A picture containing clipart

Description automatically generated

**RIGA TECHNICAL UNIVERSITY**

**Faculty of Computer Science and Information Technology**

**Department of Artificial Intelligence and Systems Engineering**

**Fundamentals of Artificial Intelligence**

**PRACTICE #2 REPORT**

Student: Fatima Karimova (201ADB002)

**DESCRIPTION OF THE DATASET**

The dataset is created the aim of classification of the types of glass, due to criminological investigation. “At the scene of the crime, the glass left can be used as evidence”, because of the reason is mentioned above data of the different kind glass has been collected and made as dataset for further use.

**Source**

* **Creator:**

B. German

Central Research Establishment

Home Office Forensic Science Service

Aldermaston, Reading, Berkshire RG7 4PN

* **Donor:**

Vina Spiehler, Ph.D., DABFT

Diagnostic Products Corporation

(213) 776-0180 (ext 3014)

* **Attribute Information:**

Id number: 1 to 214

RI: refractive index

Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)

Mg: Magnesium

Al: Aluminum

Si: Silicon

K: Potassium

Ca: Calcium

Ba: Barium

Fe: Iron

Type of glass: (class attribute)

1 building\_windows\_float\_processed

2 building\_windows\_non\_float\_processed

3 vehicle\_windows\_float\_processed

4 vehicle\_windows\_non\_float\_processed (none in this database)

5 containers

6 tableware

7 headlamps

**PART 1**

Dataset has been collected from the *UCI Machine Learning Repository: Glass Identification Data Set*

<https://archive.ics.uci.edu/ml/datasets/Glass+Identification>. The dataset includes 214 instances, 10 attributes as total with no missing data. However only 9 instances is going to used, the skipped attribute is ID column. Rest columns are included chemicals such as Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium, Iron, as type of glass such as building, and vehicle window’s float or non-float processed.

Each class (1,2,3,5,6,7) includes 70,76,17,13,9,29 records respectively. As it seems from the numbers it is not well-balanced dataset, additionally there is no record regarding to 4th group of glass which was vehicle window’s non-float glass.

All features are represented as numeric data type, there are no any string, boolean or other data types. (Figure 1)

A picture containing graphical user interface

Description automatically generated

**Figure 1, the Dataset which is used in assignment**

The dataset (Figure 1) was in .csv format, which is why no needed to additional adjustments such as arranging labels, data objects. The values which are provided in the file were numerical, there was not any yes/no, positive/neutral/negative etc. values.

The dataset includes different kind of chemicals which are used production of glasses, due to this issue scatter plot for each of them is not appropriate to display however you can check the Scatter plot, Histogram and Distribution of Iron (Fe), aluminum (Al) respectively graphs for identifying the separability of the classes.

