Phase 3

Fake News Detection Using NLP

Loading and Preprocessing the dataset Used for Our Model

Data Preprocessing In Machine Learning: What Is It?

Data preprocessing steps are a part of the data analysis and mining process responsible for converting raw data into a format understandable by the ML algorithms.

Text, photos, video, and other types of unprocessed, real-world data are disorganized. It may not only be inaccurate and inconsistent, but it is frequently lacking and doesn’t have a regular, consistent design. Machines prefer to process neat and orderly information; they read data as binary – 1s and 0s.

So, it is simple to calculate structured data like whole numbers and percentages. But before analysis, unstructured data, such as text and photos, must be prepped and formatted with the help of data preprocessing in Machine Learning.

Data preprocessing is the concept of changing the raw data into a clean data set. The dataset is preprocessed in order to check missing values, noisy data, and other inconsistencies before executing it to the algorithm.

Now that you know what is data preprocessing in machine learning, explore the major tasks in data preprocessing.

Data Preprocessing Steps In Machine Learning: Major Tasks Involved

- Data cleaning
- Data transformation
- Data reduction
- Data integration

Data Cleaning

Data cleaning, one of the major preprocessing steps in machine learning, locates and fixes errors or discrepancies in the data. From duplicates and outliers to missing numbers, it fixes them all. Methods like transformation, removal, and imputation help ML professionals perform data cleaning seamlessly.

Data Integration

Data integration is among the major responsibilities of data preprocessing in machine learning. This process integrates (merges) information extracted from multiple sources to outline and create a single dataset. The fact that you need to handle data in multiple forms, formats, and semantics makes data integration a challenging task for many ML developers.

Data Transformation

ML programmers must pay close attention to data transformation when it comes to data preprocessing steps. This process entails putting the data in a format that will allow for analysis. Normalization, standardization, and discretisation are common data transformation procedures. While standardization transforms data to have a zero mean and unit variance, normalization scales data to a common range. Continuous data is discretized into discrete categories using this technique.

Data transformation is the process of converting raw data into a format or structure that would be more suitable for model building and also data discovery in general. It is an imperative step in feature engineering that facilitates discovering insights.

Data Reduction

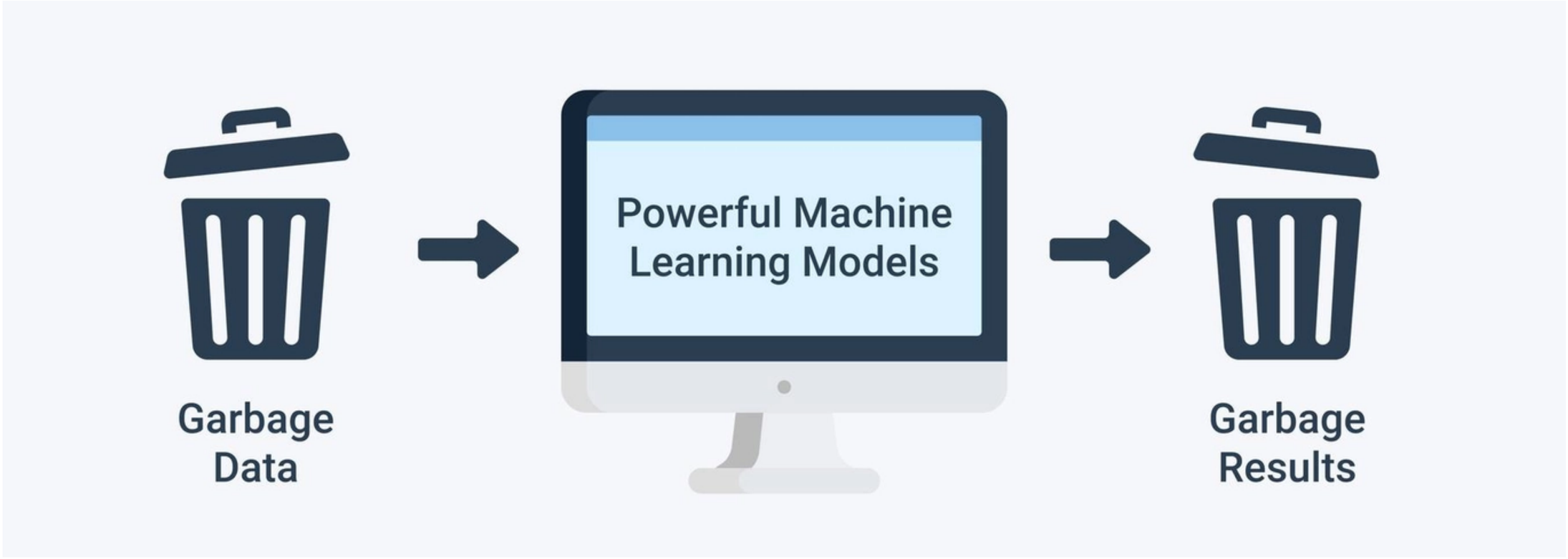
Data reduction is the process of lowering the dataset's size while maintaining crucial information. Through the use of feature selection and feature extraction algorithms, data reduction can be accomplished. While feature extraction entails translating the data into a lower-dimensional space while keeping the crucial information, feature selection requires choosing a subset of pertinent characteristics from the dataset.

