Modelling the Avila Dataset

23 May 2021

Simon Karumbi s3455453
Matthew Saunders s3782240

> *We certify that this is all our own original work. If we took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in our submission. We will show we agree to this honor code by typing "Yes": Yes*

Table of Contents

Abstract

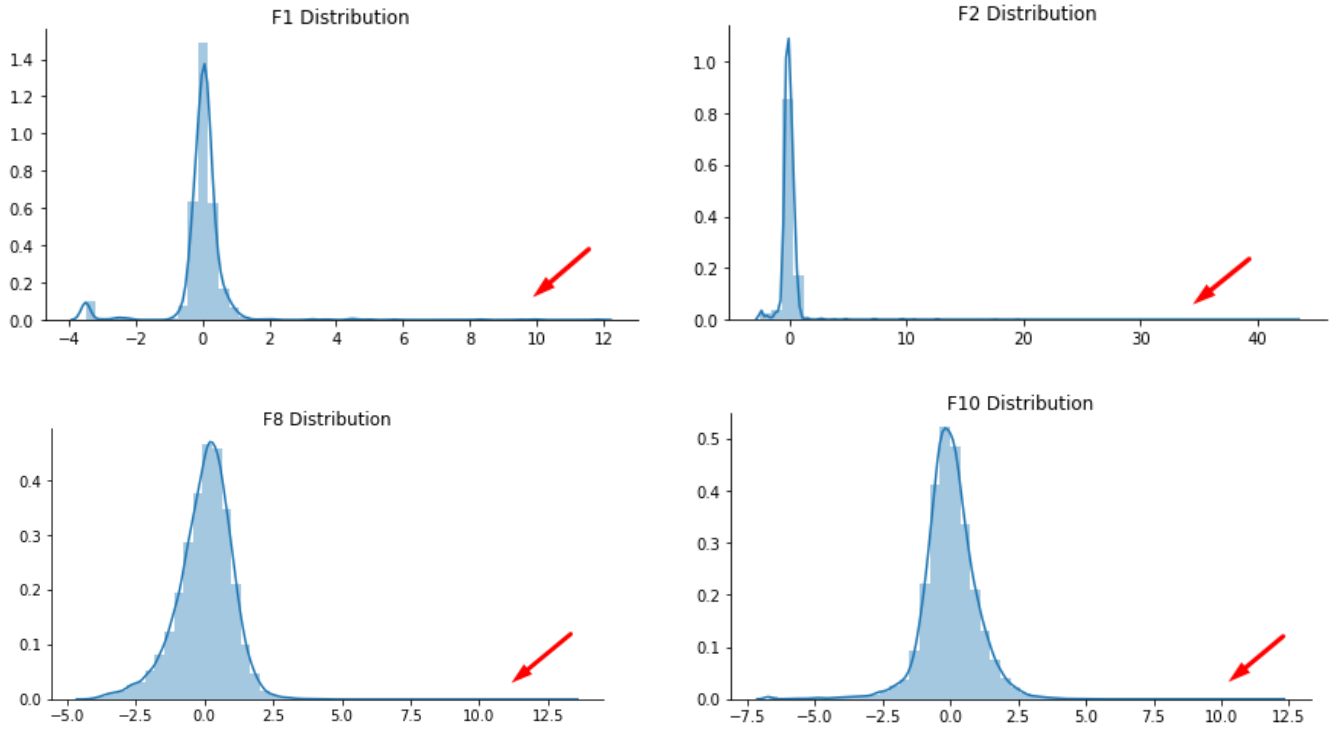
DBSCAN and KMeans Clustering algorithms were used to model the Avila Bible dataset, with little success. DBSCAN fails to identify appropriate clusters, due to an imbalance in the dataset, with a high proportion of one of the target classes overwhelming the other data, where KMeans fails to identify meaningful links within the multidimensional data. Other clustering algorithms may yield better results, however, it seems that clustering may not be suitable for this dataset.

