L'objectif de ce projet est d'étudier, en utilisant des méthodes d'apprentissage automatique, l'impact de différents critères (notes des critiques, appelation) sur le prix d'un vin.

Pour ce faire, on s'appuiera sur le site Millesima (`https://www.millesima.fr/`), qui a l'avantage de ne pas posséder de protection contre les bots. Par respect pour l'hébergeur du site, on veillera à limiter au maximum le nombre de requêtes. En particulier, on s'assurera d'avoir un code fonctionnel avant de scraper l'intégralité du site, pour éviter les répétitions.

Instructions

Ce projet est à réaliser impérativement en binôme : les projets individuels seront pénalisés. Seules les bibliothèques spécifiquement mentionnées dans le sujet sont autorisées.
Le jury donnera zéro s'il s'aperçoit lors de la soutenance que le code n'est pas maîtrisé. Ce sera en particulier le cas si le code a été produit via une IA générative.

Premier jalon : récupération des données en `python`

Tout projet de machine learning s'appuie sur un ensemble massif de données. L'objectif de cette première partie est la récupération de ces données, et leur stockage dans le format CSV.
Cette partie est à coder en `python`.

Étudier la page d'un vin sur le site `https://www.millesima.fr/`.

On pourra prendre comme exemple la page du millésime 2016 de Château Gloria :
`https://www.millesima.fr/chateau-gloria-2016.html`, reprise dans les Figures 1, 2 et 3.

Question 1 En utilisant les bibliothèques `requests` et `Beautiful soup` comme on l'a vu dans le TP2, écrivez une fonction `getsoup()` qui prend en entrée l'URL de la page d'un vin, et renvoie la soupe correspondant à cette page HTML.

Question 2 En haut de chaque page (cf. l'encadré rouge sur la Figure 1) est indiqué le prix du vin. Attention, on sera intéressé par le prix unitaire, et pas par le prix d'une caisse : ici, 60,33€ pour une bouteille, et pas 724,00€ pour une caisse de 12 bouteilles.

Écrivez une fonction `prix()` qui prend en entrée la soupe correspondant à la page d'un vin, et renvoie son prix unitaire. Dans l'exemple, il faudra renvoyer `60.33`.

L'*appellation* d'un vin reflète sa provenance géographique : il peut s'agir d'une région entière (par exemple, *Languedoc*), d'une ville ou d'un village (par exemple, *Pauillac*), ou même d'une seule parcelle dans les cas les plus précis (par exemple, *Chambolle-Musigny Les Amoureuses*, qui correspond à une parcelle bien délimitée du village de Chambolle-Musigny).

Question 3 Écrivez une fonction `appellation()` qui attend en entrée la soupe correspondant à la page d'un vin et renvoie, sous forme de chaîne de caractères, l'appellation du vin. Cette information est disponible plus bas dans la page dans le tableau "Détails", comme indiqué en rouge sur la Figure 2. Ici, on renverra `"Saint-Julien"`.

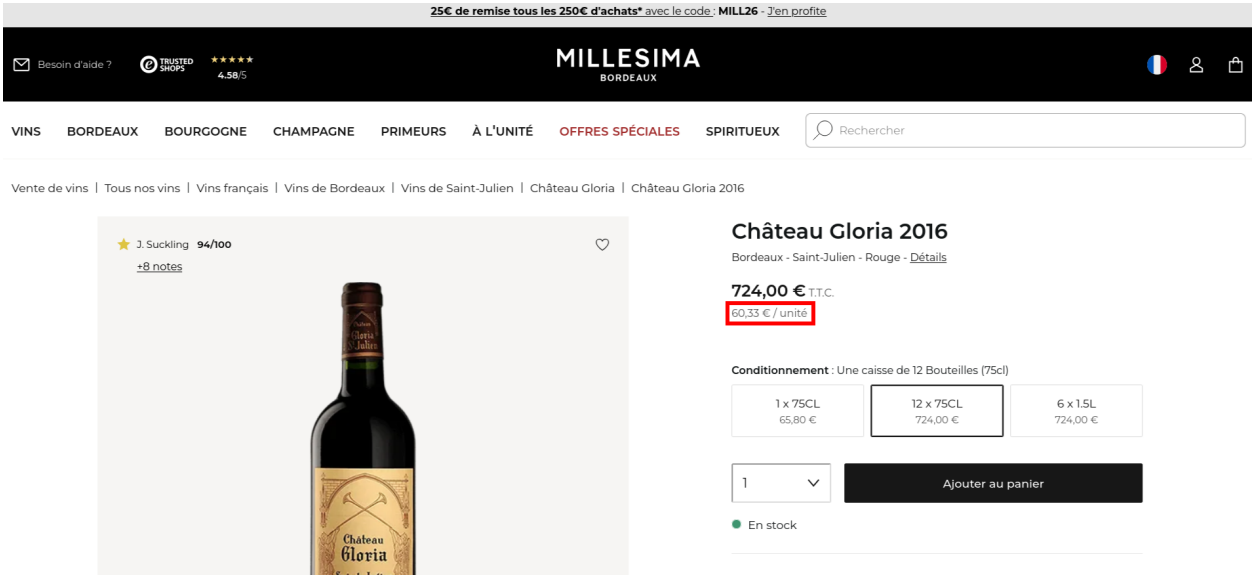

FIGURE 1 – En rouge, le prix (à l'unité) du vin.

