In [23]:
```
from future.builtins import next
import os
import csv
import re
import logging
import optparse

import dedupe
from unidecode import unidecode

import pandas as pd
```

In [24]:
```
pd.options.display.float_format = '{:20,.2f}'.format
pd.set_option('display.max_rows', 5000)
pd.set_option('display.max_columns', 5000)
pd.set_option('display.width', 1000)
pd.set_option('display.max_colwidth', -1)
```

In [25]:
```
amazon_walmart_all_path = (r'/home/ubuntu/jupyter/ServerX/1_Standard Da
                     r'/Processed Data/product_samples/amazon_walmart_al
```

Prepare df and dict corpus

In [65]:
```
fields_of_interest = [
    'Id',
    'name',
    'producer',
    'description',
    'price',
    'category',
    'source'
]
```

In [66]:
```
amazon_walmart_all_df = pd.read_csv(amazon_walmart_all_path, sep=',', q
```

In [ ]:
```
amazon_walmart_all_df.dtypes
```

In [ ]:
```
x = amazon_walmart_all_df[amazon_walmart_all_df['category'].isnull()]
x.head(1)
```

In [ ]:
```
z = amazon_walmart_all_df[amazon_walmart_all_df['producer'].isnull()]
z.head(1)
```

In [ ]:
```
y = amazon_walmart_all_df[amazon_walmart_all_df['price'].isnull()]
y.head(1)
```

In [ ]:
```
h = amazon_walmart_all_df[amazon_walmart_all_df['description'].isnull()
h.head()
```

In [ ]:
```
amazon_walmart_all_df[amazon_walmart_all_df['name'].isnull()]
```

In [67]:
```
description_corpus = amazon_walmart_all_df['description'].to_list()
description_corpus = [x for x in description_corpus if str(x) != 'nan']
```

In [68]:
```
description_corpus[1]
```

Out[68]:
```
'EPSON ELPLP12 1500HRS 200V REPL LAMP FOR LAMP POWERLITE FOR 7700P 560
0P 7600 Features Lamp Life 1500 Hour Manufacturer Epson Corporation Co
mpatible Devices LCD Manufacturer Part Number ELPLP12 Manufacturer Web
site Address www.epson.com Product Name Replacement Lamp Package Type
Retail Product Type 200W UHE Projector Lamp Tech Specs Manufacturer Ep
son Corporation Manufacturer Part Number ELPLP12 Shipping Dimensions
5.25  Depth Manufacturer Website Address www.epson.com Lamp Life 1500
Hour Compatibility Epson Powerlite 7700P Projector Epson Powerlite 760
0P Projector Epson Powerlite 5600P Projector Compatible Devices LCD Pr
oduct Name Replacement Lamp Shipping Weight 1 lb Package Type Retail P
roduct Type 200W UHE Projector Lamp'
```

In [69]:
```
category_corpus = amazon_walmart_all_df.drop_duplicates().to_dict('reco
```

In [70]:
```
categories = list(amazon_walmart_all_df['category'].unique())
categories = [x for x in categories if str(x) != 'nan']
```

In [71]:
```
producer_corpus = amazon_walmart_all_df.drop_duplicates().to_dict('reco
```

In [72]:
```
producers = list(amazon_walmart_all_df['producer'].unique())
producers = [x for x in producers if str(x) != 'nan']
```

In [77]:
```
producers.sort()
producers
```

Out[77]:
```
['-NA-',
 '1d4',
 '24/7 Cases',
 '3 in 1 Charger',
 '3D Connexion',
 '3DRose',
 '3DTV Corp',
 '3Dconnexion',
 '3M',
 '3M#',
 '3gjuice',
 '4inkjets',
 '501001717398',
 'A Days Tech',
 'A Young Life',
 'A-DATA',
 'A4TECH',
 'AAS',
 'AAXA',
 'ABC Products'
```

In [78]:
```
input_file = amazon_walmart_all_path
output_file = 'amazon_walmart_output3.csv'
settings_file = 'amazon_walmart_learned_settings3'
training_file = 'amazon_walmart_training3.json'
```

In [14]:
```
float('1.25')
```

Out[14]:
```
1.25
```

In [79]:
```
def preProcess(key, column):

    try : # python 2/3 string differences
        column = column.decode('utf8')
    except AttributeError:
        pass
    column = unidecode(column)
    column = re.sub('  +', ' ', column)
    column = re.sub('\n', ' ', column)
    column = column.strip().strip('"').strip("'").lower().strip()
    column = column.lower()
    if not column:
        return None

    if key == 'price':
        column = float(column)
    return column

def readData(filename):

    data_d = {}
    with open(filename) as f:
        reader = csv.DictReader(f)
        for row in reader:
            clean_row = [(k, preProcess(k, v)) for (k, v) in row.items(
            row_id = int(row['Id'])
            data_d[row_id] = dict(clean_row)

    return data_d
```

