l00171188-technicalproject1

February 8, 2023

[2]:
```
import pandas as pd          #importing librariries to import data to colab
import io
from google.colab import files



uploaded = files.upload()
beer_data = pd.read_csv(io.BytesIO(uploaded['beer_data.csv']), header = 0)
 ↪      #converting/reading the CSV file in form of a dataframe
```

```
<IPython.core.display.HTML object>
```

```
Saving beer_data.csv to beer_data (2).csv
```

[3]:
```
type(beer_data)               #displaying the column names with df.columns
 ↪command
beer_data.columns
```

[3]:
```
Index(['Unnamed: 0', 'Name', 'Country', 'Brand', 'Categories', 'Type',
       'Tasting Notes', 'ABV', 'IBU', 'Calories Per Serving (12 OZ/0.35L)',
       'Carbs Per Serving (12 OZ/0.35L)', 'Food Pairing',
       'Suggested Serving Temperature', 'Rating', 'Rate Count', 'Price',
       'Volume', 'Description'],
      dtype='object')
```

[4]:
```
beer_data= beer_data.drop(['Unnamed: 0'],axis=1)          #dropping a column
```

[5]:
```
beer_data = beer_data.dropna(how='all')                          #dropping the
 ↪row if it has all NA values in it
beer_data = beer_data[['Name','Brand', 'Type', 'ABV', 'Price' ]]
```

[6]:
```
beer_data = beer_data.dropna(axis=0)                    #dropping the rows if
 ↪it has any NA values in it.
beer_data
```

[6]:

	Name	Brand	\
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company	
1	Pipeworks Lizard King	Pipeworks Brewing Company	
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company	

```
3                 Pipeworks Brief Relief    Pipeworks Brewing Company
4          Pipeworks Sangremancer Red Ale    Pipeworks Brewing Company
…                                     …                            …
13461  G's Summer Vibes Hard Ginger Beer          Gs Hard Ginger Beer
13465           Merchant's Hard Lemonade                    Merchants
13467                   Blueberry Mojito  Zesty Hard Kombucha Seltzer
13477  Blueprint Pumpkin Spiced Edinbrue        Brueprint Brewing Co.
13478                          Akos White                   Akos White

                            Type   ABV   Price
0                     Craft Beer    8%  $10.00
1                     Craft Beer    6%  $11.54
2                     Craft Beer  6.5%  $11.19
3                     Craft Beer    9%  $10.99
4                     Craft Beer  8.5%   $8.99
…                            …   …     …
13461                 Craft Beer  4.5%  $15.99
13465                 Craft Beer  4.5%  $11.93
13467  Independent Craft Brewers  4.5%  $12.00
13477  Independent Craft Brewers  8.2%  $12.50
13478                 Craft Beer  6.5%  $11.99

[7742 rows x 5 columns]
```

[7]:
```
import pandas as pd             #importing librariries to import data to colab
import io
from google.colab import files


uploaded = files.upload()
spirits_data = pd.read_csv(io.BytesIO(uploaded['spirits_data.csv']), header =␣
 ↪0)                        #converting/reading the CSV file in form of a␣
 ↪dataframe
```

```
<IPython.core.display.HTML object>

Saving spirits_data.csv to spirits_data (2).csv
```

[8]:
```
type(spirits_data)
spirits_data.columns                            #displaying the column names␣
 ↪with df.columns command
```

[8]:
```
Index(['Unnamed: 0', 'Name', 'Country', 'Brand', 'Categories', 'Tasting Notes',
       'ABV', 'Base Ingredient', 'Years Aged', 'Rating', 'Rate Count', 'Price',
       'Volume', 'Description'],
      dtype='object')
```

[9]:
```
spirits_data= spirits_data.drop(['Unnamed: 0'],axis=1)
 ↪#dropping a column
```

[10]:
```
spirits_data = spirits_data.dropna(how='all')
 ↪#dropping the row if it has all NA values in it
spirits_data = spirits_data[['Name','Brand', 'Categories', 'ABV', 'Price' ]]
```

[11]:
```
spirits_data = spirits_data.dropna(axis=0)
 ↪#dropping the rows if it has any NA values in it.
spirits_data
```

