Algerian Forest Fires Prediction

Algerian Forest Fires Dataset

Krishna Kamal Adidam (adidam@usc.edu), Ashwani Kumar Pradhan (ashwanik@usc.edu)

3rd May 2022

1. Abstract

In this project, we implement a supervised machine learning model to predict forest fires in the western Algerian regions. To do so, we perform feature expansion, feature scaling and appropriate feature selection. These appropriate features are used to train multiple models by varying it's hyperparameters. The best model is selected based on accuracy and mean F1 score and tested on test dataset to give our final model.

2. Introduction

2.1 Problem Statement and Goals

We are using the Algerian Forest fire dataset provided by Professor and TA team, after some preprocessing of the original data from UCI ML repository. This dataset includes of 244 instances (including training and test dataset) from two regions in northwest of Algeria. Our dataset consists of 184 instances as Train dataset and remaining 60 as test datasets.

Our goal is to develop a classification model based on the Supervised Machine Learning concepts and models covered in class, and a few from our additional knowledge. Here we are going to train multiple ML models after performing feature engineering on the dataset provided and then choose the best out of all as final model for the prediction of the forest fires where the output '1' implies fire is predicted and '0' otherwise.

2.2 Literature Review

EE-559 class notes, lectures and discussions have been the most helpful materials while working on this dataset and designing the model along with the documentation for Scikit learn , Pandas and Numpy libraries.

3 Approach and Implementation

Following is the Machine Learning pipeline used by us while working with the given dataset and designing the various Machine Learning models:

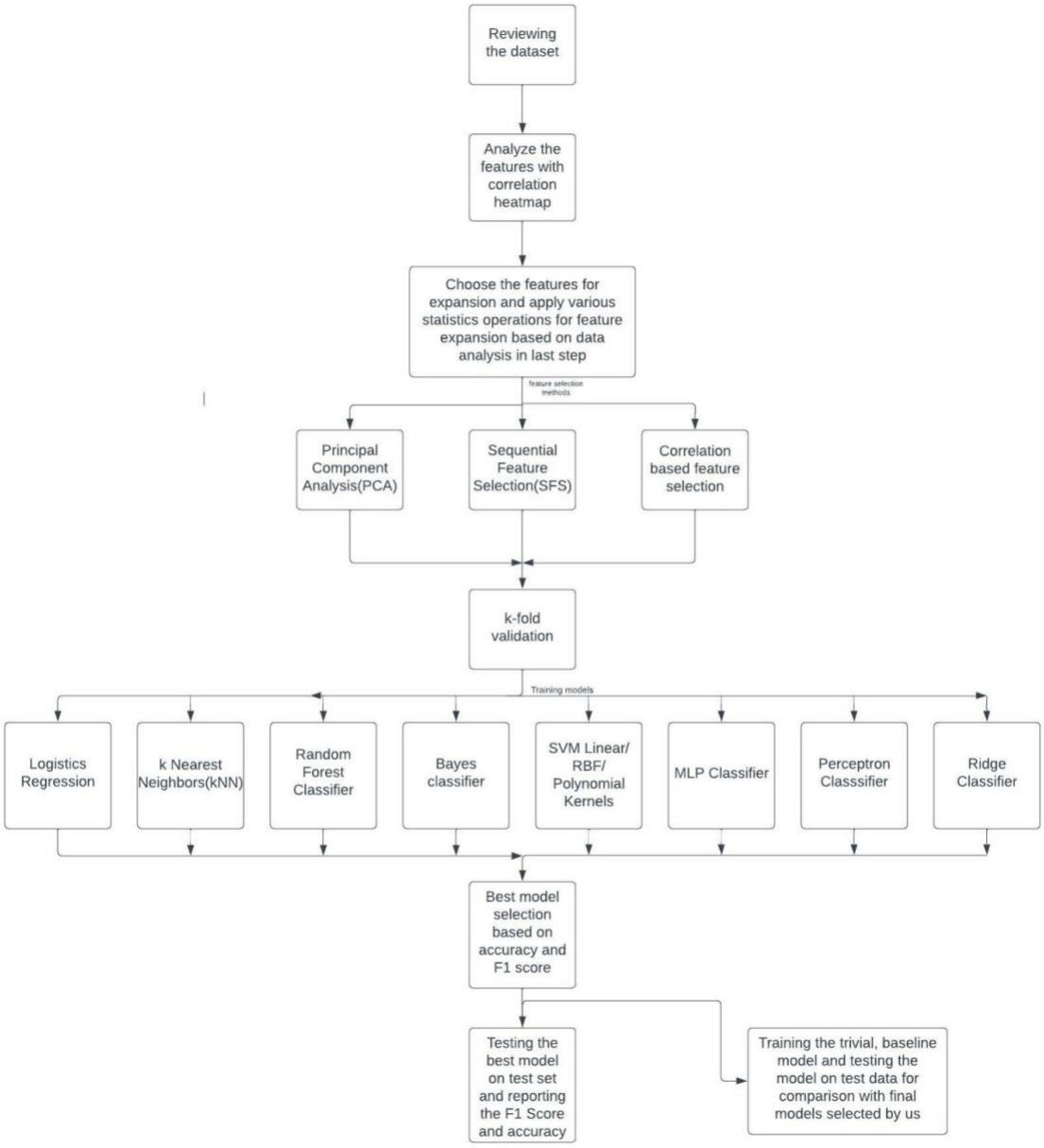


Fig 3.1: Machine Learning pipeline used for EE-559 Final Project

Following is the detailed walkthrough of our approach towards the solution for the given problem:

3.1 Dataset Usage

We have been given train and test dataset separately. So, we are concatenating the training and test datasets together (a total of 244 instances of datapoints) and performing the feature expansion on them. (Feature expansion process is explained in section 3c- Feature Engineering)
After the feature expansion is performed, we are again separating the data in two segments of training and test. After separation we have original number of rows, i.e., 184 instances of datapoints for training and 60 instances of datapoints for test set.
We are next performing data standardization on both train and test sets. Then we employed K-fold cross validation to perform model selection.
K-Fold Validation: One of the most time taking elements of this project has been the selection of Fold length (date ranges) for various k-fold validation sets, so that there is no influence of training data on validation set.
We have tried 3-fold, 5-fold, 6-fold and 10-fold validation, then trained and validated the model with the best available model and we finalized that the 5-fold validation provides us with the best accuracy and mean F1 score.

