.

Université Paris Cité

M2MO

Projet Statistiques de l'Entreprise

Classification de vins

Victor Guillard

2023

1 Introduction

La marché du vin est un des piliers de la culture française qui n'a que très peu été influencé par la révolution du machine learning. J'ai donc souhaité avec ce projet appliquer des méthodes actuelles pour pouvoir exploiter les données disponibles afin de proposer des services de meilleure qualité. Dans ce cadre, la prédiction des notes d'un vin en se basant sur des données comme par exemple le type de raisin, le lieu de production, le millésime est un enjeu majeur pour les entreprises du secteur.

Notre objectif pour ce projet est de prédire la note d'un vin à partir de multiples variables présentes dans la base de données. Nous avons utilisé diverses données telles que les descriptifs de dégustation faites par un sommelier et les informations sur le vin lui-même (millésime, cépage, région de production, etc.).

Nous avons adopté une approche de classification supervisée pour résoudre ce problème, où les différentes notes possibles (nous avons pris le parti de considérer une notation sous forme d'étoiles, de 1 à 5) étaient considérées comme des classes distinctes. Notre base de données contient 130 000 vins différents, qui sont décrits par des variables numériques telles que le prix et la note et des variables catégorielles comme la région de production, le type de raisin, et des variables textuelles telles que les descriptifs de dégustation.

Nous avons tout d'abord nettoyé et transformé les données textuelles en données utilisables par un modèle de machine learning, puis nous avons effectué une analyse exploratoire des données, nous avons ensuite procédé au prétraitement de la base de données et enfin nous avons implémenté deux modèles de classification multiclasse : un modèle de régression logistique comme benchmark, suivi d'un modèle de boosting, XGBoost.

Notre travail nous a permis d'obtenir des modèles de prédiction plutôt précis. Cela pourrait permettre à des entreprises de mieux comprendre les caractéristiques qui influencent la qualité d'un vin, et de proposer des recommandations plus précises à leurs clients.

2 Nettoyage de la base de données

2.1 Tableau récapitulatif de la base de données brute

	Names	dtypes	Missing	Uniques	Entropy
0	country	object	63	43	2.77
1	description	object	0	119955	16.83
2	designation	object	37465	37979	13.83
3	points	int64	0	21	3.64
4	price	float64	8996	390	5.77
5	province	object	63	425	5.05
6	region_1	object	21247	1229	7.89
7	region_2	object	79460	17	3.28
8	$\text{taster}_{n}ame$	object	26244	19	3.39
9	$\text{taster}_{t}witter_{h}andle$	object	31213	15	3.23
10	title	object	0	118840	16.81
11	variety	object	1	707	5.67
12	winery	object	0	16757	12.98

2.2 Description de la base de données

Cette base de données contient donc 130 000 lignes décrites par 13 variables : 'country' le pays d'origine du vin, 'description' qui est une variable textuelle contenant le résumé de la dégustation du sommelier pour ce vin, 'designation' le nom du vin, 'points' la note attribuée à ce vin par le sommelier, 'price' le prix du vin en dollars américain, ('province', 'region_1', 'region_2') sont trois variables géographiques qui nous donnent plus de précision sur le lieu de production du vin, 'taster_name' est le nom du sommelier ayant testé le vin, 'taster_twitter_handle' est l'addresse twitter du sommelier, 'title' est le nom complet du vin avec l'année de production, 'variety' comprend les types de cépages utilisés pour produire le vin et enfin 'winery' est le nom de l'entreprise vigneronne ayant produit ce vin.

2.3 Gestion des données aberrantes

Nous avons tout d'abord analysé les données aberrantes. Pour la variable points toutes les données étaient comprises entre 80 et 100 ce qui est parfaitement cohérent. En revanche pour le variable prix nous avons remarqué en comparant les prix des 50 vins les plus chers de la base avec les prix que nous avons trouvés sur internet que 3 vins étaient beaucoup plus chers que dans la réalité : nous avons donc remplacé ces données manuellement.