[11]:

	Name	Brand	Categories	ABV	Price
0	Dekuyper Triple Sec Liqueur	DeKuyper Liqueur	Citrus, Triple Sec Liqueur, Liqueur	24%	$10.99
1	DeKuyper Peachtree Schnapps Liqueur	DeKuyper Liqueur	Liqueur	20%	$11.69
2	DeKuyper Sour Apple Pucker Schnapps Liqueur	DeKuyper Liqueur	Liqueur	15%	$11.99
3	DeKuyper Blue Curacao Liqueur	DeKuyper Liqueur	Liqueur	24%	$11.99
4	DeKuyper Buttershots Schnapps Liqueur	DeKuyper Liqueur	Liqueur	15%	$12.99
…	…	…	… …		…
12862	Killepitsch	Killepitsch	Liqueur	35%	$23.99
12863	Or G French Liqueur	Or G	Liqueur	34%	$8.95
12865	Roiano Liquore	Roiano	Liqueur, Nuts, Amaretto Liqueur	40%	$16.99
12866	Very Special Chocolates Classic Assortment Liq…	Very Special Chocolates	Chocolate, Sweet Liqueur, Liqueur	5%	
12868	Don Felix Anejo Tequila	Don Felix	Anejo Tequila, Tequila	40%	

```
12866  $20.62
12868  $56.34

[10471 rows x 5 columns]
```

[12]:
```
spirits_data.columns = ['Name','Brand', 'Type', 'ABV', 'Price' ]
    #Changing the column name from category to Type
spirits_data
```

[12]:
```
                                                 Name  \
0                           Dekuyper Triple Sec Liqueur
1                     DeKuyper Peachtree Schnapps Liqueur
2              DeKuyper Sour Apple Pucker Schnapps Liqueur
3                           DeKuyper Blue Curacao Liqueur
4                   DeKuyper Buttershots Schnapps Liqueur
…                                                     …
12862                                         Killepitsch
12863                                  Or G French Liqueur
12865                                       Roiano Liquore
12866  Very Special Chocolates Classic Assortment Liq…
12868                               Don Felix Anejo Tequila

                         Brand                                    Type  ABV  \
0              DeKuyper Liqueur  Citrus, Triple Sec Liqueur, Liqueur  24%
1              DeKuyper Liqueur                                Liqueur  20%
2              DeKuyper Liqueur                                Liqueur  15%
3              DeKuyper Liqueur                                Liqueur  24%
4              DeKuyper Liqueur                                Liqueur  15%
…                             …                                …  …
12862               Killepitsch                                Liqueur  35%
12863                     Or G                                 Liqueur  34%
12865                   Roiano      Liqueur, Nuts, Amaretto Liqueur  40%
12866  Very Special Chocolates    Chocolate, Sweet Liqueur, Liqueur   5%
12868                 Don Felix               Anejo Tequila, Tequila  40%

        Price
0      $10.99
1      $11.69
2      $11.99
3      $11.99
4      $12.99
…       …
12862  $23.99
12863   $8.95
12865  $16.99
12866  $20.62
12868  $56.34
```

```
[10471 rows x 5 columns]
```

[13]:
```
import pandas as pd
import io
from google.colab import files


uploaded = files.upload()
wine_data = pd.read_csv(io.BytesIO(uploaded['wine_data.csv']), header = 0)
```

```
<IPython.core.display.HTML object>
```

```
Saving wine_data.csv to wine_data (2).csv
```

[14]:
```
type(wine_data)
wine_data.columns
```

[14]:
```
Index(['Unnamed: 0', 'Name', 'Country', 'Brand', 'Categories', 'Tasting Notes',
       'ABV', 'Food Pairing', 'Suggested Glassware',
       'Suggested Serving Temperature', 'Sweet-Dry Scale', 'Body', 'Rating',
       'Rate Count', 'Price', 'Volume', 'Description'],
      dtype='object')
```

[15]:
```
wine_data= wine_data.drop(['Unnamed: 0'],axis=1)
```

[16]:
```
wine_data = wine_data.dropna(how='all')
wine_data = wine_data[['Name','Brand', 'Categories', 'ABV', 'Price' ]]
```