We are taking the train dataset which contains of 184 data points and dividing it into 5 folds for cross-validation (i.e., nearly 36 data points in each set). While feature expansion we have taken past 2- & 3-days average, 2- & 3- days minimum, and 2- & 3- days maximum of Temperature, Relative Humidity, Rain and Wind Speed as our new features, so to not get any influence from the last 3 days of training to validation, we have dropped 3 days i.e., 6 data points from the border dates for train and validation. Also, the last three days' datapoints are dropped from the total training data as we have done the data expansion for training and test data together, this way we make sure that there is no influence of training data on test data at all. (So, the final train date range we have is from 01-Jun-2012 to 28-Aug-2012)

The date range for train and validation data of all the five folds we chose are available below:

Validation Date Range	Train Date Range	
From 01-Jun-2012 to 15-Jun-2012	From 19-Jun-2012 to 31-Jul-2012	From 1-Aug-2012 to 28-Aug-2012
From 19-Jun-2012 to 03-Jul-2012	From 01-Jun-2012 to 15-Jun-2012	From 07-Jul-2012 to 28-Aug-2012
From 07-Jul-2012 to 21-Jul-2012	From 01-Jun-2012 to 03-Jul-2012	From 25-Jul-2012 to 28-Aug-2012

From 25-Jul-2012 to 08-Aug-2012	From 01-Jun-2012 to 21-Jul-2012	From 12-Aug-2012 to 28-Aug-2012
From 12-Aug-2012 to 28-Aug-2012	From 01-Jun-2012 to 31-Jul-2012	From 1-Aug-2012 to 08-Aug-2012

Here we have not used the k-fold validation function provided by sklearn, instead we coded the k-fold validation ourselves to train and validate our models.

Test set data usage: The test data is used only two times in this whole process of model design, one being the feature transformation and the other for the testing purpose of the final model.

3.2 Preprocessing

As the dataset was already a bit preprocessed, we did not have any missing values for any of the datapoints. So, we only needed to scale the data before applying feature dimensionality reduction techniques. We are using 'Standardization' technique to scale the data instead of 'Normalization' because normalization will get affected by the outliers. But the standardization method uses mean and standard deviation for scaling, so it is much less affected by outliers.

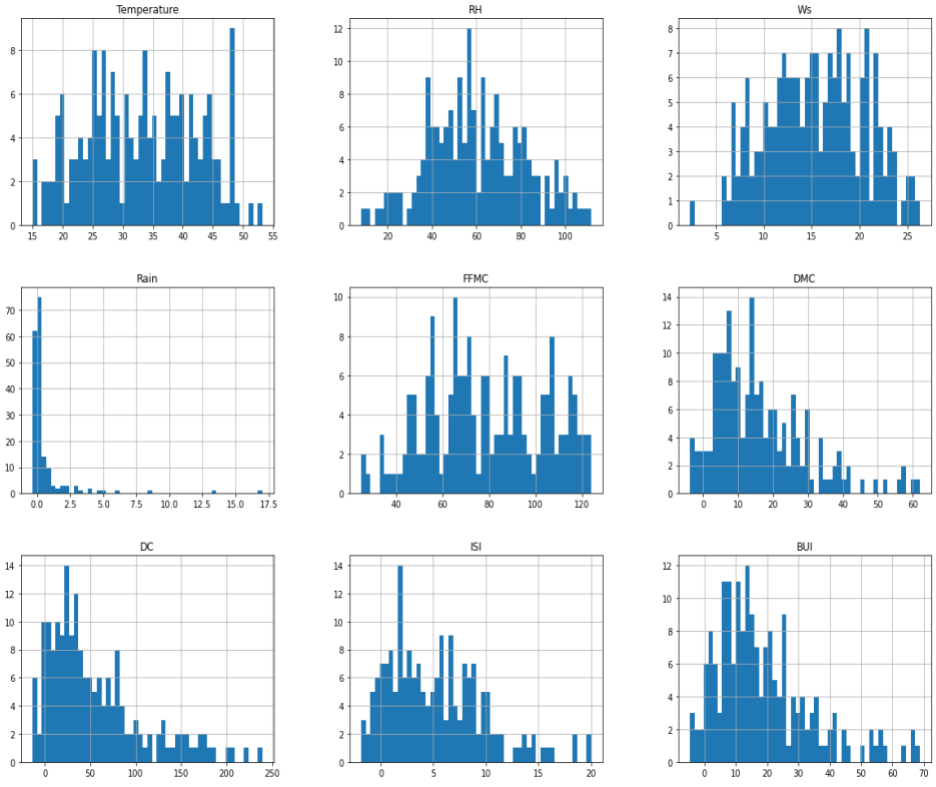


Fig 3.2: histogram plot of all features to check outliers for training data

3.3 Feature Engineering

To start with feature engineering, we checked the correlation between the original features with the help of the heatmap.