In [80]:
```
print('importing data ...')
data_d = readData(input_file)
```

```
importing data ...
```

In [81]:
```
fields = [
    {'field' : 'name', 'type': 'Name'},
    {'field' : 'name', 'type': 'String'},
    {'field' : 'description',
     'type': 'Text',
     'corpus': description_corpus,
     'has_missing': True
    },
    {'field' : 'category',
     'type': 'FuzzyCategorical',
     'categories': categories,
     'corpus': category_corpus,
     'has missing' : True
    },
    {'field' : 'producer',
     'type': 'FuzzyCategorical',
     'categories': producers,
     'corpus': producer_corpus,
     'has_missing': True
    },
    {'field' : 'price',
     'type': 'Price',
     'has_missing': True
    },
]
```

In [82]:
```
deduper = dedupe.Dedupe(fields)
```

In [ ]:
```
# took about 20 min with blocked proportion 0.8
deduper.prepare_training(data_d)
```

```
INFO:dedupe.canopy_index:Removing stop word with
INFO:dedupe.canopy_index:Removing stop word 7
INFO:dedupe.canopy_index:Removing stop word is
INFO:dedupe.canopy_index:Removing stop word than
INFO:dedupe.canopy_index:Removing stop word get
INFO:dedupe.canopy_index:Removing stop word quickly
INFO:dedupe.canopy_index:Removing stop word that
INFO:dedupe.canopy_index:Removing stop word your
INFO:dedupe.canopy_index:Removing stop word into
INFO:dedupe.canopy_index:Removing stop word and
INFO:dedupe.canopy_index:Removing stop word share
INFO:dedupe.canopy_index:Removing stop word works
INFO:dedupe.canopy_index:Removing stop word you
INFO:dedupe.canopy_index:Removing stop word protect
INFO:dedupe.canopy_index:Removing stop word of
INFO:dedupe.canopy_index:Removing stop word access
INFO:dedupe.canopy_index:Removing stop word are
INFO:dedupe.canopy_index:Removing stop word a
INFO:dedupe.canopy_index:Removing stop word easy
INFO:dedupe.canopy_index:Removing stop word to
```

In [39]:
```
dedupe.consoleLabel(deduper)
```

```
name : durable bridge
category : audio video accessories
producer : durable
price : 203.86

name : durable bridge
category : audio video accessories
producer : durable
price : 203.86

0/10 positive, 0/10 negative
Do these records refer to the same thing?
(y)es / (n)o / (u)nsure / (f)inished

y

name : hp pavilion dv6-3013nr 15.6-inch laptop - argento
category : laptops
producer : hp
price : None
```

In [46]:
```
data_d[1]
```

Out[46]:
```
{'': '0',
 'Id': '1',
 'name': 'koss eq50 3-band stereo equalizer',
 'producer': 'koss',
 'description': 'the pocket-size koss 3-band equalizer delivers high-f
idelity performance and output normally reserved for more expensive ho
me systems. with a 10db boost or -10db cut range of level it features
a 3-band equalizer that allows for convenient and individual bass midr
ange and treble adjustment. power output is greater than 20mw per chan
nel providing clean and undistorted output into your favorite stereoph
ones. ergonomically designed for easy handling a rotary volume control
and on off switch are placed for convenient usage.',
 'price': 12.65,
 'category': 'headphone accessories',
 'source': 'amazon'}
```

In [53]:
```
deduper.train()
```

```
INFO:rlr.crossvalidation:using cross validation to find optimum alph
a...
/home/ubuntu/anaconda3/lib/python3.7/site-packages/rlr/crossvalidatio
n.py:122: RuntimeWarning: invalid value encountered in double_scalars
  * (true_distinct + false_distinct)))
INFO:rlr.crossvalidation:optimum alpha: 0.000100, score 0.248698071989
76637
INFO:dedupe.training:Final predicate set:
INFO:dedupe.training:(SimplePredicate: (oneGramFingerprint, name), Sim
plePredicate: (sortedAcronym, name))
INFO:dedupe.training:(SimplePredicate: (roundTo1, price), TfidfTextCan
opyPredicate: (0.8, name))
```