Introduction

The Avila dataset is data based on images extracted from the Abila Bible, a 12th century Spanish/ Italian biblical text. There are 10 features, with a target class indicating the 12 copyists of each piece of text in the bible. It is a little unclear about what each row of the dataset refers to, and certain assumptions were made that are discussed later.

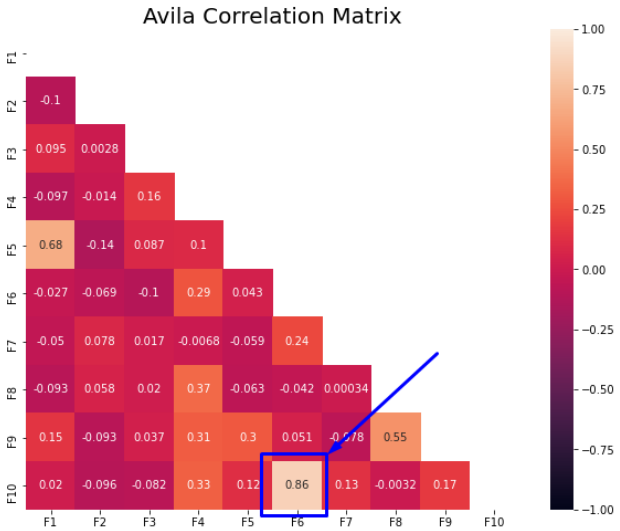
The aim of this report is to create a clustering algorithm capable of identifying the copyist responsible for each page, based on the features of the dataset. It is hypothesised that each copyist is characterised by a set of values within the feature space that will help to identify future instances of this data.

In this report, the K-means and DBSCAN clustering algorithms were used. The data was pre-processed and allegedly used Z-normalisation, however, it was found that there were several features that did not abide a standard normal distribution with extreme outliers more than 10 standard deviations away from the mean, including F1 (intercolumnar space), F2 (upper margin), F7 (interlinear spacing) and F10 (modular ratio/ interlinear spacing).

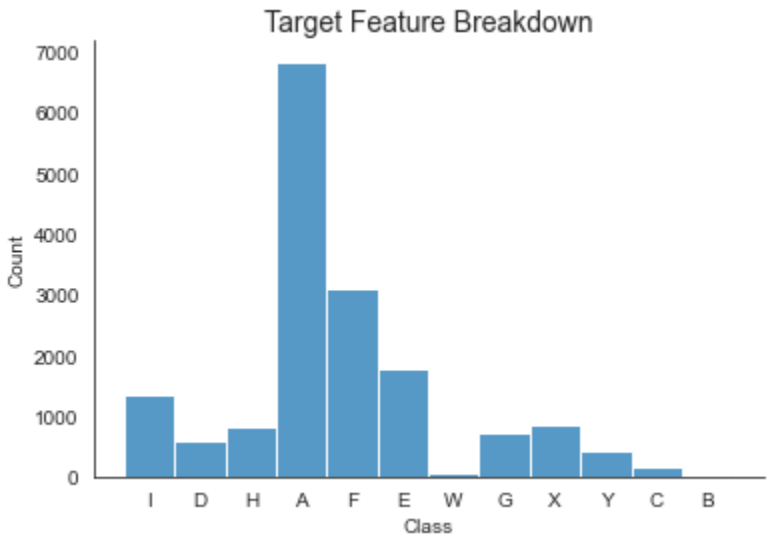


In order to address the inconsistencies in the standard normal distribution, it was decided to update all outliers (more than 5 standard deviations away from the mean) to the feature population mean (mu).

It was also decided to remove the F10 feature (modular ratio/ interlinear spacing) entirely, as it contained redundant data shown by a very high correlation with F6 (modular ratio).

