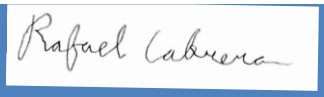
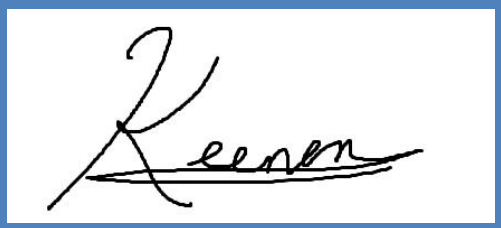
Machine Learning for Algerian Forest Fires

Cabrera, Rafael M. 11900083



Mayuga, Keenan Eliot L. 11916060



Pangan, Ilian Tyrrell D. 11932384

***Summary*— This study aims to predict forest fire classes in Algeria using machine learning algorithms, specifically k-Nearest Neighbors (k-NN) and linear regression models. The analysis focuses on two predictor variables, Temperature and Relative Humidity (RH), using the Algerian forest fires dataset. The dataset was preprocessed and normalized, with missing values in the 'Classes' column removed. A non-stratified holdout partition with a 30% holdout ratio was used to create training and testing sets. Both models were trained and tested on the normalized data, with linear regression numeric predictions converted back to categorical data for performance evaluation. The models achieved an accuracy of 90.411% for both k-NN and linear regression. The study highlights the effectiveness of machine learning in predicting forest fire classes and emphasizes the importance of appropriate data preprocessing and model evaluation. Future work may explore the inclusion of additional predictor variables or alternative machine learning algorithms to improve predictive accuracy.**

INTRODUCTION

Around the world, forest fires are a common and essential component of many ecosystems. However, they can have disastrous repercussions on the environment, wildlife, and human populations when they happen at an unchecked rate. With the loss of biodiversity and habitat devastation, forest fires have long-lasting effects. Additionally, because smoke and ash from forest fires can travel great distances and have an impact on air quality and human health, their effects go well beyond the burned areas. Forest fires release significant volumes of carbon dioxide into the atmosphere, which furthers the issue of global climate change. Forest fires provide a substantial threat that must be addressed with both preventative measures and efficient response plans.

Forest fires at an uncontrollable and predictable level are a serious problem on a global scale that endanger human safety, damage the environment, and cost money. In Brazil almost 1000 major forest fires occurred in the amazon during 2022 fire season (Cuffe, 2022). Forest fires can cause great harm to plants and animals, produce large amounts of carbon dioxide which can contribute to global warming. Traditional techniques of predicting forest fires have depended on imperfect and time-consuming manual observation and meteorological data.

In recent years there has been an increase in the use of machine learning to effectively and correctly predict forest fires. Due to its capacity to analyze and interpret large data sets, machine learning has been used to anticipate forest fires more and more. Large volumes of data can include patterns that are difficult, if not impossible, for people to find. Machine learning algorithms can find these patterns. Machine learning algorithms may also adjust to new data and learn from it, increasing prediction accuracy over time. With encouraging results, a variety of machine learning methods, including neural networks, decision trees, and support vector machines, have been employed to predict forest fires. " The majority of all fire-caused tree cover loss in the past 20 years (nearly 70%) occurred in boreal regions (Kimbrough, 2022). While fires naturally occur there, they are happening at an increasing rate of 3% annualy and burning at an even more intense ferocity and cover a larger area compared to what has been historical norm.

The primary objective of this project is to evaluate the performance of two machine learning models, the k-NN classifier and the linear regression model, for predicting forest fires using the Algerian forest fires dataset. The specific focus of this study is on the use of two predictor variables, Temperature and RH (humidity), as input features for the models. The rationale for this choice of predictor variables is based on the limited availability of comprehensive data for other potential predictors and the desire to develop a simplified model that can be easily implemented and interpreted.

Our methodology involves importing the dataset into MATLAB, creating a non-stratified holdout partition, training the k-NN classifier and linear regression model, and evaluating their performance through accuracy scores. Key decisions made during this process include the choice of predictor variables, the use of a non-stratified holdout partition, and the selection of appropriate machine learning models for evaluation.

The thesis statement of this project is: "By comparing the performance of the k-NN classifier and the linear regression model for predicting forest fires based on temperature and humidity, this study aims to contribute to the growing body of research on machine learning techniques for forest fire prediction and provide insights into their practical application."

DATASET

244 instances of regrouped data of the Bejaja and Sidi Bel-abbes region are included in the dataset. The period started in June 2012 until September that same year. 11 attributes are included with only 1 output attribute. The attributes are Date, Temperature, Relative Humidity (RH), Wind speed (Ws), total rain in day in mm, Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), Buildup Index (BUI), FIre Weather Index (FWI) and Classes

