CSE 5243 Introduction to Data Mining

Lab 2: Introduction to Scikit-Learn
(Executive Summary)

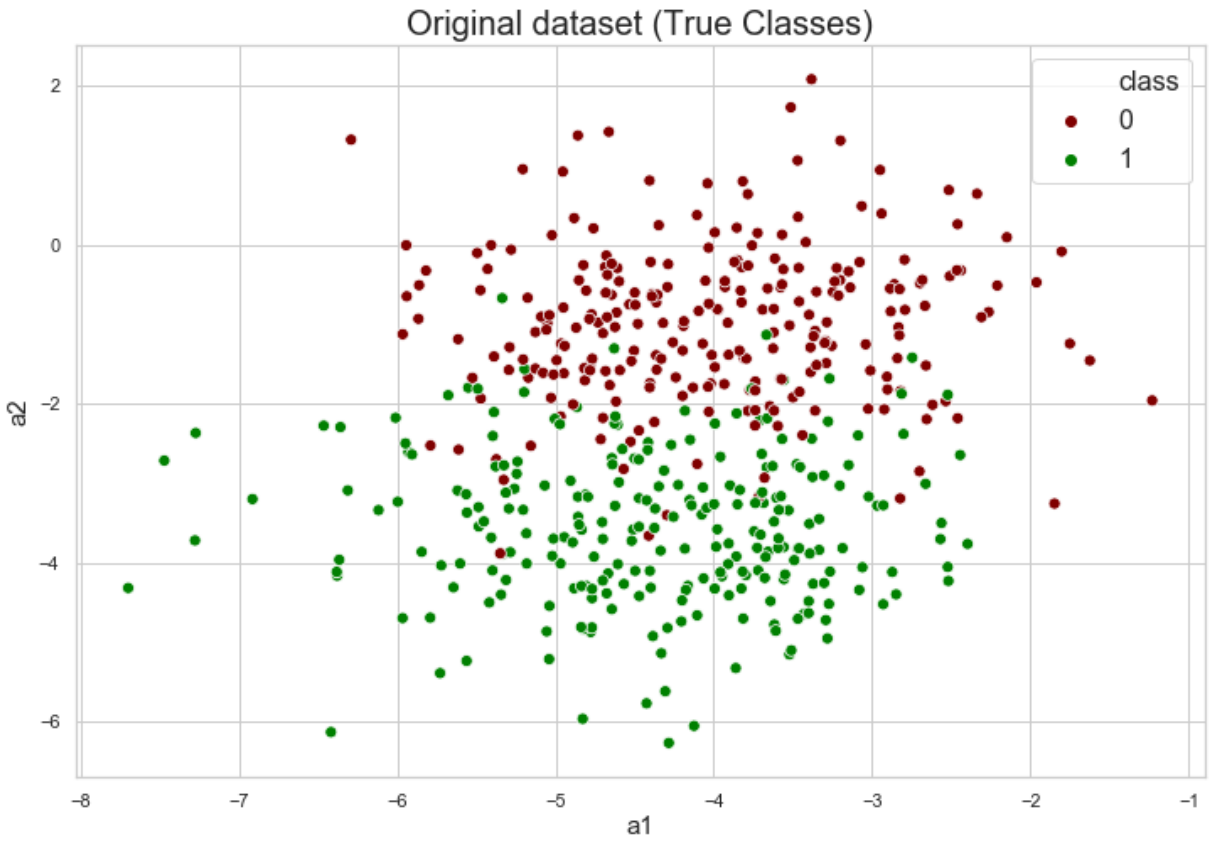
Bharat Suri (suri.40@osu.edu)
Nithin Senthil Kumar (senthilkumar.16@osu.edu)
Taught By: Michael Burkhardt
Date: 09/27/2019

Introduction

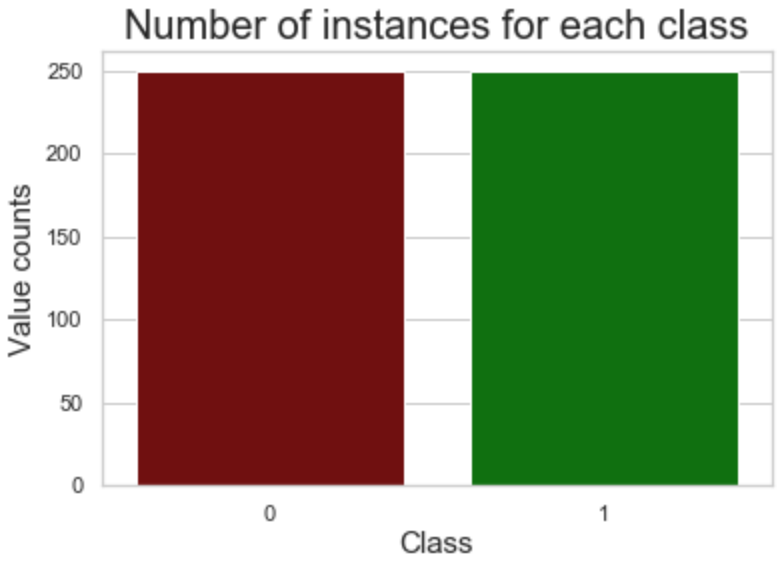
For this lab, a synthetic dataset described by two attributes was used. These attributes, namely a1 and a2 describe each record and possibly share a relationship either individually or together with a third attribute - the class label which is binary for the given data set. The task is to identify the relationship between the attributes and the class label and use this relationship to predict previously unseen records. In order to do this, the original dataset was split into two parts, the train and test set. A k-Nearest Neighbour classifier was used to train a model on the train set and then predict the classes of the data points in the test set.

