Difford's Guide

FOR DISCERNING DRINKERS

Cocktail Recommendation: a fun approach to discover new drinks

12 June 2023

Group R - Samy Maurer & Alexander Liden

Machine Learning in Business Analytics 2023

UNIL | Université de Lausanne

HEC Lausanne

Non-plagiarism statement

Alexander Liden : *alexander.liden@unil.ch*
Samy Maurer : *samy.maurer@unil.ch*

This project was written by us and in our own words, except for quotations from published and unpublished sources, which are clearly indicated and acknowledged as such. We are conscious that the incorporation of material from other works or a paraphrase of such material without acknowledgement will be treated as plagiarism, subject to the custom and usage of the subject, according to the University Regulations. The source of any picture, map or other illustration is also indicated, as is the source, published or unpublished, of any material not resulting from our own research

Abstract

The world of cocktails, while fascinating, can often be overwhelming for beginners and enthusiasts alike. Our project aims to simplify this landscape by developing a classification algorithm capable of suggesting new and exciting cocktails based on individual preferences.
Using the well-renowned Difford's Guide to Cocktails as our primary source, we meticulously build a database of cocktail recipes, including their detailed ingredient lists and preparation instructions based on a method called “Scraping” before using a one-hot encoder on the scraped data.
We followed by doing data cleaning and exploratory data analysis on the huge brut dataset. After some basic statistical analysis, we analyzed the correlation between multiple features and the distribution of each of them in order to understand the overall data and which feature is important. Through EDA, we aim to understand the data's underlying structure and the relationships between different variables, such as the type of ingredients used, the classification, as well as the resulting flavor profiles to determine whether or not the model will be able to correctly predict similar cocktails as the ones we like.

We choose four machine learning methods that will recommend cocktails based on a fixed input and compare iterations : Random Forest, Extreme Gradient Boosting, Variational Autoencoder, Cosine Similarity.

- **Random Forest** appears to suggest the two same cocktails, independently to the number of iterations which appears to be a problem for our recommendation system
- **Extreme Gradient Boosting** fits our data well as it suggests a good amount of cocktail that looks like the input.
- We leveraged subsets of input data in our **Variational Autoencoder** to overcome its inherent predictability, leading to an interesting finding that the top five recommended cocktails were frequently recurring in over a quarter of all iterations, though this came at the cost of reduced diversity and less targeted recommendations.
- **Cosine Similarity**, which is a kind of KNN method, cares a lot about the Gentle / Sweet axis but not much about the other Principal Components, which is pretty bad considering that the granularity of this axis is not very high.

The supervised learning methods appear to be way better when iterating lots of time due to its predictive function. At the end, results and graphics suggest that Extreme Gradient Boosting recommends best the input cocktails given.
It should also be interesting to see how our four models would work with a different number and names of cocktails as input.

Table of contents

1. Introduction

Cocktails are a vast and intriguing realm of flavors, ingredients, and techniques. Yet, for a beginner or someone unfamiliar with mixology, it can be quite overwhelming. You might find yourself **puzzled**, standing before your liquor cabinet, wondering what you can create with the ingredients available. You know what flavors you like, but what's the right cocktail for me?

Our project aims to solve this conundrum. We intend to develop a classification algorithm, a sort of a digital mixologist, that can help you discover new and exciting cocktails from the renowned Difford's Guide to Cocktails, often referred to as the ultimate compendium for cocktail enthusiasts.

The Difford's Guide is packed with an impressive array of cocktail recipes, complete with intricate ingredient details and step-by-step preparation instructions. We have set out to build a comprehensive **database** by scraping data from this guide, which will be the foundational resource for our cocktail suggestion algorithm.

The journey, however, does not end with gathering data. Extensive data cleaning and exploratory data analysis (EDA) follow, which are crucial steps to ensure that our algorithm operates with optimal accuracy. Every recipe in our database will be carefully scrutinized to identify and rectify any errors, inconsistencies, or issues that could compromise the algorithm's efficiency.

We will use supervised and unsupervised machine learning methods to recommend cocktails and see which one corresponds the best to the given input.

In Section 2, we will go through Data Description and how we scraped and cleaned the data from the internet. Section 3 will go through Exploratory Data Analysis and understanding how data behaves. We will be talking about our models in Section 4 and why we choose them in addition to displaying our libraries. Section 5 is about the graphs and results of our recommendations. We will end up with Section 6 and try to understand which model performs best and the limitations of our project. Section 7 and 8 are respectively for the few references and the appendix.

Our project holds exciting implications for the cocktail-making landscape. By simplifying the process of suggesting new drinks, we hope to ignite a new spark of creativity and experimentation in mixology. **Our classification algorithm is not just a tool for experienced bartenders**. It is designed for everyone. Even if you are an amateur or a newcomer, the algorithm will help you discover novel and delectable cocktails that align with your taste preferences.

So, whether you are dabbling in mixology for the first time or you are a seasoned pro, our project aims to take your cocktail-making game to the next level. All you need are your ingredients and a willingness to explore!

2. Data Description

2.1. Web-Scraping

Because the Difford's Guide does not allow direct access to their API or their databases, our only way of getting the data was up to a complicated start. We knew this term called "scraping", but we had no idea how it worked or even how to do it, so our research for resources began.

After a little digging, we found that we could transform the name part of the base URL of each cocktail (diffordsguide.com/cocktails/recipe/**1/abacaxi-ricaco**) by **index.html** (diffordsguide.com/cocktails/recipe/**1/index.html)**, which rendered the job very easy because we can now iterate through the almost 7000 cocktails without any complex problem solving.

Because the Guide has a lot of information on every cocktail, we wanted to retain as much of the information as possible. Here is the data description of the basic dataframe:

1. **Name**: Name of the cocktail. This is self-explanatory for it is different for every cocktail.

2. **Glass**: Type of glass typically used when serving this cocktail. This ranges from the shot glass to the old-fashioned. There are 44 different values to choose from.

3. **Garnish**: How to finish the cocktail, for example "Orange zest twist". Different for every cocktail, and we decided not to include this in the final dataframe.

4. **How to make**: Main way to execute the cocktail. There are 26 different values after cleaning.

5. **Contents**: Ingredients that go in the cocktail. This ranges from alcohols to liqueurs, passing by the syrups. There are 5262 different values, which is easily explainable because each cocktail is uniquely made.

6. **Ratios**: Ratio of each ingredient in every cocktail. This one will be linked to the `Contents` column to give us one column.

7. **Gentle / Boozy**: One of the most interesting parts of the Difford's Guide is that almost each cocktail was tested personally by Mr. Difford himself, and he graded each cocktail on a Gentle / Boozy, and Sweet / Sour scale. On the Gentle / Boozy scale, a Lower value means that the cocktail is Gentle, a pretty light cocktail. On the contrary, a Higher value means that the cocktail is Boozy and that the cocktail is pretty strong.

8. **Sweet / Dry Sour**: This one behaves the same way as the last one: a Lower score here means that the cocktail is Sweet, and a Higher score means that the cocktail is Dry or Sour.

9. **Calories**: The number of calories of each cocktail computed by the Guide.

10. **Calories2**: This one is a little bit tricky to explain without going into much details, we will explain its use a little later.

11. **Alcohol percentage**: The percentage of alcohol for each cocktail computed by the Guide.

12. **Alcohol percentage2**: This one is a little bit tricky to explain without going into much detail, we will explain its use a little later.

13. **Popularity**: The number of user-generated comments under each cocktail.

14. **Category**: The category of the cocktail. There are 47 unique values here.

15. **URL**: The static URL of the cocktail. Will be dropped later.

The method to scrape the site was pretty straight-forward, but pretty hard at the same time. We basically used CSS classes, more precisely the exact CSS selectors, manually for every column of the dataframe. This reduced the prettiness of the code because we could not use "direct" classes (i.e. "Alcohol Percentage", "Calories", etc...) to scrape to counter the scraping process.

This is the direct reason for the existence of the second columns for Calories and Alcohol Percentage; for some of the older or most popular cocktails, the Guide implemented a little nickname in the form a "AKA", right before the columns of interest. While it is useful to know the cute nickname of a cocktail, there is now an offset in the CSS classes that needs to be taken care of. For simplicity's sake and peace of mind, we decided to grab the two values directly for each cocktail rather than having to wait another 9 hours of scraping because of a mistake in the if-statement.

2.2. Data Cleaning

The goal of the cleaning process is to get a data frame encoded with values ranging from 0 to 1 when applicable (i.e. one-hot encoded) except for the name of the cocktail. This will definitely help us for the modeling part because it will help us determine easily what the most important variables are, and many more. One-hot encoding is a common preprocessing step used in machine learning and data science. The process involves converting categorical variables into a form that could be provided to machine learning algorithms to improve their performance.

Why One-Hot Encoding?

In our cocktails context, suppose we have a categorical variable like 'Method,' and it has three categories: 'Shaken,' 'Stirred,' and 'Blended.' Now, one way to represent these methods is by assigning each category a numerical value, say 1 for 'Shaken,' 2 for 'Stirred,' and 3 for 'Blended.' However, this representation implies an ordered relationship between the categories. The machine learning model may misinterpret these numbers as having a rank or order (i.e., 'Blended' > 'Stirred' > 'Shaken'), which is incorrect and can negatively impact the model's performance.

This is where one-hot encoding becomes useful. It allows us to create binary, or "dummy", variables for each category of a categorical variable. In our example, one-hot encoding would create three new variables: 'Method_Shaken,' 'Method_Stirred,' and 'Method_Blended.' For a 'Shaken' cocktail, 'Method_Shaken' would be 1, while 'Method_Stirred' and 'Method_Blended' would be 0.