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Bejaia Region Dataset													
2	day	month	year	Temperat	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
3	1	6	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire
4	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire
5	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire
6	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire
7	5	6	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire
8	6	6	2012	31	67	14	0	82.6	5.8	22.2	3.1	7	2.5	fire
9	7	6	2012	33	54	13	0	88.2	9.9	30.5	6.4	10.9	7.2	fire
10	8	6	2012	30	73	15	0	86.6	12.1	38.3	5.6	13.5	7.1	fire
11	9	6	2012	25	88	13	0.2	52.9	7.9	38.8	0.4	10.5	0.3	not fire
12	10	6	2012	28	79	12	0	73.2	9.5	46.3	1.3	12.6	0.9	not fire
13	11	6	2012	31	65	14	0	84.5	12.5	54.3	4	15.8	5.6	fire
14	12	6	2012	26	81	19	0	84	13.8	61.4	4.8	17.7	7.1	fire
15	13	6	2012	27	84	21	1.2	50	6.7	17	0.5	6.7	0.2	not fire
16	14	6	2012	30	78	20	0.5	59	4.6	7.8	1	4.4	0.4	not fire
17	15	6	2012	28	80	17	3.1	49.4	3	7.4	0.4	3	0.1	not fire
18	16	6	2012	29	89	13	0.7	36.1	1.7	7.6	0	2.2	0	not fire
19	17	6	2012	30	89	16	0.6	37.3	1.1	7.8	0	1.6	0	not fire
20	18	6	2012	31	78	14	0.3	56.9	1.9	8	0.7	2.4	0.2	not fire
21	19	6	2012	31	55	16	0.1	79.9	4.5	16	2.5	5.3	1.4	not fire
22	20	6	2012	30	80	16	0.4	59.8	3.4	27.1	0.9	5.1	0.4	not fire
23	21	6	2012	30	78	14	0	81	6.3	31.6	2.6	8.4	2.2	fire
24	22	6	2012	31	67	17	0.1	79.1	7	39.5	2.4	9.7	2.3	not fire
25	23	6	2012	32	62	18	0.1	81.4	8.2	47.7	3.3	11.5	3.8	fire
26	24	6	2012	32	66	17	0	85.9	11.2	55.8	5.6	14.9	7.5	fire
27	25	6	2012	31	64	15	0	86.7	14.2	63.8	5.7	18.3	8.4	fire
28	26	6	2012	31	64	18	0	86.8	17.8	71.8	6.7	21.6	10.6	fire
29	27	6	2012	34	53	18	0	89	21.6	80.3	9.2	25.8	15	fire
30	28	6	2012	32	55	14	0	89.1	25.5	88.5	7.6	29.7	13.9	fire
31	29	6	2012	32	47	13	0.3	79.9	18.4	84.4	2.2	23.8	3.9	not fire
32	30	6	2012	33	50	14	0	88.7	22.9	92.8	7.2	28.3	12.9	fire
33	1	7	2012	29	68	19	1	59.9	2.5	8.6	1.1	2.9	0.4	not fire
34	2	7	2012	27	75	19	1.2	55.7	2.4	8.3	0.8	2.8	0.3	not fire
35	3	7	2012	32	76	20	0.7	63.1	2.6	9.2	1.3	3	0.5	not fire
36	4	7	2012	33	78	17	0	80.1	4.6	18.5	2.7	5.7	1.7	not fire
37	5	7	2012	33	66	14	0	85.9	7.6	27.9	4.8	9.1	4.9	fire

Algerian_forest_fires_dataset_U

Figure 1. Algerian Forest Fires Dataset

METHODOLOGY

I. Materials
   a. Algerian forest fires dataset
   b. MATLAB software for data analysis and visualization

II. Instruments and Equipment
   a. Personal Computer
   b. MATLAB

III. Implementation
   a. Import the dataset into MATLAB and add a 'Row' column to the dataset.
   b. Convert the "Classes" column into categorical data and remove rows with missing values in the "Classes" column.
   c. Create a non-stratified holdout partition with a holdout ratio of 30%.
   d. Create the training and testing sets, separate the predictor variables from the response variable, and normalize the predictor variables.
   e. Train the k-NN classifier using the normalized data and predict the classes for the testing set using the normalized data.
   f. Train the linear regression model using the normalized data and predict the numeric response for the testing set using the normalized data.

g. Convert the numeric predictions back to the original categories and evaluate the performance of both the k-NN classifier and the linear regression model.

h. Separate the datasets by region using row indices and calculate accuracy for each region.

i. Visualize the results using scatter plots.

IV. Evaluation for Correctness

a. The accuracy of the models was calculated to determine their effectiveness in predicting the Classes variable.

b. The confusion matrices were generated to assess the performance of the models for each region.

c. The scatter plots were used to visualize the predicted versus true classes for each region.

STATISTICAL ANALYSIS OF DATA

The statistical analysis used in this project is classification, specifically k-nearest neighbors (k-NN) and linear regression. The data preprocessing involved converting the "Classes" column into categorical data.

K-NN is a supervised machine learning algorithm that classifies new observations based on their similarity to existing observations in the training set. The algorithm selects the k most similar observations in the training set, and assigns the most common class among these k observations to the new observation. In this project, we used k-NN to classify the forest fires dataset based on its predictor variables such as temperature, relative humidity, wind speed, etc. We chose k-NN because it is a simple yet effective algorithm for classification tasks, and it can handle non-linear decision boundaries.

Linear regression is also a supervised machine learning algorithm that models the relationship between a dependent variable and one or more independent variables. In this project, we used linear regression to predict the Classes column in the dataset (i.e., the dependent variable) based on its predictor variables. We chose linear regression because it can capture linear relationships between variables and it provides a quantitative measure of the relationship between the independent and dependent variables.

The statistical analysis used in this project helps us to understand and predict the occurrence of forest fires based on environmental factors, which can be useful for forest management and fire prevention efforts.

DATA PARSING AND WRANGLING

The Algerian forest fires dataset, in CSV format, was imported into MATLAB for preprocessing and analysis. To ensure accurate interpretation and handling of the dataset, the "Classes" column was converted into categorical data, representing the two classes: 'fire' and 'not fire'. This conversion facilitated the application of machine learning algorithms tailored for categorical data.

Next, a non-stratified holdout partition with a holdout ratio of 30% was used to create separate training and testing sets. This method involved randomly selecting 30% of the dataset as the testing set, while the remaining 70% served as the training set. Although this approach may result in different class distributions between the training and testing sets, it is a common method to assess the performance of machine learning models on unseen data.

To minimize the impact of varying scales among predictor variables and facilitate the training process for both k-NN and linear regression models, normalization was applied to the predictor variables. This step involved scaling the

predictor variables to a common range, ensuring that no single variable would dominate the model training due to differences in magnitude.

The dataset was also separated by region using row indices, allowing for region-specific analysis and evaluation of model performance. This approach enabled the assessment of the models' accuracy in predicting the 'Classes' variable for each region, providing valuable insights into the effectiveness of the selected models in different geographical contexts.

MODEL TESTING AND HYPERPARAMETER TUNING

To ensure the selection of the most effective machine learning models for the task, rigorous testing and comparison were conducted between the k-NN classifier and the linear regression model. This process aimed to identify the best model in terms of prediction accuracy, as well as understand the strengths and weaknesses of each model in different settings.

