Exploratory Data Analysis - report

github.com/Macsok

October 2024

1 Introduction

1.1 What is EDA?

EDA stands for Exploratory Data Analysis. It's a crucial step in the data analysis process where you summarize the main characteristics of a dataset, often using visual methods. Here's the gist:

- Purpose: Understand the data better before jumping into modeling. It helps in uncovering patterns, spotting anomalies, testing hypotheses, and checking assumptions.
- Tools and Techniques: Descriptive statistics (mean, median, mode, standard deviation), visualizations (histograms, scatter plots, box plots), and data wrangling (handling missing values, scaling).
- Outcome: You get insights that guide your next steps in the data science workflow.

You can think of it as getting to know your dataset inside out, making sure you're fully prepped before the heavy lifting begins.

1.2 An aim

This analysis aims to understand the dataset of 134 cocktails. It aims to identify main characteristics of the cocktails, understand relations between ingredients and distinguish potential groups of similar drinks.

1.3 An approach

The approach is based on uncovering pure properties, relations and information from every datum that TheCocktailDB provides us with. It all started by subtle looking and each column and examining its content. The goal was to extrude as much information as we could (even exploring 'createdAt' and 'updatedAt' columns). After testing and making brief reconnaissance we moved to analysing cocktails alone, then ingredients dataframe, next we tried to merged them together and unravel relations between them, finally we analised whole set using scikit-learn library. In other words we started with data type manipulation and clearing data, moved through exploring single

properties, and landed on clustering our set and suggesting similar drinks. **I highly suggest you to go file by file and follow thinking pattern that took us there.**

1.3.1 Brief overlook of the dataset

Main part of the data collection consists of **134** rows and **11** columns. Here is brief look at it:

id	name	category	glass	tags	instructions	imageUrl	alcoholic	createdAt	updatedAt	ingredients
11000	Mojito	Cocktail	Highball glass	[IBA, ContemporaryClassic, Alcoholic, USA, Asi...	Muddle mint leaves with sugar and lime juice. ...	https://cocktails.solvro.pl/images/ingredients...	1	2024-08-18T19:01:17.000+00:00	2024-08-18T19:06:16.000+00:00	[{'id': 170, 'name': 'Soda water', 'descriptio...
11001	Old Fashioned	Cocktail	Old-fashioned glass	[IBA, Classic, Alcoholic, Expensive, Savory]	Place sugar cube in old fashioned glass and sa...	https://cocktails.solvro.pl/images/ingredients...	1	2024-08-18T19:01:58.000+00:00	2024-08-18T19:06:17.000+00:00	[{'id': 513, 'name': 'Water', 'description': '...

Figure 1: TheCocktailDB head

The columns are: id, name, category, glass, tags, instructions, imageUrl, alcoholic, createdAt, updatedAt, ingredients. Every column excpet 'id' and 'alcoholic' have object type. Other two are represented as int64.

In 'ingredients' column there is nested information about ingredients. Each drink have own piece of ingredients information there. If you unpack such block of information there will be data about ingredients needed to prepare such drink. Every record in 'ingredients' table has 10 attributes. They are as follows: id, name, alcohol, type, percentage, imageUrl, createdAt, updatedAt, measure. After unpacking ingredients data from every cocktail there will be **531** unique records in the 'ingredients' table.

