Assignment 4: Classification in Python

Alejandro Valdés Valdés - A00999044, Patricia Angelli Sosa Silva - A01175571, and Jesús Martínez Eguiarte - A00824164

Tecnolgico de Monterrey

Abstract. Supervised machine learning consists of the application of algorithms that process supplied external instances to produce a model that could predict future cases. Computational models have been applied in order to solve classification probes of various natures where the resources of people would take longer and probably less efficient. In this paper the application of the Decision Tree and k-Nearest Neighbor classification were applied and analyzed by comparing their accuracy and running time of classification. The datasets that were classified were: post-operative recovery place for a patient, contraceptive method of choice, and cover type. The k-NN algorithm showed a higher accuracy performance than the decision tree, but this was with a higher computational running time.

Keywords: classification, decision tree, k-NN

1 Introduction

Supervised machine learning consists of the application of algorithms that reason supplied instances to produce a hypotheses that could predict future cases. Its goal is to generate a model that can predict the class of an instance by the composition of predictor features. This model is called a classifier. This paper shows the application of two classification techniques to different data sets: decision tree and k-Nearest Neighbor (k-NN).

A decision tree is a logical learning method that classifies by sorting the instances based on the feature values. As seen in Figure 1 each node represents a feature to be classified, e.g. at1, and each branch represents a value that the feature can be, e.g. a1. The classifier will define the class of an unknown instance by starting in the root node and working its way down considering its values for the features.

The feature that best divides the set should be selected to be the root node. There are various methods for finding which feature is the best one to divide it, like information gain and gini index. [2] The methodology of selecting the node will be repeated to create sub-trees until all the leaves of the tree contain the classifying class of the pathway.

The kNN is a instance-base algorithm that is based on the idea that the instances will exist in close proximity to others with similar characteristics. All instances are tagged to the value of their class and those that are undetermined

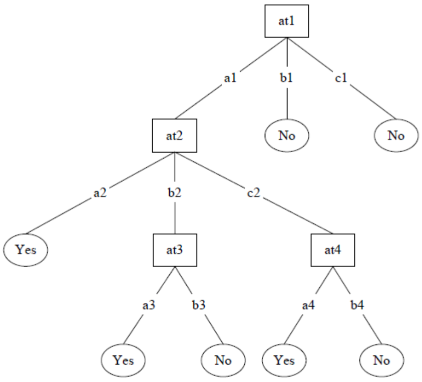


Fig. 1. Decison tree diagram. [2]

will be classified by observing those that are nearest to it. The algorithm will locate the k nearest instances to the one that wants to be classified and will label it as the one that is more frequent within the neighbors. In the case of Figure 2, the undetermined instance, represented by the red star, would be classified as being part of the green class if the selected k is 3, but will change to blue if increasing the value of k to 6.

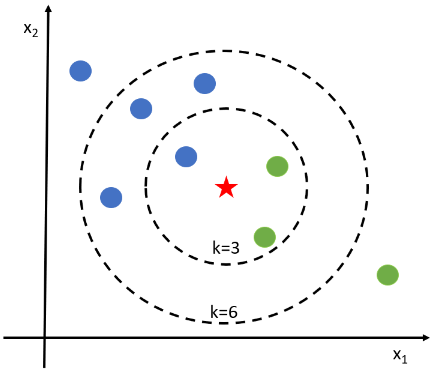


Fig. 2. The k-NN algorithm diagram, where the undetermined instance is the red star and the two possible classes are either blue or green.

The general concept that governs kNN algorithm is that all instances can be understood within a n-dimensional instance space, where each dimension correspond to one of the n features that describe each instance. The relative distance between the instances will be the factor that defines which are those that are nearest to the undefined case. There relative distance is determined with a distance metric, such as Euclidean, Minkowsky, Manhatan, Chebychev, to mention a few. [2]

In this paper the application of the two classification techniques mentioned will be analyzed by comparing its accuracy and running time of classification. The cases where the classifiers will be applied are: post-operative recovery place for a patient, contraceptive method of choice, and cover type.

