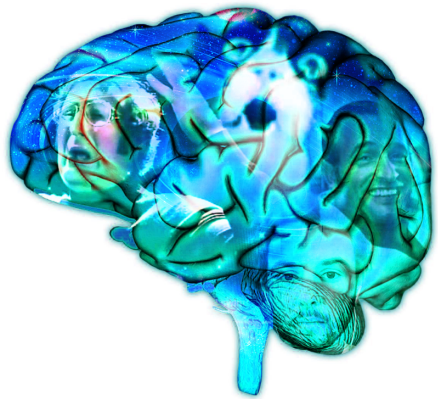
Neural Network

Jia FU, Victor JUNG, Steve MALALEL

Université Nice Sophia-Antipolis

2018

Contents



Quick presentation of the data

- Data from Kaggle, scrapped from www.winemag.com
- Approximately 130'000 values.

We want to be able to do the following :

- Recommending the price of the wine
- Choosing a wine variety
- Selecting a production's region

Let's start !

Distribution

Preparation of the data

- Jia FU

Networks' constructions

- Victor JUNG

Statistics and graphics

- Steve MALALEL

But everyone worked on everything !

Selecting the data

- We drop the useless columns,
- We drop the lines with missing information.

Refining the data

- If there's no value for the region, region = province,
- We only keep the most important varieties !

At the end, we have :

Country	Region	Province	Variety
40	1412	364	52

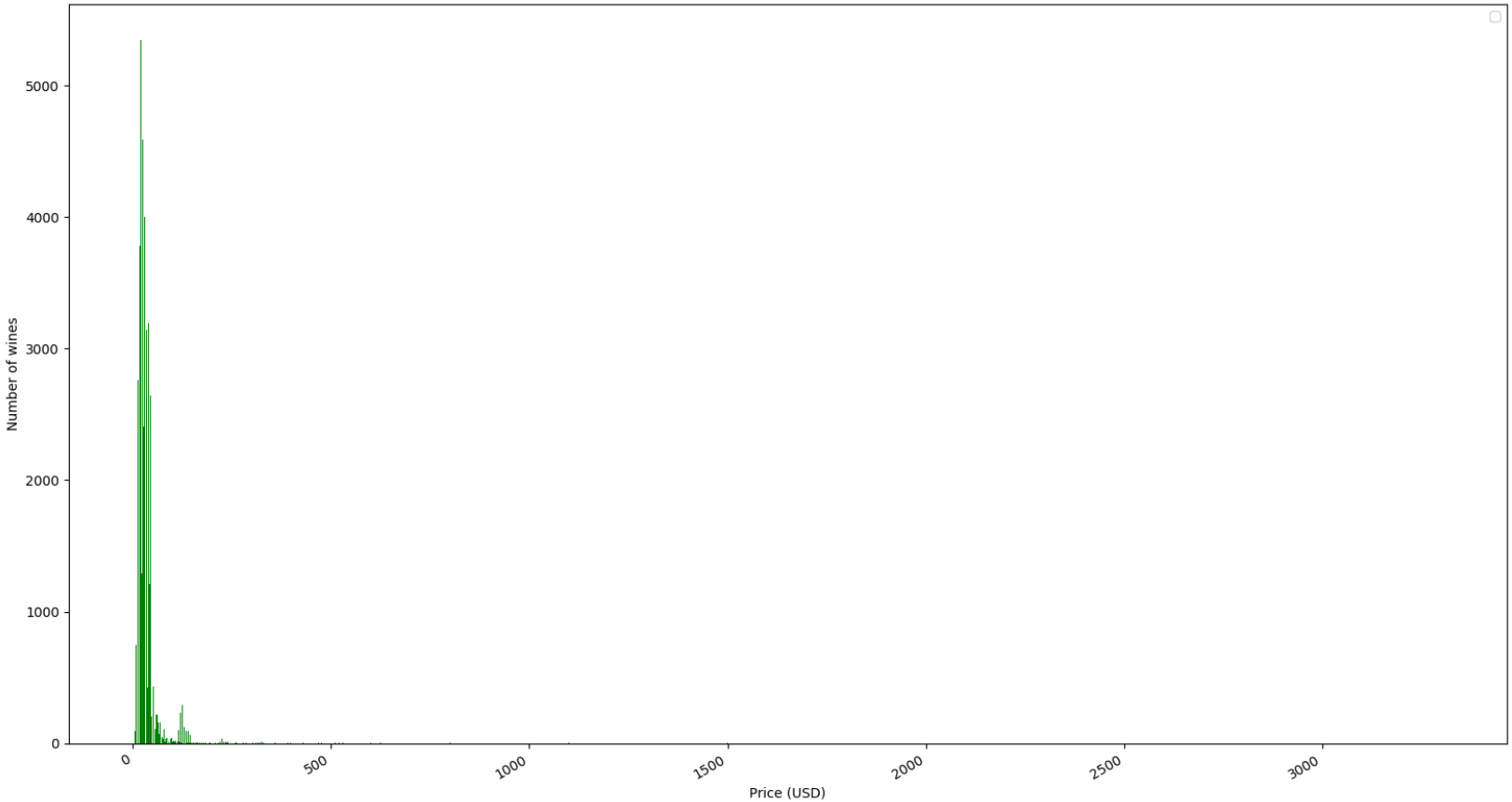
Ordinal values

For all the important values that are not numbers, we create a hash table to switch between one of these values to an integer.

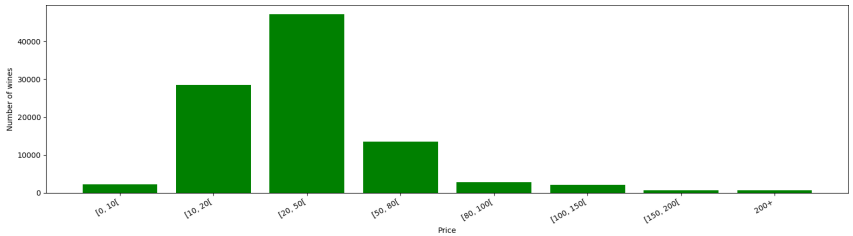
Country	Integer value
Portugal	0
US	1
France	2
...	...

Variety	Integer value
portuguese red	0
pinot gris	1
riesling	2
...	...