One-Hot Encoding in Our Project:

In our project, one-**hot encoding will be applied to all categorical variables**. For example, each unique ingredient will be transformed into a separate feature that shows whether the cocktail contains that ingredient (1) or not (0). Similarly, for methods or other categorical features, each category will become a separate binary feature.

The one-hot encoding process not only **helps in handling categorical data but also aids in maintaining the comprehensibility of the model**. For instance, after one-hot encoding, the importance of each variable (ingredient or method) in determining the cocktail can be more easily interpreted. An ingredient or method that frequently appears with certain kinds of cocktails may be given more importance by the model, and thus have a higher coefficient in a linear model or higher feature importance in tree-based models.

Moreover, **one-hot encoding** will lead to a sparse matrix (i.e., a matrix with many zeros), making the computations faster, especially when we use algorithms that can handle sparse data well.

it is important to note, though, that one-hot encoding can considerably increase the dimensionality of our data (increase the number of features). Hence, it is always **crucial to perform feature selection or dimensionality reduction post-encoding** when necessary, to ensure our model is not overly complex and to prevent overfitting.

One-hot encoding is an essential step in our data cleaning process, providing a way to transform our categorical data into a numerical form suitable for machine learning algorithms and making our task of cocktail classification and clustering easier and more effective.

Now that we have explained why we will use one-hot encoding for the project, let us dive deeper in the actual cleaning part. The first part is to combine the doubled **`Calories` and `Alcohol Percentage`** columns into one. Next, we drop the redundant columns, as well as the **`Garnish` and `URL`** columns. We decided to drop the first one because after much consideration, it is not the reason why you order a cocktail in the first place, and it can change from bar to bar and person to person.

Next up, let us deal with the missing values. To be consistent with the use of the Difford's Guide, we had to drop the cocktails where there were now values in the **`Gentle / Boozy` and `Sweet / Dry Sour`** columns. Next was the choice of keeping the information about calories and alcohol percentage, but we knew that if we kept one, then we would have to keep the other one as well for consistency's sake.

We can not only keep the value for the Gentle / Boozy columns because as you will see later on in the EDA, the values on this axis are biased towards the Boozy side and are grouped in a rather small range, so we decided to keep it. Also, we deemed it important to keep the calories as someone may not want a highly caloric cocktail if they are on a diet or just prefer a more "**light**" cocktail.

Finally, **we decided to drop the missing values in the category**. This one has the same logic as the others: if you are looking for a long drink, you most certainly do not want to be recommended a shot.

After much more technical cleaning that you will find in the Jupyter project, we are left with a clean dataframe ready for the EDA.

3. Exploratory Data Analysis

In this part, we will look at how the variables behave with each other.

The first thing we will look at is the description of the Gentle / Boozy and Sweet / Dry Sour axes.

	Gentle / Boozy	Sweet / Dry Sour
count	1207.000000	1207.000000
mean	6.957746	6.304888
std	1.503684	1.142355
min	0.000000	1.000000
25%	6.000000	6.000000
50%	7.000000	6.000000
75%	8.000000	7.000000
max	10.000000	10.000000

Fig. 1 - Cocktails in the 2 axes

The analysis shows that the data for the **'Gentle / Boozy' and 'Sweet / Dry Sour'** dimensions indeed have different characteristics. The **'Gentle / Boozy'** dimension has a higher mean, median, and standard deviation than the **'Sweet / Dry Sour'** dimension.

Mean:

The mean, or average, is the sum of all values divided by the number of values. In our case, a higher mean in the **'Gentle / Boozy'** dimension indicates that cocktails are, on average, more **'Boozy'** than they are **'Sweet' or 'Sour'**.

This might suggest that the majority of cocktails in our dataset have a stronger alcohol flavor. This makes sense considering the context – cocktails are typically consumed in social situations where the goal might be to enjoy a drink with a noticeable alcohol content.

Median:

The median is the middle value of a data set; it separates the data into two halves. The fact that the median is higher in the **'Gentle / Boozy'** dimension than in the **'Sweet / Dry Sour'** dimension further supports the idea that most cocktails lean towards the **'Boozy'** side rather than being predominantly **'Sweet' or 'Sour'**.

Standard Deviation:

The standard deviation measures the amount of variation or dispersion in a set of values. A low standard deviation indicates that the values tend to be close to the mean, while a high standard deviation suggests that the values are spread out over a wider range.

The higher standard deviation in the **'Gentle / Boozy'** dimension indicates there's a greater variability in how 'Boozy' cocktails are, compared to the variability in how 'Sweet' or 'Sour' they are. This might imply that there's a wider range of alcohol strengths among cocktails than there is of sweetness or sourness levels.

Imbalanced Information:

The analysis might indicate we have imbalanced information, particularly for more **'Gentle' and 'Sweet'** cocktails. This could be due to several reasons, such as the prevalence of 'Boozy' cocktails, a potential bias in the data collection process, or even cultural preferences for stronger drinks. This is an important consideration for our machine learning model, as imbalanced data can lead to a biased model that does not perform well across all types of cocktails.

There are several strategies to handle this, such as resampling the dataset, generating synthetic samples, or using different evaluation metrics that are less sensitive to imbalance. Understanding the nuances of our data will help us choose the most suitable approach.

Let us look at the plotted cocktails in term of their distribution:

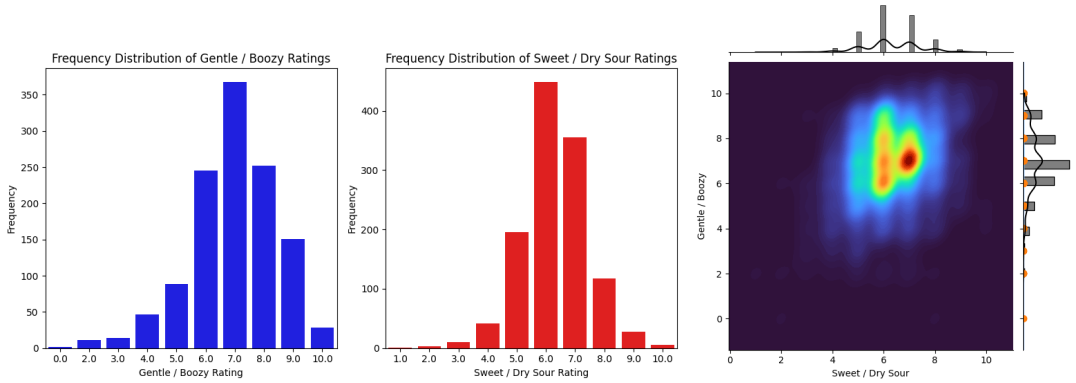


Fig. 2 - Another view at the cocktail distribution

Gentle / Boozy Dimension:

The Gentle / Boozy dimension describes how intense the alcohol taste is in a cocktail. A drink rated high on the **'Gentle'** side would be smooth and mellow, with the alcoholic content well-masked by other ingredients. Conversely, a high **'Boozy'** rating indicates a strong alcohol flavor, where the spirits are the star of the show.

The fact that the distribution is skewed towards **'Boozy'** indicates that a significant portion of the cocktails in the dataset have a more pronounced alcohol flavor. This could potentially be because many classic cocktails and mixology trends emphasize showcasing the spirits used.

The skewness in this distribution is important to remember when developing our models because it may impact the way the model learns to classify cocktails based on their **'Gentle / Boozy'** rating. For instance, a model trained on this data might be more likely to predict a 'Boozy' classification because it has seen more examples of 'Boozy' cocktails.

Sweet / Dry Sour Dimension:

The **Sweet / Dry Sour dimension** indicates the balance of sweetness and sourness in a cocktail. A high 'Sweet' rating suggests that the cocktail is sugary and less tart, while a high 'Dry Sour' rating means the cocktail leans towards acidity and lacks sweetness.

There are more cocktails with mid-range ratings in this dimension, which suggests that many cocktails achieve a balance between sweet and sour flavors. This could be because a well-balanced sweet-and-sour profile is a cornerstone of good cocktail making.

However, as there's **less skew** in this distribution compared to the **'Gentle / Boozy'** dimension, it could potentially mean that the model might have a more balanced learning from the 'Sweet / Dry Sour' ratings.

A **balanced dataset does not inherently mean the model's predictions will be more accurate**. It just means the model is less likely to favor one classification over the other due to the frequency of instances in the training data. We'll still need to carefully tune and validate our model to ensure it is making accurate predictions.

In both cases, **understanding these trends and distributions** in our data is crucial. It helps guide us during the model selection and tuning processes, as we can be aware of potential biases and take steps to address them. For instance, we could use techniques like oversampling or undersampling to balance our dataset if necessary, or adjust the class weights in our model to account for any imbalance.

