MIDS W205

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab #** | 10 | **Lab Title** | OpenRefine -- Introduction |
| **Related Module(s)** | 10 | **Goal** | Get you started on OpenRefine and Edit Distance |
| **Last Updated** | 9/27/15 | **Expected duration** | 60 minutes |

## Introduction

OpenRefine is an open source tool for working with bad data. In this Lab we will give you a quick tour of how you can use it to clean data. If you want a more comprehensive tutorial you can follow any of the tutorials listed in the resources section.

We will be using two data sets one from with earthquake data and one with customer complaint data. The first data set is the eq2015 data set which data about earthquakes of magnitude 3 or more during the first 6 months of 2015. You can download the data set here . You can fine a data attribute glossary here The second data set contains customer complaints, you can download that data set here . Please answer the following questions by using OpenRefine.

OpenRefine is to a large extent menu driven. But it also allows you to use a language for doing certain types of transformations.

The basic idea in OpenRefine is that you think of exploring your data in terms of patterns, called facets. OpenRefine also has functions for doing transformations of data. These transformations can be expressed in the language GREL although there are a few other options as well. As an example, you can decide to create a new column based on an existing column but with a transformation applied to the data. GREL is a

# Instructions, resources and prerequisites

This Lab has 3 parts. The first two are involves using OpenRefine to clean up some data files. The third one involves calculating the Levenshtein distance between two strings.

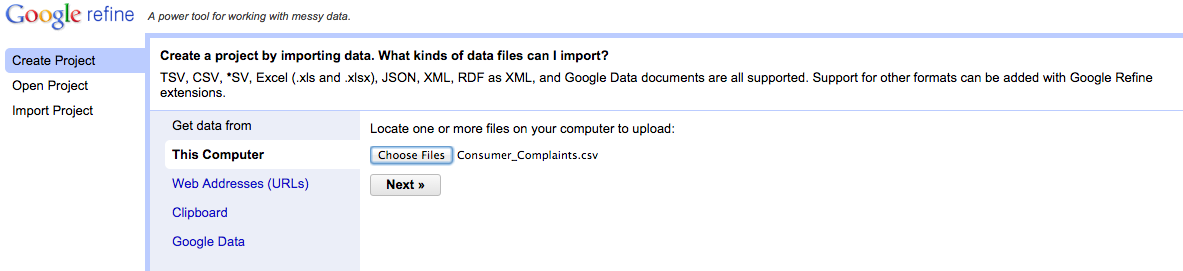
|  |  |
| --- | --- |
| **Resource** | **What** |
| http://openrefine.org/ | This is where you download OpenRefine. |
| http://arcadiafalcone.net/GoogleRefineCheatSheets.pdf | A short description of OpenRefine commands. |
| http://enipedia.tudelft.nl/wiki/OpenRefine\_Tutorial | Another tutorial on OpenRefine . |
| http://davidhuynh.net/spaces/nicar2011/tutorial.pdf | Another tutorial on OpenRefine . |
| http://schoolofdata.org/handbook/recipes/cleaningdatawithrefine/ | Programming guide for the Spark Context object. Here you can find actions available on the Spark Contexts. |
| https://github.com/OpenRefine/OpenRefine/wiki/General-Refine-Expression-Language | GREL is the language used in OpenRefine for data refinements. This is a reference guide for the GREL language. |
| http://earthquake.usgs.gov/eart | Explanation of the Earthquake data. |
| https://pypi.python.org/pypi/python-Levenshtein/0.12.0 | A Levenshtein module you can use to check your results in a Python shell. |

Cleaning Data with OpenRefine.

