Lecture 27: Case Study

The process

Harvard IACS

CS109A

Pavlos Protopapas, Kevin Rader, and Chris Tanner

ANNOUNCEMENTS

We realize this is a stressful and chaotic time for everyone, please lean on others for support, don't be shy to ask for help or to provide such to others, and be kind to yourself. I'm not just referring to 109 coursework.

Learning Objectives

- Feel prepared for the project
- Gain insights / learn considerations for solving a problem
- Feel prepared tackling the remaining course content / gain confidence!

Agenda

- Example project

How prepared do you feel for completing your course project?

DISCLAIMER:

This example project concerns predicting customers' ratings of bourbon (alcoholic drink).

Alcohol is drug. There are state and federal laws that govern the sale, distribution, and consumption of such. In the United States, those who consume alcohol must be at least 21 years of age. In no way am I, or anyone else at IACS or Harvard at large, promoting or encouraging the usage of alcohol. My intention is not to celebrate it. Anyone who chooses to consume alcohol should be of legal age and should do so responsibly. Abusing alcohol has serious, grave effects.

The point of this exercise is purely pedagogical, and it illustrates the wide range of tasks to which one can apply data science and machine learning. That is, I am focusing on a particular interest and demonstrating how it can be used to answer questions that one may be interested in for one's own personal life. You could easily imagine this being used in professional settings, too.

Step 1: formulate questions

1. Are there certain attributes of bourbons that are predictive of good bourbons? (i.e., highly rated by customers)

- Find hidden gems (i.e., should be good but current reviews are absent or unsupportive of such)

- Find over-hyped bourbons (i.e., the reviews seem high but the attributes aren't indicative)

- Are there significant results if we target experts' ratings instead of average customer ratings?

Step 1: formulate questions

2. Are there certain attributes of bourbons that are predictive of expensive bourbons?

- Find underpriced ones
- Find over-priced ones

Step 1: formulate questions

3. Which bourbons are most similar to each other?

- Which attributes are important for determining similarity? (e.g., does price play a role, or does similarity transcend price?)

Step 1: formulate questions

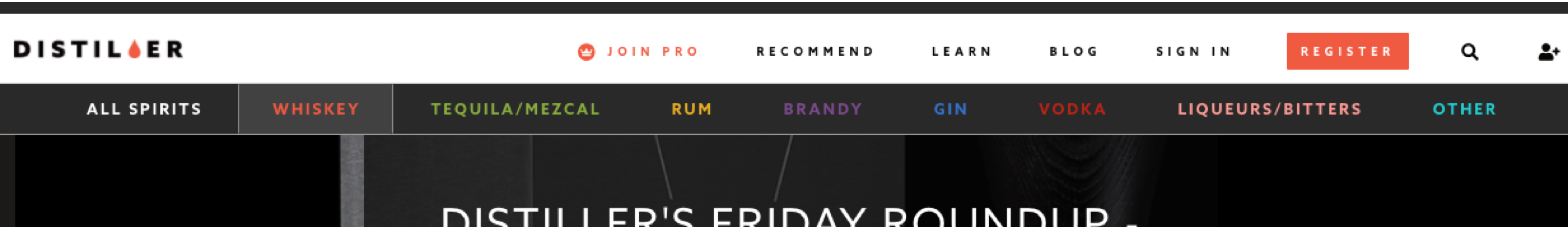
1. Are there certain attributes of bourbons that are predictive of good bourbons? (i.e., highly rated by customers)

- Find hidden gems (i.e., should be good but current reviews are absent or unsupportive of such)
- Find over-hyped bourbons (i.e., the reviews seem high but the attributes aren't indicative)
- Are there significant results if we target experts' ratings instead of average customer ratings?

