

Le but de ce projet est de déterminer la qualité d'un vin à partir de données que l'on peut obtenir en laboratoire. Ainsi on peut connaître la qualité d'un vin sans passer par un sommelier.

Extraction des données :

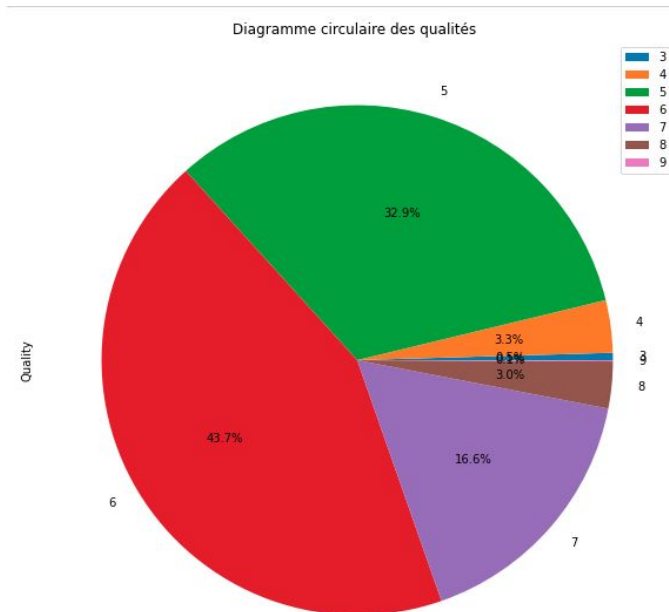
Les données sont fournies sous format csv  
Elles ont été importées dans une base de donnée Postgres.

Explication des données :

- 1 - fixed acidity: les acides qui ne s'évaporent pas rapidement
- 2 - volatile acidity: acetic acid qui lorsque sa concentration est forte, peut apporter un goût de vinaigre au vin
- 3 - citric acid: présent en petites quantités c'est un acide qui apporte de fraîcheur et du goût au vin
- 4 - residual sugar: la quantité de sucre qu'il reste après la fermentation. Il est très rare de trouver des vins avec moins d'un gramme de sucre par litre. Les vins avec plus de 45 grammes de sucre par litre sont considérés comme sucrés
- 5 - chlorides: La quantité de sel dans le vin.
- 6 - free sulfur dioxide: empêche la prolifération de microbes et l'oxydation du vin.
- 7 - total sulfur dioxide: en petite concentration le sulfure dioxide est indétectable mais au delà d'un certain seuil il devient présent dans les arômes du vin
- 8 - density: La densité du vin est semblable à celle de l'eau. elle dépend de la quantité d'alcool et de sucre.
- 9 - pH: Décrit l'acidité du vin. le vin est une boisson acide dont le ph varie en moyenne entre 3 et 4.
- 10 - sulphates: un additif qui peut influencer sur le taux de sulfur dioxide
- 11 - alcohol: Le pourcentage d'alcool présent dans le vin

**Data analyse:**

On commence par une répartition de la donnée qualité sous forme de camembert



ici on peut voir que les qualités notées 3 et 9 sont très peu représentées et donc ne sont pas assez nombreuses pour représenter une catégorie.

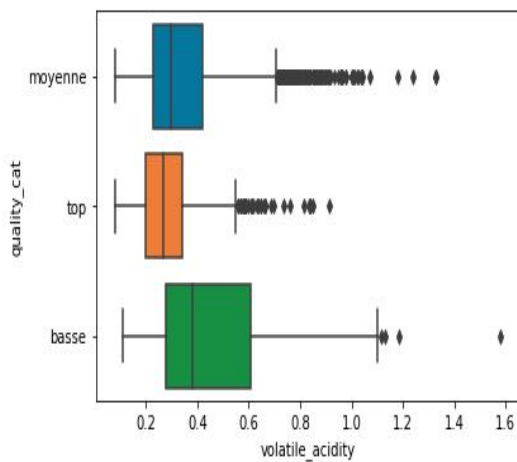
On va donc les fusionner avec la qualité la plus proche donc le 4 et le 8

