***REPORT***

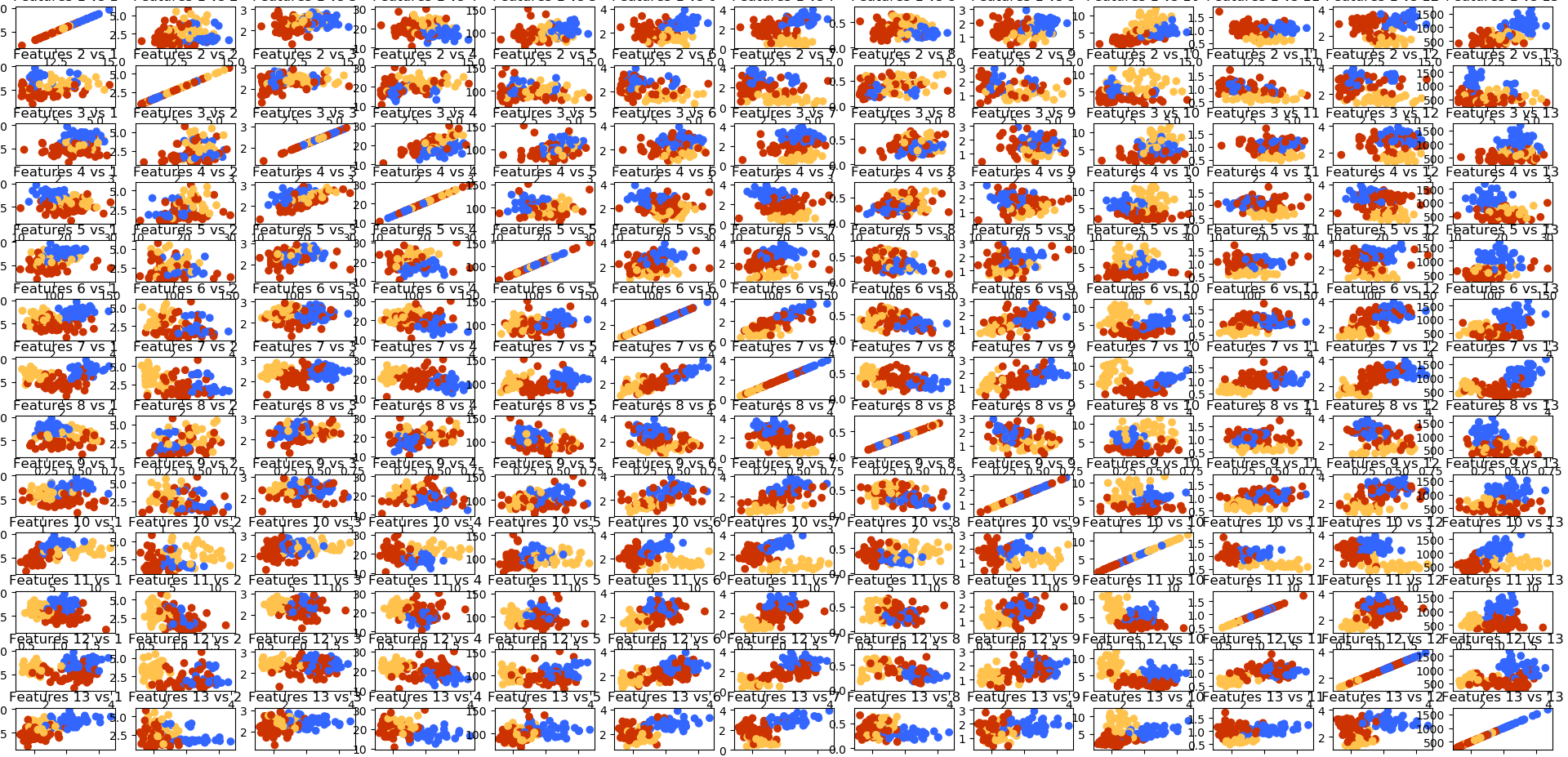
*Marc Goulding*

*Erich Reinholtz*

**FEATURE SELECTION AND THE KNN**

The first stage of our coursework was plotting the features against each other, similarly to the labs. After doing so, we have plotted the resulting figure, which we have provided below:

Scatter plot of the features:



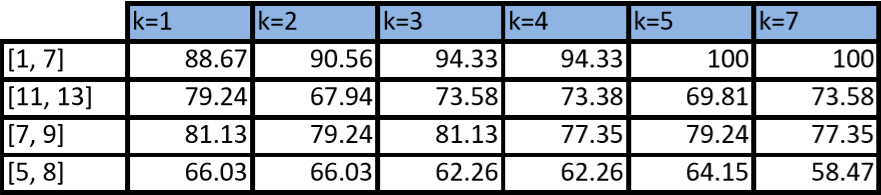
Based on further measurements, using the Knn algorithm, we have decided to opt for [1, 7] as the chosen pair of features. We have used them consistently on the other tasks in our project.

The first step in choosing the most appropriate features was based on a number of trials, to help gain a reasonable amount of perspective over the behaviour of the classifier.

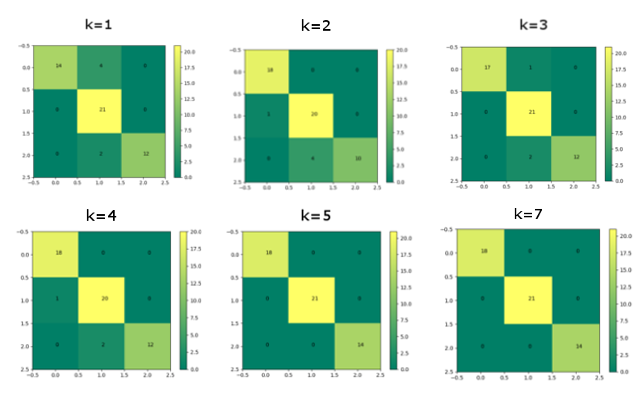
We first have tried to naively calculate the accuracies for the default value of k and pick the pair of features with the maximum value as a result. However, this has proven to not be a good approach.