Data reduction techniques, such as feature selection, feature extraction, sampling, data compression, binning, and data aggregation, enhance storage efficiency, computational speed, and model performance in machine learning and data analysis by decreasing dataset complexity and size without altering its traits.

Data Preprocessing Importance

When using data sets to train machine learning models, you'll often hear the phrase "garbage in, garbage out" This means that if you use bad or "dirty" data to train your model, you'll end up with a bad, improperly trained model that won't actually be relevant to your analysis.

Good, preprocessed data is even more important than the most powerful algorithms, to the point that machine learning models trained with bad data could actually be harmful to the analysis you're trying to do – giving you "garbage" results.


Garbage
Data
Powerful Machine
Learning Models
Garbage
Results

Depending on your data gathering techniques and sources, you may end up with data that’s out of range or includes an incorrect feature, like household income below zero or an image from a set of “zoo animals” that is actually a tree. Your set could have missing values or fields. Or text data, for example, will often have misspelled words and irrelevant symbols, URLs, etc.

When you properly preprocess and clean your data, you’ll set yourself up for much more accurate downstream processes. We often hear about the importance of “data-driven decision making,” but if these decisions are driven by bad data, they’re simply bad decisions.

Steps in Data Preprocessing in Machine Learning:

1. **Acquire the dataset**
2. **Import all the crucial libraries**
3. **Import the dataset**
4. **Identifying and handling the missing values**
5. **Encoding the categorical data**
6. **Splitting the dataset**
7. **Feature scaling**

1. Acquire the dataset

Acquiring the dataset is the first step in data preprocessing in machine learning. To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. While a business dataset will contain relevant industry and business data, a medical dataset will include healthcare-related data.

2. Import all the crucial libraries

Since Python is the most extensively used and also the most preferred library by Data Scientists around the world, we'll show you how to import Python libraries for data preprocessing in Machine Learning.

The predefined Python libraries can perform specific data preprocessing jobs. Importing all the crucial libraries is the second step in data preprocessing in machine learning. The three core Python libraries used for this data preprocessing in Machine Learning are:

- **NumPy – NumPy is the fundamental package for scientific calculation in Python. Hence, it is used for inserting any type of mathematical operation in the code. Using NumPy, you can also add large multidimensional arrays and matrices in your code.**

- **Pandas – Pandas is an excellent open-source Python library for data manipulation and analysis. It is extensively used for importing and managing the datasets. It packs in high-performance, easy-to-use data structures and data analysis tools for Python.**

- **Matplotlib – Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python. It can deliver publication-quality figures in numerous hard copy formats and interactive environments across platforms (IPython shells, Jupyter notebook, web application servers, etc.).**

Code to Import Needed Libraries:

```
import numpy as np #importing numpy.
import pandas as pd #importing pandas
import matplotlib #importing matplotlib
```

3. Import the dataset

In this step, you need to import the dataset/s that you have gathered for the ML project at hand. Importing the dataset is one of the important steps in data preprocessing in machine learning. However, before you can import the dataset/s, you must set the current directory as the working directory. You can set the working directory in Spyder IDE in three simple steps:

Save your Python file in the directory containing the dataset.
Go to File Explorer option in Spyder IDE and choose the required directory.
Now, click on the F5 button or Run option to execute the file.