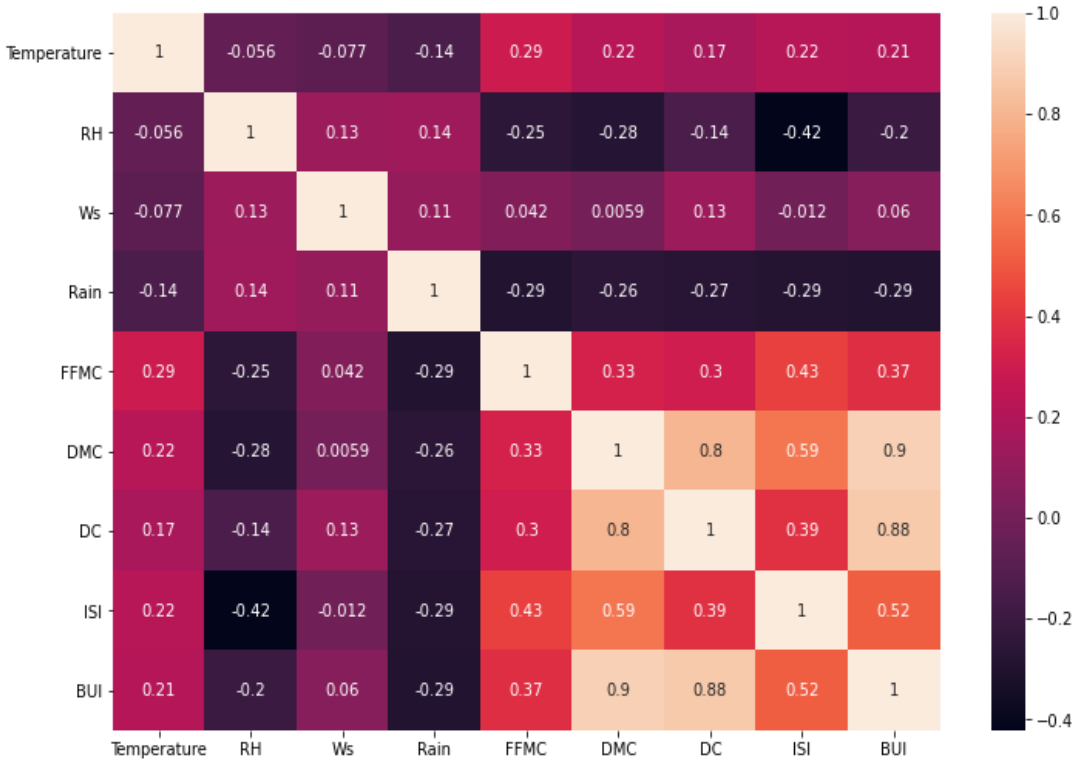


Fig 3.3: Visualization of correlation between the original features for training data

As shown above, we found that the features temperature, rain, RH and Ws seems uncorrelated with other features, so we added new features which reflect statistics of these features. These features names are as following:

Serial No.	Feature Name	Serial No.	Feature Name
i.	avg_temp_2	xiii.	avg_RH_2
ii.	max_temp_2	xiv.	max_RH_2
iii.	min_temp_2	xv.	min_RH_2
iv.	avg_temp_3	xvi.	avg_RH_3
v.	max_temp_3	xvii.	max_RH_3
vi.	min_temp_3	xviii.	min_RH_3

vii.	avg_rain_2	xix.	avg_Ws_2
viii.	max_rain_2	xx.	max_Ws_2
ix.	min_rain_2	xxi.	min_Ws_2
x.	avg_rain_3	xxii.	avg_Ws_3
xi.	max_rain_3	xxiii.	max_Ws_3
xii.	min_rain_3	xxiv.	min_Ws_3

To generate more features, we used the sklearn function for generating quadratic features. This generates a new feature matrix of all quadratic combinations of the original features. We used the quadratic feature expansion, to see if the combination of original features is going to influence the labels. We verify that through checking correlation coefficient of each compounded feature with labels.

We concatenate the features we got from polynomial feature expansion with the features we have extended using previous days statistics and overall, we have a total of 79 features excluding the 'Date' after feature expansion is completed.

The function for scaling the data in our program is written in the function implemented as: **def Scale_feature(data, data_columns,Scaletype)**

3.4 Feature dimensionality adjustment

After the feature expansion (using statistics and polynomial expansion), we have got 79 features.

Now for best feature selection, we are making use of three techniques:

i. Principal Component Analysis (PCA)

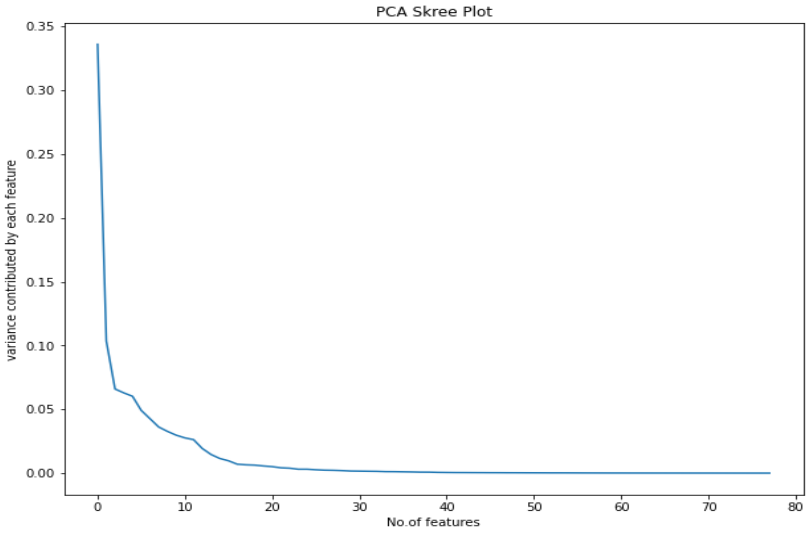
As PCA chooses a lower dimensional space where features are uncorrelated and dimensionality is reduced, it is less prone to overfitting as the degrees of freedom are reduced

