UNIVERSITAT DE BARCELONA

Work 3: Lazy Learning

Introduction to Machine Learning

Hasnain Shafqat

João Valério

Eirik Grytøyr

Date: 23/12/2022

Index

1. INTRODUCTION

The k-Nearest Neighbour (kNN) is a well-known non-parametric, supervised lazy learning classifier, which assumes that similar instances have a close link in the feature space. In this strategy, new predictions are determined by letting the new observation give votes between the k nearest examples in the training data.

In order to improve this process, there are several different parameters and pre-processing strategies that can be used. The goal of this study is to examine the impact of those modifications and search for the most satisfactory combination in the two present data sets.

2. METHODOLOGY

The datasets used for analysis are cmc and SatImage, in which the preprocessing is identical to the previous report, with two exceptions. The primary one is that, for efficiency purposes, the categorical data is only converted into numerical in the distance measure of the kNN algorithm. Secondly, this time the data is split into 10 equal-sized folds, in which each fold is used in turn as a test set with the remaining 9 as training sets.

From that standpoint, a comparative analysis in the use of feature weighting, the number of neighbours to consider, distance metrics and voting methods. Through this, it is expected to comprehend the best combination of parameters per dataset and the influence of each parameter in the quantitative results, accuracy and efficiency.

Afterwards, with the best kNN configuration, three distinct instance reduction techniques (condensed, edited and hybrid) are developed and studied as a preprocessing step in the algorithm. Through that, it is pretended to reduce the number of inconsistent and noisy cases, in order to get a more reliable dataset to support the kNN decisions. The quantitative results are obtained according to accuracy, efficiency, and storage. Finally, the best instance selection technique is determined based on a statistical method.

3. MODEL

In the third section is explained the parameters of the Grid Search in chapter 4, in order to acquire the optimal hyperparameters of the model Knn and the dataset. Through this process, it is intended to accomplish the most suitable solutions, according to accuracy and efficiency. Thus, in the following subchapters, it is explained the theory behind the 4 parameters that are being considered for the effect, along with the respective distinct approaches.

a. Feature Weighting

Firstly, it is considered feature weighting, which is applied as a preprocessing tool. This way, it is entitled as a filter, since, algorithmically, it is performed without any adjustments due to the KNN results' quality. Therefore, overfitting and biassed data situations are avoided, through a fast feature selection, which is extremely useful for the large datasets considered in this work.

To accomplish the desired outcome, three different approaches, with distinct underlying reasonings, are implemented.

- **Equal Weight:** All the features are considered to have the same weight or importance, so, each one is assigned a unitary value. Therefore, the final data set corresponds exactly to the original one.
- **Information Gain/Mutual Information:** In this technique, it is performed a weighting of the features based on interdependence, in which 0 indicates no correlation, whereas values proximate to 1 reveal a high dependency. This function relies on non-parametric methods based on entropy estimation from Knn distances. This way, as the method does not depend on the distribution of the data, it is efficient for large datasets, such as the ones in the current work.

  Furthermore, it is executed as a prior step to the function, the creation of a boolean vector to distinguish continuous and discrete variables because the algorithm does not hold the inherent capacity to do it. Therefore, the results are more trustworthy.

  Lastly, this method is applied recurring to the *sklearn* library.
- **Chi-Square:** In this approach, the dependency between stochastic variables is measured. In other words, calculates the features that are the most likely to be independent of class and therefore irrelevant for classification.

  Likewise, this model is applied via the *sklearn* library.

As the last point, it is necessary to highlight that since the algorithms return weights in different scales, these are normalised between 0 and 1. Thus, it might occur that at least one feature (the least important) is erased from the dataset.

b. K-Number

After the previous definition, points b., c. and d. are developed with direct respect to the internal functionality of the KNN.

The first parameter is k, which corresponds to the number of neighbours considered to label a data point during the consultation time of the model. In order to achieve the most suitable k, four values are tested: 1, 3, 5 and 7. Through this variation, it is intended to comprehend the effect of k in the final evaluation metrics, as well as the smoothness variation of the boundary and the quality of the data's structure caption by the model.

For the selection of k, it is expected that the larger number for k, will result in better accuracy when the data set is noisy. On the other hand, it might result in less accuracy, when the datasets are unbalanced.

c. Distance Metric

In point c., it is explored the distance metrics used by KNN to quantify the spatial closeness of 2 data points. To accomplish the desired outcome, it is implemented 3 different approaches with distinct underlying reasoning.

- **Minkowski (r = 2) / Euclidean:** The primary consideration is the Euclidean distance, which considers the shortest spatial distance between 2 points in the dimensional space as shown in equation 1.

$$d(x,\ y)\ = \sqrt{\sum_{i=1}^{n} |x_i^{\ 2} - x_i^{\ 2}|} \quad (1)$$

- **Cosine:** This method finds the similarity between the angles of the vector for the two data points, in which the same direction indicates closeness. Since the angle of the vectors is used, cosine similarity has an integrated normalisation for each observation and is often used when the magnitude of the vectors is not relevant.[2]

$$cos\theta\ = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^{\ 2}}\sqrt{\sum_{i=1}^{n} B_i^{\ 2}}} \quad (2)$$

- **Canberra:** This technique is similar to Manhattan distance, however, it is weighted, where values closer to zero have a larger impact on the distance. Furthermore, It is more sensitive to proportionality than to absolute differences [1]. Finally, It is robust to outliers compared to other metrics and can, therefore, be used as a tool to detect those outliers [3].

$$d(x,y)\ = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i + y_i|} \quad (3)$$

Overall, besides the influence on the evaluation metrics results, it will also comprehend how different mathematical considerations lead to distinct classifications, due to the various perceptions of closeness.

d. Voting Method

In a KNN machine learning algorithm, there are different ways to decide which label to assign to an observation based on the nearest neighbours. In that spectrum, three distinct approaches are implemented.

- **Majority class:** This is the simplest method. All the occurrences are counted for the K nearest neighbours of the new observation and the label of the majority class is assigned to the data point. In the case of a tie between the occurrences, the distance to the different labelled neighbours is summed per class, in which the shortest total distance is assigned to the new observation.
- **Inverse Distance Weighted Votes:** In the current method, instead of just basing the label on counting the neighbours, it is possible to use the distances so that a close labelled observation counts more than one farther away.

$$Vote(y_j)\ = \sum_{c=1}^{k} \frac{1}{d(q, x_c)^p} \quad (4)$$

Where, C is the neighbour, q is the new observation and p is a parameter to decide the importance of the closeness to the neighbours.

- **Sheppard's work:** In the third voting method, instead of using the inverse proportionality, an exponential function is employed, as shown in equation 2:

$$Vote(y_j) = \sum_{c=1}^{k} e^{-d(q, x_c)} \quad (5)$$

A tie between the weighted voting technique is unlikely since float numbers are used. Additionally, in that case, the assigned label would not be important, since the probability of choosing the right label decreases when the distance to different labels is similar. However, as the assignment requires a method to solve this potential situation, it is decided that, in the presence of the tie, the k value increases by one and the voting is repeated.

4. STATISTICAL METHODS

To decide the best model for each performance measure, they are tested on 10 folds of data. Since the performance of the configurations will depend on this data, statistical methods will be applied in this process for deciding the best parameters in chapter 4, and the best instance reduction technique in chapter 6.

