Classification and Principal Component Analysis of Algerian Woodland Fires into Multiple Classes

Mobina Bonkar
Student ID: 40277615

Github:
https://github.com/Mobi1997/INSE6220

Abstract—**Reduction procedures have been presented to allow us to choose a small subset of the essential characteristics from the original components by eliminating duplicate or unnecessary features. The Principal Component Analysis (PCA) approach is among the oldest and most effective methods for dimension reduction. PCA is used in this investigation to reduce the dimensions in order to get an uncorrelated data set. Furthermore, the data set of Algerian forest fires is more easily classified using PCA. The four classification methods employed in this work are random forest, logistic regression, K-Nearest Neighbors, and decision trees. The outcomes of each person have been compared to each other and evaluated based on a number of factors, such as accuracy. The relevance of the traits is then examined using the Shapley and its game theory tools.**

Keywords— size reduction, PCA, categorization, and feature selection

I. Introduction

The quantity of datasets has increased dramatically in recent years across a variety of factors and cases. Finding an attribute (feature) subset that maintains certain unique characteristics of the original data without duplication is the aim of FS [1]. FS selects some of the most useful characteristics from a set of features to reduce the feature space dimension [2]. These features of FS have made it a popular research topic in recent years due to its applicability in a wide range of applications, including text classification [5], optimization problems [4], and classification and regression [3]. We choose and use various machine learning methods on the Algerian forest fire data set. This dataset is used to determine the area most likely to experience a forest fire.

Initially, the PCA approach tried to reduce the number of features. Various categorization approaches are then used to create machine learning models on this data set. New observations are categorized using these models. In this way, the algorithms predict which area is most probable to catch fire. PCA is the first method utilized to the dataset in this study. Three well-known classification methods are then used. These methods are applied on the dimensionally reduced data set, the transferred data set, and the original data set. The results are then compared to each other.

This essay is organized as follows: Principal component analysis was presented in part 2. Section 3 has provided a description of the categorization algorithms. Section 4 provides a description of the data set. The experimental PCA findings are shown in Section 6, and the classification results are clarified in Section 7. The findings of this report are presented in Chapter 8.

.

II. PRINCIPA COMPONENT ANALYSIS

A. PCA is a statistical technique for picture identification and data classification [6]. It is employed when the characteristics exhibit a high association with one another. It breaks down the whole set of variables into a smaller number of fundamental elements that collectively explain the majority of the variance [7].

B. The math-based PCA technique receives the core dataset, whose features are usually coupled, and produces a new dataset with uncorrelated features. Each of the additional qualities is called a PC because of the linear relationship between the basic features. The number of variables observed in the original dataset was equal to the number of components that were recovered [8]. The first discovered primary component accounts for the majority of the volatility in the data. The first main component has nothing to do with the second component that was shown to account for the second-highest percentage of data variation [7].

C. The four stages listed below must be followed in order to apply the PCA algorithm to a database:

1. First Step: Average Centering

To ensure that every feature has the same scale, mean centering is used. The first principal component will undoubtedly be pointing in the direction of the biggest divergence if this is done.

The first step is to determine the average for each column. The value of each attribute is then subtracted from the mean. If the original data is displayed with Y, then the centralized data is displayed with Y:

Y=HX (1)

2. Second step: Determine the matrix of covariance

This phase results in a p * p matrix, where p is the number of data features. The input for this phase is the center matrix from the previous step. The covariance matrix (S) is obtained by multiplying the transposition of the centred matrix by the centred matrix itself using the following formula:

S=1/n-1YY (2)

3. Eigen breakdown

In order to create unique vectors and values, this step entails performing a specific analysis on the covariance matrix. The number of special values will be equal to the number of attributes.

4. Key elements

In this step, a new data matrix (Z) with dimensions n * p is produced. where n is the number of observations and p is the total number of features in the original dataset.

Z=YA (3)

III. CLASSIFICATION ALGORITHMS

There are several applications for data classification techniques in many different scientific fields. In classification models, training data is used to develop a model that predicts the class label for a new sample. A decision tree classifier may provide discrete outputs, while a Naive Bayes classifier may produce continuous results [8].

For this work, the x dataset has been used to test four classifiers. These four classifiers are logistic regression, K-Nearest Neighbours (KNN), lightgbm, and Naive Bayes (NB).