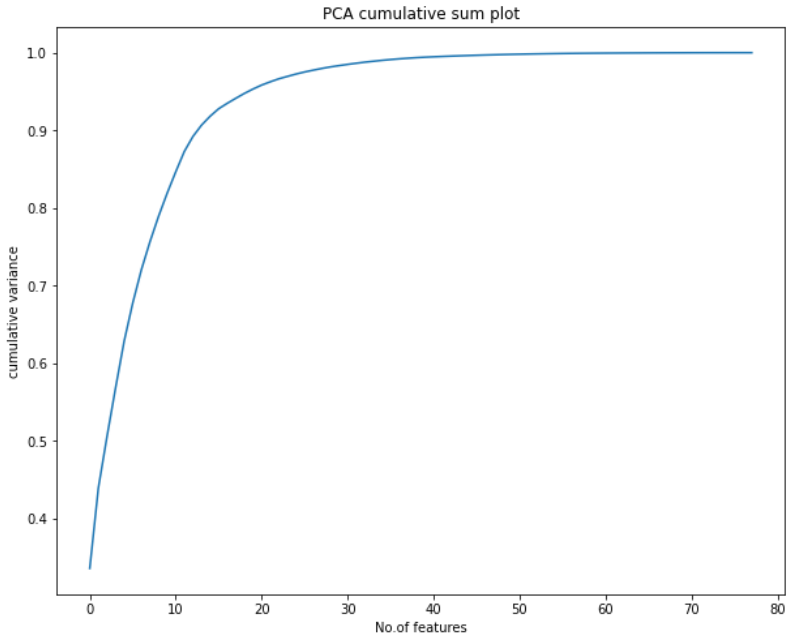
We used this to select a subset of features by transforming the expanded feature space into a lower dimensional space and choose the best number of features obtained, to do that, we ran all our Machine Learning models and chose the appropriate number of components which provide best accuracy and F1 score on the validation set. Here, while the models are training, one of the

parameters to be adjusted was the number of components chosen from PCA depending upon the variance contributed by each feature.

The PCA skree plot and the cumulative sum of variances plot for the expanded features are shown below.

As, we can see, the first feature itself retains 33 % variance, so to choose the best subset of features, we ran all our models with the number of components chosen by PCA varying from 6 to 21.





ii. Sequential Feature Selector (SFS)

As we know sequential feature selection greedily chooses the features by fitting the expanded feature space data on a simple model such as logistic regressor and gives out the best features.

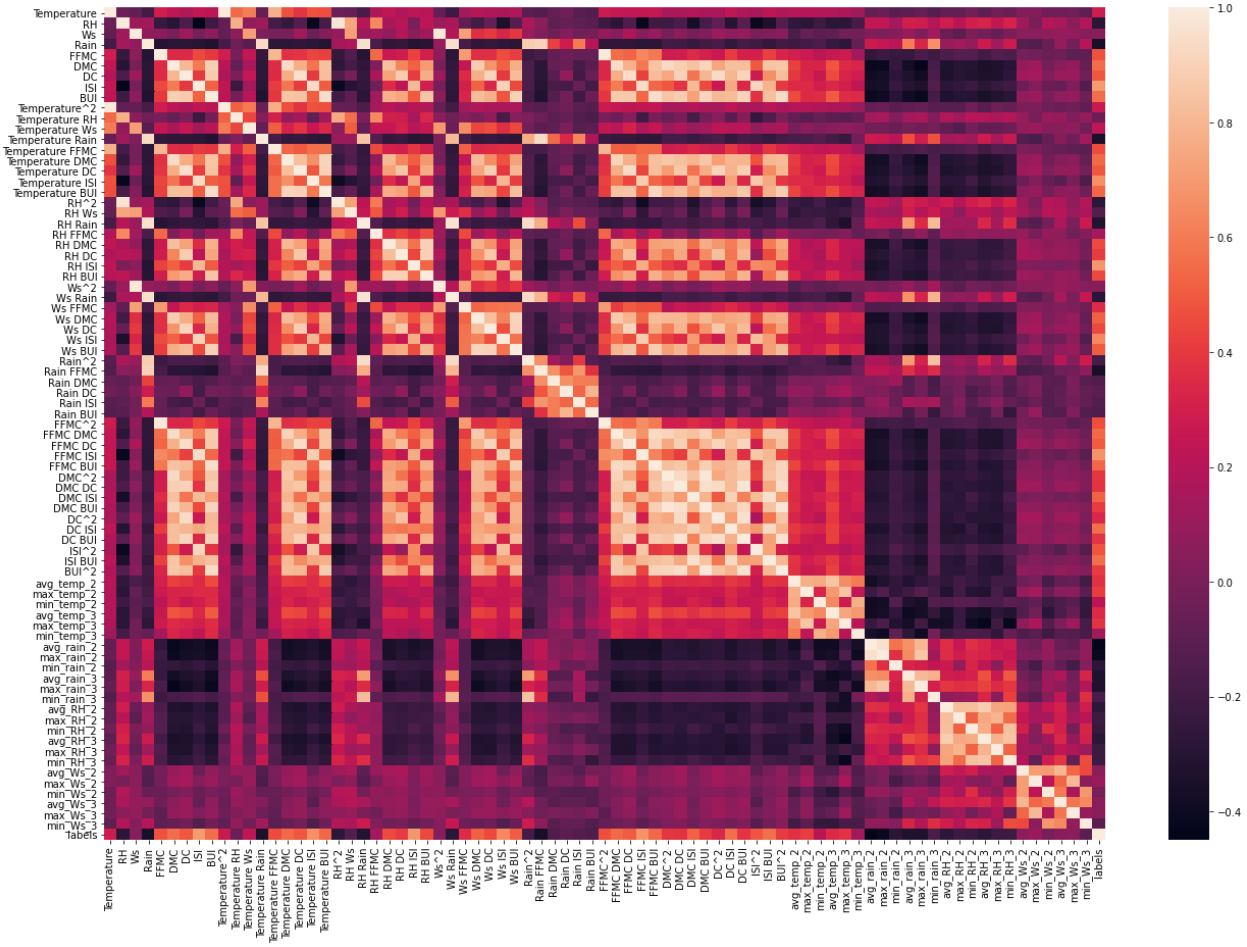
We have implemented the sequential feature selection on our expanded feature space on all the models to choose 15 features, which gives a better performance on validation set.