Puis on va séparer pour l'analyse uniquement nous vins en 3 catégories :

très bon de qualité 7 et 8

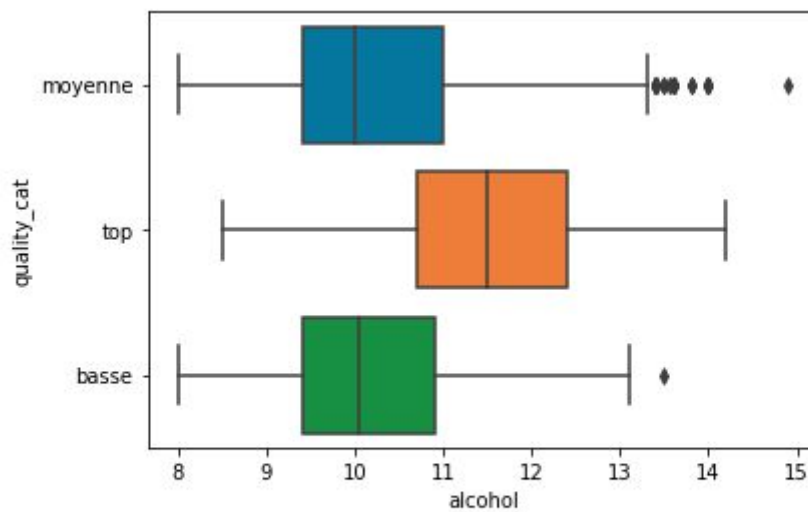
moyen : de qualité 5 et 6

basse : de qualité 4

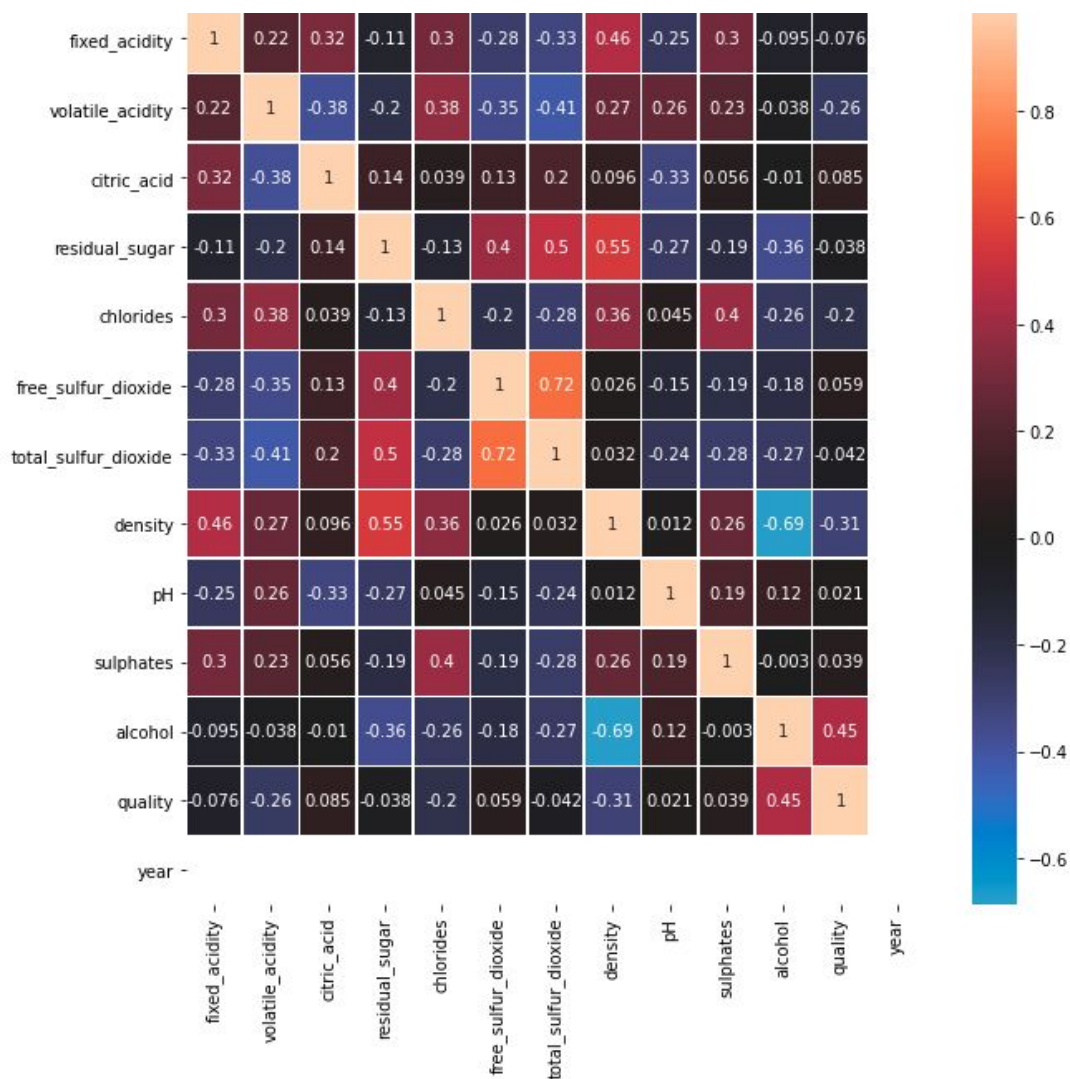


on voit que les vins de qualité ont tendance avoir une quantité d'acides volatiles (qui s'évaporent rapidement inférieur a celle des vins de moyenne est mauvaise qualité ) en moyenne plus le vin est de qualité moins il contient d'acides volatiles

