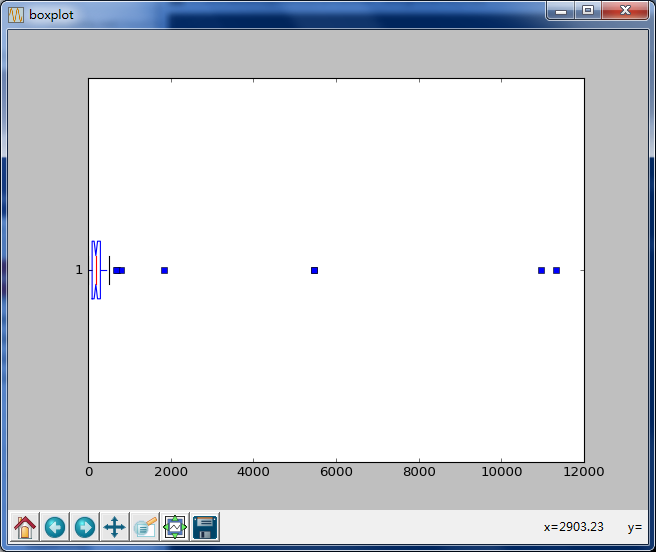
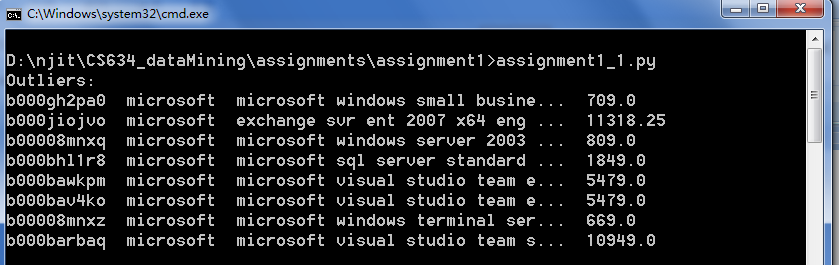
Boxplot:



Outliers:



Source Code:

import matplotlib.pyplot as plt

import pandas as pd

import io

import sys

#collect data

reader = pd.read\_csv("Amazon.csv", encoding = "ISO-8859-1", iterator = True, chunksize = 100)

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='ISO-8859-1')

index = 0;

data = [[],[],[],[]]

for chunk in reader:

id = chunk.get("id")

title = chunk.get("title")

company = chunk.get("manufacturer")

price = chunk.get("price")

for i in range(0, len(chunk), 1):

if (company.get(index) == "microsoft"):

data[0].append(id.get(index))

data[1].append(company.get(index))

data[2].append(title.get(index))

data[3].append(price.get(index))

index += 1

#print(index)

#show a boxplot and print outliers

plt.figure('boxplot')

res = plt.boxplot(data[3], notch = 1, sym = 'rs', vert = 0, patch\_artist = 0)

outliers = {}

for r in res.get('fliers'):

for d in r.get\_data()[0]:

outliers[d] = d

#retrieve complete data

print("Outliers:")

for p in outliers:

for i in range(0, len(data[3]), 1):

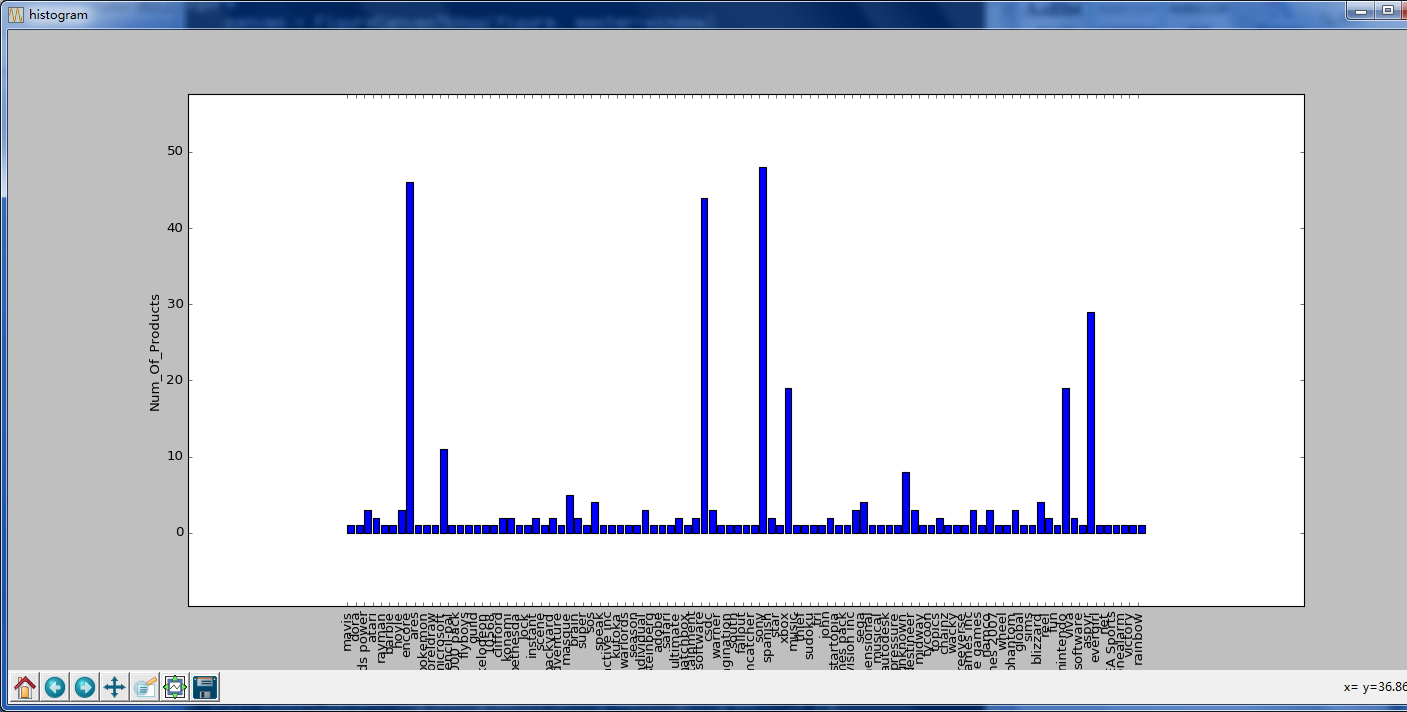
if p == data[3][i]:

