

Rapport de Lif_Projet :

What's cooking? - Kaggle



Université Claude Bernard  Lyon 1

Réalisé par

Julien GIRAUD,
~~Jean-François ROLLAND~~ (parti trop tôt)
et Melisya TUTOGLU

Suivi par

Remy CAZABET




Janvier 2020

Présentation de Kaggle



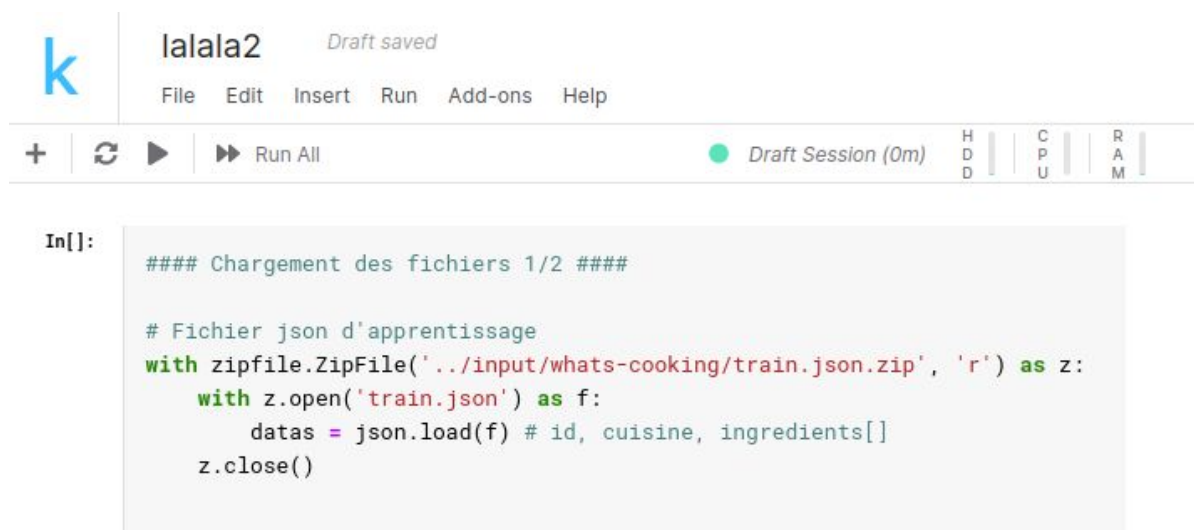
[Kaggle](#) est une plate-forme de compétition entre data-scientistes (amateurs ou non) dans le monde. Elle contient un grand nombre de problèmes à résoudre liés au machine learning. Ces problèmes sont posés par des chercheurs, des entreprises et autres organisations afin d'obtenir les meilleures solutions possibles et de mettre en concurrence des professionnels de l'information.

Kaggle leur offre ainsi la possibilité de tester leur compétences, d'essayer leurs techniques sur des jeux de données intéressants et d'améliorer leur réputation professionnelle. Kaggle leur permet aussi de remporter des prix de type mérite, argent ou job en fonction de leur classement de réussite.

17 Active Competitions		
	Deepfake Detection Challenge Identify videos with facial or voice manipulations <small>Featured · Code Competition · 3 months to go · video data, online video</small>	\$1,000,000 813 teams
	2019 Data Science Bowl Uncover the factors to help measure how young children learn <small>Featured · Code Competition · 21 days to go · video games, children, learni...</small>	\$160,000 2,720 teams
	NFL Big Data Bowl How many yards will an NFL player gain after receiving a handoff? <small>Featured · Code Competition · 5 days to go · american football, sports</small>	\$75,000 2,038 teams

Les projets proposés contiennent des données d'apprentissage et de test, et ont chacun leur propre espace de travail dans lequel les compétiteurs peuvent créer du code (exécuté en python, R, SQLight ou Julia), les partager et noter leur utilité.