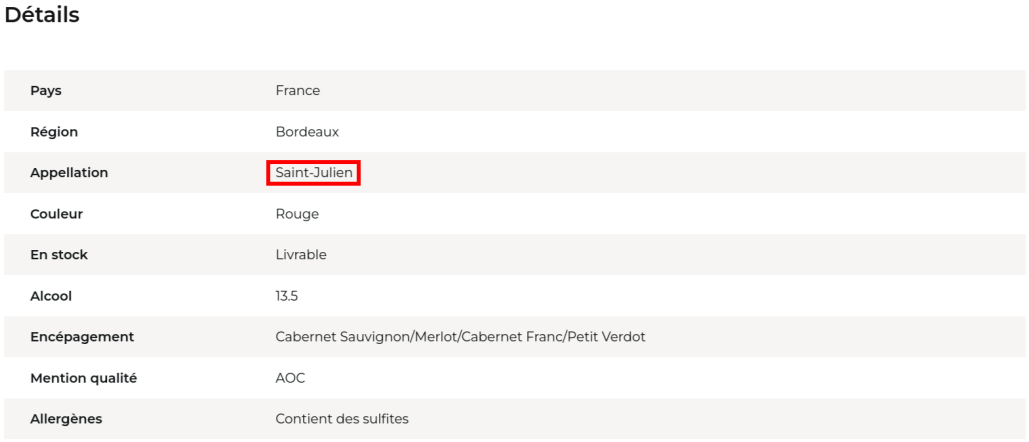

FIGURE 2 – Les détails du vin. On s'intéressera à son appellation, en rouge.

Dans le monde du vin, un certain nombre de critiques donnent des notes aux différentes cuvées. Il existe un grand nombre de ces critiques, mais on s'intéressera seulement dans ce sujet à trois des plus connus : Parker, J. Robinson et J. Suckling.

Les notes attribuées par ces critiques à la cuvée en question se retrouvent dans la section "Évaluation et notation" de la page du vin. Les trois notes qui nous intéressent apparaissent en rouge dans la Figure 3.

Question 4 Écrivez une fonction `parker()`, qui prend une soupe en argument et renvoie la note

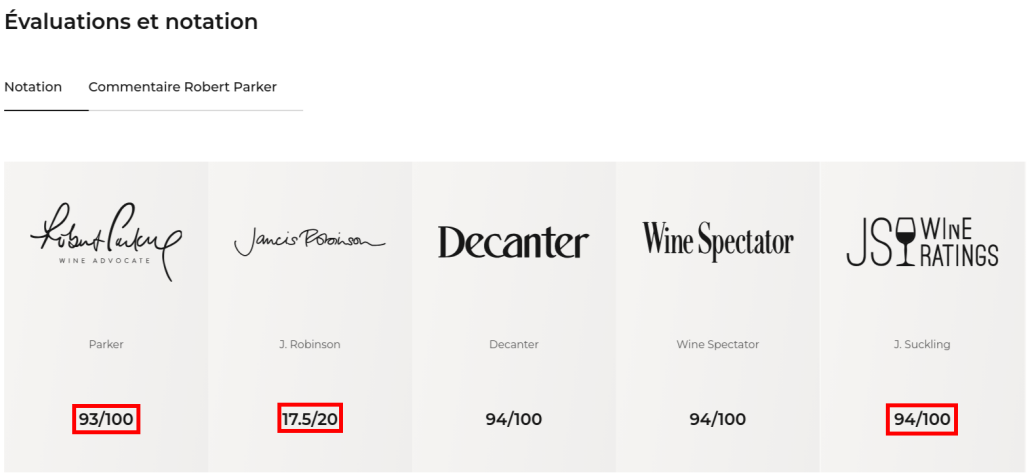

FIGURE 3 – Les notes attribuées par les critiques. En rouge, celles de Parker, J. Robinson et J. Suckling.

attribuée au vin par Parker. La réponse devra respecter les instructions suivantes :

- on omettra la base de la note (ici, `"/100"`). Dans ce cas, on renverra donc `93`.
- tous les critiques ne notent pas tous les vins : en cas d'absence de note de Parker, on renverra la constante `None`.
- la note est parfois une plage (par exemple, `"90-93/100"`). Dans ce cas, on renverra la moyenne (dans cet exemple, `91.5`).
- la note est parfois agrémentée d'un `+` (par exemple, `"96+/100"`). On ne tiendra alors pas compte du `+` (dans ce cas de figure, on renverra simplement `96`).

On testera cette fonction sur différentes pages du site pour s'assurer qu'elle couvre tous les cas de figure évoqués.

Question 5 Implémentez les fonctions `robinson()` et `suckling()` qui reprennent la question précédente, mais pour les critiques J. Robinson et J. Suckling. On fera évidemment en sorte d'éviter toute duplication de code : il faudra factoriser un maximum de code entre ces trois fonctions.

Question 6 Écrivez une fonction `informations()` qui attend la soupe de la page d'un vin en argument, et renvoie une chaîne de caractères contenant toutes les informations de l'annonce, séparées par des virgules, dans l'ordre :

`"Appelation,Parker,J.Robinson,J.Suckling,Prix"`

Sur l'annonce de Château Gloria 2016, cette fonction devra donc renvoyer :

`"Saint-Julien,93,17.5,94,60.33"`

Question 7 Nous allons limiter nos recherches aux vins de Bordeaux, détaillés sur la page `https://www.millesima.fr/bordeaux.html?page=1` et les suivantes.

En étudiant l'URL et la structure des différentes pages de résultats (il devrait y en avoir entre 50 et 100), écrivez un script qui parcourt toutes les annonces proposées par le site et appelle `informations()` sur les soupes correspondantes.

Les résultats obtenus devront être enregistrés, à raison d'une ligne par annonce, dans un fichier CSV (comma-separated values) dont la première ligne indiquera les étiquettes (sans guillemets) des différents champs :

```
Appellation,Robert,Robinson,Suckling,Prix
```

Pour la suite du projet, on travaillera à partir de ce fichier CSV local : il ne faudra surtout pas scraper le site à chaque nouvelle session de travail !