**SCATTER PLOT:**

Chart

Description automatically generated

**A picture containing timeline

Description automatically generated**

**Figure 2, Scatter plot of Iron**

**Figure 3, Scatter plot of Aluminum**

**HISTOGRAMS:**

Chart

Description automatically generated

**Chart, histogram

Description automatically generated**

**Figure 4, Histogram of Aluminum**

**Figure 5, Histogram of Iron**

Chart, histogram

Description automatically generated**DISTRUBUTION:**

**Chart, histogram

Description automatically generated**

**Figure 7, Exponential Distribution of Iron**

***Figure 6, Normal Distribution of Iron***

**STATISTICS OF DATASET:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Distribution** | **Mean** | **Median** | **Dispersion** | **Min.** | **Max.** | **Missing** |
| **R\_I** | Bar chart  Description automatically generated with medium confidence | **1.5183** | **1.5176** | **0.0019** | **1.5111** | **1.5339** | **0 (0%)** |
| **Na** |  | **13.407** | **13.30** | **0.0608** | **10.73** | **17.38** | **0 (0%)** |
| **Mg** |  | **2.6845** | **3.48** | **0.5360** | **0.00** | **4.49** | **0 (0%)** |
| **Al** |  | **1.4449** | **1.36** | **0.3447** | **0.29** | **3.50** | **0 (0%)** |
| **Si** |  | **72.650** | **72.79** | **0.0106** | **69.81** | **75.41** | **0 (0%)** |
| **K** |  | **0.4971** | **0.5550** | **1.3090** | **0.00** | **6.21** | **0 (0%)** |
| **Cl** |  | **8.9570** | **8.60** | **0.1585** | **5.43** | **16.19** | **0 (0%)** |
| **Br** |  | **0.1750** | **0.00** | **2.8339** | **0.00** | **3.15** | **0 (0%)** |
| **Fe** |  | **0.0570** | **0.00** | **1.7052** | **0.00** | **0.51** | **0 (0%)** |
| **ToG** | Chart, bar chart  Description automatically generated | **2** | **1.51** | **-** | **-** | **-** | **0 (0%)** |

As it seems from the given dataset, the records of 2nd type of glass is prevailing at the same time there are very less record regarding 6th type of glass compared with other types. Additionally, as it seems there are not any records for 4th type of glass. Due to these reasons, we can conclude that, dataset is not well-balanced.

The diagrams clearly shows that all data objects (chemicals) are clearly separable. It is visible from scatter plot. The graph allows us to see how much iron and aluminum has been identified in each type of glass, and it. As it is mentioned before it is not possible to put all graph results here due to size and visuality.

There are not merged data objects, it is divisible from the other features and their values.

From the point of difference between the data groupings, there are exceptions based on lack of record, some of them are differs from each other as the amount which is indicated.

PICTURE 12

Description automatically generatedFrom the analysis of the statistical calculations, mean value of the silicon and sodium differs very much from others, as well as their min. and max. values. However highest dispersion value belongs to iron, that means, iron identified in many of the glass making process as chemical however its median value is equal to zero. Although, barium’s median is equal to zero as well. In this case the highest value is belongs to silicon with 72.79%, that measure of central tendency for skewed distribution. As it is mentioned before, there are no missing data in dataset, that was helpful as well for as much as possible accurate results.

**PART II**

* **Hierarchical clustering, Experiment I (Figure 6)**

*Parameters:*

*Distance:* Euclidian

*Linkage* – Complete

*Annotations* – Type\_Of\_Glass

*Pruning* - Max depth 10

*Selection height ratio* – 44.0%

**Figure 6**

As it seems from the graph (Figure 6), there are 10 cluster as total. Graph depicts that, most of the 1st, 2nd and 3rd type of glasses have more similar components, as well as 7th and 6th type. At the same time the biggest cluster belongs to cluster number 7, which is include 1st, 2nd, and 3rd glass type. In comparison, the smallest cluster is number 4 which include only 6th type of glass.

The distance of the clusters is similar to each other after slicing, except c1, c5 and c9. Those clusters have been identified with less data compared with other clusters, due to reason, after other clusters has been made, the have added to other clusters as well.

**Chart

Description automatically generated with medium confidenceExperiment II (Figure 7)**

*Parameters:*

*Distance:* Euclidian

*Linkage* – Complete

*Annotations* – Potassium

*Pruning* – Max depth 5

*Selection* – Top N: 3

The graph depicts, there are 3 main cluster with amount of Potassium. And it is obvious that, most of the glass types of potassium amount range is 0.06% and 0.81% in biggest cluster. However, the distances are like each other in clusters. The clusters of potassium, indicates that the main amount of potassium that has been found is roughly 30-30%.

**Figure 7**

A picture containing diagram

Description automatically generated**Experiment III (Figure 8)**

*Parameters:*

*Distance:* Manhattan

*Linkage* – Average

*Annotations* – Aluminum

*Pruning* – Max depth 11

*Selection* – Top N: 3

The graph shows, the clusters of aluminum with distance used Manhattan, so as it depicted the similarity of the amount is greater except cluster number 3. However, if we try to slice (Figure 9)the cluster with height ration 37% in order to see the more clearly which clusters are gathered, we can observe that there is main 5 cluster has made the biggest portion of the cluster number 3

**Figure 8**

Chart

Description automatically generatedThe distance between the clusters are acceptable, because of amounts of the aluminum is close to each other, with small difference. Still, the smallest clusters with highest distance are c1 and c2 where their indicated values are 3.04% and 3.02% respectively.