Now let us look at the correlation matrix:

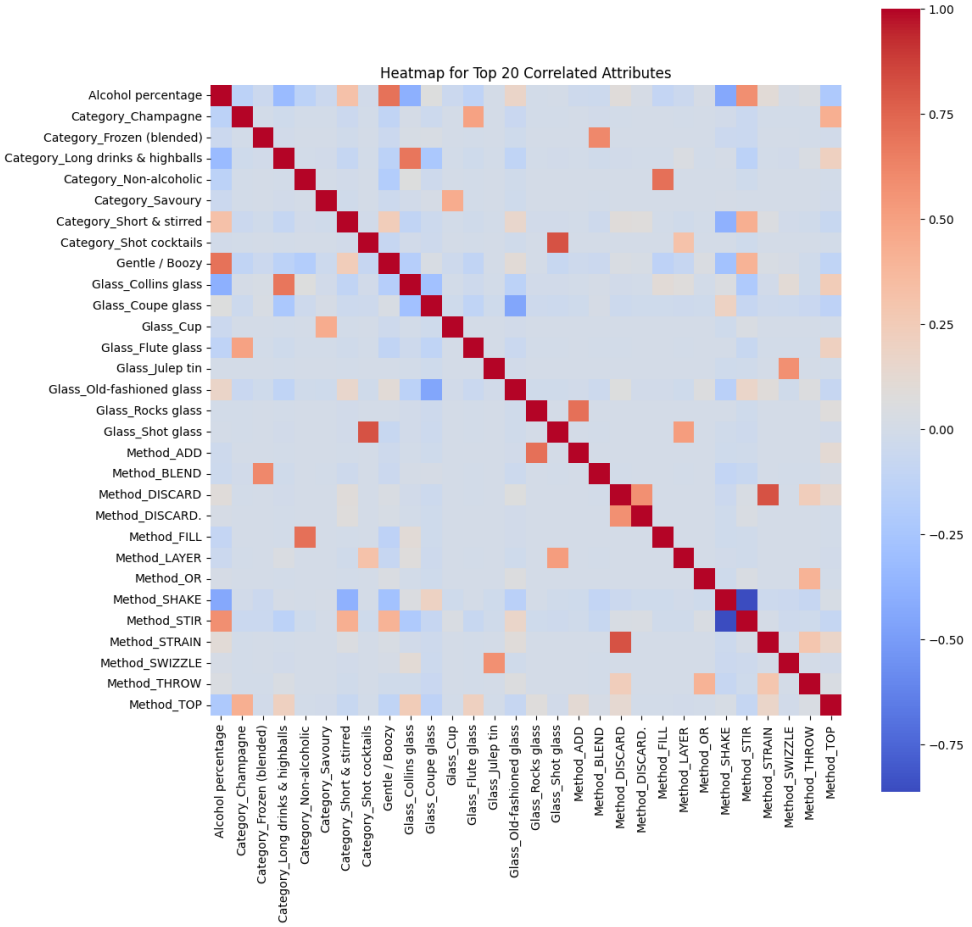


Fig. 3 - Correlation matrix for the Top 20 most correlated attributes

- **A Stirred cocktail is not Shaken**
  - These are common methods used to mix cocktails. Shaking and stirring are two different ways of combining ingredients. James Bond's famous line, "shaken, not stirred", underscores this difference. Shaking is a more aggressive technique and is usually used when a cocktail includes ingredients like fruit juices or cream, which need more force to mix evenly. Stirring, on the other hand, is a gentler method used when the cocktail mainly consists of alcoholic ingredients, preserving the clarity and subtlety of the drink. The correlation matrix confirms that these two methods are mutually exclusive in most cases.

- **You have to discard what you have strained**
  - Straining helps to remove solid ingredients (like fruit or ice) that were used in the mixing process but are not intended to be in the final drink. Hence, you "discard" the strained solids to present a clear and smooth cocktail. This is a common step in many cocktail recipes.

- **You mostly drink a shot in a shot glass**
  - This finding is fairly self-explanatory. A shot is a small, concentrated serving of alcohol, typically served in a small glass known as a shot glass. The data confirms this standard practice in the cocktail and spirits world.

- **The Gentle / Boozy axis is highly correlated with alcohol percentage, but 'only' at 0.7**
    - While a correlation of 0.7 might suggest a strong relationship, it is important to remember that correlation does not imply causation. The 'Gentle / Boozy' rating likely takes into account factors beyond just alcohol content, such as the mix of flavors in the cocktail, which can mask or highlight the alcohol's potency. Also, a correlation of 0.7, while high, still leaves room for other factors to have an impact on the 'Gentle / Boozy' scale.

- **The Gentle / Boozy axis is positively correlated with Stirred cocktails**
    - Stirred cocktails are usually made with spirits only, and therefore, tend to be stronger (or 'Boozier') as they do not contain juices or other non-alcoholic ingredients to dilute the alcohol content. Hence, it makes sense that there would be a positive correlation between the 'Gentle / Boozy' axis and stirred cocktails.

Now, moving onto the Category Distribution, we need to examine the different categories of cocktails in our data. Categories could be based on the primary alcohol, the method of preparation, the type of glassware used, or even the occasion for the drink (like aperitif or digestif). This kind of distribution can provide insights into the variety of cocktails, popularity of certain types, and potentially highlight any skewness in our data that we need to account for during modeling.

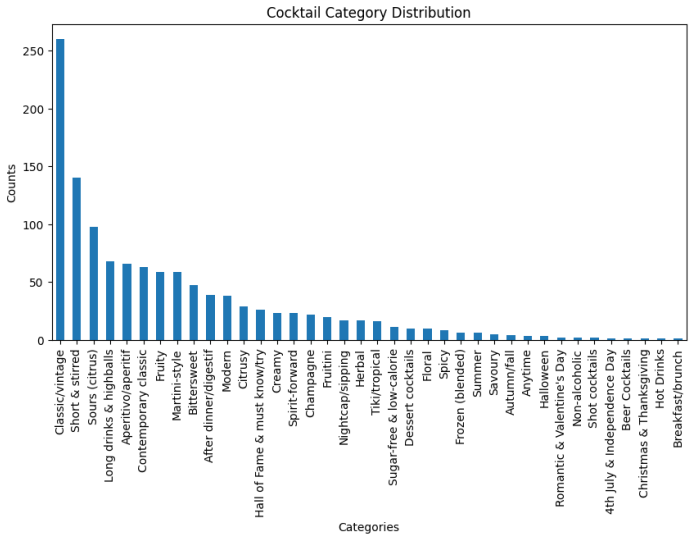


Fig. 4 - Category distribution

The predominance of **Classic / Vintage cocktails** in the dataset is not surprising. Classic cocktails like the Martini, Old Fashioned, and Negroni have stood the test of time and continue to be popular. The Difford's Guide is known for its extensive coverage of traditional drinks, providing detailed recipes and historical context. These cocktails often form the foundation for many bartenders and cocktail enthusiasts, and understanding them can provide insights into the basic principles of mixology.

As previously mentioned, **Short & Stirred cocktails** tend to be stronger ("Boozier") as they are usually made solely with spirits and are served without diluting mixers. The popularity of this category is a testament to the appreciation for potent, spirit-forward cocktails. They require less preparation compared to some other categories, which can also make them more approachable for home mixologists.

The prevalence of sours in the dataset is indicative of the widespread appeal of this cocktail category. **Sours** refer to a family of cocktails that contain a base liquor, a sour element (often citrus juice), and a sweetener. Examples include the Whiskey Sour and Margarita. The balance of sweet and sour elements in these cocktails creates a versatile and refreshing flavor profile that appeals to a wide range of palates.

Long drinks are typically served in tall glasses and are more voluminous due to the use of non-alcoholic mixers. They are often refreshing and easy-drinking, making them popular choices in various settings, from casual get-togethers to more formal events. Their representation in the dataset highlights the diversity of cocktail styles.

Aperitif cocktails are typically served before a meal to stimulate the appetite. They are usually dry and relatively low in alcohol. Common examples include the Negroni or a Dry Martini. The presence of aperitif cocktails in the dataset underscores the role cocktails can play in dining traditions and culinary experiences.

It is important to remember that these categories can intersect and overlap; a cocktail can be both a sour and a classic, for example. Each category represents a different approach to balancing the components of a cocktail, and understanding these categories can help us to appreciate the complexity and variety inherent in the world of mixology.

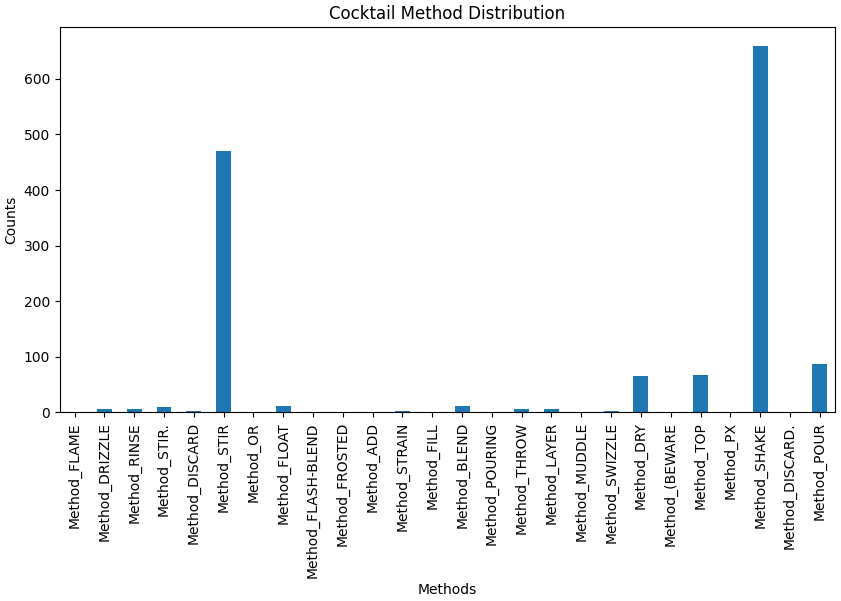


Fig. 5 - Method Distribution

Stirring is a method used primarily for spirit-forward cocktails, where all the ingredients are of similar density. The goal of stirring is to chill, dilute and mix the ingredients together without introducing too much air into the mix. This helps maintain the clarity and smooth texture of the drink. it is commonly used for cocktails like the Martini or the Negroni.

