AdvDSI-A2-beer-data-prep

March 20, 2022

1 AdvDSI - Assignment 2: Multi-Class Classification - Beer Style Predictor - Data Preparation

Train a machine learning model (using sklearn) or a custom neural networks (using pytorch) that will

accurately predict a type of beer based on some users' rating criterias such as appearance, aroma, palate or taste.

You will also need to build a web app and deploy it online (using Heroku) in order to serve your model for real time predictions.

Student Name: Nathan Fragar

Student No. : 93087548

Week: 6

Date: 20MAR2022

1.1 1. Load and Discover Dataset

[**1.1**] Task: Import required packages: Pandas, Numpy, joblib etc

[2]:

```
# Task: Import the pandas, numpy and joblib package
import pandas as pd
import numpy as np
import joblib as job

# Scaler and Encoders
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
```

[64]:

```
# Change Working Directory: /home/jovyan/work
```

[3]:

```
cd /home/jovyan/work
```

```
/home/jovyan/work
```

[4]:

```
# Task: Launch the magic commands for auto-relaoding external modules
%load_ext autoreload
```

```
%autoreload 2
```

[**1.2**] Task: Load Dataset

[5]:
```
# file_url
file_path_beer_reviews = 'data/raw/beer_reviews.csv'

# Load files into df_raw data frames
df = pd.read_csv(file_path_beer_reviews)
```

[**1.3**] Discover Dataset

[6]:
```
df.head()
```

[6]:
```
   brewery_id             brewery_name  review_time  review_overall  \
0       10325          Vecchio Birraio   1234817823             1.5
1       10325          Vecchio Birraio   1235915097             3.0
2       10325          Vecchio Birraio   1235916604             3.0
3       10325          Vecchio Birraio   1234725145             3.0
4        1075  Caldera Brewing Company   1293735206             4.0

   review_aroma  review_appearance review_profilename  \
0           2.0                2.5            stcules
1           2.5                3.0            stcules
2           2.5                3.0            stcules
3           3.0                3.5            stcules
4           4.5                4.0     johnmichaelsen

                       beer_style  review_palate  review_taste  \
0                      Hefeweizen            1.5           1.5
1              English Strong Ale            3.0           3.0
2          Foreign / Export Stout            3.0           3.0
3                 German Pilsener            2.5           3.0
4  American Double / Imperial IPA            4.0           4.5

                 beer_name  beer_abv  beer_beerid
0             Sausa Weizen       5.0        47986
1                 Red Moon       6.2        48213
2  Black Horse Black Beer       6.5        48215
3               Sausa Pils       5.0        47969
4            Cauldron DIPA       7.7        64883
```

[7]:
```
# Shape of df
df.shape
```

[7]:
```
(1586614, 13)
```

Observation

There are 1,586,614 records in total and the following fields have missing rows * brewery_name (1,586,599 records - 15 records missing - 0.0001% - categorical) * review_profilename (1,586,266 records - 348 records missing - 0.02% - categorical) * beer_abv (1,518,829 records - 67,785 records missing - 4.27% - numerical)

[8]:
```
df.describe()
```

[8]:
```
         brewery_id   review_time  review_overall  review_aroma  \
count  1.586614e+06  1.586614e+06    1.586614e+06  1.586614e+06
mean   3.130099e+03  1.224089e+09    3.815581e+00  3.735636e+00
std    5.578104e+03  7.654427e+07    7.206219e-01  6.976167e-01
min    1.000000e+00  8.406720e+08    0.000000e+00  1.000000e+00
25%    1.430000e+02  1.173224e+09    3.500000e+00  3.500000e+00
50%    4.290000e+02  1.239203e+09    4.000000e+00  4.000000e+00
75%    2.372000e+03  1.288568e+09    4.500000e+00  4.000000e+00
max    2.800300e+04  1.326285e+09    5.000000e+00  5.000000e+00

       review_appearance  review_palate  review_taste      beer_abv  \
count       1.586614e+06   1.586614e+06  1.586614e+06  1.518829e+06
mean        3.841642e+00   3.743701e+00  3.792860e+00  7.042387e+00
std         6.160928e-01   6.822184e-01  7.319696e-01  2.322526e+00
min         0.000000e+00   1.000000e+00  1.000000e+00  1.000000e-02
25%         3.500000e+00   3.500000e+00  3.500000e+00  5.200000e+00
50%         4.000000e+00   4.000000e+00  4.000000e+00  6.500000e+00
75%         4.000000e+00   4.000000e+00  4.500000e+00  8.500000e+00
max         5.000000e+00   5.000000e+00  5.000000e+00  5.770000e+01

        beer_beerid
count  1.586614e+06
mean   2.171279e+04
std    2.181834e+04
min    3.000000e+00
25%    1.717000e+03
50%    1.390600e+04
75%    3.944100e+04
max    7.731700e+04
```