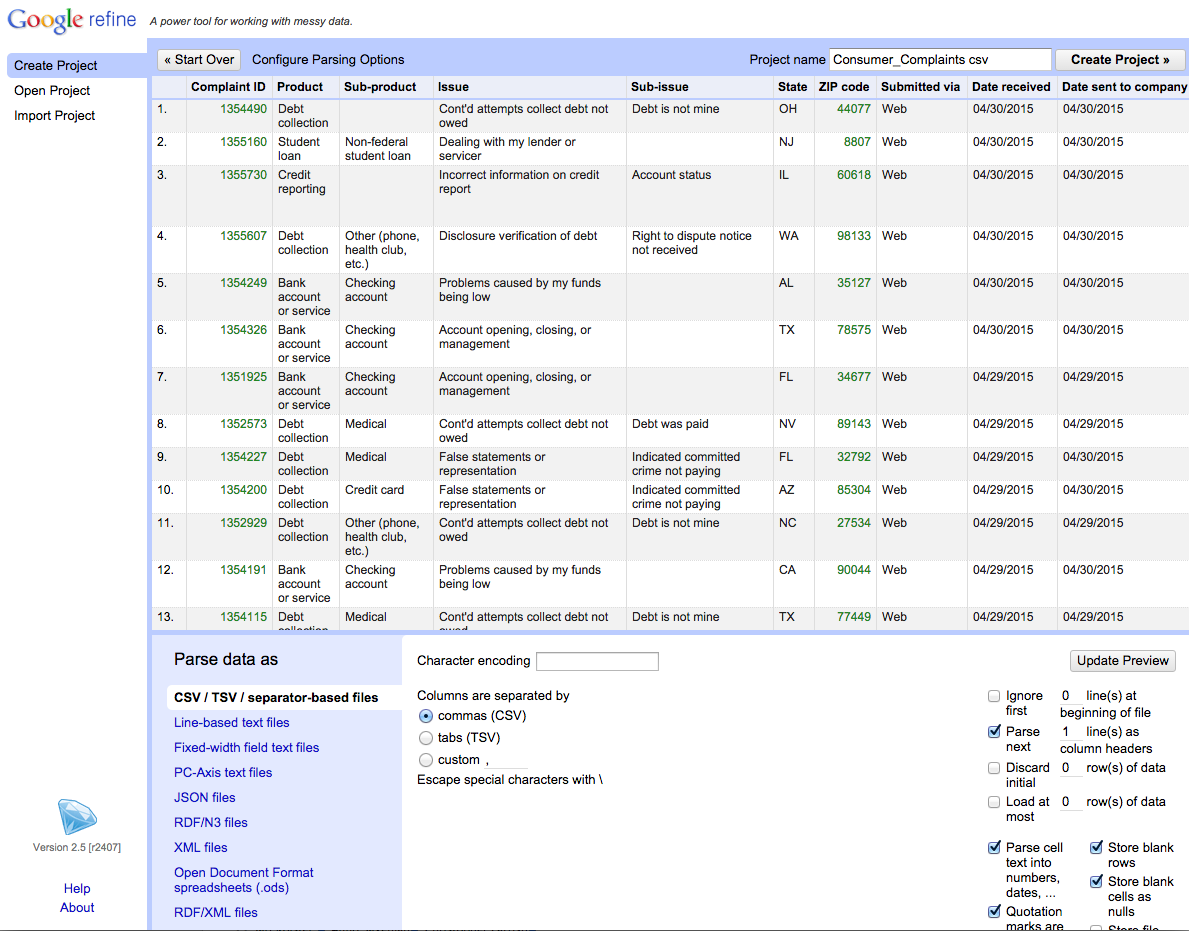
# Step-1. Wrangling the Customer Complaints Data

## Uploading data

After you started OpenRefine you can pick a data set. For this first step choose the Customer Complaints Data set.

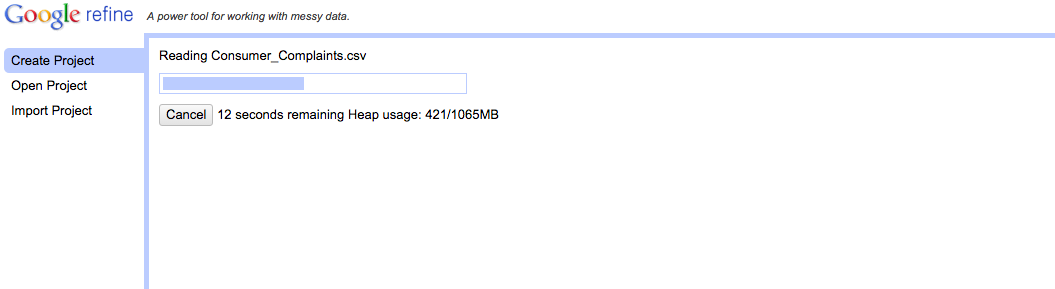


Once the data is read you can inspect it. In this case it looks ok. But lets say that it would have been tab separated, then OpenRefine would not have read it correctly. You have the opportunity to look at the data here and confirm it is ok. Ion this case we think it looks good and we click the “Create Project” button.