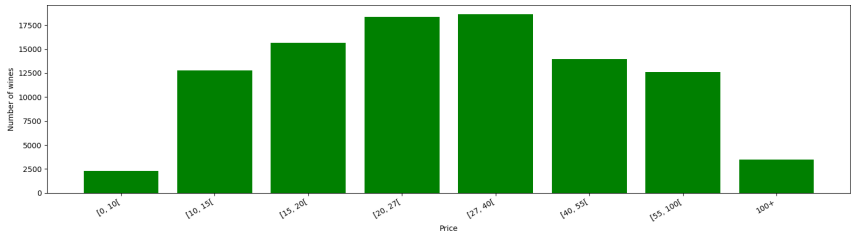
Distribution of the price



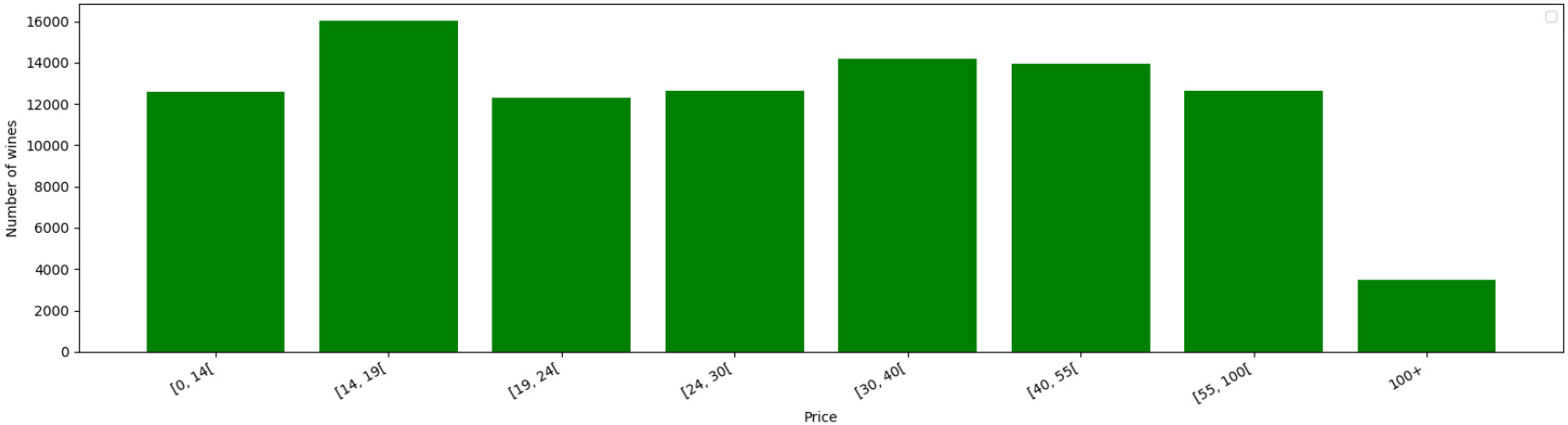
There are too much values ! Let's change that...



Real price range



'Curve' distribution



Uniform distribution

The data is now ready !

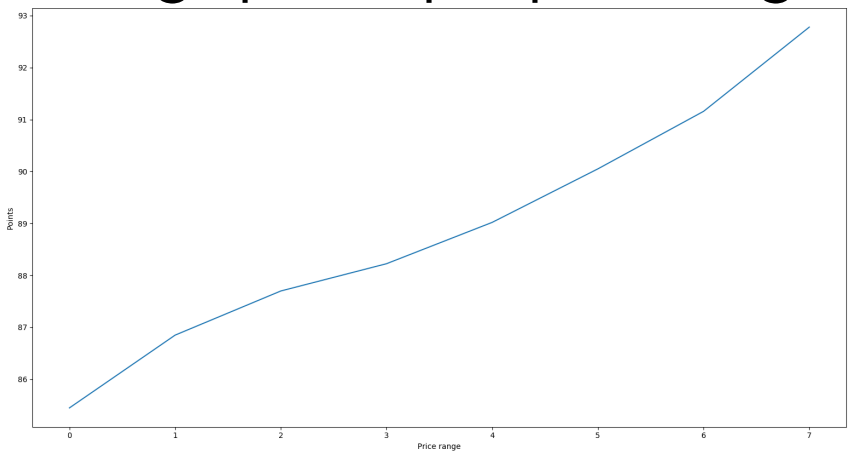
Now, we will try to provide a solution to the three different problems, using two approaches :

- Lazy learning network (KNN)
- Eager learning network (MLP)

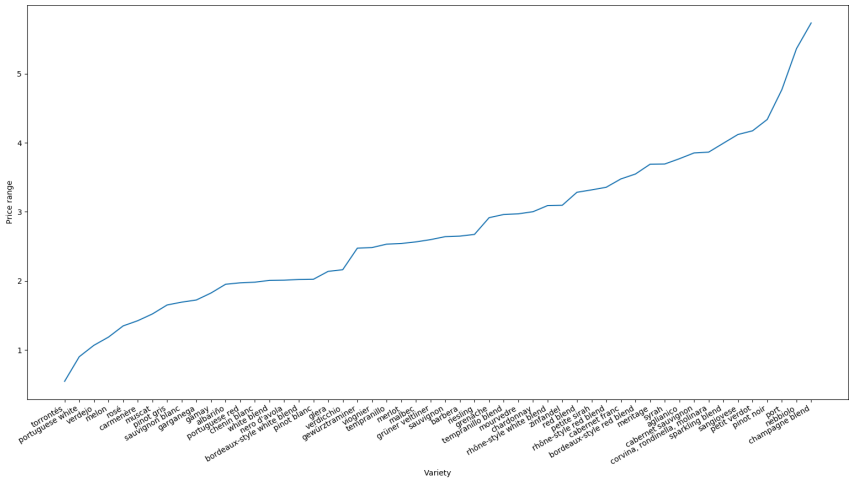
Let's begin with the lazy one...