2.4 Gestion des doublons

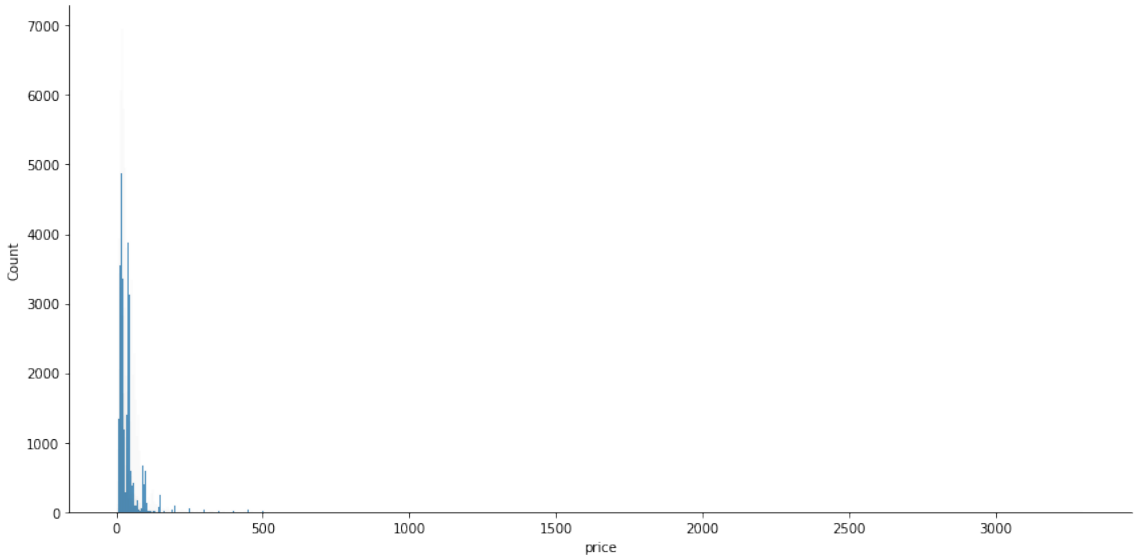
Nous avons ensuite appliqué la méthode drop duplicate de pandas pour enlever les doublons ce qui a réduit notre base de données à environ 120 000 lignes.

3 Analyse exploratoire des données

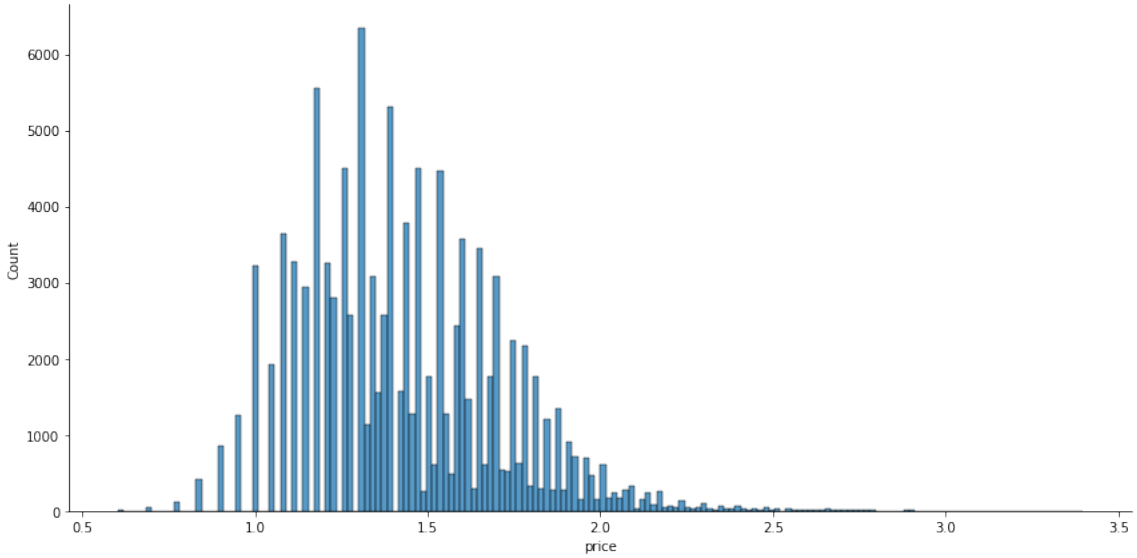
3.1 Méthodologie

Comme nous pouvons le voir dans le graphique ci-dessous la distribution de la variable prix est très étalée vers la droite ; nous avons donc choisi pour la partie exploratoire de provisoirement transformer cette colonne logarithmiquement ce qui nous donne une distribution ressemblant plus à une gaussienne commme le second graphique le montre.