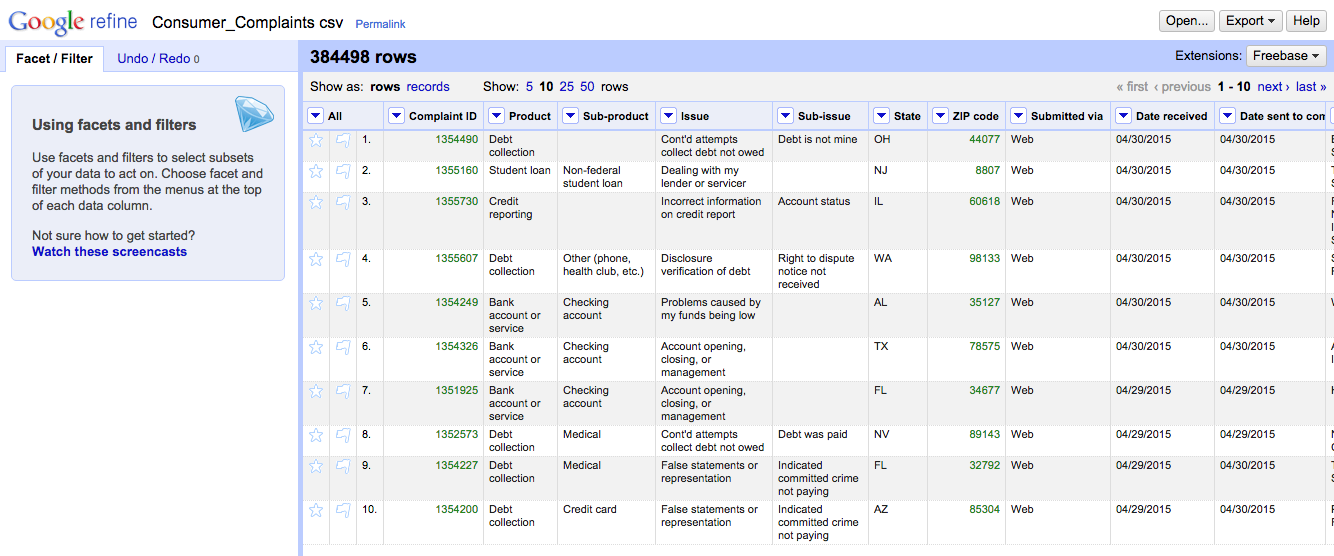


## Creating a project

Creating the project can take a little time.