Step 2: get the data

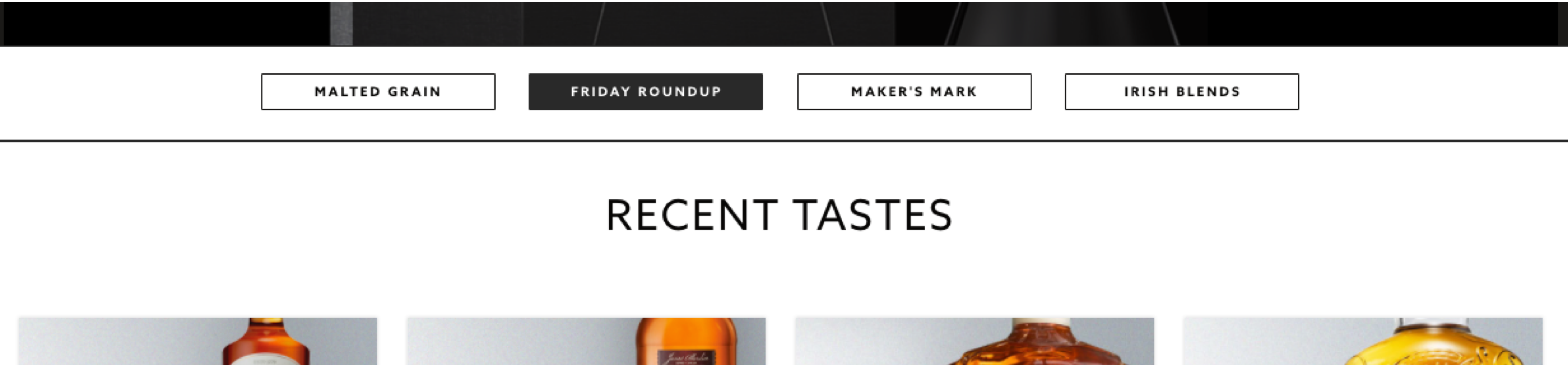
I found distiller.com which seems really comprehensive.

It has a community of reviews, an app, and even tries to make personalized recommendations. Surely we can make a better recommendation engine, though!



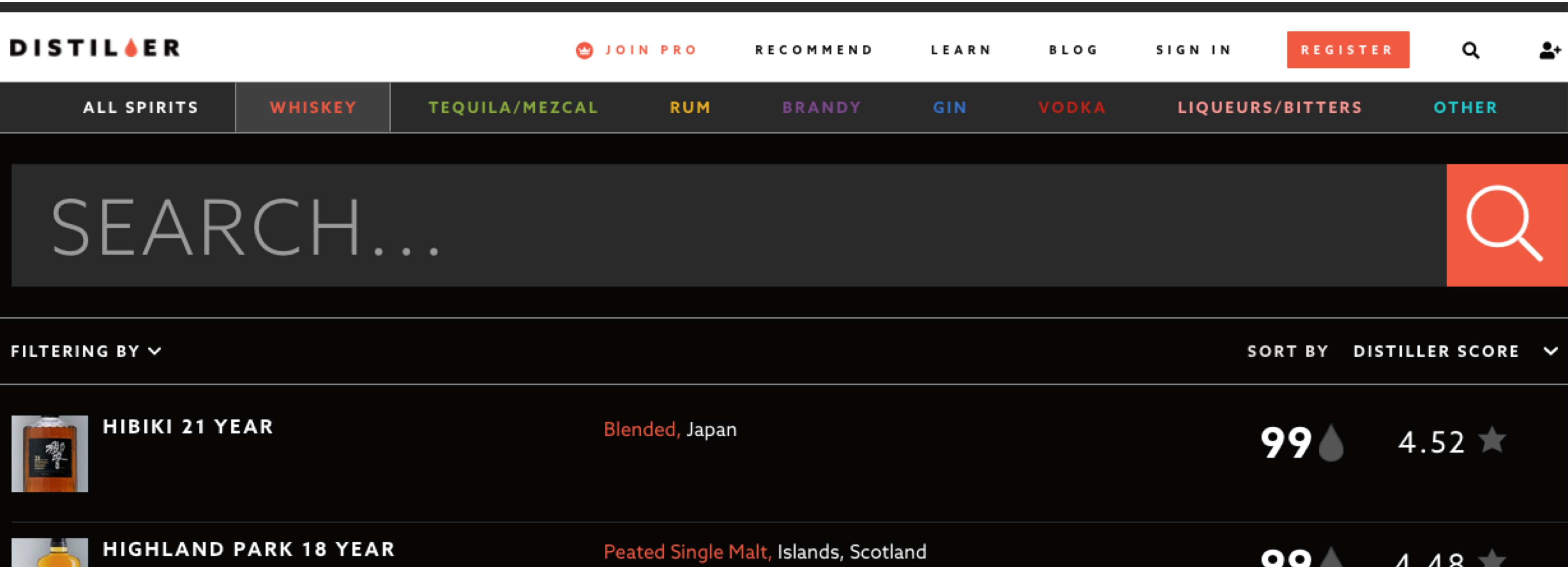
Step 2: get the data

Despite clicking the "whiskey" section, I couldn't find any listing of whiskeys, only an annoying "recent tastes" (from users) sample. If you click one, it would only show a few other related ones.

