Instructions:

1. Obtain a retail database that contains information with a relationship between users products and orders.
2. Obtain a database of recipes that contains each recipe with a list of products.
3. Products must be expressed in the same language.
4. Use products from both databases and put them into a single list.
5. Input this list into word2vec.
6. Extract vectors for each product.

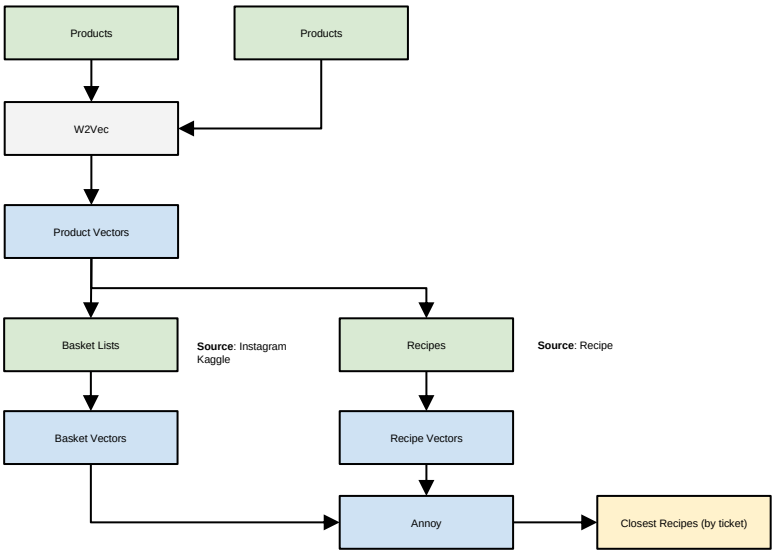
Products from different database will fall close together. Their components will be very similar on all dimensions.

There are now different ways of solving the problem of recommending a recipe to specific users:

1. Finding proximity between Baskets and Recipes.

Per ticket:

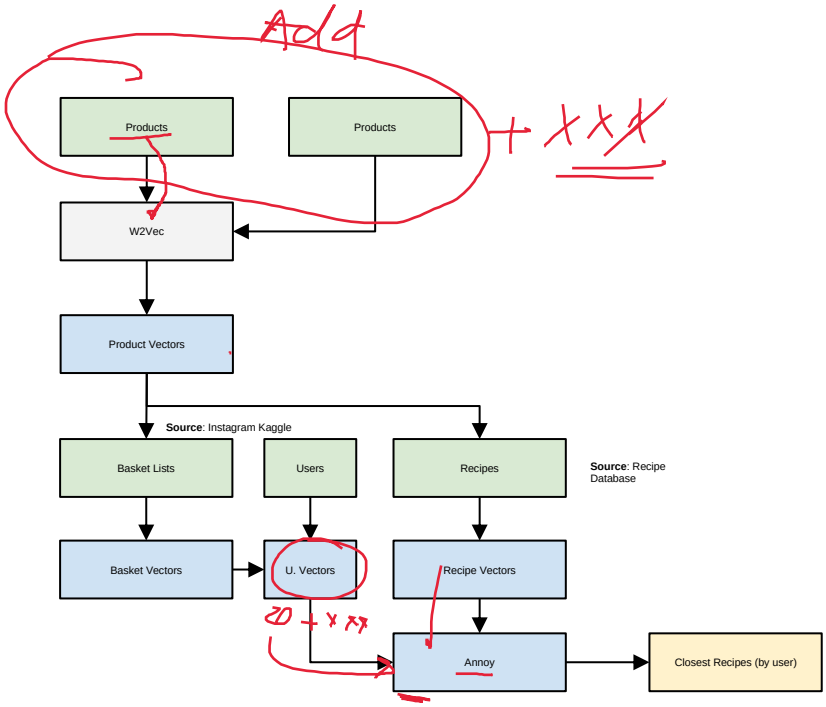
- The input to the model would be an order/basket id which would be transformed into a vector.
- Using annoy we would receive as output for example the 10 closest recipes.
- The heard of the models would be word2vec and annoy obtain distances between vectors.



