Veridion_DataCleaning

September 21, 2024

[1]:

```
import pandas as pd
import numpy as np
import pycountry
from publicsuffixlist import PublicSuffixList
import phonenumbers
from geopy.geocoders import Nominatim
from geopy.exc import GeocoderTimedOut

website_df = pd.read_csv('website_dataset.csv', sep=';')

google_df = pd.read_csv('google_dataset.csv', sep=',', on_bad_lines='skip',␣
 ↪low_memory=False)

facebook_df = pd.read_csv('facebook_dataset.csv', sep=',', on_bad_lines='skip',␣
 ↪low_memory=False)

cities_df = pd.read_csv("archive/worldcities.csv")
```

[2]:

```
# Redenumimi coloanele în website_df
website_df.rename(columns={
    'main_city': 'city',
    'main_country': 'country_name',
    'main_region': 'region_name',
    's_category': 'category',
    'root_domain': 'domain',
    'site_name': 'name'
}, inplace=True)

# curățarea numelui orașului
def clean_city_name(city_name):
    if pd.isna(city_name):
        return np.nan
    city_name = city_name.strip().strip('-').strip(',').strip('.') # Eliminăm␣
 ↪spațiile și semnele de punctuație
    city_name = ' '.join(word.capitalize() for word in city_name.split()) #␣
 ↪Capitalizăm fiecare cuvânt
    return city_name
```

```
website_df['city'] = website_df['city'].apply(clean_city_name)

# capitalizarea numelui țărilor
website_df['country_name'] = website_df['country_name'].str.title()

# curățarea numelui țării
def clean_country_name(country_name):
    if pd.isna(country_name):
        return np.nan

    country_name = ''.join([char for char in country_name if not char.
 ↪isdigit()])

    country_name = country_name.strip()

    if not country_name:
        return np.nan

    countries = [country.name for country in pycountry.countries]

    if country_name in countries:
        return country_name
    else:
        return np.nan

website_df['country_name'] = website_df['country_name'].
 ↪apply(clean_country_name)

# Capitalizează numele regiunii
website_df['region_name'] = website_df['region_name'].str.title()
```

[3]:

```
merged_df = pd.merge(website_df, cities_df[['city', 'country']], how='left',␣
 ↪on='city')

# Complet[m valorile NaN din coloana 'country_name' cu valorile corespunzătoare␣
 ↪din 'merged_df'
website_df['country_name'] = website_df['country_name'].
 ↪fillna(merged_df['country'])
```