Price recommender : Analysis

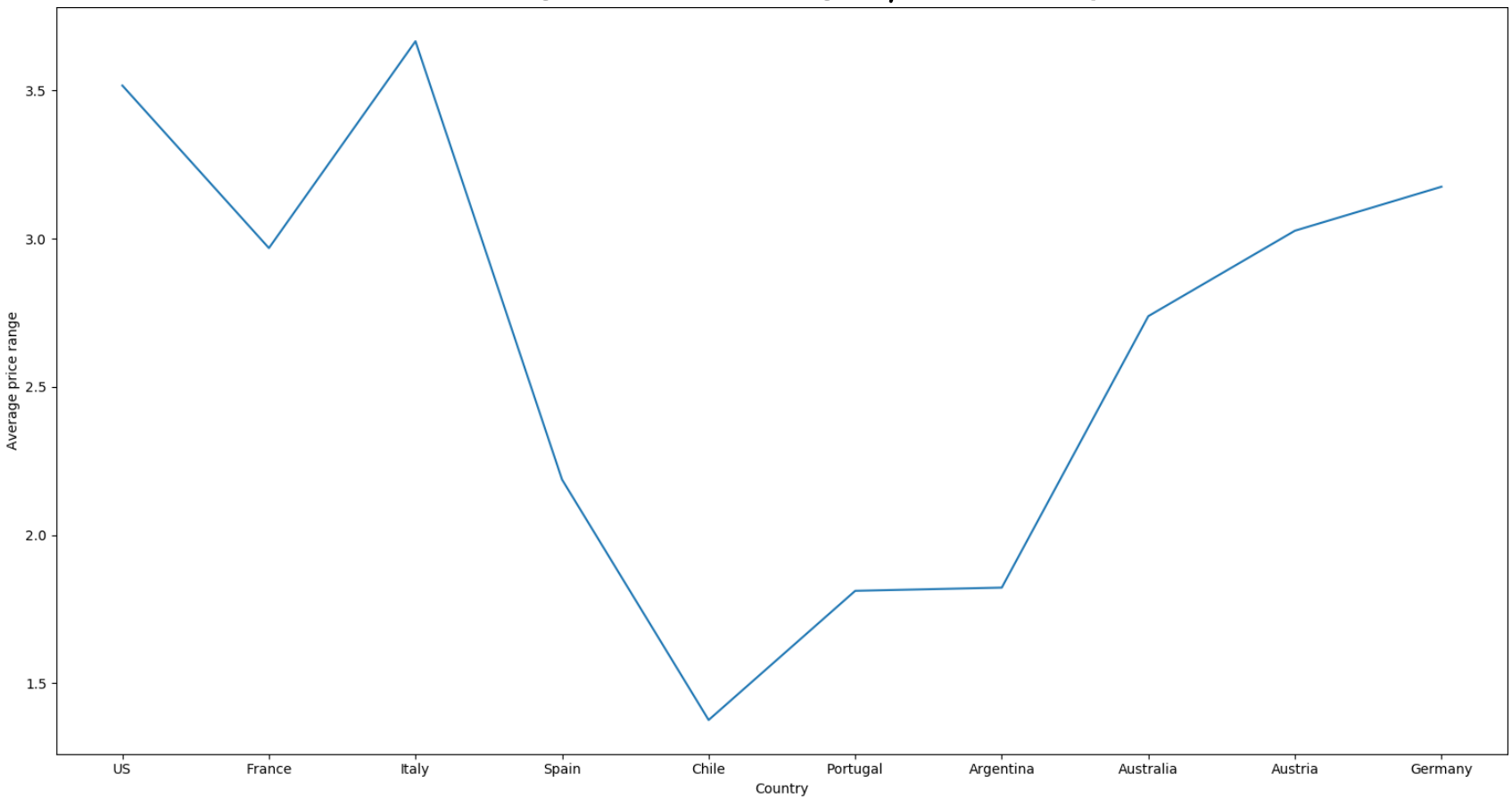
Average points per price range



Average price range / variety



Average price range / country



Price recommender : Building the network

Construction

- Classifier : KNeighborsClassifier
- K = 17
- Algorithm : Ball tree
- Weights : distance

Training set

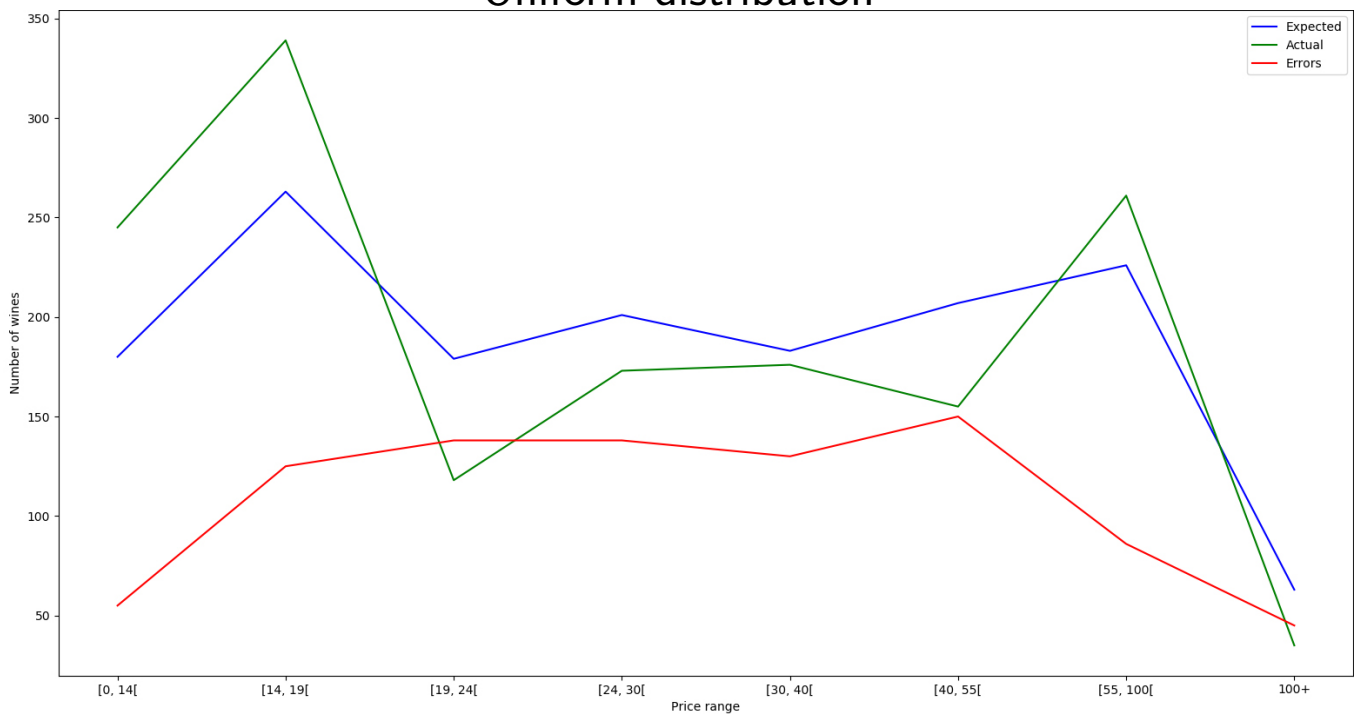
We take 99% of the data to train our network !