Visualization

The original dataset is visualized as a scatter plot below. On the x-axis is the attribute a1 and on the y-axis is the attribute a2. The class label is shown in the legend. One observation made by simply looking at the distribution of the data points was that not much information was provided when considering only a1 as the feature. This is because if the data points are projected onto the x-axis, it will not be possible to distinguish the two classes. However, the same does not apply to a2. It can be seen that somewhere around a2=-2, although not perfectly, there is a boundary that separates the two classes.

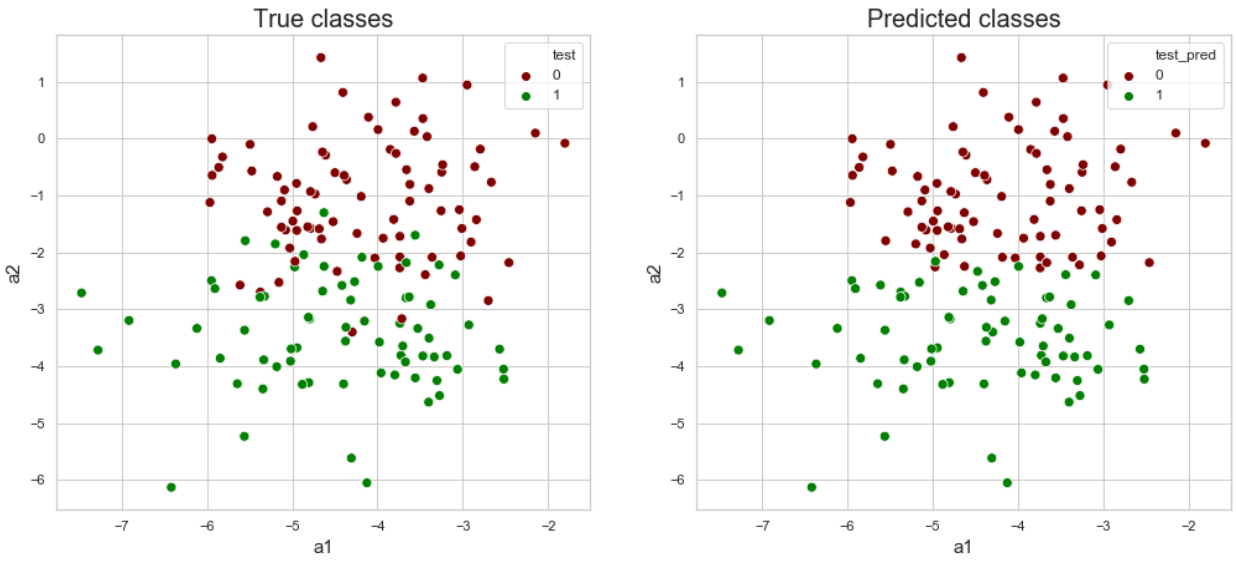


Another plot to see the number of instances of each class was used to check if there was any sort of class imbalance. The plot below shows the value counts for the two classes.



Since the data is synthetic, it appears to be balanced as there are an equal number of records belonging to each class.

Below are the scatter plots of the true and predicted labels of the test set. The records belonging to Class 0 are represented by maroon dots and the records belonging to class 1 are represented by green dots.



Findings and Conclusion

By using the built-in "score" method of the KNN classifier, an accuracy score of **0.92** was observed in the training set and **0.87** in the test set. The accuracy measure on the test set was **lower** than that of the training set, i.e. the model **performed worse** on the test set. This is in agreement with the intuition that predicting the correct class of unseen data is more difficult than that of data that the model has already seen during training. However, this is not the only case because, during different runs of the model, the accuracy measure was also different. Apart from using different hyperparameters, such as the value of k, this can also be explained by the fact that the test-train split performed in Scitkit-Learn is random.

To interpret the results, it is necessary to note from the scatter plots that the boundary between records of the two classes is not perfect. Therefore, it is likely that there will be misclassified points. These points are the ones that have more neighbours of the opposite class nearer to them. Furthermore, the accuracy would also depend on the records that lie in the test set and the train set. Since this split is random each time, the built-in score method returns marginally different accuracies on the test set for each run of the model. This is because if a test set, chosen at random, has more data points with either very high or very low value of a2, then the model will perform better when predicting the classes of data points in the test set. On the other hand, if it contains more data points in the gray area near the separating boundary, then chances are high that the model will perform worse. This means that those points are harder to classify.