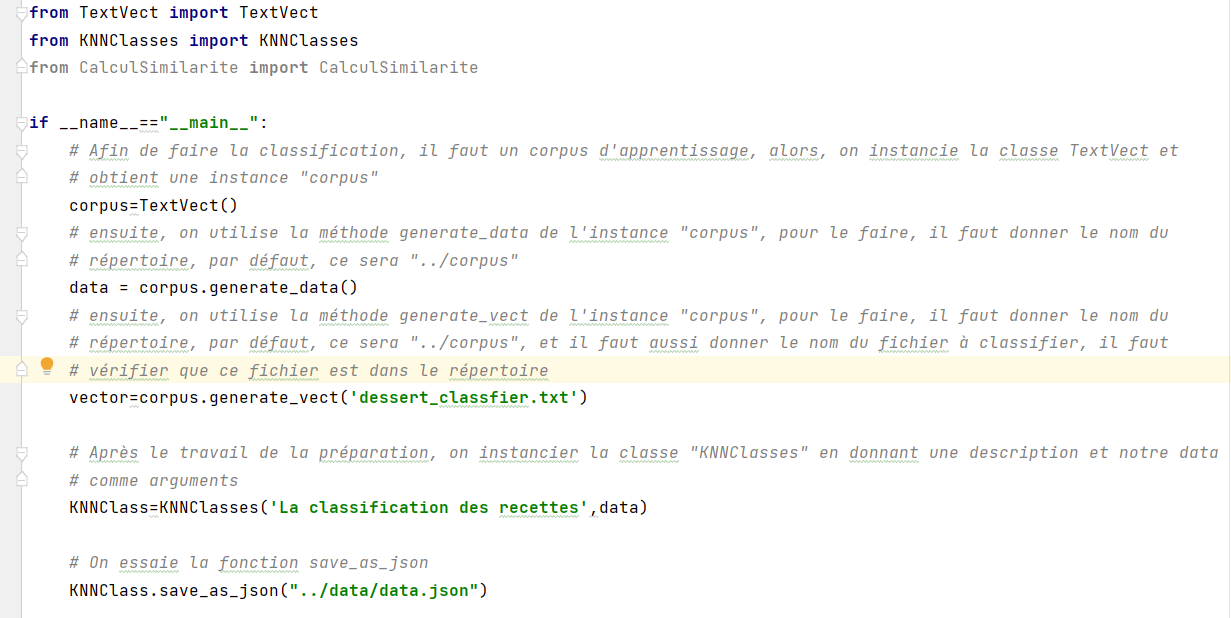
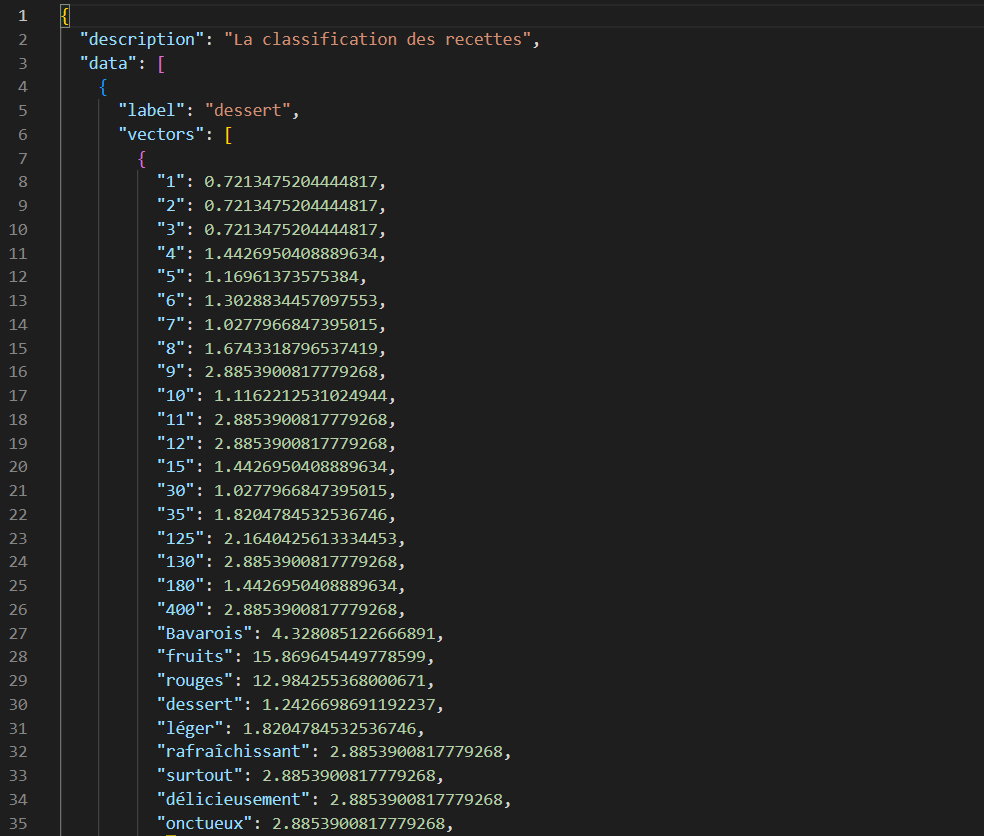
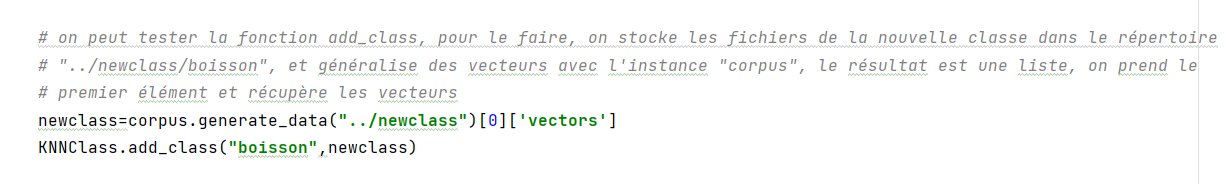
Afin de tester la fonction de nos codes, je les ai lancés dans le fichier d’entrée « main.py ». Tous les travaux de vectorisation se dérouleront avec une instance de classe TextVect qui s’appelle « corpus ». J’ai utilisé tout d’abord la méthode « generate\_data » de cette instance et obtient une base de données. Ensuite, j’ai fait une instanciation de la classe KNNClasses utilisant cette base de données, les travaux de classification suivant se dérouleront avec cette instance.