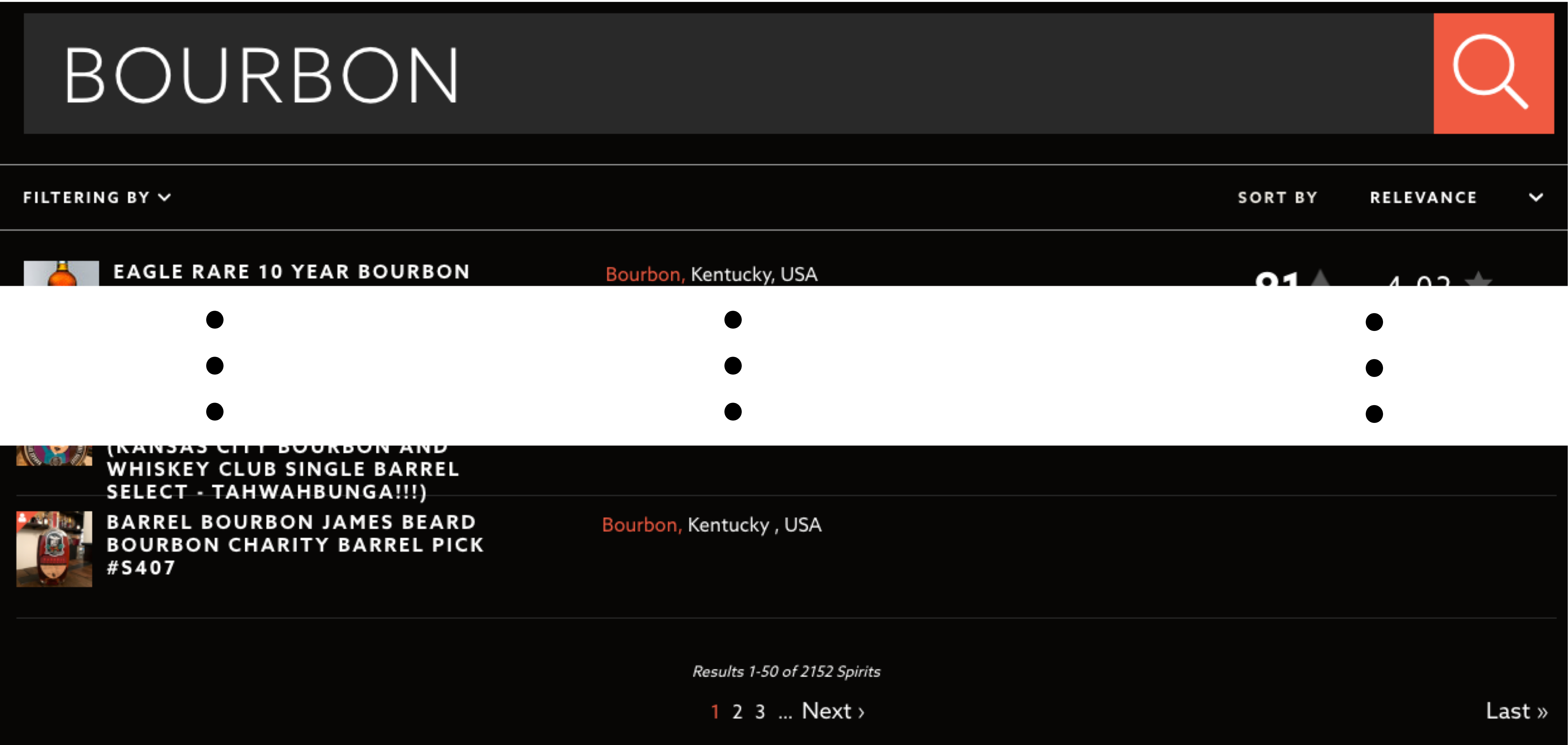


Step 2: get the data

But, there's a search bar, and if I search for "bourbon", I got tons of search results!

