Preliminary Results

1-Problem Statement :

The that was mentionned in delivrable project1 revolved around create a recipee recomender engine that recommend recipees depending on ingredients the User has and then deply it through a web app.

2-Data preprocessing :

Instead of webscrapping all the recipees, I dicovered a new dataset in Kaggle that would suit most of my need which is the RecipeNLG (cooking recipes dataset) --link of the dataset :https://www.kaggle.com/datasets/paultimothymooney/recipenlg -- this dataset has more than 2 million different Recipees , of I decided to shorten a bit the Dataset so that it's not very computationnaly expensive and that we don't have way too many labes to predict to. As of the data preprocessing I deceded to one hot encode the ingredients (Limited to the most common ones) and used that as input to my model.

3-Machine Learning model

a-I started by using K-nearest neighbors using SKlearn but quickly discovered that this one was not effective due to the large amount of data that it's trained on and the fact that it doesn twork very well in higher dimensions. So thus I decided to switch to an MLP which automatical have shown better results using Pytorch . Of course at this step I used pretty classic libraries like pandas and numpy for data preprocessing.

b-for the training testing validation split I thing I am gonna still change a bit the code so instead for each recipe to have all it's ingredients as a predicting factor I ll randomize the number of actual ingredients that would predict so that it s doesn t just overfit , that would allow me to be able to do a training testing split probably 70 percents training and 30 percent testing this would useful to be able to test it and have an actual metric to see how it performed on unseen data .

c-A this point I think that there is multiple things to change before being able to give a definitive answer on how to validating , I still have many concerns about the model and the dataset it self that I am going to talk about later in this delivrable project. For now the data set is only test on its own training set which means it 's

d – I did face many chalenges , since it s my first time implementing an MLP model from scratch I had to figure out how the model would suit better my data set though I was just using a classic ht encoding method , I must say that unexpected errors aucured more than one due to a problem in matrices dimention or

4- Preliminary results :

As preliminary results from my model , since it s testing on it s own training set then we expect to have a loss thaht is pretty low and an accuracy that is pretty high and as expected my accuracy result was 82% and my loss was 0.4461

5-Next steps :

There is a lot to change and I am gonna try to breafly explain what are all the things that I am planning to do . First data preprocessing, instead of using a one hot encding technique I am thinking of Usinf A more like Word to vector method through pretrained model for Word to Vec like GloVe or Word2Vec or FastText or some other model. The second part would be to randomize the feature to create more data point for example if we have this array [olive oil , garlic , onion , curry powder , chicken Breat , salt parprica] which would predict [chicken curry] for instance then I'll want to generate 3 more list from the initial list with random number of elements and random values so that it predicts the same chicken curry , that would be usefull for creating a train and test split since in this case ur predicting point will appear more that one time.

Furthermore I would also like to change from an MLP model to An LSTM model due to it's capability of remembering context through the array . though they might be a little computationnaly expensive.