[17]:
```
wine_data = wine_data.dropna(axis=0)
wine_data
```

[17]:

	Name	Brand	Categories	ABV	Price
9	Block 537 Merlot Dry Creek	Vineyard Block Estate	Merlot, Red Wine	13.9%	$20.99
10	Block 049 Merlot	Vineyard Block Estate	Merlot, Red Wine	14.5%	$21.99
23	Block 115 Arroyo Grande Valley Pinot Noir	Vineyard Block Estate			
24	Block 664 Pinot Noir Pommard	Vineyard Block Estate			
29	Block 012 Red Wine Oak Knoll	Vineyard Block Estate			
…	…	…			
26091	Anne Brigitte, Pays d'Oc 2018, Rosé Wine	Anne Brigitte			
26092	Tribute To Grace Rose of Grenache 2016	A Tribute To Grace			
26095	Angel Affair Rosé	Angel Affair			
26096	Accademia dei Racemi Burlesque Rose	Accademia dei Racemi			
26098	Centorri Moscato Di Pavia	Centorri			

```
23     Pinot Noir, Red Wine    14%  $24.99
24     Pinot Noir, Red Wine  13.5%  $25.99
29      Red Blend, Red Wine  14.5%  $20.99
…                       …    …      …
26091   Pink Wine, Ros Wine    13%   $0.00
26092   Pink Wine, Ros Wine  13.1%   $0.00
26095   Pink Wine, Ros Wine  12.5%  $11.99
26096   Pink Wine, Ros Wine    13%  $12.99
26098   Moscato, White Wine   6.5%  $12.99

[15672 rows x 5 columns]
```

[18]:
```
wine_data.columns = ['Name','Brand', 'Type', 'ABV', 'Price' ]
wine_data
```

[18]:
```
                                          Name                  Brand  \
9                    Block 537 Merlot Dry Creek  Vineyard Block Estate
10                             Block 049 Merlot  Vineyard Block Estate
23     Block 115 Arroyo Grande Valley Pinot Noir  Vineyard Block Estate
24                 Block 664 Pinot Noir Pommard  Vineyard Block Estate
29                 Block 012 Red Wine Oak Knoll  Vineyard Block Estate
…                                            …                     …
26091   Anne Brigitte, Pays d'Oc 2018, Rosé Wine          Anne Brigitte
26092     Tribute To Grace Rose of Grenache 2016     A Tribute To Grace
26095                          Angel Affair Rosé           Angel Affair
26096        Accademia dei Racemi Burlesque Rose   Accademia dei Racemi
26098                  Centorri Moscato Di Pavia               Centorri

                       Type    ABV   Price
9          Merlot, Red Wine  13.9%  $20.99
10         Merlot, Red Wine  14.5%  $21.99
23     Pinot Noir, Red Wine    14%  $24.99
24     Pinot Noir, Red Wine  13.5%  $25.99
29      Red Blend, Red Wine  14.5%  $20.99
…                       …    …      …
26091   Pink Wine, Ros Wine    13%   $0.00
26092   Pink Wine, Ros Wine  13.1%   $0.00
26095   Pink Wine, Ros Wine  12.5%  $11.99
26096   Pink Wine, Ros Wine    13%  $12.99
26098   Moscato, White Wine   6.5%  $12.99

[15672 rows x 5 columns]
```

[19]:
```
combined_df = pd.concat([beer_data, spirits_data, wine_data])
 ↪#joining all the three dataframes together
combined_df
```