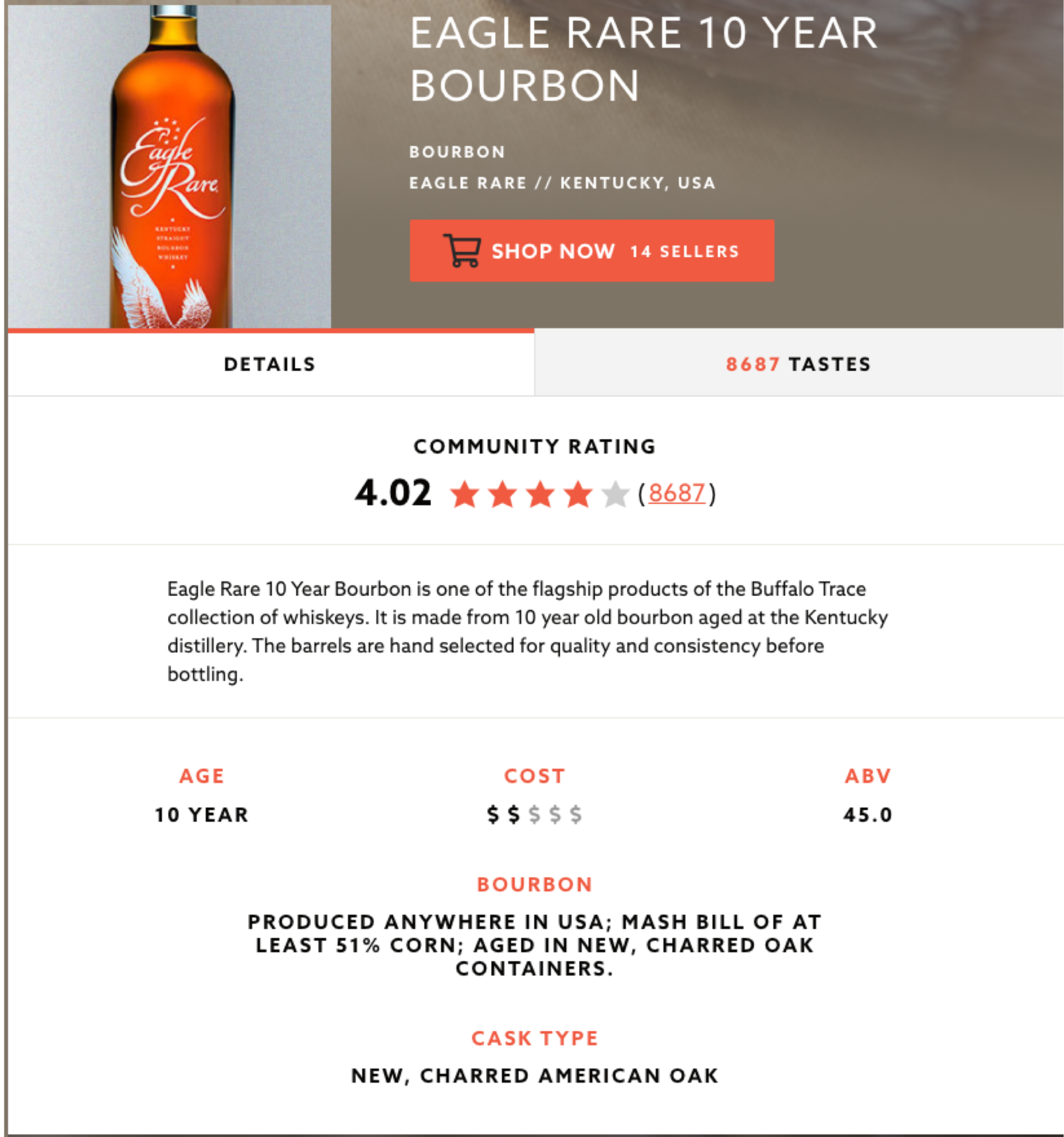


Step 2: get the data



Step 2: get the data

Top part of page



Step 2: get the data

Bottom part of page

What should we try to extract from each webpage?

Step 2: get the data

```
<div class='flavor-profile'>
<h5>Flavor Profile</h5>
<h3 class='secondary-headline flavors middleweight'>Sweet &amp;
Full Bodied</h3>
<canvas class='js-flavor-profile-chart' data-
flavors='{&quot;smoky&quot;:5,&quot;peaty&quot;:0,&quot;spicy&q
uot;:30,&quot;herbal&quot;:5,&quot;oily&quot;:0,&quot;full_bodi
ed&quot;:70,&quot;rich&quot;:65,&quot;sweet&quot;:80,&quot;brin
y&quot;:0,&quot;salty&quot;:0,&quot;vanilla&quot;:40,&quot;tart
&quot;:0,&quot;fruity&quot;:40,&quot;floral&quot;:10}'
height='250' width='900'></canvas>
</div>
```

Step 2: get the data

Do all webpages have this info?

Some are missing graphs?