3.2 Distribution originale de la variable prix



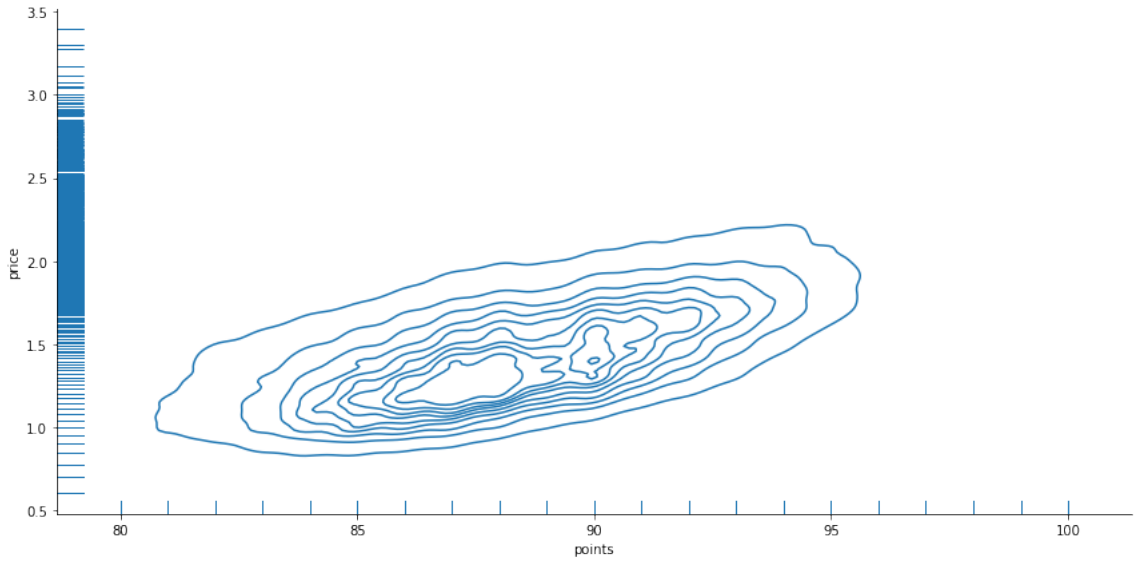
3.3 Distribution de la variable prix après transformation logarithmique



3.4 Les 20 pays ayant le plus grand nombre de vins

country	description
US	50457
France	20353
Italy	17940
Spain	6116
Portugal	5256
Chile	4184
Argentina	3544
Austria	3034
Australia	2197
Germany	1992
South Africa	1301
New Zealand	1278
Israel	466
Greece	432
Canada	226
Bulgaria	132
Hungary	129
Romania	102
Uruguay	98
Turkey	81
Slovenia	77

3.5 Relation entre les variables prix et points



3.6 Ressources supplémentaires

Pour aller plus loin, nous avons mis en annexe trois graphiques qui montrent les relations entre les variables points, country et taster plus en détails.

4 Pré-traitement, Modélisation et Apprentissage

Nous avons organisé notre démarche de résolution de problème autour de la fonctionnalité de pipeline de scikit-learn. Les deux étapes de preprocessing et l'embedding Word2Vec on été codés comme des transformers s'intégrant au pipeline et les algortihmes d'apprentissage ont été codés comme des classifiers s'ajoutant à la fin des pipelines. Nous

4.1 Pré-traitement des données

Nous avons créer deux transformer de prétraitement des données. Le premier est un transformer de gestion des données manquantes ; en effet comme le montre le tableau 2.1 nous avons beaucoup de données manquantes (sauf pour les variables description, points, title et winery). Nous avons choisi de remplacer les données manquantes de la colonne prix par la médiane de cette colonne et de remplacer les données manquantes des colonnes catégorielles par une chaîne de caractère vide. Le deuxième transformer que nous avons codé est un transformer qui nous retourne une représentation catégorielle sous forme d'entiers de toutes les variables catégorielles que nous avons.