[19]:
```
                                        Name                       Brand  \
0                  Pipeworks Ninja vs. Unicorn  Pipeworks Brewing Company
1                        Pipeworks Lizard King  Pipeworks Brewing Company
2                Pipeworks Blood Of The Unicorn  Pipeworks Brewing Company
3                        Pipeworks Brief Relief  Pipeworks Brewing Company
4                Pipeworks Sangremancer Red Ale  Pipeworks Brewing Company
…                                           …                          …
26091  Anne Brigitte, Pays d'Oc 2018, Rosé Wine              Anne Brigitte
26092    Tribute To Grace Rose of Grenache 2016         A Tribute To Grace
26095                          Angel Affair Rosé               Angel Affair
26096       Accademia dei Racemi Burlesque Rose       Accademia dei Racemi
26098                  Centorri Moscato Di Pavia                   Centorri

                       Type    ABV   Price
0               Craft Beer     8%  $10.00
1               Craft Beer     6%  $11.54
2               Craft Beer   6.5%  $11.19
3               Craft Beer     9%  $10.99
4               Craft Beer   8.5%   $8.99
…                      …    …      …
26091  Pink Wine, Ros Wine    13%   $0.00
26092  Pink Wine, Ros Wine  13.1%   $0.00
26095  Pink Wine, Ros Wine  12.5%  $11.99
26096  Pink Wine, Ros Wine    13%  $12.99
26098  Moscato, White Wine   6.5%  $12.99

[33885 rows x 5 columns]
```

[20]:
```
%matplotlib inline
import matplotlib.pyplot as plt                                  #importing matplotlib␣
 ↪and seaborn library to perform visualisation
import seaborn as sn
```

[21]:
```
import plotly.express as px

df = combined_df
fig = px.scatter(df, x="ABV", y="Price",
                 width=800, height=400)

fig.update_layout(
    margin=dict(l=20, r=20, t=20, b=20),                  #creating layout for␣
 ↪the graph
    paper_bgcolor="LightSteelBlue",
)
fig.update_xaxes(categoryorder='category ascending')          #updating axes with␣
 ↪parameter catergory ascending to have axes labels in ascending orders
fig.update_yaxes(categoryorder='category ascending')
```

```
fig.show()
```

[22]:
```
combined_df.dtypes                    #checking data types of the df
```

[22]:
```
Name     object
Brand    object
Type     object
ABV      object
Price    object
dtype: object
```

[23]:
```
#combined_df['Type'] = test_df.Type.str.replace(r'(^.*Beer.*$)', 'Beer')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Craft.*$)', 'Beer')
 ↪            # categorising the bulk data under different names to one
 ↪collective name.
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Vodka.*$)', 'Vodka')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Rum.*$)', 'Rum')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Brandy.*$)', 'Brandy')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Tequila.*$)',
 ↪'Tequila')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Liqueur.*$)',
 ↪'Liqueur')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Ready.*$)', 'Alcohol')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Whiskey.*$)',
 ↪'Whiskey')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Gin.*$)', 'Gin')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Wine.*$)', 'Wine')
combined_df
```

```
<ipython-input-23-222fe82c912a>:2: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:3: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:4: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:5: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:6: FutureWarning:
```

```
The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:7: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:8: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:9: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:10: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:11: FutureWarning:

The default value of regex will change from True to False in a future version.
```

[23]:

	Name	Brand	Type	ABV	Price
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company	Beer	8%	$10.00
1	Pipeworks Lizard King	Pipeworks Brewing Company	Beer	6%	$11.54
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company	Beer	6.5%	$11.19
3	Pipeworks Brief Relief	Pipeworks Brewing Company	Beer	9%	$10.99
4	Pipeworks Sangremancer Red Ale	Pipeworks Brewing Company	Beer	8.5%	$8.99
…	…	…	…	…	…
26091	Anne Brigitte, Pays d'Oc 2018, Rosé Wine	Anne Brigitte	Wine	13%	$0.00
26092	Tribute To Grace Rose of Grenache 2016	A Tribute To Grace	Wine	13.1%	$0.00
26095	Angel Affair Rosé	Angel Affair	Wine	12.5%	$11.99
26096	Accademia dei Racemi Burlesque Rose	Accademia dei Racemi	Wine	13%	$12.99
26098	Centorri Moscato Di Pavia	Centorri			

```
26098  Wine   6.5%  $12.99

[33885 rows x 5 columns]
```