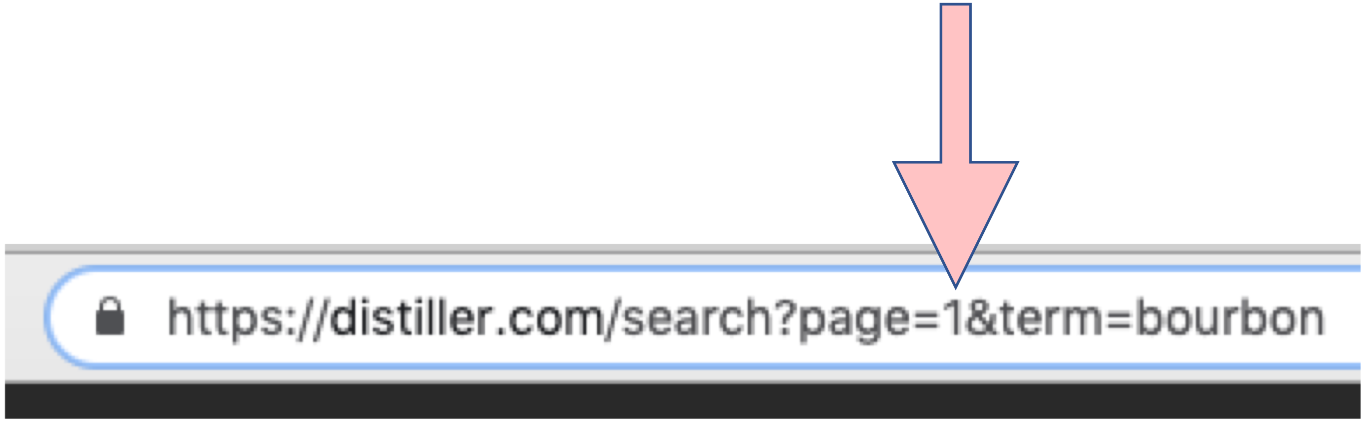
Missing Age statements?

Missing Customer Reviews?

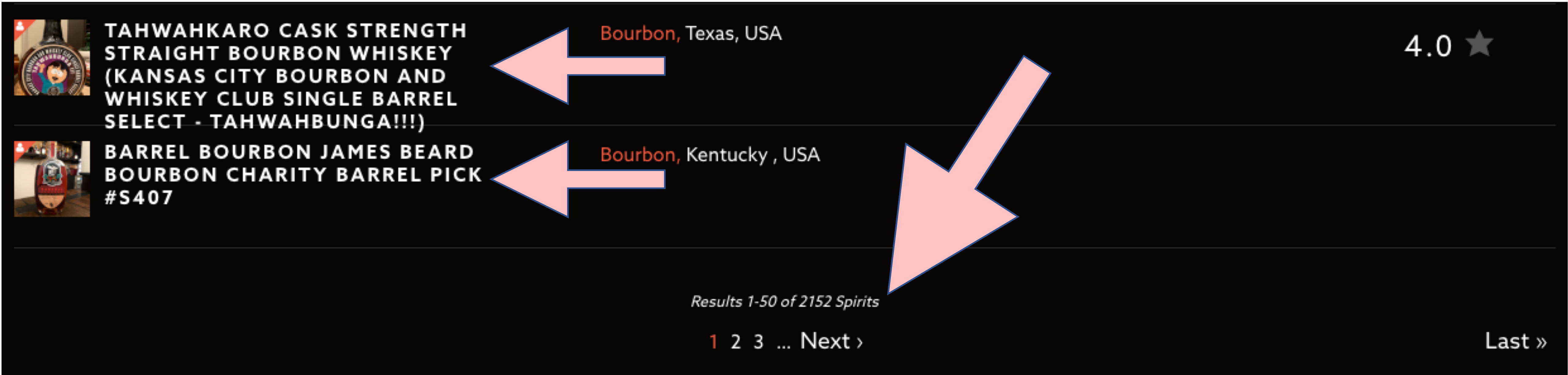
Missing Expert Scores?

Is it possible to scrape the webpages?

Step 2: get the data



1. Download the contents of each search page, while saving each whiskey's URL to a set()



Step 2: get the data

2. Visit each whiskey page, while extracting all the pertinent info that's available

Name	object
Type	object
Cask	object
Location	object
Age	object
ABV %	object
Price	int64
Badge	object
# Ratings	int64
Customers' Rating	object
Flavor Summary	object
Expert	object
Expert Score	object
Smoky	float64
Peaty	float64
Spicy	float64
Herbal	float64
Oily	float64
Full-bodied	float64
Rich	float64
Sweet	float64
Briny	float64
Salty	float64
Vanilla	float64
Tart	float64
Fruity	float64
Floral	float64
Review	object

Step 2: get the data

3. Verify it downloaded correctly and that you don’t need to change how you obtained the data

How much data? 2,139 search result pages on the site

Downloaded 2,205

Step 3: explore the data

Filter by those reviewed by an Expert → 701

Filter by those that are bourbons → 586

```
df2 = df2.loc[(df['Type'] == "Bourbon")]
```

```
df2['Type'].value_counts()
```