id	name	description	alcohol	type	percentage	imageUrl	createdAt	updatedAt	measure
170	Soda water	None	1	None	NaN	None	2024-08-18T19:01:57.000+00:00	2024-08-18T19:01:57.000+00:00	NaN
305	Light Rum	Light rums, also referred to as "silver" or "w...	1	Rum	NaN	https://cocktails.solvro.pl/images/ingredients...	2024-08-18T19:02:37.000+00:00	2024-08-18T19:02:37.000+00:00	2-3 oz
312	Lime	A lime (from French lime, from Arabic lima, fr...	0	Fruit	NaN	https://cocktails.solvro.pl/images/ingredients...	2024-08-18T19:02:40.000+00:00	2024-08-18T19:02:40.000+00:00	Juice of 1
337	Mint	Lamiaceae (/ˌleɪmiˈeɪsiˌaɪ/ or /ˌleɪmiˈeɪsiːiː/...	0	Flower	NaN	https://cocktails.solvro.pl/images/ingredients...	2024-08-18T19:02:47.000+00:00	2024-08-18T19:02:47.000+00:00	2-4
476	Sugar	Sugar is the generic name for sweet-tasting, s...	0	None	NaN	https://cocktails.solvro.pl/images/ingredients...	2024-08-18T19:03:31.000+00:00	2024-08-18T19:03:31.000+00:00	2 tsp

Figure 2: A snippet of the 'ingredients' table

2 Data review

2.1 Data loading

In this section, we will demonstrate the steps required to load the dataset into our analysis environment. As we found out data is stored in .json format, so loading it into pandas DataFrame looks like this:

```
#reading raw data
df = pd.read_json(path_or_buf='../data/cocktail_dataset.json')
```

Figure 3: Loading data and storing it as pandas DataFrame

2.2 Missing values

In this section we will examine missing values in our DataFrame and correct type of data in time columns. Figure 4 represents summed rows of missing values in each column of the dataset. We can see that most of the 'tags' column is missing (**99 out of 134**). Figure 5 shows code used to correct data type in 'createdAt' and 'updatedAt' columns.

```
df.isna().sum()
```

```
id                0
name              0
category          0
glass             0
tags             99
instructions      0
imageUrl          0
alcoholic         0
createdAt         0
updatedAt         0
ingredients       0
dtype: int64
```

Figure 4: Sums on missing values

```
df['createdAt'] = pd.to_datetime(df['createdAt'])
df['updatedAt'] = pd.to_datetime(df['updatedAt'])
```

Figure 5: Data type correction

2.3 Brief exploration

After short reconnaissance of the data in our DataFrame you can spot some important properties:

- every cocktail is alcoholic
- imageUrl won't be useful in the data analysis (at least for us)
- id's aren't consistent
- 'ingredients' column is stuffed with a .json data

In that case further explorations of the dataset will exclude 'imageUrl' and 'alcoholic' columns.

3 Exploring cocktails alone

In this section we pay attention only to cocktails - we won't unpack data from 'ingredients' column. In this case we drop four columns: 'ingredients', 'imageUrl', 'alcoholic', 'tags'.

3.1 Is there something hidden in those times?

Creating and updating times may seem harmless and boring but is there something hidden? Let's plot them and find out! Figure 6 shows the result, seems that nothing interesting was hidden there. Next we will simplify a little and assume that length of the instruction is somehow relevant to level of preparation complexity. We will replace textual value of instructions with the length of it. Below (on Figure 7) you can see brief code and the results of such operation. In this case we can, for example, find relations between type of the glass and the instruction length. Figure 8 presents lenghts of the instruction per every type of glass. Each red dot represents one cocktail. We can spot that Highball glasses tend to have a little longer instructions than Cocktail glasses. Old-fashioned glass drinks' instruction length vary more than Cocktails glass ones which are more regular.

