UNIVERSITÀ DI PISA

Data Mining and Machine Learning

Educational Version Project

2020

D. M. Maggi

M. Gomez Gomez

July 26, 2020

Contents

1.1 WikiHow

The dataset is made up of WikiHow articles.
In WikiHow, each article belongs to one and only one of 19 macrocategories: *Arts and Entertainment, Cars and Other Vehicles, Travel, Work World, Relationships, Philosophy and Religion, Family Life, Finance and Business, Computers and Electronics, Food and Entertaining, Home and Garden, Youth, Education and Communications, Health, Hobbies and Crafts, Personal Care and Style, Sports and Fitness, Holidays and Traditions, Pets and Animals.*
Each macro-category may have several subsections, but this hasn't been taken into account during our work, as WikiHow's taxonomy has a strict tree-structure.
Each article is characterized by:

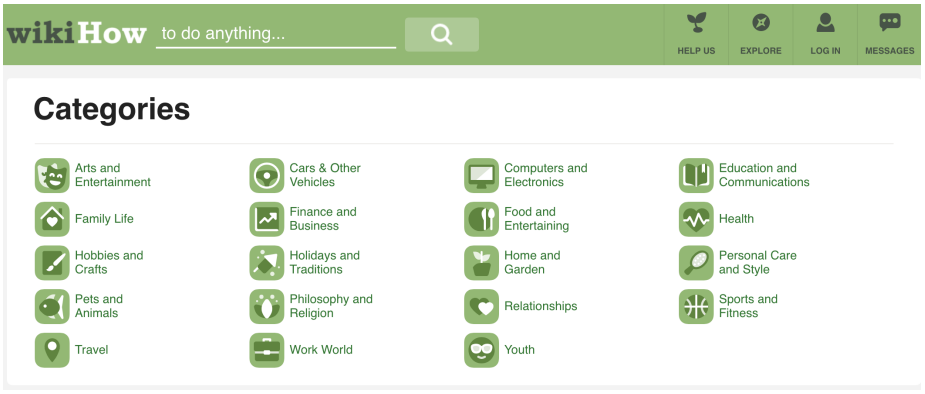

Figure 1.1: WikiHow's main categories

- its *title*, always beginning with the prefix "How to..."
- a short *summary* at the top of the page, containing the main arguments of the article.
- the *text* of the article, divided into numbered paragraphs.

At the moment, the categorization of the articles is entirely manual: the content creator has the responsibility to label the submitted work. A team of volunteers routinely double-checks the newly created or modified articles.
Our interest lies in providing an automatic tool for classifying a new document into one of the 19 macro categories.

1.2 Dataset building

1.2.1 Scraping

Data have been scraped using two Python scripts, $parser.py$ and $link_h arvester.py$. For each category, the relative paths to subcategory pages have been extracted and saved in auxiliary .txt files (which can be found in parser/texts) in order to facilitate the scraping procedure.
Below an example of the category-subcategory relationship.

```
/Category:Arts-and-Entertainment
   /Category:Amusement-and-Theme-Parks
   /Category:Carnivals
   /Category:Disneyland-and-Disney-World
   /Category:Roller-Coasters
   /Category:Water-Parks
   /Category:Artwork
   /Category:Art-Appreciation
   /Category:Art-Collection
   /Category:Art-Equipment
   /Category:Art-Journals
   /Category:Art-Media
   /Category:Art-Studies
   ...
```

The subcategories has been retrieved according to the following procedure:

```
FOR EACH CATEGORY IN \textit{CATEGORY LIST}:
    GET THE HTML PAGE OF THE SUBCATEGORY
    PARSE THE HTML
    FIND THE "subcats" <div>
    GET THE LINKS
    FOR EACH CATEGORY.TXT
        WRITE THE SUBCATEGORY RELATIVE PATH
```

For each subcategory, we proceed to scrape WikiHow pages according to this general scheme:

```
FOR EACH CATEGORY IN CATEGORY LIST:
    PARSE THE CATEGORY.TXT FILE
```

```
FOR EACH SUBCATEGORY;
    PARSE ARTICLE PAGE
    EXTRACT TITLE; SUMMARY; DESCRIPTION
    PRE-PROCESS TITLE; SUMMARY; TEXT
    SAVE [TITLE; SUMMARY; DESCRIPTION; CATEGORY] TO
    CATEGORY.CSV FILE
```

In order to balance the dataset, from each of the 19 separate .csv files 200 rows have been extracted and merged into a single *AllCategories.csv* file, which was our balanced, final dataset.

AllCategories

TITLE	SUMMARY	TEXT	CATEGORY
talent multipl area	set increas talent abil multipl disciplin audaci endeavor also feasit	practic whatev tri talent know practic especi true hope talent multipl ar	Arts-and-Entertainment
make illumin manuscript	middl age book written hand meant imag paint hand illumin manu	gather piec parchment classic manuscript made thick white parchment	Arts-and-Entertainment
peopl watch	peopl watch art cultur old citi like pari neur french word someon s	find crowd street cafe cafe classic peopl watch destin mani peopl visit	Arts-and-Entertainment
reduc entertain expens	spend money movi special event cost surpris amount time averag	purcha cabl altern first step reduc home entertain cost ditch cabl instea	Arts-and-Entertainment
support art	art allow peopl public express fun creativ meaning way art music	donat money nation art organ help big scale creativ project need fund s	Arts-and-Entertainment
contact art galleri	reach art galleri easi take lot courag confid organ well strong sens	arrang recent art sampl physic digit portfolio collect realli high qualiti pi	Arts-and-Entertainment
read ulyss	come ulyss consid mani second hardest book english languag m	understand ulyss learn read ulyss know get ulyss compri episod episo	Arts-and-Entertainment
wear cosplay wig	cosplay wig worn make intend charact come life cosplay wig high	make sure wig right size head may alreadi figur wig right size head cas	Arts-and-Entertainment
good stage presenc	good stage presenc crucial creat engag last perform talent practi	take everi opportun practic get practic make feel confid big show venu	Arts-and-Entertainment

Figure 1.2: Snapshot from AllCategories.csv

The pre-processing phase is analyzed in greater detail in the following section.