Exemple de code dans le Notebook que Kaggle met à disposition des utilisateurs



```
In[ ]: ##### Chargement des fichiers 1/2 #####

# Fichier json d'apprentissage
with zipfile.ZipFile('../input/whats-cooking/train.json.zip', 'r') as z:
    with z.open('train.json') as f:
        datas = json.load(f) # id, cuisine, ingredients[]
    z.close()
```

Vue d'ensemble

Présentation du projet

Le projet [What's cooking?](#) a pour but de prédire le type de cuisine (français, indien, italien...) d'un plat à partir de la liste de ces ingrédients. Cela semble plutôt futile mais on peut imaginer qu'un site de cuisine comme [Marmiton](#) pourrait s'en servir. Lorsqu'un utilisateur entre une nouvelle recette, le site pourrait analyser les ingrédients afin de remplir automatiquement le champs *Type de cuisine*, comme fait actuellement Facebook Marketplace en analysant les photos d'articles pour prédire la catégorie de l'article.



Pour cela Kaggle met à notre disposition un immense jeu de données dans lequel nous avons un grand nombre de recette avec vingt types de cuisines différents. Ces données viennent du site [Yummly](#) qui est justement un équivalent de Marmiton aux États-Unis.

Résolution

Pour tenter de résoudre ce problème nous avons utilisé les similarités entre les recettes. Pour cela nous avons construit une matrice contenant toutes les recettes avec un DataFrame de la librairie Pandas.

Exemple de matrice contenant 3 recettes avec une base de 3 ingrédients

Recettes / Ingrédients	Ingrédient 1	Ingrédient 2	Ingrédient 3
Recette 1	contient	ne contient pas	ne contient pas
Recette 2	contient	contient	ne contient pas
Recette 3	ne contient pas	contient	contient

On remarque que les recettes 1 et 2 sont un peu similaires, les 2 et 3 sont très similaires et les 1 et 3 n'ont rien à voir.

Puis nous avons converti ces recettes en vecteur à 6 dimensions en utilisant l'algorithme [NMF](#) (librairie [sklearn.decomposition](#)). Cet algorithme permet d'obtenir une liste de 6 valeurs (vecteur) pour chaque recette, de telle sorte que plus des vecteurs sont proches plus les recettes associées sont similaires dans leur composition.

C'est dans cette partie que se fait l'apprentissage, avec le modèle NMF. On peut ensuite convertir n'importe quelle nouvelle recette utilisant ce modèle qui a appris.

Exemple de vecteurs de recettes cohérent avec le tableau précédent

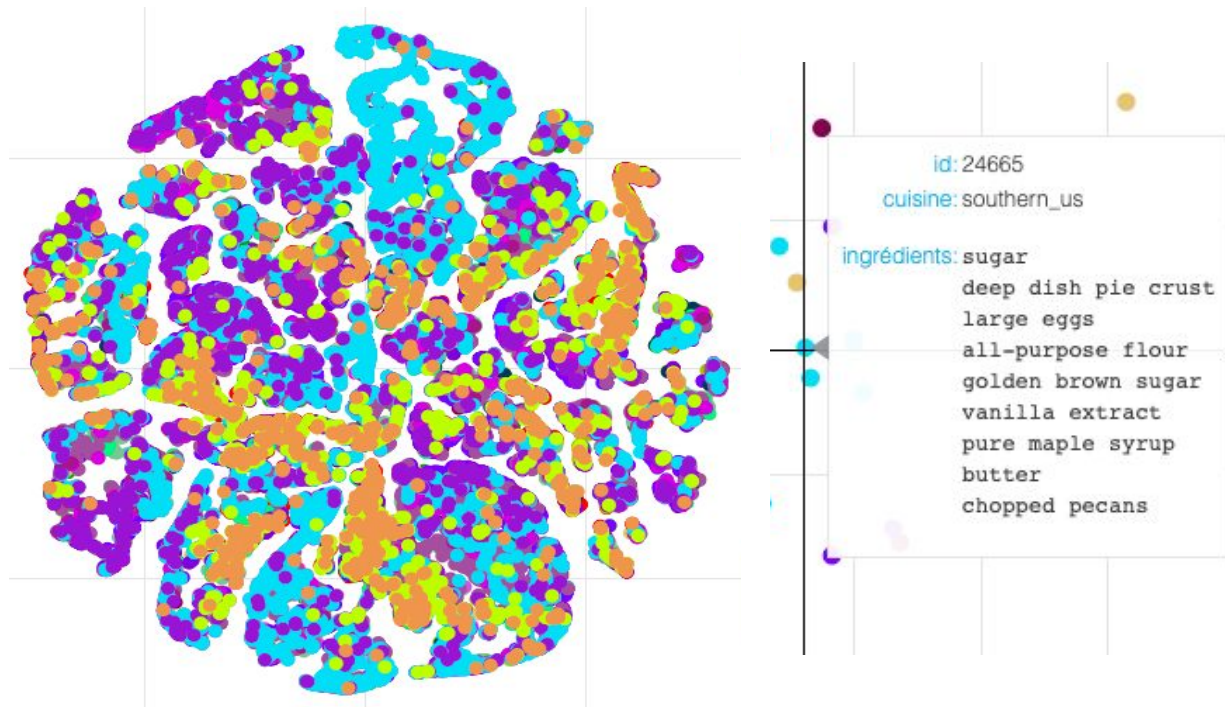
Recettes	Vecteurs
Recette 1	(24 0 0 37 0 12)
Recette 2	(22 18 10 4 31 0)
Recette 3	(4 17 8 5 34 2)