The statistical methodology, in both applications, is constituted by the **Friedman** test which is a non-parametric equivalent of the repeated-measures ANOVA and the **Nemenyi** test [6]. The methodology is as follows: Firstly, the results between the models for each fold are ranked separately. Furthermore, in a tie situation, the average rank is assigned. Then the average rank is used to compare and find the best model. The Friedman test will then be performed with a null hypothesis meaning that there is no statistical significance between the models. However, in the case where the null hypothesis from Friedman's Test is rejected, in this case with a p-value less than $\alpha = 0.01$, there is proceeded with a post-hoc test. In the present work, the statistical method defined for the purpose is the Nemenyi Test, with the aim of comparing all the classifiers to each other. The Nemenyi test produces a critical difference which the classifiers' results need to have in order to be statistically separable (with $\alpha = 0.05$). This is plotted in the relevant chapters.

Following the definitions of the parameters, it is fundamental to describe a statistical process, by which the best kNN model will be selected. Thus, these methods play a role in chapters 4, where the best combination of parameters is analysed, and 6, in which it is intended to decide the best reduction technique, as a preprocessing tool of the kNN.

The Friedman and Nemeny tests are used because they are appropriate to use when classifying multiple models [6]. However, it's also important to note that there are some limitations associated, since they don't provide a strength of the relationship between them, but just a general P-score. The models also require equal statistical distributions to be accurate. Additionally, there is also a limited number of samples (10) compared to the number of models 108 for the parameterization and 9[1] for the instance reduction.

5. RESULTS AND ANALYSIS

In this chapter, the results from the parameter combinations of Chapter 2 are debated. Firstly, in section a, the most suitable combinations are described and their set of parameters is defined.

[1] This includes the results from the data without feature reduction.

Finally, the subsequent subchapters analyse the influence of each parameter in the KNN quantitative outcome. In that perspective, in each topic, one parameter varies while the remaining are static and correspond to the optimal.

a. Optimal Parameters

In table 3, according to the criteria, the most relevant combination per dataset is presented. In order to obtain these, the results are first ranked per accuracy and time, according to the Friedman Test. Additionally, even though the best model is decided based on accuracy, according to the assignment requirements, both quantitative metrics are obtained and analysed.

Based on Friedman Test, table 1 indicates the average top 5 most and the least suitable accuracy and time average rank between all 108 models (1080 executions), along with the values for each dataset.

Table 1 - Friedman Test rank.

	cmc			SatImage		
Results	**Accuracy**	**Time [s]**	**Avg. Rank**	**Accuracy**	**Time [s]**	**Avg. Rank**
1st Rank - Acc.	0.516	0.284	12.75	0.912	4.624	14.75
2nd Rank - Acc.	0.516	0.293	12.75	0.912	4.753	14.75
3th Rank - Acc.	0.509	0.290	13.00	0.912	4.646	15.10
4th Rank - Acc.	0.508	0.408	13.50	0.911	4.615	18.10
5th Rank - Acc.	0.506	0.271	14.15	0.910	4.617	19.00
Last Rank - Acc.	0.395	0.344	96.65	0.787	3.844	106.55
1st Rank - Time	0.439	0.233	8.00	0.900	3.300	4.70
2nd Rank - Time	0.463	0.235	9.00	0.887	3.300	4.80
3th Rank - Time	0.476	0.236	10.20	0.908	3.302	4.90
4th Rank - Time	0.439	0.236	11.20	0.908	3.314	6.90
5th Rank - Time	0.453	0.238	12.10	0.898	3.313	7.00
Last Rank - Time	0.441	0.474	106.0	0.909	5.695	107.80

Firstly, it is essential to indicate that, as noted previously, the best kNN is defined as the one that achieves the highest accuracy ranking. In cmc that corresponds to 51.6% accuracy with 0.284s in prediction time, while in SatImage the accuracy and time are 91.2% and 4.624s, respectively.

For cmc, between the best and worst accuracy outcomes, the discrepancies are 12.1% and 0.060s. Furthermore, in the time ranking the difference between the first and last position is

approximately 0.241s. Hence, in cmc accuracy can be prioritised, since the time disparities are significantly diminished.

In the Satimage dataset, the pattern is dissimilar from the last dataset. The differences in the accuracy and time from the first to the last accuracy rank are 12.5% and 0.780s. Furthermore, from the best to the worst time rank the accuracy decreases by 0.9% with an increment of 2.395s in time. This way, there is not a strict relation between both metrics. Finally, the best time achieved is 3.300s, with a discrepancy of 1.324s and 0.2% for the 1st Rank accuracy chosen as the best model. This indicates that it is possible to choose a more efficient model with a reduced cost of accuracy.

Furthermore, the previous values indicate, in the best combination, discrepancies between cmc and SatImage outcomes of 39.6% in accuracy and 4.340s on time. In terms of accuracy, even though cmc (1473 observations and 3 classes) has a quarter the number of instances as SatImage (6435 observations and 6 classes), it has only half the number of classes to classify. This lack of information per class might be one of the reasons for the significant discrepancy in accuracy. Additionally, the curse of dimensionality might be a significant factor since the number of dimensions for SatImage is 36 compared to 9 for cmc[4].

Finally, the general ranking positions indicate that in both cases, as expected, the average ranking of the best values is lower in time than in accuracy, indicating more consistent results along the time folders' combination. As the folders have the same size, the time execution ranking relies only on the metrics selected. For instance, an execution with k equal to 1 will be faster than 3, independently of the data points. On the other hand, accuracy calculation depends on the metrics involved and the data points, leading to more inconsistent results. Another relevant point is that, in accuracy, the average ranking in cmc is lower than in SatImage, meaning more consistent results. This can be because the latter has double classes, is more complex, and makes the result from each data fold more model-dependent.

Furthermore, for each dataset, the best combination of parameters in accuracy and time is as follows:

Table 2 - Most suitable KNN parameters.

1st Rank - Accuracy				
Dataset	**K**	**Weight**	**Distance**	**Voting**
cmc	7	Information gain	Canberra	Majority
SatImage	3	Equal	Canberra	Inverse Distance
1st Rank - Time				
cmc	1	Equal	Minkowski	Inverse Distance
SatImage	5	Chi-Squared	Minkowski	Inverse Distance

In table 2, for accuracy, the most suitable combinations achieved for both datasets are represented in the first two rows, where it is noticeable that only Canberra as a distance metric

is coincident. Thus, it is possible to infer that the best metrics depend on the inherent characteristics and properties of the dataset. For instance, it is interesting to denote that even though cmc has fewer instances and classes, it needs a higher k to achieve better results, indicating that the classes' differentiability is not as sound as in SatImage.

On the other hand, regarding efficiency, all the metrics are similar, as expected. The exception is the number of nearest neighbours in SatImage, which is 5 instead of 1. However, this is due to the error associated with the computer executions, since, theoretically, a lower k indicates less computational cost. Thus, the metrics that lead to the most efficient classification in kNN are, at least partially, independent of the dataset.

Table 3 shows the p values obtained from the Friedman test, and since the values are lower than the limit α_T previously defined (1%), the null hypothesis is rejected.