**Experiment IV (Figure 10)**

*Parameters:*

*Distance:* Hamming

**Figure 9**

*Linkage* – Average

*Annotations* – Sodium

*Pruning* – Max depth 5

*Selection* – Top N: 4

In this experiment I used Hamming distance in order to see difference between calculating the distances of the data objects. As we know from hamming distance that, the distance is greater the similarity of data objects is more. So, as it displayed from the graph (Figure 10)that there is great distance between all clusters that means as other experiments that most of type of glasses amount of potassium are similar to each other. Additionally, if we slice the clusters, with the height ratio of 92% we can see all clusters more clearly and define the similarity of data objects.

**Graphical user interface, application

Description automatically generated**Graphical user interface, text, application

Description automatically generated

**Figure 10**

* **K-means Experiment I (Figure 11)**

*Parameters:*

*Number of clusters : from 2 to 7*

*Preprocessing: normalize columns*

*Initialize: Initialize with KMeans++*

*Re-runs: 10*

*Maximum iteration 300*

**Figure 11**

*Graphical user interface, application

Description automatically generated***Experiment II (Figure 12)**

*Parameters:*

*Number of clusters : from 3 to 9*

*Preprocessing: normalize columns*

*Initialize: Initialize with KMeans++*

*Re-runs: 10*

*Maximum iteration 300*

**Figure 12**

**Graphical user interface, text, application

Description automatically generatedExperiment III (Figure 13)**

*Parameters:*

*Number of clusters : from 7 to 9*

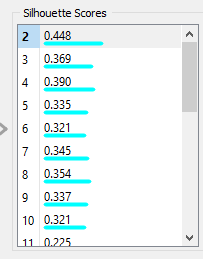
*Preprocessing: normalize columns*

*Initialize: Initialize with KMeans++*

*Re-runs: 10*

*Maximum iteration 300*

**Figure 13**

** **Experiment IV (Figure 14 and 15)**

*Parameters:*

*Number of clusters : from 2 to 25*

*Preprocessing: normalize columns*

Chart, funnel chart

Description automatically generated *Initialize: Random initialization*

*Re-runs: 50*

*Maximum iteration 300*

**Figure 14**

**Figure 15**

Graphical user interface, text, application

Description automatically generated

**Experiment V (Figure 16 )**

*Parameters:*

*Number of clusters : from 12 to 14*

*Preprocessing: normalize columns*

*Initialize: Random initialization*

*Re-runs: 50*

*Maximum iteration 300*

**Figure 16**

**Experiment VI (Figure 17 )**

*Graphical user interface, application

Description automatically generated Parameters:*

*Number of clusters : from 15 to 24*

*Preprocessing: normalize columns*

*Initialize: Random initialization*

*Re-runs: 50*

*Maximum iteration 300*

**Summary of experiments:**

**Figure 17**

As we know that K-means algorithm helps to minimize the distances between data objects inside the clusters and their centroids (ref-study materials) and when the score of the silhouette is close to 1, which is between -1 and 1, is better value. Due to these reasons, 6experiment has been conducted with different parameters. Starting from first experiment, it was one of the hopeful experiments with the silhouette scores, the score was not much high that as expected, but it was 0.419, which is higher than rest 4 other trials. The cluster numbers were from 2 to 7 and re-run amount was 10. Additionally, 4th experiment was quite successful one as well, with the cluster numbers from 2 to 25 and re-run was equal to 50. This experiment was the successful one as it seems from graph (see figure 14) with the 0.448 score. The experiment user random initialization. Coming to the worst experience that gave lowest scores was last experiment (see figure 17). The cluster numbers were highest among other experiments, from 15 to 24, with re-runs 50.

