# CSC111 Project Proposal: EAT EAT

Jinyeong Soul, Matthew Kuang, Minseok Jang, Ninjie Wang

March 8, 2023

## Problem Description and Research Question

A good diet is key to maintaining a healthy lifestyle, but for many people, taking the time to understanding nutrition and making suitable configurations to their meals is not viable. There are many challenges to overcome such as:

- Lack of expertise: Many people don't have a strong background in nutrition which can make it difficult to know where to start when creating a meal plan. Nutrition is a complex topic that involves understanding not just the basics of macro-nutrients (carbohydrates, proteins, and fats) but also the nuances of micro-nutrients, vitamins, and minerals.
- Personalization: Since everyone's body is unique, and what works for one person may not work for another, creating a meal plan that is tailored towards an individuals specific needs is difficult.
- Time-consuming: Researching and creating an effective meal plan can take a lot of time which people may not have the luxury of doing so.

Thus, we ask the question: **how can we analyze the individual nutritional goals of people to effectively generate a personalized meal plan tailored towards their goals?** We want to be able to generate a model that takes in basic information of the user, e.g, body weight, height, age, dietary restrictions, physical goals, etc., and create a personalized meal plan for them.

## Computational Plan

We plan to use the 'Epicurious' data set, which includes 17,736 columns of recipes, as well as their nutritional facts, preparation time, ingredients, etc. We will then classify these recipes into a respective diet. For example, 'Pear-Cranberry Mincemeat Lattice Pie' fits into a keto diet because of its low carbohydrate, high fat and protein composition. In general, this can be done by analyzing enough sample recipes for each category with several considerations.

- Define ingredients with high occurrence in each category.
- Define the average ratio between each nutritional facts for all categories/recipes.
- Calculate the error range for ingredients and nutritional facts similarity.

Once the program takes the input data, it will check their ingredients and nutritional facts ratio similarity, and then categorize the recipe into the appropriate category. After the recipes have been classified, we will take user-input data via food allergies, health goals (muscle gain, better digestion), and classify such information to a diet that suits them well. This can be done through the implementation of a decision tree classifier. In this classifier, each node will hold:

- 'gen_index': gives us the 'error' coefficient of the decision made in that node using gini index.
- Ways to refer to the child nodes of that node, which depends on the number of classifications.
- 'value': for leaf nodes, the majority classification/final choice of what type of diet.

In the ClassifierTree class, we will have a root value, and stopping conditions for tree generation (a max_depth, or a minimum_sample_size). This tree will have a method to recursively build the tree, up until the stop conditions are reached. There should also be a method to:

- Check how to split the tree again at a non-leaf node. This method will take the pre-existing errors of the nodes and return the next split that will minimize error.
- Split the tree. This can be done with an external library such as numpy or pandas.
- Calculate gini index using numpy and the formula.
- 'save' the tree as a root node, so we can continue to 'train' the tree.
- To make the prediction.
- To print the tree for debugging and visual aid.

After the user input information has been classified into a respective diet, we can use data filtering to filter out recipes with ingredients the user does wish to be included, and then generate recipes from the recipes under this diet classification.

To develop a user-friendly interface that is simple to use and learn, we will be utilizing Tkinter, a Python module designed for creating graphical user interfaces. Initially, the user will be presented with the opportunity to input their personal data via Radio button, Scale, Entry, and Check button widgets which will then be stored and used to determine their optimized meal plan. The user will then be presented with their personalized meal plan composed of unique breakfasts, lunches, and dinners for seven days, each meal accompanied with a description of their macro-nutrients and a link to their step-by-step instructional recipe. Furthermore, if the user is not satisfied with a given meal, they will be presented with a re-generate button, allowing them to generate another personalized meal for a particular day, or the option to re-generate an entirely new seven day meal plan. There will also be a menu implemented so that at any point in time, they may return to the data-inputting stage of the model and adjust their answers as they please. The finalized meal plan can then be converted into a PDF file for easier access which the user may download onto their computer.

# References


- Darwood, H. (2017). Epicurious - Recipes with Rating and Nutrition. Kaggle;Kaggle
- Ge, L., Sadeghirad, B., Ball, G. D., da Costa, B. R., Hitchcock, C. L., Svendrovski, A., Kiflen, R., Quadri, K., Kwon, H. Y., Karamouzian, M., Adams-Webber, T., Ahmed, W., Damanhoury, S., Zeraatkar, D., Nikolakopoulou, A., Tsuyuki, R. T., Tian, J., Yang, K., Guyatt, G. H., & Johnston, B. C. (2020). Comparison of dietary macronutrient patterns of 14 popular named dietary programmes for weight and cardiovascular risk factor reduction in adults: Systematic review and network meta-analysis of Randomised Trials. BMJ, m696. https://doi.org/10.1136/bmj.m696
- Types of diets. nutritioned.org. (2021, October 8). Retrieved March 6, 2023, from https://www.nutritioned.org/types-of-diets/
- Tkinter: Python interface to TCL/TK. Python documentation.(n.d.).Retrieved March 6, 2023, from https://docs.python.org/3/library/tkinter.html#: :text=The%20tkinter%20package%20(%E2%80%9CTk%20interface,well%20as%20on%20Windows%20systems.
- Sklearn.tree.decisiontreeclassifier. scikit. (n.d.). Retrieved March 6, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html