[10]:
```
df.sort_values(by='beer_abv', ascending=False).head()
```

[10]:
```
        brewery_id brewery_name  review_time  review_overall  review_aroma  \
12919         6513  Schorschbräu   1316780901             4.0           4.0
12939         6513  Schorschbräu   1309974178             4.0           4.0
12940         6513  Schorschbräu   1274469798             3.5           4.0
746385       16315      BrewDog   1285808609             3.5           4.0
746387       16315      BrewDog   1285274059             3.0           3.0

        review_appearance review_profilename                      beer_style  \
12919                 4.0        kappldav123                        Eisbock
12939                 3.5           Sunnanek                        Eisbock
12940                 4.0        kappldav123                        Eisbock
746385                4.0              bobsy  American Double / Imperial IPA
746387                3.0             cratez  American Double / Imperial IPA

        review_palate  review_taste                      beer_name  beer_abv  \
12919             4.0           3.5  Schorschbräu Schorschbock 57%      57.7
12939             4.0           4.0  Schorschbräu Schorschbock 43%      43.0
12940             4.0           4.5  Schorschbräu Schorschbock 43%      43.0
746385            4.0           4.0             Sink The Bismarck!      41.0
746387            3.0           3.5             Sink The Bismarck!      41.0

        beer_beerid
12919         73368
12939         57856
12940         57856
746385        57015
746387        57015
```

Observation * beer_abv has a maximum value of 57.7 * all other ratings have a maximum of 5

[**1.4**] Task: Create a for loop that will iterate through each columns and print their list of unique values

[60]:
```python
# Task: Create a list call cat_cols that contains
cat_cols = ['brewery_name','review_profilename','beer_style','beer_name']
```

[61]:
```python
# Create Data Frame df_cat_cols with categorical columns
df_cat_cols =  pd.DataFrame(df, columns=cat_cols, index=None)
```

[62]:
```python
df_cat_cols.head()
```

[62]:
```
              brewery_name review_profilename                      beer_style  \
0          Vecchio Birraio            stcules                      Hefeweizen
1          Vecchio Birraio            stcules              English Strong Ale
2          Vecchio Birraio            stcules          Foreign / Export Stout
3          Vecchio Birraio            stcules                 German Pilsener
4  Caldera Brewing Company     johnmichaelsen  American Double / Imperial IPA

                 beer_name
0             Sausa Weizen
1                 Red Moon
2   Black Horse Black Beer
3               Sausa Pils
4            Cauldron DIPA
```

