data_procesing

November 27, 2024

1 Data Processing

Data cleaning and feature engineering (excluding clustering which is done in individual model notebooks)

```
[ ]: import pandas as pd
     import numpy as np
     import sklearn as sk
     from datetime import date
     import ast
```

1.1 Read the dataset

```
[ ]: original_df = pd.read_csv('train.csv', parse_dates=['host_since',␣
      ↪'first_review', 'last_review'])
     print(original_df.columns)
```

```
Index(['name', 'description', 'property_type', 'price',
       'neighbourhood_cleansed', 'neighbourhood_group_cleansed', 'latitude',
       'longitude', 'host_since', 'host_response_time', 'host_response_rate',
       'host_acceptance_rate', 'host_is_superhost', 'host_listings_count',
       'host_total_listings_count', 'host_verifications',
       'host_has_profile_pic', 'host_identity_verified',
       'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'room_type',
       'accommodates', 'bathrooms', 'bathrooms_text', 'bedrooms', 'beds',
       'amenities', 'has_availability', 'availability_30', 'availability_60',
       'availability_90', 'availability_365', 'instant_bookable',
       'minimum_nights', 'maximum_nights', 'number_of_reviews',
       'number_of_reviews_ltm', 'number_of_reviews_l30d', 'first_review',
       'last_review', 'review_scores_rating', 'review_scores_accuracy',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'reviews_per_month', 'reviews'],
      dtype='object')
```

[ ]:
```
original_df['bathrooms_text']

#  Calculate the proportion of True values
# proportion_true = original_df['host_identity_verified'].mean()

#  Print the result
# print(f"The proportion of True values is: {proportion_true:.2%}")
```

[ ]:
```
0                2 baths
1         1 private bath
2                 1 bath
3              1.5 baths
4                 1 bath
              …
15691             1 bath
15692      1 shared bath
15693             1 bath
15694             1 bath
15695      1 shared bath
Name: bathrooms_text, Length: 15696, dtype: object
```

1.2 Remove Unnecessary and Redundant Features

All text-based descriptions and reviews will not be parsed. Neighborhood data is not necessary as this algorithm will utilize longitude and latitude for location data. host_idenity_verified will serve as the verification feature rather than parsing additional verifications One feature for the number of host listings is plenty sufficient.

[ ]:
```
df = original_df.drop(
    ["name", "description", "neighbourhood_cleansed",
     "neighbourhood_group_cleansed", "host_verifications",
     "calculated_host_listings_count_entire_homes",
     "calculated_host_listings_count_private_rooms",
     "calculated_host_listings_count_shared_rooms",
     "reviews"],
    axis=1
)
```

1.3 Resolve Two Bathroom Columns

The bathrooms text is used for more recent scrapes and typically has the word “baths” following a float value for the number of bathrooms at the property.

[ ]:
```
# Extract the numerical part from 'bathrooms_text'
df['extracted_bathrooms'] = (
    df['bathrooms_text']
    .str.extract(r'([\d\.]+)')  # Regex to capture numbers (including decimals)
    .astype(float)              # Convert to float
)

# Replace 'bathrooms' with the extracted value if it exists
df['bathrooms'] = df['extracted_bathrooms'].combine_first(df['bathrooms'])

# Drop the helper column if not needed
df.drop(columns=['extracted_bathrooms'], inplace=True)
df.drop(columns=['bathrooms_text'], inplace=True)

df['bathrooms']
```

[ ]:
```
0        2.0
1        1.0
2        1.0
3        1.5
4        1.0
        …
15691    1.0
15692    1.0
15693    1.0
15694    1.0
15695    1.0
Name: bathrooms, Length: 15696, dtype: float64
```