1) K-Nearest Neighbors, or KNN

KNN is a nonparametric classification technique. Related items are grouped together, according to the KNN technique. First, the constant k, which represents the number of neighborhoods utilized for prediction in this approach, must be selected. The value of k can then be set to the nearest objects once the distance between the data or input item and its neighbors has been established. The group having the highest quantity of the title of the class that is k to the neighbor is used to identify the objects. Next, the class label is applied to the input object. As is well known, KNN classification has at least two unsolved issues [11], specifically determining the degree of similarity between two data points and selecting the k value. To address the first issue, a number of approaches have been proposed, including the Manhattan distance, Minkowsky distance, and Euclidean distance, among others [12].

Euclidean distance: $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan distance: $\sum_{i=1}^{k}|x_i - y_i|$

Minkowsky distance: $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

It should be mentioned that for all three distance measures, only continuous variables are acceptable. When using the Hamming distance, categorical factors need to be taken into account.

2) Light Gradiant Boosting Machine

The Naive Bayes method assumes that the characteristics in the data are independent when it comes to categorization. It makes the assumption that every property exists independently of the others, which is frequently not the case in reality. In spite of this simplification, the Naive Bayes method may work well in a variety of circumstances, particularly when there are a lot of features.

- Naive Bayes classifiers come in several varieties, such as:

  Gaussian Naive Bayes: presupposes a Gaussian (normal) distribution for the characteristics.

  Word frequencies in a text document are an example of a discrete feature that is assumed to follow a multinomial distribution by Multinomial Naive Bayes.

  Similar to Multinomial Naive Bayes, Bernoulli Naive Bayes makes the assumption that the features are binary.

  Even with big datasets, naive Bayes classifiers may be trained well and are simple to implement. Although they may be used to different classification issues, they are frequently utilized in text classification tasks like sentiment analysis and spam detection.

3) Light Gradient Boosting Machinethe correct

LightGBM is a high-performance gradient boosting system for classification and regression applications that makes use of decision trees. It is a great option for big datasets because of its scalable and efficient architecture.

To forecast the target variable, which is a categorical variable, LightGBM constructs a sequence of decision trees that are trained in the classification context. The algorithm employs the gradient boosting technique, in which every succeeding tree gains knowledge from the errors of its predecessors.

LightGBM's capacity to effectively manage unbalanced datasets is one of its specialties. It uses two methods to do this: 1) Exclusive Feature Bundling (EFB) and 2) Gradient-based One-Side Sampling (GOSS).

Using GOSS, all instances of the minority class are retained while the majority class is down-sampled. In order to decrease the number of split points in the trees and increase training time, EFB is used to group together categorical data with comparable class distributions.

All things considered, LightGBM is an effective tool for classification jobs requiring great efficiency and accuracy, particularly when dealing with big and unbalanced datasets.

The logistic regression formula can be expressed as a linear equation:

(D1−D) = $\beta 0 + \beta 1 X1 + \beta 2 X2 + \cdots$ (5)

where P is the event's probability and X1, X2,... are the values of the independent variables. After the probability equation was solved, $Probability\ of\ event = P = 1\ 1{+}e{-}(\beta 0{+}\ \beta 1X1{+}\ \beta 2X2{+}\cdots)$ (6)

Forest D at random

Group techniques that employ tree classifiers are referred to as "random forest" strategies. Every tree that results from this classification

A bootstrapped sample of the initial training data is used to train method [12]. A majority vote of the trees determines the classifier's output [13]. The RF method only examines a tiny, randomly selected fraction of the input variables during training. Despite being a user-defined parameter, the number of variables has no bearing on the method. The square root of the number of inputs is the default value [14].