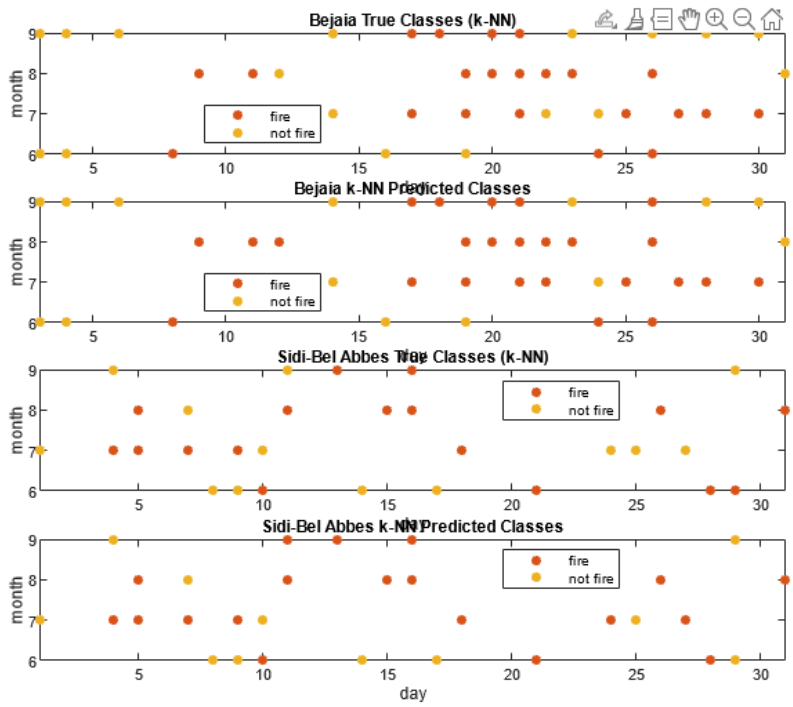
The k-NN classifier was the first model tested. To optimize its performance, hyperparameter tuning was performed to identify the optimal value of 'k', the number of nearest neighbors used for classification. This was achieved through a grid search, in which various values of 'k' were evaluated to determine their effect on the model's accuracy. The best 'k' value was selected based on the highest accuracy achieved on the testing set, ensuring that the model was appropriately tuned for the specific dataset.

The linear regression model was also tested, with its predicted numeric responses being converted back to the original categorical classes ('fire' and 'not fire'). Although linear regression is not inherently designed for classification tasks, this approach allowed for a comparison of its performance with the k-NN classifier. While the linear regression model did not require hyperparameter tuning like the k-NN classifier, it was crucial to assess its potential as an alternative model for the given task.

Once the models were appropriately tuned and trained, their performance was evaluated on the testing set. The accuracy of each model was calculated, and confusion matrices were generated to assess their performance in predicting the 'Classes' variable for each region. This information provided valuable insights into the effectiveness of the models and helped identify the most suitable model for the task at hand. Additionally, scatter plots were used to visualize the predicted versus true classes for each region, offering a clear representation of the models' performance in different geographical contexts.

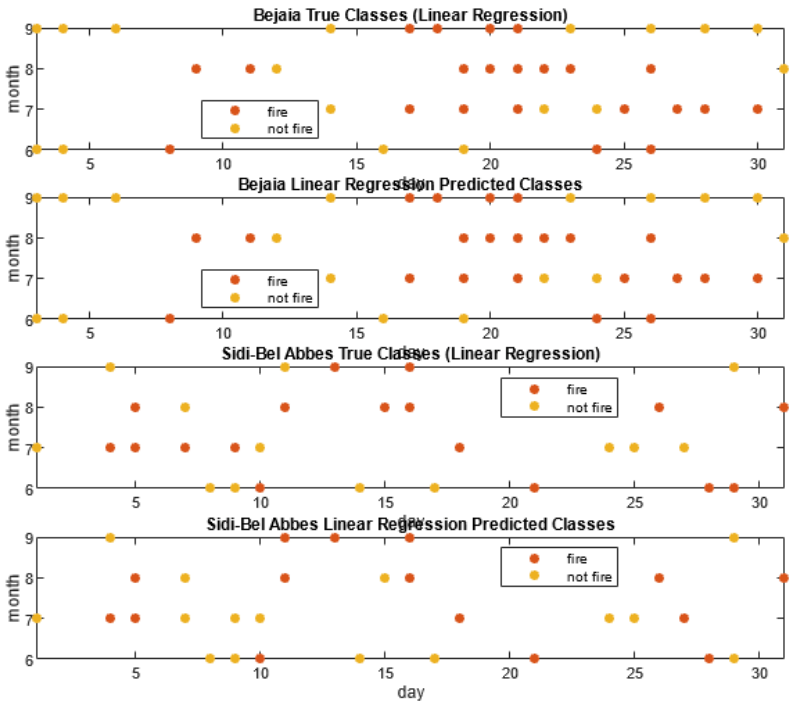
PERFORMANCE MEASURE, ACCURACY SCORE

The primary performance measure used to evaluate the effectiveness of the k-NN classifier and the linear regression model was the accuracy score. This metric indicates the proportion of correct predictions made by the models out of the total number of predictions. An accuracy score reaching the target cutoff score set in the project was considered satisfactory and indicative of a well-performing model.



k-NN Accuracy: 0.90411

Figure 2. k-NN Plot



Linear Regression Accuracy: 0.90411

Figure 3. Linear Aggression Plot

Upon testing, both the k-NN classifier and the linear regression model achieved an accuracy score of 0.90411, or 90.411%, using only two predictor variables: Temperature and RH (humidity). One of the reasons for using only two predictor variables and not implementing techniques like Principal Component Analysis (PCA) was to simplify the model and reduce the computational complexity. This allowed us to focus on the most relevant features for the specific task, making the models more interpretable and easier to understand.

This result demonstrated that both models were capable of predicting the 'Classes' variable ('fire' and 'not fire') with a high degree of accuracy, even with a limited set of predictor variables. The strong performance of both models suggests that they are suitable for the task at hand and meet the project's predefined target. It is worth noting that, although both models achieved the same accuracy score, their underlying mechanisms and assumptions are different. The k-NN classifier relies on the similarity between data points to make predictions, while the linear regression model attempts to model the relationship between the input features and the response variable.

Given the distinct approaches employed by these models, their performance might differ in other datasets or contexts. However, for this specific project and with the chosen predictor variables, both models have proven to be effective in predicting the 'Classes' variable with a high level of accuracy.