[63]:
```
# Task: Create a for loop that will iterate through each columns and print␣
 ↪their name and list of unique values
for col in df_cat_cols.columns:
  print(col)
  print(df_cat_cols[col].unique())
  # print(df_cat_cols[col].value_counts())
```

```
brewery_name
['Vecchio Birraio' 'Caldera Brewing Company' 'Amstel Brouwerij B. V.' …
 'Wissey Valley Brewery' 'Outback Brewery Pty Ltd'
 'Georg Meinel Bierbrauerei KG']
review_profilename
['stcules' 'johnmichaelsen' 'oline73' … 'hogshead' 'NyackNicky'
 'joeebbs']
beer_style
['Hefeweizen' 'English Strong Ale' 'Foreign / Export Stout'
 'German Pilsener' 'American Double / Imperial IPA' 'Herbed / Spiced Beer'
 'Light Lager' 'Oatmeal Stout' 'American Pale Lager' 'Rauchbier'
 'American Pale Ale (APA)' 'American Porter' 'Belgian Strong Dark Ale'
 'American IPA' 'American Stout' 'Russian Imperial Stout'
 'American Amber / Red Ale' 'American Strong Ale' 'Märzen / Oktoberfest'
 'American Adjunct Lager' 'American Blonde Ale' 'Euro Pale Lager'
 'English Brown Ale' 'Scotch Ale / Wee Heavy' 'Fruit / Vegetable Beer'
 'American Double / Imperial Stout' 'Belgian Pale Ale' 'English Bitter'
 'English Porter' 'Irish Dry Stout' 'American Barleywine'
 'Belgian Strong Pale Ale' 'Doppelbock' 'Maibock / Helles Bock'
 'Pumpkin Ale' 'Dortmunder / Export Lager' 'Euro Strong Lager'
 'Euro Dark Lager' 'Low Alcohol Beer' 'Weizenbock'
 'Extra Special / Strong Bitter (ESB)' 'Bock'
 'English India Pale Ale (IPA)' 'Altbier' 'Kölsch' 'Munich Dunkel Lager'
 'Rye Beer' 'American Pale Wheat Ale' 'Milk / Sweet Stout' 'Schwarzbier'
 'Vienna Lager' 'American Amber / Red Lager' 'Scottish Ale' 'Witbier'
 'American Black Ale' 'Saison / Farmhouse Ale' 'English Barleywine'
 'English Dark Mild Ale' 'California Common / Steam Beer' 'Czech Pilsener'
 'English Pale Ale' 'Belgian IPA' 'Tripel' 'Flanders Oud Bruin'
 'American Brown Ale' 'Winter Warmer' 'Smoked Beer' 'Dubbel'
 'Flanders Red Ale' 'Dunkelweizen' 'Roggenbier'
 'Keller Bier / Zwickel Bier' 'Belgian Dark Ale' 'Bière de Garde'
 'Japanese Rice Lager' 'Black & Tan' 'Irish Red Ale' 'Chile Beer'
 'English Stout' 'Cream Ale' 'American Wild Ale'
 'American Double / Imperial Pilsner'
 'Scottish Gruit / Ancient Herbed Ale' 'Wheatwine'
 'American Dark Wheat Ale' 'American Malt Liquor' 'Baltic Porter'
 'Munich Helles Lager' 'Kristalweizen' 'English Pale Mild Ale'
 'Lambic - Fruit' 'Old Ale' 'Quadrupel (Quad)' 'Braggot'
 'Lambic - Unblended' 'Eisbock' 'Berliner Weissbier' 'Kvass' 'Faro'
 'Gueuze' 'Gose' 'Happoshu' 'Sahti' 'Bière de Champagne / Bière Brut']
```

```
beer_name
['Sausa Weizen' 'Red Moon' 'Black Horse Black Beer' … 'Baron Von Weizen'
 'Resolution #2' "The Horseman's Ale"]
```

[76]:
```
# Number of Brewery Values
df['brewery_name'].value_counts()
```

[76]:
```
Boston Beer Company (Samuel Adams)    39444
Dogfish Head Brewery                  33839
Stone Brewing Co.                     33066
Sierra Nevada Brewing Co.             28751
Bell's Brewery, Inc.                  25191
                                      …
Brauerei Stolz GmbH & Co. KG              1
Hausbrauerei Düll                         1
Browar Grybów                             1
Staro&#269;eský Pivovárek Dobru ka       1
Spire Brewery                             1
Name: brewery_name, Length: 5742, dtype: int64
```

[77]:
```
# Number of Brewery Values
df['beer_style'].value_counts()
```

[77]:
```
American IPA                        117586
American Double / Imperial IPA       85977
American Pale Ale (APA)              63469
Russian Imperial Stout               54129
American Double / Imperial Stout     50705
                                     …
Gose                                   686
Faro                                   609
Roggenbier                             466
Kvass                                  297
Happoshu                               241
Name: beer_style, Length: 104, dtype: int64
```

[78]:
```
# Check Duplicates
dup = df.duplicated()
df[dup]
```

[78]:
```
Empty DataFrame
Columns: [brewery_id, brewery_name, review_time, review_overall, review_aroma,
review_appearance, review_profilename, beer_style, review_palate, review_taste,
beer_name, beer_abv, beer_beerid]
Index: []
```

[79]:
```
dup.head()
```

[79]:
```
0    False
1    False
2    False
3    False
4    False
dtype: bool
```

Observation: No Duplicates found

1.1.1 3. Prepare Data

Data preparation

Missing Values * beer_abv - mean value * brewery_name - Mode Value - 'Boston Beer Company (Samuel Adams)' * review_profilename - Mode Value - 'northyorksammy'

Label Encoding * beer_style_cat - Target - Label encode beer_style

Scaling * Use MinMax Scaling for 'beer_abv','review_aroma','review_appearance','review_palate','review_taste' *Reference Data* * Beer Style - Used to map label beer_style_cat to beer_style text * Breweries - Used to map label brewery_id to brewery_name (and vice versa)

[257]:
```python
# Task: Create a copy of df and save it into a variable called df_cleaned
df_cleaned = df.copy()
```

[258]:
```python
# Numerical Fields - fill beer_abv with mean value for column
df_cleaned['beer_abv'].fillna(df_cleaned['beer_abv'].mean(), inplace=True)
```

[259]:
```python
# Categorical Field - brewery_name - fill beer_abv with mode value for column -␣
 ↪Boston Beer Company (Samuel Adams)
df_cleaned['brewery_name'] = df_cleaned['brewery_name'].fillna('Boston Beer␣
 ↪Company (Samuel Adams)')
```