Per user:

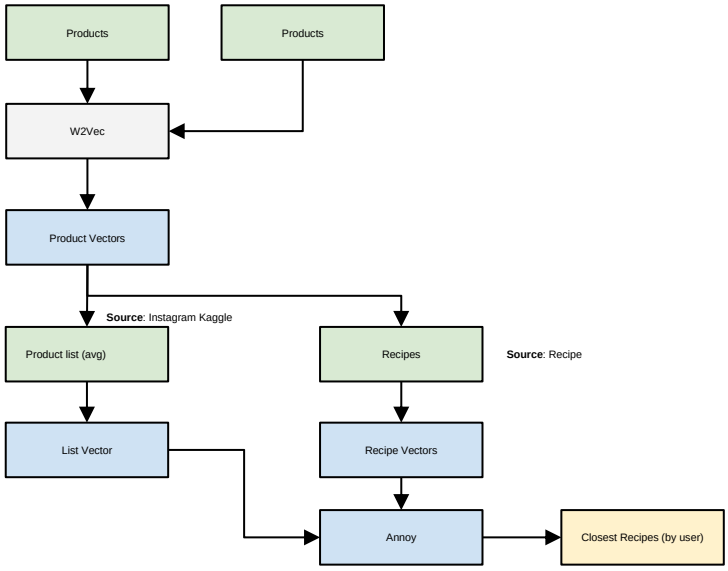
I. Input using users:

- The input to the model would be an user id which would be transformed into a vector.
- Using annoy we would receive as output for example the 10 closest recipes.
- The heard of the models would be word2vec and annoy obtain distances between vectors.



ii. Input using product list

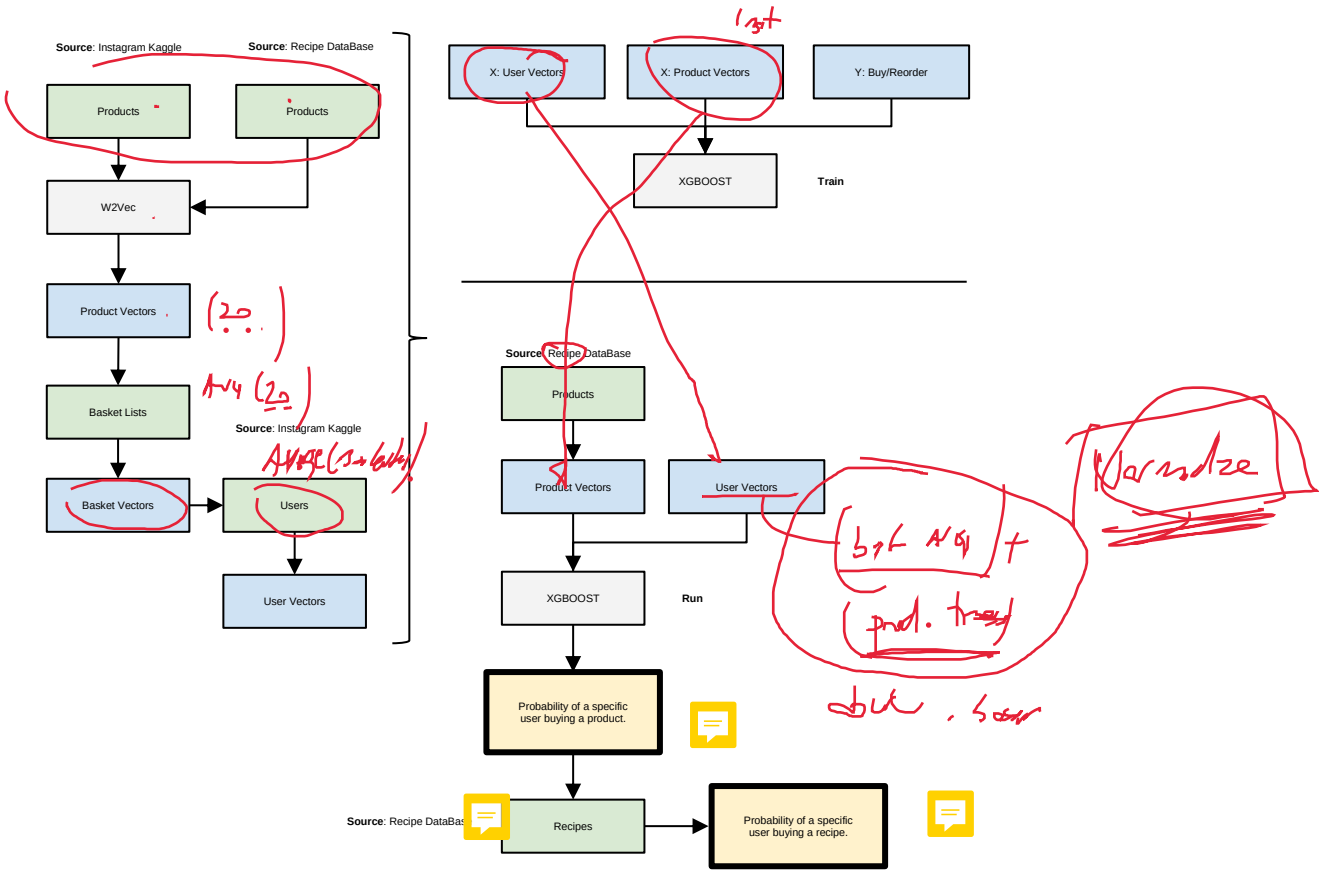
- The input to the model would be a user id which would be transformed into a vector.
- Using annoy we would receive as output for example the 10 closest recipes.
- The heard of the models would be word2vec and annoy obtain distances between vectors.