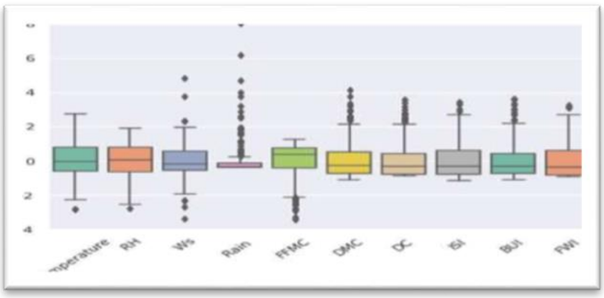
IV. DESCRIPTION OF DATA SET

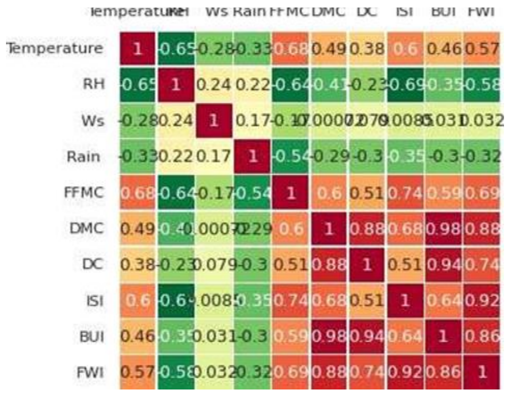
These fires are the subject of the Algerian forest fire dataset. This data collection consists of 244 samples from two separate sites in Algeria. Algeria's Sidi Bel-abbes and Berjaya areas are located in the country's northwest and northeast, respectively.

A. This data collection consists of ten characteristics and one output. The output of this dataset is divided into two classes: class 0 (not fire) and class 1 (fire). The features are provided in the following order: temperature, relative humidity, wind speed, rain, fire weather index, initial spread index, buildup index, fine fuel moisture code, duff moisture code, drought code.

The dataset and features distribution have been examined using box and whisker pots (Fig. 1). It is evident that the majority of characteristics have distributions that closely resemble the normal distribution. Outliers in six of them are skewed to the right.

However, there are outliers and a leftward bias in three features: temperature, relative humidity, and fine fuel moisture code. There are also outliers on both sides of the Wind speed characteristic.





The correlation matrix of the features is displayed in Figure 2.

Numbers with values close to one imply a high connection between two qualities. For example, the greatest association is seen between Duff Moisture Code (DMC) and Buildup Index (BUI) (0.98). Furthermore, there is a roughly strong connection (0.94) between the Buildup Index (BUI) and the Drought Code (DC). The Duff Moisture Code (DMC) and the Fire Weather Index (FWI) have a substantial association of 0.88.

The concepts discussed above are depicted in the pair plot diagram (Fig. 3). This image also illustrates how the output of characteristics that are highly associated with each other appears as a line. To put it another way, it aggregates the data from every class to produce a trend. However, features with low correlation do not show up in this sequence.

For example, there is a substantial correlation between DMC and BUI, and the pair plot result derived from these characteristics appears to be a trend. However, the findings of the pair plot indicated that there was no significant association between WS and BUI. There is no specific pattern to it.

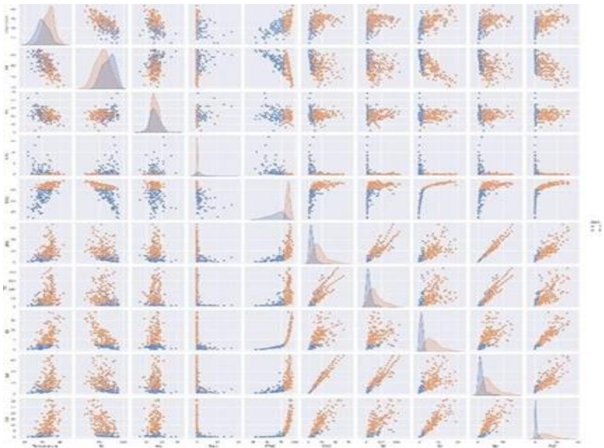


Fig. 3. Pair plot

V. PCA RESULT

The PCA method in this paper uses a dataset called "Algerian forest fires" that has ten characteristics. Transferring a dataset with dimensions n*p to a new dataset with dimensions n*r, where rp, is the goal. The value of n equals the number of samples (244). The number of features is 10, and the value of p is 10. The dimensions are minimized using the eigenvector matrix, denoted here by A as previously mentioned. The number of columns in matrix A, which corresponds to the number of fundamental characteristics, indicates the number of Principle Components (PCs). In this case, the final PC has the least amount of information, while the first PC has the most.

