Damegender Manual: Counting Males and Females in Internet Communities

for version 0.2.8, 06 May 2020

David Arroyo Menéndez (`davidam@gnu.org`)

This manual is for Damegender (version 0.2.8, 06 May 2020), which is an example in the Texinfo documentation.

Table of Contents

1 Introduction 1

2 Installation 2

3 Commands 3

4 Statistics 7

4.1 Measuring success and error 7
4.2 Principal Component Analysis (PCA) 10
4.2.1 Counting features in names 10
4.2.2 Choosing components 11

5 Use Cases 14

5.1 Introduction 14
5.2 Counting males and females in Debian 14
5.3 Counting males and females in Forbes 15
5.4 Deciding for males and females in images 16
5.5 Webscraping and Damegender because we want count scientifics .. 16
5.6 Counting males and females in a git repository 18

Appendix A GNU Free Documentation License .. 20

Index 28

1 Introduction

Damegender is a gender detection tool from the name coded by David Arroyo MEnéndez (DAME).

The gender detection tools from the names are being used usually with commercial APIs. But many countries has been doing efforts in the last years for contribute names and a number of people using each name with Open Data Licenses. So, this software is collecting this effort on an original way (we are using Machine Learning algorithms for predict names that is not appearing in our database).

Damegender is giving measures to compare in any moment our solution with the commercial APIs. So, the user can understand when it's useful to invest money or not depending of the dataset. Damegender allows to the users download a big number of names from a csv file.

This software is written oriented to tests. So you can check the right behaivour of the software with python tests for the classes and methods and with shell tests for the python commands.

Damegender is using Perceval for count males and females in a lot of Internet Communities (wikis, mailing lists, software repositories, bug tracking systems, ...). We shows source for count males and females in different situations (Ex: count-debian-gender.py)

This software is taking into account the power to predict nations and ethnicity from the surnames (Ex: surname.py, surnameincountries.py and ethnicity.py).

2 Installation

Possible Debian/Ubuntu dependencies:

```
$ sudo apt-get install python3-nose-exclude python3-dev dict dict-freedict-eng-spa dict-freedict-spa-eng dictd
```

Now, to install damegender from sources:

```
$ git clone https://github.com/davidam/damegender
$ cd damegender
$ pip3 install -r requirements.txt
```

Now, to install damegender with python package:

```
$ python3 -m venv /tmp/d
$ cd /tmp/d
$ source bin/activate
$ pip install --upgrade pip
$ pip3 install damegender
$ cd lib/python3.5/site-packages/damegender
$ python3 main.py David
```

To install apis extra dependencies:

```
$ pip3 install damegender[apis]
```

To install mailing lists and repositories extra dependencies:

```
$ pip3 install damegender[mails_and_repositories]
```

To install all possible dependencies

```
$ pip3 install damegender[all]
```

Currently you can need an api key from:

- https://store.genderize.io/documentation
- https://gender-api.com
- https://www.nameapi.org/
- https://v2.namsor.com/NamSorAPIv2/sign-in.html

To configure your api key you can execute:

```
$ python3 apikeyadd.py
```

3 Commands

You must start to check tests to understand that all is ok:

```
$ cd src/damegender
$ ./testsbycommands.sh         # It must run for you
$ ./testsbycommandsextralocal.sh  # You will need all dependencies
                                  # with: $ pip3 install damegender[all]
$ ./testsbycommandsextranet.sh    # You will need api keys
```

You can continue check python tests:

Execute all tests:

```
$ nosetests3 tests
```

Execute one file:

```
$ nosetests3 tests/test_basics.py
```

Execute one test:

```
$ nosetests3 tests/test_basics.py:TestBasics.test_indexing
```

If you are in a fresh installation, perhaps you want regenerate by your own risk some files downloaded to understand how it has been generated:

```
$ python3 postinstall.py
```

You can find an big list of commands to execute this shell scripts. Now a detailed execution of some selected examples:

The first command to learn is main.py. You can play now with this command:

```
# Detect gender from a name (INE is the dataset used by default)
$ python3 main.py David
David gender is male
363559  males for David from INE.es
0 females for David from INE.es

# Detect gender from a name only using machine learning (experimental way)
$ python3 main.py Agua --ml=nltk
Agua gender is female
0 males for Agua from INE.es
0 females for Agua from INE.es

# Detect gender from a name (all census and machine learning)
$ python3 main.py David --verbose
365196 males for David from INE.es
0 females for David from INE.es
1193 males for David from Uruguay census
5 females for David from Uruguay census
26645 males for David from United Kingdom census
0 females for David from United Kingdom census
3552580 males for David from United States of America census
12826 females for David from United States of America census
David gender predicted with nltk is male
David gender predicted with sgd is male
David gender predicted with svc is male
David gender predicted with gaussianNB is male
David gender predicted with multinomialNB is male
David gender predicted with bernoulliNB is male
David gender predicted with forest is male
David gender predicted with tree is male
David gender predicted with mlp is male
```

The first Free Software for gender detection tool was created in C language program and you can look for a python version with the name genderguesser. Some people was working in a Free dataset called name_dict.txt with 48500 names. I want to give thanks to this effort with nameincountries.py due to the good work organizing many names in different countries.

```
$ python3 nameincountries.py David
grep -i " David " files/names/nam_dict.txt > files/grep.tmp
males: [’Albania’, ’Armenia’, ’Austria’, ’Azerbaijan’, ’Belgium’, ’Bosnia and Herze-█
govina’, ’Czech Republic’, ’Denmark’, ’East Frisia’, ’France’, ’Georgia’, ’Ger-█
many’, ’Great Britain’, ’Iceland’, ’Ireland’, ’Israel’, ’Italy’, ’Kaza-
khstan/Uzbekistan’, ’Luxembourg’, ’Malta’, ’Norway’, ’Portugal’, ’Roma-
nia’, ’Slovenia’, ’Spain’, ’Sweden’, ’Swiss’, ’The Netherlands’, ’USA’, ’Ukraine’]█
females: []
both: []
```

This Free Software has been developed in the frame of a Phd in the Universidad Rey Juan Carlos I with the Phd director Jesús González Barahona, so I have developed some commands to use Perceval (Free Software where he has done good contributions)

To count gender from a git repository:

```
$ python3 git2gender.py https://github.com/chaoss/grimoirelab-perceval.git -█
-directory="/tmp/clonedir"
The number of males sending commits is 15
The number of females sending commits is 7
```

To count gender from a mailing list:

```
$ cd files/mbox
$ wget -c http://mail-archives.apache.org/mod_mbox/httpd-announce/201706.mbox█
$ cd ..
$ python3 mail2gender.py http://mail-archives.apache.org/mod_mbox/httpd-
announce/
```

Perhaps you don’t know a name, but you have obtained an free key for an api to retrieve it:

```
$ python3 api2gender.py Leticia --surname="Martin" --api=namsor
female
scale: 0.99
```

If you want to know the gender of a good number of names you can download results from an api and save in a file with downloadjson.py

```
$ python3 downloadjson.py --csv=files/names/min.csv --api=genderize
$ cat files/names/genderizefiles_names_min.csv.json
```

Now we are going to learn some commands for measure the successful of our solution:

```
$ python3 accuracy.py --csv=files/names/min.csv
################### NLTK!!
Gender list: [1, 1, 1, 1, 2, 1, 0, 0]
Guess list:  [1, 1, 1, 1, 0, 1, 0, 0]
Dame Gender accuracy: 0.875
```

```
$ python3 confusion.py --csv="files/names/partial.csv" --api=nameapi --
jsondownloaded="files/names/nameapifiles_names_partial.csv.json"
A confusion matrix C is such that Ci,j is equal to the number of obser-
vations known to be in group i but predicted to be in group j.
If the classifier is nice, the diagonal is high because there are true positives
Nameapi confusion matrix:

[[ 3, 0, 0]
 [ 0, 15, 1]]
```

```
$ python3 errors.py --csv="files/names/all.csv" --api="genderguesser"
Gender Guesser with files/names/all.csv has:
+ The error code: 0.22564457518601835
+ The error code without na: 0.026539047204698716
+ The na coded: 0.20453365634192766
+ The error gender bias: 0.0026103980857080703
```