Bourbon	586
Single Malt	27
Blended American Whiskey	14
Aged Rum	12
Peated Single Malt	11
Other Whiskey	11
Flavored Whiskey	5
Gold Rum	4
Rhum Agricole Vieux	4
Tequila Reposado	4
Blended	3
American Single Malt	3
Spiced Rum	3
Tequila Añejo	3
Flavored Rum	2
Barrel-Aged Gin	2
Rye	2
Canadian	2
Dark Rum	2
Cachaça	2
Other Brandy	1
Rhum Agricole Éléve Sous Bois	1
Rhum Agricole Blanc	1
Dairy/Egg Liqueurs	1
Silver Rum	1
White	1
Old Tom Gin	1
Other Liqueurs	1

Step 3: explore the data

Filter by those that have Customer Rating → 585

```
df2.loc[df2['Customers\' Rating'] == "N/A"]
```

	Name	Type	Cask	Location	Age	ABV %	Price	Badge	Ra
1765	Tacoma New West Bourbon	Bourbon	new, charred American oak	Heritage Distilling Co. // Washington, USA	NAS	46	2		

```
df2 = df2.loc[df2['Customers\' Rating'] != "N/A"]
df2 = df2.astype({'Customers\' Rating' : 'float64'})
```

Step 3: explore the data

A lot of missing Age statements

```
# we can keep the 'Age' feature for now but be mindful
# that it's missing for nearly half of the whiskeys
len(df2.loc[(df2['Age'] == 'NAS') | (df2['Age'] == 'nas') | (df2['Age'] == '')])
```

```
378
```

```
# let's replace all missing values with a reasonable value.
# for now, let's use 0 as a placeholder so that we can later swap it out.
df2['Age'] = df2['Age'].replace(['NAS', 'nas', 'N/A',''],'0')
```

Step 3: explore the data

```
# remove the 'Years' part of the text
df2['Age'].replace(to_replace =' [yY]ear[sS]*', value = '', regex = True)
```

```
0                    0
4                    0
12       7 y, 2 m,16 d
21                   0
22                   0
26                  17
27                   0
28                   0
38                   6
40                  17
49                   0
52                   0
53                   0
59                   0
60                  10
65                   0
67                   0
75                   0
81                   0
85                   0
```

Step 3: explore the data

```
# manually cleaning up values that otherwise would be a bit impossible to automatically clean
df2['Age'] = df2['Age'].replace(to_replace ='6.*', value = '6', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='(\d+) [Yy].*', value = '\\1', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='4 [Mm]onths', value = '4', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='9 [Mm]onths', value = '9', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='18 - 20 [Mm]onths', value = '1.5', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='32 [Mm]onths', value = '2.67', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='9 [Mm]onths', value = '9', regex = True)
df2['Age'] = df2['Age'].replace(to_replace ='9 to 11', value = '0.75', regex = True)
```

Step 3: explore the data

```
# let's look at all of the items that had an Age
# (now that all values have been cleaned-up)
df2.loc[df2['Age'] > '0']['Age']
```

```
12       7
26      17
38       6
40      17
60      10
97      15
98      12
118     12
119      6
140     22
149      3
154      6
162      6
166     11
188     13
202      9
231     17
236      6
257     14
258      6
```

```
df2 = df2.astype({'Age': 'float64'})
```

Step 3: explore the data

```
# how many had values?
len(df2.loc[df2['Age'] > 0])
```

```
206
```

```
df2['Age'].describe()
```

```
count    585.000000
mean       3.776274
std        6.010627
min        0.000000
25%        0.000000
50%        0.000000
75%        7.000000
max       28.000000
Name: Age, dtype: float64
```