1.4 Change Response Time to a Numerical Scale

[ ]:
```
# Find unique values
response_times = df['host_response_time'].dropna().unique().tolist()
print("Response times:", response_times)

# 1 = 'within an hour', 2 = 'within a day', 3 = 'within a few hours', 4 = 'a␣
 ↪few days or more'
# Map string values to corresponding numerical values
response_time_mapping = {
    'within an hour': 1,
    'within a day': 2,
    'within a few hours': 3,
    'a few days or more': 4
}

# Replace the string values with numbers, leaving NaN values as is
df['host_response_time'] = df['host_response_time'].
 ↪replace(response_time_mapping)

df
```

```
Response times: ['within a day', 'within an hour', 'within a few hours', 'a few
days or more']
```

```
[ ]:                  property_type  price   latitude  longitude host_since  \
0             Entire rental unit      4  40.684560 -73.939870 2015-05-23
1      Private room in rental unit      3  40.638991 -73.965739 2023-09-14
2             Entire rental unit      3  40.618810 -74.032380 2022-07-31
3      Private room in rental unit      0  40.673970 -73.953990 2012-08-11
4                  Room in hotel      2  40.747180 -73.985390 2014-12-23
…                            …    …        …          …          …
15691         Room in aparthotel      5  40.704777 -74.006425 2021-08-27
15692       Private room in condo      0  40.881490 -73.910130 2018-07-22
15693              Room in hotel      5  40.765440 -73.976508 2018-03-06
15694         Entire rental unit      5  40.735635 -74.005740 2016-12-16
15695        Private room in home      1  40.628170 -74.079650 2020-07-15

       host_response_time  host_response_rate  host_acceptance_rate  \
0                     2.0               100.0                 100.0
1                     1.0               100.0                  98.0
2                     1.0               100.0                 100.0
3                     1.0                99.0                  23.0
4                     1.0                93.0                  95.0
…                     …                  …                    …
15691                 1.0                99.0                  99.0
15692                 1.0               100.0                  67.0
15693                 1.0               100.0                  98.0
15694                 1.0               100.0                  96.0
15695                 1.0               100.0                  83.0

      host_is_superhost  host_listings_count  … first_review  last_review  \
0                  True                  2.0  …   2019-04-28   2024-08-10
1                  True                  1.0  …   2024-01-13   2024-09-02
2                 False                 52.0  …   2024-06-27   2024-08-17
3                 False                727.0  …          NaT          NaT
4                 False                707.0  …          NaT          NaT
…                   …                  …  …          …           …
15691              True                 15.0  …   2023-01-15   2024-08-14
15692             False                  2.0  …   2023-04-16   2023-11-18
15693              True                 28.0  …   2023-08-21   2023-08-21
15694             False               4494.0  …          NaT          NaT
15695              True                  6.0  …   2020-10-30   2023-07-21

       review_scores_rating  review_scores_accuracy review_scores_cleanliness  \
0                      5.00                    5.00                       4.97
1                      4.83                    4.87                       4.93
2                      4.60                    4.80                       4.20
3                       NaN                     NaN                        NaN
4                       NaN                     NaN                        NaN
…                      …                     …                        …
15691                  4.94                    4.94                       4.91
```

```
15692                      4.33                     4.33                       4.17
15693                      5.00                     5.00                       5.00
15694                       NaN                      NaN                        NaN
15695                      4.81                     4.90                       4.95

       review_scores_checkin  review_scores_communication  \
0                       5.00                         5.00
1                       4.80                         4.90
2                       4.80                         4.80
3                        NaN                          NaN
4                        NaN                          NaN
…                      …                           …
15691                   4.97                         4.81
15692                   4.17                         4.33
15693                   5.00                         5.00
15694                    NaN                          NaN
15695                   4.86                         4.86

       review_scores_location  review_scores_value reviews_per_month
0                        4.71                 4.94              0.52
1                        4.90                 4.63              3.81
2                        4.80                 4.20              2.14
3                         NaN                  NaN               NaN
4                         NaN                  NaN               NaN
…                       …                   …                …
15691                    5.00                 4.72              1.60
15692                    4.00                 4.33              0.35
15693                    5.00                 5.00              0.08
15694                     NaN                  NaN               NaN
15695                    4.62                 4.81              0.45

[15696 rows x 41 columns]
```

1.5 Turn Binary Columns to 1s/0s

[ ]:
```
columns_to_convert = ['host_is_superhost', 'has_availability',␣
 ↪'host_has_profile_pic', 'host_identity_verified', ]
df[columns_to_convert] = df[columns_to_convert].replace({'True': 1, 'False': 0})

# Fill NaN values with 0
df[columns_to_convert] = df[columns_to_convert].fillna(0).astype(int)
```