You can generate a lot of logs about errors, accuracies and/or confusion:

```
$ ./logs-accuracies.sh
$ ./logs-confusion.sh
$ ./logs-errors.sh
```

Perhaps you are interested on reproduce experiments to determine features:

```
$ python3 infofeatures.py
Females with last letter a: 0.4705246078961601
Males with last letter a: 0.048672566371681415
Females with last letter consonant: 0.2735841767750908
Males with last letter consonant: 0.6355328972681801
Females with last letter vocal: 0.7262612995441552
Males with last letter vocal: 0.3640823393612928
$ python3 pca-components.py --csv="files/features_list.csv" # To deter-
mine number of components
$ python3 pca-features.py                                   # To under-
stand the weight between variables for a target
```

Now we can go to play with surnames:

```
$ python3 surname.py Gil --total=es
There are 140004 people using Gil in Spain

$ python3 surname.py Lenon --total=us
There are 837 people using Lenon in United States of America
```

```
$ python3 ethnicity.py Smith
In United States of America the percentages about the race of Smith surname is:
White: 73.35
Black: 22.22
Hispanic: 1.56
Asian Pacific Indian American: 0.40
American Indian and Alaska Native: 0.85
Various races: 1.63
```

4 Statistics

In the last chapter we were learning to execute some commands such as accuracy.py, confusion.py, or errors.py, but perhaps you need to understand more theory about statistics to understand why this commands is being interesting for you.

4.1 Measuring success and error

To guess the sex, we have an true idea (example: female) and we obtain a result with a method (example: using an api, querying a dataset or with a machine learning model). The guessed result could be male, female or perhaps unknown. Remember some definitions about results about this matter:

True positive is to find a value guessed as true if the value in the data source is positive.

True negative is to find a value guessed as true if the the value in the data source is negative.

False positive is to find a value guessed as false if the the value in the data source is positive.

False negative is to find a value guessed as false if the the value in the data source is negative.

So, we can find a vocabulary for measure true, false, success and errors. We can make a summary in the gender name context about mathematical concepts:

Precision is about true positives divided by true positives plus false positives

```
(femalefemale + malemale ) /
(femalefemale + malemale + femalemale)
```

Recall is about true positives divided by true positives plus false negatives.

```
(femalefemale + malemale ) /
(femalefemale + malemale + malefemale + femaleundefined + maleundefined)
```

Accuray is about true positives divided by all.

```
(femalefemale + malemale ) /
(femalefemale + malemale + malefemale + femalemale + femaleundefined + maleundefined)█
```

The F1 score is the harmonic mean of precision and recall taking both metrics into account in the following equation:

```
2 * (
(precision * recall) /
(precision + recall))
```

In Damengender, we are using accuracy.py to apply these concepts. Take a look to practice:

```
$ python3 accuracy.py --api="damegender" --measure="f1score" --csv="files/names/partia
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Damegender!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Damegender f1score: 0.9090909090909091
```

```
$ python3 accuracy.py --api="damegender" --measure="recall" --csv="files/names/partial
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Damegender!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Damegender recall: 1.0

$ python3 accuracy.py --api="damegender" --measure="accuracy" --csv="files/names/parti
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Damegender!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Damegender accuracy: 0.8571428571428571

$ python3 accuracy.py --api="genderguesser" --measure="accuracy" --csv="files/names/pa
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Genderguesser!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Genderguesser accuracy: 0.8571428571428571

$ python3 accuracy.py --api="genderguesser" --measure="precision" --csv="files/names/p
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Genderguesser!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Genderguesser precision: 0.9090909090909091

$ python3 accuracy.py --api="genderguesser" --measure="recall" --csv="files/names/part
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Genderguesser!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Genderguesser recall: 1.0

$ python3 accuracy.py --api="genderguesser" --measure="f1score" --csv="files/names/par
-jsondownloaded=files/names/partialnoundefined.csv.json
################### Genderguesser!!
Gender list: [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
Guess list:  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
Genderguesser f1score: 0.9090909090909091
```

Error coded is about the true is different than the guessed:

```
(femalemale + malefemale + maleundefined + femaleundefined) /
(malemale + femalemale + malefemale +
femalefemale + maleundefined + femaleundefined)
```

Error coded without na is about the true is different than the guessed, but without undefined results.

```
(maleundefined + femaleundefined) /
(malemale + femalemale + malefemale +
femalefemale + maleundefined + femaleundefined)
```

Error gender bias is to understand if the error is bigger guessing males than females or viceversa.

