Data Set Categorisation

Charalambos Charalambous

Summary

Through the use of some data
we applied some clustering algorithms in both java and python in order to get some results and make a proper categorization of the data with as much accuracy as possible.

Introduction

In this work our aim was to use some data that were defined and given by the teacher with the ultimate goal and purpose to work with some clustering techniques and algorithms and draw some conclusions and generally understand the concept of clustering and how it helps us to better understand them and to make some predictions based on their characteristics.

Description of the dataset

The data I used was from the dermatology dataset, which was in .data format and had to do with various diseases of dermatology and some other characteristics such as age and if there was a family history and each patient was categorized in some class based on whether he had each disease and to what extent and depending on the characteristics and his age. I had about 367 patients and they were all categorized into 6 classes.

Pre-edited from

As for the pre-processing steps, I worked with both languages, both with
java as well as python.As far as java is concerned because my file was in .data I
I had to do it in a format where it could be read so I did the following steps.I took all the data from the file and imported it into a table in excel
so that they are saved in CSV format.Then in java through the code it reads the CSV file and takes all its data and modifies it to form it in arff format so that I can then run algorithms through the weka api because of the weka
reads better of this kind
archives.Now as far as python is concerned things are even simpler because through Collaboratory we can simply import the data both as xlsx and as CSV and I chose the first file type and then we load the file so that I can

then choose which algorithms to use.

Categorisation algorithms

The categorization algorithms that I used with both languages are three.Let me start with the one that everyone knows best, which is KNN. O
k-nearest neighbours (KNN) algorithm is a simple, easy to implement supervised machine learning algorithm that can be used to solve problems classification and regression. I used KNN because it is simple and easy to implement.
It is also not necessary to create a model and because this algorithm is
flexible. Of course It can be used for classification, regression and search .But as it turned out and when I worked on it, the disadvantage of the algorithm becoming much slower as the number of examples and/or predictions increases and that it does not give the best accuracy when dealing with a dataset that has many characteristics and the data has to be categorized based on them.

Another categorization algorithm I used is Logistic Regression. Logistic regression is a statistical
model which in its basic form uses a logistic function to model a
dependent variable. I used the
Logistic Regression performs well when the dataset is linearly separated. Logistic regression is also easier to
application, interpretation and very effective training. Even the Accounting
Regression not only gives a measure of how important a prediction is but also the
direction of the correlation. And as it turned out worked very well with my data.

The last algorithm I used is the Random forest classifier (RFC).The RFC is a classification algorithm consisting of several decision trees. It uses randomization and feature randomness when
construction of each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any single tree. I used RFC because it is comparatively less affected by noise, it can automatically handle missing values.
It also works well with both categories and continuous variables and can be used to solve both classification and regression problems. Furthermore Random Forest is based on the bagging algorithm and uses the Ensemble Learning technique. That is it creates so many trees in the subset of data and combines the output of all the trees. In this way it reduces the problem
placement in the decision trees and also reduces variation and thus improves accuracy.

Methodology

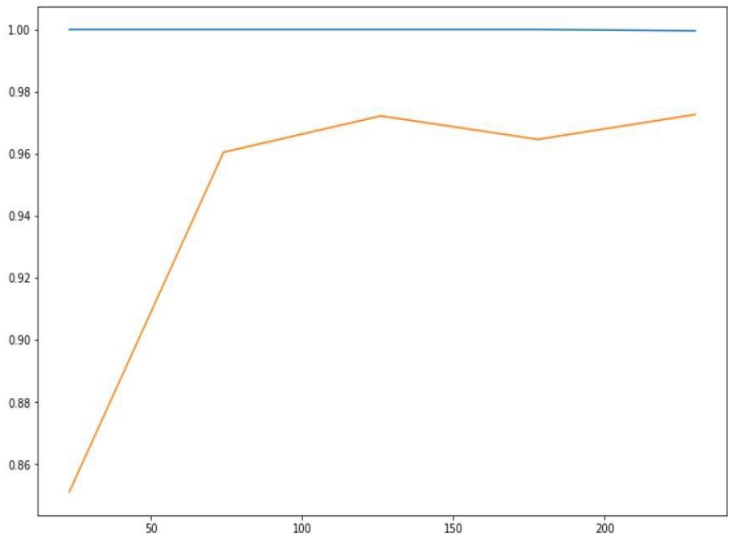
Well before I start building and using the categorization algorithms you will see in the Python code that I am going to build train and test data
. In a dataset, a training set is applied to build a model, while a testing (or validation) set is applied to validate the model that has been built. Basically I am trying to create a model to predict the test data. So I use the training data to fit the model and the test data to test it. The models that are created are to predict

unknown results called as a test set.