les vins de tres bonne qualité ont un taux d'acides volatiles inférieur a 0.6



les vins de bonne qualité contiennent plus d'alcool que le autres



On peut voir forte corrélation entre le taux de free sulfure dioxyde et le taux de total sulfure dioxyde ce qui est prévisible et normal puisque le deuxième dépend en partie du premier  
Autre forte corrélation le taux de sucre , l'alcool et la densité du vin encore une fois cette corrélation n'est pas étonnante puisque la densité dépend de ces deux éléments  
Également la corrélation entre le ph et les différents acides n'est pas perturbante.  
La densité entre le taux d'alcool et les sucres résiduels provient probablement du fait que plus un raisin est sucré plus il produira de l'alcool à la fermentation et plus le taux de sucre résiduel est susceptible d'augmenter.

### Entraînement des modèles:

Les modèles sélectionnés pour cette analyse et leur précision

Radom Forest : 68.86 %

Kn Neighbors : 47.88 %

Decision tree: 56 %

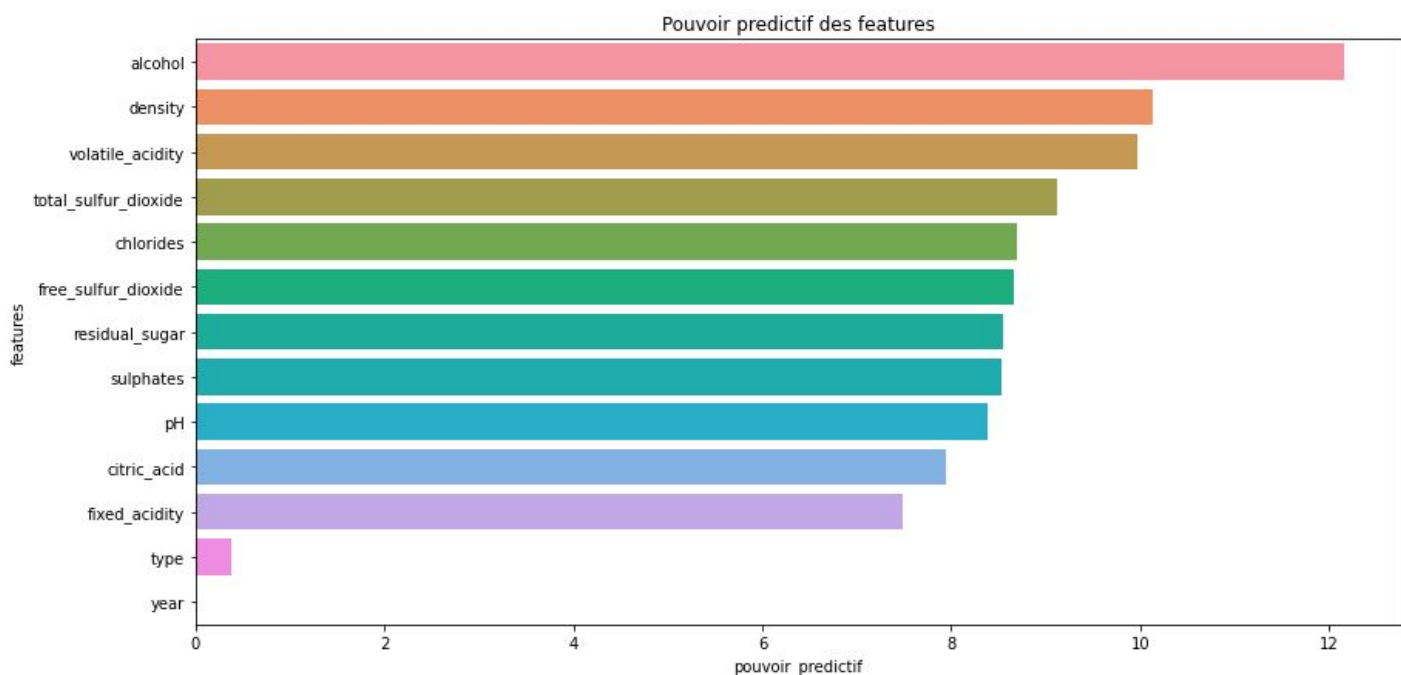
Pour l'entraînement des modèles, les données que nous avons ont été déparées en deux groupes :

données d'entraînement , données de test.

les trois algorithmes ont été entraînés puis testés.