Testing set

The rest of the set + reviews from 2018 scrapped from the same site ! (approximately 1500 reviews)

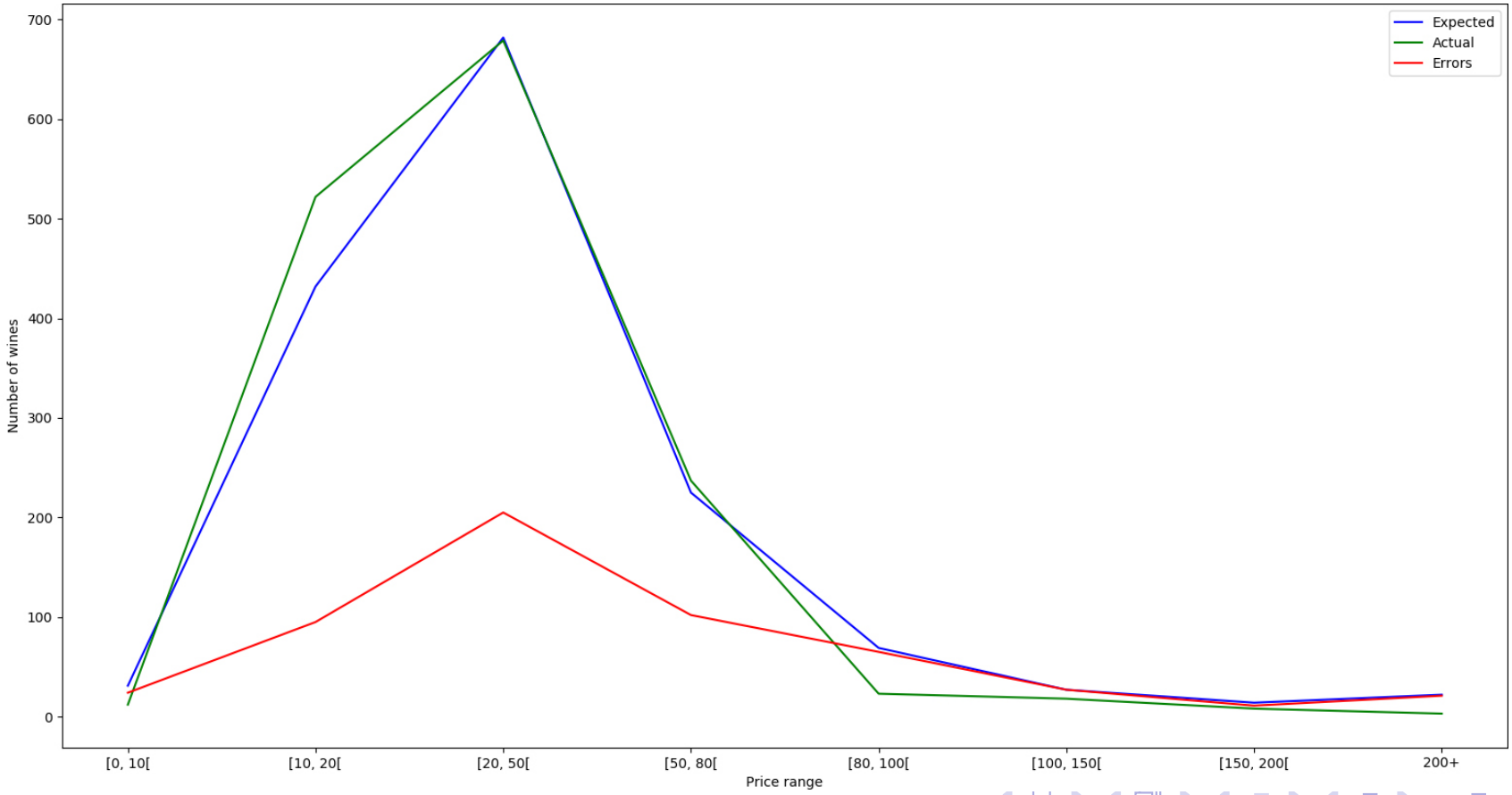
Price recommender : Results

Accuracy	Testing set (%)	2018's reviews (%)	Average difference (%)
Realist	63.9	63.4	15.5
Curve	41.1	42.9	18
Uniform	38.5	42.3	20