1.6 Resolve NaT and NaN Occurrences

As a naive approach, NaN numerical features are replaced with the means for each feature. NaT dates are replaced with today's date. Missing binary values are replaced with false.

```
[ ]: # Find the columns with Na values
     na_columns = df.columns[df.isna().any()].tolist()
     print(f"Columns with NA Value: {na_columns}")

     # Replace missing numerical features with averages
     for col in df.select_dtypes(include=['number']).columns:
         if col in na_columns:
             df[col].fillna(df[col].mean(), inplace=True)

     # Replace missing dates with today's date (date only, not time)
     for col in ['host_since', 'first_review', 'last_review']:
         if col in na_columns:
             df[col] = pd.to_datetime(df[col], errors='coerce')  # Ensure proper␣
      ↪datetime parsing
             df[col].fillna(pd.Timestamp(date.today()), inplace=True)

     df['host_since'] = pd.to_datetime(df['host_since'])
     df['first_review'] = pd.to_datetime(df['first_review'])
     df['last_review'] = pd.to_datetime(df['last_review'])

     df
```

```
Columns with NA Value: ['host_response_time', 'host_response_rate',
'host_acceptance_rate', 'bathrooms', 'bedrooms', 'beds', 'first_review',
'last_review', 'review_scores_rating', 'review_scores_accuracy',
'review_scores_cleanliness', 'review_scores_checkin',
'review_scores_communication', 'review_scores_location', 'review_scores_value',
'reviews_per_month']
```

```
[ ]:                  property_type  price   latitude  longitude host_since  \
     0             Entire rental unit      4  40.684560 -73.939870 2015-05-23
     1      Private room in rental unit      3  40.638991 -73.965739 2023-09-14
     2             Entire rental unit      3  40.618810 -74.032380 2022-07-31
     3      Private room in rental unit      0  40.673970 -73.953990 2012-08-11
     4                  Room in hotel      2  40.747180 -73.985390 2014-12-23
     …                              …    …          …          …          …
     15691         Room in aparthotel      5  40.704777 -74.006425 2021-08-27
     15692        Private room in condo      0  40.881490 -73.910130 2018-07-22
     15693                Room in hotel      5  40.765440 -73.976508 2018-03-06
     15694           Entire rental unit      5  40.735635 -74.005740 2016-12-16
     15695         Private room in home      1  40.628170 -74.079650 2020-07-15

            host_response_time  host_response_rate  host_acceptance_rate  \
     0                     2.0               100.0                 100.0
     1                     1.0               100.0                  98.0
     2                     1.0               100.0                 100.0
     3                     1.0                99.0                  23.0
```

```
4                      1.0               93.0                 95.0
…                    …                 …                    …
15691                  1.0               99.0                 99.0
15692                  1.0              100.0                 67.0
15693                  1.0              100.0                 98.0
15694                  1.0              100.0                 96.0
15695                  1.0              100.0                 83.0

       host_is_superhost  host_listings_count  …  first_review  last_review  \
0                      1                  2.0  …    2019-04-28   2024-08-10
1                      1                  1.0  …    2024-01-13   2024-09-02
2                      0                 52.0  …    2024-06-27   2024-08-17
3                      0                727.0  …    2024-11-22   2024-11-22
4                      0                707.0  …    2024-11-22   2024-11-22
…                    …                  …  …           …            …
15691                  1                 15.0  …    2023-01-15   2024-08-14
15692                  0                  2.0  …    2023-04-16   2023-11-18
15693                  1                 28.0  …    2023-08-21   2023-08-21
15694                  0               4494.0  …    2024-11-22   2024-11-22
15695                  1                  6.0  …    2020-10-30   2023-07-21

       review_scores_rating  review_scores_accuracy review_scores_cleanliness  \
0                  5.000000                5.000000                  4.970000
1                  4.830000                4.870000                  4.930000
2                  4.600000                4.800000                  4.200000
3                  4.719393                4.742812                  4.679642
4                  4.719393                4.742812                  4.679642
…                       …                     …                         …
15691              4.940000                4.940000                  4.910000
15692              4.330000                4.330000                  4.170000
15693              5.000000                5.000000                  5.000000
15694              4.719393                4.742812                  4.679642
15695              4.810000                4.900000                  4.950000

       review_scores_checkin  review_scores_communication  \
0                    5.00000                     5.000000
1                    4.80000                     4.900000
2                    4.80000                     4.800000
3                    4.82631                     4.808233
4                    4.82631                     4.808233
…                        …                          …
15691                4.97000                     4.810000
15692                4.17000                     4.330000
15693                5.00000                     5.000000
15694                4.82631                     4.808233
15695                4.86000                     4.860000
```

```
       review_scores_location  review_scores_value reviews_per_month
0                    4.710000             4.940000          0.520000
1                    4.900000             4.630000          3.810000
2                    4.800000             4.200000          2.140000
3                    4.721844             4.609505          1.245801
4                    4.721844             4.609505          1.245801
…                         …                  …                …
15691                5.000000             4.720000          1.600000
15692                4.000000             4.330000          0.350000
15693                5.000000             5.000000          0.080000
15694                4.721844             4.609505          1.245801
15695                4.620000             4.810000          0.450000

[15696 rows x 41 columns]
```