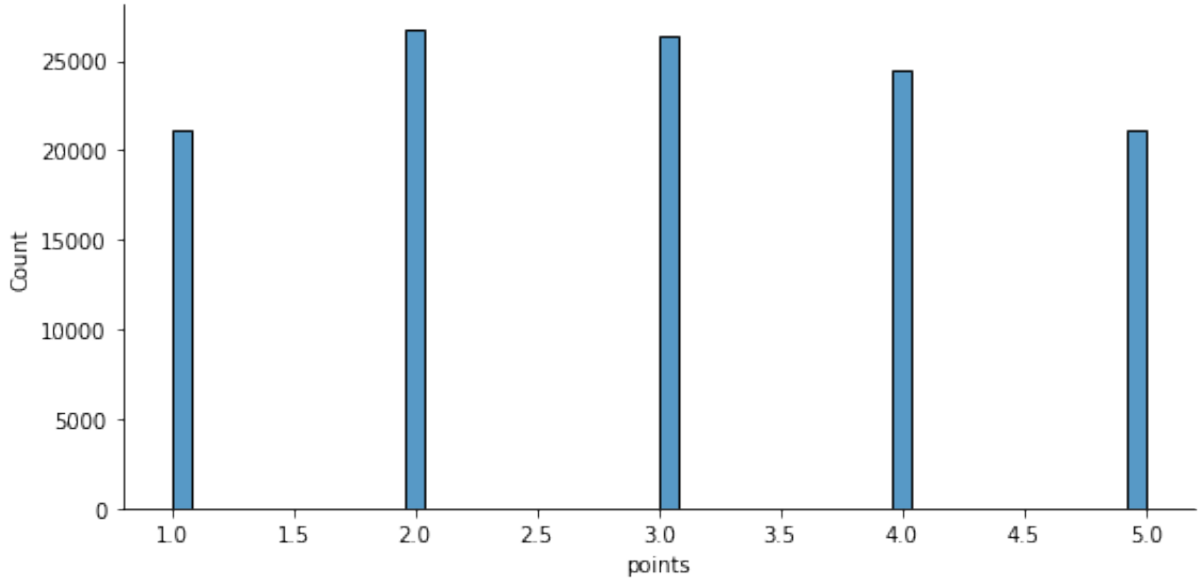
4.2 Étude de la variable objectif

La variable cible dans votre projet est la note d'un vin. Les notes vont de 0 à 100, et certaines notes peuvent être plus fréquentes que d'autres. Par exemple, les notes autour de 85-90 peuvent être plus fréquentes que les notes extrêmes de 0 ou 100. Pour prédire les notes, il est important de comprendre la distribution des notes dans les données. Nous avons de plus fait le choix de restreindre cette note à un système d'étoiles de 1 à 5 ce qui facilitera grandement l'apprentissage tout en gardant une pertinence et une interprétabilité importante (on peux voir dans le graphique suivant cette section la nouvelle distribution de la variable points que nous avons voulu le plus ressemblante à une loi uniforme pour faciliter l'apprentissage).

Il est également important de noter que différents experts peuvent attribuer des notes différentes à un même vin. Il peut donc y avoir une certaine variabilité dans les notes. Cela peut rendre la prédiction des notes plus difficile que si la notation était plus standardisée.

En résumé, pour prédire les notes d'un vin, il est important de comprendre la distribution des notes dans les données et d'étudier la corrélation entre les caractéristiques du vin et les notes attribuées. Cela peut aider à identifier les facteurs les plus importants pour la prédiction des notes et à améliorer les performances du modèle.

4.3 Distribution de la variable points après transformation



4.4 Étude de la variable textuelle : description

Dans votre projet de prédiction des notes d'un vin, nous allons utiliser la variable textuelle "description" pour entraîner nos modèles. Nous n'avons pas de données manquantes pour cette variable comme le montre le tableau 2.1.

Nous allons ensuite encoder numériquement chaque description pour pouvoir les utiliser dans nos modèles. Pour ce faire, nous utiliserons la librairie spacy pour nettoyer chaque description. Tout d'abord, nous convertirons tout en minuscule, nous enlèverons les chiffres et les ponctuations, nous enlèverons également les "stop words" (mots les plus courants en anglais tels que "the", "at", "which", etc.), nous normaliserons les espaces entre les mots et enfin, nous lemmatiserons chaque mot. La lemmatisation permet de représenter les mots sous leur forme canonique pour retrouver la racine de mots qui peuvent prendre des formes différentes. Notez que la tokenization (découpage des phrases en mots) sera intégrée au preprocessing de nos pipelines de modèle.

4.5 Modèle Word2Vec

Pour représenter numériquement les descriptions de nos vins, nous avons utilisé des méthodes de Word Embedding. Le Word Embedding est une technique qui permet de représenter les mots ou phrases d'un texte sous forme de vecteurs de nombres réels. Cette méthode permet d'améliorer les performances de méthodes de Natural Language Processing en se basant sur le contexte des mots. Les algorithmes de word embedding peuvent être utilisés pour décrire des mots ou des phrases entières et sont basés sur des modèles probabilistes et des réseaux de neurones. Dans ce projet, nous avons choisi d'utiliser le modèle Word2Vec avec l'implémentation Gensim pour représenter nos descriptions. Le modèle Word2Vec repose sur des réseaux de neurones à deux couches pour apprendre les représentations vectorielles des mots d'un texte en se

basant sur leur contexte et apprend un sur l'ensemble du corpus, ici l'ensemble des descriptions de nos vins.