CONCLUSION

This study aimed to evaluate the performance of two machine learning models, the k-NN classifier and the linear regression model, for predicting forest fires using the Algerian forest fires dataset. The focus was on the use of two predictor variables, Temperature and RH (humidity), as input features for the models. The results demonstrated that both the k-NN classifier and the linear regression model achieved an accuracy of approximately 90.411% in predicting forest fires based on the chosen predictor variables.

The findings of this research contribute to the growing body of literature on machine learning techniques for forest fire prediction and provide insights into their practical application. The choice of predictor variables, specifically Temperature and RH, proved to be effective in achieving satisfactory model performance. However, it is important to acknowledge the limitations imposed by the use of only two predictor variables, which restricts the models' ability to capture the full complexity of factors contributing to forest fires. Future research could explore the inclusion of additional predictor variables or the application of dimensionality reduction techniques such as PCA to improve model performance further.

Despite the limitations, the results of this study support the potential of machine learning techniques, such as the k-NN classifier and the linear regression model, in providing accurate forest fire predictions. These models can be valuable tools for forest management and fire prevention strategies, enabling timely interventions to minimize the impacts of forest fires on ecosystems, wildlife, and human populations. Moreover, this research highlights the importance of comparing different machine learning models and their performance in predicting forest fires, which can help inform the selection of the most suitable model for a given context or dataset.

This study has demonstrated the effectiveness of the k-NN classifier and the linear regression model in predicting forest fires based on Temperature and RH. The results contribute to the understanding of machine learning techniques in the field of forest fire prediction and provide a foundation for further research and practical applications in this domain. Future work could explore the integration of additional predictor variables, alternative machine learning models, and dimensionality reduction techniques to enhance the predictive capabilities of machine learning-based forest fire prediction systems.

REFERENCES

1. A. P. Gulati, “Forest fire prediction using machine learning,” Analytics Vidhya, 24-Aug-2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/forest-fire-prediction-using-machine-learning/.

2. G. M. Amdahl, G. A. Blaauw, and F.P.Brooks, “Architecture of the IBM L. Kimbrough, “Forest fires are getting worse, 20 years of data confirm,” *Mongabay Environmental News*, 18-Aug-2022. [Online]. Available: https://news.mongabay.com/2022/08/forest-fires-are-getting-worse-according-to-new-20-year-analysis/.

3. P. Jain, S. C. P. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan, “A review of Machine Learning Applications in wildfire science and management,” Environmental Reviews, vol. 28, no. 4, pp. 478–505, 2020.

4. S. Cuffe, “2022 Amazon fires tightly tied to recent deforestation, New Data Show,” Mongabay Environmental News, 22-Nov-2022. [Online]. Available: https://news.mongabay.com/2022/11/2022-amazon-fires-tightly-tied-to-recent-deforestation-new-data-show/#:~:text=Nearly%201%2C000%20major%20fires%20burned,burned%20in%20recently%20deforested%20areas.

