

spacy

May 18, 2019

```
In [1]: import pandas as pd
import os
#os.system("pip3 install spacy")
#os.system("python3 -m spacy download en_core_web_sm")

import spacy
```

```
In [2]: os.listdir()
```

```
Out[2]: ['diagrams.yml',
'Untitled1.ipynb',
'key_word_search-Copy1.ipynb',
'diagrams',
'tsne.ipynb',
'spacy.ipynb',
'.ipynb_checkpoints',
'data',
'kmeans.ipynb']
```

```
In [3]: path_data = "data"
path_Social_Data = path_data + os.sep+ "Social_Data"
path_Article = path_data + os.sep+ "Articles_Text"
a = []
seps = ["|", ",", ";"]
for el in os.listdir(path_Article):
    for sep in seps:
        try:
            a.append(pd.read_csv(path_Article + os.sep + el, sep=sep))
            break
        except:
            pass
df = pd.concat(a)
```

/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:13: FutureWarning: Sorting because
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

```
del sys.path[0]
```

```
In [4]: #os.listdir(path_Article), os.listdir(path_Social_Data)
```

```
In [5]: df.keys()
df.head()["text"]
```

```
Out[5]: 0    Kenzo presents a psychedelic extravaganza for ...
1    Retro is a whole mood in this brand new fashio...
2    The model influencer stars in Jeremy Scotts C...
3    Take five designers, two photographers, one fa...
4    Feminine gets a modern rework with pop-bright ...
Name: text, dtype: object
```

```
In [6]: def key_word_search(keywords,df):
        elements_ids = []
        for index, row in df.iterrows():
            for el in keywords :
                if el in str(row["text"]) and str(row["text"]) is not None :
                    elements_ids.append(index)
        return elements_ids
```

```
In [7]: import en_core_web_sm
nlp = en_core_web_sm.load()
```

```
In [8]: def get_entities(df,nlp):
        import time
        candidates = []
        start = time.time()

        for index, row in df.iterrows():
            if( typerow["text"] != )
            doc = nlp(str(row["text"]))
            #for ent in doc.ents:
            #    if(ent.label_ == "org"):
            #        candidates.append(ent.text)
            for token in doc:
                if(token.tag_ == "NNP"):
                    candidates.append(token.text)

            if(time.time() - start > 100):
                return candidates
            #print("time taken:", time.time() - start )
            #print(len(candidates))
        return candidates
```

```
In [9]: candidates = get_entities(df,nlp)
```

```
In [10]: candidates = list(set(candidates))
```

```
In [11]: candidates
```

```
Out[11]: ['ZARA',  
          'Lacombe',  
          'Bourdin',  
          'Elicriso',  
          'Faux',  
          'PST',  
          'Jackets',  
          'Woodgate',  
          'Grégory',  
          'Annemie',  
          'VBdile',  
          'Tautz',  
          'Noon',  
          'Trousers',  
          'Burlington',  
          'Ramla',  
          'Rietveld',  
          'Maris',  
          'prometteur',  
          'carréUn',  
          'Time',  
          'MARKET',  
          'Binx',  
          'Amina',  
          'Cavallin',  
          'Mwad',  
          'Koi',  
          'MONSTER',  
          'KAELEN',  
          'Faubourg',  
          'Cassidy',  
          'Short',  
          'Jam',  
          'Hawaii',  
          'TogethHer',  
          'Rubinski',  
          'Saltspin',  
          'Augusto',  
          'Pouvez',  
          'Hardison',  
          'Hitchcock',  
          'Jorja',  
          'Condé',  
          'BASE',
```

'Lawrence',
'Legend',
'ALAIN',
'Louis',
'MKG0',
'Endocare',
'STÜSSY',
'Roversi',
'Zara',
'Town',
'Xenia',
'ARCHIVE',
'Brando',
'KANGOL',
'Boutemy',
'Dorit',
'Crochet',
'MARIANNA',
'J12',
'Krause',
'OLIVER',
'Graham',
'Mower',
'Kiko',
'hell',
'Desrues',
'Groupe',
'Liam',
'Hydrater',
'SIMMONS',
'Causse',
'DIABOLI',
'Keanu',
'SEEMANN',
'Plastiques',
'Italia',
'Tag',
'Halatte',
'ParisMaillot',
'Explorer',
'Bags',
'Isherwood',
'Que',
'Sister',
'Levi',
'Keystone',
'Inspiré',
'Shack',