[ ]:
```
# Check that all NA are gone
na_columns = df.columns[df.isna().any()].tolist()
print(f"Columns with NA Value: {na_columns}")
```

```
Columns with NA Value: []
```

1.7 Convert DateTime Features to Numeric

[ ]:
```
# Convert datetime columns to numeric (e.g., number of days since the reference␣
↪date)
df['host_since'] = (df['host_since'] - pd.Timestamp('1970-01-01')) // pd.
↪Timedelta('1D')
df['first_review'] = (df['first_review'] - pd.Timestamp('1970-01-01')) // pd.
↪Timedelta('1D')
df['last_review'] = (df['last_review'] - pd.Timestamp('1970-01-01')) // pd.
↪Timedelta('1D')
```

1.8 One-hot Encode room_type

[ ]:
```
# List all unique strings in 'property_type' and 'room_type' columns
property_types = df['property_type'].dropna().unique().tolist()
room_types = df['room_type'].dropna().unique().tolist()

print("Unique property types:", property_types)
print("Unique room types:", room_types)

# Upon investigation, property_type has too many categories and is largley␣
↪redundant
df.drop(columns=['property_type'], inplace=True)

# one-hot encoding
df = pd.get_dummies(df, columns=['room_type'], prefix='room', dummy_na=False)
```

```
df
```

```
Unique property types: ['Entire rental unit', 'Private room in rental unit',
'Room in hotel', 'Private room in home', 'Private room in condo', 'Shared room
in rental unit', 'Private room in serviced apartment', 'Entire guest suite',
'Entire townhouse', 'Private room in loft', 'Entire serviced apartment',
'Private room in townhouse', 'Entire guesthouse', 'Entire vacation home',
'Entire home', 'Entire place', 'Entire condo', 'Entire loft', 'Private room in
guest suite', 'Private room in bungalow', 'Room in boutique hotel', 'Room in
aparthotel', 'Private room in bed and breakfast', 'Entire bungalow', 'Private
room in casa particular', 'Shared room in guest suite', 'Private room in
resort', 'Entire villa', 'Private room in kezhan', 'Private room', 'Shared room
in home', 'Entire cottage', 'Private room in guesthouse', 'Camper/RV', 'Boat',
'Private room in hostel', 'Shared room in hostel', 'Houseboat', 'Shared room in
serviced apartment', 'Shared room in vacation home', 'Private room in ranch',
'Private room in camper/rv', 'Room in serviced apartment', 'Private room in
vacation home', 'Private room in religious building', 'Shared room in
townhouse', 'Private room in cottage', 'Shared room', 'Private room in tiny
home', 'Private room in villa', 'Private room in houseboat', 'Shared room in
guesthouse', 'Shared room in condo', 'Private room in tower', 'Shared room in
casa particular', 'Tiny home', 'Private room in earthen home', 'Private room in
castle', 'Casa particular']
Unique room types: ['Entire home/apt', 'Private room', 'Hotel room', 'Shared
room']
```