Les recettes 1 et 2 sont un peu similaires (1^{ère} dimension), les recettes 2 et 3 sont très similaires (dimensions 2, 3, 4 et 5) et les recettes 1 et 3 sont très différentes en tous points.

Enfin nous avons utilisé la librairie [sklearn.neighbors](#) pour récupérer les 10 plus proches voisins de n'importe quel vecteur à 6 dimensions. De cette façon lorsqu'on veut connaître le type d'une nouvelle recette, nous la convertissons en vecteur avec le modèle NMF puis nous relevons le type de cuisine prédominant des 10 plus proches voisins de celui-ci.

Afin de pouvoir visualiser notre travail et rendre ces données plus parlantes nous avons converti tous les vecteurs des plats en coordonnées à 2 dimensions avec l'algorithme [TSNE](#) (librairie [sklearn.manifold](#)) afin de les afficher sur une carte interactive de la librairie [Bokeh](#).

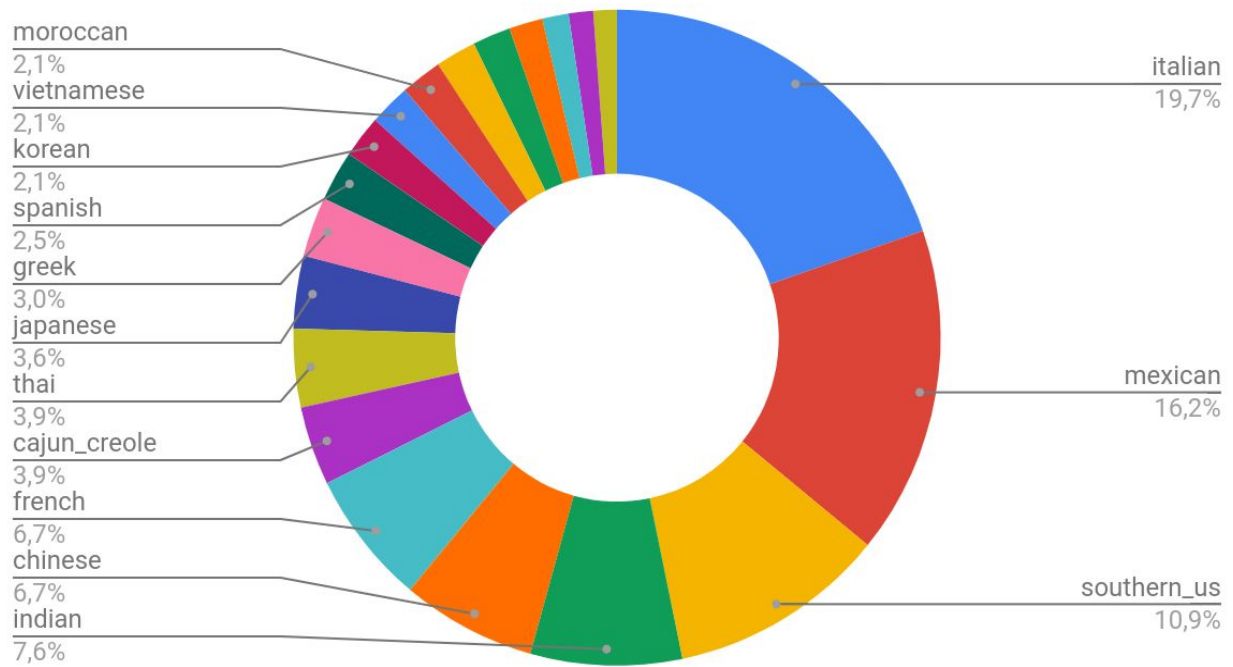
Chaque couleur représente un type de cuisine. la carte interactive nous permet de zoomer et ainsi de voir plus en détail chacune de nos recettes.