Nous avons donc créer un transformer Word2Vec qui apprend son vocabulaire sur les données qu'il a vues (les données train) et qui puisse gérer les données inconnues (les mots dans la base test qui ne sont pas dans train) ce qui nous évite un leakage, et qui nous retourne la base de données transformée avec la variable description remplacée par sa version vectorielle dans le modèle Word2Vec appris.

4.6 Algorithmes d'apprentissage

Nous avons tout d'abord effectué un split train test de 80/20 pourcents des données.

Nous avons ensuite testé deux algorithmes pour effectuer notre classification : 1. Une regression logistique comme benchmark 2. Un algorithme de boosting XGBoost avec 'hist' comme weak classifier pour accélerer le processus d'apprentissage

5 Mesure de la performance

Nous avons mesuré la performance des deux modèles en utilisant des mesures de précision et les matrice de confusion. Les résultats obtenus pour les modèles de classification que nous avons utilisés sont cohérents avec les performances attendues. Nous avons utilisé la régression logistique comme modèle de référence, qui a donné une précision de seulement 29%. Nous avons ensuite utilisé XGBoost qui a donné une précision de 41%. On voit de plus avec les heatmaps des matrices de confusion que la regression logistique n'arrive à prédire correctement qu'une seule classe là où le classifier XGBoost se généralise beaucoup mieux, malgré une précision légèrement moins bonne pour les classes au milieu.

Le modèle de régression logistique est donc peu performant, tandis que XGBoost a des performances améliorées. Les résultats obtenus sont cohérents avec les attentes pour cette tâche de classification.

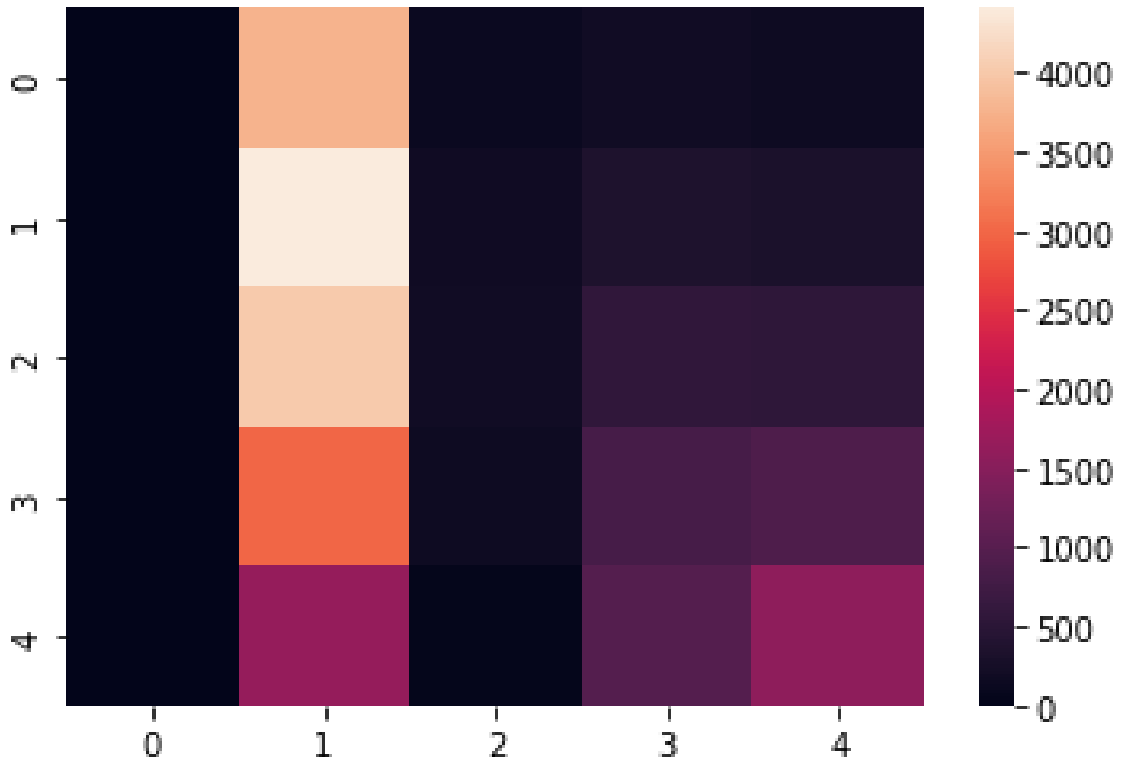
5.1 Précision par classes de la régression logistique