For the pairs of features where the separate data points belonging to different classes intertwine a lot, the accuracy may even decrease the more neighbours we use. That is because, the more chaotically the points are displayed, the harder it is to tell which group one of them should belong to. In the case of our selected features, where all datum of a certain kind are, mostly, grouped into closely-tightened clusters, enlarging the value of k helps increase the accuracy drastically. The decisive moment for choosing to select features 1 and 7 was when, whilst using them, we achieved 100% accuracy using at least 5 neighbours. This proved that, the more information we’d feed into our classifier, the more efficient it becomes.

This chart features a few of the measurements we undertook while looking for the most appropriate pair of features to select.

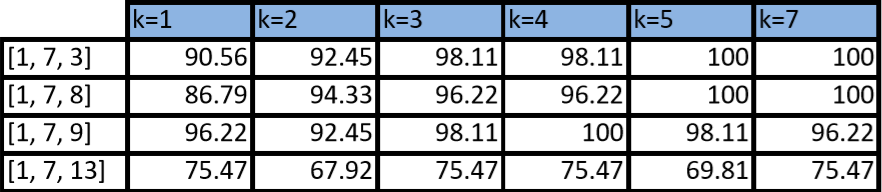


Next up, we have added the corresponding confusion matrices, which show the appropriate behaviour. The higher the accuracy, the less wrongly-labelled entries are. Obviously, in the cases where the accuracy is 100%, only the main diagonal values are non-null.

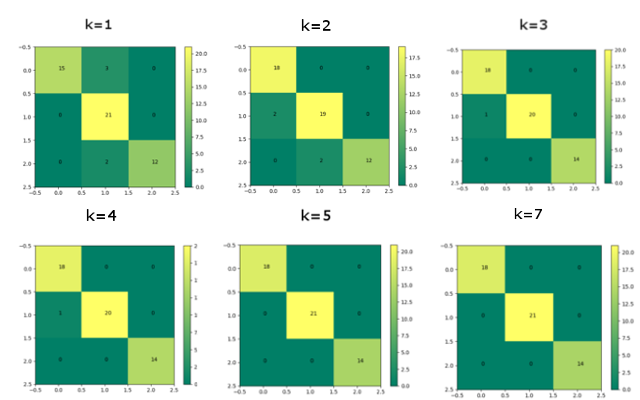


**KNN 3D**

This chart showcases different variations of the accuracy depending on various third features, selected for the Knn 3D function. Eventually, we opted to use feature number 3 as our third feature.

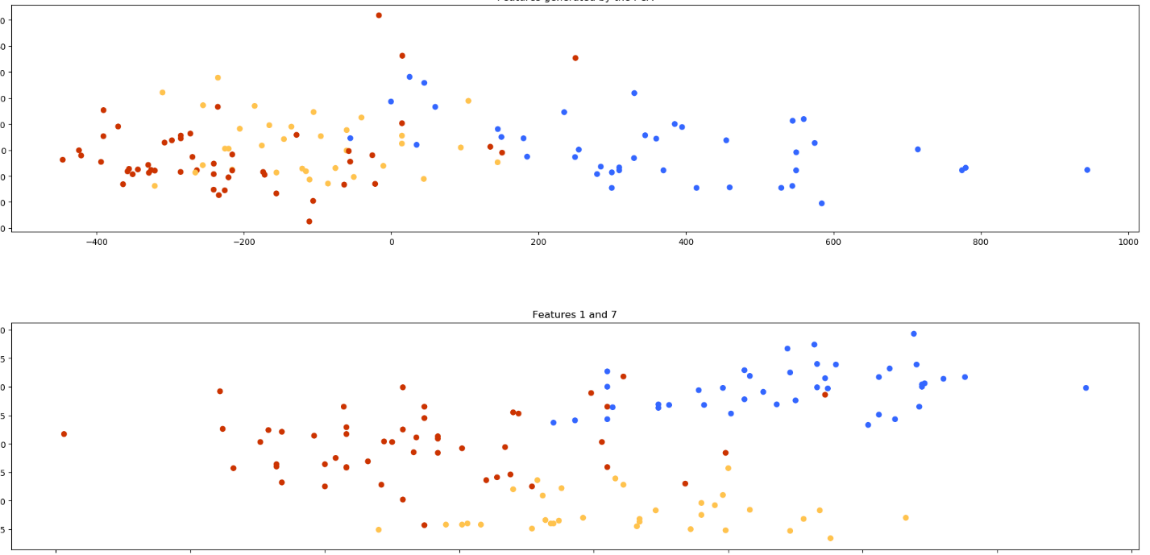


We’ve also plotted the confusion matrices for this function, similarly to the case of the regular KNN.

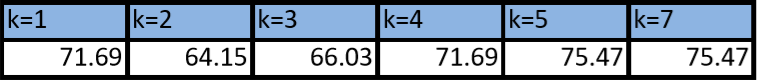


**PCA**

After comparing our own PCA, developed during the lab sessions, with Scipy’s PCA, we have decided to use the latter for this task. Below, we have provided the corresponding scatter plot, followed by another one, which showcases the features we manually chose during the first task.



After using Scipy’s PCA to reduce the data set and applying our Knn algorithm, we have measured the accuracy depending on the number of neighbours used. We have written down the obtained results below:



Furthermore, as always, we have produced the confusion matrices for this classifier.