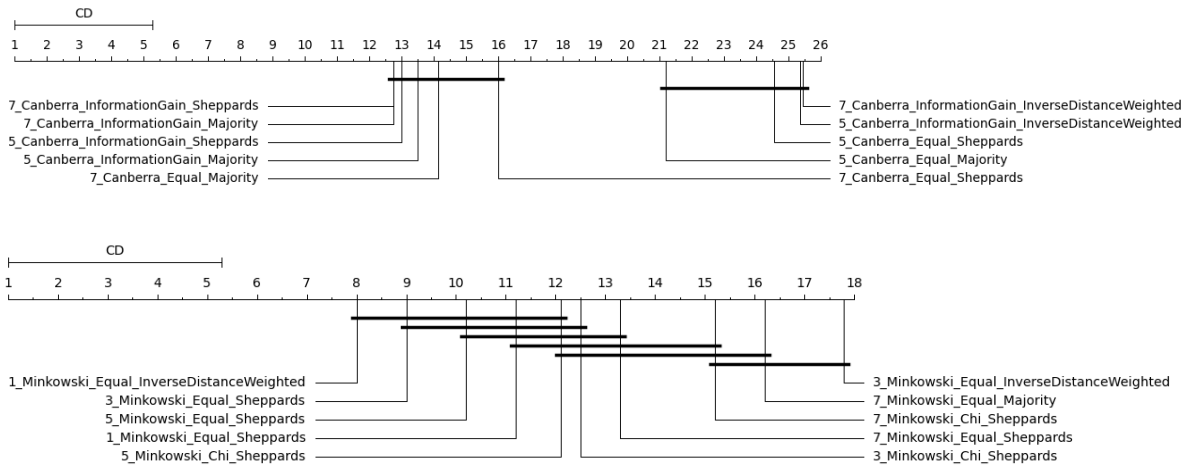
Table 3 - p-values from Friedman Test.

Results	Accuracy	Time
cmc	1.27e-41	3.13e-131
SatImage	3.48e-127	3.39e-155

Hence, it is necessary to perform the Nemenyi Test in order to understand the differentiability between the models in both datasets.

Plot 1 is the top 10 accuracies and times, respectively, on the Nemenyi Test's results, in cmc, for α_N equal to 5%.

Plot 1 - Rank 10 in Nemenyi Tests for accuracy and time, respectively, for cmc.



According to the accuracy results for the highest ten ranks of Friedman, it is understandable by the Nemenyi Test that the H_0 Hypothesis withstand and the first 6 models can't be considered different. However, this does not mean that the models are considered equal, but rather that there is not enough evidence to conclude that they are different. From that perspective, it would

be plausible for the 6 best, to select according to efficiency. Furthermore, with this outcome, it is possible to indicate that the total 7 optimal values in the kNN for cmc are:

- **K:** 5 or 7
- **Weight:** Information Gain or Equal Weight
- **Distance:** Canberra
- **Voting:** Majority or Sheppards

In which the only two non-optimal combinations out of the eight possible are: 5 + Canberra + Equal Weight + Majority or Sheppards.

In terms of efficiency, it is noticeable that the separability between models is, as expected, higher than in accuracy. In fact, some of the models are considered the same due to the error associated with time in the executions and the low level of the computational cost of cmc dataset, since algorithmically it is intuitive that a k equal to 3 leads to more operations than when its value is unitary. However, according to the results, the most efficient parameters are:

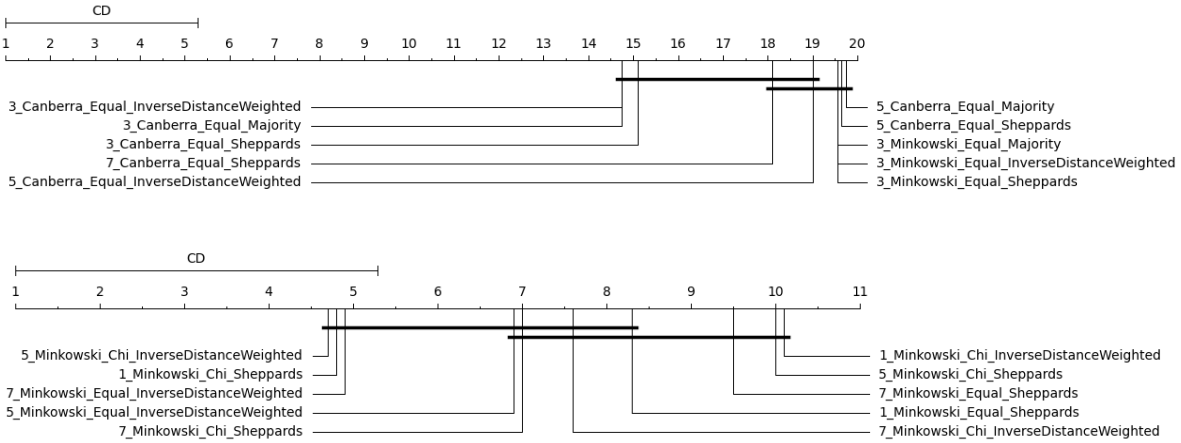
- **K:** 1 or 3 or 5
- **Weight:** Equal Weight or Chi-Squared
- **Distance:** Minkowski
- **Voting:** Inverse or Sheppards

When combined in the 5 ways exposed in plot1.

As the last point, between metrics, the ranks associated with accuracy are higher than in time. As stated previously, for the same fold size, the accuracy depends on the data's inherent characteristics, while time relies, almost, exclusively on the methods applied.

Likewise, plot 2 illustrates the same information in SatImage.

Plot 2 - Rank 10 in Nemenyi Test for accuracy and time, respectively, for SatImage.



In SatImage, in terms of accuracy, the possible optimal configurations are only 5, however, the variability of parameters (total of 8) is higher than in cmc (total of 7).

- **K:** 3 or 5 or 7
- **Weight:** Equal Weight
- **Distance:** Canberra
- **Voting:** Majority or Inverse or Sheppards

This difference might be due to the size of the datasets and the number of classes. Moreover, the previous premises are also the origin of the increase in the accuracy ranking from cmc to SatImage. However, it is essential to note that these patterns do not have a strong correlation with the accuracy per dataset, because, as it is known, the model performs better at 39.6% in SatImage.

In the efficiency, it is evident that from cmc to SatImage the average ranks decrease. This behaviour is expected because for a bigger dataset the differences between combinations are more sound. Thus, it is easier for the Nemenyi Test to project these discrepancies. Nonetheless, the number of optimal efficient parameters is the same as in cmc:

- **K:** 3 or 5 or 7
- **Weight:** Equal Weight or Chi-Squared
- **Distance:** Minkowski
- **Voting:** Inverse or Sheppards

In which seven possible configurations are possible.

Overall, in both datasets, it is possible to infer, from the Friedman Test ranking followed by the Nemenyi Test, that it is possible to create an intersection of parameters between the most suitable in accuracy and time. The only persistent exception is the distance metric, where Canberra outweighs the remaining in accuracy, while Minkowski does it in time.