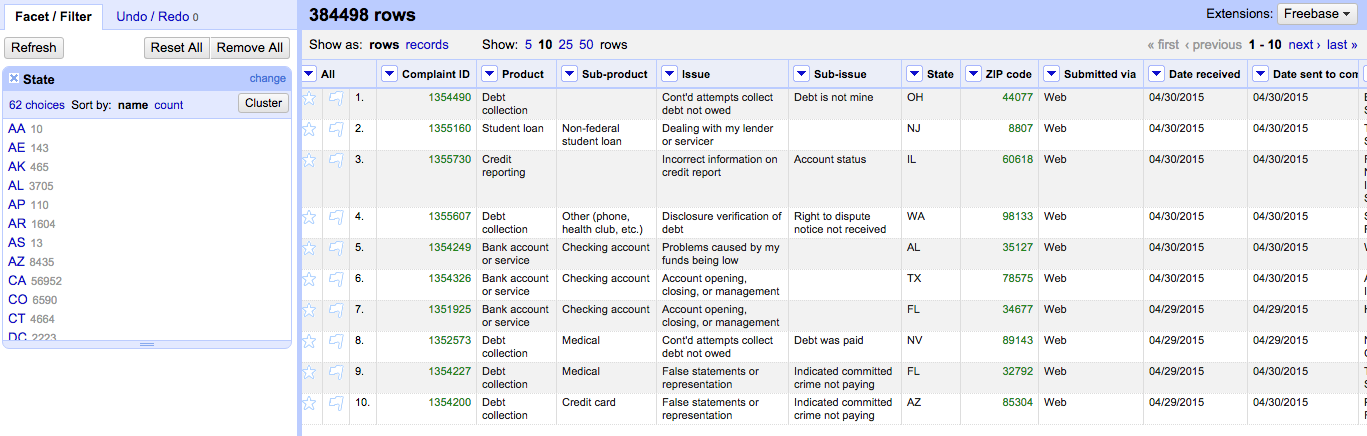


Once the project is created you can see that it has 384498 rows.



Check states with text facet

If you select text facet for state you will see a summary in the left column pane. It indicates you we have 62 different state value (?). Try to figure out why.



● A1: How many rows are missing value in the state column? Explain how you came up

with the number?

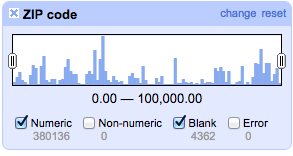
Checking zip codes

Try the text facet on zip codes, what happens? You can see that there Is 24748 different zip codes in this data set. Is that reasonable? Eye ball the data, does all zip codes look valid?

Now try the numeric facet. With the numeric facet the zip code attribute is treated is a numeric value. What would you say the scalar type is for zip codes, can be treated as a numeric attribute? Histogram below shows the distribution when the attribute is treated as numeric.

By un checking numeric you can get a list of row that are missing.

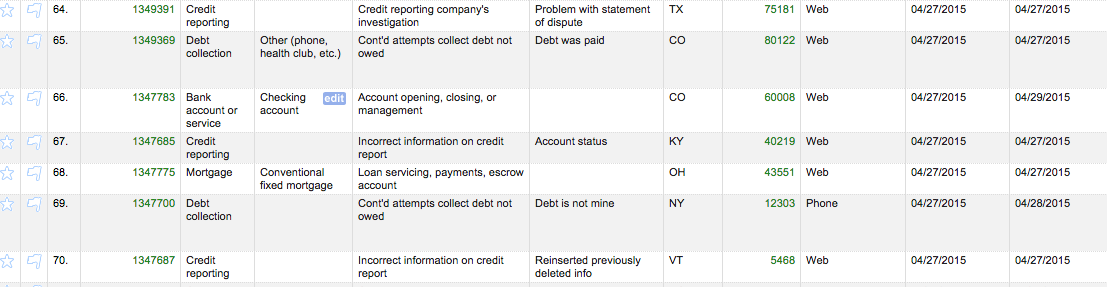
● A2: How many rows with missing zip codes do you have?



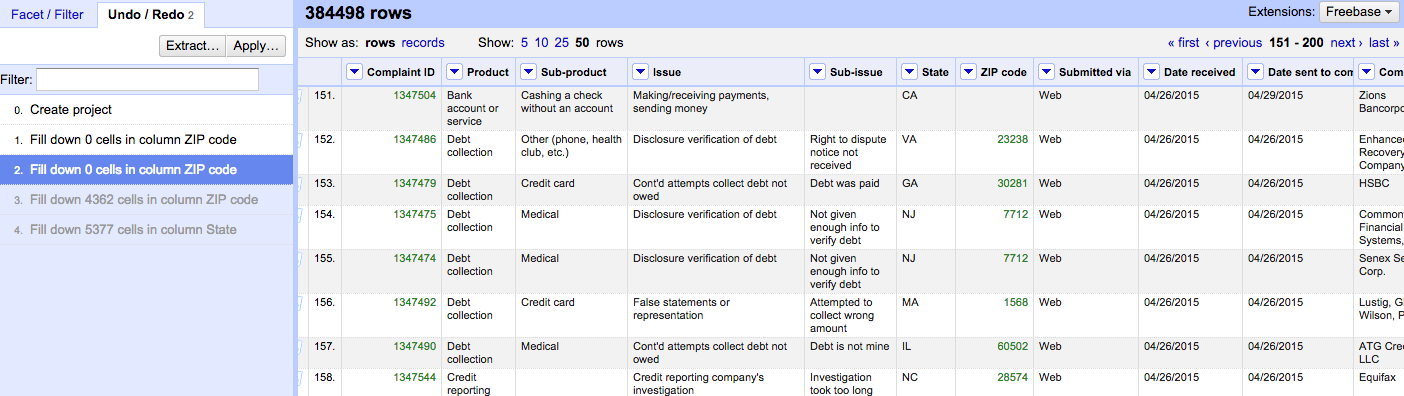
One way of filling in missing values if to take the previous value and use that. In OpenRefine it is called fill-down. Find a row that is blank. Apply fill down to the fill down by:

Edit Cell->Fill Down

What happened to the empty cell.



If you need to undo the operation, switch to the Undo/Redo tab. Select the a previous state for the data. In this example I went back to state 2. As you can see in this screen shot row 151 has a missing zip code, so presumely the fill downs for Zip code and State has been un done.



● A3: Transforming zip code column. Lets create a new column called “ZipCode5” with all zip

codes that contains 5 digits preserved. All other rows should have the zip code 99999. (Technically speaking the 4-digit zip codes may be valid zip codes, we do this more to illustrate transformations).

Transformations are expressed in some language. OpenRefine supports a few alternatives, we will be using GREL. You can find a link to a language reference in the resources section. For this simple transnformation we will be using an if statement.



For the Zip code column select:

Edit Column -> Add column based on this column.

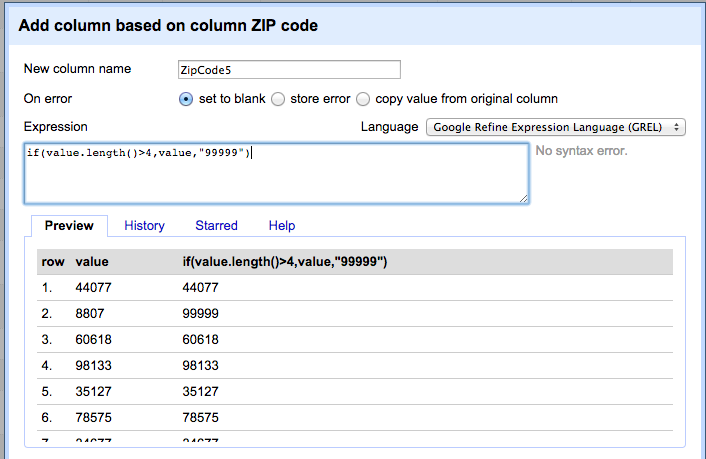
You will get the dialogue below. Insert the name of the new column and the expression:

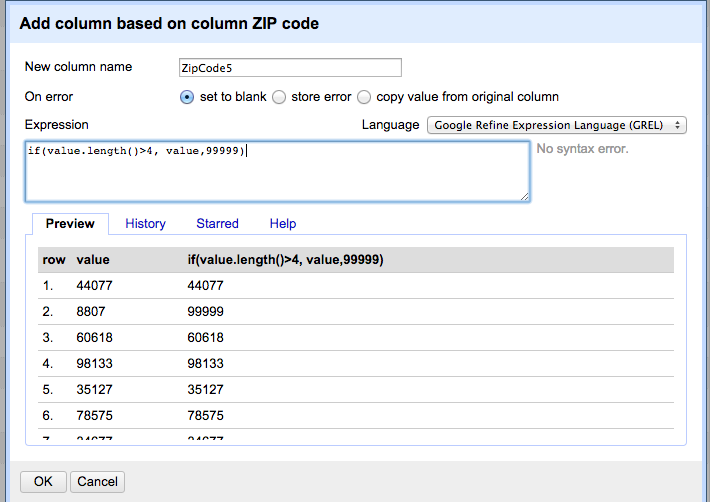
If(value.length() > 4, value, “99999”)

This expressen states that if the length of value is more than 4 insert value, otherwise insert the string “99999”.