The matrix A is created as follows:

$$
\begin{matrix}
0.298 & -0.348 & 0.117 & 0.098 & -0.786 & -0.342 & -0.175 & 0.019 & -0.006 & -0.003 \\
-0.277 & 0.399 & -0.421 & -0.019 & -0.164 & -0.627 & 0.407 & 0.025 & 0.015 & 0.011 \\
-0.040 & 0.539 & 0.420 & -0.640 & -0.304 & 0.125 & -0.085 & 0.078 & -0.008 & 0.002 \\
-0.196 & 0.215 & 0.646 & 0.608 & -0.083 & 0.015 & 0.345 & -0.050 & 0.003 & 0.005 \\
0.348 & -0.232 & -0.041 & -0.238 & -0.105 & 0.289 & 0.813 & 0.058 & -0.087 & 0.023 \\
0.374 & 0.266 & -0.089 & 0.213 & 0.047 & 0.041 & -0.070 & 0.637 & 0.243 & 0.512 \\
0.330 & 0.374 & -0.249 & 0.18 & -0.168 & 0.244 & -0.052 & -0.699 & 0.082 & 0.266 \\
0.368 & -0.073 & 0.311 & -0.189 & 0.366 & -0.457 & 0.072 & -0.236 & 0.568 & -0.058 \\
0.37 & 0.313 & -0.145 & 0.201 & -0.022 & 0.1 & -0.041 & 0.187 & 0.045 & -0.809 \\
0.393 & 0.119 & 0.159 & -0.007 & 0.281 & -0.333 & -0.056 & -0.043 & -0.776 & 0.096
\end{matrix}
$$

the eigenvalues match:

$$
\lambda = \begin{bmatrix} 5.75 \\ 1.589 \\ 0.92 \\ 0.799 \\ 0.39 \\ 0.25 \\ 0.22 \\ 0.092 \\ 0.0162 \\ 0.0038 \end{bmatrix}
$$

As can be seen, the values are displayed in decreasing sequence, with the greatest value appearing first and the lowest value at the end.

As a result, the first PC shows the most variation in the data.

The amount of variation that each PC displays is shown using a scree plot (Fig. 4). Prior to drawing an elbow diagram or a scree plot, you must ascertain the value of each l. In this instance, Formula 1 is used:

$$
l_j = \frac{\lambda_j}{\sum_{j=1}^{p} \lambda_j} \times 100 \,, For\ j = 1, \ldots, p \qquad (7)
$$

Getting 80% of the data for this report from computers is the goal. According to the Scree plot figure, the elbow is evident at the third PC. Furthermore, the pareto diagram (Fig. 5) illustrates it. Hence the quantity of information is the sum of the information from the first and second PCs plus the information from the third PC.

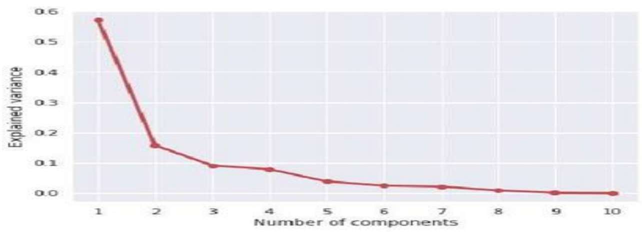


Fig. 4. Scree plot

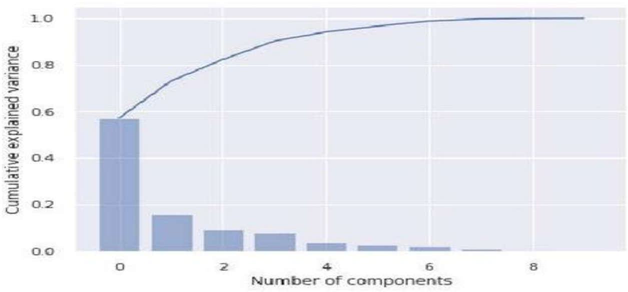


Fig. 5. Pareto plot (Explained Variance)

exceeds 80%. Thus, r is found to be three in this report. As can be seen in the two figures above, the PCA technique successfully converted the original dataset with 10 features into a new dataset with 3 additional features and offers a reasonable decrease.