print(str(data[0][i]) + " " + str(data[1][i]) + " " + str(data[2][i])[:30] + "... " + str(data[3][i]))

plt.show()



Histogram:



Source Code:

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import io

import sys

import re

#collect data

reader = pd.read\_csv("GoogleProducts.csv", encoding = "ISO-8859-1", iterator = True, chunksize = 100)

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='ISO-8859-1')

index = 0;

data = [[],[],[],[]]

# popular game companies' prefixes and full names

# popular game products and their companies

companies = {

"blizzard" : "blizzard",

"aspyr" : "aspyr",

"egames" : "egames",

"got" : "got game entertainment",

"nintendo" : "nintendo",

"sony" : "sony",

"kids" : "kids power",

"eagle" : "eagle games",

"sega" : "sega",

"flyboys" : "flyboys",

"rock" : "rock star",

"namco" : "namco",

"konami" : "konami",

"evergirl" : "evergirl",

"vivendi-universal": "vivendi-universal",

"xbox": "xbox",

"freeverse" : "freeverse software",

"enteractive" : "enteractive inc",

"warcraft" : "blizzard",

"starcraft" : "blizzard",

"wingnuts 2: raina's revenge" : "Freeverse",

"midway arcade treasures" : "midway",

"activision" : "activision inc",

"vivendi-universal" : "vivendi-universal games inc",

"11273" : "encore",

"fifa" : "EA Sports"

}

# other keywords

keywords = [

"warcraft",

"starcraft",

"nintendo",

"xbox",

"wingnuts 2: raina's revenge",

"midway arcade treasures",

"game",

"gaming"

]

for chunk in reader:

id = chunk.get("id")

title = chunk.get("name")

description = chunk.get("description")

price = chunk.get("price")

manufacturer = chunk.get("manufacturer")

for i in range(0, len(chunk), 1):

#parse company roughly

comp = None

res = None

found = False

res = re.match("([\w]+(?=:))",str(title.get(index))) #match company name ends with :

if (res == None) : res = re.match("([\w'-]+(?=\s))",str(title.get(index))) #match company name ends with space

if (res != None) : comp = res.group(0)

else: comp = str(title.get(index))

if (companies.get(comp) != None):

comp = companies.get(comp)

found = True

else:

for keyword in keywords:

if ((keyword in str(title.get(index))) or (keyword in str(description.get(index)))):

found = True

if (companies.get(keyword) != None): comp = companies.get(keyword)

else :

if (('microsoft' in str(title.get(index))) or ('microsoft' in str(description.get(index)))): comp = 'microsoft'

break

# get a game related item

if (found == True):

data[0].append(re.findall("(?<=/)\w+", id.get(index))[-1])

if ('\'' in comp or len(comp) <= 2): comp = 'unknown'

data[1].append(comp)

data[2].append(title.get(index))

data[3].append(price.get(index))

index += 1

#print(index)

count = {}

for i in range(0, len(data[1]), 1):

if (count.get(data[1][i]) == None): count[data[1][i]] = 1

else : count[data[1][i]] = count.get(data[1][i]) + 1

# print(str(data[0][i]) + " " + str(data[1][i]) + " " + str(data[2][i])[:20] + "... " + str(data[3][i]))

plt.figure('histogram', figsize = (24,8))

X = np.arange(len(count))

plt.bar(X, count.values())

plt.xticks(X, count.keys(), rotation='vertical')

plt.xlabel('Manufacturer')

plt.ylabel('Num\_Of\_Products')

plt.margins(0.2)

plt.show()

Algorithm:

To find identical items, we need pick each element from one dataset, and then compute the similarity of the element to each element of dataset 2 and record the max similarity and the index of element of dataset 2 that gives the maximum similarity.