[24]:
```
combined_df['Price'] = combined_df['Price'].str.replace('$', '')
 ↪#replacing the speacial characters in Price and ABV rows with spaces
combined_df['Price'] = combined_df['Price'].astype(float)
 ↪#converting the Price and ABV column values to float from object data type.
combined_df['ABV'] = combined_df['ABV'].str.replace('%', '')
combined_df['ABV'] = combined_df['ABV'].astype(float)
combined_df
```

```
<ipython-input-24-66dfd5eb1eaf>:1: FutureWarning:

The default value of regex will change from True to False in a future version.
In addition, single character regular expressions will *not* be treated as
literal strings when regex=True.
```

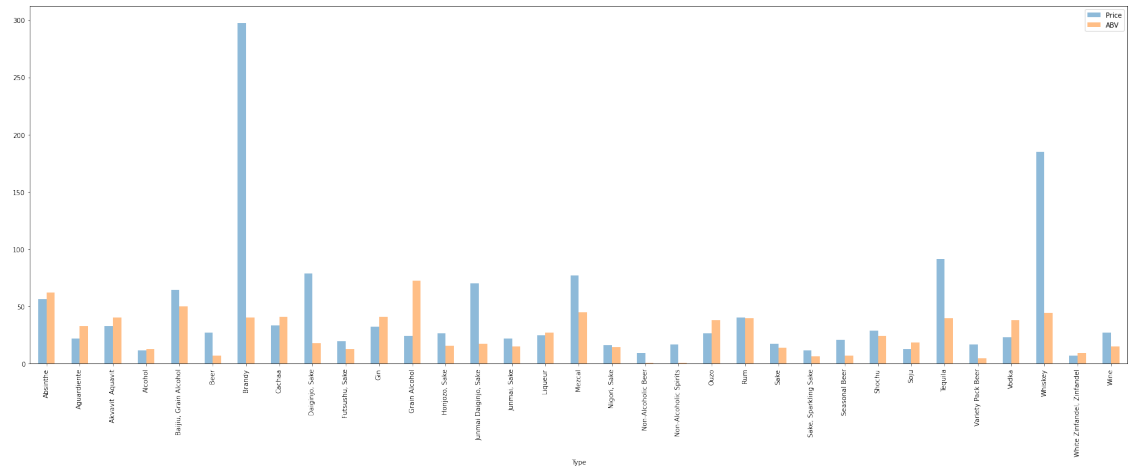
[24]:

	Name	Brand	Type	ABV	Price
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company	Beer	8.0	10.00
1	Pipeworks Lizard King	Pipeworks Brewing Company	Beer	6.0	11.54
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company	Beer	6.5	11.19
3	Pipeworks Brief Relief	Pipeworks Brewing Company	Beer	9.0	10.99
4	Pipeworks Sangremancer Red Ale	Pipeworks Brewing Company	Beer	8.5	8.99
…	…	…	…	…	…
26091	Anne Brigitte, Pays d'Oc 2018, Rosé Wine	Anne Brigitte	Wine	13.0	0.00
26092	Tribute To Grace Rose of Grenache 2016	A Tribute To Grace	Wine	13.1	0.00
26095	Angel Affair Rosé	Angel Affair	Wine	12.5	11.99
26096	Accademia dei Racemi Burlesque Rose	Accademia dei Racemi	Wine	13.0	12.99
26098	Centorri Moscato Di Pavia	Centorri	Wine	6.5	12.99

```
[33885 rows x 5 columns]
```

[25]:
```
combined_df[['Type', 'Price','ABV']].groupby(by=['Type']).mean().plot.
↪bar(alpha=0.5,figsize=(30,10))              #visualising the data in bar␣
↪graph with matplotlib library
```

[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7fefb7cd93a0>



[26]:
```
combined_df
```

[26]:

	Name	Brand
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company
1	Pipeworks Lizard King	Pipeworks Brewing Company
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company
3	Pipeworks Brief Relief	Pipeworks Brewing Company
4	Pipeworks Sangremancer Red Ale	Pipeworks Brewing Company
…	…	…
26091	Anne Brigitte, Pays d'Oc 2018, Rosé Wine	Anne Brigitte
26092	Tribute To Grace Rose of Grenache 2016	A Tribute To Grace
26095	Angel Affair Rosé	Angel Affair
26096	Accademia dei Racemi Burlesque Rose	Accademia dei Racemi
26098	Centorri Moscato Di Pavia	Centorri