The algorithm offers the following three PCs:

$Z1 = 0.298X1 - 0.27X2 - 0.19X4 + 0.34X5 + 0.37X6 + 0.329X7 + 0.36X8 + 0.37X9 + 0.39X10$

$Z2 = -0.348X1 + 0.398X2 + 0.53X3 + 0.21X4 - 0.23X5$
$\quad + 0.269X6 + 0.374X7 + 0.31X9 + 0.11X10$

$Z3 = 0.116X1 - 0.42X2 + 0.42X3 + 0.646X4 - 0.249X7$
$\quad + 0.311X8 - 0.144X9 + 0.11X10$

While certain aspects contribute negatively to the first PC(Z1) others do the opposite.
For example, with values of 0.374, 0.37, and 0.39, respectively, traits 6, 9, and 10 provide the largest contributions. Features 2 and 4 have the worst impacts, with values of -0.277 and -0.19. The influence of the third feature can also be ignored because of its small magnitude.

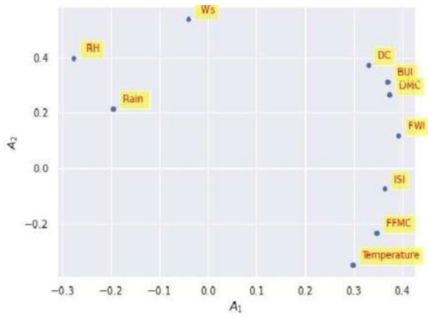


Fig. 6. coefficient plot

The most advantageous contributions are made by the second and third features on the second PC, which have respective values of 0.398 and 0.539. This suggests that the two factors that affect this PC the most are relative humidity and wind speed. Furthermore, the first feature (Temperature) and the fifth feature (FFMC, or Fine Fuel Moisture Code) had the highest negative impacts, with values of -0.348 and -0.23. The influence of Feature 8 (ISI) may also be overlooked.

With values of 0.646 and 0.42, respectively, the third and fourth characteristics (rain and wind speed) for the third PC had the most beneficial impacts. However, with a value of -0.42, the second characteristic (Relative Humidity) has the most negative influence.

A plot of coefficients is shown in Figure 6. To make the earlier ideas of positive and negative contributions for each feature on the first two PCs easier to grasp, the coefficient plot is created for the first two PCs even though r = 3 in this report. The first PC is most positively impacted by the FWI (Fire Weather Index), BUI (Buildup Index), and DMC (Duff Moisture Code) characteristics, which are located in the rightmost section of the figure and have values near 0.4. Furthermore, the first PC is most negatively impacted by the RH (Relative Humidity) characteristic, which is located at the far left of the chart.

It is also evident that the feature with the most beneficial influence is WS (Wind Speed), which is located at the top of the graph for the second PC. Furthermore, the most harmful impacts are caused by temperature and FFMC, which are located close to the base of the coefficient graph and have a value of around -0.2.

The impact of each characteristic on PCs is also displayed in the biplot graph. This graph's result is identical to that of the previous two methods. This report uses a three-dimensional graph since r = 3 (three PCs are selected).

Whether the impact is positive or negative is also indicated by the direction of the vector. The vector's angle decreases as the feature's influence on the PC increases. The effect of a feature on the PC can be ignored when the angle of a vector is around 90 degrees.

To assist the reader better understand how each feature affects PCs, fig. 7 includes both two-dimensional and three-dimensional biplot graphs.
As observed in the picture, the FWI vector has the most advantageous impact on this PC since it has the lowest angle with the PC1 axis. Furthermore, the first PC is at a modest angle to the BUI and DMC vectors, suggesting that they have a positive effect on it. The direction and angle of the RH and rain vectors also demonstrate that they are the most harmful to the original PC.
In the case of PC2, the WS vector, which represents the wind speed feature, has the shortest angle with the PC 2 axis. Given that it is traveling in the correct direction, this vector has a positive contribution. The Relative Humidity characteristic, represented by the RH vector, has the lowest angle with the PC2 axis and contributes positively due to its direction.

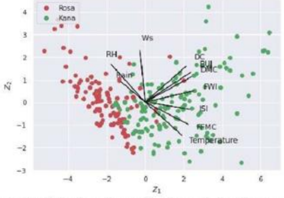


Fig.7 .a) 2D biplot for understanding the angel of each vector

Fig.7 .b) 3D biplot, Investigation of vector angles in three-dimensional space

VI. CLASSIFICATION RESUT

This section examines how well the categorization algorithms performed on the Algerian forest fire dataset.