[ ]:
```
       price   latitude  longitude  host_since  host_response_time  \
0          4  40.684560 -73.939870       16578                 2.0
1          3  40.638991 -73.965739       19614                 1.0
2          3  40.618810 -74.032380       19204                 1.0
3          0  40.673970 -73.953990       15563                 1.0
4          2  40.747180 -73.985390       16427                 1.0
…      …        …          …         …                   …
15691      5  40.704777 -74.006425       18866                 1.0
15692      0  40.881490 -73.910130       17734                 1.0
15693      5  40.765440 -73.976508       17596                 1.0
15694      5  40.735635 -74.005740       17151                 1.0
15695      1  40.628170 -74.079650       18458                 1.0

       host_response_rate  host_acceptance_rate  host_is_superhost  \
0                   100.0                 100.0                  1
1                   100.0                  98.0                  1
2                   100.0                 100.0                  0
3                    99.0                  23.0                  0
4                    93.0                  95.0                  0
…                     …                     …                  …
15691                99.0                  99.0                  1
```

```
15692           100.0             67.0              0
15693           100.0             98.0              1
15694           100.0             96.0              0
15695           100.0             83.0              1

       host_listings_count  host_total_listings_count  …  \
0                      2.0                        2.0  …
1                      1.0                        1.0  …
2                     52.0                       55.0  …
3                    727.0                     1336.0  …
4                    707.0                     2453.0  …
…                      …                        …   …
15691                 15.0                       15.0  …
15692                  2.0                        2.0  …
15693                 28.0                       35.0  …
15694               4494.0                     4784.0  …
15695                  6.0                        9.0  …

       review_scores_cleanliness  review_scores_checkin  \
0                       4.970000                5.00000
1                       4.930000                4.80000
2                       4.200000                4.80000
3                       4.679642                4.82631
4                       4.679642                4.82631
…                            …                      …
15691                   4.910000                4.97000
15692                   4.170000                4.17000
15693                   5.000000                5.00000
15694                   4.679642                4.82631
15695                   4.950000                4.86000

       review_scores_communication  review_scores_location  \
0                         5.000000                4.710000
1                         4.900000                4.900000
2                         4.800000                4.800000
3                         4.808233                4.721844
4                         4.808233                4.721844
…                              …                       …
15691                     4.810000                5.000000
15692                     4.330000                4.000000
15693                     5.000000                5.000000
15694                     4.808233                4.721844
15695                     4.860000                4.620000

       review_scores_value  reviews_per_month  room_Entire home/apt  \
0                 4.940000           0.520000                     1
1                 4.630000           3.810000                     0
```

```
2                 4.200000          2.140000                     1
3                 4.609505          1.245801                     0
4                 4.609505          1.245801                     0
…                    …                 …                  …
15691             4.720000          1.600000                     1
15692             4.330000          0.350000                     0
15693             5.000000          0.080000                     0
15694             4.609505          1.245801                     1
15695             4.810000          0.450000                     0

      room_Hotel room  room_Private room  room_Shared room
0                   0                  0                 0
1                   0                  1                 0
2                   0                  0                 0
3                   0                  1                 0
4                   1                  0                 0
…                 …                  …                 …
15691               0                  0                 0
15692               0                  1                 0
15693               0                  1                 0
15694               0                  0                 0
15695               0                  1                 0

[15696 rows x 43 columns]
```