2. Finding the probability of a user buying specific products contained in a recipe.

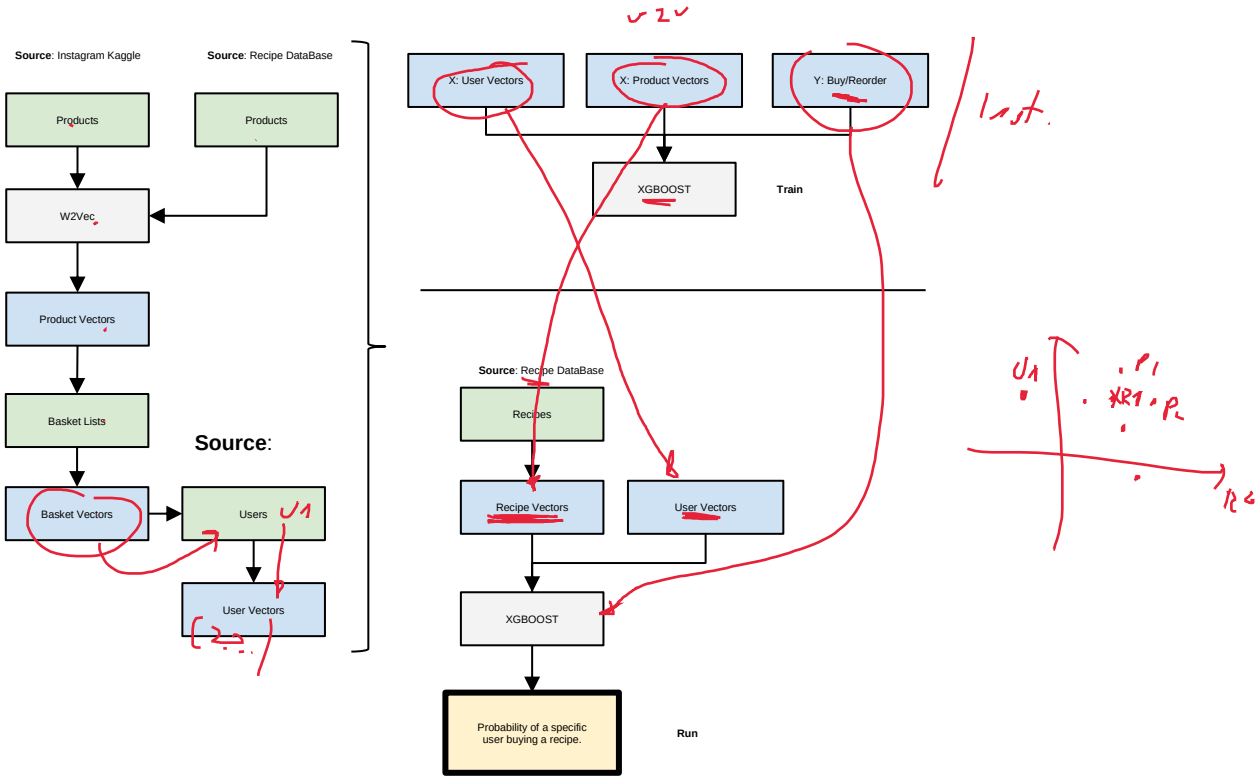
A. Buy predicting each product individually and then averaging to obtain a recipe.

- The input to the model would be a user id which would be transformed into a vector.
- The output would be the probability of the user buying a specific recipe, for all recipes.
- The heard of the models would be word2vec and xgboost to give a probability score.



B. By predicting the whole recipe.

C. The input to the model would be a user id and a recipe id which would be transformed into a vector.
D. The output would be the probability of the user buying a specific recipe.
E. The heard of the models would be word2vec and xgboost to give a probability score.


Source: Instagram Kaggle
Source: Recipe DataBase
Products
Products
W2Vec
Product Vectors
Basket Lists
Source:
Basket Vectors
Users
User Vectors
X: User Vectors
X: Product Vectors
Y: Buy/Reorder
XGBOOST
Train
Source: Recipe DataBase
Recipes
Recipe Vectors
User Vectors
XGBOOST
Probability of a specific user buying a recipe.
Run