Shaking is used when a cocktail contains a mix of ingredients with different densities, like juice, cream, or egg whites, and you want to fully integrate them. Shaking introduces more air, chill and dilution into a cocktail than stirring, and results in a frothy, cloudy, well-mixed drink. Examples of shaken cocktails include the Margarita or the Whiskey Sour.

The **Pouring** technique involves directly pouring the ingredients into the glass in which the cocktail will be served, either over ice or into a previously chilled glass. This method is usually used for simple cocktails that do not require a lot of mixing or when layering of ingredients is desired.

In cocktail terminology, "**dry**" can refer to a few things: the lack of sweetness in a drink, a method of shaking a cocktail with no ice (a "dry shake"), or referring to using "dry" vermouth. In this context, it refers to the former.

Topping a cocktail usually refers to the addition of a small quantity of an ingredient after the cocktail has been prepared. This could be a splash of soda or tonic, a dash of bitters, or a garnish added at the end. This method is used to add the finishing touch to a cocktail and can dramatically influence the overall flavor and presentation of the drink.

Understanding these various methods is essential as they can significantly impact the texture, temperature, and flavor balance of the cocktail. Each method brings out different qualities from the ingredients, contributing to the overall experience of the drink.

Let us look at the most used ingredients:

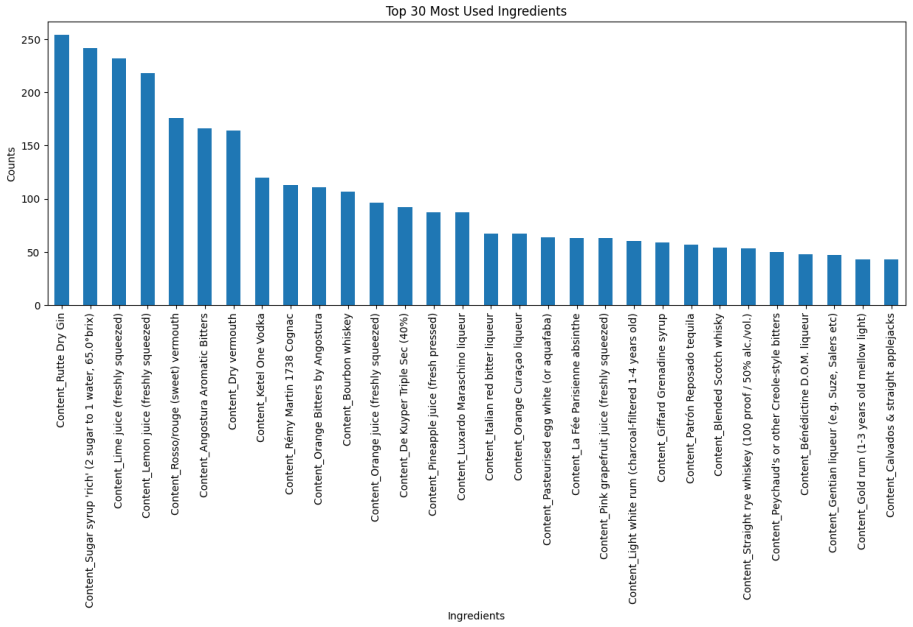


Fig. 6 - Ingredient distribution.

Many of the alcohol names do include brand names. This provides additional detail about the specific flavor profiles desired in a cocktail recipe, as different brands of the same type of alcohol can have distinct flavors. For example, one brand of gin might have a more pronounced juniper flavor, while another might have a more balanced botanical flavor. This specificity can help users recreate the cocktail as accurately as possible, this is not the case here as it seems that the Guide has special contracts with famous liquor brands to only offer one brand per alcohol.

As expected in any culinary field, there are staple ingredients that form the backbone of many recipes. In the case of cocktails, ingredients like lime juice, lemon juice, sugar syrup, and various forms of

alcohol are frequently used. They provide the basic structure of many cocktails - acidity from the citrus, sweetness from the syrup, and of course, the alcohol base.

The popularity of **lime and lemon juices** in cocktails is quite notable. Citrus is a crucial component in many cocktails as it provides acidity to balance the sweetness and bitterness of other ingredients. The freshness and tang of citrus also help to enhance the flavors of the alcohol.

it is worth noting that many **Classic and Vintage** cocktails rely on simple, tried-and-true combinations of ingredients. These cocktails have stood the test of time and are enjoyed by many, which explains their prevalence in the data.

The prominence of the **Sour** category in the cocktail world is in line with the frequent use of lime and lemon juices. Sour cocktails are based on the principle of balance between the base spirit, the sweetness (usually from a syrup), and the sour (usually from citrus). This balance makes them very popular and versatile, leading to a wide range of variations.

This data can provide insights into what makes a cocktail popular or classic, and can also guide the creation of new cocktail recipes, by understanding what ingredients and ratios tend to be preferred.

To finish this EDA, let us look at the Correlation Matrix between ingredients:

1	Content_Celery saccharum (celery syrup)	Content_Rutte Celery Dry Gin	0.749169
2	Content_Ketel One Vodka (from freezer)	Content_Cranberry juice (sweetened) (chilled)	0.706814
3	Content_Apple sugar syrup	Content_Kombucha	0.706814
4	Content_Savoia Americano Rosso	Content_Rose sugar syrup	0.706814
5	Content_Ketel One Vodka (from freezer)	Content_Giffard Fraise des Bois liqueur	0.706814
6	Content_Bigallet Genepi Grand Tetras	Content_Celery saccharum (celery syrup)	0.706520
7	Content_Cola (e.g. Coca Cola or Pepsi Cola)	Content_Thomas Henry Ginger Ale	0.692511
8	Content_Angostura Aromatic Bitters (optional)	Content_Sugar-free sweetener	0.629551
9	Content_St. George Spiced Pear liqueur	Content_Pear juice (freshly pressed)	0.576871

Fig. 7 - Top 9 Ingredient correlation.

Celery Syrup, Celery Dry Gin and Genepi: This combination is a great example of flavor pairing and enhancement. Celery syrup, with its fresh, light, and slightly savory flavor, pairs well with the botanical notes of gin. In this case, Celery Dry Gin is used, which likely has even more pronounced celery flavor notes. Genepi is an herbal liqueur known for its slightly bitter, complex botanical flavor, which adds an extra layer of depth to this combination. In cocktails, it is common to pair ingredients with similar flavor profiles to enhance and complement each other.

Vodka, Cranberry Juice, and Liqueur de Fraise des bois : This combination is characteristic of many sweet, fruity cocktails. Vodka is often used in these types of cocktails for its neutral flavor that allows the other ingredients to shine. Cranberry juice provides a balance of tartness and sweetness, while the Liqueur de Fraise des bois adds a rich, sweet strawberry flavor. The popularity of this combination may be due to the overall balance of flavors and the general appeal of sweet, fruity cocktails.

Sugar Syrup and Kombucha: Sugar syrup, with its sweet, straightforward flavor, is often used in cocktails to balance out more potent or sour flavors. Kombucha, on the other hand, is a fermented tea drink with a distinct sour and slightly sweet flavor, often with fruity or floral undertones depending on the type of tea and any added flavorings. The pairing of sugar syrup and kombucha is interesting as it

points to a trend in more health-conscious or unique cocktails. Kombucha, being a source of probiotics, adds a healthful twist and a unique sour flavor to cocktails, which is balanced out by the sweetness of the sugar syrup.

By understanding these ingredient pairings, we can better understand how flavors are combined in cocktail creation and how to create a balanced, tasty cocktail.

4. Methodology

Throughout the study, we employed in a sequential manner, following a well-defined pipeline. The project involved both supervised and unsupervised learning methods to build the cocktail recommendation system. The main methodologies employed were Random Forests and XGBoost for supervised learning, while for unsupervised learning, we used Variational Autoencoder (VAE) and Cosine Similarity based k-Nearest Neighbors (KNN).

For each model, we created a function that used our trained model to recommend cocktails based on input cocktails. The function accepts a list of cocktails, retrieves their feature values (specific properties or characteristics used by the model to make predictions) from the scaled dataframe (data that has been standardized to aid in model performance), and predicts the clusters (output that groups cocktails based on their similarity) for each cocktail. The function then recommends cocktails from the same clusters. This provides users with recommendations for cocktails that share similar characteristics to their inputs.

Supervised Learning

4.1. Random Forest

This machine learning method operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes of the individual trees for classification tasks.

It is a popular method because it is good at avoiding overfitting, where the model is too tailored to the training data and performs poorly with new data which is here perfect for our massive data that this model manages well.

We created a Random Forest Classifier with, arbitrarily, 500 decision trees ensuring a robust model. We trained our classifier on the training data, teaching it to understand the patterns and relationships in the cocktail features and then was used to do predictions on the test set in order to recommend cocktails.

4.2. Extreme Gradient Boosting

Another supervised learning method employed in our project was the Extreme Gradient Boosting, or XGBoost. This method is a decision-tree-based ensemble machine learning algorithm that uses a

gradient boosting framework. It is renowned for its speed and performance, as well as its capability to handle large-scale data.

In the context of our project, we chose the XGBoost over its counterparts for its performance, efficiency, and its capability to provide a better predictive accuracy with a fewer number of tuning parameters. Similar to Random Forest, XGBoost builds trees, but in a sequential manner, where each new tree helps to correct errors made by the previously trained tree. This way, it boosts the performance of our model in predicting the labels for cocktails, thereby enhancing the accuracy of our cocktail recommendation system.