Despite the previous reasoning, the assignment requirement is to decide the model through accuracy. So, in summary, the best kNNs per dataset have the following characteristics:

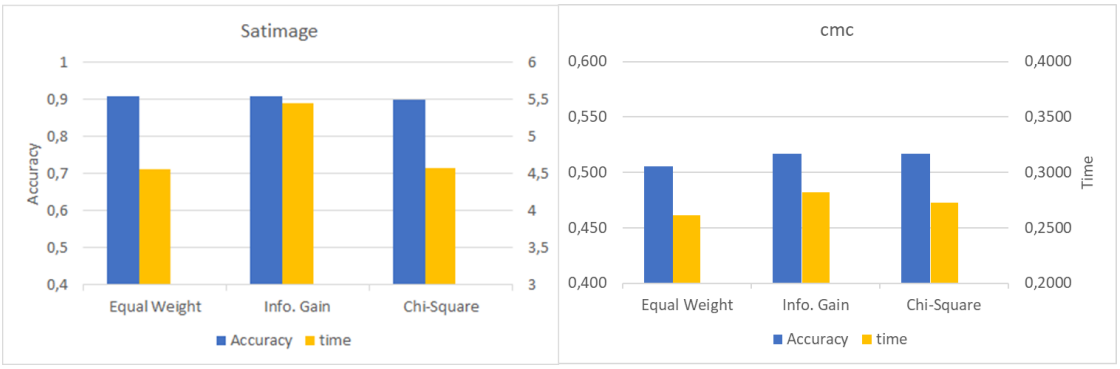
- **cmc:** 51.6% accuracy and 0.284s with Information gain weight, k=7, Canberra distance and Majority voting;
- **SatImage:** 91.1% accuracy and 4.624s with equal weight, k=3, Canberra distance and inverse distance voting.

b. Feature Weighting

With the best combinations as a base, with the parameters fixed, in this subchapter the influence of weighting type variation is going to be analysed for both datasets.

Plot 3 illustrates the values obtained for accuracy and efficiency.

Plot 3 - Weighting influence on the most suitable KNN.



According to the graphical values obtained it is noticeable that in both datasets, the influence of the weighting in the accuracy results is irrelevant. However, related to time, the influence is

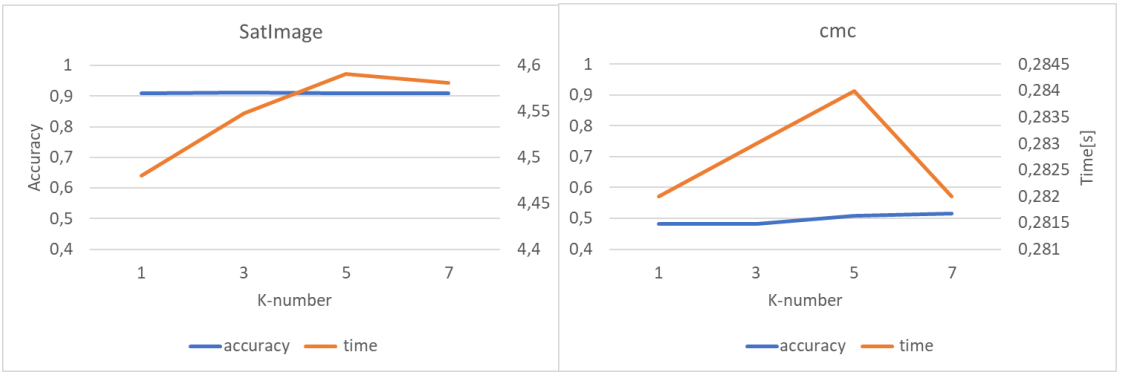
extremely dependent on the feature space dimensions. As expected, equal weighting is the most efficient and information gain is the least one. So, overall, the trade-off between accuracy and time is unbalanced, especially for larger datasets, such as SatImage. Thus, usually, the elected weighting feature is the equal one.

As a final note, the relation between equal weight, information gain and Chi-Square are similar for both datasets, indicating, once again, that these efficiencies only depend on feature dimensionality and not on other inherent characteristics or properties of the dataset. So, the discrepancies between approaches grow along with the feature space.

c. K-Number

Following the previous reasoning, the influence of k is represented in plot 4.

Plot 4 - k influence on the most suitable KNN.

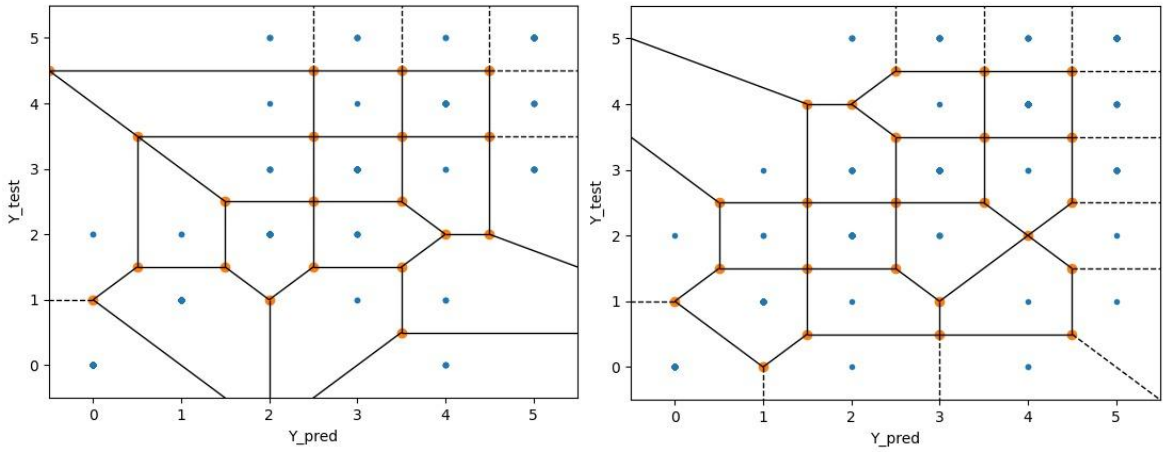


In general, the plots demonstrate that, in terms of accuracy, the value is steady in the SatImage dataset, while in cmc it is noticed a slight improvement, with the increment of k. This indicates that the noise and balance in the data are not compensated or uncompensated significantly in the range of k between 1 and 7.

On the efficiency subject, as debated in a., it is noticed an overall pattern in which the computational cost increases with k. The exceptions from this convention emerge from the error associated with the algorithms' execution.

Moreover, it is possible to study the smoothness of the decision boundary for different k values. In that panorama, it is presented in the SatImage dataset, for the same test fold, the Voronoi Diagram for k equal to 1 (left) and 7 (right), the most extreme cases.

Plot 5 - Voronoi Diagram in SatImage dataset, for k equal 1 (left) and 7 (right).