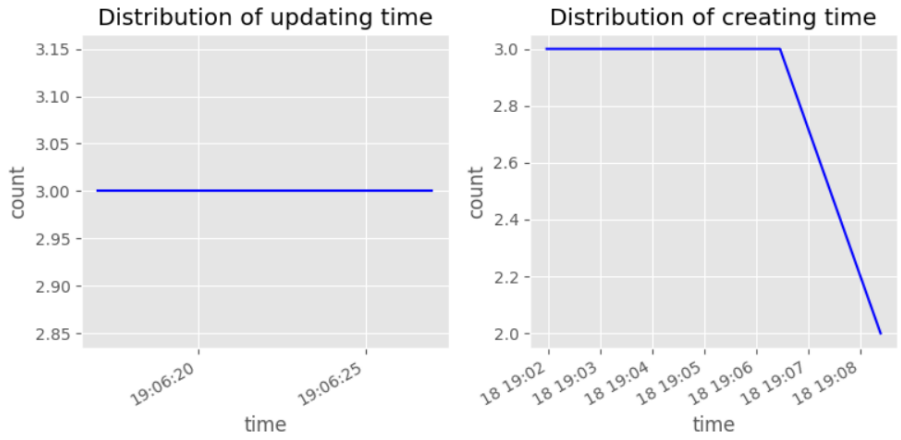

Figure 6: Times of creating and upgrading cocktails

```
df['instructions'] = df['instructions'].apply(lambda a: len(a))

df.head()
```

id	name	category	glass	instructions	createdAt	updatedAt
11000	Mojito	Cocktail	Highball glass	177	2024-08-18 19:01:17+00:00	2024-08-18 19:06:16+00:00
11001	Old Fashioned	Cocktail	Old-fashioned glass	218	2024-08-18 19:01:58+00:00	2024-08-18 19:06:17+00:00
11002	Long Island Tea	Ordinary Drink	Highball glass	152	2024-08-18 19:01:58+00:00	2024-08-18 19:06:17+00:00
11003	Negroni	Ordinary Drink	Old-fashioned glass	44	2024-08-18 19:01:58+00:00	2024-08-18 19:06:17+00:00
11004	Whiskey Sour	Ordinary Drink	Old-fashioned glass	148	2024-08-18 19:01:59+00:00	2024-08-18 19:06:18+00:00

Figure 7: Replacement of the instruction and current look at head of our cocktails data

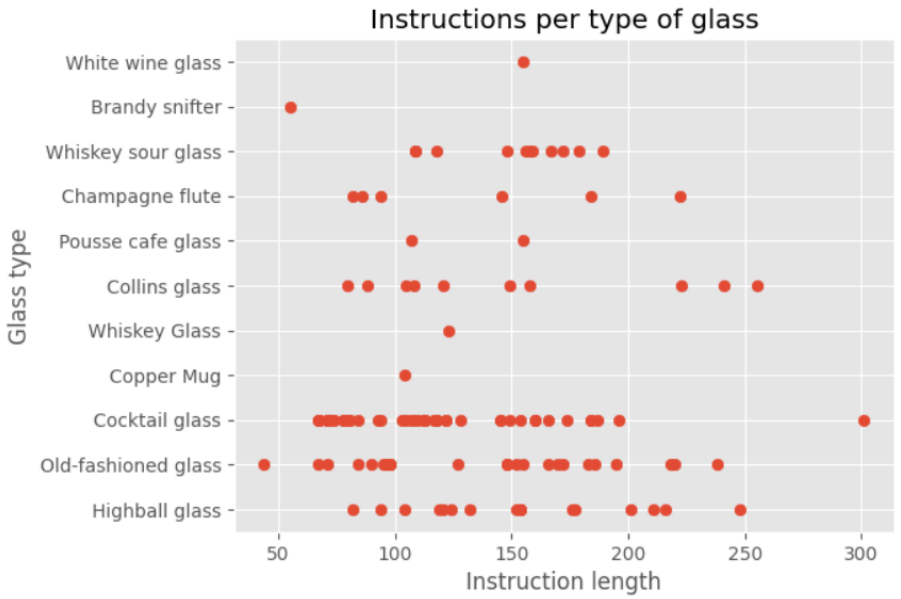

Figure 8: Length of the instruction per type of the glass

3.2 Dividing into groups

Let's take a look at popularity of each drink category. There are three such categories: Punch / Party Drink, Cocktail, Ordinary Drink. To count how many drinks are in each category we will use function: value_counts(), then we will simply plot it as horizontal bar plot. Here is the result (Figure 9):