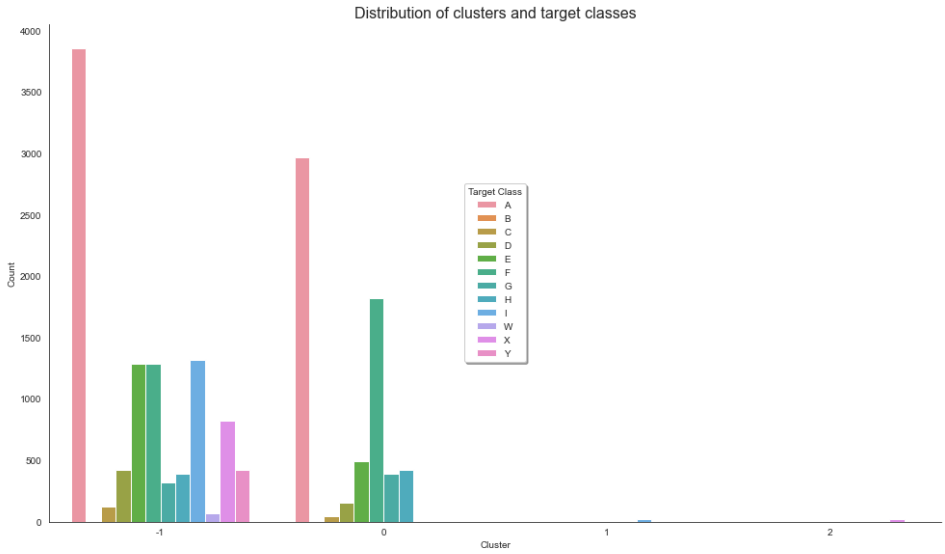


Once preprocessing of the data was completed, it was observed that the data was heavily imbalanced, with certain target classes represented with a much higher proportion in the data and some like ‘W’ and ‘B’ only account for several instances, however, addressing this is outside the scope of this report.



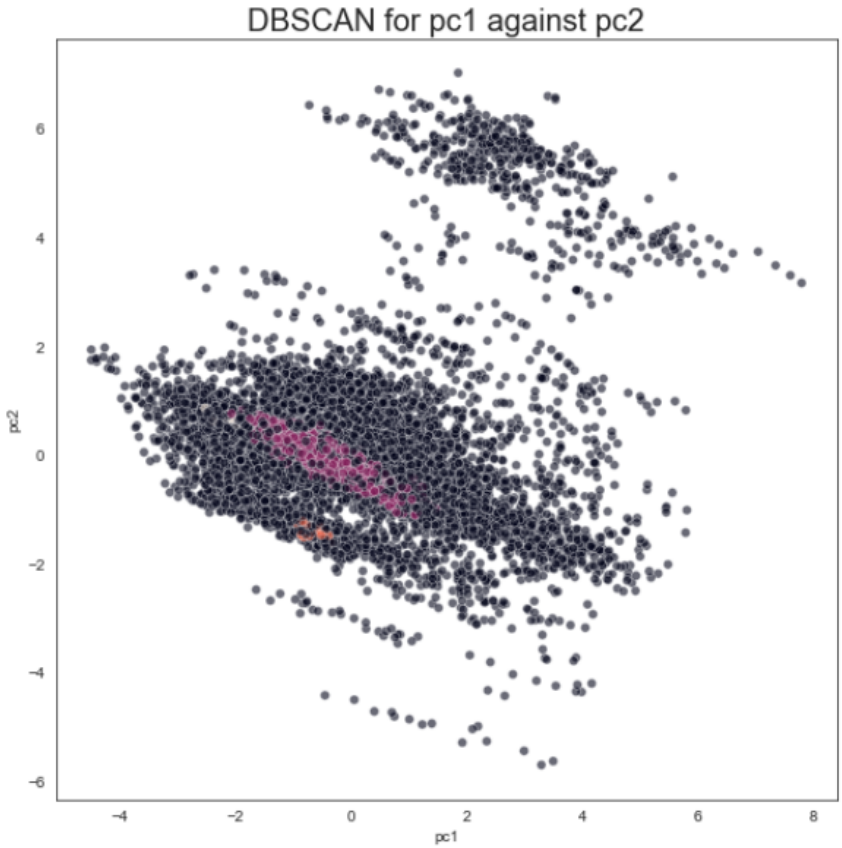
Methodology

The DBSCAN Clustering Algorithm was used as the first model in this report. The default epsilon distance of 0.5 was attempted, as well as minimal samples of two times the number of features in the dataset.