The **weighted error** is about the true is different than the guessed, but giving a weight to the guessed as undefined.

```
(femalemale + malefemale +
+ w * (maleundefined + femaleundefined)) /
(malemale + femalemale + malefemale + femalefemale +
+ w * (maleundefined + femaleundefined))
```

In Damegender, we have coded errors.py to implement the different definitions in diffrent apis.

The confusion matrix creates a matrix about the true and the guess. If you have this confusion matrix:

```
[[ 2, 0, 0]
 [ 0, 5, 0]]
```

It means, I have 2 females true and I've guessed 2 females and I've 5 males true and I've guessed 5 males. I don't have errors in my classifier.

```
[[ 2  1  0]
[ 2 14  0]]
```

It means, I have 2 females true and I've guessed 2 females and I've 14 males true and I've guessed 14 males. 1 female was considered male, 2 males was considered female.

In Damegender, we have coded confusion.py to implement this concept with the different apis.

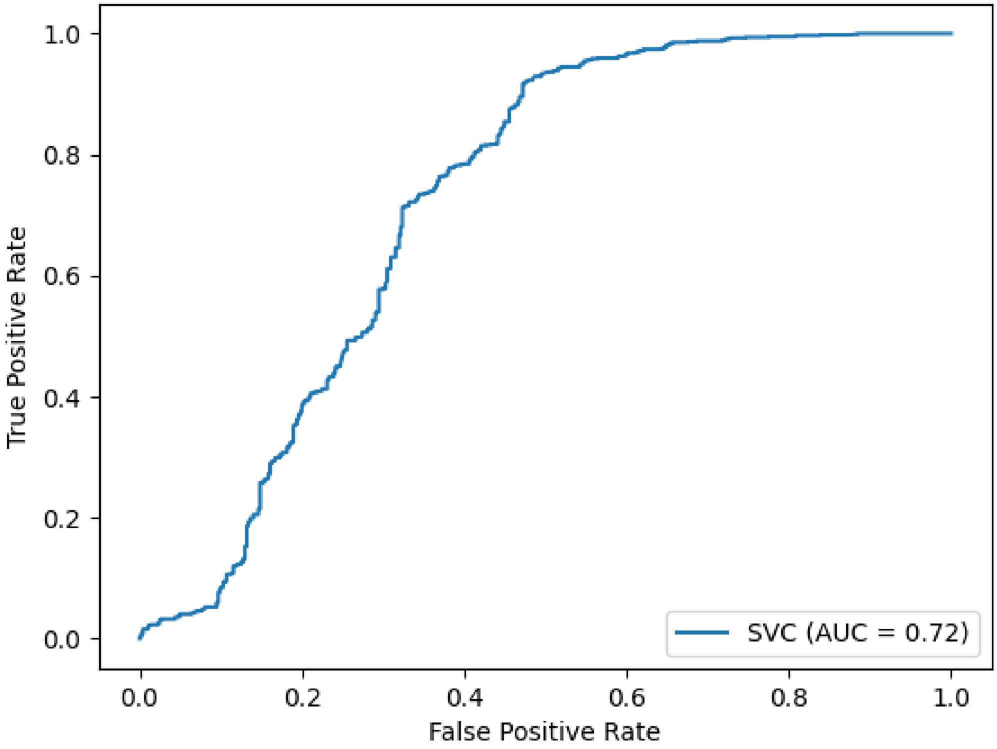
```
python3 confusion.py --csv=files/names/min.csv --api=damegender --jsondownloaded=files
A confusion matrix C is such that Ci,j is equal to the number of obser-
vations known to be in group i but predicted to be in group j.
If the classifier is nice, the diagonal is high because there are true positives
Damegender confusion matrix:

      M   F   U
M  [[ 5,  0,  0 ]
F   [ 0,  1,  0 ]]
```