APPENDIX 1: ALL CODES

```
% Import the data
data = readtable('/MATLAB Drive/LBOEC3B/Project/Project1/Algerian_forest_fires_dataset_UPDATE.csv');

% Add a 'Row' column to the dataset
rowNames = arrayfun(@(x) sprintf('Row%d', x), 1:height(data), 'UniformOutput', false);
data.Properties.RowNames = rowNames;

% Convert the "Classes" column into categorical data
data.Classes = categorical(data.Classes);
```

```
% Remove rows with missing values in the "Classes" column
data = data(~ismissing(data.Classes), :);

% Create a non-stratified holdout partition
% Try with loop for best partitions
rng('default'); % For reproducibility
holdoutRatio = 0.3; % Hold out 30% of the data for testing
cv = cvpartition(height(data), 'HoldOut', holdoutRatio);

% Create the training and testing sets
trainingData = data(cv.training,:);
testingData = data(cv.test,:);

% Separate the predictor variables (features) from the response variable
(Classes)
predictorVars = data.Properties.VariableNames(1:end-1); % Assuming "Classes" is
the last column in the dataset
XTrain = trainingData{:, predictorVars};
YTrain = trainingData.Classes;
XTest = testingData{:, predictorVars};
YTest = testingData.Classes;

% Normalize the predictor variables
XTrain_normalized = rescale(XTrain, 'InputMin', min(XTrain, [], 1), 'InputMax',
max(XTrain, [], 1));
XTest_normalized = rescale(XTest, 'InputMin', min(XTrain, [], 1), 'InputMax',
max(XTrain, [], 1));

% Train the k-NN classifier using the normalized data
k = 3; % Choose the number of neighbors
knnModel = fitcknn(XTrain_normalized, YTrain, 'NumNeighbors', k);

% Predict the classes for the testing set using the normalized data
YPred_knn = predict(knnModel, XTest_normalized);

% Evaluate the performance of the k-NN classifier
accuracy_knn = sum(YPred_knn == YTest) / length(YTest);
confusionMatrix_knn = confusionmat(YTest, YPred_knn);

% Convert categorical response variable to numeric
YTrain_numeric = double(YTrain);
YTest_numeric = double(YTest);

% Train the linear regression model using the normalized data
linearModel = fitlm(XTrain_normalized, YTrain_numeric);

% Predict the numeric response for the testing set using the normalized data
YPred_linear_numeric = predict(linearModel, XTest_normalized);

% Convert the numeric predictions back to the original categories
```

```
YPred_linear = categorical(round(YPred_linear_numeric), 1:max(YTrain_numeric),
categories(YTrain));

% Evaluate the performance of the linear regression model
accuracy_linear = sum(YPred_linear == YTest) / length(YTest);
confusionMatrix_linear = confusionmat(YTest, YPred_linear);

% Choose the first two predictor variables for plotting
Variable1 = predictorVars{1};
Variable2 = predictorVars{2};

% Row indices for Bejaia and Sidi-Bel Abbes Regions in the dataset
bejaia_rows = arrayfun(@(x) sprintf('Row%d', x), 3:124, 'UniformOutput', false);
sidi_rows = arrayfun(@(x) sprintf('Row%d', x), 128:249, 'UniformOutput', false);

% Separate the datasets by region using row indices
testingData_bejaia = testingData(ismember(testingData.Row, bejaia_rows), :);
testingData_sidi = testingData(ismember(testingData.Row, sidi_rows), :);

% Define YTest_bejaia and YTest_sidi
YTest_bejaia = testingData_bejaia.Classes;
YTest_sidi = testingData_sidi.Classes;

% Define YPred_knn_bejaia and YPred_knn_sidi
YPred_knn_bejaia = YPred_knn(ismember(testingData.Row, bejaia_rows));
YPred_knn_sidi = YPred_knn(ismember(testingData.Row, sidi_rows));

% Define YPred_linear_bejaia and YPred_linear_sidi
YPred_linear_bejaia = YPred_linear(ismember(testingData.Row, bejaia_rows));
YPred_linear_sidi = YPred_linear(ismember(testingData.Row, sidi_rows));

% Calculate accuracy for each region
accuracy_knn_bejaia    =    sum(YPred_knn_bejaia    ==    YTest_bejaia)    /
length(YTest_bejaia);
accuracy_knn_sidi = sum(YPred_knn_sidi == YTest_sidi) / length(YTest_sidi);
accuracy_linear_bejaia   =   sum(YPred_linear_bejaia   ==   YTest_bejaia)   /
length(YTest_bejaia);
accuracy_linear_sidi    =    sum(YPred_linear_sidi    ==    YTest_sidi)    /
length(YTest_sidi);

% Plotting
figure('Position', [100, 100, 1000, 800]);

subplot(4, 2, [1, 2]);
gscatter(testingData_bejaia.(Variable1),       testingData_bejaia.(Variable2),
YTest_bejaia);
xlabel(Variable1);
ylabel(Variable2);
title('Bejaia True Classes (k-NN)');
legend('Location', 'best');
```

```
axis tight;

subplot(4, 2, [3, 4]);
gscatter(testingData_bejaia.(Variable1),       testingData_bejaia.(Variable2),
YPred_knn_bejaia);
xlabel(Variable1);
ylabel(Variable2);
title('Bejaia k-NN Predicted Classes');
legend('Location', 'best');
axis tight;

subplot(4, 2, [5, 6]);
gscatter(testingData_sidi.(Variable1),           testingData_sidi.(Variable2),
YTest_sidi);
xlabel(Variable1);
ylabel(Variable2);
title('Sidi-Bel Abbes True Classes (k-NN)');
legend('Location', 'best');
axis tight;

subplot(4, 2, [7, 8]);
gscatter(testingData_sidi.(Variable1),           testingData_sidi.(Variable2),
YPred_knn_sidi);
xlabel(Variable1);
ylabel(Variable2);
title('Sidi-Bel Abbes k-NN Predicted Classes');
legend('Location', 'best');
axis tight;

% Display k-NN accuracy
annotation('textbox', [0.05, 0.02, 0.9, 0.04], 'String', ['k-NN Accuracy: '
num2str(accuracy_knn)],  'HorizontalAlignment',  'center',  'FontSize',  14,
'FontWeight', 'bold', 'LineStyle', 'none');

% Second figure for linear regression plots
figure('Position', [100, 100, 1000, 800]);

subplot(4, 2, [1, 2]);
gscatter(testingData_bejaia.(Variable1),       testingData_bejaia.(Variable2),
YTest_bejaia);
xlabel(Variable1);
ylabel(Variable2);
title('Bejaia True Classes (Linear Regression)');
legend('Location', 'best');
axis tight;

subplot(4, 2, [3, 4]);
gscatter(testingData_bejaia.(Variable1),       testingData_bejaia.(Variable2),
YPred_linear_bejaia);
xlabel(Variable1);
ylabel(Variable2);
```

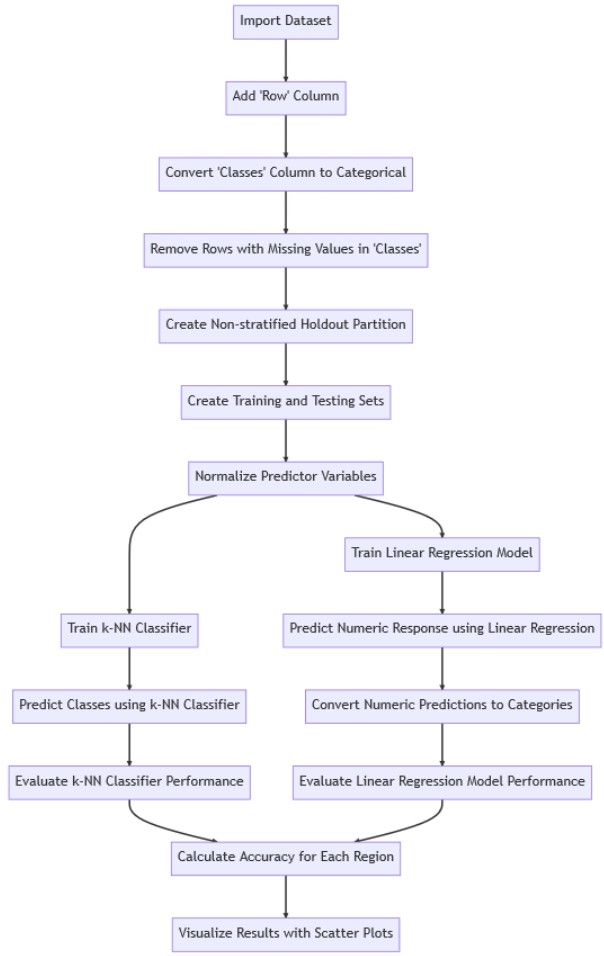
```
title('Bejaia Linear Regression Predicted Classes');
legend('Location', 'best');
axis tight;

subplot(4, 2, [5, 6]);
gscatter(testingData_sidi.(Variable1),           testingData_sidi.(Variable2),
YTest_sidi);
xlabel(Variable1);
ylabel(Variable2);
title('Sidi-Bel Abbes True Classes (Linear Regression)');
legend('Location', 'best');
axis tight;

subplot(4, 2, [7, 8]);
gscatter(testingData_sidi.(Variable1),           testingData_sidi.(Variable2),
YPred_linear_sidi);
xlabel(Variable1);
ylabel(Variable2);
title('Sidi-Bel Abbes Linear Regression Predicted Classes');
legend('Location', 'best');
axis tight;

% Display accuracy
annotation('textbox', [0.05, 0.02, 0.9, 0.04], 'String', ['Linear Regression
Accuracy:   '   num2str(accuracy_linear)],   'HorizontalAlignment',   'center',
'FontSize', 14, 'FontWeight', 'bold', 'LineStyle', 'none');
```