The classifiers are initially run on the original dataset, and each classifier's output is compared to the others.

The output of the PCA technique is then used to analyze the classifier's performance. In this instance, the classifier's input is the PCA method's output rather than the original data set containing the important characteristics. Instead of being the primary features in this new dataset, the PCs generated by the PCA approach are believed to represent the dimensions.

The two primary metrics used in this study to evaluate the algorithms' performance are accuracy and F1-Score. The accuracy measure shows how well the algorithm performs in accurately identifying data. Additionally, the F1-Score index is employed here since multiclass classification models are being studied. Precision and recall, two important metrics that are calculated using the following method, are combined in this index:

$$F_{1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (9)$$

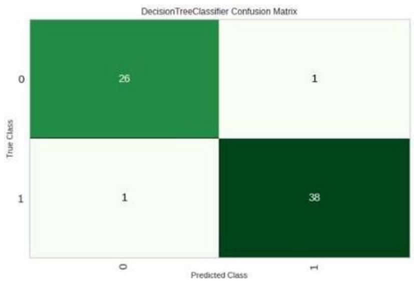


Fig. 8. Confusion Matrix for Decision Tree

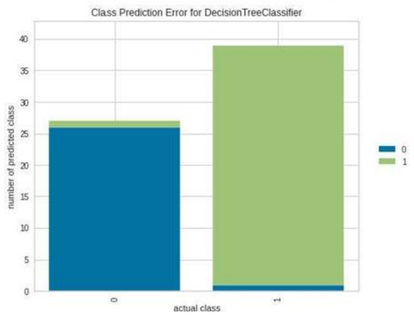


Fig. 9. Prediction error for Decision Tree

The decision tree technique yields 96% accuracy with a standard deviation of 0.053 on the original data set. The F1-Score for this method is 96.17%, with a standard deviation of 0.0504. Additionally, while using the adjusted model of this technique, the metrics have altered significantly, although this difference is negligible.

However, when the technique is used to the dataset produced by the PCA approach, accuracy and F1-Score are much reduced. In this case, the F1-Score is 88.6% and the algorithm's accuracy is 87.75%. This proves that the decision tree method performed better on the old dataset than the new dataset. Just one incident out of 27 that belonged to class zero (not fire) was mistakenly assigned to class one, according to the confusion matrix (Fig. 8). Furthermore, just one sample—fire—out of 39 examples from class one was inadvertently included in class zero. This is further demonstrated by the prediction error plot (Fig. 9).

The algorithm's performance is also examined differently in the AUC-ROC diagram (Fig. 10). The two indicators used in this instance are AUC and ROC. When selecting various algorithmic thresholds, this curve examines the classification problems. In this case, AUC stands for the level or resolution criterion. The graph shows that the AUC for both classes is 0.99, which is also the micro-average.

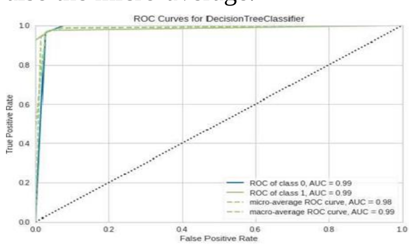


Fig. 10. AUC and ROC curve for Decision Tree

The next algorithm to be looked at is the KNN algorithm. This method produces a 91.50% accuracy rate with a 0.0667 standard deviation and a 92.32% F1-Score with a 0.0608 standard deviation on the original data set. In tune mode, this algorithm's accuracy rose by more than 2% to 93.46%, and its F1-Score was equal to 93.48%.

Applying this method to the output of the PCA algorithm yields a result that is almost equivalent to the AUC-ROC curve (Fig. 13) shows that the separability requirement of both classes is the same in this algorithm (equal to 0.91).

One may claim that, when applied to the original data, the Logistic Regression method has one of the highest accuracy metrics. The accuracy is 96.71%, the standard deviation is 0.329, and the F1-Score is 96.89%. The accuracy in tune mode has improved by around 1% to 97.38% with an F1-Score of 97.55%. This method performed poorly on the PCA algorithm's output, reducing its accuracy by around 4.5 percent. In this case, the method yields an F1-Score of 92.42% and an accuracy of 92.25%.

