­Lab 3 – Machine learning

Group: 2, “Joseph, Adam”, “Alfredji, Zainab”

# INTRODUCTION

This assignment aims to strengthen our knowledge about machine learning algorithms such as the k-mean algorithm and the ID3 decision tree algorithm.

The k-mean algorithm is useful for many real-life scenarios such as market segmentation, anomaly detection or document clustering. The usefulness lies in the algorithm ability to discover underlaying patterns in data by grouping similar items/data points together. It does this be identifying “k” number of centroids (one for each cluster) and assign each data point to its nearest centroid. When all data points have been assigned to clusters the centroids are then recalculated as the mean of all points in each cluster. This is then repeated iteratively until the centroids stabilize and the cluster assignments no longer change significantly[[1]](#footnote-2).

By using the k-mean algorithm, businesses can segment customers based in purchasing behavior or healthcare can identify regions of interest in complex datasets.

The Iterative Dichotomiser 3 (ID3) algorithm is a method used in machine learning to create decision trees, mainly for classification tasks. The ID3 uses information gain to determine which attribute in the data set gives the most information and then selects that attribute as the root node for the decision tree. This is done by calculating the whole data set entropy and also each attribute features entropy [[2]](#footnote-3).

As mentioned above the algorithm is used for classifications tasks, which could be e.g useful in healthcare, by diagnosing patients where the tree represent a medical test and the leaves represent potential diagnoses.

# DESIGN AND IMPLEMENTATION

**Clustering: k-mean**

The k-means algorithm was designed to divide the dataset into ‘num\_clusters’ clusters. It iteratively assigns data points to the nearest centroid and the recalculated centroids based on the current cluster groups. The algorithm uses the SSE(Sum of Squared Errors) to measure the tightness of the clustering and stops when improvements fall below the ‘termination\_tol’. This is briefly shown in the pseudo code below.

**Pseudo code**

* function kmeans(data\_points, num\_clusters, termination\_tol, max\_iter):
* initialize centroids by randomly selecting num\_clusters data points
* repeat max\_iter times:
* for each data point:
* calculate distance to each centroid
* assign data point to the closest centroid
* for each cluster:
* recalculate the centroid as the mean of all points in the cluster
* calculate SSE for the current clustering
* if change in SSE < termination\_tol:
* break

return final centroids, assigned clusters, and SSE

The elbow method is used to determine the optimal number of clusters. Where it ranges between 1 and 15 of the value of ‘k’. The plot below shows us the total within-cluster sum of square(WSS) against the number of clusters ‘k’. The point where the plot starts bending and gains less reduction in WSS is considered the ‘elbow’, indicating the optimal number of cluster which is 5.

A graph of a number of clusters

Description automatically generated

After determining the optimal ‘k’ from the elbow plot, which is 5. Here is the plot for the clustering of the dataset.

A diagram of different colored dots

Description automatically generated

**Classification: split decision tree**

The algorithm worked by creating a tree where each node represents a feature field, and the branches represent the outcomes of the test on that feature, leading to a decision or prediction. The recursive splitting leads to a tree structure where the paths from root to leaf represent classification rules (shown in the pseudo code below). To apply the algorithm to the golf dataset, it will be used to construct a tree with a maximum height of 3. This involves selecting attributes at each level of the tree that best split the data, up to three levels deep (shown in the text representation of the tree below).

**Pseudo code:**

function split(node):

if node is a leaf (node.height == max\_height or all targets are the same):

node.class\_name = find\_dominating\_class(node.target\_column)

return

else:

calculate initial entropy for the node's target column

for each feature in the node's input columns:

calculate weighted entropy for splitting by this feature

if this split's info gain is the best so far:

record this feature as the best for splitting

if a best feature is found:

node.split\_variable = best\_feature

for each unique value of the best feature:

create a child node with the subset of data for this value

recursively call split on the child node

else:

node.class\_name = find\_dominating\_class(node.target\_column)

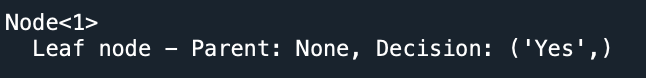
return

**Text representation of the tree:**

A computer screen shot of a black screen

Description automatically generated

Different maximum heights in the decision tree affects its complexity. A lower height might lead to underfitting, where the tree does not capture the underlying patterns well, while a higher height might lead to overfitting, where the tree becomes too specific to the training data. Here are some text illustrations of the different height constant.  
  
**Height 1:**

****

**Height 2:**

**A computer screen shot of a program

Description automatically generated**

**Height 4:**

**A screenshot of a computer program

Description automatically generated**

By these results, it is shown that both height 1 and 2 is underfitting the dataset we have, while height 4 is overfitting/unchanged because of how small the dataset is. Which makes height 3 most accurate for this specific dataset.

# CONCLUSION AND FUTURE WORK

In this assignment we have demonstrated the effectiveness of two machine learning algorithms were each is serving different purposes in data analysis.

The correctness of the k-mean algorithm was showcased through an elbow plot determining the optimal number of clusters for our dataset, which was 5 in our case. While for the decision tree we tried out trees of different heights and found that a tree with a height of 3 gave the most accurate model without overfitting or underfitting.

More thorough research about these algorithms could have provided more fine tuning of the algorithms or maybe bigger data sets used for the ID3 decision tree could provide more possibilities for testing different tree heights and see the algorithm effectiveness.

What we’ve learned is that machine learning is a huge part of AI and that there are many algorithms that can be used and tuned for solving multiple real-life problems. But we’ve also learned the importance of really get to know your data and understand the context you’re working in.

Because of the weaknesses of k-mean algorithms which is its way of handling outliers, it would be an improvement to try to implement the k-mediod algorithm instead. But because the assignment was about handling numerical data only this was not necessary because the k-medoid is a better choice when there is categorical data to be handled as well.

In short, this assignment was a great hands-on experience with two key machine learning techniques, understanding them better and building knowledge on how to use them for future work and problems.

# REFERENCES

1. GeeksforGeeks, [K means Clustering – Introduction](https://www.geeksforgeeks.org/k-means-clustering-introduction/) [internet]; 2023.
2. Tom M. Mitchell, [Decision Tree Algorithm (ID3)](https://www.cse.unsw.edu.au/~cs9417ml/DT1/decisiontreealgorithm.html) [internet]; 2023.

1. GeeksforGeeks, [K means Clustering – Introduction](https://www.geeksforgeeks.org/k-means-clustering-introduction/) [internet]; 2023. [↑](#footnote-ref-2)
2. Tom M. Mitchell, [Decision Tree Algorithm (ID3)](https://www.cse.unsw.edu.au/~cs9417ml/DT1/decisiontreealgorithm.html) [internet]; 2023. [↑](#footnote-ref-3)