Initially, we simplified our cocktail data, reducing a broad range of features down to three principal ones retaining all the rich information. Following this, we employed a method called KMeans clustering, which allowed us to group cocktails into 'clusters' or families based on their inherent similarities. To ascertain the optimal number of these groups, we used a scoring system that quantifies how well the cocktails fit into their respective clusters. Once we had our optimal clusters, we utilized the XGBoost technique to train the system to understand and distinguish between these groups based on the simplified cocktail features. Lastly, we devised our function that uses this learned information to recommend cocktails.

As a note, the specific parameters used in our XGBoost model were adjusted based on the characteristics of our dataset and the performance requirements of our recommendation system. As with any machine learning project, the parameters chosen for a model are crucial in ensuring its accuracy and reliability. The parameters for our model were carefully chosen after several iterations and tests, ensuring the optimal balance between overfitting and underfitting, thereby delivering accurate and reliable cocktail recommendations.

We also tried a XGBoost Classifier to train our data in order to observe different results. Those results are displayed in the appendix.

We wanted to give a try to mix the two models of supervised learning which is not included in here due to its poor performance but is good to be mentioned. Those results are also displayed in the appendix.

Unsupervised Learning

4.3. Variational Autoencoder

The Variational Autoencoder (VAE) method we used in our study is a type of autoencoder, a neural network that learns to compactly represent data in a way that can be reversed, recreating the original input. VAEs, in particular, make the compact representation (latent space) structured and continuous, making them useful for generating new, similar data.

Variational Autoencoders (VAEs) have proven to be an excellent choice for our cocktail recommendation system for several reasons. In our project, we are dealing with high-dimensional data—each cocktail has many features, such as ingredients and their proportions. VAEs shine in this environment as they compress these high-dimensional data into a lower-dimensional latent space, while preserving the essential information about the data. This latent space serves as an organized, structured map of our cocktail data, where similar cocktails are placed close to each other.

Moreover, the VAE model's capability to generate new, unseen data is beneficial for our project. It provides the opportunity for the system to suggest creative and novel cocktails based on the learned characteristics from the training data, enriching the user's experience. By capturing the underlying patterns and relations between the features, VAE aids in tailoring more personalized and innovative recommendations, giving our system an edge over traditional methods.
The training of VAEs also involves ensuring that the encoded data is similar to a standard normal distribution. This property is advantageous as it allows us to leverage statistical methods to analyze and manipulate the data further if required. In summary, VAE's capacity for data compression, generation, and its statistical properties align perfectly with the needs of our cocktail recommendation project.

In our project, we first determined the size of the latent space—1024 in this case—representing the dimensions in which our data will be compressed.

The first part of our VAE is the encoder. This neural network takes the scaled cocktail data and compresses it into the latent space. We applied a statistical trick called reparameterization to sample from the distribution without affecting the network's differentiability, a vital property for training our model.

The second part of our VAE is the decoder, another neural network that takes the points in the latent space and reconstructs the original data from them. It takes as input the compressed data from the encoder and tries to recreate the original input.

For training the VAE, we defined a special loss function that takes into account both the quality of our reconstructions and how closely the learned distributions align with a standard normal distribution (a process called KL divergence).

Once the VAE was trained, we used it to compress our scaled cocktail data into the latent space, giving us a new, structured representation of our cocktail data.
Finally, we used a nearest neighbor search in this latent space to find similar cocktails to a given input thanks to our recommendation function.

4.4. Cosine Similarity

In this project, another approach we used for recommending cocktails is the Cosine Similarity based k-Nearest Neighbors (KNN). The concept of cosine similarity is drawn from vector space, where items are treated as multidimensional vectors. For us, each dimension represents a feature of the cocktail. The cosine of the angle between two vectors (in our case, cocktails) is used as a measure of similarity.

Cosine Similarity and k-Nearest Neighbors (KNN) method presents a suitable fit for our cocktail recommendation project due to its feature-based approach to similarity, which is key in recommendation systems. This method calculates the similarity between different cocktails based on their features, implying that cocktails with high similarity share many common features, thus are likely to be enjoyed by the same individual. Therefore, Cosine Similarity and KNN allow us to provide targeted, relevant, and reliable cocktail recommendations that are specifically aligned with the unique taste profiles of the input cocktails. By deploying this method, we are ensuring that our recommendations not only make intuitive sense but are also data-driven, accurate, and highly personalized.

We computed cosine similarities among all cocktails, essentially creating a large similarity matrix that maps how closely each cocktail resembles every other cocktail in our dataset. This matrix served as the foundation for our KNN model.

Next, we used k-Nearest Neighbors (KNN) to generate recommendations. KNN is an algorithm that finds the 'k' nearest points (in our case, cocktails) to a given point (the input cocktail). This "nearness" is measured based on the cosine similarity we calculated earlier. When an input cocktail is given, our KNN model looks up its 'k' most similar cocktails from the similarity matrix and suggests those as recommendations. The notion of "nearest" here is conceptually akin to the input cocktail and the recommended cocktails sharing many common features.

The Cosine Similarity and KNN approach provides robust results for our task because it works based on the principle of feature similarity, which is a natural fit for recommendation systems. It presumes that if two cocktails share a high degree of similarity (i.e., they have similar features), they are likely to be enjoyed by the same person, thereby making them good recommendations for each other. By utilizing this methodology, we can generate meaningful and reliable recommendations that are tailored to the unique features of each input cocktail.

Software and tools

The analysis for this project was carried out using a variety of software tools and libraries designed for data analysis and machine learning. The main programming language used throughout this project was Python due to its extensive support for scientific computing and machine learning libraries.

For web Scraping:

requests: Used for making HTTP requests in Python.
BeautifulSoup: Used for parsing HTML and XML documents and web scraping.

For Data Manipulation and Analysis:

pandas: Used for data manipulation and analysis.
numpy: Used for mathematical operations on large, multi-dimensional arrays and matrices.

For Visualization:

matplotlib.**pyplot**: Used for creating static, animated, and interactive visualizations in Python.
seaborn: Based on matplotlib, used for creating more attractive and informative statistical graphics.
matplotlib.patches: Used for creating shapes that can be added to figures.

For Parallel Computing:

concurrent.futures: Used for creating high-level interface for asynchronously executing callables.

For Machine Learning and Data Modeling:

- **sklearn**: A comprehensive library for machine learning and modeling, which includes:
- **GridSearchCV**: Used for hyperparameter tuning.
- **train_test_split**, **StratifiedKFold**: Used for splitting data into train and test sets.
- **MinMaxScaler**: Used for scaling features.
- **KMeans**: Used for KMeans clustering.
- **PCA**: Used for Principal Component Analysis.
- **silhouette_score**: Used for evaluating clustering performance.
- **TSNE**: Used for dimensionality reduction.
- **NearestNeighbors**: Used for implementing neighbor searches.
- **cosine_similarity**: Used for calculating cosine similarity.
- **RandomForestClassifier**: Used for the random forest algorithm.
- **accuracy_score**, **f1_score**: Used for evaluating model performance.
- **xgboost**, **XGBClassifier**: An implementation of gradient boosted decision trees designed for speed and performance.
- **tensorflow**: An open-source platform for machine learning, used here for creating the Variational Autoencoder.

For Utilities and Others:

- **os**: Used for interacting with the operating system.
- **time**: Used for time-related tasks.
- **re**: Used for working with Regular Expressions.
- **csv**: Used for reading and writing csv files.
- **random**: Used for generating pseudo-random numbers.
- **warnings**: Used for warning control.
- **fractions**: Used for working with rational numbers.
- **unicodedata**: Used for Unicode Database.
- **tabulate**: Used for creating pretty-printed tabular data.
- **collections**.Counter: Used for counting hashable objects.
- **eli5**: Used for debugging machine learning classifiers and explaining their predictions.

5. Results

Our five cocktails main inputs are : *Abbey*, *Hot tub*, *The Frank*, *Green Ghost*, *Honey Cosmopolitan* which share a lot in common. We added five more cocktails for our Random Forest model that needs more cocktails in order to output recommendations. The cocktails are : *Absinthe Sour*, *Absolutely Fabulous*, *Ace Of Clubs Daiquiri*, *Mr. Bali Hai*, *Achilles Heel*.
The recommendations were then computed into a two dimensional scatter plot based on two features : Gentle/Boozy and Sweet/Dry Sour. Blue dots represent the input cocktails while the orange dots are the recommended ones.

After the recommendations provided of each model based on those cocktails, we want to iterate the process a thousand times in order to have a better idea of the model's overall behavior and constantly see how the model recommends cocktails over others.

Based on those new recommendations, we then perform an EDA to understand the output of the models.

5.1. Random Forest

Cocktail recommandation (Basic)

Based on our 10 cocktails, the cocktails recommendations were : *Pago Pago Cocktail*, *Warsaw Cooler.*

On the scatter-plot below, we observe the 5 cocktails that we imputed in blue against the two cocktails that have been recommended to us.