1.9 One-hot Encode Key Amenities

[ ]:
```python
# List of selected amenities to encode
selected_amenities = [
    "Air conditioning", "Kitchen", "Dedicated workspace", "Heating",
    "Hot water", "Refrigerator", "Free street parking", "Self check-in",
    "Shampoo", "Washer"
]

# Step 1: Ensure the amenities column is properly formatted as a list
df['amenities'] = df['amenities'].apply(lambda x: ast.literal_eval(x) if␣
 ↪isinstance(x, str) else x)

# Step 2: Expand the list of amenities into individual rows
df_expanded = df.explode('amenities').reset_index()

# Check the columns after explode to avoid mismatch
print(df_expanded.columns)

# Rename columns for clarity
df_expanded = df_expanded[['index', 'amenities']]
df_expanded.columns = ['property_index', 'amenity']
```

```
# Step 3: Handle potential inconsistencies in strings
df_expanded['amenity'] = df_expanded['amenity'].str.strip()

# Step 4: Create one-hot encoding for the expanded amenities
df_expanded_one_hot = pd.get_dummies(df_expanded['amenity'], prefix='',␣
 ↪prefix_sep='')

# Step 5: Group by property_index and aggregate by taking the max
df_expanded_one_hot = df_expanded_one_hot.
 ↪groupby(df_expanded['property_index']).max()

# Step 6: Ensure selected amenities are in the columns, add them if not present
for amenity in selected_amenities:
    if amenity not in df_expanded_one_hot.columns:
        df_expanded_one_hot[amenity] = 0

# Align the columns with the selected amenities order
df_expanded_one_hot = df_expanded_one_hot[selected_amenities]

# Step 7: Merge the one-hot encoded amenities into the original df
df = df.merge(df_expanded_one_hot, left_index=True, right_index=True,␣
 ↪how='left')

# Drop the amenities column
df = df.drop(['amenities'], axis=1)

# Display the updated df
df
```

```
Index(['index', 'price', 'latitude', 'longitude', 'host_since',
       'host_response_time', 'host_response_rate', 'host_acceptance_rate',
       'host_is_superhost', 'host_listings_count', 'host_total_listings_count',
       'host_has_profile_pic', 'host_identity_verified',
       'calculated_host_listings_count', 'accommodates', 'bathrooms',
       'bedrooms', 'beds', 'amenities', 'has_availability', 'availability_30',
       'availability_60', 'availability_90', 'availability_365',
       'instant_bookable', 'minimum_nights', 'maximum_nights',
       'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30d',
       'first_review', 'last_review', 'review_scores_rating',
       'review_scores_accuracy', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value', 'reviews_per_month',
       'room_Entire home/apt', 'room_Hotel room', 'room_Private room',
       'room_Shared room'],
      dtype='object')
```

```
[ ]:        price   latitude  longitude  host_since  host_response_time  \
    0          4  40.684560 -73.939870       16578                 2.0
    1          3  40.638991 -73.965739       19614                 1.0
    2          3  40.618810 -74.032380       19204                 1.0
    3          0  40.673970 -73.953990       15563                 1.0
    4          2  40.747180 -73.985390       16427                 1.0
    …        …        …          …         …                  …
    15691      5  40.704777 -74.006425       18866                 1.0
    15692      0  40.881490 -73.910130       17734                 1.0
    15693      5  40.765440 -73.976508       17596                 1.0
    15694      5  40.735635 -74.005740       17151                 1.0
    15695      1  40.628170 -74.079650       18458                 1.0

           host_response_rate  host_acceptance_rate  host_is_superhost  \
    0                   100.0                 100.0                  1
    1                   100.0                  98.0                  1
    2                   100.0                 100.0                  0
    3                    99.0                  23.0                  0
    4                    93.0                  95.0                  0
    …                     …                    …                 …
    15691                99.0                  99.0                  1
    15692               100.0                  67.0                  0
    15693               100.0                  98.0                  1
    15694               100.0                  96.0                  0
    15695               100.0                  83.0                  1

           host_listings_count  host_total_listings_count  …  Air conditioning  \
    0                      2.0                        2.0  …                 0
    1                      1.0                        1.0  …                 1
    2                     52.0                       55.0  …                 1
    3                    727.0                     1336.0  …                 0
    4                    707.0                     2453.0  …                 1
    …                      …                        …  …              …
    15691                 15.0                       15.0  …                 1
    15692                  2.0                        2.0  …                 0
    15693                 28.0                       35.0  …                 1
    15694               4494.0                     4784.0  …                 1
    15695                  6.0                        9.0  …                 1

           Kitchen  Dedicated workspace  Heating  Hot water  Refrigerator  \
    0            1                    0        1          1             1
    1            1                    0        0          1             0
    2            0                    1        1          1             0
    3            1                    1        1          1             1
    4            0                    1        1          1             0
    …         …                 …      …        …          …
    15691        0                    1        1          1             1
```

```
15692         1                    1        0          1             1
15693         1                    0        1          0             0
15694         1                    0        1          1             1
15695         1                    1        1          1             0

       Free street parking  Self check-in  Shampoo  Washer
0                        1              1        1       0
1                        1              1        1       0
2                        0              1        1       0
3                        0              0        0       1
4                        0              1        1       0
…                    …            …      …     …
15691                    0              1        1       0
15692                    1              1        1       0
15693                    0              1        1       0
15694                    0              1        1       0
15695                    0              1        0       0

[15696 rows x 52 columns]
```

1.10 Save as CSV

[ ]:
```
# Save the DataFrame as a CSV file
df.to_csv('train_processed.csv', index=False)
```