[260]:
```python
df_cleaned[df_cleaned['brewery_name'].isna()]
```

[260]:
```
Empty DataFrame
Columns: [brewery_id, brewery_name, review_time, review_overall, review_aroma,
review_appearance, review_profilename, beer_style, review_palate, review_taste,
beer_name, beer_abv, beer_beerid]
Index: []
```

[261]:
```python
# Task: Find mode for review_profilename column
df_review_profilename_mode = df_cleaned['review_profilename'].mode()
print(df_review_profilename_mode)
```

```
0    northyorksammy
Name: review_profilename, dtype: object
```

[262]:

```
# Categorical Field - review_profilename - fill beer_abv with mode value for␣
 ↪column - northyorksammy
df_cleaned['review_profilename'] = df_cleaned['review_profilename'].
 ↪fillna('northyorksammy')
```

[263]:

```
df_cleaned[df_cleaned['review_profilename'].isna()]
```

[263]:

```
Empty DataFrame
Columns: [brewery_id, brewery_name, review_time, review_overall, review_aroma,
review_appearance, review_profilename, beer_style, review_palate, review_taste,
beer_name, beer_abv, beer_beerid]
Index: []
```

[234]:

```
df_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1586614 entries, 0 to 1586613
Data columns (total 13 columns):
 #   Column              Non-Null Count    Dtype
---  ------              --------------    -----
 0   brewery_id          1586614 non-null  int64
 1   brewery_name        1586614 non-null  object
 2   review_time         1586614 non-null  int64
 3   review_overall      1586614 non-null  float64
 4   review_aroma        1586614 non-null  float64
 5   review_appearance   1586614 non-null  float64
 6   review_profilename  1586614 non-null  object
 7   beer_style          1586614 non-null  object
 8   review_palate       1586614 non-null  float64
 9   review_taste        1586614 non-null  float64
 10  beer_name           1586614 non-null  object
 11  beer_abv            1586614 non-null  float64
 12  beer_beerid         1586614 non-null  int64
dtypes: float64(6), int64(3), object(4)
memory usage: 157.4+ MB
```

[264]:

```
df_cleaned.head()
```

[264]:

```
   brewery_id             brewery_name  review_time  review_overall  \
0       10325          Vecchio Birraio   1234817823             1.5
1       10325          Vecchio Birraio   1235915097             3.0
2       10325          Vecchio Birraio   1235916604             3.0
3       10325          Vecchio Birraio   1234725145             3.0
4        1075  Caldera Brewing Company   1293735206             4.0

   review_aroma  review_appearance review_profilename  \
0           2.0                2.5            stcules
```

```
1           2.5                3.0           stcules
2           2.5                3.0           stcules
3           3.0                3.5           stcules
4           4.5                4.0    johnmichaelsen

                       beer_style  review_palate  review_taste  \
0                      Hefeweizen            1.5           1.5
1              English Strong Ale            3.0           3.0
2          Foreign / Export Stout            3.0           3.0
3                 German Pilsener            2.5           3.0
4  American Double / Imperial IPA            4.0           4.5

                beer_name  beer_abv  beer_beerid
0            Sausa Weizen       5.0        47986
1                Red Moon       6.2        48213
2  Black Horse Black Beer       6.5        48215
3              Sausa Pils       5.0        47969
4           Cauldron DIPA       7.7        64883
```

[265]:

```
# Import label encoder
from sklearn import preprocessing
# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
```

[266]:

```
# Encode labels in column 'beer_style'.
df_cleaned['beer_style_cat']= label_encoder.
 ↪fit_transform(df_cleaned['beer_style'])
```

[267]:

```
df_cleaned.head()
```

[267]:

```
   brewery_id               brewery_name  review_time  review_overall  \
0       10325            Vecchio Birraio   1234817823             1.5
1       10325            Vecchio Birraio   1235915097             3.0
2       10325            Vecchio Birraio   1235916604             3.0
3       10325            Vecchio Birraio   1234725145             3.0
4        1075  Caldera Brewing Company   1293735206             4.0

   review_aroma  review_appearance review_profilename  \
0           2.0                2.5           stcules
1           2.5                3.0           stcules
2           2.5                3.0           stcules
3           3.0                3.5           stcules
4           4.5                4.0    johnmichaelsen

                       beer_style  review_palate  review_taste  \
0                      Hefeweizen            1.5           1.5
1              English Strong Ale            3.0           3.0
```

```
2          Foreign / Export Stout            3.0           3.0
3                 German Pilsener            2.5           3.0
4  American Double / Imperial IPA            4.0           4.5

                beer_name  beer_abv  beer_beerid  beer_style_cat
0            Sausa Weizen       5.0        47986              65
1                Red Moon       6.2        48213              51
2  Black Horse Black Beer       6.5        48215              59
3              Sausa Pils       5.0        47969              61
4           Cauldron DIPA       7.7        64883               9
```