To sum up, from conducted experiments and depicted results we can say, in order to get high score cluster number, initialization and re-run plays important role. Based on the successful results, random initialization helps to increase the score which can lead to creating wrong cluster in the dataset. It can be one of the drawbacks of it. However high re-run value is helpful to re generate centroid of the cluster and which can give successful results.

**PART III Experiment I (Table 2)**

* **Linear regression**

Parameters :

Fit intercept (underchecking it fixes it to zero)

*Regularization:* No regularization

* **kNN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | ***MSE*** | ***RMSE*** | ***MAE*** | ***R2*** |
| ***KNN*** | *1.821* | *1.349* | *0.765* | *0.637* |
| ***ANN*** | *2.055* | *1.433* | *0.985* | *0.591* |
| ***LINEAR R*** | *2.268* | *1.506* | *1.059* | *0.548* |

Parameters:

*Neighbors:* Number of neighbors 3

*Metric:* Ecliudean

*Weight:* Uniform

* **Neural network**

Parameters:

*Neurons in hidden layers* 100

*Activation:* ReLu

**Table 2**

*Solver:* Adam

*Regularization*, a=0.0001

*Maximal number of iterations:* 200

*Replicable training*

* **Test and Score**

Cross validation: Number of folds – 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | ***MSE*** | ***RMSE*** | ***MAE*** | ***R2*** |
| ***KNN*** | *1.970* | *1.404* | *0.826* | *0.608* |
| ***ANN*** | *2.000* | *1.414* | *0.967* | *0.602* |
| ***LINEAR R*** | *2.059* | *1.435* | *0.998* | *0.590* |

**Experiment II (Table 3)**

* **Linear regression**

Parameters :

Fit intercept (underchecking it fixes it to zero)

*Regularization:* Ridge regression (L2)

*Alpha:* 0.1

* **kNN**

Parameters:

*Neighbors:* Number of neighbors 5

*Metric:* Manhattan

*Weight:* Uniform

* **Neural network**

Parameters:

*Neurons in hidden layers* 200,100,50

**Table 3**

*Activation:* ReLu

*Solver:* Adam

*Regularization*, a=0.0005

*Maximal number of iterations:* 210

*Replicable training*

* **Test and Score**

Cross validation: Number of folds – 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | ***MSE*** | ***RMSE*** | ***MAE*** | ***R2*** |
| ***KNN*** | *1.971* | *1.404* | *0.892* | *0.607* |
| ***ANN*** | *3.081* | *1.755* | *1.414* | *0.386* |
| ***LINEAR R*** | *2.091* | *1.446* | *1.007* | *0.583* |

**Experiment III (Table 4)**

* **Linear regression**

Parameters :

*Fit intercept (underchecking it fixes it to zero)*

*Regularization:* Ridge regression (L2)

*Alpha:* 0.05

* **kNN**

Parameters:

**Table 4**

Table

Description automatically generated*Neighbors:* Number of neighbors 10

*Metric:* Ecliudean

*Weight:* Distance

* **Neural network**

Parameters:

*Neurons in hidden layers* 50,10,5

*Activation:* Logistic

*Solver:* Adam

*Regularization*, a=0.0008

*Maximal number of iterations:* 190

*Replicable training*

* **Test and Score**

Cross validation: Number of folds – 5

**Figure 18, fragment of predictions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | ***MSE*** | ***RMSE*** | ***MAE*** | ***R2*** |
| ***KNN*** | *2.114* | *1.454* | *0.897* | *0.579* |
| ***ANN*** | *2.210* | *1.487* | *1.021* | *0.560* |
| ***LINEAR R*** | *1.907* | *1.381* | *0.948* | *0.620* |

**Experiment IV (Table 4)**

* **Linear regression**

Parameters :

Fit intercept (underchecking it fixes it to zero)

*Regularization:* Lasso regression (L1)