'SHANNON',
'ACKERMAN',
'Emboîtant',
'Dunn',
'Plateformes',
'CRUISE',
'Woodshock',
'Il',
'Thomas',
'Shabazz',
'FXT',
'Spinel',
'livre',
'Hundle',
'Rich',
'Theatre',
'MERMAID',
'Dolby',
'Jenny',
'Carhartt',
'Spencer',
'Cole',
'Rose',
'Heathcliff',
'Engagés',
'Guardian',
'dAnne',
'Jeremy',
'ANNA',
'Spotify',
'LUCY',
'González',
'Lenczner',
'Tailleur',
'américain',
'Lucia',
'KINSHIP',
'@a8lia',
'Bart',
'Givenchy',
'USSR',
'Doublet',
'Zhang',
'TWA',
'Terrain',
'Artcurial',
'Smallable.com',
'Slashed',

'Boum',
'Erin',
'Philippines',
'ZADIG',
'Aaron',
'LN',
'lAustralie',
'Tracey',
'Dimanche',
'Christina',
'Colovos',
'Cosha',
'Hawke',
'Grede',
'Aspera',
'MULBERRY',
'Mozambique',
'Absolument',
'Eve',
'puisquelle',
'White',
'USA',
'MCQ',
'MIYAKE',
'Cam',
'H&M.',
'Paradise',
'Lockheed',
'd**Alexander',
'Ariane',
'Thebe',
'Qian',
'Coco',
'Turquie',
'Janis',
'Tenney',
'Métiers',
'Roy',
'CAITLIN',
'Confused',
'India',
'King',
'Brigitte',
'Orion',
'BEAUFILLE',
'Nice',
'GUISEPPE',
'Fluid',

'Hole',
'Cate',
'BEGÚM',
'HYUN',
'Bogey',
"d'Instagram",
'JAMES',
'Parisien',
'Bol',
'PRANGSTA',
'Cardigan',
'Bélier',
'Jiapei',
'Grainger',
'BERSANI',
"J'adore",
'Macintosh',
'Potts',
'McDonald',
'Weber',
'Hurel',
'XXème',
'Salemton',
"d'Angleterre",
'Fini',
'Bush',
'Sergio',
'paradise',
'Shrimps',
'd**Anja',
'Etichettificio',
'Elite',
'Brunschwig',
'Marketing',
'Express',
'Mélanger',
'KHALE',
'',
'Dishoom',
'lentrée',
'AI',
'MINE',
'Lockhart',
'Californien',
'Greyhound',
'lorigine',
'Astrologue',
'Juillet',

'Coleman',
'A.F.',
'Pippa',
'Queen',
'Boost',
'Sibbel',
'Brooker',
'Younger',
'Money',
'Richardson',
'Falmouth',
'Toe',
'CARHARTT',
'deau',
'Gummo',
'Seynes',
'Jetlag',
'ON',
'Clio',
'Easton',
'Shy',
'Liline',
'Alizé',
'Touch',
'Courtesy',
'Carel',
'eu',
'EMMA',
'Odyssey',
'O.K',
'Roadside',
'Pozzi',
'Diane',
'Fischer',
'Simons',
'Kazuhito',
'Ž',
'EFFETS',
'Chiquito',
'Sun',
'Arctic',
'ELISABETTA',
'TOMORROWLAND',
'Katty',
'Desmos',
'Olympia',
'MARA',
'Donald',

'Hupkes',
'HORIZN',
'RacilVeste',
'Class',
'Dogg',
'Uccello',
'Pepe',
'Bethann',
'Vincent',
'Val',
'Loris',
'Borges',
'SHERMAN',
'IKEA',
'terme',
'Dutti',
'Humberto',
'COMMON',
'Angel',
'Scout',
'York].',
'ERDEM',
'jeal',
'KANE',
'Résultat',
'GROSS',
'ATTIC',
'Xavi',
'Oursin',
'CK',
'Gianfranco',
'ANGELIA',
'Fashionistas',
'Segal',
'Scotch',
'Actors',
'Howell',
'm'aide",
'Galliera',
'l'Ange",
'DellAcqua',
'Trotter',
'Gypsy',
'MGSM',
'DEMEULEMEESTER',
'Rony',
'De',
'GAULTIER',