[268]:

```
# Reindex with target field at end
new_col_order =␣
 ↪['brewery_name','brewery_id','beer_abv','review_aroma','review_appearance','review_palate',
df_cleaned = df_cleaned.reindex(columns=new_col_order)
```

[269]:

```
df_cleaned.head()
```

[269]:

```
             brewery_name  brewery_id  beer_abv  review_aroma  \
0          Vecchio Birraio       10325       5.0           2.0
1          Vecchio Birraio       10325       6.2           2.5
2          Vecchio Birraio       10325       6.5           2.5
3          Vecchio Birraio       10325       5.0           3.0
4  Caldera Brewing Company        1075       7.7           4.5

   review_appearance  review_palate  review_taste  \
0                2.5            1.5           1.5
1                3.0            3.0           3.0
2                3.0            3.0           3.0
3                3.5            2.5           3.0
4                4.0            4.0           4.5

                       beer_style  beer_style_cat
0                      Hefeweizen              65
1              English Strong Ale              51
2          Foreign / Export Stout              59
3                 German Pilsener              61
4  American Double / Imperial IPA               9
```

[270]:

```
# Save to CSV - Full Dataset
df_cleaned.to_csv('data/interim/beer_reviews_full.csv', index=False)
```

[79]:

```
df_cleaned.head()
```

[79]:

```
   brewery_id           brewery_name  review_time  review_overall  \
0       10325        Vecchio Birraio   1234817823             1.5
1       10325        Vecchio Birraio   1235915097             3.0
```

```
2       10325           Vecchio Birraio   1235916604             3.0
3       10325           Vecchio Birraio   1234725145             3.0
4        1075   Caldera Brewing Company   1293735206             4.0

   review_aroma  review_appearance review_profilename  \
0           2.0                2.5            stcules
1           2.5                3.0            stcules
2           2.5                3.0            stcules
3           3.0                3.5            stcules
4           4.5                4.0     johnmichaelsen

                       beer_style  review_palate  review_taste  \
0                      Hefeweizen            1.5           1.5
1              English Strong Ale            3.0           3.0
2          Foreign / Export Stout            3.0           3.0
3                 German Pilsener            2.5           3.0
4  American Double / Imperial IPA            4.0           4.5

                beer_name  beer_abv  beer_beerid  brewery_name_cat
0            Sausa Weizen       5.0        47986              5438
1                Red Moon       6.2        48213              5438
2  Black Horse Black Beer       6.5        48215              5438
3              Sausa Pils       5.0        47969              5438
4           Cauldron DIPA       7.7        64883              1480
```

[119]:
```
# Reindex with target field at end
new_col_order =␣
 ↪['brewery_id','brewery_name','brewery_name_cat','beer_abv','review_palate','review_aroma','
df_cleaned = df_cleaned.reindex(columns=new_col_order)
```

[120]:
```
df_cleaned.head()
```

[120]:
```
   brewery_id             brewery_name  brewery_name_cat  beer_abv  \
0       10325          Vecchio Birraio               NaN       5.0
1       10325          Vecchio Birraio               NaN       6.2
2       10325          Vecchio Birraio               NaN       6.5
3       10325          Vecchio Birraio               NaN       5.0
4        1075  Caldera Brewing Company               NaN       7.7

   review_palate  review_aroma  review_appearance  review_taste  \
0            1.5           2.0                2.5           1.5
1            3.0           2.5                3.0           3.0
2            3.0           2.5                3.0           3.0
3            2.5           3.0                3.5           3.0
4            4.0           4.5                4.0           4.5

                       beer_style                beer_name  beer_beerid  \
0                      Hefeweizen             Sausa Weizen        47986
1              English Strong Ale                 Red Moon        48213
2          Foreign / Export Stout  Black Horse Black Beer        48215
3                 German Pilsener               Sausa Pils        47969
4  American Double / Imperial IPA            Cauldron DIPA        64883

   review_time  review_overall review_profilename
0   1234817823             1.5            stcules
1   1235915097             3.0            stcules
2   1235916604             3.0            stcules
3   1234725145             3.0            stcules
4   1293735206             4.0     johnmichaelsen
```