Similar to confusion is ROC (Receiver Operating Characteristic) is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

In Damegender, you can use ROC relative to machine learning algorithms with the next command:

```
$ python3 roc.py svc
```



4.2 Principal Component Analysis (PCA)

4.2.1 Counting features in names

We have developed a script infofeatures.py with our datasets to visualize data about some features chosen by us.

```
$ python3 infofeatures.py ine
```

Take a look to the results with the different datasets:

Dataset	**Letter A**	**Last Letter A**	**Last Letter O**	**Last Letter Consonant**	**Last Letter Vocal**	**First Letter Consonant**	**First Letter Vocal**
Uruguay (females)	0.816	0.456	0.007	0.287	0.712	0.823	0.177
Uruguay (males)	0.643	0.249	0.062	0.766	0.234	0.771	0.228
Australia (females)	0.922	0.588	0.033	0.272	0.728	0.772	0.228

Australia (males)	0.818	0.03	0.269	0.57	0.43	0.763	0.237
Canada (females)	0.659	0.189	0.005	0.591	0.408	0.838	0.161
Canada (males)	0.752	0.22	0.025	0.54	0.456	0.818	0.181
Spain (females)	0.922	0.588	0.03	0.271	0.728	0.772	0.228
Spain (males)	0.818	0.03	0.268	0.569	0.43	0.763	0.236
United Kingdom (females)	0.825	0.374	0.013	0.322	0.674	0.765	0.235
United Kingdom (males)	0.716	0.036	0.039	0.78	0.218	0.799	0.2
USA (females)	0.816	0.456	0.007	0.287	0.712	0.823	0.177
USA (males)	0.643	0.02	0.061	0.765	0.234	0.84	0.159

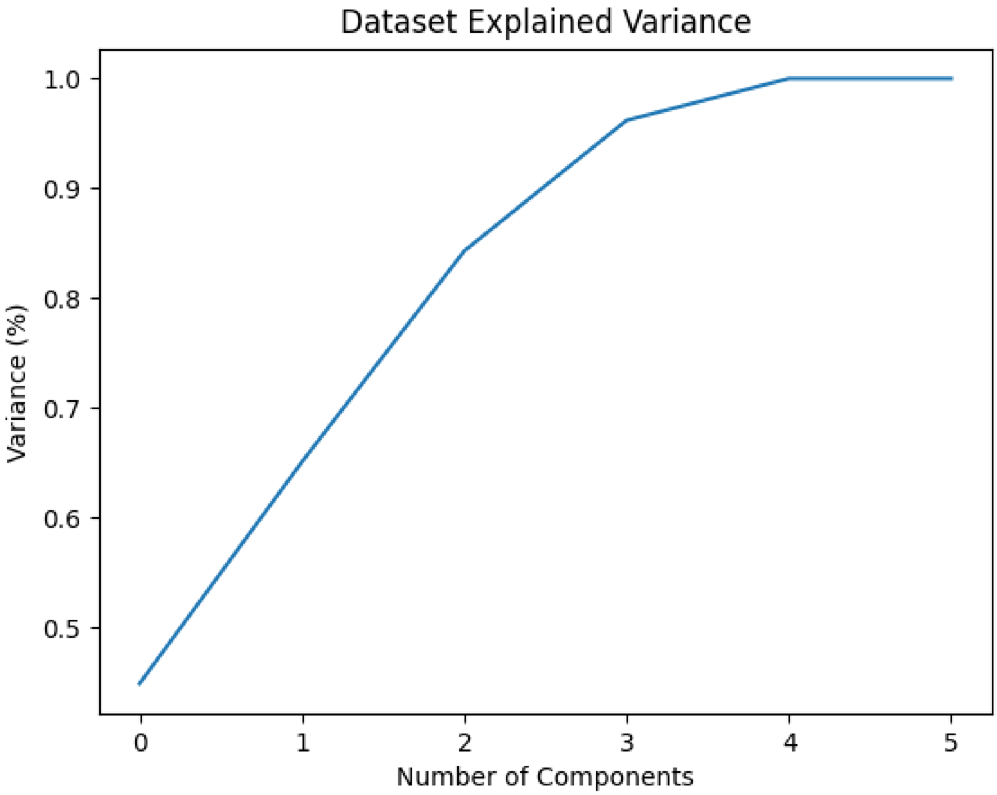
The countries where the main language is spanish (Uruguay + Spain) and english (USA + United Kingdom + Australia) are having very similar variation with the features chosen between males and females with these datasets (remember is the datasets extracted from official statistics provided by the states). Canada, a country french centric has different rules with this features.