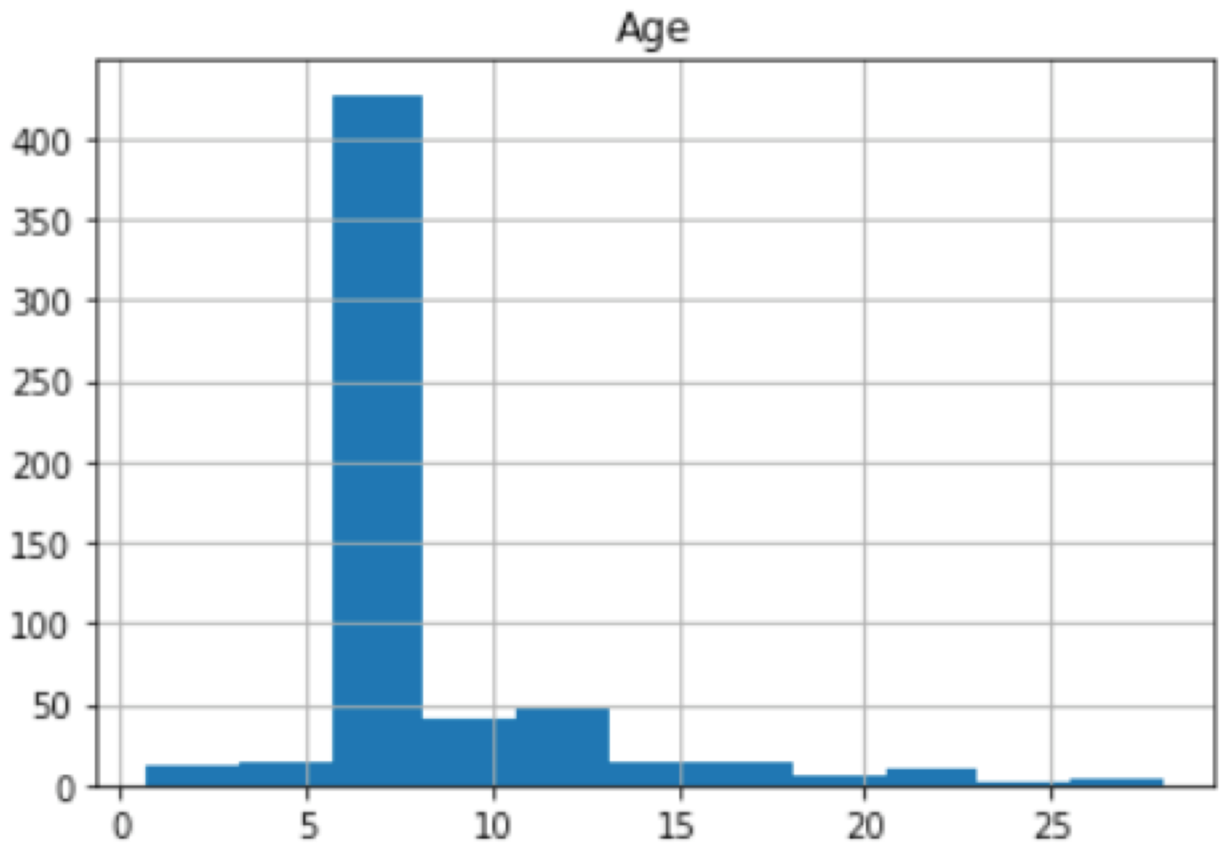
Step 3: explore the data

```
df2['Age'] = df2['Age'].replace(0,7)
```

```
df2['Age'].describe()
```

```
count    585.000000
mean       8.311316
std        3.607727
min        0.750000
25%        7.000000
50%        7.000000
75%        7.000000
max       28.000000
Name: Age, dtype: float64
```

Step 3: explore the data



Step 3: explore the data

What's the distribution of the Flavor Summary?

```
df2['Flavor Summary'].value_counts()
```

```
Rich & Full Bodied          54
Sweet & Rich                40
Sweet                       36
Vanilla & Sweet             34
Spicy                       33
Vanilla & Rich              24
Full Bodied & Rich          20
Sweet & Vanilla             20
Spicy & Rich                18
Vanilla                     18
Vanilla & Full Bodied       17
Fruity & Sweet              17
Full Bodied & Spicy         17
Spicy & Vanilla             16
Rich & Vanilla              13
Sweet & Spicy               13
Rich & Spicy                13
Full Bodied                 11
Vanilla & Spicy             11
Full Bodied & Vanilla       10
Spicy & Sweet               10
Spicy & Full Bodied         10
```

Step 3: explore the data

What's the Badge feature like?

```
df2['Badge'].value_counts()
```

```
                                               428
RARE                                           119
Requested By\nElw00t                             2
Requested By\njd139                              1
Requested By\ntjbriley                           1
Requested By\nBourbon_Obsessed_Lexington         1
Requested By\nCymru-and-the-Ferg                 1
Requested By\ndanmeister33                       1
Requested By\nCblake34                           1
Requested By\ndjriebesell                        1
Requested By\nandrewls24                         1
Requested By\nspectorjuan                        1
Requested By\ncubfanccccc                        1
Requested By\nsamueljcarlson                     1
Requested By\nJFForbes                           1
Requested By\nJamesSpears                        1
```

Step 3: explore the data

What's the Expert feature like?

```
df2['Expert'].value_counts()
```

```
Jacob Grier            92
Jake Emen              85
Amanda Schuster        76
Stephanie Moreno       66
Rob Morton             62
Keith Allison          26
Colin Howard           23
Sam Davies             21
Nicole Gilbert         17
Distiller Staff        15
Brock Schulte          14
Paul Belbusti          13
Ryan Conklin           12
Jack Robertiello       10
Tim Knittel            10
Katrina Niemisto        8
Dennis Gobis            4
```

Step 3: explore the data

Now that our data is clean, let's explore it. EDA time!