[122]:

```
cols_drop =␣
 ↪['beer_name','beer_beerid','review_time','review_overall','review_profilename']
df_cleaned_2 = df_cleaned.drop(cols_drop, axis=1)
```

[123]:

```
df_cleaned_2.head()
```

[123]:

```
   brewery_id             brewery_name  brewery_name_cat  beer_abv  \
0       10325          Vecchio Birraio               NaN       5.0
1       10325          Vecchio Birraio               NaN       6.2
2       10325          Vecchio Birraio               NaN       6.5
3       10325          Vecchio Birraio               NaN       5.0
4        1075  Caldera Brewing Company               NaN       7.7

   review_palate  review_aroma  review_appearance  review_taste  \
0            1.5           2.0                2.5           1.5
1            3.0           2.5                3.0           3.0
2            3.0           2.5                3.0           3.0
3            2.5           3.0                3.5           3.0
4            4.0           4.5                4.0           4.5

                       beer_style
0                      Hefeweizen
1              English Strong Ale
2          Foreign / Export Stout
3                 German Pilsener
4  American Double / Imperial IPA
```

[97]:

```
df_cleaned.to_csv('data/interim/beer_reviews_full.csv', index=False)
```

[98]:

```
df_cleaned_2.to_csv('data/interim/beer_reviews_full_primary_cols.csv',␣
 ↪index=False)
```

[92]:

```
# df_cleaned = pd.read_csv('data/interim/beer_reviews_full.csv')
```

Scale Data Simple Scaler

```
[271]: # Task: Import Standard Staler and instantate as sc
       from sklearn.preprocessing import MinMaxScaler

       mms = MinMaxScaler()
```

```
[272]: num_cols =␣
         ↪['beer_abv','review_aroma','review_appearance','review_palate','review_taste']
```

```
[273]: df_cleaned[num_cols] = mms.fit_transform(df_cleaned[num_cols])
```

```
[274]: df_cleaned[num_cols].head()
```

```
[274]:    beer_abv  review_aroma  review_appearance  review_palate  review_taste
       0  0.086497         0.250                0.5          0.125         0.125
       1  0.107298         0.375                0.6          0.500         0.500
       2  0.112498         0.375                0.6          0.500         0.500
       3  0.086497         0.500                0.7          0.375         0.500
       4  0.133299         0.875                0.8          0.750         0.875
```

```
[275]: df_cleaned.head()
```

```
[275]:              brewery_name  brewery_id  beer_abv  review_aroma  \
       0          Vecchio Birraio       10325  0.086497         0.250
       1          Vecchio Birraio       10325  0.107298         0.375
       2          Vecchio Birraio       10325  0.112498         0.375
       3          Vecchio Birraio       10325  0.086497         0.500
       4  Caldera Brewing Company        1075  0.133299         0.875

          review_appearance  review_palate  review_taste  \
       0                0.5          0.125         0.125
       1                0.6          0.500         0.500
       2                0.6          0.500         0.500
       3                0.7          0.375         0.500
       4                0.8          0.750         0.875

                               beer_style  beer_style_cat
       0                       Hefeweizen              65
       1               English Strong Ale              51
       2           Foreign / Export Stout              59
       3                  German Pilsener              61
       4  American Double / Imperial IPA               9
```

1.1.2 Create List of Beer Styles

```
[277]: df_beer_style = df_cleaned.groupby(['beer_style','beer_style_cat']).size().
         ↪reset_index().rename(columns={0:'count'})
       df_beer_style = df_beer_style.drop(['count'], axis=1)
       df_beer_style.sort_values(by='beer_style', ascending=True).head(10)
```

```
[277]:                       beer_style  beer_style_cat
       0                         Altbier               0
       1          American Adjunct Lager               1
       2        American Amber / Red Ale               2
       3      American Amber / Red Lager               3
       4             American Barleywine               4
       5              American Black Ale               5
       6             American Blonde Ale               6
       7              American Brown Ale               7
       8         American Dark Wheat Ale               8
       9  American Double / Imperial IPA               9
```

```
[279]: # Save to CSV - Full Dataset - Scaled
       df_beer_style.to_csv('data/processed/beer_style.csv', index=False)
```