Un quatrième algorithme a été préparé. Cependant, cet algorithme étant chronophage, il était préférable de se concentrer sur la suite du projet. Le code de cet algorithme (SVC) est néanmoins présent dans wine.ipynb, libre a vous de le déployer si vous le souhaitez.

### Importance des données pour la prédiction:



Ici on peut voir qu'en dehors du type toutes les données sont importantes pour la prédiction de la qualité. Ce qui peut amener à penser qu'un la qualité du vin dépend avant tout d'un certain équilibre dans sa composition

Api de prédiction:

L'api attend les données suivantes obtenues à partir d'un test psychochimique:

- 1 - fixed acidity (tartaric acid - g / dm<sup>3</sup>)
- 2 - volatile acidity (acetic acid - g / dm<sup>3</sup>)
- 3 - citric acid (g / dm<sup>3</sup>)
- 4 - residual sugar (g / dm<sup>3</sup>)
- 5 - chlorides (sodium chloride - g / dm<sup>3</sup>)
- 6 - free sulfur dioxide (mg / dm<sup>3</sup>)
- 7 - total sulfur dioxide (mg / dm<sup>3</sup>)
- 8 - density (g / cm<sup>3</sup>)
- 9 - pH
- 10 - sulphates (potassium sulphate - g / dm<sup>3</sup>)
- 11 - alcohol (% by volume)
- 12 - type: white ou red selon le type de vin (blanc ou rouge)

La requête envoyée à l'api doit être un POST qui cible un des trois chemins suivants (selon le modèle que l'on veut utiliser):

- <https://winepredictioncamelia.herokuapp.com/knn>
- <https://winepredictioncamelia.herokuapp.com/dt>
- <https://winepredictioncamelia.herokuapp.com/rf>

si vous héberger cette application ailleurs ou sous un autre nom de domaine il faudra changer <https://winepredictioncamelia.herokuapp.com/> dans cet exemple.

le body de la requête doit être au format json. Son schéma est le suivant:

```
{
  "fixed_acidity": "float",
  "volatile_acidity": "float",
  "citric acid": "float",
  "residual sugar": "float",
  "chlorides": "float",
  "free_sulfur_dioxide": "float",
  "total sulfur dioxide": "float",
  "density": "float",
  "pH": "float",
  "sulphates": "float",
  "alcohol": "float",
  "type": "int"
}
```

voici un exemple :

```
curl -d '{"fixed_acidity": 0.35,"volatile_acidity": 2.43,"citric acid":1.8,"residual sugar" :
2.4,"chlorides": 0.72,"free_sulfur_dioxide" : 0.5,"total sulfur dioxide":1.7,"density":1.2,"pH":
3.8,"sulphates": 4.2,"alcohol":14.1,"type": 0}' -H "Content-Type: application/json" -X POST
http://localhost:8000/dt
```

```
curl -d '{"fixed_acidity": 0.35,"volatile_acidity": 2.43,"citric acid":1.8,"residual sugar" :  
2.4,"chlorides": 0.72,"free_sulfur_dioxide" : 0.5,"total sulfur dioxide":1.7,"density":1.2,"pH":  
3.8,"sulphates": 4.2,"alcohol":14.1,"type": 0}' -H "Content-Type: application/json" -X POST  
http://localhost:8000/knn
```

## Mise en ligne:

La mise en ligne a été faite sur heroku de la façon suivante :

- heroku login => permet de se connecter à heroku
- heroku create => crée une application heroku
- heroku container:login => transmet les informations de connexion à la fonction container de heroku
- heroku git:remote -a fastapi heroku . ajoute un git heroku à notre dossier
- Login to your Heroku dashboard and open your projects.
- Go to Settings.
- Delete heroku/python from the list of buildpacks
- Then click Add buildpack → Choose "Python" → Save Changes.
- Activate your environment in your code.
- heroku container:push api : ici api est l'identifiant de conteneur qui héberge l'api
- heroku container:release api : met en production l'api
- heroku ps:scale web=1.

Pour mettre en ligne une évolution de l'application il faut :

- Modifier le code
- Créer un nouveau conteneur de l'application en lançant 'docker-compose up' (modifier le fichier docker\_compose.yml au besoin)
- publier la nouvelle version avec la commande suivante
- heroku container:push api
- heroku container:release api