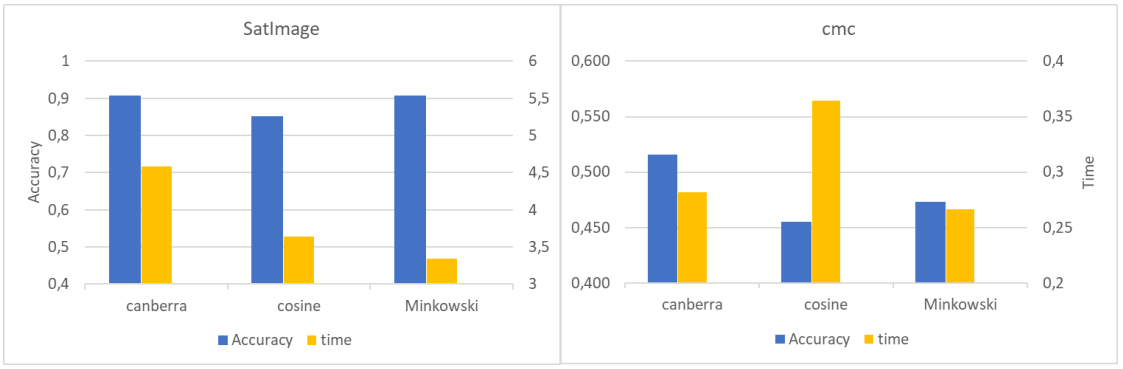


According to the Voronoi Diagrams obtained, it is understandable that the increment of the number of nearest neighbours in the kNN classification increases the smoothness of the boundary decision. This is discernible by the number of vertices in each decision boundary, which, in general, is higher for k equal to 7. Furthermore, by overlapping both diagrams it is also possible to understand the regions with a higher difficulty of classification in which a higher number of neighbours is necessary.

d. Distance Metric

In the distance metric perspective, the results from Canberra, Cosine and Minkowski are displayed in plot 6.

Plot 6 - Distance metric influence on the most suitable KNN.



According to the results, it is visible that for both datasets the Canberra distance is the best metric in terms of accuracy. The fact that the Canberra metric is the most suitable method may indicate that the data is either on a different scale or it's mainly constituted by categorical features since this metric is independent of the order of categories.

Precisely, in the SatImage dataset the discrepancy for the second most suitable accuracy, provided by Minkowski, is not significant. However, the difference in time can be considered relevant, so, it would be appropriate, as stated in point a., to opt for a more efficient method, since the accuracy cost is reduced. On the contrary, as seen previously, in cmc the efficiency differences are irrelevant.

Finally, Cosine never seems to be a suitable choice, particularly in cmc, where it is the poorest in both metrics.

e. Voting Method

Finally, the last feature analysed is the voting method in plot 7.

Plot 7 - Voting influence on the most suitable KNN.

In the voting methods, it achieved the most balanced results between methods from all the features discussed in the current configuration and the weighting of the votes didn't improve significantly. Weighted voting methods are useful when the variables in the data have different levels of importance or when the decision boundary is complex or has a significant level of noise[7]. This might indicate that the data in general is equally important or has smooth decision boundaries. This also corresponds well to the result from the SatImage set, which has the highest k-value.

In cmc the same pattern is verified, however, the Majority is in fact the most accurate method. Nonetheless, Sheppards' approach would also be feasible as an election for the kNN.

6. REDUCTION METHODS

Succeeding the subsequent analysis, with the optimised kNN, it is pretended to study the impact of instance reduction techniques on the quality of the results, according to accuracy, efficiency and storage. Thus, in the three main units, condensed, edited and hybrid, the self-implemented algorithms selected and the brief correspondent descriptions are:

- **Condensed Reduction Technique:** In this section the algorithm selected is the **Generalised Condensed Nearest Neighbour (GCNN)**. GCNN's objective is to keep the most important observations, called prototypes, by iteratively adding one more prototype to the list, per class. The iterations end when all the instances' closest prototype is of the same class (they are absorbed). Additionally, it is possible to add a margin, so the instances are only absorbed if the closest prototype is of the same class with a margin of ρ. Equation 6 shows this criterion.

$$||x - q|| - ||z - p| > \rho, \quad \rho \in [0, 1] \qquad (6)$$

Where p and q, both prototypes, are, respectively, the nearest homogeneous prototype to x and the nearest heterogeneous prototype to x [5].

According to equation (6), for a larger value, fewer examples are removed, while for a unitary value no observations are removed. Finally, in order to get the optimal configuration, ρ will be tested with values of 0.01, 0.001 and 0.0001.

- **Edited Reduction Technique:** For the current section, the algorithm is entitled **Edited Nearest Neighbor Estimating Class Probabilistic and Threshold (ENNTh)**. This technique is based on local information of an instance, by considering the underlying probability distribution in the neighbourhood of the data point.

  Furthermore, the data points are erased from the training data if either the highest probability does not coincide with the real class or it is not higher than the threshold μ. Since the datasets have distinct characteristics, the thresholds tested are distinct. While cmc is tested with 10%, 12%, 14% and 16%, SatImage the values are 20%, 30%, 40% and 50%.

- **Hybrid Reduction Technique:** In the final topic, the model executed is the **Decremental Reduction Optimization Procedure 3 (DROP3)**. DROP3 intends to determine if it is feasible to remove an instance from the data, in which the reduction process occurs. This if at least as many of its associates would still be classified correctly without it. Through this evaluation, each associate (every instance that has the referred data point as one of its neighbours) is examined to understand the impact of the imputation. However, priorly, it is fundamental to apply a noise filter using a rule to prevent overfitting and slightly smooth the decision boundary. Then, after eliminating the noisy examples, the instances are sorted based on the proximity of their nearest enemy, removing first the points farthest from the actual decision boundary. This allows points internal to clusters to be removed early in the process, even if there were noisy points nearby. After completing this preprocess, it is necessary to delete disposable instances.

  Finally, in this method, there are no parameters to be tested.

a. Models' Results

The parameters that are coincident with the kNN, such as k and distance, are considered with the optimal values from chapter 4. In the remaining, the reduction techniques are applied with different parameter values, in order to obtain the optimal one. Table 4, 5 and 6 indicate the accuracy, time and storage values for each parameter variation.

Table 4 - GCNN.

	Accuracy		Prediction Time [s]		Storage [% of Data]	
	SatImage	**cmc**	**SatImage**	**cmc**	**SatImage**	**cmc**
No Reduction	0.912	0.516	3.490	0.210	100	100
ρ=0.01	0.909	0.496	1.401	0.172	38.8	78.9
ρ=0.001	0.894	0.496	0.860	0.172	18.7	70.3
ρ=0.0001	0.893	0.481	0.821	0.158	18.9	69.7

According to GCNN reductions, none of the configurations tested is able to increase the accuracy values performed by the kNN. However, the reductions were not significant, since with ρ equal to 0.0001, the soundest reduction, the differences of 1.9% (SatImage) and 3.5% (cmc) in accuracy, allow for time and storage drops, respectively, of 2.669s and 81.1% (SatImage), and 0.052s and 30.3% (cmc). These results are particularly interesting in SatImage, where the time and storage requirements are costly.

As the last point, most of the accuracy is achieved when ρ equals 0.01, in which a difference of 0.3%, allows for a considerable reduction in storage of 21.1%.

Table 5 - ENNTh.

	Accuracy		Prediction Time [s]		Storage [% of Data]	
	SatImage	**cmc**	**SatImage**	**cmc**	**SatImage**	**cmc**
No Reduction	0.912	0.516	3.493	0.353	100	100
μ=0.32 | 0.10	0.910	0.530	2.514	0.244	84.6	67.5
μ=0.34 | 0.12	0.810	0.525	0.869	0.251	22.7	67.0
μ=0.36 | 0.14	0.574	0.514	0.783	0.379	17.3	62.8
μ=0.38 | 0.16	0.467	0.483	0.660	0.217	12.8	41.8

* The configuration μ= value 1 | value 2, indicates that value 1 corresponds to SatImage, while value 2 is related to cmc.