	Type	ABV	Price
0	Beer	8.0	10.00
1	Beer	6.0	11.54
2	Beer	6.5	11.19
3	Beer	9.0	10.99
4	Beer	8.5	8.99
…	…	…	…
26091	Wine	13.0	0.00
26092	Wine	13.1	0.00
26095	Wine	12.5	11.99

```
26096  Wine  13.0  12.99
26098  Wine   6.5  12.99

[33885 rows x 5 columns]
```

[27]:
```
combined_df = combined_df.iloc[:,:].values                    #converting a␣
 ↪df to array to transform the values
combined_df
```

[27]:
```
array([['Pipeworks Ninja vs. Unicorn', 'Pipeworks Brewing Company',
        'Beer', 8.0, 10.0],
       ['Pipeworks Lizard King', 'Pipeworks Brewing Company', 'Beer',
        6.0, 11.54],
       ['Pipeworks Blood Of The Unicorn', 'Pipeworks Brewing Company',
        'Beer', 6.5, 11.19],
       …,
       ['Angel Affair Rosé', 'Angel Affair', 'Wine', 12.5, 11.99],
       ['Accademia dei Racemi Burlesque Rose', 'Accademia dei Racemi',
        'Wine', 13.0, 12.99],
       ['Centorri Moscato Di Pavia', 'Centorri', 'Wine', 6.5, 12.99]],
      dtype=object)
```

[28]:
```
from sklearn.preprocessing import LabelEncoder
```

[29]:
```
transform = LabelEncoder()
combined_df[:,0]= transform.fit_transform(combined_df[:,0])                #␣
 ↪coverting the string data to numerical values using Label encoder
combined_df[:,1]= transform.fit_transform(combined_df[:,1])
combined_df[:,2]= transform.fit_transform(combined_df[:,2])

combined_df
```

[29]:
```
array([[24057, 7898, 5, 8.0, 10.0],
       [24047, 7898, 5, 6.0, 11.54],
       [23996, 7898, 5, 6.5, 11.19],
       …,
       [1489, 405, 32, 12.5, 11.99],
       [689, 143, 32, 13.0, 12.99],
       [6463, 2038, 32, 6.5, 12.99]], dtype=object)
```

[30]:
```
combined_df = pd.DataFrame(combined_df)                                    #␣
 ↪coverting the array back to data frame
combined_df.columns = ['Name','Brand', 'Type', 'ABV', 'Price' ]
combined_df
```

[30]:
```
        Name Brand Type   ABV  Price
0      24057  7898    5   8.0   10.0
```

```
1       24047  7898    5   6.0  11.54
2       23996  7898    5   6.5  11.19
3       24002  7898    5   9.0  10.99
4       24070  7898    5   8.5   8.99
…         …    …     …    …     …
33880    1547   428   32  13.0    0.0
33881   30800    99   32  13.1    0.0
33882    1489   405   32  12.5  11.99
33883     689   143   32  13.0  12.99
33884    6463  2038   32   6.5  12.99

[33885 rows x 5 columns]
```

[32]:
```
combined_df['ABV'] = combined_df['ABV'].astype(float)
 ↪#converting the all the column data types to float
combined_df['Name'] = combined_df['Name'].astype(float)
combined_df['Brand'] = combined_df['Brand'].astype(float)
combined_df['Type'] = combined_df['Type'].astype(float)
combined_df['Price'] = combined_df['Price'].astype(float)
```

[34]:
```
price_col = combined_df['Price']
price_col.replace(to_replace = 0, value =pd.NA, inplace=True)
 ↪ #replacing the 0 values with Na in Price column
```

[35]:
```
abv_col =combined_df['ABV']
abv_col.replace(to_replace = 0, value = pd.NA, inplace=True)
 ↪ #replacing the 0 values with Na in ABV column
```