	0
0	0.0
1	0.2617421768303545
2	0.2853185595567867
3	0.27913372292884153
4	0.4499146272054639

5.2 Matrice de confusion de la régression logistique

	0	1	2	3	4
0	0	3770	125	200	170
1	0	4408	178	361	310
2	0	4029	206	567	548
3	0	2985	166	812	905
4	0	1649	47	969	1581

5.3 Heatmap de la matrice de confusion de la régression logistique



5.4 Performance de XGBoost

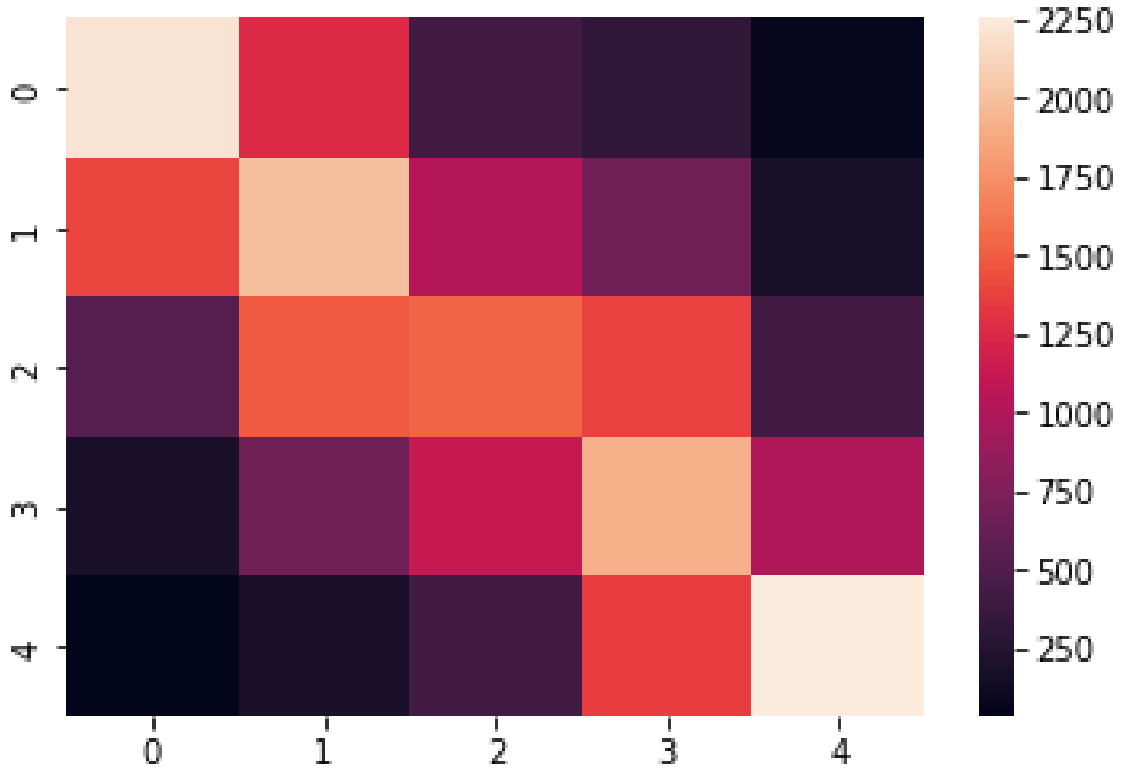
5.5 Précision par classes de XGBoost

	0
0	0.5074661153227659
1	0.3582839705618381
2	0.3385024368630926
3	0.3387125376839865
4	0.5768738807879253

5.6 Matrice la matrice de confusion de XGBoost

	0	1	2	3	4
0	2209	1254	420	314	68
1	1392	1996	1023	673	173
2	535	1484	1528	1386	417
3	183	652	1127	1910	996
4	34	185	416	1356	2255

5.7 Heatmap de confusion de XGBoost



6 Conclusion

En conclusion, nous avons réalisé un projet de prédiction des notes de vins en utilisant différents modèles, notamment la régression logistique comme benchmark et le XGBoost, qui ont permis d'obtenir une accuracy de 29% et 41% respectivement. Cependant, il reste des axes d'amélioration possibles pour optimiser la performance de notre modèle.

Premièrement, l'utilisation de techniques de recherche de grille (GridSearchCV) pour trouver les meilleurs hyperparamètres pour notre modèle XGBoost pourrait améliorer significativement les performances de notre modèle. GridSearchCV permet d'optimiser les hyperparamètres en les cherchant sur une grille prédéfinie, permettant ainsi de trouver les meilleurs paramètres pour notre modèle.

