in a given area. It affects the ease with which fires can ignite and spread.
  - Manifest variables: PDSI

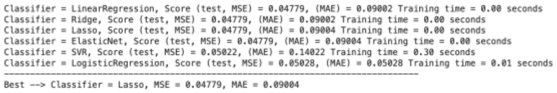


```
Classifier = LinearRegression, Score (test, MSE) = 0.03959, (MAE) = 0.07750 Training time = 0.01 seconds
Classifier = Ridge, Score (test, MSE) = 0.03959, (MAE) = 0.07747 Training time = 0.01 seconds
Classifier = Lasso, Score (test, MSE) = 0.04125, (MAE) = 0.08514 Training time = 0.01 seconds
Classifier = ElasticNet, Score (test, MSE) = 0.04125, (MAE) = 0.08514 Training time = 0.01 seconds
Classifier = SVR, Score (test, MSE) = 0.02411, (MAE) = 0.11731 Training time = 0.77 seconds
Classifier = LogisticRegression, Score (test, MSE) = 0.02210, (MAE) = 0.02210 Training time = 0.04 seconds
--------------------------------------------------------------------------------
Best --> Classifier = LogisticRegression, MSE = 0.02210, MAE = 0.02210
```

Fig 15: Muller loop result

- Weather Moisture: Temperature: The temperature at the time of the wildfire.Relative Humidity is referred to the humidity level in the area.

○ Manifest variables: temperature and precipitation



```
Classifier = LinearRegression, Score (test, MSE) = 0.04779, (MAE) = 0.09002 Training time = 0.00 seconds
Classifier = Ridge, Score (test, MSE) = 0.04779, (MAE) = 0.09002 Training time = 0.00 seconds
Classifier = Lasso, Score (test, MSE) = 0.04779, (MAE) = 0.09004 Training time = 0.00 seconds
Classifier = ElasticNet, Score (test, MSE) = 0.04779, (MAE) = 0.09004 Training time = 0.00 seconds
Classifier = SVR, Score (test, MSE) = 0.05022, (MAE) = 0.14022 Training time = 0.30 seconds
Classifier = LogisticRegression, Score (test, MSE) = 0.05028, (MAE) = 0.05028 Training time = 0.01 seconds
--------------------------------------------------------------------------------
Best --> Classifier = Lasso, MSE = 0.04779, MAE = 0.09004
```

Fig 16: Muller loop result

## 6.1. Heat Map :

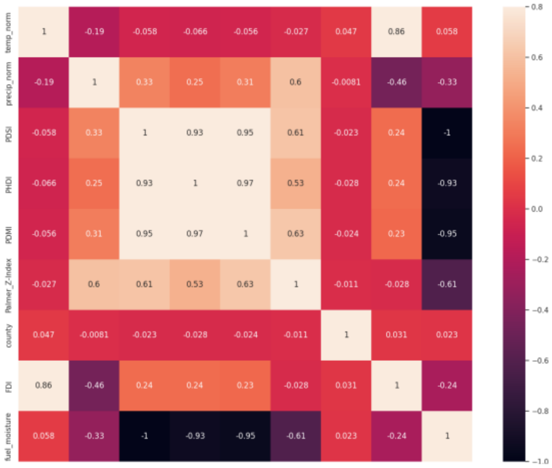


Fig 17: Heat map

## 6.2. Multimodel MLP :

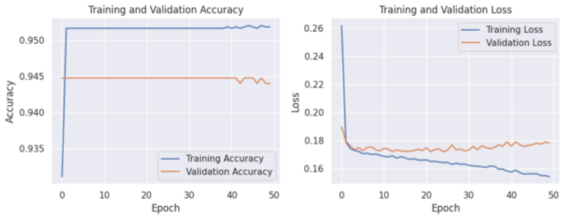


Fig 18: Multimodel MLP

The model is trained, and its performance is evaluated using the Keras Algorithm. The evaluation visualized is in the form of subplots for accuracy and loss during training. It will help us visualize how our model's performance changes over the training epochs.

The graph depicts the performance of a multi-layer perceptron neural network trained over 50 epochs for binary classification. Both training and validation accuracy start high and remain closely aligned, indicating the model's proficiency in capturing data patterns and its commendable generalization to unseen data. However, while the training loss consistently decreases, suggesting continuous learning, the validation loss reveals a different tale: after an initial descent, it fluctuates with a subtle uptrend in the latter epochs. This divergence hints at emerging overfitting, where the model, despite its adeptness with the training data, might be losing its generalization edge on new data. Given the model's structure and training approach, measures like dropout, regularization, or early stopping might be considered to counteract this overfitting trajectory.

## 6. **Conclusion**:

In our pursuit to forecast California wildfires, our project has embraced data-driven solutions and machine-learning methodologies. By leveraging a diverse dataset and implementing advanced models, we have taken a step towards early wildfire prediction. While our models demonstrate promise, the complexity of this task, coupled with the ever-evolving nature of wildfire dynamics, necessitates continued data refinement and model enhancement. The project's potential to enhance disaster preparedness and environmental preservation is undeniable, making it a crucial step in safeguarding California's communities and ecosystems from the growing threat of wildfires.

## 7. **References:**

1) K. Pham et al., "California Wildfire Prediction using Machine Learning," 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), Nassau, Bahamas, Dec - 2022.

2) A. Malik, N. Jalin, S. Rani, P. Singhal, S. Jain and J. Gao, "Wildfire Risk Prediction and Detection using Machine Learning in San Diego, California," 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), Atlanta, GA, USA, Oct - 2021.

3) T. Jiang, S. K. Bendre, H. Lyu and J. Luo, "From Static to Dynamic Prediction: Wildfire Risk Assessment Based on Multiple Environmental Factors," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, Dec - 2021.

4) Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.