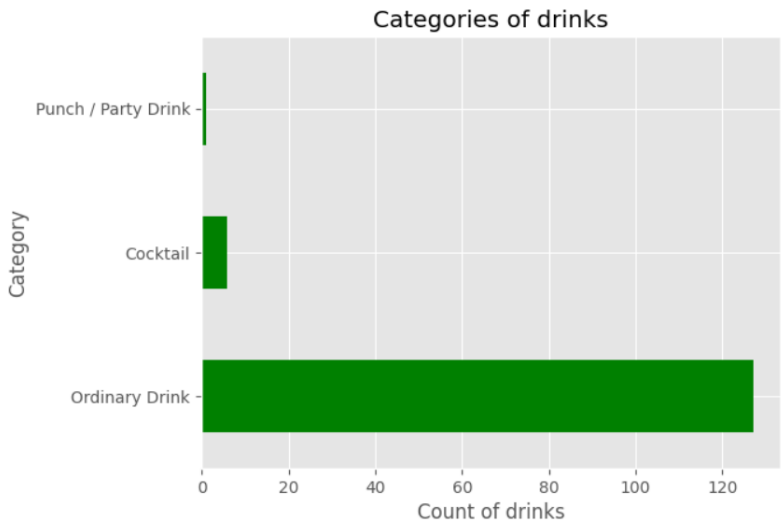

Figure 9: Count of drinks in each category

Similarly we can count the frequency of using each glass type in drinks and present the mon bar plot. This time we will focus on top 10 used glass types.

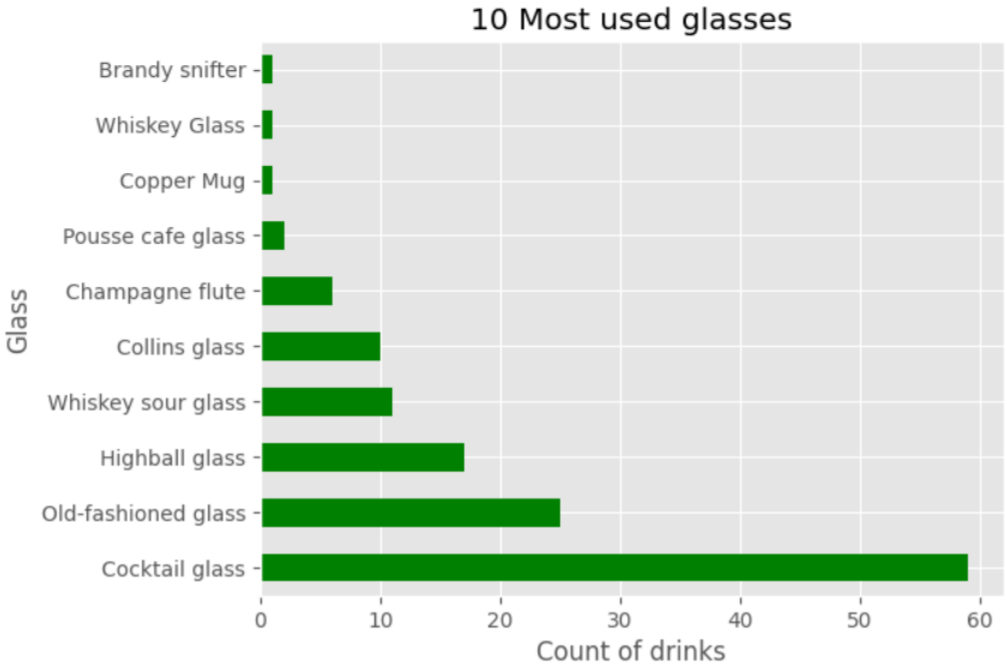

Figure 10: Top 10 used glasses

As we can see most commonly used glass type is Cocktail glass, second one is Old-fashioned glass and on third position there is Highball glass. The top one leaves opponents far behind.