15 features are chosen to maintain the balance between number of constraints and degrees of freedom.

iii. Selecting Feature Subset Based on Features' Correlation with Labels-

A feature's correlation with label indicates how much it influences the output we are trying to predict.

Basing on this, from our expanded feature space we have selected the subset of features which have the highest correlation with the labels. We use these features to train all our models and obtain the best validation accuracy.



Following are the features selected by us after observing the above visualization:

'DMC', 'ISI', 'BUI', 'Temperature DMC', 'Temperature * ISI','Temperature * BUI', 'RH * ISI', 'Ws * DMC', 'Ws * ISI', 'Ws * BUI', 'FFMC * DMC', 'FFMC * DC', 'FFMC * ISI', 'FFMC * BUI', 'DMC * ISI', 'DC * ISI', 'ISI * BUI'.

As we can see, we have selected the original features plus the other features which are combinations of original features, which affect our output label.

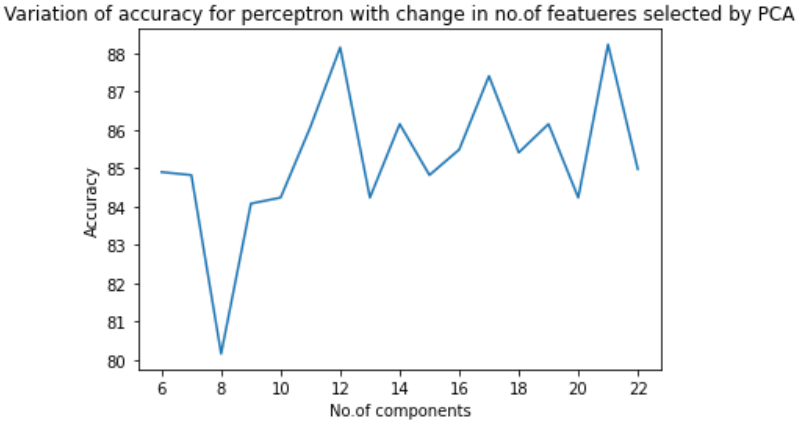
3.5 Training, Classification, and model selection:

We have used the following models:

i. **Linear Perceptron Classifier:**
As we know this is a supervised algorithm which will change the optimal weight vector repeatedly based on the number of misclassified datapoints until converged. The decision boundary is linear, and we

have used it for 2 class classification problem. We have used this method for:

- Features selected using PCA – This model is trained by varying the number of components selected by PCA, and accuracy and F1 score are observed at each iteration and finally the one with maximum accuracy is chosen.



The best validation accuracy is obtained at no. of components chosen by PCA=12 and corresponding accuracy is 88.23 %.

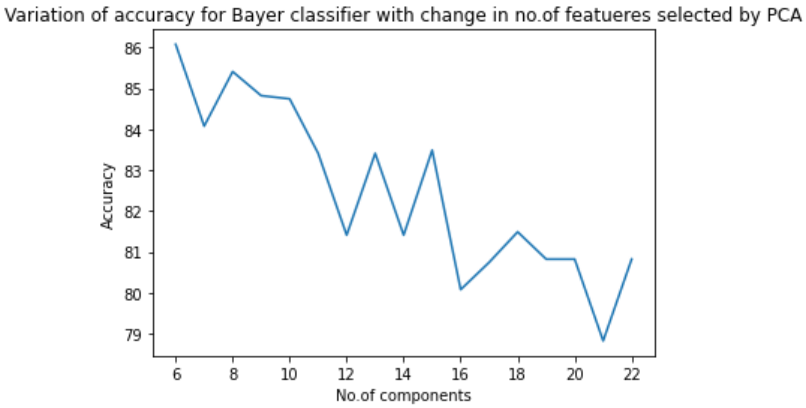
- Choosing the features which are the most correlated with the labels – Perceptron is trained on features selected. The validation accuracy in this case is: 83.49 %.

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained these on perceptron.
  The validation accuracy in this case is: 82.23 %.

ii. Bayes Minimum Error Classifier:

This classifier works under the naïve assumption that there is independence between features. This assumes that the probability density functions of features are distributed according to Gaussian distribution. We have used this method for:

- Features selected using PCA - This model is trained by varying the number of components selected by PCA, and accuracy and F1 score are observed at each iteration and finally the one with maximum accuracy is chosen.



The best validation accuracy is obtained at no .of components chosen by PCA=12 and corresponding accuracy is 86.07 %.

- Choosing the features which are the most correlated with the labels – Bayes minimum classifier is trained on features selected.
  The validation accuracy in this case is: 91.41 %.

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained these on Naïve Bayes classifier.
  The validation accuracy in this case is: 88.74 %.

iii. K Nearest Neighbors:

This model assigns a class to the input based on the majority vote of its neighboring datapoints' corresponding class labels. We have implemented this model by changing the number of neighbors, which is the hyperparameter for this model.
The number of neighbors changes from 3 to 19.