APPENDIX 2: FLOW CHART OR PSEUDOCODE OF EACH CODE

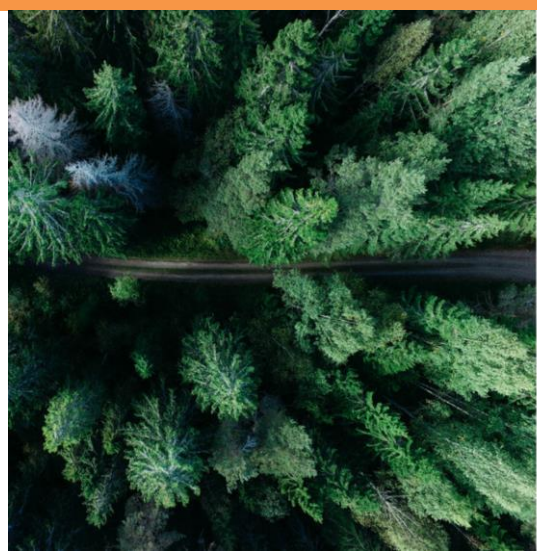


Project Program Flowchart

APPENDIX 3: DLSU GDRIVE OR ONEDRIVE LINK TO THE PRESENTATION AND DEMONSTRATION VIDEO

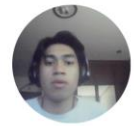
https://drive.google.com/file/d/1tNM41PuABcaA1uxptxTqyJxpXrKSqvqi/view?usp=sharing

APPENDIX 4: PRESENTATION SLIDES



MACHINE LEARNING FOR ALGERIAN FOREST FIRES

CABRERA, MAYUGA, PANGAN



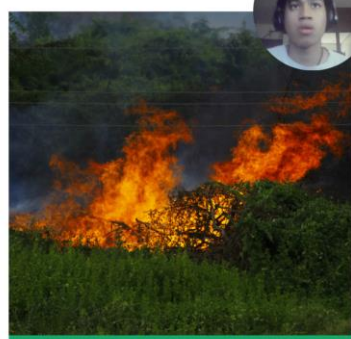
Introduction

Around the world, forest fires are a common and essential component of many ecosystems. However, they can have disastrous repercussions on the environment, wildlife, and human populations when they happen at an unchecked rate. With the loss of biodiversity and habitat devastation, forest fires have long-lasting effects

In recent years there has been an increase in the use of machine learning to effectively and correctly predict forest fires. Due to its capacity to analyze and interpret large data sets, machine learning has been used to anticipate forest fires more and more.

The primary objective of this project is to evaluate the performance of two machine learning models, the k-NN classifier and the linear regression model, for predicting forest fires using the Algerian forest fires dataset. The specific focus of this study is on the use of two predictor variables, Temperature and RH (humidity), as input features for the models

The thesis statement of this project is: "By comparing the performance of the k-NN classifier and the linear regression model for predicting forest fires based on temperature and humidity, this study aims to contribute to the growing body of research on machine learning techniques for forest fire prediction and provide insights into their practical application."



In Brazil almost 1000 major forest fires occurred in the amazon during 2022 fire season (Cuffe, 2022).

DATASET

ALGERIAN FOREST FIRE DATASET

244 instances of regrouped data of the Bejaia and Sidi Bel-abbes region are included in the dataset.

It starts in June 2012 until September that same year.

The attributes are Date, Temperature, Relative Humidity (RH), Wind speed (Ws), total rain in day in mm, Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), Buildup Index (BUI), FIre Weather Index (FWI) and Classes

PICTURE OF DATASET

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Bejaia Region Dataset													
2	day	month	year	Temperat	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
3	1	6	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire
4	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire
5	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire
6	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire
7	5	6	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire
8	6	6	2012	31	67	14	0	82.6	5.8	22.2	3.1	7	2.5	fire
9	7	6	2012	33	54	13	0	88.2	9.9	30.5	6.4	10.9	7.2	fire
10	8	6	2012	30	73	15	0	86.6	12.1	38.3	5.6	13.5	7.1	fire
11	9	6	2012	25	88	13	0.2	52.9	7.9	38.8	0.4	10.5	0.3	not fire
12	10	6	2012	28	79	12	0	73.2	9.5	46.3	1.3	12.6	0.9	not fire
13	11	6	2012	31	65	14	0	84.5	12.5	54.3	4	15.8	5.6	fire
14	12	6	2012	26	81	19	0	84	13.8	61.4	4.8	17.7	7.1	fire
15	13	6	2012	27	84	21	1.2	50	6.7	17	0.5	6.7	0.2	not fire
16	14	6	2012	30	78	20	0.5	59	4.6	7.8	1	4.4	0.4	not fire
17	15	6	2012	28	80	17	3.1	49.4	3	7.4	0.4	3	0.1	not fire
18	16	6	2012	29	89	13	0.7	36.1	1.7	7.6	0	2.2	0	not fire
19	17	6	2012	30	89	16	0.6	37.3	1.1	7.8	0	1.6	0	not fire
20	18	6	2012	31	78	14	0.3	56.9	1.9	8	0.7	2.4	0.2	not fire

Methodology

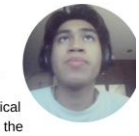
I. Materials

a. Algerian forest fires dataset
b. MATLAB software for data analysis and visualization