In tune mode, the Logistic Regression algorithm performs admirably and, under ideal conditions, yields an accuracy of about 100%. An overview of the algorithm's output is given by the graphs (Figs. 14, 15, and 16) that show the confusion matrix, prediction error, and class report. In this case, it is clear that every sample has been appropriately classified. Along with accuracy, the two markers—recall and precision—are also 100%.

Both the normal and tune modes of the Random Forest algorithm yield the same level of accuracy. In both cases, the F1-Score is 97.41% and the accuracy is 97.33%. The method was unable to maintain its high performance when used on fresh datasets, such as Logistic Regression. In this case, the F1-Score is 91.57% and the accuracy is around 90.88%, having dropped by 7%. The confusion matrix (Fig. 17) and prediction error diagram (Fig. 18) demonstrate that this approach correctly classified every sample as being in class 0. Furthermore, three samples were incorrectly assigned to class 0 out of 39 samples from class one, whereas 36 samples were correctly recognized. The class report diagram (Fig. 19) shows the precision and recall status for each class.

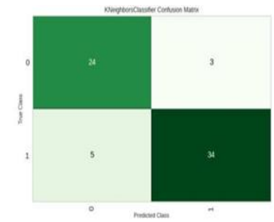


Fig. 11. Confusion Matrix for KNN

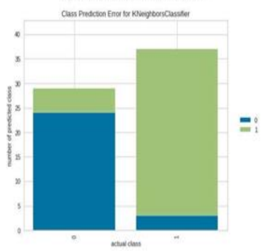


Fig. 12. Prediction error for KNN

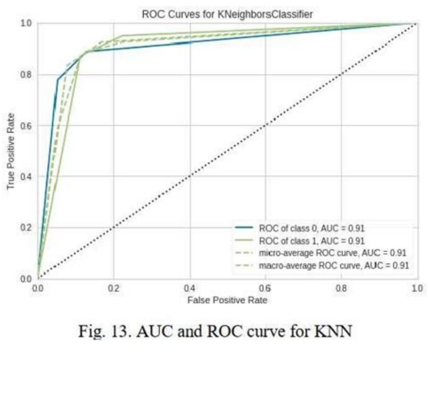


Fig. 13. AUC and ROC curve for KNN

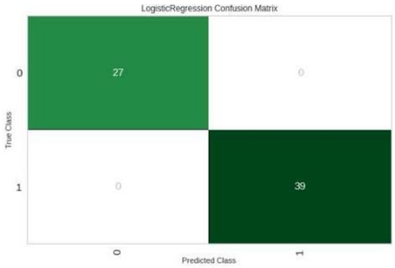


Fig. 14. Confusion Matrix for Logistic Regression

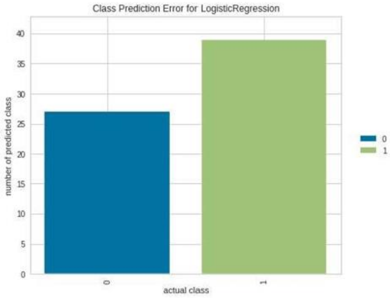


Fig. 15. Prediction error for Logistic Regression

Fig. 19. Class report for Random forest

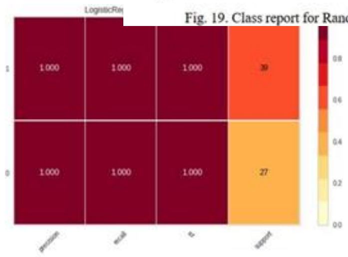


Fig. 16. Class report for Logistic Regression

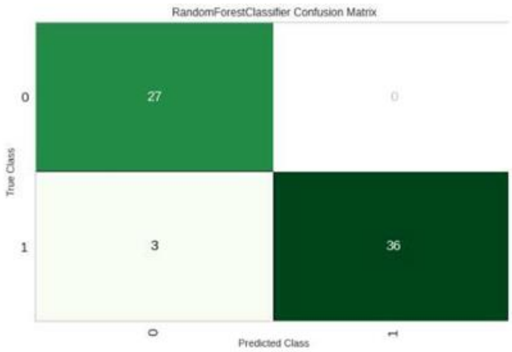


Fig. 17. Confusion Matrix for Random forest

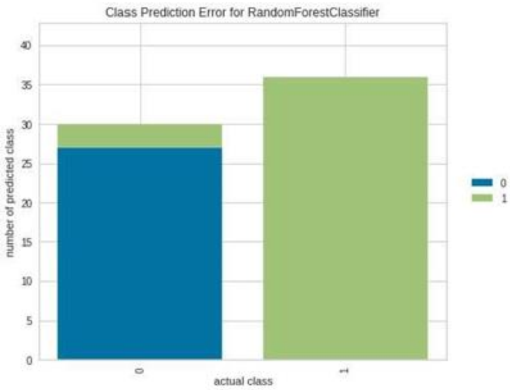


Fig. 18. Prediction error for Random Forest

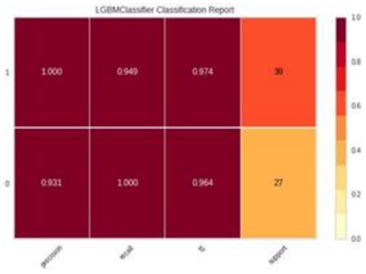


Fig. 20. Class report for best model

It is clear from Table 2 that logistic regression, naive bayes, and knn each provide the best outcomes when used on the new datasets. In that order, the Ada boost, Decision Tree, and Dummy algorithms all exhibit subpar performance. In terms of run time, the three quickest algorithms are SVM, Ridge, and Dummy.

Random Forest, Extra Tree, and KNN are the slowest algorithms.