Code To Import DataSet ：

```
data_set= pd.read_csv(‘Dataset.csv’)
```

Sample Data Set:

Assignment-1_Data.csv

```
BillNo;Itemname;Quantity;Date;Price;CustomerID;Country
536365;WHITE HANGING HEART T-LIGHT HOLDER;6;01.12.2010 08:26;2,55;17850;United Kingdom
536365;WHITE METAL LANTERN;6;01.12.2010 08:26;3,39;17850;United Kingdom
536365;CREAM CUPID HEARTS COAT HANGER;8;01.12.2010 08:26;2,75;17850;United Kingdom
536365;KNITTED UNION FLAG HOT WATER BOTTLE;6;01.12.2010 08:26;3,39;17850;United Kingdom
536365;RED WOOLLY HOTTIE WHITE HEART.;6;01.12.2010 08:26;3,39;17850;United Kingdom
536365;SET 7 BABUSHKA NESTING BOXES;2;01.12.2010 08:26;7,65;17850;United Kingdom
536365;GLASS STAR FROSTED T-LIGHT HOLDER;6;01.12.2010 08:26;4,25;17850;United Kingdom
536366;HAND WARMER UNION JACK;6;01.12.2010 08:28;1,85;17850;United Kingdom
536366;HAND WARMER RED POLKA DOT;6;01.12.2010 08:28;1,85;17850;United Kingdom
536367;ASSORTED COLOUR BIRD ORNAMENT;32;01.12.2010 08:34;1,69;13047;United Kingdom
536367;POPPY'S PLAYHOUSE BEDROOM;6;01.12.2010 08:34;2,1;13047;United Kingdom
536367;POPPY'S PLAYHOUSE KITCHEN;6;01.12.2010 08:34;2,1;13047;United Kingdom
536367;FELTCRAFT PRINCESS CHARLOTTE DOLL;8;01.12.2010 08:34;3,75;13047;United Kingdom
536367;IVORY KNITTED MUG COSY;6;01.12.2010 08:34;1,65;13047;United Kingdom
536367;BOX OF 6 ASSORTED COLOUR TEASPOONS;6;01.12.2010 08:34;4,25;13047;United Kingdom
536367;BOX OF VINTAGE JIGSAW BLOCKS;3;01.12.2010 08:34;4,95;13047;United Kingdom
536367;BOX OF VINTAGE ALPHABET BLOCKS;2;01.12.2010 08:34;9,95;13047;United Kingdom
536367;HOME BUILDING BLOCK WORD;3;01.12.2010 08:34;5,95;13047;United Kingdom
536367;LOVE BUILDING BLOCK WORD;3;01.12.2010 08:34;5,95;13047;United Kingdom
536367;RECIPE BOX WITH METAL HEART;4;01.12.2010 08:34;7,95;13047;United Kingdom
536367;DOORMAT NEW ENGLAND;4;01.12.2010 08:34;7,95;13047;United Kingdom
536368;JAM MAKING SET WITH JARS;6;01.12.2010 08:34;4,25;13047;United Kingdom
536368;RED COAT RACK PARIS FASHION;3;01.12.2010 08:34;4,95;13047;United Kingdom
536368;YELLOW COAT RACK PARIS FASHION;3;01.12.2010 08:34;4,95;13047;United Kingdom
536368;BLUE COAT RACK PARIS FASHION;3;01.12.2010 08:34;4,95;13047;United Kingdom
536369;BATH BUILDING BLOCK WORD;3;01.12.2010 08:35;5,95;13047;United Kingdom
536370;ALARM CLOCK BAKELIKE PINK;24;01.12.2010 08:45;3,75;12583;France
536370;ALARM CLOCK BAKELIKE RED;24;01.12.2010 08:45;3,75;12583;France
536370;ALARM CLOCK BAKELIKE GREEN;12;01.12.2010 08:45;3,75;12583;France
536370;PANDA AND BUNNIES STICKER SHEET;12;01.12.2010 08:45;0,85;12583;France
536370;STARS GIFT TAPE;24;01.12.2010 08:45;0,65;12583;France
536370;INFLATABLE POLITICAL GLOBE;48;01.12.2010 08:45;0,85;12583;France
536370;VINTAGE HEADS AND TAILS CARD GAME;24;01.12.2010 08:45;1,25;12583;France
536370;SET/2 RED RETROSPOT TEA TOWELS;18;01.12.2010 08:45;2,95;12583;France
536370;ROUND SNACK BOXES SET OF4 WOODLAND;24;01.12.2010 08:45;2,95;12583;France
536370;SPACEBOY LUNCH BOX;24;01.12.2010 08:45;1,95;12583;France
536370;LUNCH BOX I LOVE LONDON;24;01.12.2010 08:45;1,95;12583;France
536370;CIRCUS PARADE LUNCH BOX;24;01.12.2010 08:45;1,95;12583;France
536370;CHARLOTTE BAG DOLLY GIRL DESIGN;20;01.12.2010 08:45;0,85;12583;France
536370;RED TOADSTOOL LED NIGHT LIGHT;24;01.12.2010 08:45;1,65;12583;France
536370;SET 2 TEA TOWELS I LOVE LONDON;24;01.12.2010 08:45;2,95;12583;France
```