'McQueen',
'Armani',
'Légendes',
'Leonetta',
'Strongbow',
'Fund',
'ERES',
'Iart',
'MICRO',
'Wonderland',
'Symboliquement',
'Higgins',
'Kendall',
'Erroz',
'Paco',
'Vente',
'Warsaw',
'Déjà',
'Malèna',
'HAWAIIANAS',
'Pier',
'Bert',
'Marlene',
'Jelloun',
'Byers',
'Fall',
'Twisté',
'Abbagnato',
'Comète',
'Helmut',
'E. ',
'Ken',
'Nation',
'Emary',
'Dylan',
'Turquoise',
'Becoming',
'Oscar',
'Yasmin',
'Targaryen',
'Jones',
'Whiteley',
'Wranglers',
'TATTY',
'Fondation',
'Claudia',
'Kensit',
'Clare',

'Royce',
'CAO',
'PAKE',
'Mytheresa',
'Nations',
'Scandinave',
'Powerlong',
'Membre',
'Madonna',
'dartisanat',
'Duke',
'Cotton',
'Mirae',
'dAlexander',
'GRAY',
'348,31',
'Ratajkwski',
'J.W.',
'MBE',
'BURBERRY',
'Saoirse',
'Dayton',
'Ferguson',
'KidsSemainier',
'Rory',
'GivenchyBaskets',
'Vegas',
'Ophelia',
'Shop',
'Beyonce',
'Serpentine',
'DiMaggio',
'PANTHERELLA',
'LORENZO',
'Philipp',
'Aldridge',
'LOW',
'HD',
'd'aujourd'hui",
'Bloomland',
'Char',
'HAYFORD',
'Vicki',
'UI',
'LAURENCE',
'Krump',
'Tissu',
'Koizumi',

'ALLINA',
'FALL',
'Benedetta',
'AMI',
'e',
'dit',
'Montenapoleone',
'Christian',
'ref',
'Irina',
'lAsie',
'Juarroz',
'Rahman',
'Darwinisme',
'Helm',
'Rykiel',
'Arizona',
'SELMAN',
'XU',
'Sanchez',
'Chaque',
'Shinn',
'Dumbo',
'Ariel',
'SHOES',
'Way',
'Spiderman',
'Ambrosio',
'Nayab',
'Simpson',
'Noyce',
'Marino',
'Bloomsbury',
'CBE',
'GRM',
'Standard',
'Fearless',
'Wartime',
'HAT',
'Lazaro',
'Franz',
'Thomson',
'Jaw',
'Menna',
'BIBI',
'que',
'Baines',
'Miho',

'Aiche',
'KITE',
'PUGLISI',
'Carré',
'Clotilde',
'Princesse',
'BILLY',
'Recruiting',
'Floral',
'ROBERTO',
'Yayoi',
'Abloh',
'Cult',
'Uganda',
'DJ',
'Hindmarch',
'Américain',
'Aimé',
'Asymétrique',
'trois',
'Yorker',
'Korean',
'mexprime',
'Royer',
'Redchurch',
'Brain',
'Lolita',
'Pomme',
'Gogh',
'Skate',
'Look',
'ALISTAIR',
'Annakiki',
'Une',
'VACCARELLO',
'Stylistiquement',
'Méditerranée',
'Theseus',
'Cudi',
'Monte',
'ACHOUR',
'+1633',
'Tennessee',
'CLAIRE',
'Sidival',
'Tuscany',
'Ferré',
'Dhouib',

'KIM',
'Patrick',
'Shearling',
'Translate',
'Captivée',
'French',
'Chipperfield',
'Rianne',
'Geller',
'Lison',
'Sprouse',
'NAMILIA',
'VICKI',
'Corrèze',
'Mods',
'Zanaughtti',
'Fondatrice',
'Blaaberg',
'Movie',
'SERGIDEVCIA',
'Intern',
'Steve',
'Luca',
'GaiaSac',
'Olivine',
'Covent',
'Garcia',
'BrownsFashion.com',
'Homme',
'Brûlerie',
'PLUMES',
'Sabbat',
'Vuitton',
'Beauty',
'Stella',
'Rottingdean',
'Harness',
'Reloaded',
'AW11',
'Sonia',
'Taiwan',
'Buatsi',
'Brother',
'Herin',
'Taussauds',
'Alvarez',
'quOlivia',
'ECKHAUS',