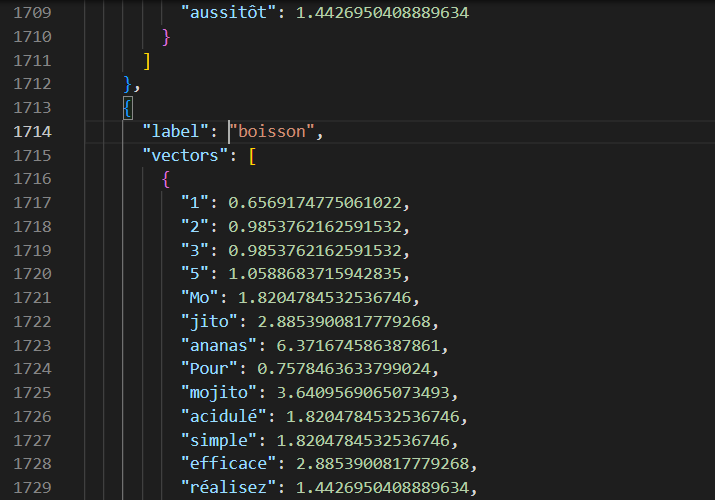
Par la suite, j’ai testé la méthode « save\_as\_json », vérifiant en même temps la réussite de la vectorisation de notre corpus d’apprentissage, et obtenu le résultat suivant. On peut voir que les données téléchargées dans le fichier « data.json » correpondent bien à ce que l’on attend.



