Meal-to-Goal Mapping using Embeddings (TFIDF, Word2Vec)

Description of the dataset:

The dataset contains the following features namely

'goal_id', 'goal_short_name', 'expert_assessment', 'user_id', 'preferred_locale', 'meal_id', 'meal_type', 'meal_title', 'meal_ingredients', 'expert_explanation', 'carbs_grams', 'protein_grams', 'fat_grams', 'fiber_grams', 'calories', 'carbs_RD_explanation', 'protein_RD_explanation', 'fat_RD_explanation', 'fiber_RD_explanation', 'calories_RD_explanation'

Our main focus is on the following features:
'goal_short_name', 'expert_assessment', 'meal_title', 'meal_ingredients'

goal_short_name: This feature is basically a goal that has been set by the user that they need to accomplish

The goals can be any of the following choices

Make 1/4 of my meal grains and starchy vegetables
Make 1/2 of my meal non-starchy vegetables
Make ¼ of my meal protein
Choose low fat foods
Choose foods without added sugar
Drink water with my meal
Reduce the portion size of my meal
Choose whole grain carbs
Choose whole fruits
Choose a variety of fruits and vegetables
Choose lean proteins
Choose plant proteins
Choose vegetable fats instead of animal fats
Choose non-starchy vegetables
Drink water instead of sugary beverages

Note: For this problem, we will be focusing only on the goals that do not have any quantities (Eg: Choose low fat foods and not Make ¼ of my meal protein)

expert_assessment: This feature is basically a binary feature having the values {'not_really', 'yes'}. This is an assessment given by a physician whether a user goal has been met or not based on the description of the features {'meal_title', 'meal_ingredients'}. For this feature, ‘not_really’ indicates that the user has not met the set goal and ‘yes’ means that the user was able to meet the set goal.

meal_title: This is the tile of the meal that the user has eaten

meal_ingredients: This describes the ingredients that a particular meal eaten by the user contains

Problem Description:

Here, we need to find a way that can predict whether a particular goal is met or not based on the meal title/ingredients. For each goal, we'll be using meal title, meal ingredients and a combination of meal ingredients and meal titles as our independent variables/predictors and expert assessment as our dependent variable

Our approach:

Since our dependent variables are free form responses (text), we need to preprocess them in order to get a cleaner dataset. We perform the following pre-processing steps

Preprocessing

- Replace all non-alphabets with blank spaces
- Remove all stop words
- Perform lemmatization (Eg: Convert history, histories to histor)
- Convert the dependent variable (expert_assessment) to 1s or 0s

Computers tend to understand numbers better than text. Hence, we need to find a way to represent our dependent variables (meal ingredients/title) in the form of numbers instead of text. We use word embeddings for this. We use two types of embeddings TF-IDF (Term frequency–inverse document frequency) and Word2Vec.

TFIDF (Term frequency–inverse document frequency)

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. For more information, you can read this

- Meal ingredients

  We first convert the meal ingredients into embeddings using TF-IDF and then perform a classification task to see which meal meets the goal set by the user. We perform the classification task using logistic regression.
- Meal title

We perform a similar task as above to convert the meal titles to TF-IDF word embeddings and then perform a classification task to see which meal meets the goal set by the user. We perform the classification task using logistic regression.

- Meal ingredients + Meal title

  We create a new feature combining meal title and meal ingredients as this will tend to give more robust and trustworthy information about the meal. We perform TF-IDF embeddings and a classification task similar to the ones we carried out above using this derived feature

Word2Vec

Word2vec is a two-layer neural net that processes text by “vectorizing” words. Its input is a text corpus (dictionary of all the words) and its output is a set of vectors: feature vectors that represent words in that corpus. For more information, you can read this

- Meal ingredients

  We first convert the meal ingredients into embeddings using Word2Vec and then perform a classification task to see which meal meets the goal set by the user. We perform the classification task using logistic regression.
- Meal title

  We perform a similar task as above to convert the meal titles to Word2Vec word embeddings and then perform a classification task to see which meal meets the goal set by the user. We perform the classification task using logistic regression.

- Meal ingredients + Meal title

  We create a new feature combining meal title and meal ingredients as this will tend to give more robust and trustworthy information about the meal. We perform Word2Vec embeddings and a classification task similar to the ones we carried out above using this derived feature

Evaluation

We use accuracy as our metric to evaluate the performance of our classification model on different sets of features and embeddings. Instead of relying on a single number, we perform cross validation and run the model 1000 times to get a distribution of accuracy as a distribution will be more accurate than a single number. After plotting the distribution we get the mean of the distribution and represent it as our final accuracy.