'ANGELS',
'Casino',
'DHANAK',
'FALKE',
'Nickelodeon',
'Rousseau',
'Reggio',
'Frankie',
'PHIPPS',
'Shimatsu',
'Sischy',
'Waters',
'Phalle',
'UNISA',
'Harington',
'l'extravagance",
'Ex',
'Photo',
'Ukraine',
'Chadwick',
'Frank',
'Oséree',
'Senior',
'succède',
'Delettrez',
'AQUAZZURA',
'Year',
'Ada',
'dactualité',
'Diamond',
'Exécutif',
'Aurora',
'Don',
'Scott',
'Mayhoola',
'déclectisme',
'US',
'MOSCHINO',
'Ashish',
'jen',
'Gambino',
'NINAMOUNAH',
'Freya',
'Bissett',
'Stussy',
'Bourjois',
'Ewan',
'I',

'Nicki',
'Halloween',
'HILFIGER',
'Ozzy',
'né',
'Hawaiï',
'DORFF',
'Nottingham',
'Book',
'Saturne',
'Florinaopolis',
'Japan',
'Clement',
'BEES',
"d'Authentic",
'Sies',
'Mathieu',
'McIver',
'MISBHV',
'ACNE',
'ATIKA',
'Maccabee',
'Tali',
'Texas',
'Valentino',
'HOLLOWAY',
'pavé',
'Ichon',
'Prague',
'Cindy',
'Trevor',
'Près',
'store.com',
'Grenelle',
'Consciente',
'Noreskär',
'Cabochon',
'Mechanical',
'Kiliwatch',
'Kalliala',
'Dart',
'Thompson',
'Ruth',
'Freddie',
'Wilfried',
'Courteille',
'Level',
'Femme',

'Lopatin',
'Nouveau',
'Ibeyi',
'Leah',
'MNÁ',
'Liberty',
'Bulb',
'LEROY',
'Kang',
'Dunaway',
'Darré',
'H7000253',
'Anais',
'Archeological',
'Nash',
'Whaaat',
'Lebon',
'Agnès',
'Binga',
'Sky',
'Rebecca',
'PEBBLE',
'Fish',
'Leonor',
'RALPH',
'Buenos',
'par',
'Hynde',
'Shirts',
'Bonsai',
'Lake',
'DSM',
'Cottage',
'Opéra',
'PeaceMinusOne',
'WWAKE',
'Reeves',
'Annabelle',
'Ballets',
'Fifi',
'WOORYOUNG',
'Scooby',
'Catherine',
'Bjarke',
'Edimbourg',
'Shahidi',
'Paris',
'JACOBSEN',

'Julie',
'NEIGE',
'Roberto',
'Third',
'FERRÉ',
'Uomo',
'Chapelier',
'ERIKA',
'DAVID',
'Hero',
'Rata',
'Komono',
'Kye',
'LI',
'Dès',
'Biela',
'Adieu',
'Taxidermy',
'NOUS',
'Editorial',
'Selfridges',
'Beth',
'International',
'il',
'Bodysuit',
'Leonardo',
'Infusé',
'Inspirées',
'Debeh',
'Ursula',
'CMMN',
'Bonjour',
'Daily',
'Canadien',
'Zhao',
'Thor',
'WEAVE',
'Emilia',
'Luhrmann',
'France',
'TMC',
'Sharon',
'Turlington',
'Wetherall',
'Bill',
'AW88',
'NY',
'Saks',

'Lindbergh',
'Massiah',
'Matter',
'dune',
'Joakim',
'Chi',
'LP',
'HERON',
'Combiér',
'DANIEL',
'Star',
'DOLLS',
'Lux',
'Izabel',
'Sole',
'Barry',
'Alek',
'Vision',
'javance',
'Winnie',
'Endo',
'Ellsworth',
'Mac',
'Rawla',
'Beauté',
'Berry',
'Adèle',
'Lolo',
'Habit',
'Current',
'Dédié',
'Imaginez',
'DAFNA',
'Art',
'DEMUELEMEESTER',
'Masculin',
'JacquemusSandales',
'vest',
'KUNERT',
'Ly',
'Mitterrand',
'model',
'Adjani',
'LAV',
'Rumours',
'Manfield',
'Arte',
'Lafawndah',