- Features selected using PCA - This model is trained by varying the number of components selected by PCA, and accuracy and F1 score are observed at each iteration and finally the one with maximum accuracy is chosen.
  The best validation accuracy is obtained for number of neighbors= 3, at no. of components chosen by PCA=12 and corresponding accuracy is 87.411 %.

- Choosing the features which are the most correlated with the labels – KNN is trained on features selected by varying the number of neighbors.
  The best validation accuracy in this case is obtained for number of neighbors = 15 and the corresponding accuracy is: 92.82 %.

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained this KNN classifier by varying the number of neighbors.
  The best validation accuracy in this case is obtained for number of neighbors = 12 and the corresponding accuracy is: 91.56 %.

iv. Logistic Regression:

Logistic regression is a model which uses sigmoid function as its hypothesis rather than affine hypothesis. It models the probability of a class and gives a hard output based on thresholding. The hypothesis function is chosen from Maximum Likelihood Estimation.

We have changed the regularization constant 'C' as our hyper parameter and trained the models. The 'C' is chosen from {0.05, 0.1, 0.7 ,1, 5 ,10}

- Features selected using PCA - This model is trained by varying the number of components selected by PCA
  The best validation accuracy is obtained for 'C' = 0.05, at no. of components chosen by PCA=6 and corresponding accuracy is 90.078 %.

- Choosing the features which are the most correlated with the labels – Logistic regression is trained on features selected by varying the regularization constant.
  The best validation accuracy in this case is obtained for 'C' = 1 and the corresponding accuracy is: 91.41 %.

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained this classifier by varying 'C'.
  The best validation accuracy in this case is obtained for 'C' = 1 and the corresponding accuracy is: 90.82 %.

v. Random Forests Classifier:

Random forest is ensemble method which trains many decision trees. A decision tree makes a tree like decisions and gives the probabilities. This is prone to overfitting; hence we use random forests.

- Features selected using PCA - This model is trained by varying the number of components selected by PCA.
  The best validation accuracy is obtained at no. of components chosen by PCA=17 and corresponding accuracy is 91.411 %.

- Choosing the features which are the most correlated with the labels – Random Forest classifier is trained on features selected.
  The best validation accuracy in this case is obtained as: 91.49 %.

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained this classifier.
  The best validation accuracy in this case is obtained as: 88.90 %.

vi. Ridge Classifier:

Ridge Classifier is a classifier which uses ridge regression taught in EE-559. This classifier converts the target values into {-1,1} and performs a regression task.

- Features selected using PCA - This model is trained by varying the number of components selected by PCA.
  The best validation accuracy is obtained at no. of components chosen by PCA=6 and corresponding accuracy is 88.82 %.

- Choosing the features which are the most correlated with the labels – Ridge classifier is trained on features selected.
  The best validation accuracy in this case is obtained as: 91.49 %.

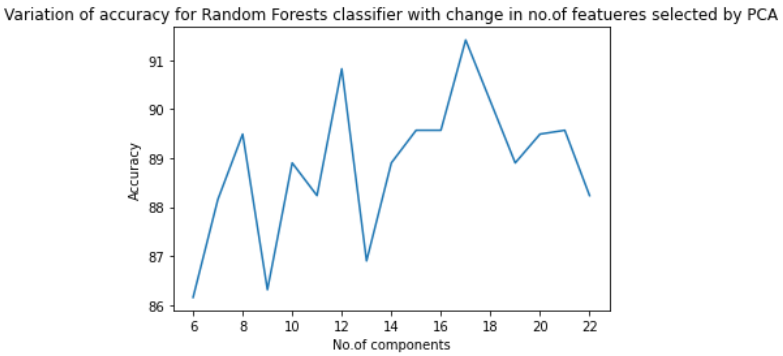
- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained this ridge classifier.
  The best validation accuracy in this case is obtained as: 88.47 %.

vii. Artificial Neural Networks (ANN):

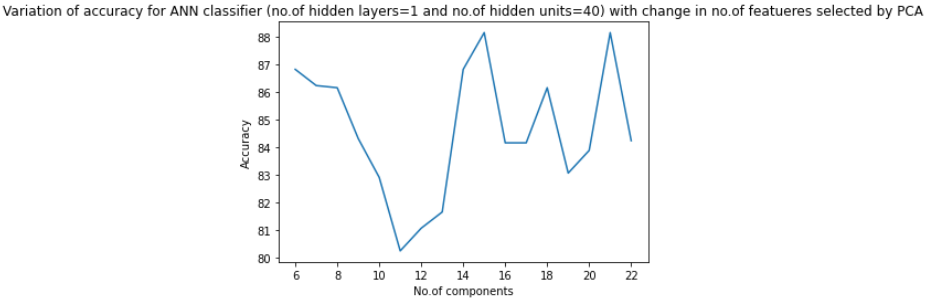
ANN helps in giving us a non-linear decision boundary and the hidden layers allow the model to learn new features. Here we implemented three architectures of ANN with ReLU as our hidden layer activation function.

- Features selected using PCA - This model is trained by varying the number of components selected by PCA.