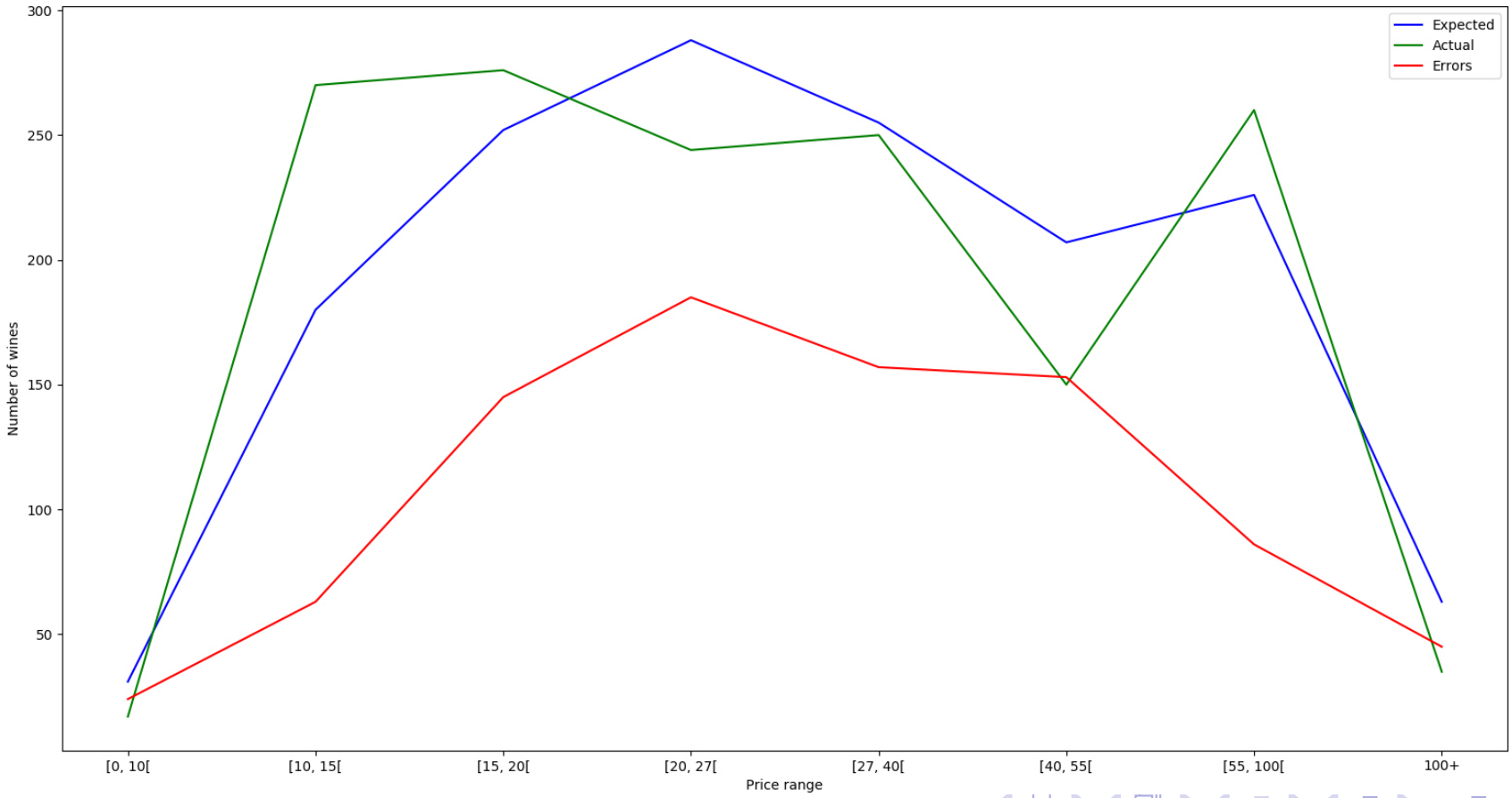
Price recommender : Results

Realistic distribution



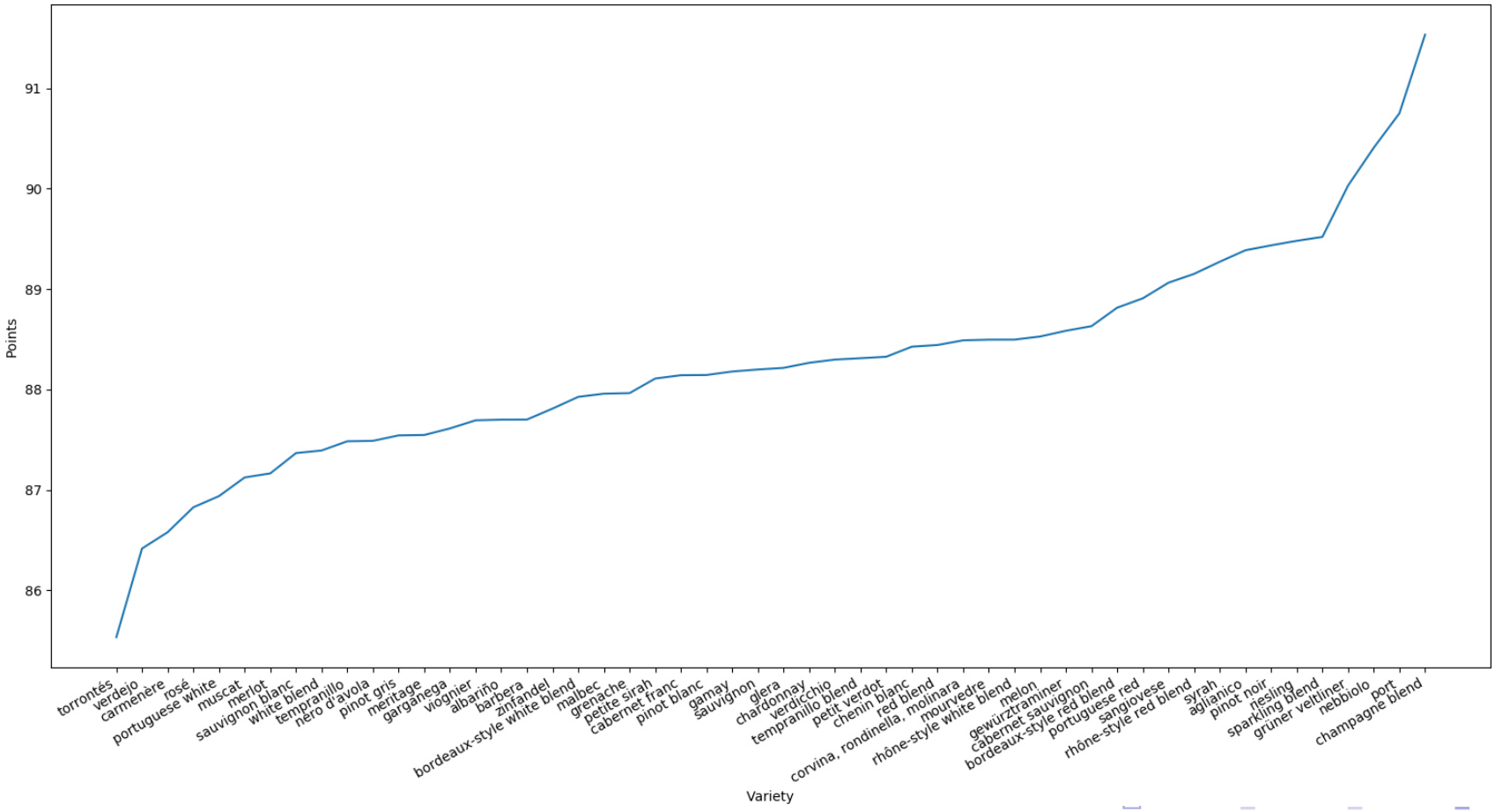
Price recommender : Results

Curve distribution



Variety selector : Analysis

Average points per variety



Variety selector : Analysis

Most used varieties per country

Country	1st variety	2nd variety	3rd variety
US	Pinot Noir	Cabernet Sauvignon	Chardonnay
France	Bordeaux-style Red Blend	Chardonnay	Rosé
Italy	Red Blend	Sangiovese	Nebbiolo
Spain	Tempranillo	Red Blend	Tempranillo Blend
Chile	Cabernet Sauvignon	Sauvignon Blanc	Carmenère
Portugal	Portuguese Red	Portuguese White	Port
Argentina	Malbec	Cabernet Sauvignon	Chardonnay
Australia	Syrah	Chardonnay	Cabernet Sauvignon
Austria	Grüner Veltliner	Riesling	Sauvignon Blanc
Germany	Riesling	Pinot Noir	Gewürztraminer

Repeating : Riesling, Sauvignon Blanc, Red Blend, Pinot Noir, Chardonnay and Cabernet Sauvignon.

Variety selector : Building the network

Construction

- Classifier : KNeighborsClassifier
- K = 15
- Algorithm : Ball tree
- Weights : distance

Training set

Again, 99% of the data.

Testing set

Still the same testing set ! (2018's reviews + rest of the set)

Variety selector : Results

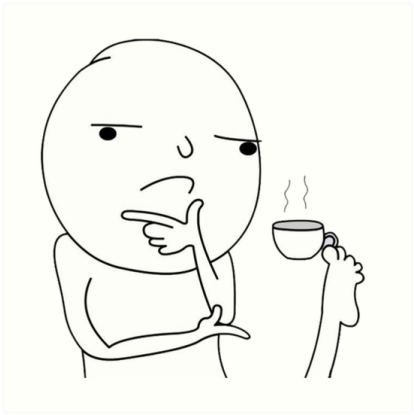
Accuracy	Testing set (%)	2018's reviews (%)
Realist	50.5	53.7
Curve	51.1	52.7
Uniform	50.1	54

At least 50% !

Region selector : Analysis

1412 regions ? No way we can be precise !