Deuxièmement, l'utilisation du one hot encoding pour représenter les variables catégorielles pourrait améliorer la qualité de notre modèle en permettant une meilleure prise en compte des caractéristiques de chaque vin.

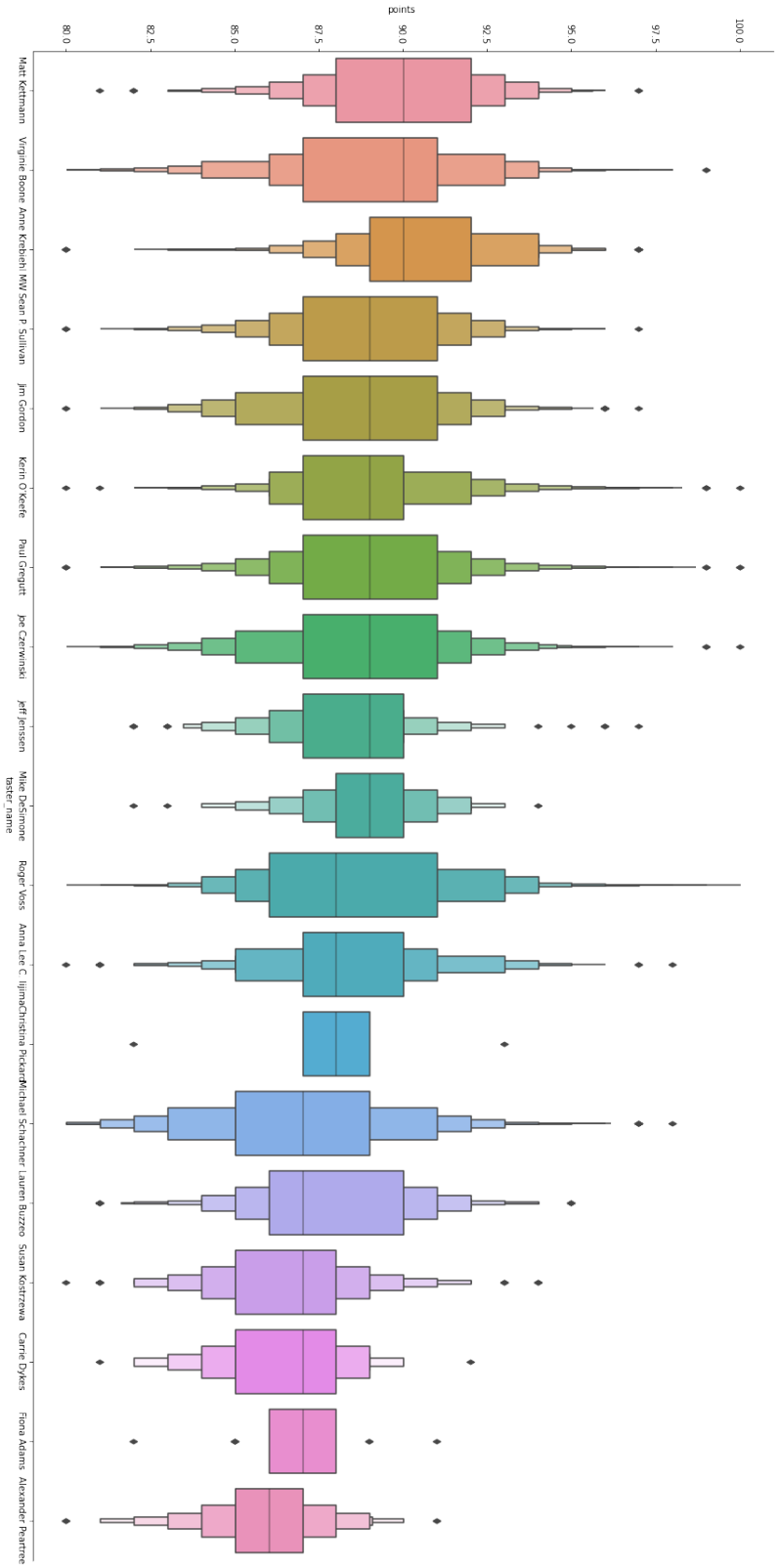
Troisièmement, l'utilisation des réseaux de neurones récurrents (RNN) tels que le LSTM avec une couche d'embedding pourrait aider à améliorer les prévisions. Les RNN permettent de capturer les relations temporelles entre les variables d'entrée, ce qui peut être utile pour prédire les notes de vins qui peuvent varier selon les années.