[4]:

```
# domain_suffix cleaning
psl = PublicSuffixList()

def is_valid_domain(domain):
    if pd.isna(domain):
        return False
    if ' ' in domain:
        return False
    if '.' not in domain:
        return False
    if len(domain) < 5 or len(domain) > 253:
        return False
    for char in domain:
        if not (char.isalnum() or char in {'.', '-'}):
            return False
    return True

def extract_and_validate_suffix(domain):
    if pd.isna(domain) or not is_valid_domain(domain):
        return np.nan
    extracted_suffix = psl.publicsuffix(domain)
    return extracted_suffix if psl.is_public(extracted_suffix) else np.nan

website_df['domain_suffix'] = website_df['domain'].apply(extract_and_validate_suffix)

# Filtrează website-urile valide care au sufix de domeniu
valid_websites_df = website_df[website_df['domain_suffix'].notna()]

def extract_tld(domain):
    if pd.isna(domain):
        return np.nan
    else:
        parts = domain.split('.')
        if len(parts) > 1:
            return parts[-1]
        else:
            return np.nan

website_df['tld'] = website_df['domain'].apply(extract_tld)

def fill_legal_name(row):
    # Use legal_name if it's available
    if not pd.isna(row['legal_name']):
        return row['legal_name']

    if not pd.isna(row['name']):
        return row['name']

    if not pd.isna(row['domain']):
        domain_parts = row['domain'].split('.')
        if len(domain_parts) > 1:
            return domain_parts[0].replace('-', ' ').title()
    return np.nan

website_df['legal_name'] = website_df.apply(fill_legal_name, axis=1)

# Name cleaning
def fill_name(row):
    if not pd.isna(row['name']):
        return row['name']

    if not pd.isna(row['legal_name']):
        return row['legal_name']

    if not pd.isna(row['domain']):
        domain_parts = row['domain'].split('.')
        if len(domain_parts) > 1:
            return domain_parts[0].replace('-', ' ').title()

    return np.nan

website_df['name'] = website_df.apply(fill_name, axis=1)

# Elimină rândurile cu name = NaN
website_df = website_df.dropna(subset=['name'])

# Eliminam randurile cu phone = NaN
website_df = website_df.dropna(subset=['phone'])

# Elimină rândurile cu valori NaN în 'city' și 'region_name'
website_df = website_df.dropna(subset=['city', 'region_name'])

# phone cleaning
def get_iso_country_code(country_name):
    try:
        country = pycountry.countries.get(name=country_name)
        if country:
            return country.alpha_2
        return None
    except LookupError:
        return None

def validate_and_format(phone, country_name):
    try:
        if pd.isna(phone) or pd.isna(country_name):
            return np.nan

        region_code = get_iso_country_code(country_name)
        if not region_code:
            return np.nan

        parsed_number = phonenumbers.parse(phone, region_code)

        if phonenumbers.is_valid_number(parsed_number):
            return phonenumbers.format_number(parsed_number, phonenumbers.
 ↪PhoneNumberFormat.E164)
        else:
            return np.nan
    except phonenumbers.phonenumberutil.NumberParseException:
        return np.nan

validated_phone_numbers = []

for index, row in website_df.iterrows():
    validated_phone = validate_and_format(row['phone'], row['country_name'])

    validated_phone_numbers.append(validated_phone)

website_df['original_phone'] = website_df['phone']

website_df['phone'] = validated_phone_numbers

# Elimină rândurile cu numere de telefon NaN
website_df = website_df.dropna(subset=['phone'])

# Completează categoriile NaN cu 'Other'
website_df['category'] = website_df['category'].fillna('Other')

# Completează limbile NaN cu 'Unknown'
website_df['language'] = website_df['language'].fillna('Unknown')

# Creează o coloană 'address' din numele țării, regiunea și orașul
website_df['address'] = website_df['country_name'] + ', ' +␣
 ↪website_df['region_name'] + ', ' + website_df['city']

print("\nWebsite dataset: \n")
print(website_df.head(10))
```

```
Website dataset:

                   domain domain_suffix language  \
1     clothesencounter.ca            ca       en
2          investa.com.au        com.au       en
3       timminsgarage.com           com       en
```

```
5             ironcrow.ca            ca       en
6       springboarddm.com           com       en
7    stoneandtilesrus.com           com       en
8            mdaccpap.com           com       en
9             micacchi.ca            ca       en
10       skicollision.com           com       en
11  libertelightdance.com           com       en

                                    legal_name         city country_name  \
1                            Clothes Encounter     Cardigan       Canada   
2   Investa Wholesale Funds Management Limited     Brisbane    Australia   
3                          Timmins Garage Inc.      Timmins       Canada   
5                                    Iron Crow      Calgary       Canada   
6                          The Hershey Company  Mississauga       Canada   
7                         STONE AND TILES R US     Brampton       Canada   
8                                MDAC CPAP INC  Thunder Bay       Canada   
9                        Micacchi Architecture      Toronto       Canada   
10                       SKI Collision & Glass     Winnipeg       Canada   
11   LIBERTE LIGHT DANCE, SCHOOL OF DANCE INC.    Moose Jaw       Canada   

             region_name         phone                         name  tld  \
1   Prince Edward Island  +13066937766            Clothes Encounter   ca   
2             Queensland  +61282269300       Investa Property Group   au   
3                Ontario  +18775896640               Timmins Garage  com   
5                Alberta  +14032878770                    Iron Crow   ca   
6                Ontario  +19053690553  SpringBoard Data Management  com   
7                Ontario  +19054940660         STONE AND TILES R US  com   
8                Ontario  +18076834405                    MDAC CPAP  com   
9                Ontario  +16477257799        Micacchi Architecture   ca   
10              Manitoba  +12042989266        SKI Collision & Glass  com   
11          Saskatchewan  +13069900067          Liberte Light Dance  com   

                               category original_phone  \
1         Shoes & Other Footwear Stores    13066937766   
2                Real Estate Developers    61282269300   
3    Automobile Dealers & Manufacturers    18775896640   
5                      Furniture Stores    14032878770   
6                        Data Solutions    19053690553   
7                            Tile Store    19054940660   
8          Medical Supply Manufacturers    18076834405   
9   Architects & Architectural Services    16477257799   
10                      Auto Body Shops    12042989266   
11                        Dance Schools    13069900067   

                                   address  
1   Canada, Prince Edward Island, Cardigan  
2          Australia, Queensland, Brisbane  
3                 Canada, Ontario, Timmins  
```

```
5                 Canada, Alberta, Calgary
6             Canada, Ontario, Mississauga
7                Canada, Ontario, Brampton
8             Canada, Ontario, Thunder Bay
9                 Canada, Ontario, Toronto
10              Canada, Manitoba, Winnipeg
11         Canada, Saskatchewan, Moose Jaw
```

[5]:

```
# Schimba numele coloanei categories
facebook_df.rename(columns={
    'categories': 'category'
}, inplace=True)

def extract_domain_suffix(domain):
    if pd.notna(domain) and '.' in domain:
        parts = domain.split('.')
        return parts[-1]  # Returnează ultimul element (sufixul)
    return None

# Adaugă coloana domain_suffix în datasetul Facebook
facebook_df['domain_suffix'] = facebook_df['domain'].
 ↪apply(extract_domain_suffix)

# Capitalizeaza numele țărilor și orașelor
facebook_df['country_name'] = facebook_df['country_name'].str.title()
facebook_df['city'] = facebook_df['city'].str.title()

# Elimină coloana phone_country_code
facebook_df.drop(columns=['phone_country_code'], inplace=True)

# Funcție pentru a curăța numerele de telefon din Facebook
def clean_facebook_phone(raw_phone):
    if pd.isna(raw_phone):
        return None
    cleaned = ''.join(filter(str.isdigit, str(raw_phone)))  # Convertim în␣
 ↪string și păstrăm doar cifrele
    return cleaned if cleaned else None

# Curățăm coloana de telefon
facebook_df['cleaned_phone'] = facebook_df['phone'].apply(clean_facebook_phone)

# Validăm și formatază numerele de telefon folosind aceeași logică
validated_facebook_numbers = []

for index, row in facebook_df.iterrows():
    validated_phone = validate_and_format(row['cleaned_phone'],␣
 ↪row['country_name'])  # Folosim funcția de validare
```

```
    validated_facebook_numbers.append(validated_phone)  # Adăugăm numărul␣
  ↪validat în listă

# Salvează numărul original
facebook_df['original_phone'] = facebook_df['phone']

# Înlocuim coloana de telefon cu numerele validate
facebook_df['phone'] = validated_facebook_numbers

# Curățăm coloana de telefon pentru a elimina orice NaN
facebook_df.dropna(subset=['phone'], inplace=True)

# Crează un mapping pentru țări
country_mapping = dict(zip(cities_df['country'], cities_df['iso2']))

# Completează country_code folosind country_name
facebook_df['country_code'] = facebook_df['country_code'].
  ↪fillna(facebook_df['country_name'].map(country_mapping))

# Crează un mapping invers pentru codurile țărilor
reverse_country_mapping = dict(zip(cities_df['iso2'], cities_df['country']))

# Completează country_name folosind country_code
facebook_df['country_name'] = facebook_df['country_name'].
  ↪fillna(facebook_df['country_code'].map(reverse_country_mapping))

# Elimină rândurile cu country_name NaN
facebook_df = facebook_df.dropna(subset=['country_name'])

# Completează adresa cu orașul dacă este NaN
facebook_df['address'] = facebook_df['address'].fillna(facebook_df['city'])

# Setează country_code la 'NA' pentru Namibia
facebook_df.loc[facebook_df['country_name'] == 'Namibia', 'country_code'] = 'NA'

# Completează valorile lipsă în datasetul Facebook
facebook_df['category'] = facebook_df['category'].fillna('Other')
facebook_df['description'] = facebook_df['description'].fillna('Check the␣
  ↪website to find more')
facebook_df['email'] = facebook_df['email'].fillna('Check the website to find␣
  ↪more')

# Completează region_code și region_name cu 'Unknown' dacă sunt NaN
facebook_df['region_code'] = facebook_df['region_code'].fillna('Unknown')
facebook_df['region_name'] = facebook_df['region_name'].fillna('Unknown')

# Completează zip_code cu 'Unknown' dacă este NaN
```

```python
facebook_df['zip_code'] = facebook_df['zip_code'].fillna('Unknown')

# Elimină rândurile unde city este NaN
facebook_df = facebook_df[facebook_df['city'].notna()]

print("\nFacebook dataset: \n")
print(facebook_df.columns)

print(facebook_df.head(10))

print(facebook_df.isnull().sum())
```

```
Facebook dataset:

Index(['domain', 'address', 'category', 'city', 'country_code', 'country_name',
       'description', 'email', 'link', 'name', 'page_type', 'phone',
       'region_code', 'region_name', 'zip_code', 'domain_suffix',
       'cleaned_phone', 'original_phone'],
      dtype='object')
                      domain  \
984             mymuesli.com
1532            findex.co.nz
1668           inditrade.com
1866      pema-stuck-putz.de
1986  permasteelisagroup.com
2295   digitalprinting.co.id
2474   haarchitekt-landau.de
2776   city-personalbuero.de
3474          global-office.de
3618           libo-media.de

                                                address  \
984       sailerwöhr 16, 94032, passau, germany, bayern
1532  level 29, 188 quay street, 1010, auckland, new…
1668  fifth floor, phoenix house, senapati bapat mar…
1866  sachsstraße 26, 50259, pulheim, germany, nordr…
1986  viale e. mattei 21/23, 31029, vittorio veneto,…
2295  jl. asshirot ii no.15 kebayoran lama,jakarta b…
2474  neustadterstr. 24 a, 76829, landau in der pfal…
2776  ostbahnstraße 17, 76829, landau in der pfalz, …
3474   gautinger str. 26a, 82061, neuried (bei münchen)
3618  berliner straße 13, 14547, beelitz, germany, b…

                                               category                 city  \
984    Organic farms|Salvage Merchandise & Thrift Store               Passau
1532                  Accounting & Bookkeeping Services             Auckland
1668        Investment Consultants & Financial Advisors               Mumbai
```

```
1866  General Contractors|Home Builders & Renovation…             Pulheim  
1986                       Other Engineering Services      Vittorio Veneto
2295                                            Other              Jakarta
2474                                      Hair Salons  Landau In Der Pfalz
2776                                            Other  Landau In Der Pfalz
3474          Business Consulting|Business Consulting              Neuried
3618                     Digital & Marketing Agencies              Beelitz

     country_code country_name                 description  \
984            de      Germany  Check the website to find more
1532           nz  New Zealand  Check the website to find more
1668           in        India  Check the website to find more
1866           de      Germany  Check the website to find more
1986           it        Italy  Check the website to find more
2295           id    Indonesia  Check the website to find more
2474           de      Germany  Check the website to find more
2776           de      Germany  Check the website to find more
3474           de      Germany  Check the website to find more
3618           de      Germany  Check the website to find more

                               email                              link  \
984   Check the website to find more               https://mymuesli.com
1532  Check the website to find more               https://findex.co.nz
1668  Check the website to find more              https://inditrade.com
1866  Check the website to find more         https://pema-stuck-putz.de
1986  Check the website to find more     https://permasteelisagroup.com
2295  Check the website to find more      https://digitalprinting.co.id
2474  Check the website to find more      https://haarchitekt-landau.de
2776  Check the website to find more      https://city-personalbuero.de
3474  Check the website to find more  https://global-office.de/muenchen
3618  Check the website to find more              https://libo-media.de

                              name      page_type           phone  \
984                       mymuesli  LocalBusiness  +49851986990010
1532            Findex New Zealand  LocalBusiness    +648004945690
1668     Inditrade Capital Limited  LocalBusiness  +91180026687030
1866          Stuckateur Pema GmbH  LocalBusiness  +49492234834030
1986           Permasteelisa Group  LocalBusiness   +3904385050000
2295         Digitalprinting.co.id   Organization   +6281181998860
2474                   Haarchitekt  LocalBusiness  +49634155750950
2776  CiP city personalbüro Landau  LocalBusiness  +49496341987800
3474         global office München  LocalBusiness  +49892441056300
3618      Libo Media Druck&Werbung  LocalBusiness  +49332046049750

     region_code             region_name zip_code domain_suffix  \
984           by                 bavaria    94032           com
1532         auk                auckland     1010            nz
1668          mh             maharashtra  Unknown           com
```

```
1866          nw  north rhine-westphalia    50259            de
1986          34                  veneto    31029           com
2295          jk                 jakarta    11480            id
2474          rp         rheinland-pfalz    76829            de
2776          rp         rheinland-pfalz    76829            de
3474          by                  bayern    82061            de
3618          bb             brandenburg    14547            de

       cleaned_phone  original_phone
984   49851986990010    4.985199e+12
1532    648004945690    6.480049e+10
1668  91180026687030    9.118003e+12
1866    492234834030    4.922348e+10
1986   3904385050000    3.904385e+11
2295   6281181998860    6.281182e+11
2474  49634155750950    4.963416e+12
2776    496341987800    4.963420e+10
3474  49892441056300    4.989244e+12
3618  49332046049750    4.933205e+12
domain            0
address           0
category          0
city              0
country_code      0
country_name      0
description       0
email             0
link              0
name              0
page_type         0
phone             0
region_code       0
region_name       0
zip_code          0
domain_suffix     0
cleaned_phone     0
original_phone    0
dtype: int64
```

[ ]:

```
```

[6]:

```python
# Adaugă coloana domain_suffix în datasetul Google
google_df['domain_suffix'] = google_df['domain'].apply(extract_domain_suffix)

print("\nGoogle dataset: \n")
print(google_df.columns)
```

```
# Elimină rândurile cu name = NaN
google_df = google_df.dropna(subset=['name'])

# Curățarea și validarea telefonului
def clean_phone(raw_phone):
    if pd.isna(raw_phone):
        return None
    # Curățăm numărul
    cleaned = ''.join(filter(str.isdigit, raw_phone))
    return cleaned if cleaned else None

google_df['cleaned_raw_phone'] = google_df['raw_phone'].apply(clean_phone)

# Completează coloana phone cu valorile din cleaned_raw_phone dacă sunt␣
 ↪disponibile
google_df['phone'] = google_df['phone'].
 ↪combine_first(google_df['cleaned_raw_phone'])

# Elimină coloana temporară cleaned_raw_phone
google_df.drop(columns=['cleaned_raw_phone'], inplace=True)

# Elimină rândurile unde phone este NaN
google_df = google_df.dropna(subset=['phone'])

# Elimină rândurile unde city este NaN
google_df = google_df.dropna(subset=['city'])

for index, row in google_df.iterrows():
    if pd.isna(row['raw_address']):
        google_df.at[index, 'raw_address'] = f"{row['city']},␣
 ↪{row['region_name'] if pd.notna(row['region_name']) else ''}"

google_df['country_name'] = google_df['country_name'].str.title()

# Schimbă denumirea pentru Coreea de Sud
google_df.loc[google_df['country_name'] == 'South Korea', 'country_name'] =␣
 ↪'Korea, South'

country_mapping = dict(zip(cities_df['country'], cities_df['iso2']))

# Completează country_code folosind country_name
google_df['country_code'] = google_df['country_code'].
 ↪fillna(google_df['country_name'].map(country_mapping))

print(google_df['country_code'].isna().sum())

# Elimină coloana phone_country_code
```

```python
google_df = google_df.drop(columns=['phone_country_code'])

google_df['category'] = google_df['category'].fillna('Other')

print(google_df.head(10))
```

```
Google dataset:

Index(['address', 'category', 'city', 'country_code', 'country_name', 'name',
       'phone', 'phone_country_code', 'raw_address', 'raw_phone',
       'region_code', 'region_name', 'text', 'zip_code', 'domain',
       'domain_suffix'],
      dtype='object')
0
                                              address  \
0   28 Central Coast Hwy, West Gosford NSW 2250, A…
1    400 Scott St, St. Catharines, ON L2M 3W2, Canada
2       191 Pleasant St, Yarmouth, NS B5A 2J9, Canada
3   11040 Santa Monica Blvd Suite 370, Los Angeles…
5                         Ferndale, MI, United States
7           55 S Cleveland Ave, Westerville, OH 43081
8   4050 S Torrey Pines Dr, Las Vegas, NV 89103, U…
10              125 Chesterfield Ln, Maumee, OH 43537
11                    29 Clement St, Nashua, NH 03060
12         600 N Wolfe St #7-113, Baltimore, MD 21287

                             category            city country_code  \
0             Fabric-Based Home Goods         gosford           au
1                         Book Stores  st. catharines           ca
2   Other Building Material Retailers        yarmouth           ca
3             Plastic Surgery Clinics     los angeles           us
5                         Pubs & Bars        ferndale           us
7          Preschools & Kindergartens     westerville           us
8          Preschools & Kindergartens       las vegas           us
10         Preschools & Kindergartens          maumee           us
11         Preschools & Kindergartens          nashua           us
12                        Pediatrists       baltimore           us

     country_name                                        name         phone  \
0       Australia                      Spotlight West Gosford  +61243355946
1          Canada               Heritage Christian Book Store  +19059374553
2          Canada                        Pleasant Timber Mart  +19027429181
3   United States          Skin Specifics Medical Spa West LA  +18184268353
5   United States  B. Nektar Meadery - Taproom & Headquarters  +13137446323
7   United States           South Cleveland Avenue KinderCare  +16148990026
8   United States                     Torrey Pines KinderCare  +17023670822
10  United States                           Maumee KinderCare  +14198938206
```

```
11  United States             KinderCare at Rivier University  +16038880442
12  United States                               Brem Henry MD  +14109556406

                                          raw_address        raw_phone  \
0                         West Gosford NSW, Australia  +61 2 4335 5946
1                    400 Scott St · In Grantham Plaza  +1 905-937-4553
2         7+ years in business · Yarmouth, NS, Canada  +1 902-742-9181
3   7+ years in business · 11040 Santa Monica Blvd…  +1 818-426-8353
5                         Ferndale, MI, United States  +1 313-744-6323
7              7+ years in business · Westerville, OH   (614) 899-0026
8   7+ years in business · Las Vegas, NV, United S…  +1 702-367-0822
10                  7+ years in business · Maumee, OH   (419) 893-8206
11                                         Nashua, NH   (603) 888-0442
12                                      Baltimore, MD   (410) 955-6406

   region_code      region_name  \
0          nsw  new south wales
1           on          ontario
2           ns      nova scotia
3           ca       california
5           mi         michigan
7           oh             ohio
8           nv           nevada
10          oh             ohio
11          nh    new hampshire
12          md         maryland

                                                 text zip_code  \
0   4.1 (766) · Craft store West Gosford NSW, Aust…     2250
1   4.7 (100) · Book store 400 Scott St · In Grant…  l2m 3w2
2   4.7 (40) · Building materials store 7+ years i…  b5a 2j9
3   4.3 (15) · Medical spa 7+ years in business · …    90025
5   4.8 (296) · $$ · Bar Ferndale, MI, United Stat…      NaN
7   3.6 (20) · Preschool 7+ years in business · We…    43081
8   3.8 (11) · Preschool 7+ years in business · La…    89103
10  4.6 (10) · Preschool 7+ years in business · Ma…    43537
11  5.0 (1) · Day care center Nashua, NH Closed   …   03060
12  No reviews · Neurologist Baltimore, MD (410) 9…    21287

                 domain domain_suffix
0   spotlightstores.com           com
1       bookmanager.com           com
2         timbermart.ca            ca
3             linktr.ee            ee
5             linktr.ee            ee
7        kindercare.com           com
8        kindercare.com           com
10       kindercare.com           com
```

```
11       kindercare.com           com
12  hopkinsmedicine.org           org
```

[7]:

```
# Completează zip_code cu 'Unknown' unde este NaN
google_df.loc[google_df['zip_code'].isna(), 'zip_code'] = 'Unknown'

# Completează region_name cu city unde este NaN
google_df.loc[google_df['region_name'].isna(), 'region_name'] =␣
 ↪google_df['city']

google_df.loc[google_df['region_code'].isna(), 'region_code'] = 'Unknown'

# Elimină rândurile unde domain este NaN
google_df = google_df[google_df['domain'].notna()]
```

[8]:

```
## Check for missing values
print("Website dataset missing values:")
print(website_df.isnull().sum())

print("\nGoogle dataset missing values:")
print(google_df.isnull().sum())

print("\nFacebook dataset missing values:")
print(facebook_df.isnull().sum())
```

```
Website dataset missing values:
domain            0
domain_suffix     0
language          0
legal_name        0
city              0
country_name      0
region_name       0
phone             0
name              0
tld               0
category          0
original_phone    0
address           0
dtype: int64

Google dataset missing values:
address          0
category         0
city             0
country_code     0
country_name     0
name             0
```

```
phone            0
raw_address      0
raw_phone        0
region_code      0
region_name      0
text             0
zip_code         0
domain           0
domain_suffix    0
dtype: int64

Facebook dataset missing values:
domain            0
address           0
category          0
city              0
country_code      0
country_name      0
description       0
email             0
link              0
name              0
page_type         0
phone             0
region_code       0
region_name       0
zip_code          0
domain_suffix     0
cleaned_phone     0
original_phone    0
dtype: int64
```

[9]:

```python
# Check for duplicate rows
print("\nWebsite dataset duplicates:")
print(website_df.duplicated().sum())

print("\nGoogle dataset duplicates:")
print(google_df.duplicated().sum())

print("\nFacebook dataset duplicates:")
print(facebook_df.duplicated().sum())
```

```

Website dataset duplicates:
0

Google dataset duplicates:
0
```

```
Facebook dataset duplicates:
0
```

[10]:

```
# Check data types
print("\nWebsite dataset data types:")
print(website_df.dtypes)

print("\nGoogle dataset data types:")
print(google_df.dtypes)

print("\nFacebook dataset data types:")
print(facebook_df.dtypes)
```

```

Website dataset data types:
domain            object
domain_suffix     object
language          object
legal_name        object
city              object
country_name      object
region_name       object
phone             object
name              object
tld               object
category          object
original_phone    object
address           object
dtype: object

Google dataset data types:
address          object
category         object
city             object
country_code     object
country_name     object
name             object
phone            object
raw_address      object
raw_phone        object
region_code      object
region_name      object
text             object
zip_code         object
domain           object
domain_suffix    object
dtype: object
```

```
Facebook dataset data types:
domain               object
address              object
category             object
city                 object
country_code         object
country_name         object
description          object
email                object
link                 object
name                 object
page_type            object
phone                object
region_code          object
region_name          object
zip_code             object
domain_suffix        object
cleaned_phone        object
original_phone      float64
dtype: object
```

[11]:

```
# Number of rows in each dataset
website_rows = website_df.shape[0]
google_rows = google_df.shape[0]
facebook_rows = facebook_df.shape[0]

print(f"Website dataset rows: {website_rows}")
print(f"Google dataset rows: {google_rows}")
print(f"Facebook dataset rows: {facebook_rows}")
```

```
Website dataset rows: 50919
Google dataset rows: 282842
Facebook dataset rows: 156
```

[12]:

```
merge_website_df = website_df[['name', 'address', 'phone', 'category']]
merge_google_df = google_df[['name', 'address', 'phone', 'category']]
merge_facebook_df = facebook_df[['name', 'address', 'phone', 'category']]

merged_df = pd.concat([merge_website_df, merge_google_df, merge_facebook_df],␣
 ↪ignore_index=True)

print(merged_df)
```

```
                                             name  \
0                               Clothes Encounter
1                          Investa Property Group
2                                  Timmins Garage
3                                       Iron Crow
```

```
4                     SpringBoard Data Management
…                                           …
333912           Landau mit allen Sinnen genießen
333913                        Minigolf Arena Köln
333914                                 Al Ain Zoo
333915                             CreaCheck GmbH
333916  Café1739 im  Heiligenthaler Hof in Landau

                                                  address             phone  \
0                  Canada, Prince Edward Island, Cardigan      +13066937766
1                         Australia, Queensland, Brisbane      +61282269300
2                                Canada, Ontario, Timmins      +18775896640
3                                Canada, Alberta, Calgary      +14032878770
4                            Canada, Ontario, Mississauga      +19053690553
…                                                   …               …
333912  hans-stichter-strasse 34, landau, germany, rhe…   +49496341513300
333913  gut clarenhof, 50226, frechen, germany, nordrh…  +494916372960110
333914  al ain, 74750, al ain, united arab emirates, a…       +9718009660
333915  hertelsbrunnenring 10, 67657, kaiserslautern, …   +49496313661300
333916  martin-luther-str. 17 (gegenüber poseidon), 76…  +494917329032780

                                                 category
0                           Shoes & Other Footwear Stores
1                                  Real Estate Developers
2                      Automobile Dealers & Manufacturers
3                                        Furniture Stores
4                                          Data Solutions
…                                                   …
333912              Monuments & Memorials|Travel Agencies
333913                       Golf Courses & Country Clubs
333914                                                Zoo
333915  Digital & Marketing Agencies|Commercial Printi…
333916                                  Bed and Breakfast

[333917 rows x 4 columns]
```

[13]:
```
# Save the merged DataFrame to a CSV file
merged_df.to_csv('merged_dataset.csv', index=False)

print(merged_df.head(10))
```

```
                          name                                 address  \
0            Clothes Encounter  Canada, Prince Edward Island, Cardigan
1       Investa Property Group         Australia, Queensland, Brisbane
2               Timmins Garage                Canada, Ontario, Timmins
3                    Iron Crow                Canada, Alberta, Calgary
4  SpringBoard Data Management            Canada, Ontario, Mississauga
5         STONE AND TILES R US               Canada, Ontario, Brampton
```

```
6                    MDAC CPAP            Canada, Ontario, Thunder Bay
7        Micacchi Architecture                Canada, Ontario, Toronto
8        SKI Collision & Glass              Canada, Manitoba, Winnipeg
9          Liberte Light Dance         Canada, Saskatchewan, Moose Jaw

          phone                             category
0  +13066937766        Shoes & Other Footwear Stores
1  +61282269300               Real Estate Developers
2  +18775896640   Automobile Dealers & Manufacturers
3  +14032878770                     Furniture Stores
4  +19053690553                       Data Solutions
5  +19054940660                           Tile Store
6  +18076834405         Medical Supply Manufacturers
7  +16477257799  Architects & Architectural Services
8  +12042989266                      Auto Body Shops
9  +13069900067                        Dance Schools
```

[ ]:

```

```