Look at the result, this it do what you wanted? That seems to be wrong with that? What happens if you instead insert a numeric value using the expression:

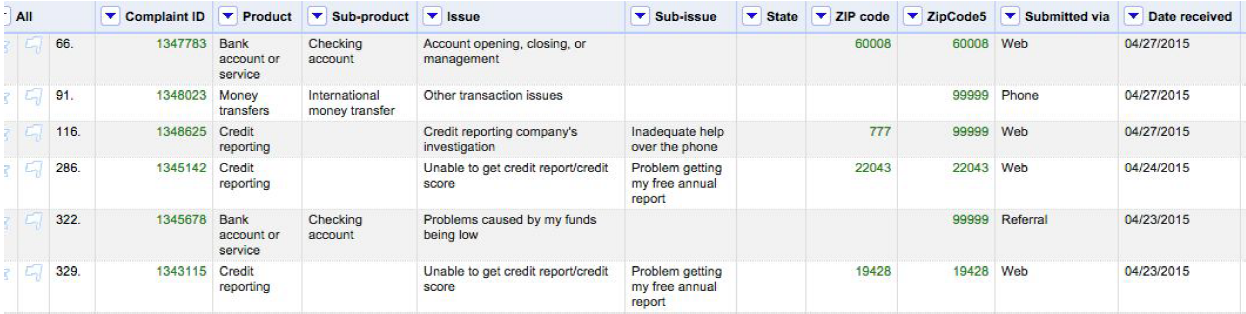
If(value.length() > 4, value, 99999)





You should have the same type for all cells in the created column.

Example of result:



● A4: If you consider all zip codes less than 99999 valid zip codes. How many valid and

invalid zip codes do you have respectively.

# Step-2. Cleaning up eq2015 Data.

Upload the data

● A5: For column “nst” fill in missing values.

● A6: Clean up the place column so that it has state or country name depending on what is

in the text.

● A7: From the column “updated” extract the Date without time into a new column called

“eventdate”

● A8: Run cluster en edit on “location” column. Run nearest neighbor and levenshtein

distance. Answer the following questions:

○ Does it make sense to merge detected values?

○ Why or why not?

● A9: Try to do nearest neighbor clustering on “place’ column.

○ What happens?

○ Explain why it is happening.

# Step-3 Levenshtein Distance

In this lab we will go over a simple example of Levenshtein distance calculation. We will then ask you to calculate the distance for two strings gumbarrel" and "gunbarell". We will point you to a simple implementation of the Levenshtein distance that you can use to check your result.

## Installing Levenshtein python module

The following steps will just clone and build a Python Levenshtein module in a directory. It does not fully install the module. But you can use it to run a distance function from your shell to check your results.

$ git clone <https://github.com/ztane/python-Levenshtein/>

$ cd python-Levenshtein/

$ python setup.py build

$ cd Levenshtein/

$ python

>>> from Levenshtein import \*

>>> distance("hej","hei")

1

>>> distance("monthgomery st","montgomery street")

5

## Example: Levenshtein Calculation

Lets step through the calculation of distance between the words LOYOLA and LAJOLLA. We will denote a cell with the d[i,j], where i is the row and j is the column. The first column and row indicates the indices we will be using.

As a reminder the algorithms is as follows:

Denote the column by c and row by r. We have n rows and m columns. d[i,j] denotes the value on row i and columns j.

d[i,j] is to be set to the minimum of: d[i-1,j]+1 or d[i,j-1]+1 or d[i-1, j-1]+cost[I,j]

Distance is found in the resulting value d[n,m]

We first set up the matrix. The blue row and column just contains the i and j values. We then insert values 0-m in first row i==1 and 0-n in the column j==1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  | L | O | Y | O | L | A |
| 1 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | L | 1 |  |  |  |  |  |  |
| 3 | A | 2 |  |  |  |  |  |  |
| 4 | J | 3 |  |  |  |  |  |  |
| 5 | O | 4 |  |  |  |  |  |  |
| 6 | L | 5 |  |  |  |  |  |  |
| 7 | L | 6 |  |  |  |  |  |  |
| 8 | A | 7 |  |  |  |  |  |  |