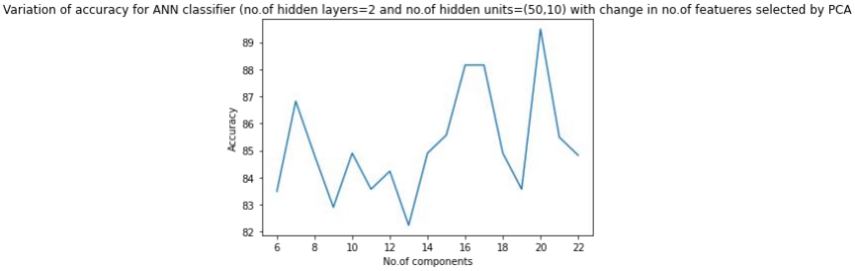
- Architecture 1: One hidden layer with number of hidden units = 40



- Architecture 2: Two hidden layers with number of hidden units = (50, 10)



- Architecture 3: Two hidden layers with number of hidden units = (35, 15)



The best validation accuracy is given for 9 number of components with one hidden layer and the accuracy is: 88.82 %

- Choosing the features which are the most correlated with the labels – ANN is trained for three different architectures (same architectures as shown in PCA case).

The best validation accuracy in this case is obtained as: 91.49 % with one hidden layer.

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, ANN is trained for three different architectures (same architectures as shown in PCA case).
  The best validation accuracy in this case is obtained as: 90.23 % with 2 hidden layers.

viii. Support Vector Machines (SVM):

SVMs are maximum margin classifiers, they use kernel trick to transform the space given the dual representation of input. This is less prone to overfitting as we are optimizing the cost function with respect to a constraint that maximum margin should be ensured while as many datapoints are classified correctly.

There are various kernels. We used linear, radial basis function and polynomial kernel by varying regularization parameter 'C' and kernel coefficient(gamma) on all the data obtained from various feature selection techniques.

The 'C' is chosen from {0.0001, 0.001, 0.1, 1, 10, 100, 500}

The gamma is chosen from {0.0001, 0.0005, 0.001, 0.01, 0.1, 1}

- Features selected using PCA - This model is trained by varying the number of components selected by PCA.
  The hyperparameters are 'C', gamma and kernel used.
  The best validation accuracy is obtained using linear kernel with 'C' as 0.01 and gamma as 0.0001 at no. of components chosen by PCA=15 and corresponding accuracy is 92.745 % with mean F1 score as 0.89

- Choosing the features which are the most correlated with the labels – The hyperparameters are 'C', gamma and kernel used.
  The best validation accuracy is obtained using linear kernel with 'C' as 0.01 and gamma as 0.0001 and corresponding accuracy is 92.8235 % with mean F1 score as 0.8854

- Features selected using Sequential Feature Selection- After obtaining 15 features from SFS, we trained SVMS.
  The hyperparameters are 'C', gamma and kernel used.
  The best validation accuracy is obtained using linear kernel with 'C' as 0.01 and gamma as 1, corresponding accuracy is 92.82 % with mean F1 score as 0.8845.

4 Results and Analysis: Comparison and Interpretation

Following is the cross-validation performance metrics results:

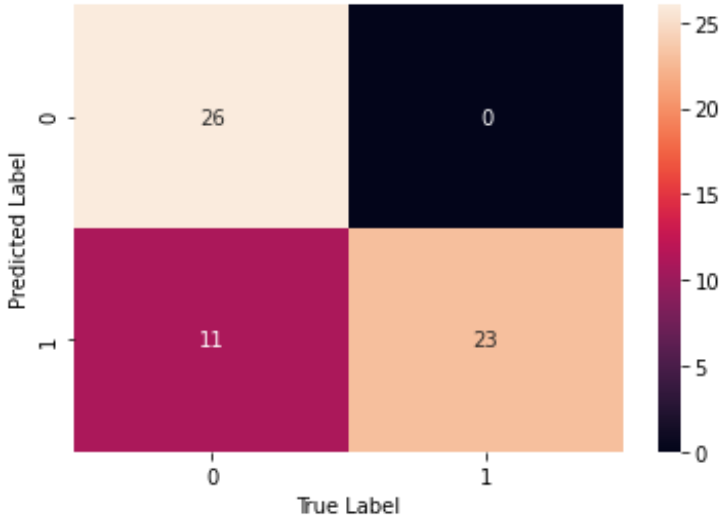
Model	PCA			SFS		Features selected from maximum correlation with labels	
	Accuracy (%)	Mean F1 Score	No. of Features selected	Accuracy	Mean F1 Score	Accuracy	Mean F1 Score
Perceptron Classifier	88.2352	0.8283	21	82.23	0.754	83.49	0.749
Logistic Regression	90.07	0.863	6	90.82	0.863	91.411	0.8816
k Nearest Neighbors (kNN)	87.41	0.8632	7	91.56	0.864	92.82	0.8781
Random Forest Classifier	91.411	0.874	17	88.9	0.83	91.49	0.87
Bayes Classifier	86.078	0.825	6	88.74	0.84	91.411	0.8788
SVM Linear Classifier	92.74	0.893	15	92.823	0.884	92.82	0.885
MLP Classifier (ANN)	89.49	0.847	20	91.49	0.87	92.15	0.876
Ridge Classifier	88.8	0.839	6	88.47	0.845	91.49	0.871

Finally, the best performing model from model selection is SVM with linear kernel with regularization parameter ‘C’ as 0.01, kernel coefficient(gamma) as

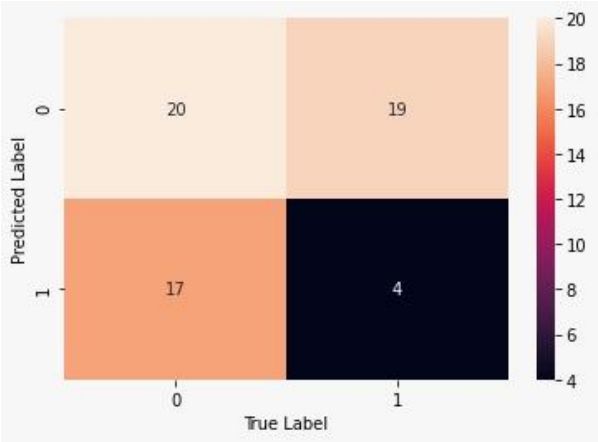
0.0001 for the data obtained from features selected by choosing the mostly correlated ones with the labels.