Table 1. some classification report before PCA

Model		Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.98	1.00	0.99	0.98	0.98	0.96	0.96	0.07
rf	Random Forest Classifier	0.97	0.99	0.96	0.99	0.97	0.95	0.95	0.55
lr	Logistic Regression	0.97	1.00	0.96	0.98	0.97	0.93	0.94	0.47
ada	Ada Boost Classifier	0.97	0.99	0.96	0.98	0.97	0.93	0.94	0.10
et	Extra Trees Classifier	0.97	0.99	0.98	0.97	0.97	0.93	0.94	0.47
gbc	Gradient Boosting Classifier	0.96	0.98	0.96	0.97	0.96	0.92	0.92	0.09
dt	Decision Tree Classifier	0.96	0.96	0.95	0.98	0.96	0.92	0.92	0.02
ridge	Ridge Classifier	0.92	0.00	0.93	0.93	0.93	0.84	0.85	0.03
knn	K Neighbors Classifier	0.92	0.97	0.94	0.91	0.92	0.83	0.84	0.12
lda	Linear Discriminant Analysis	0.92	0.97	0.92	0.93	0.92	0.83	0.84	0.02
svm	SVM - Linear Kernel	0.83	0.00	0.79	0.82	0.79	0.67	0.70	0.02
nb	Naive Bayes	0.80	0.93	0.98	0.75	0.84	0.59	0.64	0.02
qda	Quadratic Discriminant Analysis	0.58	0.57	0.63	0.61	0.61	0.14	0.15	0.02
dummy	Dummy Classifier	0.54	0.50	1.00	0.54	0.70	0.00	0.00	0.01

Table 2. some classification report after PCA

VII. CNCLUSION

The algorithms of PCA, Random Forest, KNN, Logistic Regression, and Decision Tree were investigated in this work. The data set of Algerian forest fires, which is related to estimating the risk of fire in combat, was used to evaluate these methods.

Before being used in the PCA approach, the dataset underwent statistical analysis. The PCA technique, which sought to minimize the number of variables and maintain 80% of the data, could only add three PCs.

Additionally, the original dataset was fed into several classifiers. Out of the four algorithms analyzed for the research, the Random Forest approach was shown to perform the best. Applying classifiers to the new dataset (the output PCs of the PCA technique) improved the performance of the Logistic Regression algorithm. When comparing different methods, it may be claimed that in some situations, the accuracy of the classifier can be improved by applying PCA output. In other situations, it could potentially negatively affect accuracy. Overall, it can be concluded that the classifiers being studied may provide a decent F1-Score together with a respectable degree of accuracy.

VIII. REFERENCES

.Dy, Jennifer G., et al. "Unsupervised feature selection applied to content-based retrieval of lung images." IEEE transaction on pattern analysis and machine intelligence 25.3 (2003): 373-378.