Based on the values of table 5, the accuracy of cmc dataset increased in two distinct situations, in which the most prominent improvement is 1.4%, with fewer 0.109s and 32.5% in time prediction and storage, by the order mentioned. This indicates that a relevant part of the original data in cmc is misleading. In fact, this pattern can be seen by the probabilistic values associated with each class per data point, where, usually, the highest probability does not outweigh the remaining, leading to lower probabilistic values. Precisely, in 41.8% of the data, the probability of the points belonging to their real classes is, only, at least 0.16%.

Relatively to SatImage, it is possible to set a higher value for the minimum probability accepted (μ), indicating a sounder quality of the data. Moreover, even though it was not possible to improve the accuracy, a 0.2% decrement enables a drop of 0.979s in prediction and 15.4% in storage, which might be a suitable trade-off. Finally, 61.9% of the data points have a probability of belonging to the right class between 32% and 34%.

Table 6 - Drop3.

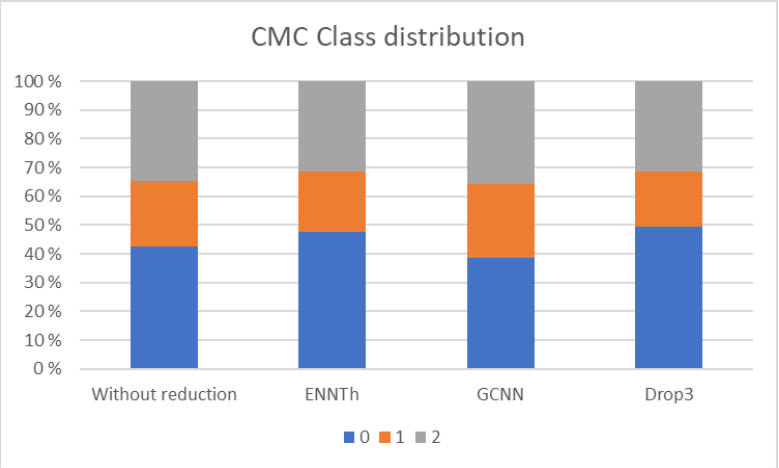
	Accuracy		Prediction Time [s]		Storage [% of Data]	
	SatImage	**cmc**	**SatImage**	**cmc**	**SatImage**	**cmc**
No Reduction	0.912	0.516	3.459	0.210	100	100
Reduction	0.879	0.511	0.945	0.069	22.6	26.7

In DROP3 the number of parameters to adjust is lower than in the previous models, indicating that it is a less flexible algorithm. However, the results achieved are suitable, since the prediction time decreased by 2.514s and 0.141s on SatImage and cmc, by the respective order. Furthermore, even though the final storage in both datasets is approximately 23%, the accuracies only decreased by 3.3% (SatImage) and 0.5% (cmc). So, even though there is no improvement in terms of accuracy, it is possible to achieve roughly the same outcomes with less prediction time and storage. This indicates that a considerable amount of data in both datasets is not util, which allows for memory and time benefits with low accuracy cost, as previously predicted in this report.

b. Class Distribution

Plot 8 illustrates the class distribution in cmc with and without the optimal reduction techniques. Since the most suitable model is only defined in chapter 6, the reduction techniques of the following plot correspond to the parameters with the highest accuracy in the tables.

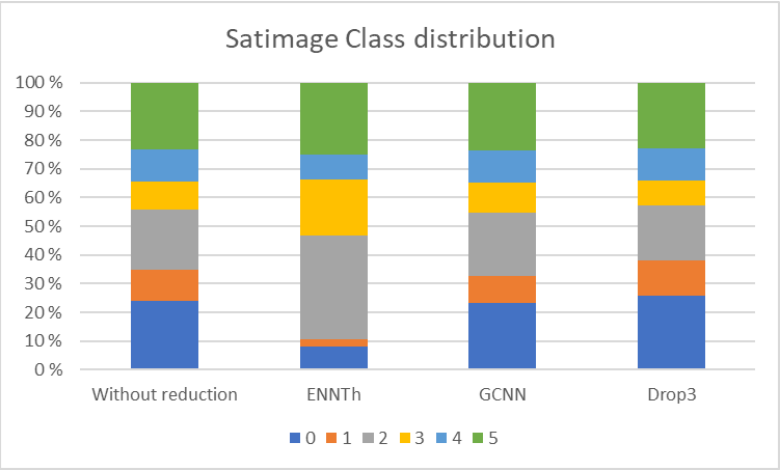
Plot 8 - Class distribution in cmc.



Through the analysis of plot 8, it is comprehended that the original dataset is not totally balanced, in which the least represented class is 1. From these instances, the interpretability of the reduction techniques about the instances' utility is distinct. While ENNTh and DROP3 increase the representation of class 0 and decrease class 2, while roughly keeping 1, GCNN reduces class 0 by augmenting 1 and keeping 2 representations. Taking that into account, it is inferable that even though ENNTh model does not lead to a more balanced dataset, it is in fact the only model with the capacity to improve accuracy, along with time prediction and storage benefits. Furthermore, since ENNth and DROP3 lead to similar class distributions, the discrepancy in the accuracy values is due to the number of instances reduced and which ones are perceived as non-relevant.

Plot 9 follows the previous analysis for SatImage.

Plot 9 - Class distribution in SatImage.



According to plot 9, once again, the algorithms have different perspectives related to the utility of the data points, leading to separate distributions. Firstly, the original dataset is not balanced, where classes 1, 3 and 4 are misrepresented. From that, the algorithms seem to agree on classes 4 and 5 in terms of representativity. However, in the remaining, the understandings are slightly different. While ENNTh reduces drastically 0 and 1, by increasing 2 and 3, GCNN keeps roughly the same distribution as the original dataset and DROP3 reduces classes 2 and 3, by increasing 0 and 1, the total opposite of ENNTh.

Taking the previous distributions into account and the tables, the highest accuracy (0.910) is achieved by ENNTh. However, the class distribution is completely unbalanced. So, in fact, this achievement is only possible because the storage reduction is 15.4%. On the other hand, GCNN and DROP3 have more uniform distributions between classes. Particularly, GCNN reaches 0.909, only 0.1% less than ENNTh, with a 61.2% storage reduction. Additionally, it is notable that the DROP3 instance removal precision, since only 22.6% of storage remaining allows kNN to acquire an accuracy of 0.879.