Now as for avoiding overfitting . Overfitting refers to a model that models the training data too well. Overfitting occurs when a model learns the details and noise in the training data to the extent that it negatively affects the model's performance on new data. Overfitting is bad because: The model has extra ability to learn random noise in the observation. To mask the noise, an overfit model stretches and ignores areas not covered by data. Consequently, the overfit overfit has an overfit that overlaps the data that is not covered by the data.
Model makes bad predictions everywhere except the training set. Some techniques I have used to reduce overfitting or limit it are by increasing the number of samples or reducing the number of
features as you will see e.g. in the RFC where I used the max_features variable.Even the best combination of parameters is a good solution.

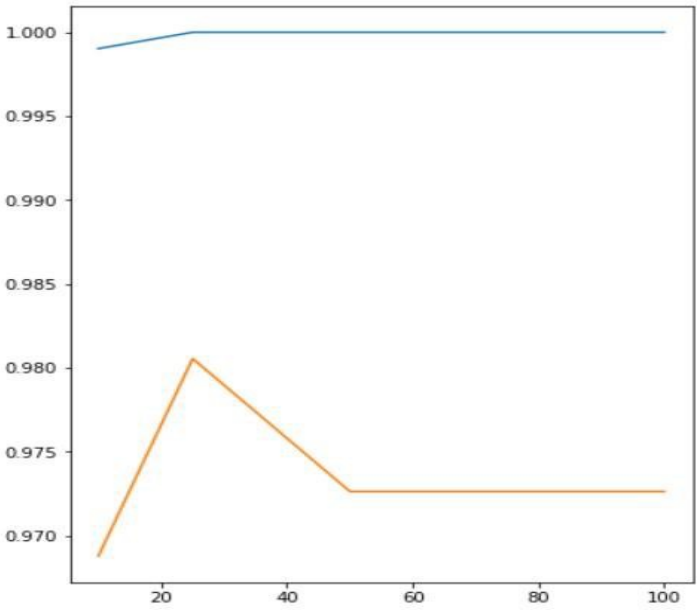
Experimental Evaluation

As you will see in two of the three algorithms we had a very good accuracy rate because they worked and



applied too well with the data

.



To achieve this we altered some of the parameters of the algorithms. Like for example the ordering parameter that I tweaked in logistic regression.Even the results were helped by both the 'n_neighbours' in KNN and the 'n_estimators' in RFC .But in general I used GridSearch to get the best possible parameters or the best combination of them. I also used F1 measure in KNN but as you will see and it was shown the score was not high enough which confirmed that KNN is not the best for our data .

Conclusions

Categorization is a difficult process there is no note on which algorithms will work best with your data, so what you have to do is to run them and the results will show it.For me my file had many characteristics for each patient I had to

become more laborious process than any algorithm to categorize it .As it was shown by the use of test and train data the logistic regression and random forest gave the best result and the highest accuracy rate .The most difficult part has to do with overfitting that you try to eliminate it as much as possible by modifying either the characteristics or take a larger sample of data .

Literary sources

Books :

- Python Data Science Handbook
- DataClassification: Algorithms and Applications
- Data Mining: The Textbook
- Computational Methods of Feature Selection

Websites :

- **https://www.varonis.com/blog/dataclassification/**
- **https://digitalguardian.com/blog/whatdata-classification-data-classificationdefinition**
- **https://www.forcepoint.com/cyberedu/data-classification**
- **https://dl.acm.org/doi/book/10.5555/2535015**
- **https://www.edureka.co/blog/classificati on-algorithms/**
- **https://www.sciencedirect.com/topics/co mputer-science/classification-algorithm**