1.1.3 Create a list of Breweries

```
[ ]: df_brewery = df_cleaned_prep.groupby(['brewery_name','brewery_id']).size().
       ↪reset_index().rename(columns={0:'count'})
```

```
[ ]: df_brewery = df_brewery.drop(['count'], axis=1)
```

```
[ ]: df_brewery.sort_values(by='brewery_name', ascending=True).head(10)
```

```
[ ]:                      brewery_name  brewery_id
     0               't Hofbrouwerijke       13160
     1           (512) Brewing Company       17863
     2           10 Barrel Brewing Co.       16873
     3            1516 Brewing Company        4473
     4         16 Mile Brewing Company       20688
     5        1648 Brewing Company Ltd        8396
     6  1702 / The Address Brewing Co.       17783
     7             192 Brewing Company       22972
     8      1st City Brewery and Grill        4437
     9              2 Brothers Brewery       16847
```

```
[215]: # Save to CSV - Dataset = Remaining - 90%
       df_brewery.to_csv('data/processed/breweries.csv', index=False)
```

[276]:

```
# Save to CSV - Full Dataset - Scaled
df_cleaned.to_csv('data/interim/beer_reviews_full_scaled.csv', index=False)
```

1.1.4 Prepare data for split

[280]:

```
df_cleaned_prep = df_cleaned
```

[281]:

```
# Drop fields that are not going to be used - brewery_id, review_time,␣
 ↪review_overall, review_profilename, beer_name, beer_beerid
df_cleaned_prep.drop(['brewery_name','beer_style'], axis=1, inplace=True)
```

[289]:

```
df_cleaned_prep.head()
```

[289]:

```
   brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
0       10325  0.086497         0.250                0.5          0.125
1       10325  0.107298         0.375                0.6          0.500
2       10325  0.112498         0.375                0.6          0.500
3       10325  0.086497         0.500                0.7          0.375
4        1075  0.133299         0.875                0.8          0.750

   review_taste  beer_style_cat
0         0.125              65
1         0.500              51
2         0.500              59
3         0.500              61
4         0.875               9
```

[283]:

```
# Save to CSV - Dataset = Pre Split
df_cleaned_prep.to_csv('data/interim/beer_reviews_pre_split.csv', index=False)
```

[ ]:

```
# Load files
# df_cleaned_prep = pd.read_csv('data/interim/beer_reviews_pre_split.csv')
```

[284]:

```
df_cleaned_prep.head()
```

[284]:

```
   brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
0       10325  0.086497         0.250                0.5          0.125
1       10325  0.107298         0.375                0.6          0.500
2       10325  0.112498         0.375                0.6          0.500
3       10325  0.086497         0.500                0.7          0.375
4        1075  0.133299         0.875                0.8          0.750

   review_taste  beer_style_cat
0         0.125              65
1         0.500              51
2         0.500              59
3         0.500              61
```

```
4          0.875                    9
```

Split Data

Split data into Train, Validate and Test. * Split data into 10% of total dataset (using stratfy = True) to reduce the dataset * Stratefy and use a split of 0.2 for final dataset

[290]:

```
from src.data.sets import split_sets_random
```

[291]:

```
X_train_cleaned, y_train_cleaned, X_val_cleaned, y_val_cleaned, X_test_cleaned,␣
  ↪y_test_cleaned, X_remaining_cleaned, y_remaining_cleaned =␣
  ↪split_sets_random(df = df_cleaned_prep, target_col = 'beer_style_cat',␣
  ↪to_numpy=False, test_ratio=0.2, stratify_dataset= 'Yes',␣
  ↪reduce_dataset=True, reduce_ratio=0.1 )
```

[292]:

```
X_remaining_cleaned.head()
```

[292]:

```
         brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
708938         1471  0.129832         0.625                0.8           0.50
1390396        1525  0.121900         0.750                0.7           0.50
1091788         337  0.084763         0.375                0.4           0.50
422279           73  0.100364         0.750                0.9           1.00
870776          147  0.164500         0.750                0.8           0.75

         review_taste
708938          0.750
1390396         0.750
1091788         0.500
422279          0.875
870776          0.875
```

[293]:

```
y_remaining_cleaned.value_counts(normalize=True)
```

[293]:

```
12     0.074111
9      0.054189
14     0.040003
89     0.034116
11     0.031958
         …
62     0.000432
56     0.000384
88     0.000293
72     0.000187
64     0.000152
Name: beer_style_cat, Length: 104, dtype: float64
```

[294]:

```
y_train_cleaned.value_counts(normalize=True)
```

```
[294]: 12    0.074110
       9     0.054183
       14    0.040002
       89    0.034109
       11    0.031966
               …
       62    0.000431
       56    0.000389
       88    0.000305
       72    0.000189
       64    0.000147
       Name: beer_style_cat, Length: 104, dtype: float64
```

```
[295]: y_val_cleaned.value_counts(normalize=True)
```

```
[295]: 12    0.074118
       9     0.054202
       14    0.040021
       89    0.034129
       11    0.031954
               …
       48    0.000441
       56    0.000378
       88    0.000284
       72    0.000189
       64    0.000158
       Name: beer_style_cat, Length: 104, dtype: float64
```

```
[296]: y_test_cleaned.value_counts(normalize=True)
```

```
[296]: 12    0.074118
       9     0.054202
       14    0.039990
       89    0.034129
       11    0.031954
               …
       48    0.000441
       56    0.000378
       88    0.000284
       72    0.000189
       64    0.000158
       Name: beer_style_cat, Length: 104, dtype: float64
```