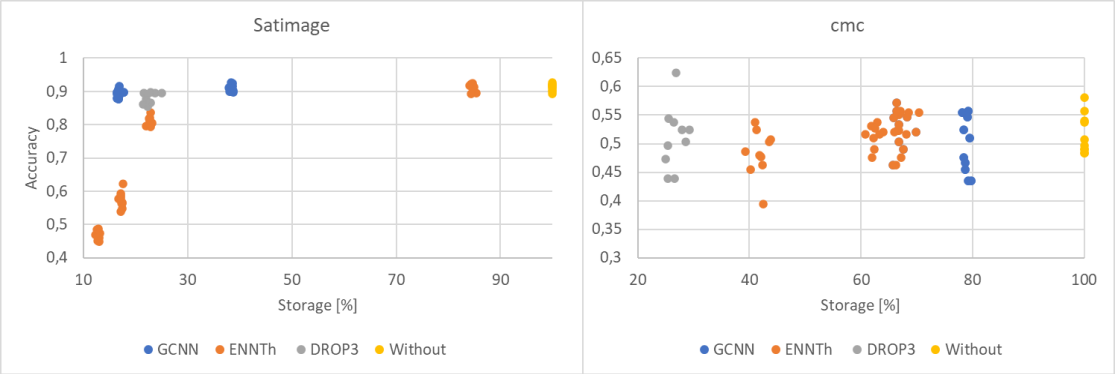
It is also important to denote that the performance metric is accuracy, which produces more satisfactory results when the training and test data share a common distribution. For instance, a method such as F1-Score would reflect more faithfully an improvement in class balance.

In general, this demonstrates that these reduction techniques when applied to all the classes are not efficient to downsample the overrepresented classes.

c. Accuracy and Storage comparison

Another aspect to study is the correlation between instance reduction and accuracy. Plot 10 shows this relation for the two datasets and for each distinct algorithm.

Plot 10 - Relation between instance reduction and accuracy in each model for both datasets.



Firstly, in the SatImage dataset, it is visible that the three reduction techniques considered in this work hold the capacity to reduce the storage, and consequently the prediction time, while kNN keeps, roughly, the same accuracy. However, in ENNth this pattern depends heavily on the parameters since the increment of μ leads to a set of wrongly removed instances. Thus, GCNN and DROP3 perform a better selection of the relevant instances than ENNTh.

Finally, in the cmc dataset, it is verified to be a distinct condition from the previously stated, since the dispersion between the data points. This way, the accuracy achieved per model depends heavily on the parameters, indicating a clear correlation between these and the accuracy achieved. This pattern might be due to the general unsatisfactory accuracy of the classification of the cmc data.

In general, the plots show that there is not a clear correlation between accuracy and the size of the remaining data set above a fundamental level for the algorithms.

7. STATISTICAL ANALYSIS OF THE REDUCTION TECHNIQUES

Finally, the aim of the last analysis topic is to provide continuity to the previous analysis by comprehending which of the previous instance reduction techniques is the most suitable as a preprocessing tool of the kNN with respect to accuracy, time consumption and storage reduction. For that purpose, the two statistical algorithms explained in topic 3 are employed.

In this chapter, the ranking performed by Friedman Test is not shown in a single table. Since tables 4, 5 and 6 already expose this information it is easy to infer the proper sort of values per category. Furthermore, this can be inferred from plots 10 and 11 relative to the Nemenyi Test.

The P-values obtained from the results of the Friedman test are presented in table 7.

Table 7 - Friedman Test rank

Results	Accuracy	Time	Storage
cmc	5.97e-4	2.02e-12	3.27e-15
SatImage	4.81e-13	1.74e-15	1.86e-15

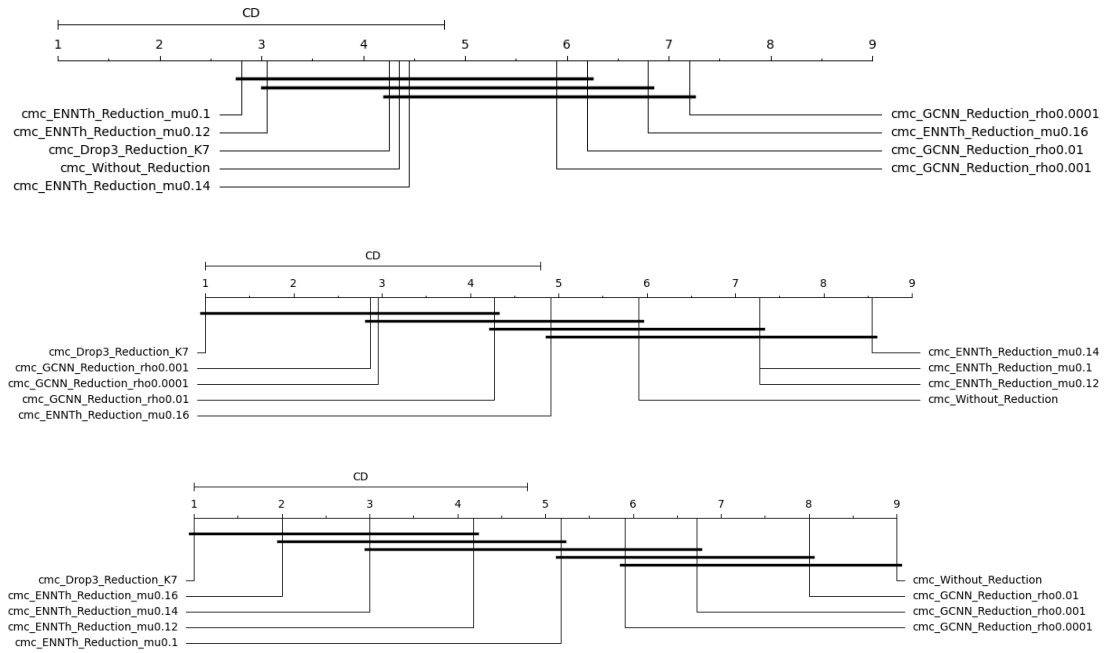
As the values indicate, with a defined confidence level of α_F= 0.01, the models are significantly distinguishable, with a confident margin, and as in chapter 4 the Nemenyi Tests are performed in

order to do the individual statistical analysis. The plots from the Nemenyi tests are used to show the average rankings from the Friedman tests.

a. cmc

Plot 11 shows the ranking between the models according to accuracy time and storage. These are obtained taking into account a statistical significance with a confidence level of $\alpha_N = 0.05$ as distinguishable.

Plot 11 - Rank 10 in Nemenyi Test for accuracy, time and storage, respectively, for cmc.



The shown rankings have a suitable correspondence to the result of the average values in chapter 5. Apart from that, the following statistical observations can be made:

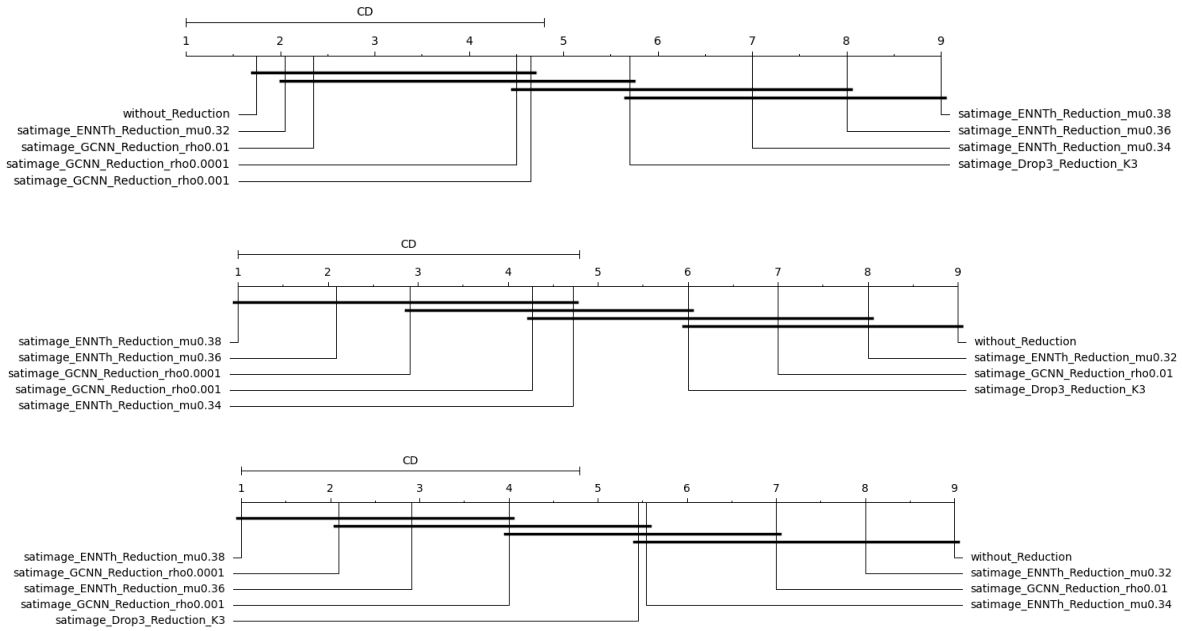
- Comparing the best configuration for each model related to the accuracy, none of them is significant.
- None of the models are proven to improve the accuracy, from tests on the complete data, but on the other hand, none of the models gives significantly worse results.
- Drop 3 gives a dataset with better time performance than the ENNTh reduction.
- Drop3 and the three best ENNTh reduce more storage than GCNN, for the tested parameters.

In general, the plots show that the storage metric is the most consistent measure from these tests, with the models well distributed along the axis, while the accuracy is the metric with the most variation and largest dependency between the model and the data folds.

b. SatImage

The same Friedman and Nemenyi test is performed on the Knn results after the reduction of the SatImage dataset, which is shown in plot 12.

Plot 12 - Rank 10 in Nemenyi Test for accuracy, time and storage, respectively, for Satimage.



From the previous illustrations, the general noticeable patterns are

- It is not proved that the algorithms will improve the accuracy, in a dataset like SatImage but all the GCNN configurations and the ENNTh, with mu = 0.32, might still be able to keep the accuracy, at the same time that GCNN with μ 0.001 and 0.0001 is significantly reducing the data size and the computation time.
- For the time plot, ENNTh with μ ranging from 0.34 to 0.38 and GCNN with ρ from 0.0001 to 0.001 are the model configurations that reduce the time the most.
- In the storage reduction plot, ENNTh with the two largest values of μ and GCNN with the two smallest values of ρ have the most impact, as expected from the description of the algorithms in chapter 5.

In general, the result reveals that there is a correlation between time consumption and storage reduction, even though it is not as significant as expected. The main reasons for this issue might be related to the hardware performance during the tests.

It is also displayed that when a feature reduction method needs to be selected, it is necessary to test different strategies and fine-tune the parameters since datasets with distinct characteristics require different methods. Nevertheless, this report might give an indication of a preferable starting point if a similar dataset is used.

Due to the reduced ability to conclude on the best models in these tests, the number of data folds should be greater compared to the number of algorithms, in order to make the critical distance smaller for a clearer conclusion.

An alternative method to reduce the size of the dataset would be to use feature reduction instead, which often is the preferable method [8]. This technique has the ability to remove irrelevant features and reduce the complexity of the data and still keep the relevant information.

This was applied in the previous report [9], which revealed that especially the dataset SatImage is formed by a considerable amount of unnecessary features since only four components after a

PCA reduction counts for 90% of the variance. This indicates that feature reduction will be the preferable option for the Satimage dataset.

Finally, there are some situations where instance reduction may be preferred over feature selection. For example, when the dataset is considerably large and the model is inefficient, or as a downsampling technique for unbalanced classes. However, since feature selection can also prevent overfitting and improve interpretability and then potentially cause an easier model optimization, feature selection will be the preferable alternative.

8. CONCLUSION

All the goals initially proposed were achieved with success and will be explained during the conclusion.

Firstly, it was comprehended that kNN is a Lazy Learning algorithm, which performs, each time, the class classification of a new data point according to the training data, leading to a high consultation time.

Secondly, the different metrics considered (weight, k, distance and voting) contribute to distinct algorithm configurations, in which the most satisfactory combination depends on the data. Usually, in accuracy, the main dependence relies on its inherent characteristic and properties, while in efficiency the outcomes lean, almost exclusively, on the parameters selected for the kNN.

Furthermore, performing instance reduction techniques as a preprocessing tool of the kNN might be useful for distinct circumstances and purposes. Usually, the first aim of this approach is to select the important instances and through that increase a higher accuracy with a lower consultation time. However, even though sometimes the accuracy growth is not achievable, the reduction of time and storage rates contribute to a good trade-off. Through this, more elevated flexibility in the kNN is accomplished, due to the fact that the model can be adjusted depending on the preferable viewpoint.

By analysing the behaviour of the instance selection, of the current work, and the feature selection, of the previous one, it is inferred that the most prominent approach depends on the dataset characteristics.

The statistical methods are useful to provide suitable information about the models according to different considerations. Through that, it is easier to understand which one of the model's configurations is elected for the project aim. Finally, the statistically significant parameters for a KNN classifier, which were common for both datasets, are a k of 5/7, Equal Weight, Canberra and Majority/Sheppards. In the feature reduction, the results are vague and depend more on the data and the parameters of the methods. As an improvement, to determine a more clear conclusion between the algorithms, a study containing more sets of data should be performed.

To conclude, the kNN is a suitable and flexible algorithm in the classification process. However, its maximum potential is explored when combined with preprocessing and statistical tools.

9. BIBLIOGRAPHY

[1] V. B. Surya Prasatha, Haneen Arafat Abu Alfeilate , Ahmad B. A. Hassanate , Omar Lasassmehe , Ahmad S. Tarawnehf , Mahmoud Bashir Alhasanatg,h, Hamzeh S. Eyal Salmane," Effects of Distance Measure Choice on KNN Classifier Performance - A Review" 2019;

[2] Chris Emmery, "Euclidean vs. Cosine Distance", 2017;

[3] Michał Szałański, "In Defense of Manhattan Distance - On the Performance Impact of Distance Metrics in Clustering Analysis", 2019;

[4] Grant Peter, "k-Nearest Neighbors and the Curse of Dimensionality", towardsdatascience.com, 2022;

[5] Fu Chang, Chin-Chin Lin, Chi-Jen Lu , "Adaptive Prototype Learning Algorithms: Theoretical and Experimental Studies, 2006;

[6] Janez Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets", Faculty of Computer and Information Science Trza˘ ska 25 ˘ Ljubljana, Slovenia, 2006;

[7] Ming Zhao and Jingchao Chen, Improvement and Comparison of Weighted k Nearest Neighbors Classifiers for Model Selection, 2015;

[8] Milad Malekipirbazari, Vural Aksakallib, Waleed Shafqatb, Andrew Eberhard "Performance comparison of feature selection and extraction methods with random instance selection", 2021;

[9] Hasnain Shafqat, João Valério, Eirik Grytøyr & Roberto Lopez "Work 2: Dimensionality Reduction and Visualization", 2022.