Enfin, l'utilisation de techniques de word embedding plus sophistiquées telles que l'Universal Sentence Encoder ou Bert peut également améliorer notre modèle. Les word embeddings sont des méthodes qui permettent de représenter les variables textuelles sous forme de vecteurs numériques, ce qui permet de mieux capturer les nuances sémantiques des variables textuelles.

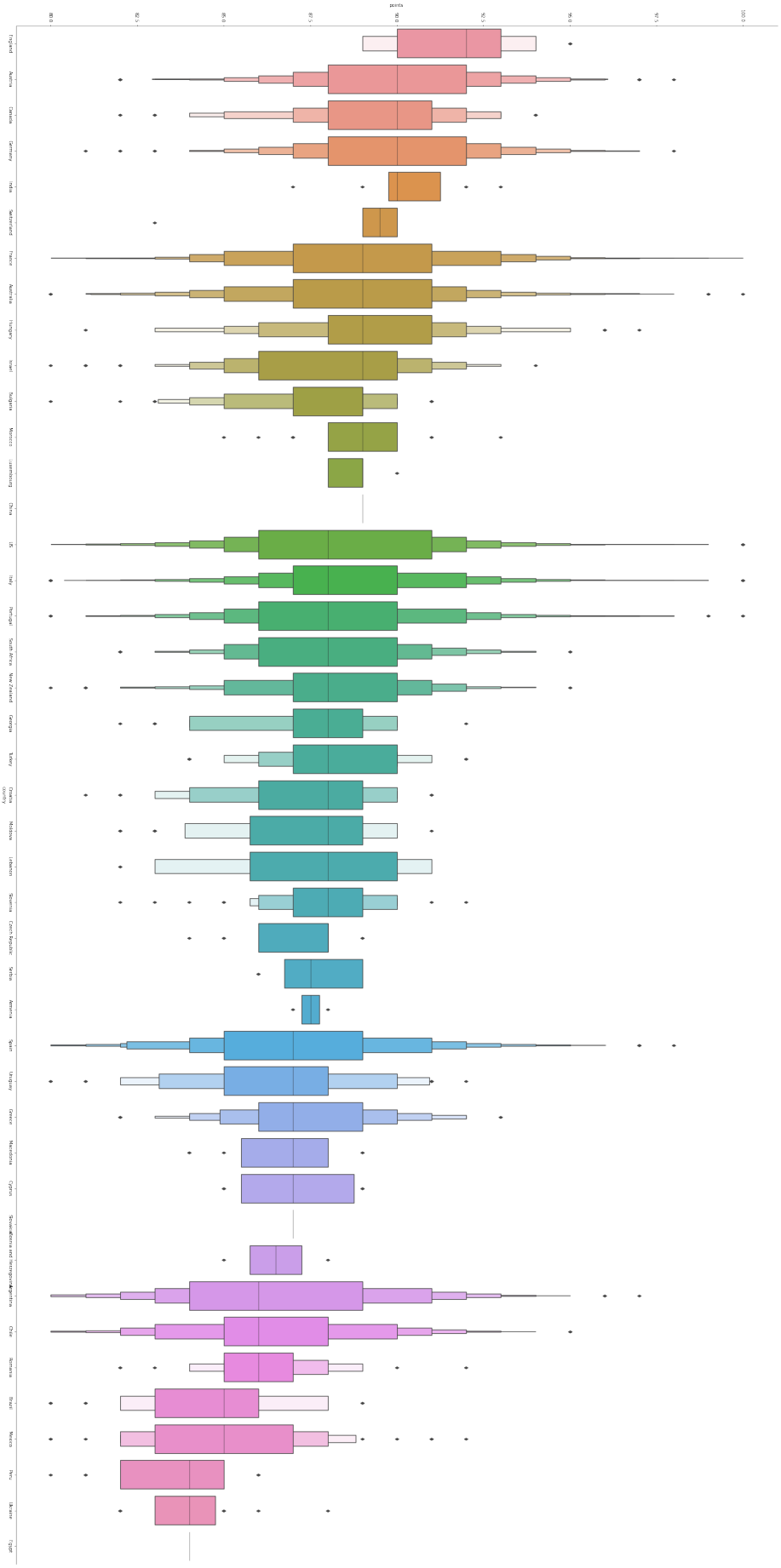
En somme, en combinant ces axes d'amélioration, il serait possible d'améliorer significativement la performance de notre modèle de prédiction des notes de vins.

7 Annexe

7.1 Relation entre les variables points et taster



7.2 Relation entre les variables points et country



7.3 Relation entre les variables price et country