Grâce à cette visualisation, nous pouvons visionner directement quels sont les plus proches voisins de chacune de nos recettes même si la précision est supérieure en 6 dimensions.

Après avoir rendu notre résultat sur Kaggle nous obtenons un score de 40 % ce qui montre que notre code fonctionne mieux que de l'aléatoire (voir répartition des recettes ci-dessous). Cependant il y a une grosse marge de progression.

Répartition des recettes par pays



Difficultés et essais infructueux

Mauvaises pistes

Au début nous avons essayé d'utiliser les occurrences des ingrédients dans les types de cuisine pour calculer les similarités entre plats. Il a fallu que le développement soit suffisamment avancé pour lancer les tests avant que nous découvrions que cela ne fonctionne pas. Heureusement il n'y avait pas beaucoup de changements à faire pour utiliser les recettes à la place des occurrences.

Nous avons également mal interprété les vecteurs des matrices factorisées, ce qui nous a bien sûr donné des résultats qui n'avaient aucun sens.

Apprentissage des technologies

La visualisation de nos recettes s'est d'abord réalisée sur un graphique assez simple, non interactif de la librairie [matplotlib.pyplot](#). Cependant, à cause du grand nombre de recettes, cette visualisation ne nous permettait pas de comprendre et d'exploiter nos données. Ainsi nous nous sommes penchés sur une carte interactive (Bokeh) mais sa manipulation était bien plus compliquée.

En effet, nous avons eu des difficultés sur l'intégration de la légende dans la carte (toutes les données liées aux points) et l'utilisation d'une couleur par pays. Nous voulions afficher le type de cuisine, l'id du plat et les ingrédients associés à celui-ci. Le problème est que cette carte n'est pas faite pour placer un amas de points de même couleur...

La carte de Bokeh utilise une source pour intégrer la légende de chaque élément du graphique. Dans cette source il est nécessaire d'avoir des listes de même taille. Par exemple nous avons une liste de couleurs de 20 éléments car nous avons 20 types de cuisines. Cependant le tableau contenant les coordonnées des recettes n'a pas le même nombre d'éléments. L'ajout d'une couleur par pays n'est possible que si nous intégrons la couleur plus toutes les données nécessaire à la légende dans un DataFrame. Avec cette solution, nous avons par la suite exploité le DataFrame qui contient la couleur, les coordonnées, le type, l'id et les ingrédients dans la source.

	x	y	id	ingredient	color	cuisine
0	26.717314	-56.082382	31634	ice cubes\nclub soda\nwhite rum\nlime\nturbina...	#9B86D8	brazilian
1	54.312527	-16.432930	21052	eggs\nhearts of palm\nCILANTRO\ncoconut cream\...	#9B86D8	brazilian
2	-23.151327	-67.249519	623	sweetened condensed milk\nbutter\ncocoa powder\n	#9B86D8	brazilian
3	26.720102	-56.076916	26667	lime\nCRUSHED ICE\nsimple syrup\ncachaca\n	#9B86D8	brazilian
4	-37.560867	-42.324726	15482	sugar\ncorn starch\negg whites\nboiling water\...	#9B86D8	brazilian

Globalement nous avons eu beaucoup de difficultés avec les conversions entre liste Python, matrice Numpy et DataFrame Pandas.