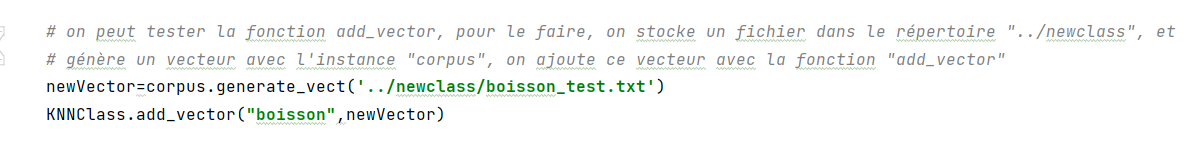
J’ai testé ensuite la fonction pour l’ajoute d’une nouvelle classe, et a obtenu le résultat suivant.



On voit bien que dans le fichier JSON, un nouvel dict est ajouté dans la liste de « data ».



De la même façon, j’ai testé la fonction « add\_vector » et un nouveau vecteur a été ajouté dans le data avec succès.

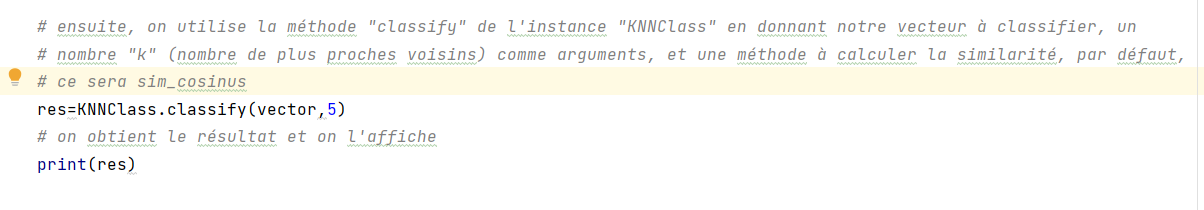




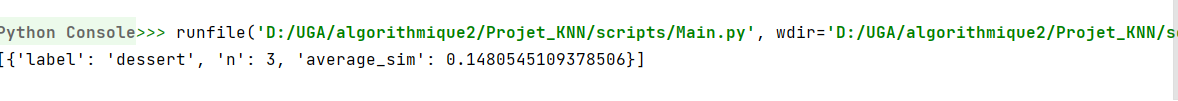
Après avoir effectué les codes pour tester la fonction « del\_class », la classe « boisson » est bien disparu et il ne reste que les classes « entrée » « plat » « dessert ».



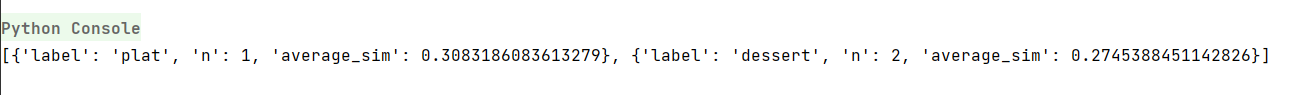
Finalement, j’ai testé la fonction « classify », et obtenu les résultats suivants :



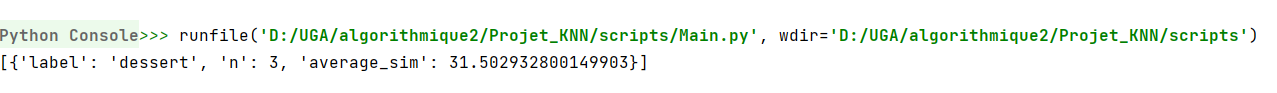
1 : sim\_cosinus



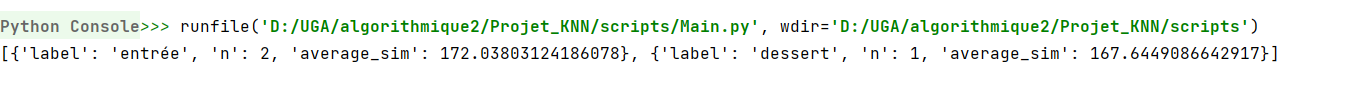
2 : sim\_PearsonCoefficient



3 : sim\_EucDist



4 : sim\_ManDist



On peut noter que, lorsque l’on a une recette d’un dessert à classifier, nos codes peuvent globalement satisfaire à notre besoin.