2 Methods

2.1 Data sets

All data sets were obtained from the UCI Machine Learning Data Repository, University of California Irvine. [1] The data sets were multivariate and consisted on a classification problems. The selected sets were: post-operative patients, contraceptive method of choice, and cover type.

Post-Operative Patient The classification task consists on determining where a postoperative patient in a recovery area should be sent next. Therefore the patients were classified intro three categories of where they should be sent: intensive care unit (I), general hospital floor (A) or home (S). The data set contained 90 instances with 8 attributes, which mostly corresponded to body temperature measurements for the concern of hypothermia. Some of the attributes included: internal temperature (L-CORE), surface temperature (L-SURF), oxygen saturation (L-O2), blood pressure (L-BP), stability of surface temperature (SURF-STBL), stability of core temperature (CORE-STBL), stability of blood pressure (BP-STBL), and comfort. It is important to mention that in this data base there were instances where values were missing.

Contraceptive Method Choice This data set was obtained from some of the parameters in the 1987 National Indonesia Contraceptive Prevalence Survey, where the samples were taken from married women that were not pregnant or did not know it when interviewed. The idea was to classify type of contraceptive method used from their demographic and socio-economic characteristics . The types of contraception were classified into three classes: no use, long-term or short-term. The dataset consisted of 1473 instances with 9 attributes, which included: woman's age, woman's and husband's education, number of children, woman's religion, working status of woman, husband's education, standard-of-living index, and media exposure. No instance was missing attribute information.

Cover type The data was recollected in order to create a model which could classify the forest cover type from only cartographic variables. The forest cover type classes included: Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, and Krummholz. The data set is composed of 581012 instances that include 54 attributes, where 10 consisted of quantitative variables and 44 of binary qualitative indicators. The quantitative variables included: elevation, aspect, slope, vertical and horizontal distance to hydrology, horizontal distance to roadways, horizontal distance to fire points, and hill shade at 9 a.m., noon and 3 p.m.. The qualitative variables consistence on the absence or presence, 0 or 1 respectively, to define if it was part of one of the four types of wilderness areas and which of the forty types of soil is present. No instance was missing attribute information.

2.2 Classification

Each data set was shuffled and split into two parts: the training set by using two-thirds for training (66%) and the other third for estimating performance(34%). [2] Afterwards, each training set was used as described in the next subsections depending on the algorithm used. The accuracy of the algorithms is reported. This process was repeated different times according to the computational capabilities available depending on each database and each method in order to find an average behavior. The amount of repetitions are reported in the Results section.

Decision Tree Algorithm The information gain method was used in order to determine the structure of the decision tree. It consists of a recursive method where, in a training set T, the feature with most information, a, will be chosen. This will subdivide T into T_{ai}subclasses. Then, for every T_{ai}, a new feature with the most information will further subdivide the problem. This will happen until all elements have as a final feature the class of the element.

The information gain I given by an element a in a training set T will be given by:

$$I(T, a) = H(T) - H(T, a)$$

where $H(T)$ consists of the entropy, the pondered average of information contained of the whole set, and $H(T, a)$ is the entropy given the division by element a. The entropy is given by:

$$H(T) = - \sum_{C \epsilon ClassValue} p_C \log_2 p_C$$

where p_c corresponds to the probability of the class to exist in the set T and c is one of the possible values that the set can be classified.

The entropy given the division by element a is calculated by:

$$H(T, a) = \sum_{i \epsilon AttributeValue} p_{ai} H(T_{ai})$$

where $p_{ai} = |T_{ai}|/T$ and $H(T_{ai})$ is calculated with the same formula as $H(T)$ but the whole is considered as the subset T_{ai}.

The attribute with the highest information gain is the one selected as the node. Then, the process is repeated for the subset which generates given the division of the possible values for the selected attribute. As mentioned earlier this process is done recursively until all final features, or leaves, are the class of the element.