'Wizard',
'Nichée',
'Museum',
'Wise',
'Muses',
'MUSED',
'Barcelos',
'Gerrit',
'Realisation',
'Bioderma',
'Queensland',
'Icelandic',
'Cullen',
'WENDY',
'VICTOR',
'Lodi',
'Adriana',
'KIT',
'Olsen',
'Choo',
'Tania',
'Blouse',
'Lipa',
'PEOPLES',
'Lognon',
'Patrice',
'Unknown',
'Grand',
'Lay',
'Coca',
'Sincere',
'Irlande',
'Blavatsky',
'Westfield',
'von',
'Kaia',
'Broadway',
'Dying',
'Takadoi',
'Ganymède',
'Latina',
'MAGDA',
'BETHANY',
'Brighton',
'Brazil',
'Chen',
'MatchesFashion.com',
'VANDERVORST',

'STEFAN',
'Apartment',
'Luis',
'Wonder',
'Permettre',
'Forss',
'PHILIPP',
'Sézalory',
'L'exposition',
'Rasika',
'Thaets',
'Beats',
'Burch',
'Miuccia',
'Delevingne',
'Europe',
'MYKITA',
'Afonzo',
'SAFILO',
'Stephania',
'Keira',
'Principal',
'Tsumori',
'Shayne',
'Eugenie',
'qu'Helmut',
'Matanda',
'ROXANNE',
'Terminator',
'MILO',
'Dorigo',
'Ekholm',
'Conservation',
'McFly',
'Londoniens',
'BGosh',
'Sluiz',
'Boy',
'RICH',
'Bisbiki',
'Kering',
'Shamballa',
'Hepburn',
'Francisco',
'Belfast',
'BEAR',
'l'art',
'trans',

'quil',
'podiumsMode',
'DOWN',
'will.i.am',
'fell/',
'Douleur',
'Rocha',
'Lucan',
'HALPERN',
'Imperial',
'BIG',
'LVHM',
'Pump',
'Feng',
'Chromat',
'Pavel',
'Cet',
'Vermenou',
'Camel',
'Ino',
'9h30',
'gamme',
'LeCoultre',
'NARCISO',
'.',
'Babies',
'Michele',
'Bobbie',
'SW19',
'Snoop',
'Pis',
'Culmone',
'Karamolegkou',
'Naomi',
'Fedez',
'Néerlandaise',
'Cambridge',
'Louie',
'Nouchi',
'Volumizer',
'Germanier',
'BOGO',
'Barralon',
'Biarritz',
'Delevigne',
'Kanye',
'Toll',
'y',

```
'Orwellian',
'YEOMAN',
'Bardot',
'ARIGATO',
'Mino',
'LIZ',
'Columbia',
'Ellington',
'Arapage',
'Lafayette',
'Iceland',
'Jacquie',
'CDTS',
'jupe',
'Ceretti',
'Kee',
'Tyler',
'McCallum',
'PALM',
'Nouvelle',
'FILIPPA',
'Thai',
'Brooke',
'Chylak',
'Voyagez',
'Aussie',
'Tjäder',
'Wipe',
'Français',
'Miró',
'Ritz',
'Forcément',
'WINNER',
'Yesterday',
'Tong',
'Bollente',
'été',
'Romee',
"l'oeuvre",
'Brooks',
'SCHUMACHER',
'Mercury',
'LIPPES',
'Joueur',
...]
```

```
In [12]: scoring = key_word_search(candidates[:100],df)
```

```
In [14]: top_brands = {}
```

```

for i in range(min(len(candidates),100)):
    top_brands[candidates[i]] = scoring[i]

```

```

In [28]: results = sorted(top_brands.items(), key=lambda kv: -kv[1])

```

```

In [29]: results
        #"https://citronclothing.com"

```

```

Out[29]: [('Il', 194),
          ('Woodshock', 191),
          ('CRUISE', 189),
          ('Plateformes', 187),
          ('Dunn', 186),
          ('SHANNON', 184),
          ('ACKERMAN', 184),
          ('Emboîtant', 184),
          ('Inspiré', 182),
          ('Shack', 182),
          ('Keystone', 175),
          ('Levi', 173),
          ('Sister', 171),
          ('Que', 170),
          ('Isherwood', 167),
          ('Bags', 165),
          ('ParisMaillot', 162),
          ('Explorer', 162),
          ('Halatte', 160),
          ('Tag', 156),
          ('Italia', 155),
          ('Plastiques', 151),
          ('SEEMANN', 150),
          ('Keanu', 147),
          ('DIABOLI', 143),
          ('Causse', 142),
          ('SIMMONS', 141),
          ('Hydrater', 138),
          ('Liam', 136),
          ('Groupe', 134),
          ('Desrues', 128),
          ('hell', 124),
          ('Kiko', 121),
          ('Mower', 120),
          ('Graham', 116),
          ('OLIVER', 113),
          ('Krause', 109),
          ('J12', 108),
          ('MARIANNA', 107),
          ('Dorit', 102),