Note: After performing all the above steps mentioned for every goal, embeddings and feature combination, we have come up with a final evaluation table as shown below

Goal		TFIDF			Word2vec		
		Ingredients	Title	Title + Ingredients	Ingredients	Title	Title + Ingredients
Lean Proteins	Train	Acc: 0.671 Std: 0.01	Acc: 0.671 Std: 0.01	Acc: 0.69 Std: 0.01	Acc: 0.622 Std: 0.006	Acc: 0.621 Std: 0.006	Acc: 0.619 Std: 0.0075
	Test	Acc: 0.677 Std: 0.024	Acc: 0.677 Std: 0.024	Acc: 0.698 Std: 0.025	Acc: 0.6212 Std: 0.025	Acc: 0.623 Std: 0.024	Acc: 0.618 Std: 0.024
Plant Proteins	Train	Acc: 0.84 Std: 0.006	Acc: 0.84 Std: 0.006	Acc: 0.84 Std: 0.006	Acc: 0.838 Std: 0.006	Acc: 0.838 Std: 0.0057	Acc: 0.839 Std: 0.006
	Test	Acc: 0.843 Std: 0.024	Acc: 0.838 Std: 0.024	Acc: 0.842 Std: 0.023	Acc: 0.84 Std: 0.023	Acc: 0.84 Std: 0.022	Acc: 0.839 Std: 0.025
Low fat	Train	Acc: 0.65 Std: 0.024	Acc: 0.658 Std: 0.02	Acc: 0.68 Std: 0.021	Acc: 0.51 Std: 0.011	Acc: 0.52 Std: 0.019	Acc: 0.515 Std: 0.01
	Test	Acc: 0.656 Std: 0.047	Acc: 0.664 Std: 0.04	Acc: 0.685 Std: 0.048	Acc: 0.48 Std: 0.056	Acc: 0.49 Std: 0.062	Acc: 0.48 Std: 0.055
Choose Whole grains	Train	Acc: 0.852 Std: 0.0082	Acc: 0.837 Std: 0.008	Acc: 0.852 Std: 0.008	Acc: 0.792 Std: 0.0075	Acc: 0.792 Std: 0.0073	Acc: 0.792 Std: 0.007
	Test	Acc: 0.86 Std: 0.03	Acc: 0.85 Std: 0.03	Acc: 0.864 Std: 0.028	Acc: 0.792 Std: 0.03	Acc: 0.798 Std: 0.03	Acc: 0.792 Std: 0.0313
Choose Whole fruits	Train	Acc: 0.812 Std: 0.012	Acc: 0.76 Std: 0.013	Acc: 0.80 Std: 0.012	Acc: 0.742 Std: 0.01	Acc: 0.743 Std: 0.011	Acc: 0.743 Std: 0.011
	Test	Acc: 0.824 Std: 0.04	Acc: 0.765 Std: 0.042	Acc: 0.814 Std: 0.0415	Acc: 0.74 Std: 0.043	Acc: 0.741 Std: 0.04	Acc: 0.74 Std: 0.045
Low Glycemic index	Train	Acc: 0.67 Std: 0.04	Acc: 0.68 Std: 0.041	Acc: 0.687 Std: 0.038	Acc: 0.529 Std: 0.016	Acc: 0.531 Std: 0.017	Acc: 0.53 Std: 0.016
	Test	Acc: 0.68 Std: 0.075	Acc: 0.65 Std: 0.07	Acc: 0.681 Std: 0.079	Acc: 0.472 Std: 0.07	Acc: 0.47 Std: 0.07	Acc: 0.471 Std: 0.078
No Added sugar	Train	Acc: 0.84 Std: 0.011	Acc: 0.83 Std: 0.01	Acc: 0.84 Std: 0.01	Acc: 0.811 Std: 0.009	Acc: 0.811 Std: 0.009	Acc: 0.81 Std: 0.009
	Test	Acc: 0.843 Std: 0.036	Acc: 0.83 Std: 0.03	Acc: 0.843 Std: 0.035	Acc: 0.80 Std: 0.03	Acc: 0.8 Std: 0.036	Acc: 0.81 Std: 0.03
Fruits and veggies	Train	Acc: 0.71 Std: 0.008	Acc: 0.648 Std: 0.009	Acc: 0.727 Std: 0.0084	Acc: 0.58 Std: 0.0058	Acc: 0.57 Std: 0.0058	Acc: 0.578 Std: 0.006
	Test	Acc: 0.72	Acc: 0.65	Acc: 0.73	Acc: 0.588	Acc: 0.57	Acc: 0.58

		Std: 0.02	Std: 0.023	Std: 0.021	Std: 0.02	Std: 0.023	Std: 0.0237