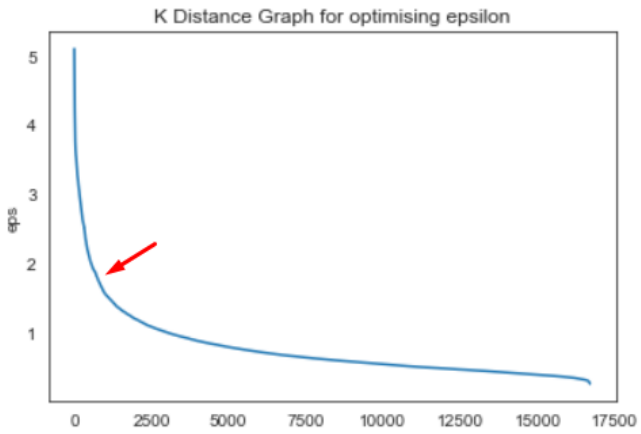


This first iteration yielded poor results, only identifying 3 potential clusters, and labelling most of the training dataset as noise.

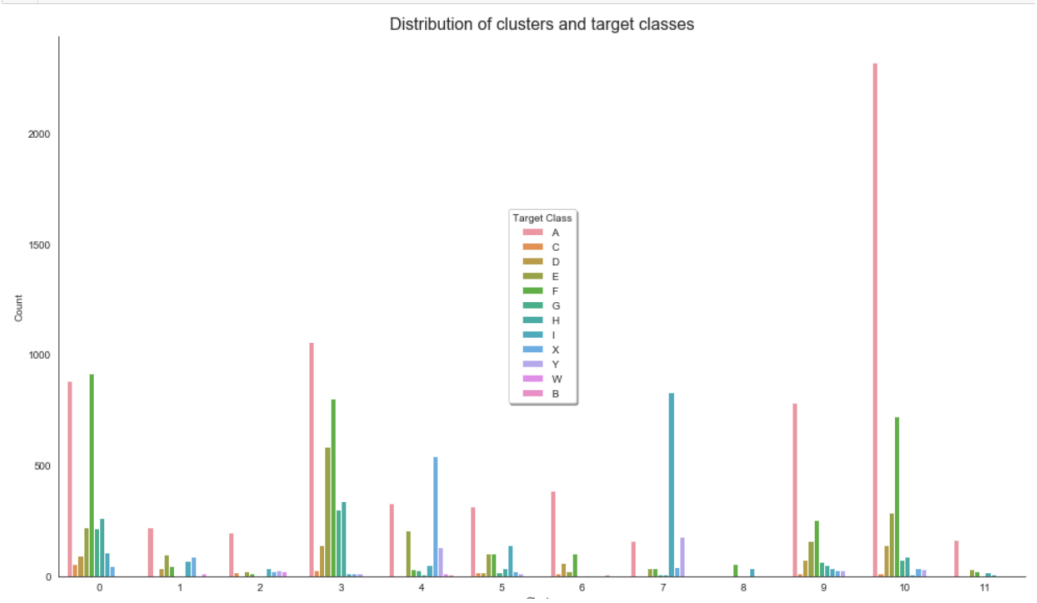
In order to visualise the data more effectively, Principal Component Analysis was used in order to reduce the data into principal components that best describe the variation within the data. We can see in our first iteration that there is some kind of cluster forming amongst the principal components.



In order to tune the model, a K distance graph was used to visualise the optimal range for epsilon where a range of 1.5 to 2.5 was decided to be the best location.