Please see Case_Study_PART_1-4.ipynb, which I will make available after the matinee lecture.

Step 4: model the data

1. Are there certain attributes of bourbons that are predictive of good bourbons? (i.e., highly rated by customers)

- Find hidden gems (i.e., should be good but current reviews are absent or unsupportive of such)
- Find over-hyped bourbons (i.e., the reviews seem high but the attributes aren't indicative)
- Are there significant results if we target experts' ratings instead of average customer ratings?

Step 4: model the data

Goal: predict Customers' Ratings

Data: train/dev/test splits

Features to use: ???

Question: which features should we use?
Could we use?

```
Name                   object
Type                   object
Cask                   object
Location               object
Age                   float64
ABV %                 float64
Price                   int64
# Ratings               int64
Customers' Rating     float64
Flavor Summary         object
Expert                 object
Expert Score            int32
Smoky                 float64
Peaty                 float64
Spicy                 float64
Herbal                float64
Oily                  float64
Full-bodied           float64
Rich                  float64
Sweet                 float64
Briny                 float64
Salty                 float64
Vanilla               float64
Tart                  float64
Fruity                float64
Floral                float64
Review                 object
Rare                     bool
```

Break-out room time!

(Discussion, no coding)

Step 4: model the data

Goal: predict Customers' Ratings

Data: train/dev/test splits

Features to use:

- All 14 flavors
- Age
- ABV %
- Price
- Badge
- Expert Score

Accuracy Metric: MSE

```
Name                   object
Type                   object
Cask                   object
Location               object
Age                   float64
ABV %                 float64
Price                   int64
# Ratings               int64
Customers' Rating     float64
Flavor Summary         object
Expert                 object
Expert Score            int32
Smoky                 float64
Peaty                 float64
Spicy                 float64
Herbal                float64
Oily                  float64
Full-bodied           float64
Rich                  float64
Sweet                 float64
Briny                 float64
Salty                 float64
Vanilla               float64
Tart                  float64
Fruity                float64
Floral                float64
Review                 object
Rare                     bool
```

Step 4: model the data

Model #1: Linear Regression

Question: should we scale our data?

Question: should we use polynomial features?

Please see Case_Study_PART_5.ipynb, which I will make available after the matinee lecture.

```
Name                    object
Type                    object
Cask                    object
Location                object
Age                    float64
ABV %                  float64
Price                    int64
# Ratings                int64
Customers' Rating      float64
Flavor Summary          object
Expert                  object
Expert Score             int32
Smoky                  float64
Peaty                  float64
Spicy                  float64
Herbal                 float64
Oily                   float64
Full-bodied            float64
Rich                   float64
Sweet                  float64
Briny                  float64
Salty                  float64
Vanilla                float64
Tart                   float64
Fruity                 float64
Floral                 float64
Review                  object
Rare                      bool
```

Step 1: formulate questions

1. Are there certain attributes of bourbons that are predictive of good bourbons? (i.e., highly rated by customers)

- Find hidden gems (i.e., should be good but current reviews are absent or unsupportive of such)
- Find over-hyped bourbons (i.e., the reviews seem high but the attributes aren't indicative)
- Are there significant results if we target experts' ratings instead of average customer ratings?

Step 4: model the data

Good, important practices:

- Heavily inspect your data first
- Spend the extra time to clean it
- Start with the most simple models
- If the model seems sensitive to a particular run, use bootstrapping
- Inspect results, and allow that to guide your next choices
- Reflect about your modelling choices
- Leverage as much of your data as possible (cross-validation)

Step 4: model the data

Good, important practices:

- When comparing different models, make everything as fair as possible
  - Same data splits
  - Fix all random seeds so your experiments are repeatable
- Look at your worst mistakes. Any patterns to these errors?
- For classification tasks, look at false positives and false negatives
- Do you have any indication that your data is limiting?
  - Clean up data further or get more data?

Step 5: communicate your results

- A long notebooks I provided are only good for exploring and getting work done. By no means is it an attempt to communicate the results
- Think of what would be the easiest, most succinct way to discern:
  - All of your models' results (e.g., heatmap table?).
  - The effectiveness and ineffectiveness of your model
  - Examples of it working / results
- It should be clear and compelling which model we should use (instead of the baseline)
- Review the Visualization lecture for more inspiration

Step 1: formulate questions

2. Are there certain attributes of bourbons that are predictive of expensive bourbons?

- Find underpriced ones
- Find over-priced ones

Step 1: formulate questions

3. Which bourbons are most similar to each other?

- Which attributes are important for determining similarity? (e.g., does price play a role, or does similarity transcend price?)

Good luck on your projects! You can do it.