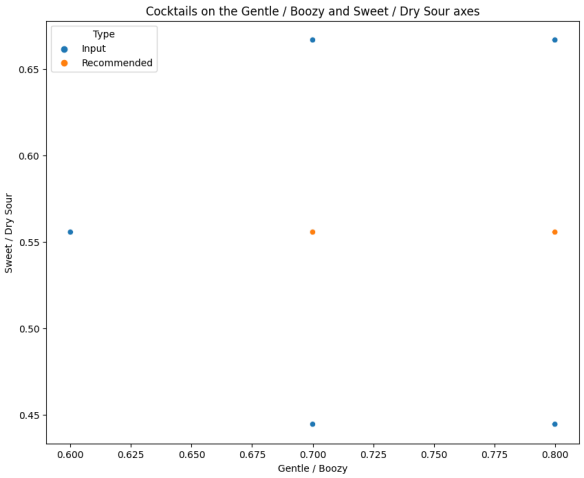


Fig. 8 - Scatterplot: Gentle/Boozy vs Sweet/Dry Sour

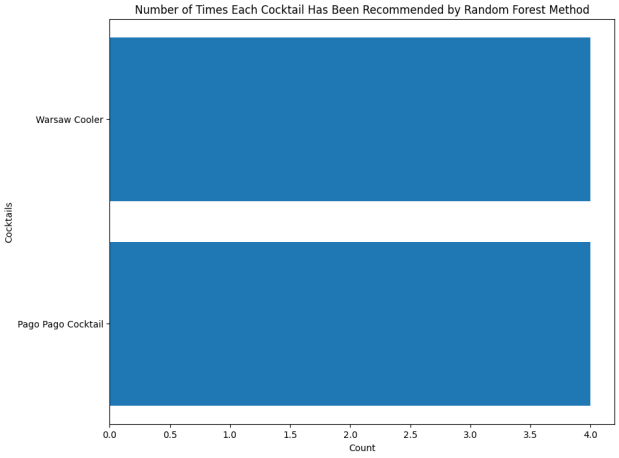
The recommended cocktails appear to have been computed by the mean of the of the cocktails in both axes, with their Sweet axis being equal to the far left cocktail. The resulting figure is a C shape for the input cocktails, and a straight line for the recommended. There is no outlier here, which is a good news for the fact that we only have 2 cocktails in the output.

Cocktail recommendation (1000 iterations)

Let us now iterate a thousand times. The output of the recommended cocktails appears to be :

Pago Pago Cocktail, Warsaw Cooler, Pago Pago Cocktail, Warsaw Cooler, Pago Pago Cocktail, Warsaw Cooler, Pago Pago Cocktail, Warsaw Cooler ...

In brief, we have the two same cocktails as before as it is displayed on the figure X.
The Figure X is a new scatter plot based on the top two predictors which are the percentage of alcohol of a cocktail and the popularity of this one.



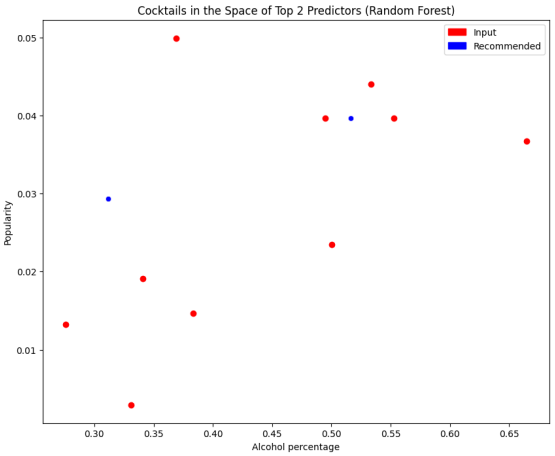


Fig. 9 - Cocktail distribution

Fig. 10 - Alcohol percentage vs popularity

After 1000 times, we can see that the cocktails are always the same, so we wanted to explore the underlying structure of the selection, which is more interesting than only being the mean of every cocktail in the Boozy / Sweet axes. We can see that the most important factors here are the popularity, and the alcohol percentage, which makes sense because some of the input cocktails are very famous among cocktail enthusiasts. One interesting thing here is that the model recommended a pretty average cocktail in terms of alcohol percentage, and a pretty low one, which is represented well by the lower value in the figure before.

5.2. Extreme Gradient Boosting

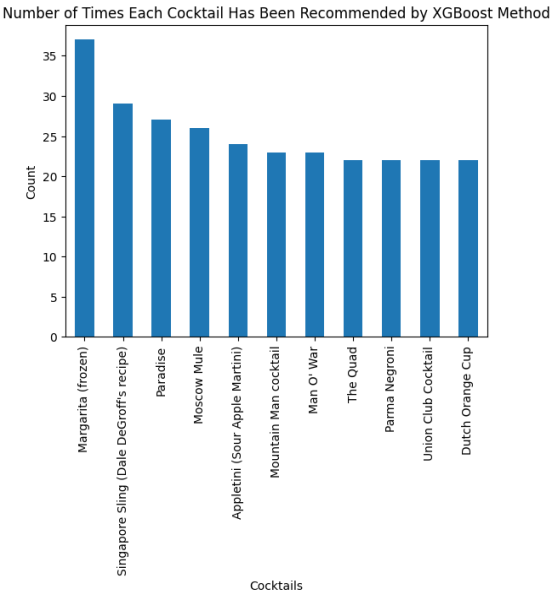


Fig. 11 - Cocktail Distribution (XGB)

After 1000 iterations, here is the distribution of all the cocktails. We can see that one cocktail is clearly going out of the lot (the Frozen Margarita) with 37 times, instead of around 25 times for the others in the Top Ten.

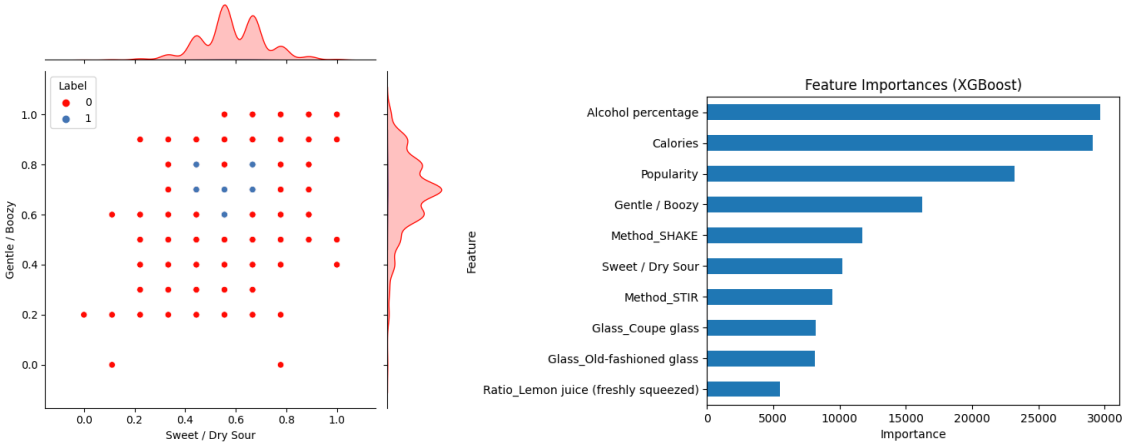


Fig. 12 - Cocktail Scatterplot & Feature Importance (XGB)

Regarding the scatterplot & distribution graph on the left, we can see two important phenomenons:

1. A lot of different cocktails were recommended, much more than any of the other methods presented in this project,
2. But the bulk of the cocktail distribution stayed around the input we gave the algorithm.
    a. We can see 2 spikes in the X axis where most cocktails were given as an input, and a big spike in the Y axis too.

Regarding the feature importance, we can see that the most important categories were almost the same as the ones from the Random Forest, which is not surprising to see.

5.3. Variational Autoencoder

Cocktail recommendation (Basic)

From our five cocktails input, the five cocktail recommendations for the VAE are :
Cazador, Blood, Smoke & Tears, London Fog, Vodka Sour (no added sugar & low-calorie), Green Isaac's Special.

To have a better representation, we plot again the input and the output in a scatter plot :

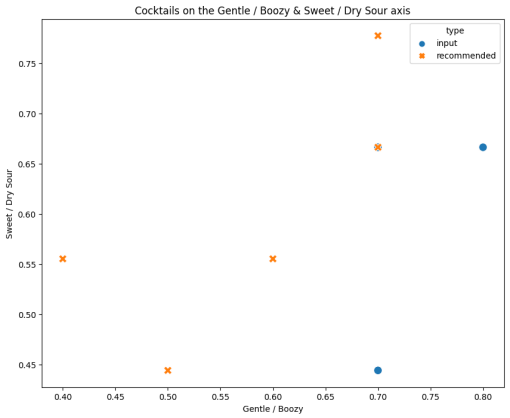


Fig. 13 - Cocktails in the Gentle / Boozy & Dry / Sour axis

Here we can see that the VAE did not choose the Boozy / Sweet axes as important, which is why the output seems all over the place. Let us see what happens if we use the Principal Component Analysis in order to have as axes the data that has the maximum information by reducing the dimensionality of the overall data:

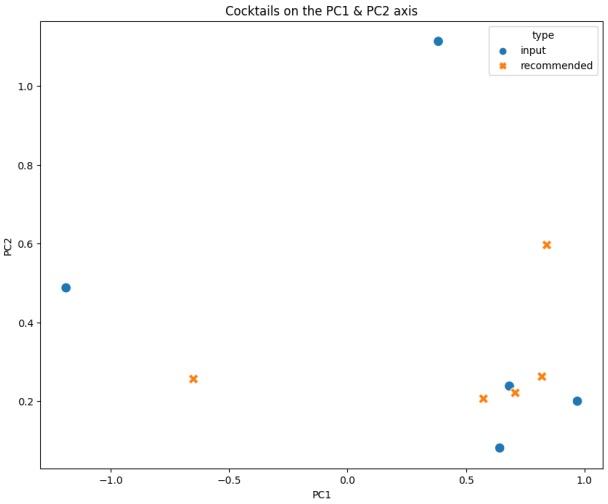


Fig. 14 - Cocktails in the PC1 / PC2 axis

Now the model makes much more sense, and we can see that the weighting of the above-mentioned axes in the PC1 & PC2 were not as high as other components, such as Popularity or Alcohol percentage.