In order to reduce the amount of branches caused by calculative data, for example age of the woman in the contraceptive method case or elevation for the cover type data set, the instances were grouped into three groups given the distribution of the data. The first group were all the numbers less than the mean minus one standard deviation. The middle group was within the values of one standard deviation more and one standard deviation less than the mean. The final group was those who were larger than one standard deviation more than the mean. The values given were 1, 2, and 3, respectively. It was done before the training of the algorithm was realized. This allowed a faster computational performance and a greater generalization capability for the algorithm.

If the testing set presented a scenario where the features in the attribute were not completely found, the pathway through the tree stopped and the highest probability, in the subset where the attribute stopped, was the one selected to classify the undefined attribute.

***k*-NN Algorithm** In order to find the k-NN, the Euclidean distance was selected as the metric to define the distance between each instance. The Euclidean distance between two instances x_i and y_i is given by:

$$D(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^2)^{1/2}$$

where n is the total number of attributes.

All the distances for each instance compared to the unclassified one were calculated in order to consider them posteriorly. An array of values for k was applied and then the value with the best accuracy was selected. The array of was of the next numbers: 1,3,5,7,9,30,45 and 60. This was posteriorly used for the next repetitions of the algorithm.

It is important to note that the attributes were changed in order to satisfy the application of this algorithm. Those qualitative attributes that were not in a binary format, for example low, medium and high for temperature in the post-operative patients, was changed to numeric values that were between 0 and 1, and that were equally distributed between the parameters. In the same example of temperature, low medium and high were changed for 0, 0.5 and 1, respectively. Additionally, those qualitative values that were larger than 1 were normalized by dividing the whole set of characteristics by the largest one. This was done so all parameters were equally weighted in the distance metric.

3 Results

3.1 Decision tree

The construction of the decision tree as a final result of the training set was printed as a key value structure within the tree. The example for the post-operative patient decision tree is shown as follows.

```
'BP–STBL': {
  'mod–stable': {
    'L–BP': {
      'mid': {
        'L–SURF': {
          'mid': 'S',
          'high': 'A'
        }
      },
      'high': 'A',
      'low': 'A'
    }
  },
  'unstable': {
    'L–O2': {
      'good': {
        'SURF–STBL': {
          'unstable': {
            'L–CORE': {
              'mid': 'S',
              'low': 'A'
            }
          },
          'stable': 'I'
        }
      },
      'excellent': 'A'
    }
  },
  'stable': {
    'COMFORT': {
      '10': {
        'L–SURF': {
          'mid': {
            'L–CORE': {
              'mid': {
                'SURF–STBL': {
                  'unstable': 'A',
                  'stable': 'A'
```

```
                                    }
                                },
                                'low': 'S'
                            }
                        },
                        'high': 'A',
                        'low': 'A'
                    }
                },
                '15': 'S',
                '07': 'S'
            }
        }
    }
}
```

In order to comprehend the results in the dictionary decision tree, Figure 3 was realized in order to graphically explain the result of the algorithm application. In the tree, the feature with the highest information gain was the stability of patient's blood pressure (BP-STBL). It is also possible to see that not all the pathways for each possible scenario are complete. For example, the stability of patient's surface temperature (SURF-STBL) in the penultimate node of the pathway to the left shows a classification only for the values of stable and unstable, even though it could also be mid-stable. In the case an attribute had all the characteristics until it arrived to this node but had a mid-stable surface temperature stability, the value of A (send the patient to the general hospital floor) would be selected since most scenarios have this classification.

