The goal is to create a shopping list that can eventually be categorized and predict each person's shopping list.

There are several ways to do this:

For example:

1. customer classification based on type of diet and age.

2. classification of customers based on type of diet, number and age of family members, e.g. single parents, etc.

3. classification of customers based on type of diet, age and time of purchase e.g. occasions, etc

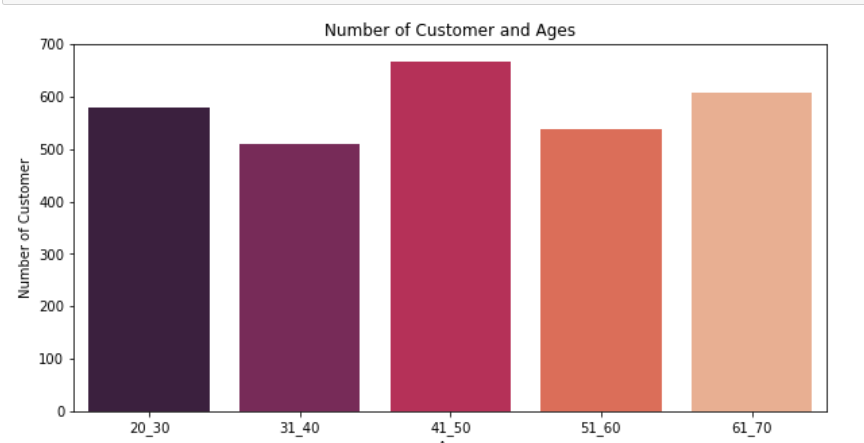
I chose the first way, 5 age groups between 20/30, 31/40, 41/51, 50/60, 61/70.

I considered 3 diet types for each age group and created random shopping lists for 50 customers.

The shopping lists finally look like this



The final simulated table



Number of customers in a specific age group

There are many ways to create a model that predicts the customer diet based on customer class. Here I have tried the three most commonly used models.

1. KNN: works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label. So the classes of customers, together with their shopping card together as points in a space and in test phase, we try to find the nearest neighbors to the test point and we take their label as predicted diet.
2. Multi-Layer:  is a class of feedforward artificial neural network (ANN).  consists of at least three [layers](https://en.wikipedia.org/wiki/Layer_(deep_learning)) of nodes: an input [layer](https://en.wikipedia.org/wiki/Layer_(deep_learning)), a hidden [layer](https://en.wikipedia.org/wiki/Layer_(deep_learning)) and an output [layer](https://en.wikipedia.org/wiki/Layer_(deep_learning)). Except for the input nodes, each node is a neuron that uses a nonlinear [activation function](https://en.wikipedia.org/wiki/Activation_function). If a multilayer perceptron has a linear [activation function](https://en.wikipedia.org/wiki/Activation_function) in all neurons, that is, a linear function that maps the [weighted inputs](https://en.wikipedia.org/wiki/Synaptic_weight) to the output of each neuron, then [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra) shows that any number of layers can be reduced to a two-layer input-output model.
3. Log Decision Tree: use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

Issues are that the simulation is not make a huge difference between for example a customer from category 1, who is vegetarian and a customer from the same category who is vegan. If a customer buys at the first sight only Fruits, vegetables and drinks, it is not easy to decide if the person is vegan, vegetarian or has a normal diet.

I tried to make some big changes in shopping list to show how important they are for accuracy. Example below is a one of the shopping cards:



If we consider the KNN model for this input the accuracy will be 82%.

For another shopping card like below



The accuracy is 84%. This means that if the shopping card has Dairy for Vegetarians, meats for Normal, it is easier for the model to predict the output.