Lets calculate the d[i,2]. Meaning the value for each row in the column 2.

d[2,2], cost is 0, minimum is d[1,1]+0=>0

d[3,2], cost is 1, minimum is d[2,2]+1=>1

d[4,2], cost is 1, minimum is d[3,2]+1=>2

d[5,2], cost is 1, minimum is d[4,2]+1=>3

d[6,2], cost is 0, minimum is d[5,1]+0=>4, or d[5,2]+1

d[7,2], cost is 0, minimum is d[6,2]+0=>5, or d[6,2]+1

d[8,2], cost is 1, minimum is d[7,2]+1=>6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  | L | O | Y | O | L | A |
| 1 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | L | 1 | 0 |  |  |  |  |  |
| 3 | A | 2 | 1 |  |  |  |  |  |
| 4 | J | 3 | 2 |  |  |  |  |  |
| 5 | O | 4 | 3 |  |  |  |  |  |
| 6 | L | 5 | 4 |  |  |  |  |  |
| 7 | L | 6 | 5 |  |  |  |  |  |
| 8 | A | 7 | 6 |  |  |  |  |  |

Lets calculate the d[i,3]. Meaning the value for each row in the column 3.

d[2,3], cost is 1, minimum is d[2,2]+1=>1

d[3,3], cost is 1, minimum is d[2,2]+1=>1

d[4,3], cost is 1, minimum is d[3,2]+1=>2, or d[3,3]+1

d[5,3], cost is 0, minimum is d[4,2]+0=>2

d[6,3], cost is 1, minimum is d[5,3]+1=>3

d[7,3], cost is 1, minimum is d[6,2]+1=>4, or d[6,3]+1

d[8,3], cost is 1, minimum is d[7,2]+1=>4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  | L | O | Y | O | L | A |
| 1 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | L | 1 | 0 | 1 |  |  |  |  |
| 3 | A | 2 | 1 | 1 |  |  |  |  |
| 4 | J | 3 | 2 | 2 |  |  |  |  |
| 5 | O | 4 | 3 | 2 |  |  |  |  |
| 6 | L | 5 | 3 | 3 |  |  |  |  |
| 7 | L | 6 | 3 | 4 |  |  |  |  |
| 8 | A | 7 | 4 | 4 |  |  |  |  |

Now if the you do the same thing for the rest of the columns we will get the following matrix. You see the calculated edit distance in the cell d[8,7].

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  | L | O | Y | O | L | A |
| 1 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | L | 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 3 | A | 2 | 1 | 1 | 2 | 3 | 4 | 4 |
| 4 | J | 3 | 2 | 2 | 2 | 3 | 4 | 5 |
| 5 | O | 4 | 3 | 2 | 3 | 2 | 3 | 4 |
| 6 | L | 5 | 3 | 3 | 3 | 3 | 2 | 3 |
| 7 | L | 6 | 3 | 4 | 4 | 4 | 3 | 3 |
| 8 | A | 7 | 4 | 4 | 5 | 5 | 4 | **3** |

If you use the Levenshtein function to check the result you will see

>>> distance("loyola","lajolla")

3

So we are assuming we got the calculation right.

## Calculation: gumbarrel v.s gunbarell

Now calculate the edit distance of the words: “gumbarrel” and “gunbarell". Use the python levenshtein function to check you result.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  | G | U | M | B | A | R | R | E | L |
| 1 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | G | 1 |  |  |  |  |  |  |  |  |  |
| 3 | U | 2 |  |  |  |  |  |  |  |  |  |
| 4 | N | 3 |  |  |  |  |  |  |  |  |  |
| 5 | B | 4 |  |  |  |  |  |  |  |  |  |
| 6 | A | 5 |  |  |  |  |  |  |  |  |  |
| 7 | R | 6 |  |  |  |  |  |  |  |  |  |
| 8 | E | 7 |  |  |  |  |  |  |  |  |  |
| 9 | L | 8 |  |  |  |  |  |  |  |  |  |
| 10 | L | 9 |  |  |  |  |  |  |  |  |  |

SUBMISSION 3: submit a representation of the resulting matrix from the leveshtein edit distance calculation. The resulting value should be correct.