[36]:
```
#combined_df = combined_df.dropna(how='all')
combined_df
```

[36]:
```
          Name   Brand  Type   ABV  Price
0      24057.0  7898.0   5.0   8.0   10.0
1      24047.0  7898.0   5.0   6.0  11.54
2      23996.0  7898.0   5.0   6.5  11.19
3      24002.0  7898.0   5.0   9.0  10.99
4      24070.0  7898.0   5.0   8.5   8.99
…        …      …    …    …    …     …
33880   1547.0   428.0  32.0  13.0   <NA>
33881  30800.0    99.0  32.0  13.1   <NA>
33882   1489.0   405.0  32.0  12.5  11.99
33883    689.0   143.0  32.0  13.0  12.99
33884   6463.0  2038.0  32.0   6.5  12.99

[33885 rows x 5 columns]
```

[37]:
```
combined_df = combined_df.dropna(axis=0)
 ↪#dropping the Na values from the data frame
combined_df
```

[37]:

	Name	Brand	Type	ABV	Price
0	24057.0	7898.0	5.0	8.0	10.0
1	24047.0	7898.0	5.0	6.0	11.54
2	23996.0	7898.0	5.0	6.5	11.19
3	24002.0	7898.0	5.0	9.0	10.99
4	24070.0	7898.0	5.0	8.5	8.99
…	…	…	…	…	…
33878	1600.0	454.0	32.0	12.0	29.95
33879	1149.0	277.0	32.0	12.6	9.99
33882	1489.0	405.0	32.0	12.5	11.99
33883	689.0	143.0	32.0	13.0	12.99
33884	6463.0	2038.0	32.0	6.5	12.99

[32592 rows x 5 columns]

[38]:
```
q_low = combined_df["Price"].quantile(0.01)
 ↪#removing the outliers from the data
q_hi  = combined_df["Price"].quantile(0.99)

combined_df = combined_df[(combined_df["Price"] < q_hi) & (combined_df["Price"]
 ↪> q_low)]
```

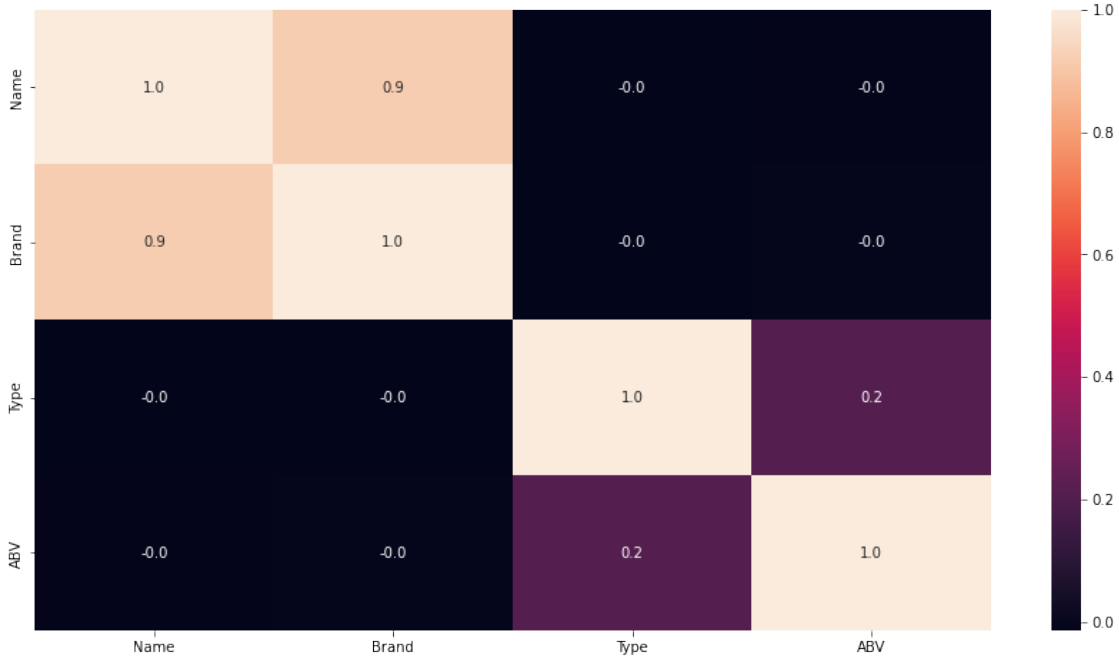
[39]:
```
q_low = combined_df["ABV"].quantile(0.01)
 ↪#removing the outliers from the data
q_hi  = combined_df["ABV"].quantile(0.99)

combined_df = combined_df[(combined_df["ABV"] < q_hi) & (combined_df["ABV"] >
 ↪q_low)]
```