*Alpha:* 0.05

* **kNN**

**Table 5**

Parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | ***MSE*** | ***RMSE*** | ***MAE*** | ***R2*** |
| ***KNN*** | *1.400* | *1.183* | *0.726* | *0.721* |
| ***ANN*** | *1.331* | *1.154* | *0.827* | *0.735* |
| ***LINEAR R*** | *1.468* | *1.468* | *0.827* | *0.708* |

*Neighbors:* Number of neighbors 8

*Metric:* Manhattan

*Weight:* Uniform

* **Neural network**

Parameters:

*Neurons in hidden layers* 75,4,2

*Activation:* Identity

*Solver:* Adam

*Regularization*, a=0.005

*Maximal number of iterations: 175*

**Table 6, Test on train data results**

*Replicable training*

* **Test and Score**

Cross validation: Number of folds – 5

**Experiment V (Table 6)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | ***MSE*** | ***RMSE*** | ***MAE*** | ***R2*** |
| ***KNN*** | *0.809* | *0.900* | *0.529* | *0.839* |
| ***ANN*** | *0.670* | *1.292* | *0.913* | *0.667* |
| ***LINEAR R*** | *1.293* | *1.137* | *0.849* | *0.743* |

* **Linear regression**

Parameters :

Fit intercept (underchecking it fixes it to zero)

*Regularization:* No regularization

* **kNN**

Parameters:

*Neighbors:* Number of neighbors 3

*Metric:* Ecliudean

**Table 6**

Table

Description automatically generated*Weight:* Uniform

* **Neural network**

Parameters:

*Neurons in hidden layers* 100,5,3

*Activation:* Identity

*Solver:* Adam

*Regularization*, a=0.05

*Maximal number of iterations:* 100

*Replicable training*

* **Test and Score**

Cross validation: Number of folds – 5

**Figure 19, fragment of predictions**

First of all, for this experiment 3 main supervised algorithm has been selected, Linear regression, kNN and Neural network.

The reason of choosing the linear regression and kNN algorithms in order to see best possible fits for my dataset. Because data set is big enough to give wrong predictions. Additionally, as we know kNN algorithm is supports non-linear solutions as well, so it gives different insight about the experiment result as well. Comping to the experiments and its results, the experiment was tricky because of I decided to do all 3 experiments for each algorithm all together, at the end from my opinion I conducted good experiments with different parameters.

For these experiments I choose type of glass as target value because I wanted to predict the type of glasses based on other chemicals. Because the main aim of this data set and supervised algorithms are to predict. For this goal, the experiments and its results were pretty interesting.

We know from the previous lecture notes and professor’s explanations, that main results that we must notice are RMSE and R2 , which form RMSE must give smaller error value, for R2 the best score which is close to 1.

Starting from that in all experiments kNN and Linear regression results were competing as it seems from the results (tables). But it was obvious that kNN algorithm was depicting more accurate results that we can diffidently say that 1st,2nd , 3rd and 5th experiment results that kNN algorithm showed better values. Coming to the 3rd experiment that we can see the predictions of the algorithms (see figure 18 ) that predictions are not satisfactory, although from scores (see table 4) kNN has more satisfactory scores. However, the predictions of algorithms including kNN was not correct. So, it is the result of the parameters as well. Because of the linear regression alpha value, for kNN the neighbor number was high as well. Coming to the best Neural network (ANN) results were in 4th experiment where I did test in train data (see table 6) that results were more higher comparing to other experiments as total. The scores of ANN were better than other algorithms in those experiments.

``coming to the last experiment that I have conducted for this part, the parameters were more acceptable for those results. As previous experiments that I can say kNN had best results as before, additionally the prediction values were accurate as well. Although, other algorithms predictions were satisfactory either comparing with previous predictions. This time the I decide to put lower, kind of default hypermeters for each of algorithm and experiment resulted best values.

To sum it up these experiments in this part, I can say it was tricky to observe the differences, however Orange tool was easy to conduct all experiments with different parameters and see immediate result. As it seems from the tables and scores kNN algorithm was best suit among other algorithms for my dataset. Because of the predictions were accurate most of times, and scores were more satisfactory.