Cocktail recommendation (1000 Iterations)

A lot of cocktails have been listed so we will display a bar plot to see the most selected cocktails and have an idea about the general model.

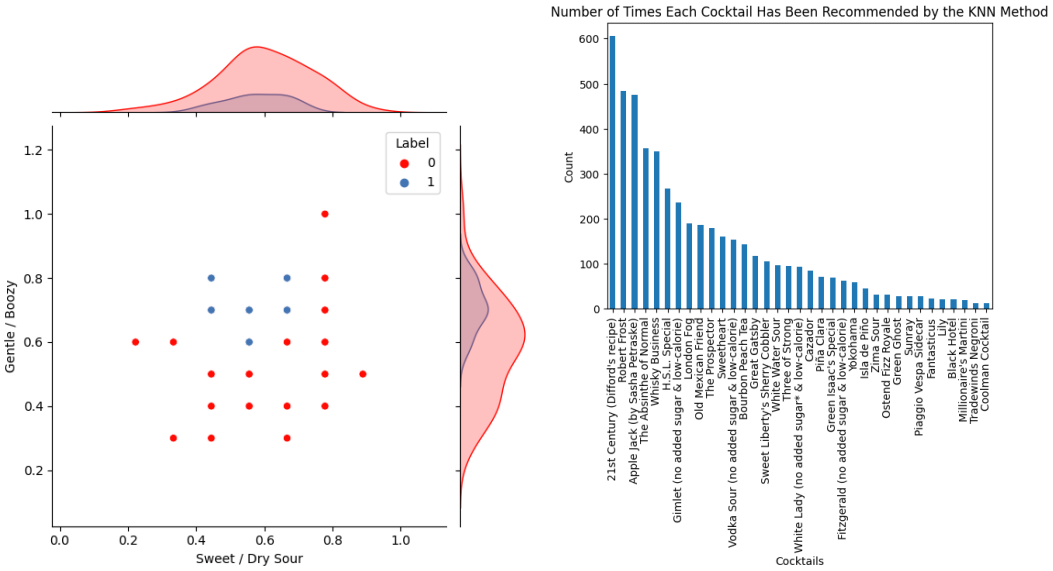


Fig. 15 - Gentle/Boozy vs Sweet/Dry Sour (Thousand Iteration)

For this part, we iterated through subsets of the input list to get this output, as the predictable nature of the VAE would not let us see the biases of the output. And we were pleasantly surprised to see that the Top five cocktails were present in more than 25% percent of all iterations, but this came with the drawback of a low diversity. Furthermore, the cocktails were not as concentrated around the input cocktails as in the ones before, which can lead to inappropriate recommendations.

5.4. Cosine Similarity

Cocktail recommendation (Basic)

From our five cocktails input, the Cosine Similarity recommends us twenty cocktails :
Blue Star, Income Tax Cocktail, Houla Houla Cocktail, Fibber McGee, Twinkle, King of Orange, Transylvanian Martini, Very French Martini, Mr President, Valentino, Celebration, Violet Affinity, London Cocktail, Mujer Verde, London Fog, Gimlet Cocktail (Difford's recipe), The Flirt, Lolita Margarita, Pink Grapefruit Margarita, Tailor Made

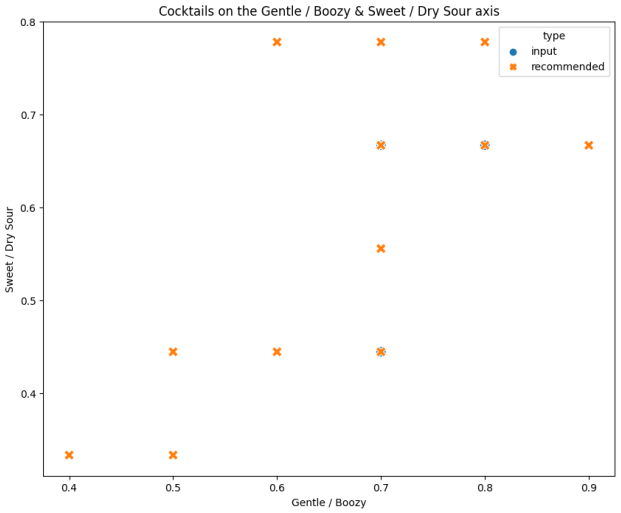


Fig. 16 - Gentle/Boozy vs Sweet/Dry Sour

For this model, we can see that the model found cocktails in a somewhat random pattern around the Boozy / Sweet axes, but some cocktails were found with the same values as the input, which is nice to see. Right now, we need to iterate 1000 times to see how the model really behaves.

Cocktail recommandation (1000 Iterations)

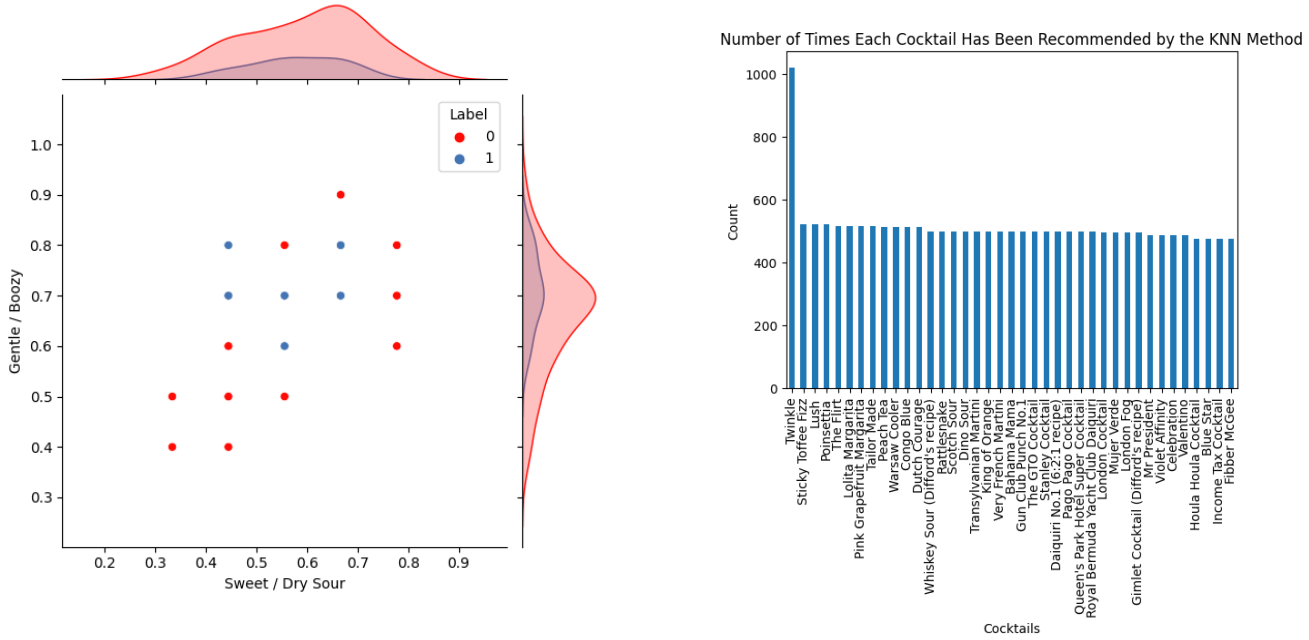


Fig. 17 - Gentle/Boozy vs Sweet/Dry Sour & Distribution

Like we did for the VAE model, we used subsets of the input to iterate 1000 times and get different results at each iteration. One surprising factor is that the Twinkle was recommended at each and every iteration, which may have arrived because it has exactly the same values as one of the input cocktails regarding the Boozy / Sweet axis. Also, most top 20 cocktails were recommended half the time, which is not really good considering that there is a lot more afterwards. Here the model seems to not have taken other components of the cocktails as seriously as the others before it, which is a problem because it is more prone to error due to the low granularity of the values in the axis on the left.

6. Recommandations

So, which model should be considered as the best?

After considering our four models, we consider taking the XGBoost as the best model in order to recommend cocktails based on the input.

The problem for our unsupervised learning is that it can easily go wrong when going through multiple times but appears to be correct most of the time when going through one because of its non-random and predictive aspect. On the other hand supervised learning can go wrong at the start while adjusting itself when going through multiple iterations which will give good predictive results. This is exactly the case for our XGBoosting method. The distribution behind this one is convenient as it recommends the right cocktails with the right amount of underlying weight of the other components of a cocktail as well as the Boozy / Sweet axis.

Random Forest is a bad model due to its capacity to recommend only two and the same two cocktails after a few iterations. It is a good model if our goal is to recommend a really small number of cocktails as this model understands the principal components very well.

Limitations

Our study faced some constraints and challenges due to the limited amount of time available, and certain aspects warrant further examination. For example, we did not have the opportunity to investigate how our models would perform when given a varied number of inputs or different cocktail names. Examining this could yield insightful information about how the recommendation system responds to diverse inputs. Additionally, as the number of inputs increases, computational cost may become a concern, raising questions about scalability. Understanding how the model scales with larger inputs could be an intriguing area of future work.