[40]:
```
corr_matrix = combined_df.corr()

import seaborn as sn
import matplotlib.pyplot as plt
 ↪#visualising with the heat map to see correlation between the features in
 ↪data.

plt.figure(figsize=(15,8))
sn.heatmap(corr_matrix, annot=True, fmt=".1f")
plt.show()
```



[41]:
```
combined_df = combined_df[['Name','Type', 'ABV', 'Price' ]]
  ↪   #calling the df with new columns as Name and Brand has high correlation
combined_df
```

[41]:

	Name	Type	ABV	Price
0	24057.0	5.0	8.0	10.0
1	24047.0	5.0	6.0	11.54
2	23996.0	5.0	6.5	11.19
3	24002.0	5.0	9.0	10.99
4	24070.0	5.0	8.5	8.99
…	…	…	…	…
33878	1600.0	32.0	12.0	29.95
33879	1149.0	32.0	12.6	9.99
33882	1489.0	32.0	12.5	11.99
33883	689.0	32.0	13.0	12.99
33884	6463.0	32.0	6.5	12.99

[31295 rows x 4 columns]

[42]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

[43]:
```
X = combined_df.drop(columns=['Price'])
  ↪#seperating the df with the true label to train and test
Y = combined_df['Price']
```

[44]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=3)                                    #applying PCA␣
 ↪with n_components parameter to remove unnecessary features.
X2 = pca.fit_transform(X)
```

[45]:
```
from sklearn import preprocessing
from sklearn import utils

#convert y values to categorical values
lab = preprocessing.LabelEncoder()
y_transformed = lab.fit_transform(Y)                         #tranforming Y to␣
 ↪make sure it works with classification algorithms too
```

[46]:
```
X.dtypes
```

[46]:
```
Name    float64
Type    float64
ABV     float64
dtype: object
```

[47]:
```
Y.dtypes
```

[47]:
```
dtype('O')
```

[48]:
```
combined_df.dtypes
```

[48]:
```
Name     float64
Type     float64
ABV      float64
Price     object
dtype: object
```

[49]:
```
combined_df['Price'] = combined_df['Price'].astype(float)
```

[50]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, mean_absolute_error,␣
 ↪mean_squared_error
import matplotlib
import matplotlib.pyplot as plt
import numpy as np


X_train, X_test, Y_train, Y_test = train_test_split(X,y_transformed, train_size␣
 ↪= 0.8)                 #splitting data into train and test
```

```
clf = LinearRegression()
clf.fit(X_train, Y_train)                                    #applying the model to the␣
 ↪data

predictions = clf.predict(X_test)                            # prediction using test␣
 ↪data

print("Score:", clf.score(X_test, Y_test))                   #checking accuracy score␣
 ↪by comparing the actual results with the values from true label
```

```
Score: 0.2698086964582588
```

[52]:
```
combined_df.info()                          # checking the information of dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 31295 entries, 0 to 33884
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    31295 non-null  float64
 1   Type    31295 non-null  float64
 2   ABV     31295 non-null  float64
 3   Price   31295 non-null  float64
dtypes: float64(4)
memory usage: 1.2 MB
```

[ ]:
```
#X.shape
```

[ ]:
```
#Y.shape
```

[61]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
clf = RandomForestClassifier(n_estimators=10)
clf = clf.fit(X_train, Y_train)                              #applying the model to␣
 ↪the data

predictions = clf.predict(X_test)                           # prediction using test␣
 ↪data

score = accuracy_score(Y_test, predictions)                    #checking accuracy␣
 ↪score by comparing the actual results with the values from true label
score
```

[61]:
```
0.07397347819140437
```