* We define similarity = (s\_title \* 2 + s\_comp + s\_price) / 4

Checking description will be too slow, therefore just skip it this time

The similarity of title will be more reliable than that of company and price; we give more weight on title

* If similarity > 0.6, we think they are the same item
* For computing the similarity of text attributes title and company, we use difflib.SequenceMatcher which is based on finding the longest contiguous matching subsequence that contains no “junk” elements and convert the result into (0,1) range
* For numeric attribute price, we define s\_price = 1 - abs(price1 - price2) / max(price1, price2), which will also give a value between(0,1)
* Since for some of items of Amazon file, the prices of items are 0(missing), we just simply ignore price attribute, and therefore our similarity becomes similarity = (s\_title \* 2+ s\_comp) / 3

pseudo-code:

for each i in dataset1{

double max = 0; //max similarity for element i so far

int maxJ = -1;// max index j that gives that maximum similarity for element i

for each j in dataset2{

s\_title = difflib.SequenceMatcher(d1[title][i],d2[title][ j])

s\_comp= difflib.SequenceMatcher(d1[comp][i],d2[comp][ j])

if (d2[price][ j] != 0 ){

s\_price = 1 - abs(data[price][i] - data2[price][j]) / max(data[price][i] , data2[price][j])

s = (s\_title \* 2 + s\_comp + s\_price) / 4

}

Else s = (s\_title \* 2 + s\_comp) / 3

If (s > max) {

max = s

maxj = j

}

}

If (max > 0.6) output to file(i, maxj)

}

Source Code:

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import io

import sys

import re

import difflib

import datetime

import csv

#collect data

reader = pd.read\_csv("GoogleProducts.csv", encoding = "ISO-8859-1", iterator = True, chunksize = 100)

sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='ISO-8859-1')

index = 0;

data = [[],[],[],[],[]]

for chunk in reader:

id = chunk.get("id")

title = chunk.get("name")

description = chunk.get("description")

price = chunk.get("price")

manufacturer = chunk.get("manufacturer")

for i in range(0, len(chunk), 1):

#parse company roughly

comp = None

res = None

res = re.match("([\w]+(?=:))",str(title.get(index))) #match company name ends with :

if (res == None) : res = re.match("([\w'-]+(?=\s))",str(title.get(index))) #match company name ends with space

if (res != None) : comp = res.group(0)

else: comp = str(title.get(index))

data[0].append(id.get(index))

if (str(manufacturer.get(index)) != "nan") : comp = str(manufacturer.get(index)) # adjust comp by original data

data[1].append(str(comp).lower())

data[2].append(str(title.get(index)).lower())

if ("gbp" in str(price.get(index))):

v = 1.22 \* float(str(price.get(index))[:len(price.get(index)) - 4])

data[3].append(v)#convert into dollar

#print(v)

else : data[3].append(float(price.get(index)))

data[4].append(str(description.get(index)).lower())

index += 1

reader = pd.read\_csv("Amazon.csv", encoding = "ISO-8859-1", iterator = True, chunksize = 100)

index = 0;

data2 = [[],[],[],[],[]]

for chunk in reader:

id = chunk.get("id")

title = chunk.get("title")

company = chunk.get("manufacturer")

price = chunk.get("price")

description = chunk.get("description")

for i in range(0, len(chunk), 1):

data2[0].append(id.get(index))

comp = ""

res = re.match("([\w]+(?=\s))",str(company.get(index)))

if (res == None):

comp = str(company.get(index))

else:

comp = res.group(0)

data2[1].append(str(comp).lower())

data2[2].append(str(title.get(index)).lower())

data2[3].append(price.get(index))

data2[4].append(str(description.get(index)).lower())

index += 1

f = open("result.csv", "w", encoding='ISO-8859-1')

writer = csv.writer(f, lineterminator='\n')

writer.writerow(("G\_ID", "G\_TITLE", "A\_ID", "A\_TITLE"))

t1 = datetime.datetime.now().time()

print(t1)

for i in range(0, len(data[0]), 1):

max = 0

maxJ = -1

for j in range(0, len(data2[0]), 1):

#s\_des = difflib.SequenceMatcher(None, data[4][i], data2[4][j]).ratio() too slow....

s\_title = difflib.SequenceMatcher(None, data[2][i], data2[2][j]).ratio()

s\_comp = difflib.SequenceMatcher(None, data[1][i], data2[1][j]).ratio()

s\_price = 0

if (data2[3][j] != 0):

m = data[3][i]

if (data[3][i] < data2[3][j]): m = data2[3][j]

s\_price = 1 - abs(data[3][i] - data2[3][j]) / m

s = (s\_title \* 2 + s\_comp + s\_price) / 4

else: s = (s\_title \* 2 + s\_comp) / 3 # eliminate missing data term

if (s > max):

max = s

maxJ = j

if (max > 0.6):

writer.writerow((data[0][i],data[2][i],data2[0][maxJ],data2[2][maxJ]))

t2 = datetime.datetime.now().time()

f.close()

print(t2)

Matching result:

See result.csv