II. Instruments

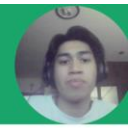
a. Personal Computer
b. MATLAB

III. Implementation



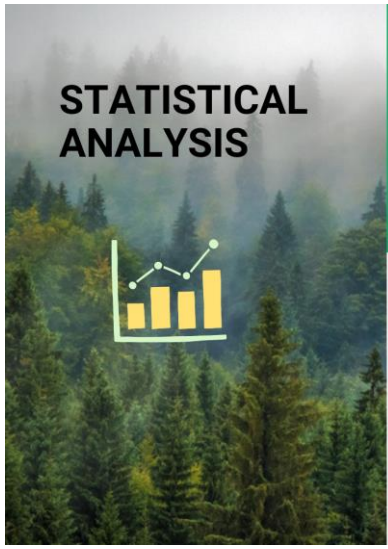
a.Import the dataset into MATLAB and add a 'Row' column to the dataset.
b. Convert the "Classes" column into categorical data and remove rows with missing values in the "Classes" column.
c. Create a non-stratified holdout partition with a holdout ratio of 30%.
d. Create the training and testing sets, separate the predictor variables from the response variable, and normalize the predictor variables.
e. Train the k-NN classifier using the normalized data and predict the classes for the testing set using the normalized data.
f. Train the linear regression model using the normalized data and predict the numeric response for the testing set using the normalized data.
g. Convert the numeric predictions back to the original categories and evaluate the performance of both the k-NN classifier and the linear regression model.
h. Separate the datasets by region using row indices and calculate accuracy for each region.
i. Visualize the results using scatter plots.

Methodology



IV. Evaluation of Correctness

a. The accuracy of the models was calculated to determine their effectiveness in predicting the Classes variable.
b. The confusion matrices were generated to assess the performance of the models for each region.
c. The scatter plots were used to visualize the predicted versus true classes for each region.



The statistical analysis used in this project is classification, specifically k-nearest neighbors (k-NN) and linear regression.

In this project, we used k-NN to classify the forest fires dataset based on its predictor variables such as temperature, relative humidity, wind speed, etc. We chose k-NN because it is a simple yet effective algorithm for classification tasks, and it can handle non-linear decision boundaries.

In this project, we used linear regression to predict the Classes column in the dataset (i.e., the dependent variable) based on its predictor variables. We chose linear regression because it can capture linear relationships between variables and it provides a quantitative measure of the relationship between the independent and dependent variables.

DATA PARSING AND WRANGLING

To ensure accurate interpretation and handling of the dataset, the "Classes" column was converted into categorical data, representing the two classes: 'fire' and 'not fire'. This conversion facilitated the application of machine learning algorithms tailored for categorical data.

Next, a non-stratified holdout partition with a holdout ratio of 30% was used to create separate training and testing sets. This method involved randomly selecting 30% of the dataset as the testing set, while the remaining 70% served as the training set

To minimize the impact of varying scales among predictor variables and facilitate the training process for both k-NN and linear regression models, normalization was applied to the predictor variables.

Model Testing and Hyperparameter Tuning

To ensure the selection of the most effective machine learning models for the task, rigorous testing and comparison were conducted between the k-NN classifier and the linear regression model. This process aimed to identify the best model in terms of prediction accuracy, as well as understand the strengths and weaknesses of each model in different settings.

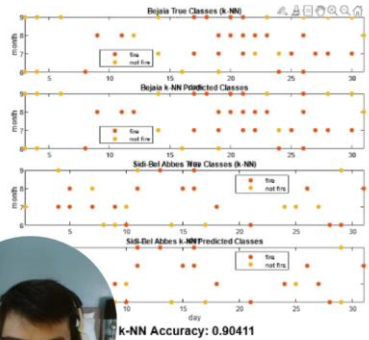
The k-NN classifier was the first model tested. To optimize its performance, hyperparameter tuning was performed to identify the optimal value of 'k', the number of nearest neighbors used for classification. This was achieved through a grid search, in which various values of 'k' were evaluated to determine their effect on the model's accuracy. The best 'k' value was selected based on the highest accuracy achieved on the testing set, ensuring that the model was appropriately tuned for the specific dataset.

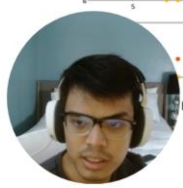
Model Testing and Hyperparameter Tuning

The linear regression model was also tested, with its predicted numeric responses being converted back to the original categorical classes ('fire' and 'not fire'). Although linear regression is not inherently designed for classification tasks, this approach allowed for a comparison of its performance with the k-NN classifier. While the linear regression model did not require hyperparameter tuning like the k-NN classifier, it was crucial to assess its potential as an alternative model for the given task.

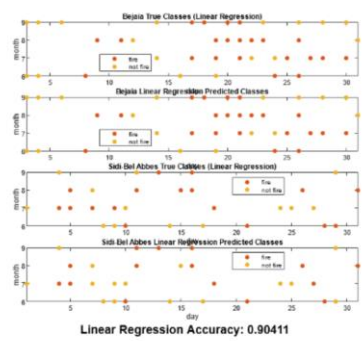
Once the models were appropriately tuned and trained, their performance was evaluated on the testing set. The accuracy of each model was calculated, and confusion matrices were generated to assess their performance in predicting the 'Classes' variable for each region. This information provided valuable insights into the effectiveness of the models and helped identify the most suitable model for the task at hand. Additionally, scatter plots were used to visualize the predicted versus true classes for each region, offering a clear representation of the models' performance in different geographical contexts.

PERFORMANCE MEASURE, ACCURACY SCORE





k-NN Accuracy: 0.90411



Linear Regression Accuracy: 0.90411

PERFORMANCE MEASURE, ACCURACY SCORE

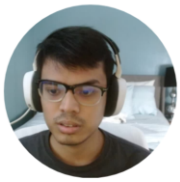
The primary performance measure used to evaluate the effectiveness of the k-NN classifier and the linear regression model was the accuracy score. This metric indicates the proportion of correct predictions made by the models out of the total number of predictions. An accuracy score reaching the target cutoff score set in the project was considered satisfactory and indicative of a well-performing model.

Upon testing, both the k-NN classifier and the linear regression model achieved an accuracy score of 0.90411, or 90.411%, using only two predictor variables: Temperature and RH (humidity). One of the reasons for using only two predictor variables and not implementing techniques like Principal Component Analysis (PCA) was to simplify the model and reduce the computational complexity. This allowed us to focus on the most relevant features for the specific task, making the models more interpretable and easier to understand.