4.1 Final Test Set Results:

- On the best performing model:
  - The accuracy is 81.67 %.
  - The F1 scores for each label [0.82539 0.8070]
  - The mean F1 score is 0.8162
  - The confusion matrix is:

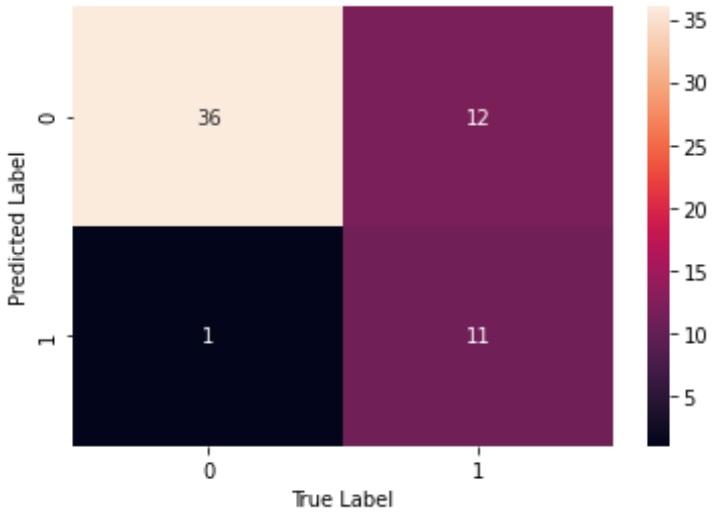


- On the trivial system
  - The accuracy is 40%
  - The F1 score for each label [0.201 0.163]
  - The mean F1 score is 0.1818
  - The confusion matrix is



- On the baseline system (Nearest Means Classifier)

- The accuracy is 51.66
- The F1 score for each label [0.8470 0.6285]
- The mean F1 score is 0.6285
- The confusion matrix is



5 Libraries and Self-implemented Functions:

Libraries Used: Following is the list of libraries used by us:

a. NumPy Library
b. Pandas Library for data analysis from csv file
c. Matplotlib library for visualization
d. Random module from python
e. Seaborn library for data visualization (using heatmap)
f. Datetime module form python for working with various segmentation of dates
g. Sklearn.preprocessing package for standardization of features and polynomial feature expansion
h. Sklearn.decomposition package for Principal Component Analysis (PCA)
i. Sklearn.feature_selection package for Sequential Feature Selector (SFS)
j. sklearn.linear_model package for Logistic Regression, Perceptron and Ridge Classifier models
k. sklearn.metrics package for accuracy score, f1_score calculations and confusion matrix

l. sklearn.svm package for Support Vector Classifier (SVC) model
m. sklearn.neighbors package for K-Neighbors Classifier model
n. sklearn.naive_bayes package for Bayes minimum error classification
o. sklearn.ensemble package for Random Forest Classifier model
p. sklearn.neural_network package for MLP Classifier

Self-Implemented Functions: Following is the information on the functions implemented by us:

i. **read_data(data):** Function to read the dataframe and split the data and its labels. This function takes the file name as input argument.
ii. **add_features(data,feature_num,column_name_1,column_name_2,column_name_3,n):** Function to create new features, which are average, min and max for last 'n' number of days.
iii. **add_statistic_features_to_data(data):** Function to add statistics of previous days to the given data with the help of add_features function.
iv. **Scale_feature(data, data_columns,Scaletype):** Function to scale the data. We use standardization function from sklearn for scaling the data of all columns except the Date column.
v. **cv_split(data,start_date_train_1,end_date_train_1,start_date_val, end_date_val,start_date_train_2,end_date_train_2)**: Function to split data into validation fold and training folds based on its start and end dates. We are not using the
vi. **PCA_transform(data, no_of_components,date_column):** Function to perform PCA transformation on data, based on number of components selected after dropping the 'Date' column and then concatenating the 'Date' column back.
vii. **PCA_select_components(data,no_of_comp,labels):** Function to split data into folds based on number of components given for PCA
viii. **Models_train(data,Model,Kernel='linear',no_of_neighbours=5,Hidden_layer_sizes=(50),C_val=1,Gamma=0.1):** Function which trains the Models on data the data provided after feature selection. The function accepts the model name(eg: SVC, MLPClassifier, KNeighborsClassifier) to be trained as input and provides accuracy and mean F1 score as output for the selected model.

ix. **train_models_initial(data,Model,Model_str,hidden_layer_size=(50)):** Function to train the selected model and return the accuracy. It prints out the F1 Score and accuracy for specified range of PCA components and then finally prints out the number of components and F1 score pertaining to best accuracy for the model.

x. **Nearest_means_train(trian_data_values, input_data_labels):** Function to calculate the class means of the training data

xi. **prediction_nearest_means(values, mean_list):** Function to predict the classes based on nearest means

6 Contributions of each team member

In all phases of this project, both the team members have been equally involved.

7 Summary and conclusions:

In this project a classification model to predict the forest fires was implemented using a Support Vector Machine Classifier (SVM), in addition to that to select this model we have also trained Linear Perceptron, Artificial Neural Networks, K Nearest Neighbors classifier, Ridge Classifier and Bayes Minimum Error Classifier which are clearly explained in EE-559 course. On top of that we have also trained Random Forest Classifier, Logistic Regression on the data obtained after feature expansion and reduction using Principal Component Analysis (PCA), Sequential Feature Selection (SFS) and choosing the features which have most correlation with labels in the expanded feature space. We tested our final model on our test and obtained the performance matrix.