The letter a is varying 0.2 from males to females in (USA and Uruguay) and 0.1 from males to females (United Kingdom, Australia and Spain). The last letter a is varying 0.5 from males to females in (Australia, Spain) around 0.4 in (USA, United Kingdom) and 0.2 in Uruguay. The last letter o from females to males is varying 0.2 in (Spain, Australia) and is equal in (Uruguay, USA, United Kingdom). For the last letter consonant all countries is giving the result that is for males, with results from 0.2 to 0.5: Uruguay and USA (0.5), United Kingdom (0.4), Australia and Spain (0.3). So last letter vocal is reverse tha last letter consonant. First letter consonant or first letter vocal is a non significative feature due to so similar results in english and spanish.

Surely, the rules it's a coincidence but we think that is a coincidence between languages due to that there are a good number of names to think different.

4.2.2 Choosing components

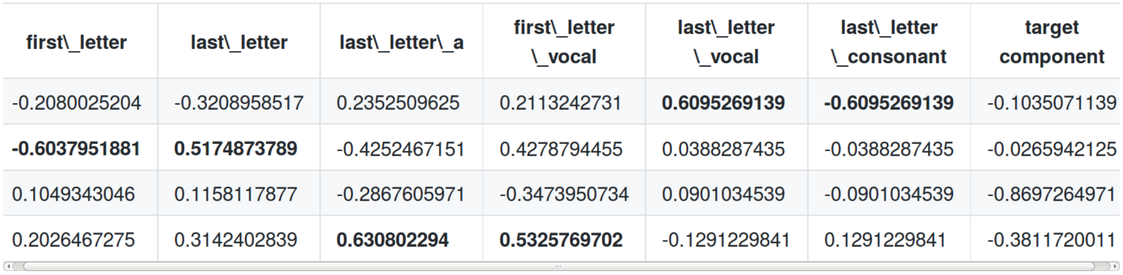
After, to choose features for our machine learning task, we can understand if this features makes sense with Principal Component Analysis. We have written 2 scripts for this task `pca-components.py` and `pca-features.py`. With `pca-components.py` we are giving a csv (files/features_list.csv, files/features_list_no_cat.csv, ...) and the output is an image where we can visualize a curve to determine when this curve stops the growth the number of components.



In the image, we can see that the curve stops the growth in the fourth component.

When you know the components you can execute `pca-features.py` so:

```
$ python3 pca-features.py  --categorical=both --components=4
The json file is created in files/pca.json
The html file is created in files/pca.html
```



first_letter	last_letter	last_letter_a	first_letter _vocal	last_letter _vocal	last_letter _consonant	target component
-0.2080025204	-0.3208958517	0.2352509625	0.2113242731	**0.6095269139**	**-0.6095269139**	-0.1035071139
-0.6037951881	**0.5174873789**	-0.4252467151	0.4278794455	0.0388287435	-0.0388287435	-0.0265942125
0.1049343046	0.1158117877	-0.2867605971	-0.3473950734	0.0901034539	-0.0901034539	-0.8697264971
0.2026467275	0.3142402839	**0.630802294**	**0.5325769702**	-0.1291229841	0.1291229841	-0.3811720011

To simplify and to learn, we can observe this analysis without letters. In this analysis, we can observe 4 components.

The first component is about if the last letter is vocal or consonant. If the last letter is vocal we can find a male and if the last letter is a consonant we can find a male.

The second component is about the first letter. The last letter is determining females and the first letter is determining males.

The third component is not giving relevant information.

The fourth component is giving the last_letter_a and the first_letter_vocal is for females.