This result demonstrated that both models were capable of predicting the 'Classes' variable ('fire' and 'not fire') with a high degree of accuracy, even with a limited set of predictor variables. The strong performance of both models suggests that they are suitable for the task at hand and meet the project's predefined target. It is worth noting that, although both models achieved the same accuracy score, their underlying mechanisms and assumptions are different. The k-NN classifier relies on the similarity between data points to make predictions, while the linear regression model attempts to model the relationship between the input features and the response variable.

PERFORMANCE MEASURE, ACCURACY SCORE

Given the distinct approaches employed by these models, their performance might differ in other datasets or contexts. However, for this specific project and with the chosen predictor variables, both models have proven to be effective in predicting the 'Classes' variable with a high level of accuracy.



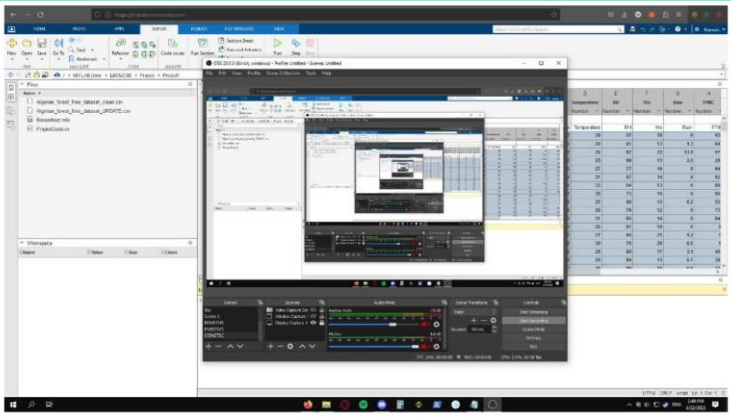
Conclusion

This study aimed to evaluate the performance of two machine learning models, the k-NN classifier and the linear regression model, for predicting forest fires using the Algerian forest fires dataset. The focus was on the use of two predictor variables, Temperature and RH (humidity), as input features for the models.

The results demonstrated that both the k-NN classifier and the linear regression model achieved an accuracy of approximately 90.411% in predicting forest fires based on the chosen predictor variables.

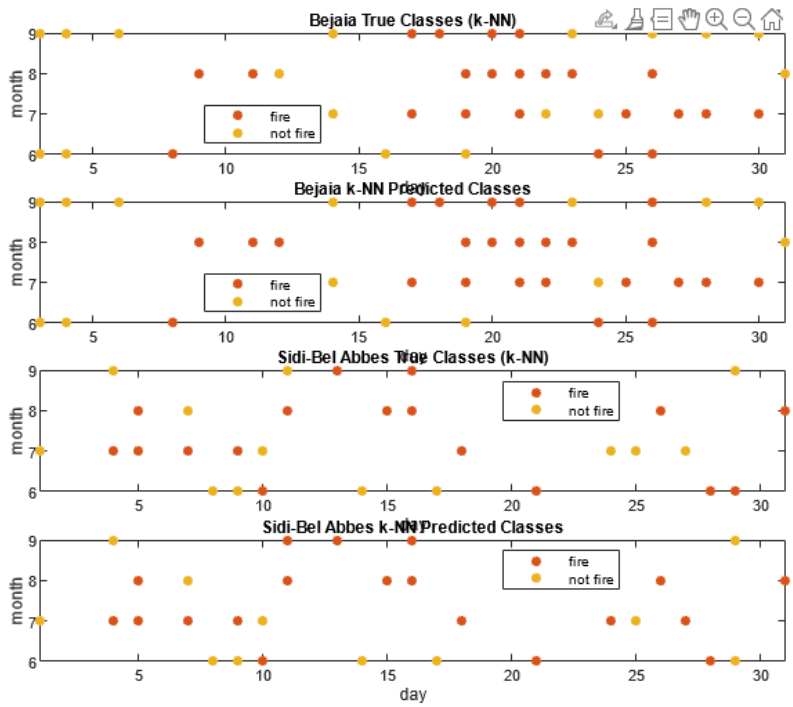
The choice of predictor variables, specifically Temperature and RH, proved to be effective in achieving satisfactory model performance. However, it is important to acknowledge the limitations imposed by the use of only two predictor variables

Demonstration Video



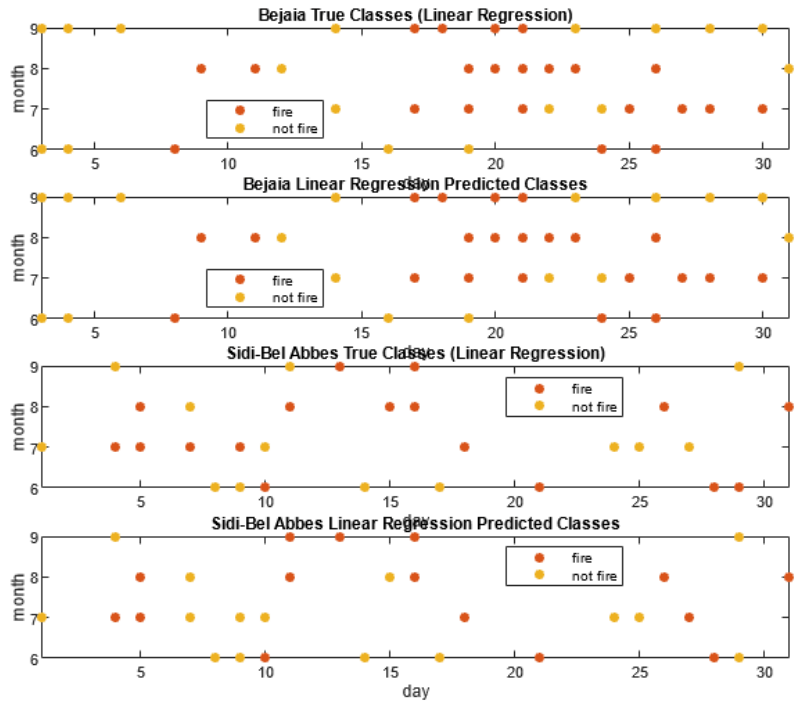
Thank You

APPENDIX 5: PERFORMANCE MEASURES, ACCURACY SCORES



k-NN Accuracy: 0.90411

k-NN Plot



Linear Regression Accuracy: 0.90411

Linear Aggression Plot