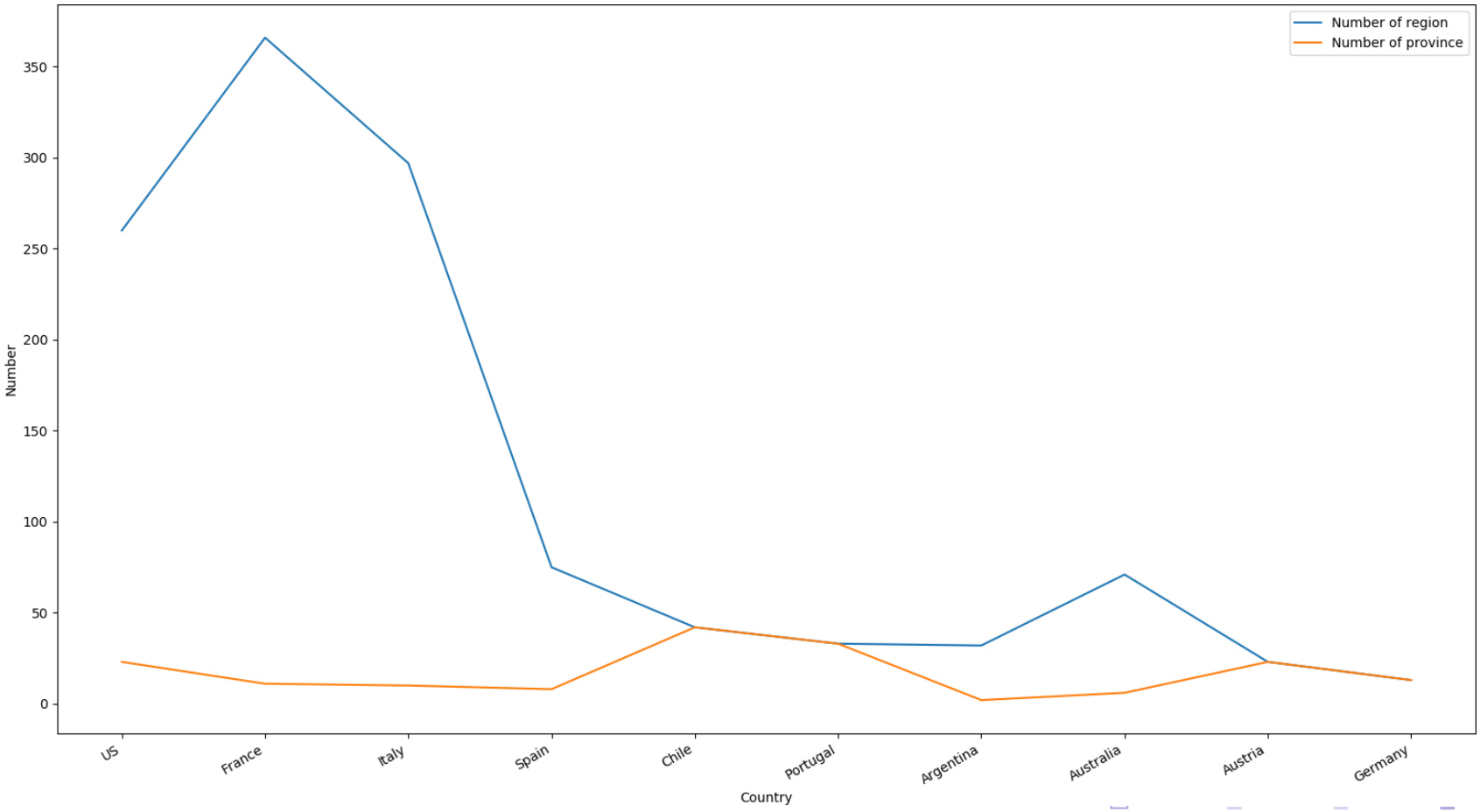
We have to do it in another way...



We can first guess a larger location, and then the region ?
Okay, let's do it !

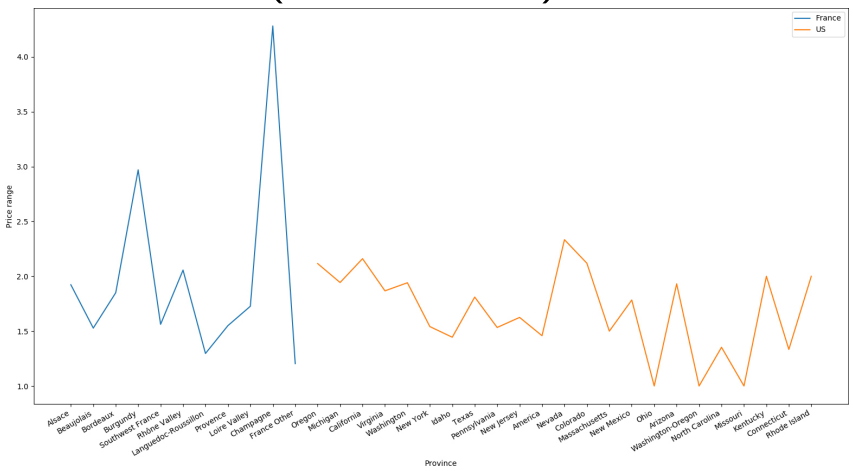
Region selector : Analysis

Number of region and province per country (top 10)

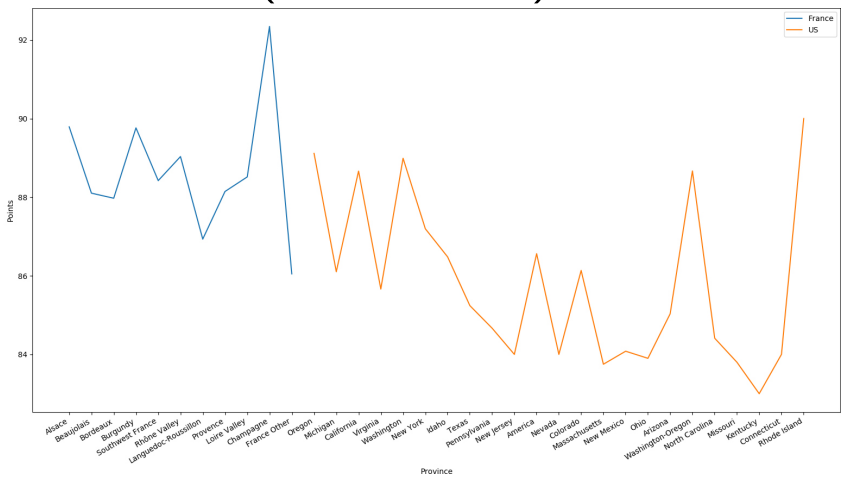


Region selector : Analysis

Average price range per province (France, US)



Average points per province (France, US)



Region selector : Building the network

Construction

- Classifier : KNeighborsClassifier x2
- K = 10 / 16
- Algorithm : Ball tree / KD tree
- Weights : uniform

Training set

Surprise ! 99% of the data for both.