Table

Description automatically generated**Additional Experiment I (Table 7)**

Parameters

* **kNN**

*Neighbors* : Number of neighbors -3

*Metric* : Ecliudean

*Weight*: Uniform

* **Logistic Regression**

**Table 7**

Regularization type : Ridge (L2)

Strength : 1

**Experiment II (Table 8)**

* **kNN**

Table

Description automatically generated*Neighbors* : Number of neighbors -5

*Metric* : Manhattan

*Weight*: Uniform

* **Logistic Regression**

Regularization type : Ridge (L2)

Strength : 3

**Experiment III (Table 9)**

**Table 8**

* **kNN**

*Neighbors* : Number of neighbors -2

*Metric* : Euclidean

Text, table

Description automatically generated*Weight*: Distance

* **Logistic Regression**

Regularization type : Ridge (L2)

Strength : 0.700

**Table 9**

**A picture containing calendar

Description automatically generated**

In this part of task, I have conducted 3 other algorithms, however based on last experiment kept kNN algorithm to analyze more.

The algorithms were kNN, Naïve Bayes and Logistic Regression. The reason of why I have chosen those algorithms in this part of task is to predict that can we assign certain chemicals while classify the types of A picture containing calendar

Description automatically generatedglass. Logistic regression is used for binary classification as everyone knows, but I wanted to test this in my data in order to check the results and understand working way of algorithm.

**Table 10, Prediction of kNN**

Table

Description automatically generated with low confidenceComing to results, I have learned from the practical classes when we are analyzing these results we need to pay attention to CA and F1 results more in order to say something satisfactory. The rule was, The highest value of CA and F1 is the best result.

**Table 11,prediction of Naïve Bayes**

From last experiment results we can say that in this experiment kNN algorithm shows accurate and successful results, as other 2 experiments. Based on the result of kNN algorithm, we can say that better scores come from low neighbor number. When we increase the value over 3 the score increases dramatically.

**Table 12, prediction of Logistic Regression**

We can analyze the given confusion matrix as well, that we see (see table 10) for example to take 3rd type of glass : **kNN**

False Positive (FP) – 10 True Positive (TP) - 5

False Negative (FN) – 12 True Negative (TN) – 78

**Naïve bayes:**

False Positive (FP) – 19 True Positive (TP) - 4

False Negative (FN) – 12 True Negative (TN) – 62

**Logistic Regression:**

False Positive (FP) – 0 True Positive (TP) - 0

False Negative (FN) – 16 True Negative (TN) – 76

From these results we can say that kNN algorithm is predicted only true classified ones are greater than false negative (22<83), so this result is satisfactory for my dataset that kNN gave better results compared the result of other algorithms. We can say that logistic regression was the least algorithm for using for this dataset, however the predictions of 2 algorithm were over my expectation still. But it is crystal clear that kNN shows best results for previous (Part III) experiments as well as current one.

Moreover, in 2nd experiment, Logistic regression showed satisfactory score as well after kNN algorithm. However, it is obvious from the first 2 experiment that logistic algorithm does not give those reliable solutions, since my dataset is not that much suitable for this. But with this experiment I learned that logistic regression is more suitable for the more binary classification predictions such as 0/1 or true/false.

To sum up this practical task, firstly, I can say it was so interesting to work with Orange tool for Machine learning. Which we can test and use different kind of algorithms and see immediate results for our datasets. Another thing that I really like from this practice to see how it differs that your expected values from the real simulation scores. Comparing and analyzing the results were quite tricky too.

For this practice task I used dataset of different kind of glass and the ingredients that has been used while creating the glasses. Luckily, there were not any missing data for recover, however in Orange tool there is that functionality that to recover or refill the missing data as well, which I think so good and helpful.

From my point of view this task taught me to work with algorithms in practice more and see the results and analyze those scores. Based on this knowledge and practice I got; I will use that knowledge in the future as much as I can.

For accessing the orange tool document and .csv file click [here](https://github.com/kaarimovva/PRACTICAL_2).

(<https://github.com/kaarimovva/PRACTICAL_2>)