1.2.2 Pre-processing phase

After gathering all of the documents making up our dataset, we have applied to the instances the following operations:

- lowercase the text and remove everything that isn't a letter (using *re* package).
- remove all the English *stop-words*, that is, those considered too common to be significant, present in the text (using *nltk.corpus*).
- reduce words to *stems*, that is, separate suffixes from radixes of the words (using *SnowballStemmer* from *nltk.stem.snowball* package).

After this initial clean-up, the newly-balanced dataset *All_Categories.csv*
INSERIRE SCHERMATE DI SETTING UP MODELLO

1.3 Data Mining

The one posed in this project is a *multi-class classification* problem, because each article can belong to one and only one class. A multi-class model is employed for training for each type of classifier.
Our aim was to compare different classifier and evaluate their results on our dataset. In order to do so, we will compare the results of each classifier with respect to their *precision*, *recall*, *f-measure* and *accuracy*.

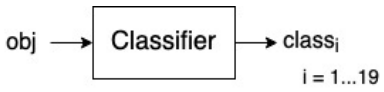

Figure 1.3: Classification pipeline in a multi-class classification scenario

1.3.1 Random Forest

Random Forest is a classification *ensemble method*. The "forest" is made up of an ensemble of *decision tree* classifiers. Each tree depends on the values of a random vector sampled independently and sharing the same distributions as all the other trees in the forest.

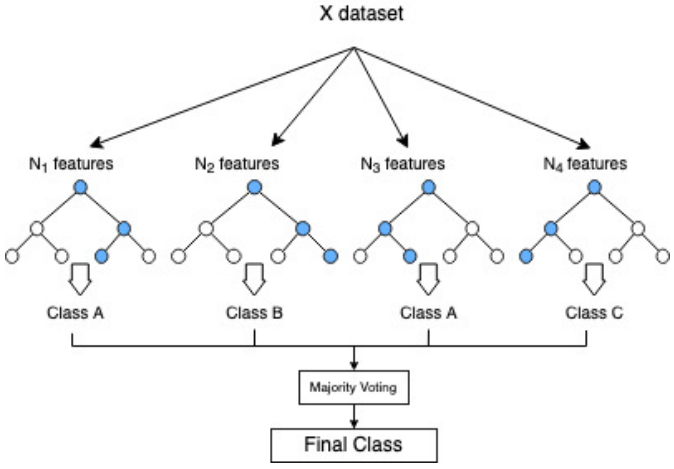

Figure 1.4: Random Forest ensemble voting procedure

During the classification phase, each tree expresses its vote and the class returned globally is the one reaching the highest consensus.

1.3.2 Naïve Bayes

Statistical classifier expressing class membership as a *probability*.
Let X be the *evidence* whose class label is unknown.
Let H be the *hypotesis* that X belongs to class C.
Bayes theorem helps computing $P(H|X)$, that is the probability that X belongs to class C, given its characteristics:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

The aim of a Bayesian classifier will be to determine the class C_i having the higher posterior probability $P(C_i|X)$.
In the context of multi-class text classification, this can be reformulated as finding

$$\arg\max_k P(class_k|doc_i), \quad i = 1 \dots k$$

In other words, for each class k, the probability that document i belongs to it is computed and the class with the highest value is selected.

1.3.3 J48

It is the Java implementation, used by Weka, of the decision tree algorithm C4.5.
The algorithm builds a decision tree from the training data. At each node, the attribute having the higher information gain is chosen as a splitting criterion.

1.3.4 SMO

The Sequential Minimimal Optimization (SMO) algorithm is employed for solving the quadratic problem arising in the training of Support Vector Machines (SVM).
It employs heuristics to partition the training problem into smaller problems that can be solved analytically.

1.3.5 Validation

The results of the various classifiers employed have been compared using a k-fold cross validation, with k=10.
This method randomly partitions the dataset in k mutually exclusive subsets,

of approximately equal numerosity.

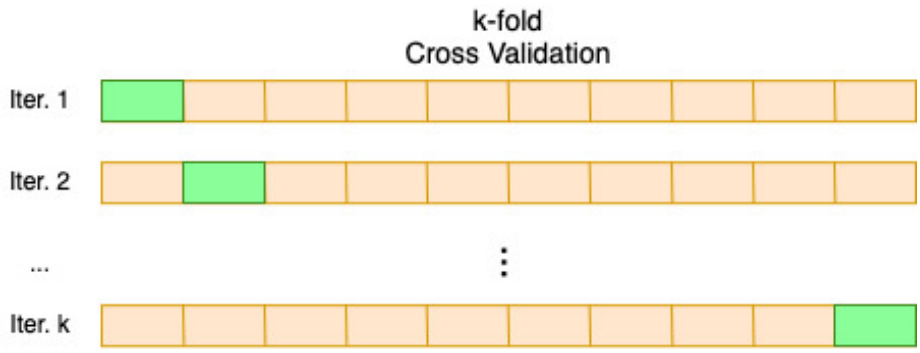

Figure 1.5: k-fold Cross Validation pipeline

At each iteration, subset D_i is used as a *test set* and the $k-1$ remaining ones as *training sets*.

1.4 Implementation

1.4.1 Use Case Diagrams

1.4.2 UML Class

Create.java

1.4.3 Libraries used

Weka API

1.4.4 Models

Models have been derived using WEKA.

1.4.5 Code structure