5 Use Cases

5.1 Introduction

There are many research studies count males and females in specific communities such as Twitter, StackOverflow, ... We hope that with this manual software

A specific community has some clues to determine male or female, for example, in Twitter you observe the photo, nickname, real name, ...

5.2 Counting males and females in Debian

In the Debian community all member must have a gpg key to collaborate, so we can count males and females from the keyring. With gpg commands you can import a the debian keyring and dump the debian keyring in a csv file.

```
$ rsync -az --progress keyring.debian.org::keyrings/keyrings/ .
```

We have generated a script to count males and females:

```
~/git/damegender/src/damegender$ python3 count-debian-gender.py
Perhaps you need wait some minutes. You can take a tea or coffe now
debian males: 795
debian females: 24
```

In the dump of the debian keyring dataset we have divided name, surname and email in different fields. So, it's easy detect the name, although some names has several emails

We have choosen the United States of America dataset and we are using the method name_frec to decide for male or female in the row. Take a look to the source:

```
import csv
import unicodedata
import unidecode
from pprint import pprint
import re
from app.dame_gender import Gender
from app.dame_utils import DameUtils

du = DameUtils()
g = Gender()

result=""
dm = []

with open('files/debian-maintainers-gpg-2020-04-01.csv') as csvfile:
    reader = csv.reader(csvfile, delimiter=',', quotechar='|')
    aux = ""
    cnt = 0
    for row in reader:
        cnt = cnt +1
        if (aux != row[0]):
            dm.append(row[0])
        aux = row[0]

print("Perhaps you need wait some minutes. You can take a tea or coffe now")

females = 0
males = 0
for rowdm in dm:
    if (int(g.name_frec(str(rowdm.upper()), ’us’)[’females’]) > int(g.name_frec(str(ro
        females = females + 1
    else:
        males = males + 1

print("debian males: %s" % males)
print("debian females: %s" % females)

csvfile.close()
```

The advantage using name_frec is about to understand how you are deciding male or female in the script counting males and females. In this script the decision is simple: a name is male if there are more males than females and female if there are more females than males.

The United States of America dataset is a good choice for Free Software communities, due to that this communities is based on english as main language and United States of America is a leader country in software development. United States of America hosts people from different countries due to migrations towards good companies and universities.

5.3 Counting males and females in Forbes

In the second example, we are using guess without machine learning instead of name_frec. If you are using guess you are trusting on damegender to take the decision, but perhaps you are not agree.

Please take a look about our guess method in the current state:

```
    def guess(self, name, binary=False, *args, **kwargs):
        # guess list method
        dataset = kwargs.get(’dataset’, ’es’)
        # guess method to check names dictionary
        guess = ’’
        name = unidecode.unidecode(name).title()
        name.replace(name, "")
        dicc = self.name_frec(name, dataset)
        m = int(dicc[’males’])
        f = int(dicc[’females’])
        if ((m == 0) and (f == 0)):
            if binary:
                guess = 2
            else:
                guess = "unknown"
        elif (m > f):
            if binary:
                guess = 1
            else:
                guess = "male"
        elif (f > m):
            if binary:
                guess = 0
            else:
                guess = "female"
        else:
            if binary:
                guess = 2
            else:
                guess = "unknown"
        return guess
```

We are using the spanish dataset by default and the rest is the same idea that in the last script: more people using the name.

5.4 Deciding for males and females in images

There are many free software tools for decide gender in images files, we have selected the next tool:

```
$ git clone https://github.com/davidam/damephoto
$ cd damephoto/bin
$ python3 damephoto.py girl1.jpg
```

We can use this tool to decide gender about images from Twitter, Github, ...