Testing set

2018's reviews

Region selector : Results

	Province (%)			Region (%)		
	ALL	US	NOT US	ALL	US	NOT US
Realist	49.2	69	53.5	41.1	26.2	57.3
Curve	49.8	68.5	55	43.6	27.3	60.1
Uniform	51.2	69.3	55	42.7	28.4	56.7

realised with 2018's reviews

We are better at guessing the region outside the US, but better at guessing the province inside the US.
Overall, good results !

Price recommender : Building the network

Construction

- Classifier : MLPClassifier
- activation : tanh (f(x) = tanh(x))
- learning rate : adaptive
- hidden layer sizes = (40, 30, 20, 10)

Training set

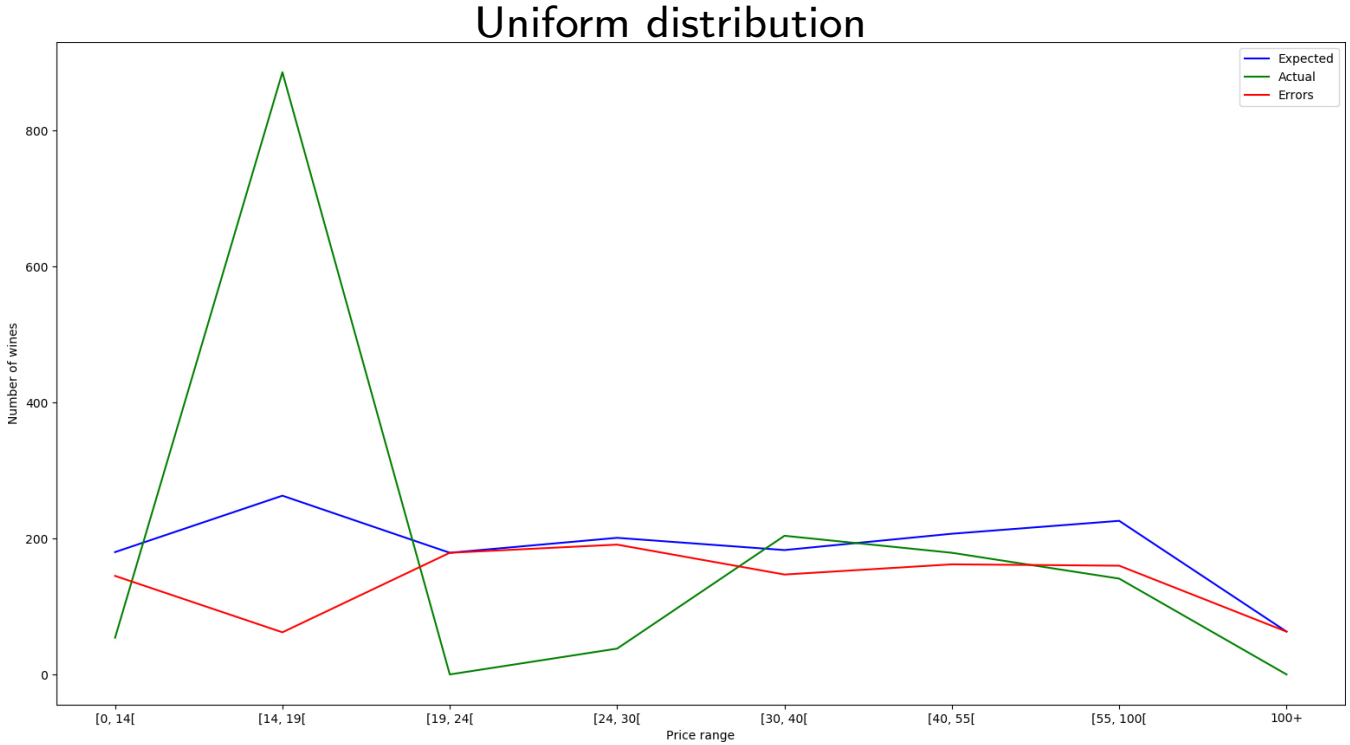
20% of the data.

