Project Report

Artificial Intelligence (Minor)

Project title:

Glass Identification

Group members:

Syed Adil Hasan Naqvi (124165)

Umar Rehman (244475)

Submitted to:

Sir Farhan Khan

Date:

07-Aug-19

School of Electrical Engineering and Computer Sciences
National University of Sciences and Technology
H 12, Islamabad

The Problem

Often at a crime scene among the debris, shards of different types of glass can prove to be useful evidence but only if they can be identified correctly. Methods of correct identification of glass can prove very useful in criminology and forensic sciences.

The objective of this project is to train different models to identify different types of glass based on their composition.

Dataset

Quick characteristics

Attributes	9
Classes	6
Size	214
Format	All numeric
Source	UCI Machine Learning Repository, CA: University of California (archive.ics.uci.edu)

Attributes

Following are all the features of the dataset:

1. RI: refractive index
2. Na: Sodium
3. Mg: Magnesium
4. Al: Aluminum
5. Si: Silicon
6. K: Potassium
7. Ca: Calcium
8. Ba: Barium
9. Fe: Iron

Classes

Class	**Examples**
building_windows_float_processed	70
building_windows_non_float_processed	76
vehicle_windows_float_processed	17
containers	13
tableware	9
headlamps	29

The Solution

Create a model that takes glass composition (attributes) as inputs and tells its type (class) as output.

Now the question remains as to what model to use that can achieve this. For this project, we chose three methods to identify glass.

1. K Nearest Neighbors
2. Support Vector Machines
3. Neural Networks

The choice of these methods was somewhat arbitrary. The main purpose here (apart from identifying glass of course) was to get a better understanding of the chosen methods, understand their working, see how their performance changes by tweaking their parameters and try and find the reasoning behind their (good or bad) performance.

Python was used as programming language and following libraries were used:

- pandas
- sklearn
- time

Preparing data for training/testing

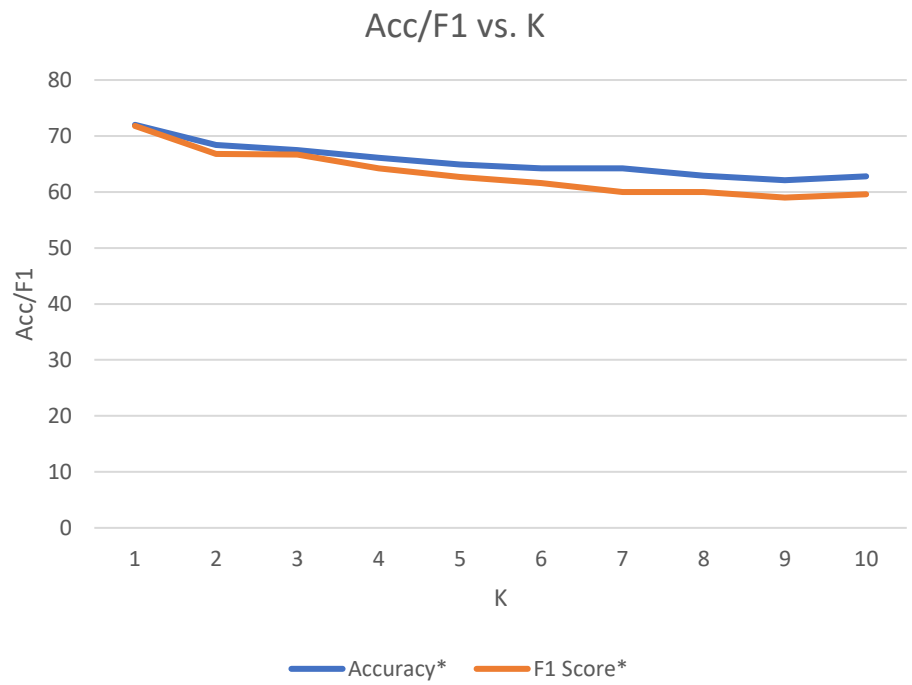
To make the comparison of the models better, all three models were to be fed the exact same training and testing data (by controlling random seed) and only then their performance was compared. This was to ensure that random sampling does not affect the performance of the model.

Training	80%
Testing	20%
Random sampling	Yes
Stratification	Yes

There was a great disparity in the number of examples of each class. To make sure every class had a proportionate representation in the training and testing data, stratification was used.

K Nearest Neighbors

The most important parameter in KNN is the value of K. This was chosen by testing different values of K against their respective accuracies and F1 scores.



The highest accuracy and F1 score were recorded when K was equal to 1 and hence a value of 1 was chosen for K. Euclidian distance was used in this model.

Performance metric	**Value achieved**
Accuracy	72.0%
F1 score (weighted average)	71.4%
Runtime	1.17s
Seed used	1

Support Vector Machine

sklearn SVC (support vector classifier) was used with following parameter values:

Parameter	**Value**
C	1
Kernel	rbf
gamma	auto

The rest of the parameters were left as default (available in sklearn documentation).

The values of the above parameters were chosen by testing them against their respective accuracy and F1 score and the best ones were chosen.

The results are as following:

Performance metric	**Value achieved**
Accuracy	81.3%
F1 score (weighted average)	76.8%

Seed used	66
Runtime	1.03s

Neural Network

sklearn MLP (multilevel perceptron) was used with following parameter values:

Parameter	**Value**
no. of hidden layers	1
neurons in hidden layer	20
learning rate	0.025
activation function	logistic
max iterations	1000

The rest of the parameters were left as default (available in sklearn documentation).

The values of the above parameters were chosen by testing them against their respective accuracy and F1 score and the best ones were chosen.

The results are as following:

Performance metric	**Value achieved**
Accuracy	81.3%
F1 score (weighted average)	80.85%
Seed used	93
Runtime	1.68s

References

1. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science
2. http://neuroph.sourceforge.net/tutorials/GlassIdentification/GlassIdentificationUsingNeuralNetworks.html
3. https://www.kaggle.com/eliekawerk/glass-type-classification-with-machine-learning
4. https://scikit-learn.org/stable/index.html