5.5 Webscraping and Damegender because we want count scientifics

Sometimes, we can reach the database of names from a website, for example, we can retrieve a list of scientifics from Spain thanks to webometrics and the next script:

```
from lxml import html
import requests

print("Introduce an url from webometrics, for example, https://www.webometrics.info/en

import argparse

parser = argparse.ArgumentParser()
parser.add_argument("url", help="display the gender")
args = parser.parse_args()

page = requests.get(args.url)
tree = html.fromstring(page.content)
```

```
scientifics = tree.xpath('//tr/td/a/strong/text()')

print('Scientifics: %s' % scientifics)
```

If you have retrieved the list of names in a file `files/scientifics.txt`, you could count males and females with the next script called `count-scientifics.py`:

```
import csv
import unicodedata
import unidecode
import re

from pprint import pprint
from app.dame_gender import Gender
from app.dame_utils import DameUtils
from ast import literal_eval
from app.dame_sexmachine import DameSexmachine

du = DameUtils()
g = Gender()
s = DameSexmachine()

with open('files/scientifics.txt') as f:
    mainlist = [list(literal_eval(line)) for line in f]

l = mainlist[0]

ll = []
for i in l:
    ll.append(i.split())


ten = ll[0:10]
hundred = ll[0:100]
thousand = ll[0:1000]

x = 0
y = 0
males = 0
females = 0
for j in hundred:
    if (len(j[0]) == 1):
        x = x + 1
    else:
        sex = g.guess(j[0], binary=False)
        y = y +1
        if (sex == "male"):
            males = males + 1
        elif (sex == "female"):
            females = females + 1

print("Number of scientifics with a single letter as first name: %s" % x)
print("Number of scientifics with the first name normal: %s" % y)
print("Number of females scientifics: %s" % females)
print("Number of males scientifics: %s" % males)

for j in thousand:
    if (len(j[0]) == 1):
        x = x + 1
    else:
        sex = g.guess(j[0], binary=False)
        y = y +1
        if (sex == "male"):
            males = males + 1
        else:
            females = females + 1

print("Number of females scientifics: %s" % females)
print("Number of males scientifics: %s" % males)
```

And the results are:

```
Number of females scientifics: 31425
Number of males scientifics: 47945
```

5.6 Counting males and females in a git repository

We can think a simple version of git2gender.py:

```
from app.dame_sexmachine import DameSexmachine
from app.dame_perceval import DamePerceval
from app.dame_utils import DameUtils
import sys
import argparse
parser = argparse.ArgumentParser()
parser.add_argument("url", help="Uniform Resource Link")
parser.add_argument(’--directory’)
parser.add_argument(’--version’, action=’version’, version=’0.1’)
args = parser.parse_args()
if (len(sys.argv) > 1):
    ds = DameSexmachine()
    du = DameUtils()
    dp = DamePerceval()
    l1 = dp.list_committers(args.url, args.directory)
    l2 = du.delete_duplicated(l1)
    l3 = du.clean_list(l2)
```

```
females = 0
males = 0
unknowns = 0
for g in l3:
    sm = ds.guess(g, binary=True)
    if (sm == 0):
        females = females + 1
    elif (sm == 1):
        males = males + 1
    else:
        unknowns = unknowns + 1

print("The number of males sending commits is %s" % males)
print("The number of females sending commits is %s" % females)
```

Try to execute this script:

```
$ python3 git2gender.py https://github.com/davidam/davidam.git --directory="/tmp/clone
The number of males sending commits is 3
The number of females sending commits is 0
```

This count is not so good because in a git repository the same person can have been called with the same name:

```
['David Arroyo Menéndez <davidam@es.gnu.org>', 'David Arroyo Menendez <davidam@gmail.c
```

If you look up 'David Arroyo' in Google Scholar you can find several researchers, but David Arroyo Menéndez is unique in this context. If appears with accent or not could a trouble about spelling, but could be the same person. On other hand, if you find 'David Arroyo' and 'David Arroyo Menéndez' with the same, the he is the same person, but in some data centers several people with different names can be using the same email account.

def same_email(string1, string2): firstName

def same_identity(string1, string2): same_identity = false string1 = remove_accents(string1) string2 = remove_accents(string2) if (same_email(string1, string2) and ((contains(string1, string2)) or (contains(string2, string1)))): same_identity = true else: same_identity = false return same_identity

Appendix A GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
https://fsf.org/

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `https://www.gnu.org/copyleft/`.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with…Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

C

Commands 3
Commands about Statistics 3
Configuring Api Keys 2

E

Executing tests 3

I

Installation 2

P

Perceval 3
Python Virtual Environment 2

R

Regenerating files in post installation 3