Meal Match Algorithm:

Part 1:

Notes:

Find Plans That fulfill requirements in a moderate amount of time

* Right now, it’s just guessing until it finds some food item that has the exact delta. This is missing the step of the meal planner to recognize imbalances and start building towards that imbalance
* If not match, it should seek to add a match most similar to the delta proportions. This is what a real meal-planner would do
  + accomplish this by creating 3 new vars for each macro:
    - prot\_req\_prop: %protein of total req’s
    - fat\_req\_prop: % fat of total req’s
    - carb\_req\_prop: % carb of total req’s
    - IF not an exact match, how do we find the closest?
      * should all in the possibilities space calculate a distance in total values to the delta, and a distance in proportions to the delta in parallel as the algorithm runs?
      * Or it could iterate, adding n (starting at 1) to each req until it finds a match
      * Proportions are continuous, so a minimization would be best
      * Could a parallel machine also create random chunks of foods attempting to find the same proportion as the requirements. Then when a proportion is found, it is combined with other parallel jobs to build a final result.
    - It could first blindly sort them by total distance sqt((prot\_prop-prot\_propT)^2+…)
      * and then start building from the top of that stack.
    - Rather than just matching, it can start doing a check whether each plan is within the allowed percentage (default 5%)
  + When in production it can also store meal plans with certain macro reqs and use them immediately or riff on them s

Pseudocode:

1. Calculate protein, fat and carb proportions of final requirements
2. Calculate likely pieces based on size of requirements (compared to average ingredient macro size, expected total ingredients needed)
3. Sort main df by proportions
4. Calculate proportions needed

**Complete – now need to work out different groups and get a better mix**

Part 2: Bunching to vary meals

Create algorithm to determine if item falls into category ‘protein bar’, ‘beef jerky’, homemade meat, etc

After having foods classified into groups, write rules (1 bar per day, only in this meal, etc). Run the algorithm at the *meal* level, with target macros per meal, but more flexibility in running over. Run it piecewise, starting at breakfast.

Can have users select % from raw ingredients, branded products, restaurant food, and recipes

Groups

* Meal Type
  + Branded Products
  + Restaurant Food
  + Recipes
  + Raw Ingredients
* Food Group Classification
  + 'American Indian/Alaska Native Foods',
  + 'Baby Foods',
  + 'Baked Products',
  + 'Beef Products',
  + 'Beverages',
  + 'Breakfast Cereals',
  + 'Cereal Grains and Pasta',
  + 'Dairy and Egg Products',
  + 'Fast Foods',
  + 'Fats and Oils',
  + 'Finfish and Shellfish Products',
  + 'Fruits and Fruit Juices',
  + 'Lamb, Veal, and Game Products',
  + 'Legumes and Legume Products',
  + 'Meals, Entrees, and Side Dishes',
  + 'Nut and Seed Products',
  + 'Pork Products',
  + 'Poultry Products',
  + 'Restaurant Foods',
  + 'Sausages and Luncheon Meats',
  + 'Snacks',
  + 'Soups, Sauces, and Gravies',
  + 'Spices and Herbs',
  + 'Sweets',
  + 'Vegetables and Vegetable Products'
* Pyramid Group Classification
  + Fats, Oils, Sweets
  + Milk, Yogurt, Cheese
  + Meat, Poultry, Fish, Dry Beans, Eggs, Nuts
  + Vegetable
  + Fruit
  + Bread, Cereal, Rice, Pasta
* MyPlate Food Groups
  + Fruits
  + Vegetables
  + Grains
  + Protein Foods
  + Dairy
  + Oils
* Diet Type
  + Keto
  + Mediterranean
  + Low Carb
  + USDA recommended
* User Preference Groups
  + Favorite
  + Liked
  + May Like (collaborative filtering)
  + Neutral
  + Dislike
* Pricepoint

May need to find or make a new classification based on the usda food group pyramid

Will probably need to do something like 50% group-based 50% delta resolution on full list.

To create food type bunches, can use the ingredients lists compared to the nutrition database

The ability to bunch in multiple dimensions will have a big impact on the quality of the meals

* I.e. no more than 1 soup per day, 1 bar a day, variations in types of foods

Liked foods could also go in bunches

Pseudocode:

1. Sort all groups by proportion

2. Select based on group rules.

3. calculate new delta

4. Sort full group again.

First bunched groups: Getting food group of food item based on ingredients list

Part 3: Incorporating more recipes, fast food and fast casual meals.

Optimize plan finder algorithms using Monte Carlos and heuristics for finding likely matches (like finding items with similar proportions to requirements)

Part 4:

Optimize the mix between ‘liked’ items, ‘recommended’ items, and brand new items