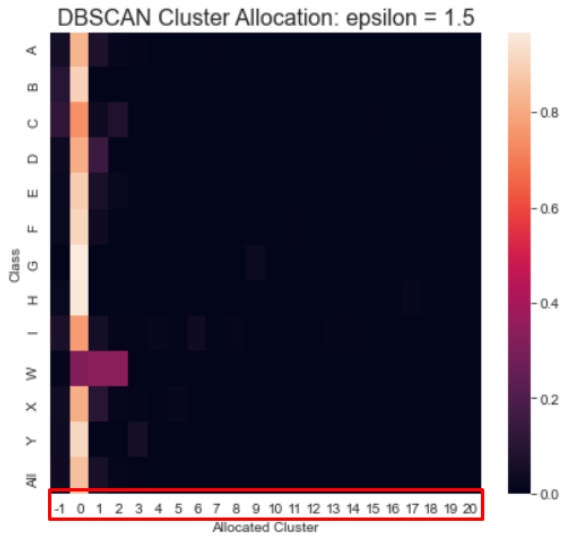
The K means algorithm was the second model used. Max iterations and number times the algorithm was run with different centroid seeds were both left at their default values. The K value was chosen to be 12 as that was the number of classes in the dataset.

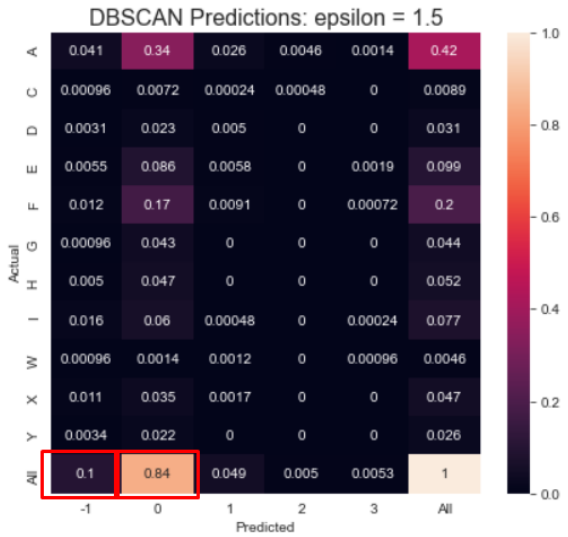


Results

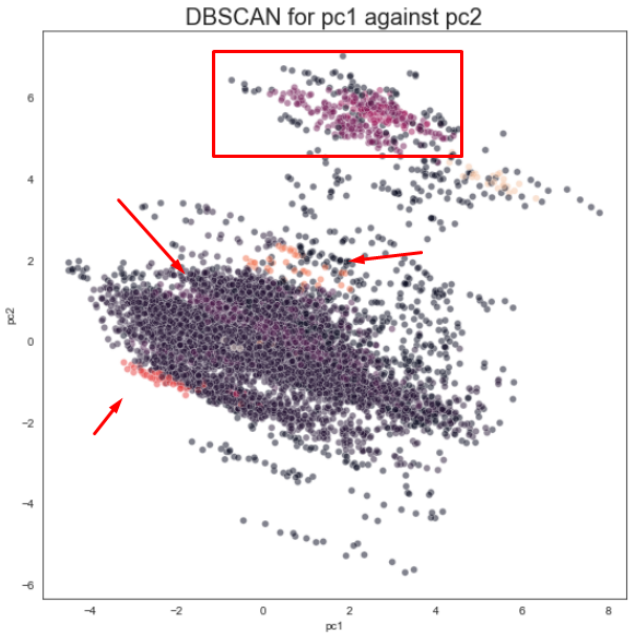
DBSCAN

This model was not particularly good at discerning more than 2 clusters on the dataset. When testing the optimal value for epsilon, the best model identified 21 potential clusters, yet when it came to predicting unseen instances, performed terribly, labelling most of the data in the first cluster and only predicting 3 clusters in total. In total, it allocated 86% of the training set to the first cluster, and predicted that 84% of the test data belonged to that group.





When visualising this data using the principal components calculated earlier, it can be seen that there was an improvement over the default hyperparameters, however, not a particularly useful or significant difference.