Testing set

2018's reviews + rest of the set

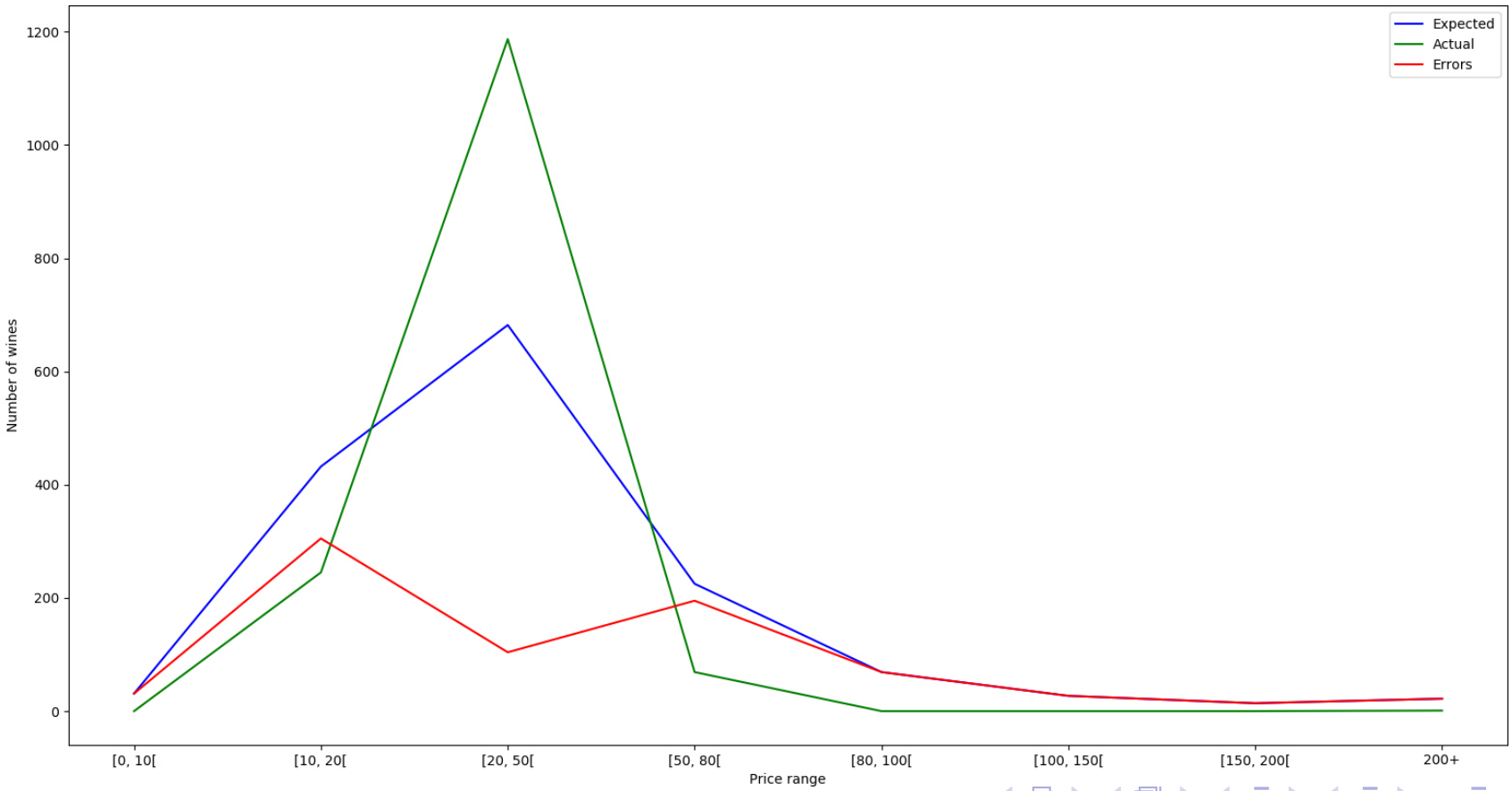
Price recommender : Results

Accuracy	Testing set (%)	2018's reviews (%)	Average difference (%)
Realist	52.1	48.9	16
Curve	26.2	25.4	21.2
Uniform	25.1	27	27.5



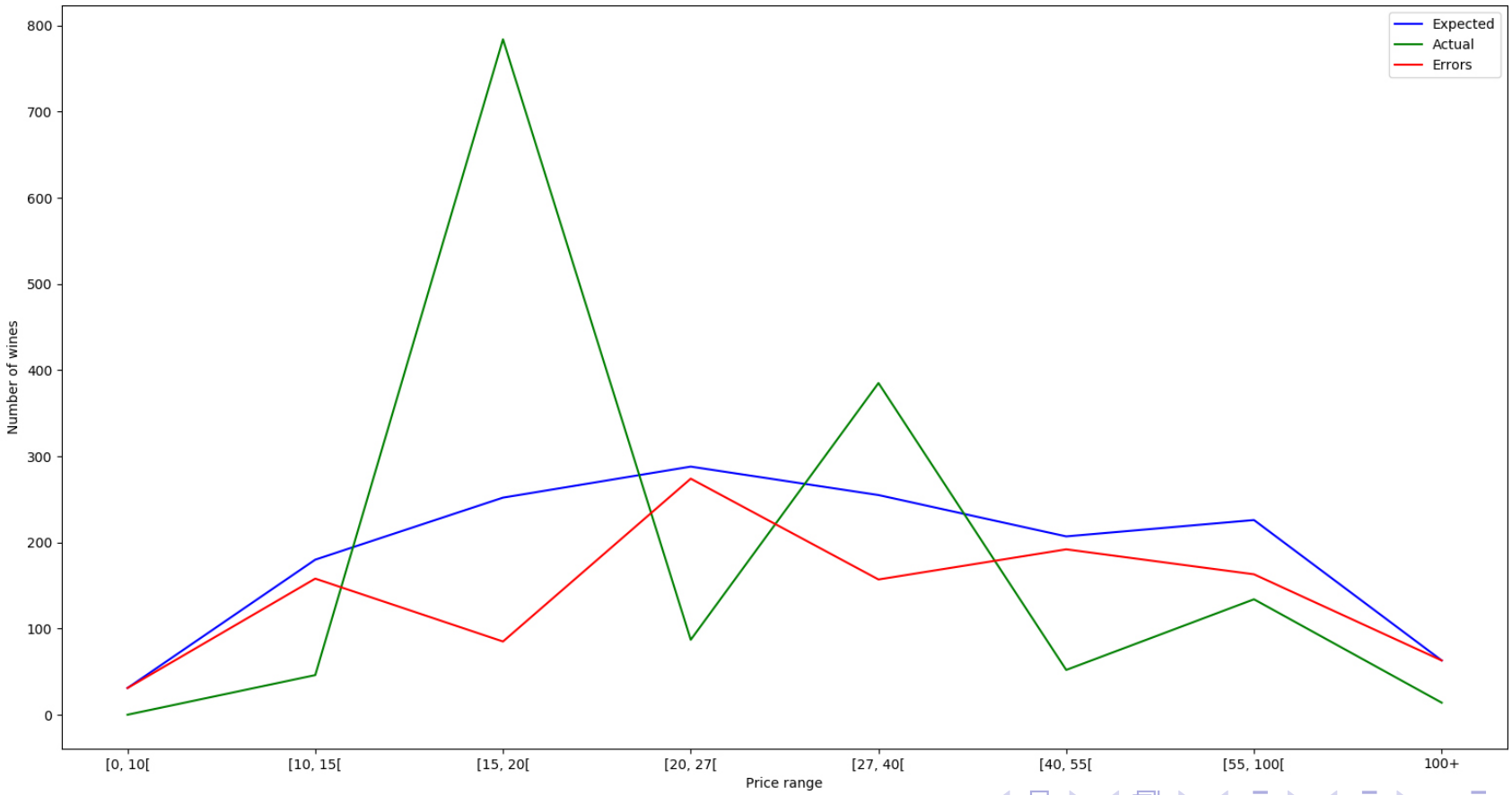
Price recommender : Results

Realistic distribution



Price recommender : Results

Curve distribution



Variety selector : Building the network

Construction

- Classifier : MLPClassifier
- activation : logistic (f(x) = 1 / (1 + exp(-x)))
- learning rate : adaptive
- hidden layer sizes = (100, 100)

Training set

20% of the data.

Testing set

2018's reviews + rest of the set

Variety selector : Results

Accuracy	Testing set (%)	2018's reviews (%)
Realist	31.1	33
Curve	30.7	32.7
Uniform	27.9	29.2

Way under the KNeighborsClassifier...
30% is still fine.

Region selector : Building the network

Construction

- Classifier : MLPClassifier
- activation : logistic (f(x) = 1 / (1 + exp(-x)))
- learning rate : adaptive
- hidden layer sizes = (30, 30, 30)

Training set

30% of the data.

Testing set

2018's reviews

Region selector : Results

	Province (%)			Region (%)		
	ALL	US	NOT US	ALL	US	NOT US
Realist	49.2	69	53.5	39.4	25.6	53.8
Curve	49.8	68.5	55	40	25.4	53.8
Uniform	51.2	69.3	55	40.2	27.8	52.1

Almost as good as the lazy one !

The end