Prepare Data - Standard

```
[297]: X_test_cleaned.head()
```

[297]:
```
          brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
1511694          323  0.138499         1.000                0.8          0.875
45638              1  0.076096         0.625                0.7          0.625
282074            35  0.084763         0.625                0.7          0.750
1578181        22511  0.121165         0.750                0.8          0.750
129557            31  0.089964         0.750                0.7          0.750

         review_taste
1511694         0.875
45638           0.750
282074          0.750
1578181         0.875
129557          0.750
```

[298]:
```
save_sets(X_train=X_train_cleaned, y_train=y_train_cleaned,␣
 ↪X_val=X_val_cleaned, y_val=y_val_cleaned, X_test=X_test_cleaned,␣
 ↪y_test=y_test_cleaned, path='data/processed/standard_10/')
```

[120]:
```
# chunks = pd.read_csv('data/interim/beer_reviews_full.csv', chunksize =␣
 ↪100000, iterator = False )
# for n, chunk in enumerate(chunks):
#     chunk.to_csv(f'data/interim/beer_reviews_full_{n}.csv', index=False,␣
 ↪header=True)
```

[299]:
```
df_clean_remaining = pd.concat([X_remaining_cleaned, y_remaining_cleaned],␣
 ↪axis=1)
```

[300]:
```
df_clean_remaining.tail()
```

[300]:
```
         brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
720283          689  0.115965         0.625                0.8          0.625
495274          132  0.103831         0.750                0.9          0.750
948133          287  0.086497         0.625                0.8          0.625
715982          811  0.100364         0.750                0.8          0.875
178749        24453  0.121165         0.750                0.8          0.750

         review_taste  beer_style_cat
720283          0.750              12
495274          0.750               2
948133          0.625             103
715982          0.750               2
178749          0.875               2
```

[301]:
```
# Save to CSV - Dataset = Remaining - 90%
df_cleaned_prep.to_csv('data/interim/beer_reviews_remaining_post_split.csv',␣
 ↪index=False)
```

```
[302]: df_train_cleaned = pd.concat([X_train_cleaned, y_train_cleaned], axis=1)
```

```
[303]: df_train_cleaned.head()
```

```
[303]:          brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
       209458          192  0.121165         0.750                0.9          0.875
       470022          392  0.096897         0.625                0.8          0.750
       715177          811  0.096897         0.750                0.7          0.750
       1390594        2147  0.077830         0.500                0.7          0.625
       189154         2743  0.121165         0.750                0.8          0.750

                review_taste  beer_style_cat
       209458          0.875              12
       470022          0.750             102
       715177          0.750              18
       1390594         0.625              38
       189154          0.625               2
```

```
[304]: df_val_cleaned = pd.concat([X_val_cleaned, y_val_cleaned], axis=1)
```

```
[305]: df_test_cleaned = pd.concat([X_test_cleaned, y_test_cleaned], axis=1)
```

```
[306]: df_train_cleaned.shape
```

```
[306]: (95196, 7)
```

```
[307]: df_val_cleaned.shape
```

```
[307]: (31733, 7)
```

```
[308]: df_train_val = pd.concat([df_train_cleaned, df_val_cleaned], axis=0)
```

```
[309]: df_train_val.shape
```

```
[309]: (126929, 7)
```

```
[310]: df_train_val.tail()
```

```
[310]:          brewery_id  beer_abv  review_aroma  review_appearance  review_palate  \
       777471        13014  0.121900         0.750                1.0          1.000
       530510          743  0.096897         0.625                0.8          0.625
       1197430        9629  0.187034         0.750                0.6          0.750
       1036712       11031  0.176634         0.750                0.9          1.000
       403152          694  0.190501         0.875                0.8          0.750

                review_taste  beer_style_cat
       777471          0.875              11
```

```
530510          0.625             14
1197430         0.750             11
1036712         0.750             89
403152          0.750             25
```

[311]:

```
df_train_val.to_csv('data/interim/beer_reviews_split_train_post_split.csv',␣
 ↪index=False)
```

[312]:

```
df_train_cleaned.to_csv('data/interim/beer_reviews_split_test_post_split.csv',␣
 ↪index=False)
```