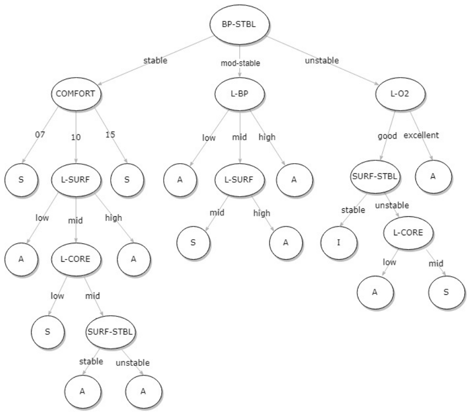


Fig. 3. Decision tree classifier generated by the implementation of the algorithm as a diagram of the coded result.

The accuracy results for the implementation of the decision tree algorithm for each data set were included in the Table 1. It is possible to view how the accuracy for the algorithm reduces between the Post-Operative Patient and Contraceptive Method Choice. It is important to highlight that the amount of attributes between both only increments by one and the iterations of the second is 16.3 times the first. On the other hand, the Cover type decision tree has greater generalization capabilities as seen by the higher mean accuracy rate with a low standard distribution. It is important to note that they all had results better than a random guess of the classification, which is a good indicator.

Table 1. Mean accuracy results for decision tree algorithm

Data set	Repetitions	Mean accuracy	Standard distribution
Post-Operative Patient	100	53.47%	8.41
Contraceptive Method Choice	100	43.98%	2.07
Cover type	1	68.53%	0.33

In order to further compare the performance of the algorithm for each data set, the running time for one iteration of the algorithm was recorded as seen in Table 2. As expected, as the amount of data increases between the data sets this increases the running time. Additionally, it is important to note that the amount of attributes changes considerably between the first two data sets, with 8 and 9 attributes, and the Cover type, with 54 attributes. This increments the complexity of the calculation and therefore the running time.

Table 2. Running time results for decision tree algorithm

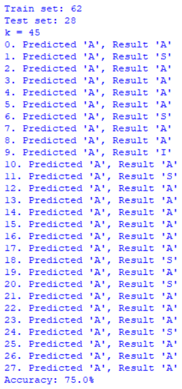
Data set	Running time (s)
Post-Operative Patient	0.0104
Contraceptive Method Choice	0.0775
Cover type	845.3673

3.2 *k*-NN Algorithm

In order to select the best k out of the array of possible values, a test for each possible k was done as viewed in Figure 4. It shows the results for the predictions in the Post-Operative Patient data set for the training set. The best k in this case was 45 with an accuracy of 75%.

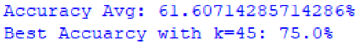
A comparison to the average accuracy of all possible k values was reported of 61.60%, as seen in Figure 5. This shows how the accuracy of the algorithm was improved by selecting the best value.

Finally, with the selected k value of 45, a repetition of the algorithm was done in order to have the average behavior given the selected characteristics.



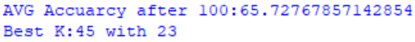
```
Train set: 62
Test set: 28
k = 45
0. Predicted 'A', Result 'A'
1. Predicted 'A', Result 'S'
2. Predicted 'A', Result 'A'
3. Predicted 'A', Result 'A'
4. Predicted 'A', Result 'A'
5. Predicted 'A', Result 'A'
6. Predicted 'A', Result 'S'
7. Predicted 'A', Result 'A'
8. Predicted 'A', Result 'A'
9. Predicted 'A', Result 'I'
10. Predicted 'A', Result 'A'
11. Predicted 'A', Result 'S'
12. Predicted 'A', Result 'A'
13. Predicted 'A', Result 'A'
14. Predicted 'A', Result 'A'
15. Predicted 'A', Result 'A'
16. Predicted 'A', Result 'A'
17. Predicted 'A', Result 'A'
18. Predicted 'A', Result 'S'
19. Predicted 'A', Result 'A'
20. Predicted 'A', Result 'S'
21. Predicted 'A', Result 'A'
22. Predicted 'A', Result 'A'
23. Predicted 'A', Result 'A'
24. Predicted 'A', Result 'S'
25. Predicted 'A', Result 'A'
26. Predicted 'A', Result 'A'
27. Predicted 'A', Result 'A'
Accuracy: 75.0%
```