More 522065 rows

4. Identifying and handling the missing values

In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project.

Basically, there are two ways to handle missing data:

- **Deleting a particular row** – In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75% of the values are missing. However, this method is not 100% efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias.

- **Calculating the mean** – This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to the first method (omission of rows/columns). Another way of approximation is through the deviation of neighbouring values. However, this works best for linear data.

5. Encoding the categorical data

Categorical data refers to the information that has specific categories within the dataset. In the dataset cited above, there are two categorical variables – country and purchased.

Machine Learning models are primarily based on mathematical equations. Thus, you can intuitively understand that keeping the categorical data in the equation will cause certain issues since you would only need numbers in the equations.

6. Splitting the dataset

Splitting the dataset is the next step in data preprocessing in machine learning. Every dataset for Machine Learning model must be split into two separate sets – training set and test set.

Training set denotes the subset of a dataset that is used for training the machine learning model. Here, you are already aware of the output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes.

Sample code to Split DataSet:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2,
random_state=0)
```

- **x_train – features for the training data**
- **x_test – features for the test data**
- **y_train – dependent variables for training data**
- **y_test – independent variable for testing data**

7. Feature scaling

Feature scaling marks the end of the data preprocessing in Machine Learning. It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds.

```
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
```

Sample Program Which Has all this Steps:

```
import numpy as np #importing numpy.
import pandas as pd #importing pandas
import matplotlib #importing matplotlib
df = pd.read_csv("Assignment-1_Data.csv",sep = ';')
df=df.dropna()
df
d=df[df['Quantity']>=0]
d
dd = list(d["Itemname"].apply(lambda x:x.split(" ")))
dd
one_hot_transformer = TransactionEncoder()
df_transform = one_hot_transformer.fit_transform(df)
df = pd.DataFrame(df_transform,columns=one_hot_transformer.columns_)
df
```

Expected Output:

```
[76]: import pandas as pd

[77]: df = pd.read_csv("Assignment-1_Data.csv",sep = ';')

C:\Users\akfla\AppData\Local\Temp\ipykernel_20560\2820828009.py:1: DtypeWarning: Columns (0) have mixed types. Specify dtype option on import or set low_
memory=False.
  df = pd.read_csv("Assignment-1_Data.csv",sep = ';')

[ ]:

[79]: df=df.dropna()