Ressources et puissance de calcul

La matrice qui rassemble nos 39000 recettes et nos 1500 ingrédients est bien sûr gigantesque, et faire de l'apprentissage dessus demande une quantité faramineuse de RAM sans parler de la puissance de calcul. C'est le même problème lorsqu'on converti une liste de 39000 vecteurs à 6 dimensions en liste de points 2D.

Au début nous attendions quelques minutes le temps que nos pauvres processeurs fassent leur travail, mais nous avons dû revoir cette stratégie lorsque nos machines se sont mises à planter. C'est pour cette raison que nous avons utilisé Kaggle pour exécuter notre code avant d'exporter nos objets dans des fichiers avec [Pickle](#).

Et encore, dans la réduction de dimension des vecteurs de recettes nous avons réussi à faire planter notre noyau Python sur Kaggle ! Certaines des stratégies que nous avons essayé d'utiliser ont lentement rempli les 16 Go de RAM disponibles avant de déclencher une erreur. Finalement pour placer la nouvelle recette sur la carte nous avons utilisé une moyenne en fonction des 10 plus proches voisins.

Bilan du projet

Bilan humain

Tout d'abord nous souhaitons remercier M. Cazabet pour son implication dans ce projet. Ses conseils nous ont été d'une grande aide !

En ce qui concerne notre groupe nous n'avons pas eu de surprise, mis à part le départ prématuré de Jean-François. Nous sommes tous deux issus de la même formation (DUT Info à la Doua) et avons déjà travaillé ensemble. De plus à deux la répartition des tâches est très facile à organiser.

Au début du projet nous avons tous deux effectués des recherches et des tests sur nos machines respectives, puis nous mettions nos découvertes en commun lorsqu'elles semblaient intéressantes. C'est au niveau du milieu du projet que nous avons commencé à travailler sur des points opposés. Julien s'est occupé du traitement des recettes (reconnaissance des recettes) pendant que Melisya s'est occupée de la carte qui permet justement de visualiser le traitement de reconnaissance.

Bilan technique

Sur le plan technique nous avons appris beaucoup de choses ! En plus de s'être familiarisé avec Kaggle, Python et Jupyter-Notebook nous avons découvert les bases de la manipulation de données en Python (Pandas) ainsi que quelques stratégies en Intelligence Artificielle (du côté de Sklearn).

Notre projet fonctionne, même s'il pourrait fonctionner bien mieux ; et en prime nous avons une super carte des plats du monde ! Il faut dire qu'à deux, nous n'avons pas eu le temps de faire beaucoup de tests pour optimiser notre score de 40 %. Nous avons cependant établi une liste des points à ajuster pour optimiser ce résultat.

Dans les pistes d'amélioration nous avons trouvé trois points à explorer :

- Le nombre de plus proches voisins à utiliser pour supposer le type d'une nouvelle recette est actuellement à 10. Il faudrait essayer d'autres valeurs.
- Le nombre de dimensions des vecteurs de recettes est à 6 car lors de nos tests nous avons trouvé ça bien, rien ne dit que c'est optimum.
- La valeur donnée à un ingrédient présent dans un plat est de 1 dans la matrice sur laquelle on fait l'apprentissage. Cela mériterait une normalisation au prorata du nombre d'ingrédients de la recette (plus il y a d'ingrédients moins ils sont importants) et au prorata de la taille du type de cuisine (les 4 cuisines les plus répandues représentent 50 % de toutes les recettes).