Fig. 4. Best K value of iteration for Post-Operative Patient dataset



```
Accuracy Avg: 61.607142857142866%
Best Accuarcy with k=45: 75.0%
```

Fig. 5. Summary of iteration for Post-Operative Patient dataset



```
AVG Accuarcy after 100:65.72767857142854
Best K:45 with 23
```

Fig. 6. Overall Result of Knn with Post-Operative Patient dataset

Table 3 shows the results for accuracy given the different data sets. It is possible to see how there was a considerable increase between the accuracy of the first and last data sets. This is an indication of the effect of the increment of normalized attributes are able to increase the possibility of grouping similar classes together.

Even though the k-NN had a good accuracy, considerably better than a random guess, the running time performance was costly. The time it took to run the algorithm was highly due to the need to find a value of k that was ideal for the set. This resulted in a long waiting time for the system to give a result. The increase is of two orders of magnitude between the first two data sets and of three between the last two.

4 Discussion

As seen in Table 1, the accuracy for the algorithm reduces between the Post-Operative Patient and Contraceptive Method Choice was reduced. This shows how having to fit the data of a larger number of iterations gives a system lower generalization capabilities. The second data set has 16.3 times more iterations than the first. However, it is interesting to note how it was possible to generate a more accurate system with the largest data set by changing the quantitative

Table 3. Mean accuracy results for k-NN algorithm

Data set	**Repetitions**	**Mean accuracy**	**Standard distribution**
Post-Operative Patient	100	65.72%	7.21
Contraceptive Method Choice	1	45.84%	0
Cover type	1	81.26%	0

Table 4. Running time results for k-NN algorithm

Data set	**Running time (s)**
Post-Operative Patient	2.8
Contraceptive Method Choice	Above 130
Cover type	Above 170k

characteristics to qualitative ones with the parameters selected. Since the middle group of separation contained around 68.3% if the data was normally distributed.

A similar behavior happened to the k-NN algorithm, where the largest data set had the best performance, then the first one and at last the contraceptive method choice prediction. This highlights that a greater number of attributes with comparable weights, since they all were normalized, were better in order to generalize the predictions.

In the running time aspect of the performance for the different algorithms behaved as expected with the increase of iterations and attributes. As seen in Table t: dtTime, the amount of attributes considerably increased the time. The amount of attributes changes considerably between the first two data sets, with 8 and 9 attributes, and the Cover type, with 54 attributes, as does the time. Of course the amount of time for execution also is influenced by the quantity of iterations, but if the time increase was proportionate to the amount of data then it would have increased 16.3 times, as the increase of iterations. However, this is not the case since the time only increased 13.4 times.

When comparing the running time performance of k-NN to the decision tree algorithm the results are completely different. Just comparing the value for the running time for the first data set, the time increased 558 times for the k-NN algorithm. The others even have greater increments between them. The greater generalization capabilities of k-NN was achieved through a process that had to repeat many iterations in order to find the optimal value for *k*. This was a computationally-expensive technique. Additional to this disadvantage, it requires large storage requirements and they are sensitive to the choice of the similarity function. [2]

5 Conclusion

Computational classification algorithms have been applied to various problems of diverse natures. In this paper three different data sets were used to apply the Decision Tree and k-NN algorithm classification in order to compare their performance. It was possible to see that the k-NN had greater generalization capabilities than the decision tree algorithm. However, it required a computationally-expensive technique in order to find the value of k that gave the best accuracy for the system. This exercise showed the nature of these algorithms and the need to determine new ways to process the iterations to reduce the computational cost. Finally it is concluded that the price of accuracy could be achieved, but it could increase the computational cost of running these algorithms.

References

1. Dheeru, D., and Karra Taniskidou, E. UCI machine learning repository, 2017.
2. Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering 160* (2007), 3–24.