[80]: df
```

[80]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	09.12.2011 12:50	0,85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	09.12.2011 12:50	2,1	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	09.12.2011 12:50	4,15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	09.12.2011 12:50	4,15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	09.12.2011 12:50	4,95	12680.0	France

388023 rows × 7 columns

[88]:
```
d=df[df['Quantity']>=0]
```

[87]:
```
d
```

[87]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
46	536371	PAPER CHAIN KIT 50'S CHRISTMAS	80	01.12.2010 09:00	2,55	13748.0	United Kingdom
96	536378	PACK OF 72 RETROSPOT CAKE CASES	120	01.12.2010 09:37	0,42	14688.0	United Kingdom
102	536378	RED CHARLIE+LOLA PERSONAL DOORSIGN	96	01.12.2010 09:37	0,38	14688.0	United Kingdom
174	536386	JUMBO BAG BAROQUE BLACK WHITE	100	01.12.2010 09:57	1,65	16029.0	United Kingdom
175	536386	JUMBO BAG RED RETROSPOT	100	01.12.2010 09:57	1,65	16029.0	United Kingdom
...	...	...	...	...	...	...	...
521860	581566	HOME SWEET HOME BLACKBOARD	144	09.12.2011 11:50	3,26	18102.0	United Kingdom
521861	581567	COCKLE SHELL DISH	84	09.12.2011 11:56	0,79	16626.0	United Kingdom
521869	581567	AGED GLASS SILVER T-LIGHT HOLDER	144	09.12.2011 11:56	0,55	16626.0	United Kingdom
521901	581571	SMALL CERAMIC TOP STORAGE JAR	96	09.12.2011 12:00	0,69	15311.0	United Kingdom
522022	581584	RED FLOCK LOVE HEART PHOTO FRAME	72	09.12.2011 12:25	0,72	13777.0	United Kingdom

10116 rows × 7 columns

[89]:
```
dd = list(d["Itemname"].apply(lambda x:x.split(" ")))
```

[90]:
```
dd
```

[90]:
```
[['WHITE', 'HANGING', 'HEART', 'T-LIGHT', 'HOLDER'],
 ['WHITE', 'METAL', 'LANTERN'],
 ['CREAM', 'CUPID', 'HEARTS', 'COAT', 'HANGER'],
 ['KNITTED', 'UNION', 'FLAG', 'HOT', 'WATER', 'BOTTLE'],
 ['RED', 'WOOLLY', 'HOTTIE', 'WHITE', 'HEART.'],
 ['SET', '7', 'BABUSHKA', 'NESTING', 'BOXES'],
 ['GLASS', 'STAR', 'FROSTED', 'T-LIGHT', 'HOLDER'],
 ['HAND', 'WARMER', 'UNION', 'JACK'],
 ['HAND', 'WARMER', 'RED', 'POLKA', 'DOT'],
 ['ASSORTED', 'COLOUR', 'BIRD', 'ORNAMENT'],
 ["POPPY'S", 'PLAYHOUSE', 'BEDROOM'],
 ["POPPY'S", 'PLAYHOUSE', 'KITCHEN'],
 ['FELTCRAFT', 'PRINCESS', 'CHARLOTTE', 'DOLL'],
 ['IVORY', 'KNITTED', 'MUG', 'COSY'],
 ['BOX', 'OF', '6', 'ASSORTED', 'COLOUR', 'TEASPOONS'],
 ['BOX', 'OF', 'VINTAGE', 'JIGSAW', 'BLOCKS'],
 ['BOX', 'OF', 'VINTAGE', 'ALPHABET', 'BLOCKS'],
 ['HOME', 'BUILDING', 'BLOCK', 'WORD'],
```

[73]:
```
one_hot_transformer = TransactionEncoder()
df_transform = one_hot_transformer.fit_transform(df)
```

[74]:
```
df = pd.DataFrame(df_transform,columns=one_hot_transformer.columns_)
```

[75]:
```
df
```

[75]:

	B	C	D	I	N	P	Q	a	c	e	i	l	m	n	o	r	s	t	u	y
0	True	False	False	False	True	False	False	False	False	False	True	True	False	False	True	False	False	False	False	False
1	False	False	False	True	False	False	False	True	False	True	False	False	True	True	False	False	False	True	False	False
2	False	False	False	False	False	False	True	True	False	False	True	False	False	True	False	False	False	True	True	True
3	False	False	True	False	False	False	False	True	False	True	False	False	False	False	False	False	False	True	False	False
4	False	False	False	False	False	True	False	False	True	True	True	False	False	False	False	True	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
388018	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
388019	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
388020	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
388021	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
388022	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

388023 rows × 20 columns