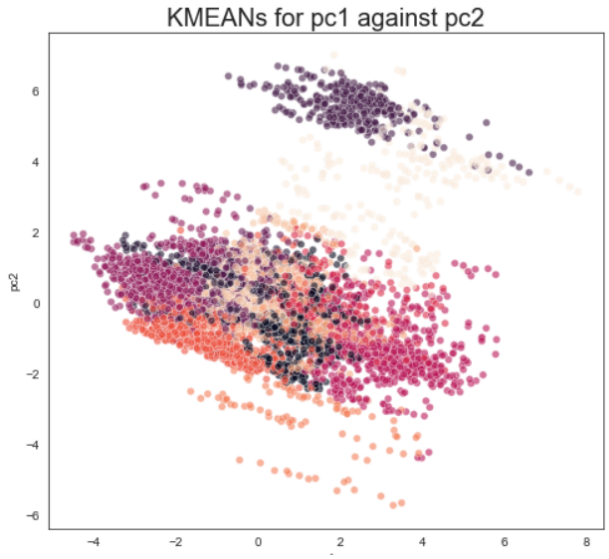


K Means

K Means was not very good for this data set as it struggled with identifying which record belonged in the various clusters. Since there was no identifiable pattern within the clusters, the algorithm was unable to separate the records accurately. This was made worse by

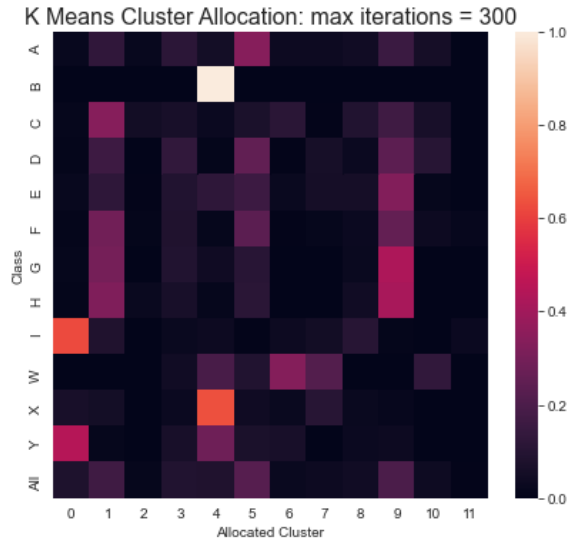
the fact that several of the average values used in the algorithm were located almost on top of each other.

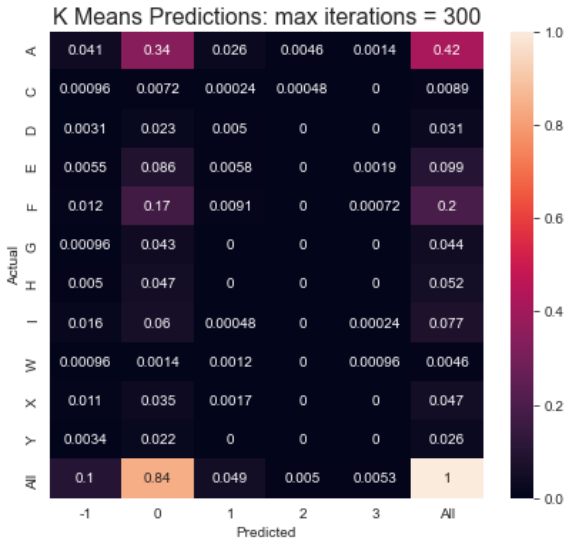
As with the previous algorithm, K means did not perform well on this data set. While it was able to successfully run the algorithm the results were wildly inaccurate and it was clear that the algorithm was struggling.