```


('Crochet', 102),
('Boutemy', 101),
('KANGOL', 100),
('Zara', 98),
('Town', 98),
('Xenia', 98),
('ARCHIVE', 98),
('Brando', 98),
('Roversi', 96),
('STÜSSY', 95),
('Endocare', 93),
('MKGO', 92),
('Louis', 90),
('ALAIN', 86),
('Legend', 84),
('Lawrence', 83),
('BASE', 82),
('Jorja', 80),
('Condé', 80),
('Hitchcock', 78),
('Pouvez', 77),
('Hardison', 77),
('Augusto', 72),
('Rubinski', 71),
('Saltspin', 71),
('TogethHer', 68),
('Hawaii', 67),
('Jam', 65),
('Short', 61),
('Cassidy', 60),
('Faubourg', 59),
('KAELEN', 58),
('MONSTER', 54),
('Koi', 53),
('Mwad', 49),
('Cavallin', 47),
('Amina', 45),
('MARKET', 44),
('Binx', 44),
('Time', 43),
('prometteur', 33),
('carréUn', 33),
('Maris', 31),
('Rietveld', 24),
('Ramla', 21),
('Burlington', 20),
('Trousers', 17),
('Tautz', 14),

```

('Noon', 14),
('Grégory', 13),
('Annemie', 13),
('VBdile', 13),
('Jackets', 11),
('Woodgate', 11),
('PST', 10),
('Faux', 8),
('Elicriso', 6),
('Bourdin', 4),
('Lacombe', 1),
('ZARA', 0)]

```

```

In [36]: import requests
import json
import json#,httplib
import http.client

```

```

In [77]: results_dict = [ ]
for el in results :
    di = {}
    di[el[0]] = str(el[1])
    results_dict.append(di)

```

```

In [79]: def send_to_back(results_dict):
import json,http.client
connection = http.client.HTTPSConnection('devuv.back4app.io')
headers = {'Content-type': 'application/json'}

connection.request('PUT', '/classes/TrendingBrand/6xJmBY0mRF',
                    json.dumps({"objects": results_dict})
                    , {
                        "X-Parse-Application-Id": "Z4nEXPFGTTcJNvQ84WucVrYdtN81Yhv4CNu9FPoL",
                        "X-Parse-REST-API-Key": "QrvAUbLlLr0Eu0JfjTKkuBkjiePjWrmWANUaj7G",
                        "Content-Type": "application/json"
                    })
results_1 = json.loads(connection.getresponse().read())
print(results_1)

```

```

In [62]: results_dict

```

```

Out[62]: [{ 'Il': '194'},
           { 'Woodshock': '191'},
           { 'CRUISE': '189'},
           { 'Plateformes': '187'},
           { 'Dunn': '186'},
           { 'SHANNON': '184'},
           { 'ACKERMAN': '184'},
           { 'Emboîtant': '184'},

```

```
    {'Inspiré': '182'},  
    {'Shack': '182'}]
```

```
In [81]: results_dict_TOP_10 = results_dict[:10]  
        send_to_back(results_dict_TOP_10)
```

```
{'updatedAt': '2019-05-18T07:16:17.771Z'}
```

```
In [72]: black_list = [  
        "I1",  
  
        ]
```

```
In [76]: new_results_dict = []  
        for el in results_dict:  
            for e in el.keys():  
                if e not in black_list :  
                    new_results_dict.append(el)  
        print(new_results_dict)
```

```
[{'Woodshock': '191'}, {'CRUISE': '189'}, {'Plateformes': '187'}, {'Dunn': '186'}, {'SHANNON':
```

```
In [80]: send_to_back(new_results_dict)
```

```
{'updatedAt': '2019-05-18T07:15:52.867Z'}
```

```
In [ ]:
```