**Air Quality Analysis using Machine Learning Algorithms**

(Final)

**Machine Learning Models**

1. **General**

Machine Learning is the ability of machines to learn, where a machine is built up using certain algorithms through which it can take its own decisions and provide the result to the user. Basically, it is considered the subfield of Artificial Intelligence [1]. Today Machine Learning is used for complex data classification and decision making. In simple terms it is the development of algorithms that enables the system to learn, and to make necessary decisions. It has strong ties to mathematical optimization that delivers methods, theory, and application domain to the field and, it is employed in a range of computing tasks where designing and programming explicit algorithms is infeasible [2]. In our project, three models were used, which are as follows:

1. **Decision Tree**.
2. **Definition**. Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems. It is called a decision tree because, like a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. To build a tree, the CART algorithm is popular , which stands for Classification and Regression Tree algorithm. A decision tree can contain categorical data as well as numeric data [2]. Decision Trees usually mimic human thinking ability while deciding, so it is easy to understand.
3. **Specifications**. The model’s name DecisionTreeRegressor. It is derived from the popular library SkLearn. SkLearn or Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects.

c. **Methods**. They work as follows:

* Data Pre-processing step
* Fitting a Decision-Tree algorithm to the Training set
* Predicting the test result
* Test accuracy of the result(Creation of Confusion matrix)
* Visualizing the test set result.

1. **Expectations**.

The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It is expected that differences between training score and test score is minimal. There should be linear relationship between expected and predicted.

3. KNN.

1. **Definition**. The abbreviation KNN stands for “K-Nearest Neighbour”. It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbours to a new unknown variable that must be predicted or classified is denoted by the symbol 'K'. The k value is the core of the algorithm. The value of a data point is determined by the data points around it. The number of data points that are taken into consideration is determined by the k value. Thus, the k value is the core of the algorithm [3]. KNN classifier determines the class of a data point by the majority voting principle.
2. **Specification**. The model’s name is KNeighborsClassification. It is derived from the popular library SkLearn. SkLearn or Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects.
3. **Methods.** Steps to implement the K-NN algorithm:

* Data Pre-processing step
* Fitting the K-NN algorithm to the Training set
* Predicting the test result
* Test accuracy of the result(Creation of Confusion matrix)
* Visualizing the test set result.

1. **Expectations.** The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It is expected that differences between training score and test score is minimal. There should be linear relationship between expected and predicted.
   1. **Random Forest.**
2. **Definition**. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

1. **Specifications.** The model’s name is “RandomForestClassifier”. It is derived from the popular library SkLearn, SkLearn or Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects.
2. **Methods.** Implementation Steps are given below:

* Data Pre-processing step
* Fitting the Random Forest algorithm to the Training set
* Predicting the test result
* Test accuracy of the result (Creation of Confusion matrix)
* Visualizing the test set result.

1. Expectations**.** It enhances the accuracy of the model and prevents the overfitting issue.

**Result**

* 1. **Decision Tree.**

1. **Checking the overfit**. It is expected that difference between training and resting score to be minimal, Testing score found to be 81% and testing score found to be 71%. Differences to is moderate. Scores from the code shown below in Table 1 :

Table 1: Score of train and test

|  |  |
| --- | --- |
| Items | Score |
| Training set | 0.8109409420137816 |
| Testing set | 0.7159876620671785 |

1. **Decision Tree Model Evaluation.** Mean Squared Error (2345) is far more than zero. Mean absolute error also nearer to zero. R2 score is 71%. Model run can be termed as moderate.

Table 1: Error values

|  |  |  |
| --- | --- | --- |
| Errors | Value | Remarks |
| Mean Squared Error | 2345.1841746259192 | Should be as less as possible |
| Mean Absolute Error | 34.206046867823986 | Should be as less as possible |
| R2 Score | 0.7159876620671785 | Should be nearer to 1 |

1. **Evaluation by visualization using bell curve.**

Predicted values away from bell curve boundary which is as follows:

Histogram

Description automatically generated

Figure 1. Evaluation of output

1. **Evaluation by visualization using bell curve**.

In the following image, expected, and predicted value do not match.

Chart, scatter chart

Description automatically generated

Figure 2. Relationship between expected and predicted.

* 1. **KNN**

1. **Selection of K value by training.** Model is trained using n\_neighbors = 2, as it is maximum.

Table 3: Accuracy with standard deviation

|  |  |  |  |
| --- | --- | --- | --- |
| K | Mean Accuracy | Standard deviation | Remarks |
| 2 | 74.36% | 0.066 | Maximum |
| 3 | 72.29% | 0.049 |  |
| 4 | 69.36% | 0.04 | Minimum |
| 5 | 70.33% | 0.03 |  |
| 6 | 70.45% | 0.045 |  |

1. **Mismatched prediction**

Total number of mismatched predictions is 53, which are shown below:

[('A', 'B'), ('B', 'C'), ('B', 'C'), ('C', 'B'), ('A', 'B'), ('B', 'A'), ('B', 'A'), ('A', 'B'), ('B', 'C'), ('C', 'B'), ('A', 'C'), ('B', 'A'), ('B', 'C'), ('B', 'C'), ('A', 'B'), ('B', 'C'), ('A', 'C'), ('A', 'B'), ('B', 'A'), ('B', 'C'), ('B', 'C'), ('B', 'C'), ('A', 'C'), ('C', 'B'), ('B', 'A'), ('B', 'A'), ('C', 'B'), ('A', 'B'), ('C', 'B'), ('A', 'C'), ('A', 'B'), ('B', 'A'), ('C', 'B'), ('A', 'C'), ('A', 'B'), ('B', 'A'), ('B', 'A'), ('A', 'B'), ('C', 'B'), ('A', 'B'), ('B', 'C'), ('C', 'B'), ('C', 'B'), ('B', 'C'), ('A', 'B'), ('A', 'B'), ('C', 'B'), ('B', 'A'), ('B', 'C'), ('A', 'B'), ('B', 'C'), ('A', 'B'), ('C', 'B')]

1. **Confusion matrix.** In the following output, we have seen that there were some incorrect predictions, so for knowing the number of correct and incorrect predictions, we need to use the confusion matrix.

Chart

Description automatically generated

Figure 3: Confusion matrix of KNN model

In the above image, we can see there are 91+56+74= 221 correct predictions and 14+5+13+10+11= 53 incorrect predictions. So, we can say that the performance of the model needs to be improved.

1. **Classification report .** Accuracy of model performance is 81%, which is as follows:

precision recall f1-score support

A 0.83 0.90 0.86 101

B 0.71 0.69 0.70 81

C 0.87 0.80 0.84 92

accuracy 0.81 274

macro avg 0.80 0.80 0.80 274

weighted avg 0.81 0.81 0.81 274

* 1. **Random Forest**.

1. **Model evaluation by mismatched Prediction**. Total number of mismatched predictions is 34, which are shown below:

[('C', 'A'), ('B', 'C'), ('C', 'B'), ('A', 'B'), ('B', 'A'), ('B', 'A'), ('B', 'C'), ('C', 'B'), ('C', 'A'), ('B', 'A'), ('B', 'C'), ('A', 'B'), ('B', 'A'), ('A', 'B'), ('B', 'C'), ('B', 'A'), ('B', 'A'), ('C', 'B'), ('B', 'A'), ('A', 'B'), ('A', 'B'), ('C', 'B'), ('C', 'B'), ('B', 'C'), ('B', 'A'), ('A', 'B'), ('A', 'C'), ('B', 'A'), ('C', 'A'), ('A', 'B'), ('C', 'B'), ('A', 'B'), ('C', 'B'), ('C', 'B'), ('B', 'A'), ('B', 'A'), ('B', 'C'), ('A', 'B')]

1. **Confusion matrix.** In the following output, we have seen that there were some incorrect predictions, so for knowing the number of correct and incorrect predictions, we need to use the confusion matrix.

A screenshot of a cell phone

Description automatically generated with low confidence

Figure 4: Confusion matrix of Random Forest

In the above image, we can see there are 88+67+85 = 240 correct predictions and 6+6+1+11+2+8= 34 incorrect predictions., whereas, in KNN, there were 11 incorrect predictions. So, we can say that the performance of the model is improved by using the random forest algorithm.

1. **Classification report**. Accuracy of model performance is 86%, which is as follows:

precision recall f1-score support

A 0.90 0.86 0.88 101

B 0.79 0.79 0.79 81

C 0.89 0.92 0.90 92

accuracy 0.86 274

macro avg 0.86 0.86 0.86 274

weighted avg 0.86 0.86 0.86 274

**Discussions**

* 1. **Comparative analysis of correct and incorrect prediction**.

Table 4: Comparative prediction vector

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Correct prediction | Incorrect prediction | Remarks |
|  |  |  |  |
| Decision Tress | Training set score: 0.8109409420137816 | Testing set score: 0.7117912269016089 | Difference moderate |
| KNN | 221 | 53 | Overfitting |
| Random Forest | 240 | 34 | Overfittings minimize |

* 1. **Accuracy.**

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **Remarks** |
| Decision Tree | 71% | Regression |
| KNN | 81% | Classification |
| Random Forest | 88 % | Classification |

**Conclusion**

* 1. Air quality prediction will help the researcher to foresee the environmental degradation. It will give a wakeup call to the decision maker. Accordingly, we will be able to make the environment livable to the future generation.

**References**

1. Bhalgat, P., S. Bhoite, and S. Pitare, Air quality prediction using machine learning algorithms. International Journal of Computer Applications Technology and Research, 2019. 8(9): p. 367-390.

2. Iskandaryan, D., F. Ramos, and S. Trilles, Air quality prediction in smart cities using machine learning technologies based on sensor data: a review. Applied Sciences, 2020. 10(7): p. 2401.

3. Madan, T., S. Sagar, and D. Virmani. Air Quality Prediction using Machine Learning Algorithms–A Review. in 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). 2020. IEEE.