One of the significant challenges we encountered is the interpretability of the results. Typically, in a classification or regression problem, performance can be quantified using metrics like accuracy, AUC-ROC, sensitivity, etc. However, our task was unique - we needed to recommend cocktails based on certain features, which essentially falls under the umbrella of unsupervised learning. Standard metrics of success don't apply in the same way here. One potential future direction could be developing new ways to assess the effectiveness of the recommendation system, possibly by integrating user feedback or employing other user satisfaction metrics.

Another potential limitation lies in the feature representation of cocktails. Currently, our approach represents cocktails based on selected features such as "Sweet / Dry Sour" and "Gentle / Boozy". Although these features provide a reasonable approximation of a cocktail's taste, they might not capture all the intricacies. For example, the unique flavors resulting from the interaction of individual ingredients are not represented in the current approach. An interesting avenue for further study could be to expand the feature set, possibly including more information about the ingredients and their proportions.

By addressing these limitations, we believe there's potential for substantial improvement and refinement of the current cocktail recommendation system.

7. References

Social Drinking | Responsible Drinking in Social Situations. (2023, April 11). Orlando Recovery Center. Retrieved June 12, 2023, from https://www.orlandorecovery.com/drug-addiction-resources/alcohol/social-drinking/

Yiu, T. (2019, June 12). *Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu*. Towards Data Science. Retrieved June 12, 2023, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2

8. Appendix

In this appendix, we will briefly show the results of two different extensions of the XGBoosing methods that we talked about briefly in our appendix.

XGBoosting classifier :

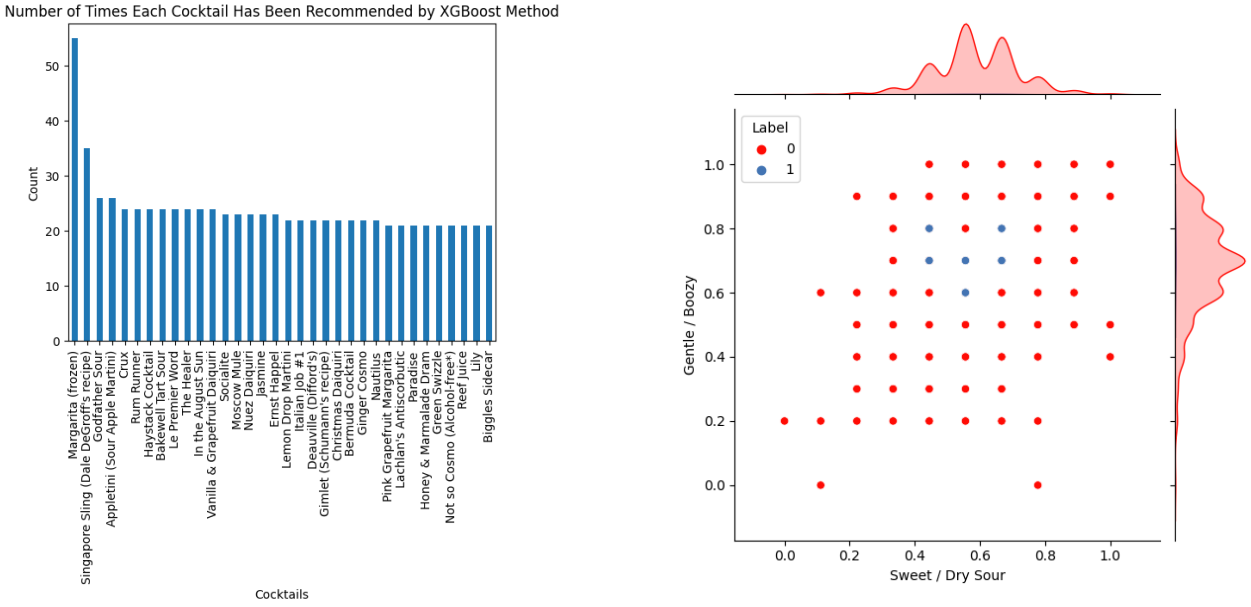


Fig.18 - Scatter Plot & Distribution

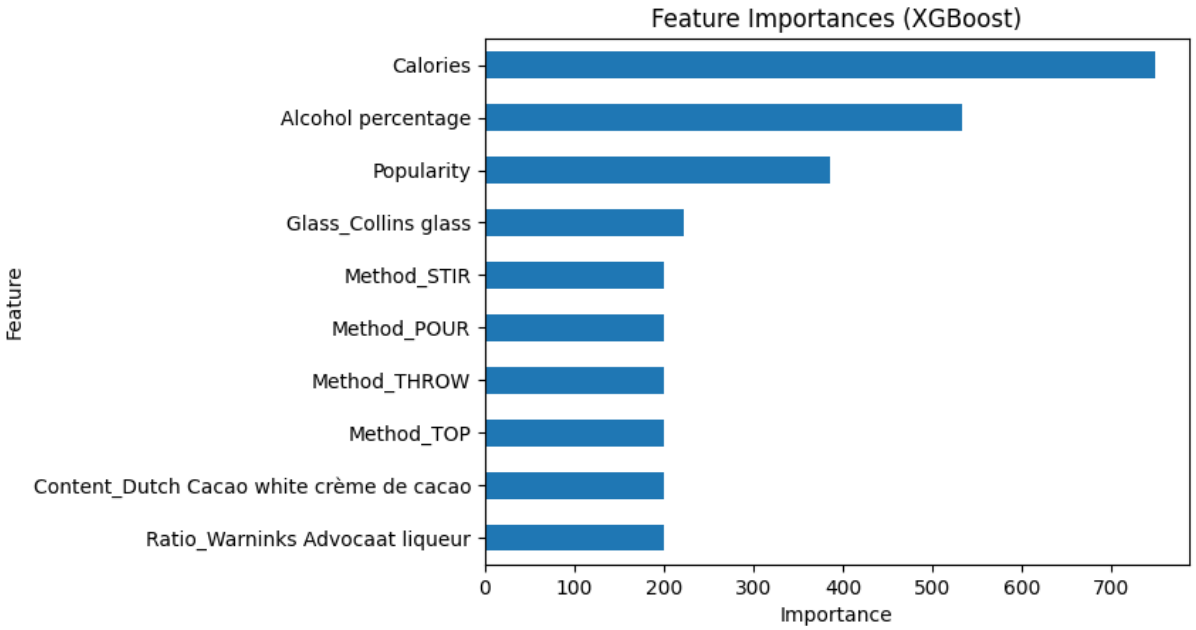


Fig. 19 - Features of Importances

Mix of random forest and XGboosting :

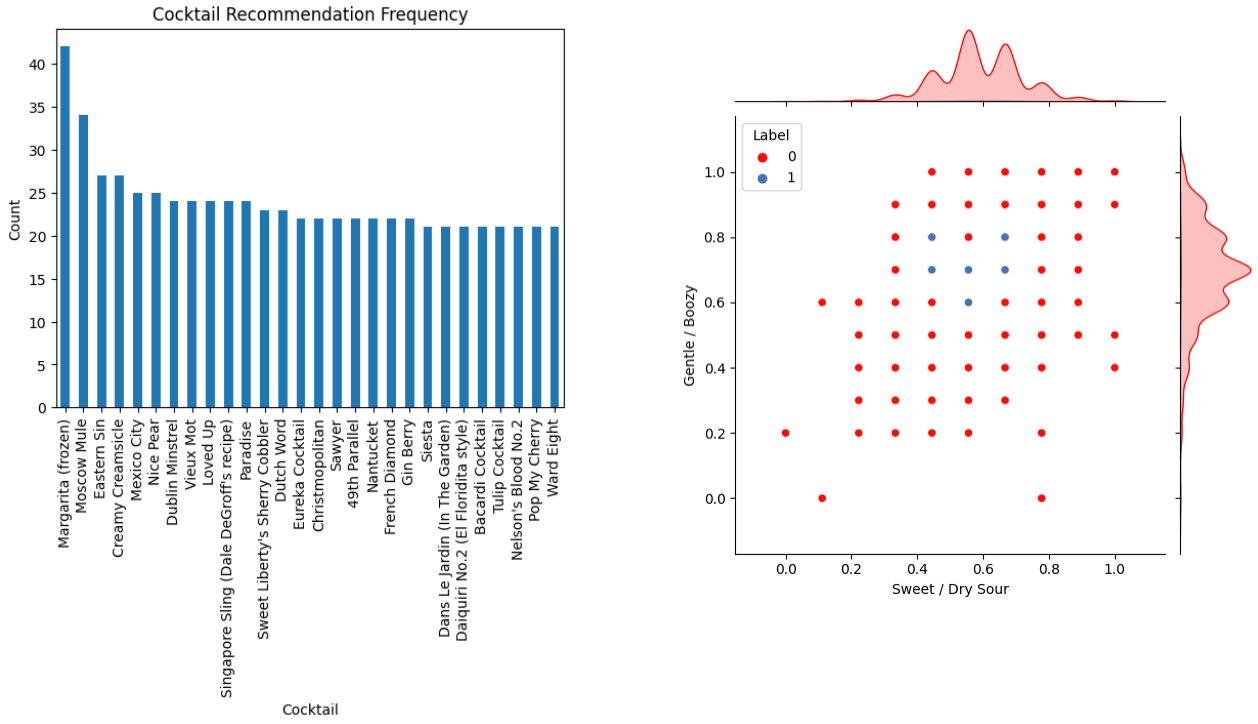


Fig. 20 - Scatter Plot & Distribution