In [54]:
```
threshold = deduper.threshold(data_d, recall_weight=1)
threshold
```

```
INFO:dedupe.canopy_index:Removing stop word new
INFO:dedupe.canopy_index:Removing stop word 4
INFO:dedupe.canopy_index:Removing stop word black
INFO:dedupe.canopy_index:Removing stop word digital
INFO:dedupe.canopy_index:Removing stop word with
INFO:dedupe.canopy_index:Removing stop word and
INFO:dedupe.canopy_index:Removing stop word white
INFO:dedupe.canopy_index:Removing stop word case
INFO:dedupe.canopy_index:Removing stop word x
INFO:dedupe.canopy_index:Removing stop word inch
INFO:dedupe.canopy_index:Removing stop word 1
INFO:dedupe.canopy_index:Removing stop word for
INFO:dedupe.canopy_index:Removing stop word gb
INFO:dedupe.canopy_index:Removing stop word 2
INFO:dedupe.canopy_index:Removing stop word usb
INFO:dedupe.canopy_index:Removing stop word 8
INFO:dedupe.canopy_index:Removing stop word 3
INFO:dedupe.blocking:10000, 2.3534082 seconds
INFO:dedupe.blocking:20000, 4.5438192 seconds
INFO:dedupe.api:Maximum expected recall and precision
INFO:dedupe.api:recall: 0.891
INFO:dedupe.api:precision: 0.688
INFO:dedupe.api:With threshold: 0.424
```

Out[54]:
```
0.42378402
```

In [55]:
```
print('clustering...')
clustered_dupes = deduper.match(data_d, threshold)

print('# duplicate sets', len(clustered_dupes))
```

```
clustering...

INFO:dedupe.canopy_index:Removing stop word new
INFO:dedupe.canopy_index:Removing stop word 4
INFO:dedupe.canopy_index:Removing stop word black
INFO:dedupe.canopy_index:Removing stop word digital
INFO:dedupe.canopy_index:Removing stop word with
INFO:dedupe.canopy_index:Removing stop word and
INFO:dedupe.canopy_index:Removing stop word white
INFO:dedupe.canopy_index:Removing stop word case
INFO:dedupe.canopy_index:Removing stop word x
INFO:dedupe.canopy_index:Removing stop word inch
INFO:dedupe.canopy_index:Removing stop word 1
INFO:dedupe.canopy_index:Removing stop word for
INFO:dedupe.canopy_index:Removing stop word gb
INFO:dedupe.canopy_index:Removing stop word 2
INFO:dedupe.canopy_index:Removing stop word usb
INFO:dedupe.canopy_index:Removing stop word 8
INFO:dedupe.canopy_index:Removing stop word 3
INFO:dedupe.blocking:10000, 2.2952282 seconds
INFO:dedupe.blocking:20000, 4.4590712 seconds

# duplicate sets 104
```

In [56]:
```
for key, values in data_d.items():
    values['price'] = str(values['price'])
```

In [57]:
```
cluster_membership = {}
cluster_id = 0
for (cluster_id, cluster) in enumerate(clustered_dupes):
    id_set, scores = cluster
    cluster_d = [data_d[c] for c in id_set]


    canonical_rep = dedupe.canonicalize(cluster_d)
    for record_id, score in zip(id_set, scores):
        cluster_membership[record_id] = {
            "cluster id" : cluster_id,
            "canonical representation" : canonical_rep,
            "confidence": score
        }

singleton_id = cluster_id + 1

with open(output_file, 'w') as f_output, open(input_file) as f_input:
    writer = csv.writer(f_output)
    reader = csv.reader(f_input)

    heading_row = next(reader)
    heading_row.insert(0, 'confidence_score')
    heading_row.insert(0, 'Cluster ID')
    canonical_keys = canonical_rep.keys()
    for key in canonical_keys:
        heading_row.append('canonical_' + key)

    writer.writerow(heading_row)

    for row in reader:
        row_id = int(row[0])
        if row_id in cluster_membership:
            cluster_id = cluster_membership[row_id]["cluster id"]
            canonical_rep = cluster_membership[row_id]["canonical repres
            row.insert(0, cluster_membership[row_id]['confidence'])
            row.insert(0, cluster_id)
            for key in canonical_keys:
                row.append(canonical_rep[key].encode('utf8'))
        else:
            row.insert(0, None)
            row.insert(0, singleton_id)
            singleton_id += 1
            for key in canonical_keys:
                row.append(None)
        writer.writerow(row)
```

In [58]:
```
fields_of_interest = ['Cluster ID', 'confidence_score', 'Id', 'name', '
```

In [59]:
```
amazon_walmart_output = pd.read_csv('amazon_walmart_output2.csv', sep='
```

In [ ]:
```
amazon_walmart_output[amazon_walmart_output['confidence_score'] == None
```

In [ ]:
```
amazon_walmart_output = amazon_walmart_output[fields_of_interest]
```

In [61]:
```
amazon_walmart_output[amazon_walmart_output['confidence_score'] > 0.9].s
```

Out[61]:

	Cluster ID	confidence_score	Id	name	producer	description
101	1	1.00	102	Case Logic SLR Camera	Case Logic	This SLR backpack combines practicality and style with customizable organization and sleek lines. Carry everythingSLR camera lenses flash and laptopcomfortably on your back. Product Material Nylon Product Weight 3.39 lbs. Laptop Compartment Dimensions 14.9 or 14 inch PC fit in the padded laptop compartment Unzip side pocket slide your tripod inside and secure it at the top with the adjustable buckle serves as additional accessory

In [ ]: