data_cleaning

May 3, 2025

1 Data Cleaning Exercise

Cleaning your data is crucial when starting a new data engineering project because it ensures the accuracy, consistency, and reliability of the dataset. Dirty data, which may include duplicates, missing values, and errors, can lead to incorrect analysis and insights, ultimately affecting the decision-making process. Data cleaning helps in identifying and rectifying these issues, providing a solid foundation for building effective data models and analytics. Additionally, clean data improves the performance of algorithms and enhances the overall efficiency of the project, leading to more trustworthy and actionable results.

Use Python, `numpy`, `pandas` and/or `matplotlib` to analyse and clean your batch data:

1.1 Import Libraries

[30]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

1.2 Load Data

Link to data source: https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets?resource=download&select=Bitcoin_tweets_dataset_2.csv

[17]:
```
df = pd.read_csv(
    'Bitcoin_tweets_dataset_2.csv',
    sep=',',
    on_bad_lines='skip',
    engine='python',
    encoding='utf-8'
)
```

1.3 Understand the Data

View the first few rows, get summary statistics and check data types

[7]:
```
df.head()
```

[7]:
```
        user_name   user_location  \
0        ChefSam  Sunshine State
1            Roy             NaN
2  Ethereum Yoda             NaN
3        Viction   Paris, France
4           Rosie          London

                                    user_description         user_created  \
0  Culinarian | Hot Sauce Artisan | Kombucha Brew…  2011-03-23 03:50:13
1  Truth-seeking pleb   • Science   • Nature    …  2022-01-30 17:41:41
2      UP or DOWN…\n.\n.\n.\n.\nPrice matters NOT.  2022-07-24 04:50:18
3  https://t.co/8M3rgdjwEe\n\n#bitcoin #blockchai…  2010-03-26 10:15:26
4  The flower language of jasmine is loyalty, res…  2013-02-16 09:57:56

   user_followers  user_friends user_favourites user_verified  \
0          4680.0        2643.0            6232         False
1           770.0        1145.0            9166         False
2           576.0           1.0               0         False
3           236.0        1829.0            2195         False
4         12731.0          46.0             134         False

                  date                                                text  \
0  2023-03-01 23:59:59  Which #bitcoin books should I think about read…
1  2023-03-01 23:59:47  @ThankGodForBTC I appreciate the message, but …
2  2023-03-01 23:59:42  #Ethereum price update: \n\n#ETH $1664.02 USD\…
3  2023-03-01 23:59:36  CoinDashboard v3.0 is here\nAvailable on ios a…
4  2023-03-01 23:59:32  #Bitcoin Short Term Fractal (4H) \n\nIn lower …

                                        hashtags              source  \
0                                   ['bitcoin']   Twitter for iPhone
1                                   ['Bitcoin']   Twitter for iPhone
2  ['Ethereum', 'ETH', 'Bitcoin', 'BTC', 'altcoin…      Twitter Web App
3                                   ['Bitcoin']  Twitter for Android
4                              ['Bitcoin', 'BTC']      Twitter Web App

  is_retweet
0      False
1      False
2      False
3      False
4      False
```

[9]:
```
df.describe(include='all')
```

[9]:
```
            user_name user_location  \
count          174336         85028
unique          39502         10595
```

```
top     Ethereum Yoda           USA
freq              8721          2146
mean               NaN           NaN
std                NaN           NaN
min                NaN           NaN
25%                NaN           NaN
50%                NaN           NaN
75%                NaN           NaN
max                NaN           NaN

                                     user_description         user_created  \
count                                          159563                170627
unique                                          33318                 37110
top     UP or DOWN…\n.\n.\n.\nPrice matters NOT.  2022-07-24 04:50:18
freq                                             8714                  8714
mean                                              NaN                   NaN
std                                               NaN                   NaN
min                                               NaN                   NaN
25%                                               NaN                   NaN
50%                                               NaN                   NaN
75%                                               NaN                   NaN
max                                               NaN                   NaN

        user_followers   user_friends user_favourites user_verified  \
count     1.698200e+05  169820.000000          169820        169820
unique             NaN            NaN           17478            61
top                NaN            NaN               0         False
freq               NaN            NaN           24354        168353
mean      1.059798e+04     771.399694             NaN           NaN
std       1.308698e+05    2677.266627             NaN           NaN
min       0.000000e+00       0.000000             NaN           NaN
25%       1.190000e+02       9.000000             NaN           NaN
50%       5.460000e+02     122.000000             NaN           NaN
75%       1.956000e+03     606.000000             NaN           NaN
max       1.878937e+07  254276.000000             NaN           NaN

                     date  \
count              169820
unique             142505
top     2023-03-01 14:00:03
freq                   14
mean                  NaN
std                   NaN
min                   NaN
25%                   NaN
50%                   NaN
75%                   NaN
```

```
max                      NaN

                                                          text     hashtags  \
count                                                   169820       169013
unique                                                  167092        40810
top     Top 5 #cryptocurrency #price jumps in last min…  ['Bitcoin']
freq                                                       287        29312
mean                                                       NaN          NaN
std                                                        NaN          NaN
min                                                        NaN          NaN
25%                                                        NaN          NaN
50%                                                        NaN          NaN
75%                                                        NaN          NaN
max                                                        NaN          NaN

                source is_retweet
count           169013     168954
unique             720          1
top     Twitter Web App      False
freq             53653     168954
mean               NaN        NaN
std                NaN        NaN
min                NaN        NaN
25%                NaN        NaN
50%                NaN        NaN
75%                NaN        NaN
max                NaN        NaN
```

[10]:
```
df.dtypes
```

[10]:
```
user_name            object
user_location        object
user_description     object
user_created         object
user_followers      float64
user_friends        float64
user_favourites      object
user_verified        object
date                 object
text                 object
hashtags             object
source               object
is_retweet           object
dtype: object
```

[11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174397 entries, 0 to 174396
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   user_name         174336 non-null  object
 1   user_location     85028 non-null   object
 2   user_description  159563 non-null  object
 3   user_created      170627 non-null  object
 4   user_followers    169820 non-null  float64
 5   user_friends      169820 non-null  float64
 6   user_favourites   169820 non-null  object
 7   user_verified     169820 non-null  object
 8   date              169820 non-null  object
 9   text              169820 non-null  object
 10  hashtags          169013 non-null  object
 11  source            169013 non-null  object
 12  is_retweet        168954 non-null  object
dtypes: float64(2), object(11)
memory usage: 17.3+ MB
```

1.4 Handle Missing Data

Identify missing values and fill or drop missing values

[13]:
```
df.isnull().sum()
```

[13]:
```
user_name              61
user_location       89369
user_description    14834
user_created         3770
user_followers       4577
user_friends         4577
user_favourites      4577
user_verified        4577
date                 4577
text                 4577
hashtags             5384
source               5384
is_retweet           5443
dtype: int64
```

[14]:
```
(df.isnull().sum() / len(df)) * 100
```

[14]:
```
user_name            0.034978
user_location       51.244574
user_description     8.505880
user_created         2.161734
```

```
user_followers       2.624472
user_friends         2.624472
user_favourites      2.624472
user_verified        2.624472
date                 2.624472
text                 2.624472
hashtags             3.087209
source               3.087209
is_retweet           3.121040
dtype: float64
```

[15]:
```
df = df.dropna()
```

[16]:
```
df.isnull().sum().sum()
```

[16]:
```
0
```

1.5 Handle Duplicates

Identify duplicates and remove them

[18]:
```
df[df.duplicated()]
```

[18]:
```
                          user_name user_location user_description  \
1534       https://t.co/dwh8jR9uho          None             None
3130       https://t.co/dwh8jR9uho          None             None
3380       https://t.co/dwh8jR9uho          None             None
3862       https://t.co/dwh8jR9uho          None             None
3975       https://t.co/Rms8bPm9sh          None             None
…                              …           …                …
172063  Via https://t.co/p6iec02QvX          None             None
172698  Via https://t.co/p6iec02QvX          None             None
173795  Via https://t.co/p6iec02QvX          None             None
173836  Via https://t.co/p6iec02QvX          None             None
174159  Via https://t.co/p6iec02QvX          None             None

       user_created  user_followers  user_friends user_favourites  \
1534           None             NaN           NaN            None
3130           None             NaN           NaN            None
3380           None             NaN           NaN            None
3862           None             NaN           NaN            None
3975           None             NaN           NaN            None
…              …                …             …              …
172063         None             NaN           NaN            None
172698         None             NaN           NaN            None
173795         None             NaN           NaN            None
173836         None             NaN           NaN            None
```

```
174159         None              NaN           NaN            None

       user_verified  date  text hashtags source is_retweet
1534            None  None  None     None   None        NaN
3130            None  None  None     None   None        NaN
3380            None  None  None     None   None        NaN
3862            None  None  None     None   None        NaN
3975            None  None  None     None   None        NaN
…                …    …     …        …      …           …
172063          None  None  None     None   None        NaN
172698          None  None  None     None   None        NaN
173795          None  None  None     None   None        NaN
173836          None  None  None     None   None        NaN
174159          None  None  None     None   None        NaN

[1060 rows x 13 columns]
```

[19]:
```
df.duplicated().sum()
```

[19]:
```
1060
```

[20]:
```
df = df.drop_duplicates()
```

[21]:
```
df.duplicated().sum()
```

[21]:
```
0
```

1.6 Handle Outliers

Identify outliers and remove or correct them

[22]:
```
df.select_dtypes(include='number').columns
```

[22]:
```
Index(['user_followers', 'user_friends'], dtype='object')
```

[24]:
```
numeric_df = df.select_dtypes(include='number')

Q1 = numeric_df.quantile(0.25)
Q3 = numeric_df.quantile(0.75)
IQR = Q3 - Q1

condition = ~((numeric_df < (Q1 - 1.5 * IQR)) | (numeric_df > (Q3 + 1.5 *␣
 ↪IQR))).any(axis=1)

df_no_outliers = df[condition]
```

[25]:

```python
print(f"Vorher: {len(df)} Zeilen")
print(f"Nachher: {len(df_no_outliers)} Zeilen")
```

```
Vorher: 173337 Zeilen
Nachher: 134832 Zeilen
```

1.7 Handle Incorrect Data Types

[26]:

```python
df.dtypes
```

[26]:

```
user_name            object
user_location        object
user_description     object
user_created         object
user_followers      float64
user_friends        float64
user_favourites      object
user_verified        object
date                 object
text                 object
hashtags             object
source               object
is_retweet           object
dtype: object
```

[27]:

```python
for col in df.select_dtypes(include='object'):
    try:
        df[col] = pd.to_numeric(df[col])
        print(f"Konvertiert: {col} → numerisch")
    except:
        try:
            df[col] = pd.to_datetime(df[col])
            print(f"Konvertiert: {col} → datetime")
        except:
            print(f"Unverändert: {col}")
```

```
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
```

```
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
/tmp/ipykernel_1412/1078092930.py:7: UserWarning: Could not infer format, so
each element will be parsed individually, falling back to `dateutil`. To ensure
parsing is consistent and as-expected, please specify a format.
  df[col] = pd.to_datetime(df[col])
```

```
Unverändert: user_name
Unverändert: user_location
Unverändert: user_description
Unverändert: user_created
Unverändert: user_favourites
Unverändert: user_verified
Unverändert: date
Unverändert: text
Unverändert: hashtags
Unverändert: source
Konvertiert: is_retweet → numerisch
```

[28]:
```
df['date'] = pd.to_datetime(df['date'], format='%Y-%m-%d %H:%M:%S',␣
↪errors='coerce')
```

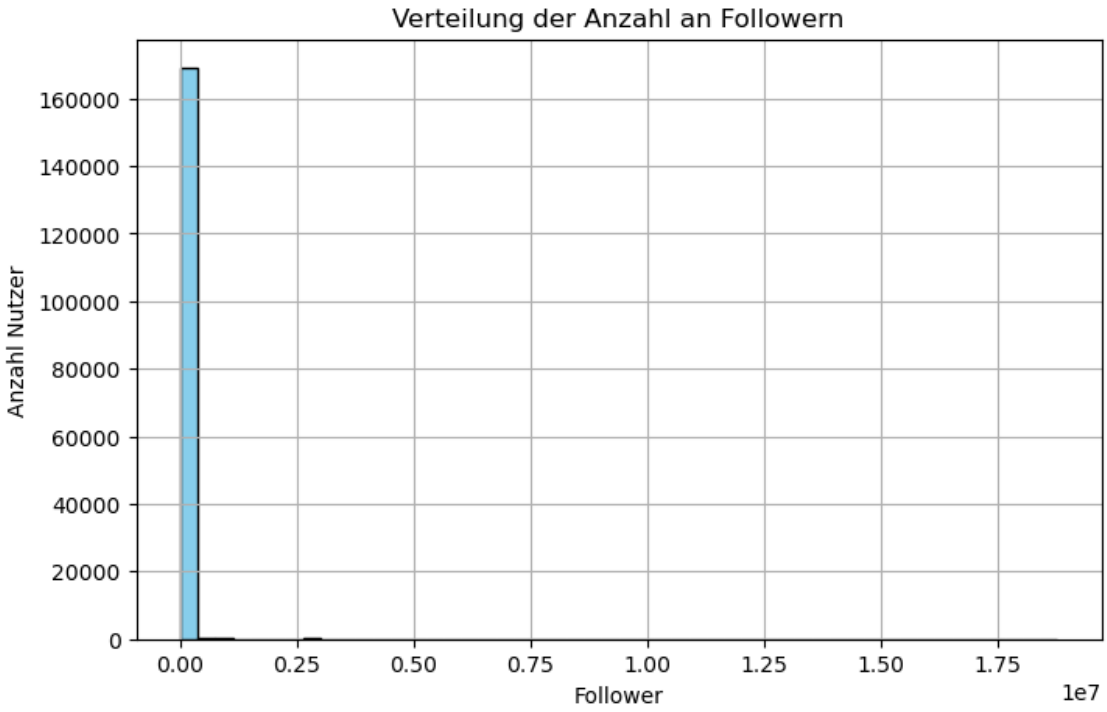
[32]:
```
df['is_retweet'] = df['is_retweet'].map({'True': 1, 'False': 0})
```

1.8 Visualize Data

Use graphes, plots and/or diagrams to visualize the data

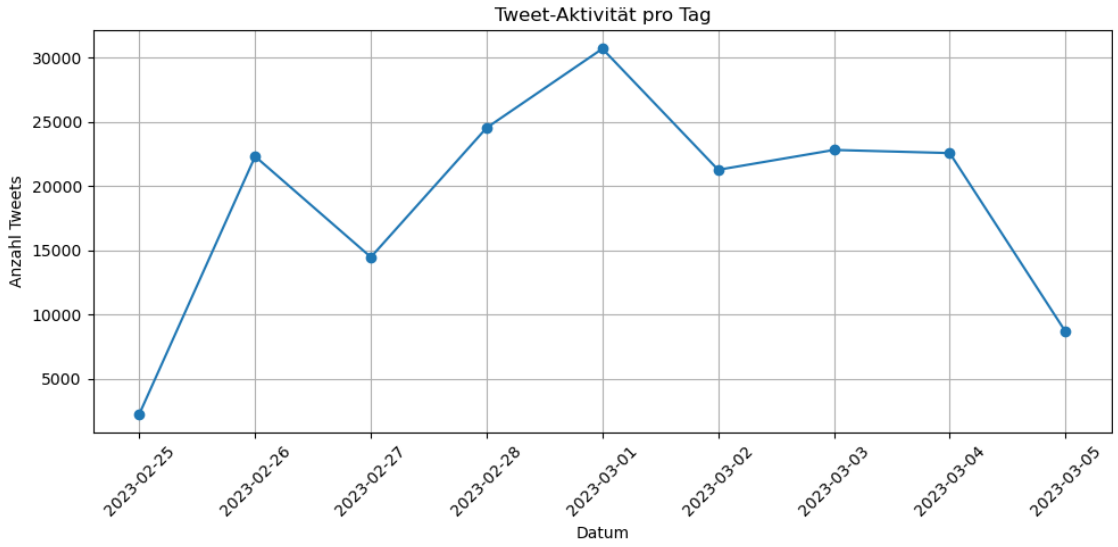
[36]:
```
plt.figure(figsize=(8, 5))
plt.hist(df['user_followers'], bins=50, color='skyblue', edgecolor='black')
plt.title('Verteilung der Anzahl an Followern')
plt.xlabel('Follower')
plt.ylabel('Anzahl Nutzer')
plt.grid(True)
```

```
plt.show()
```

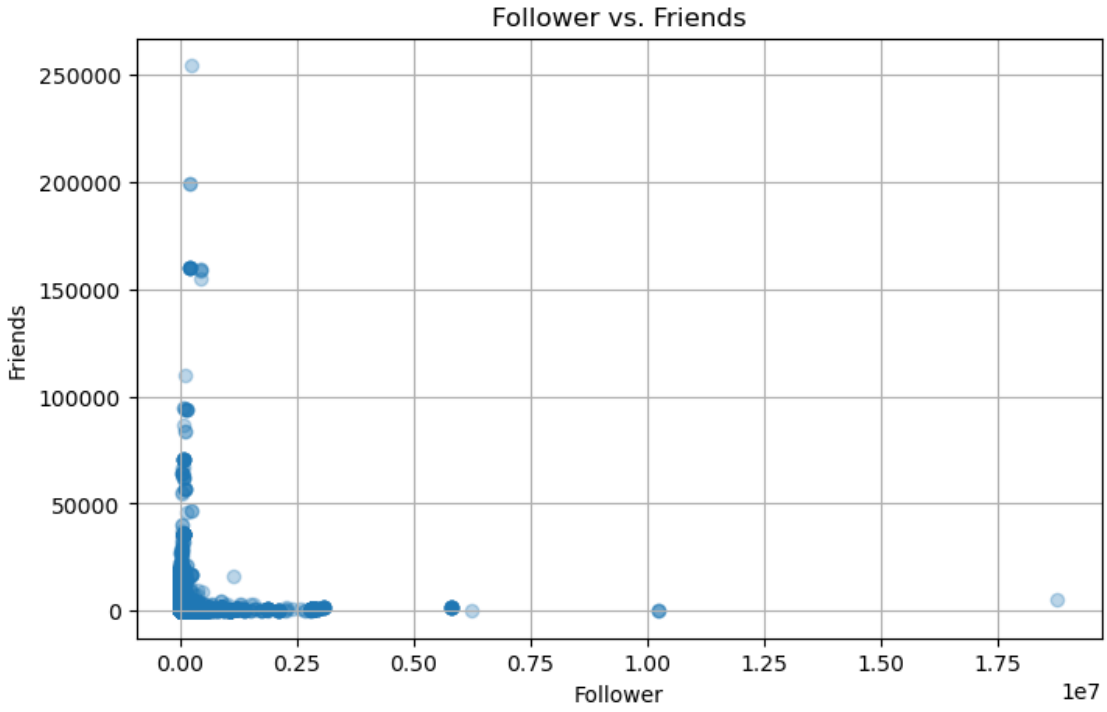


[37]:
```
tweets_per_day = df['date'].dt.date.value_counts().sort_index()

plt.figure(figsize=(10, 5))
plt.plot(tweets_per_day.index, tweets_per_day.values, marker='o', linestyle='-')
plt.title('Tweet-Aktivität pro Tag')
plt.xlabel('Datum')
plt.ylabel('Anzahl Tweets')
plt.xticks(rotation=45)
plt.tight_layout()
plt.grid(True)
plt.show()
```



[38]:
```
plt.figure(figsize=(8, 5))
plt.scatter(df['user_followers'], df['user_friends'], alpha=0.3)
plt.title('Follower vs. Friends')
plt.xlabel('Follower')
plt.ylabel('Friends')
plt.grid(True)
plt.show()
```



1.9 Save Cleaned Data

[40]:
```
save_path = "bitcoin_tweets_cleaned.csv"

df.to_csv(save_path, index=False)

print(f"Gespeicherte Datei: {save_path}")
```

```
Gespeicherte Datei: bitcoin_tweets_cleaned.csv
```