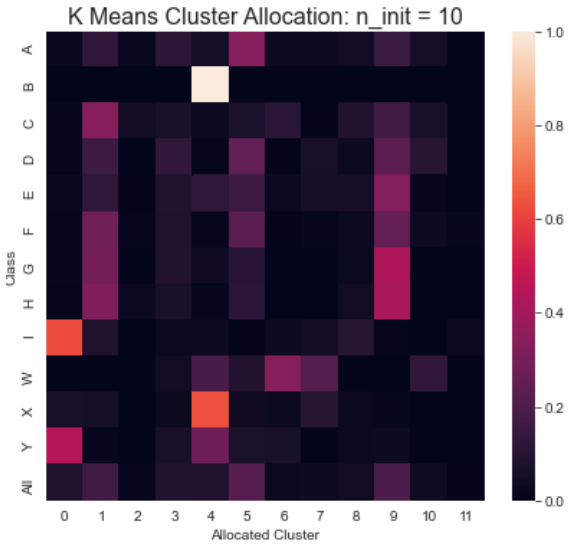


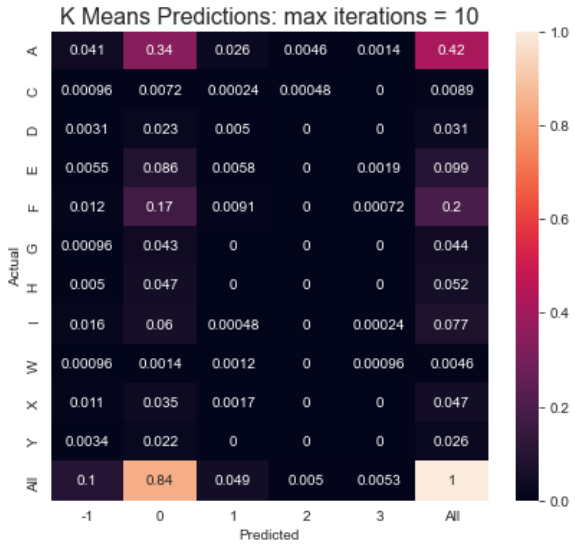
Principal Component Analysis was used to visualise clusters. Unlike DBSCAN, there were plenty of different clusters forming on the graph. However, these clusters were inaccurate and did not represent the true spread of the data.

Changing the hyperparameters of max_iter and n_init had no effect on the effectiveness of the algorithm at all.









Discussion

DBSCAN

DBSCAN struggled identifying the correct number of clusters based on the target class. More hyperparameter tuning could potentially have been done with the min_samples value, however, fundamentally it seems as if this is the wrong clustering algorithm to use on this data.

K Means

While the raw number of clusters could be manually imputed in the algorithm, K Means was unable to accurately predict which record belonged in which cluster. There were simply too many widespread data entries for K means to become an effective algorithm for this data set.

Data Preparation

Data preparation could have been attempted differently, without the exclusion of outlier values, however, we hypothesise that this may not have positively benefited the DBSCAN model, as those values would probably have been identified as further noise. It was realised that the outlier values may have been drop-cap initials or decorated letters, a feature of historical texts. This would explain some of the huge variation within the standard deviation of some features, however, this would still have been difficult to deal with, without raw, unstandardised data.

(Image: Getty images 2021)

Visualising the data to identify discernible clusters in multidimensional data is difficult, however, there are methods other than Principal Component Analysis that could've been explored.

Finally, it seems that K Means clustering worked slightly better to model this data, where the advantage lies in being able to determine the number of clusters to find. It was difficult tuning DBSCAN to find the appropriate values.

Conclusion

Clustering does not seem to be a useful algorithm for this particular dataset, where K Means clustering performed better than the DBSCAN clustering algorithm. In future studies, alternate data preparation with discernment between decorative, drop down letters and an equal representation of the target classes could improve the success of these models.

References

Data set

C. DeÂ Stefano, M. Maniaci, F. Fontanella, A. ScottoÂ diÂ Freca, Reliable writer identification in medieval manuscripts through page layout features: The 'Avila' Bible case, Engineering Applications of Artificial Intelligence, Volume 72, 2018, pp. 99-110, viewed 6 May 2021 <https://archive.ics.uci.edu/ml/machine-learning-databases/00459/>