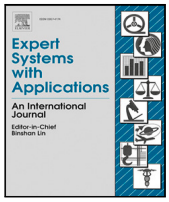
Expert Systems With Applications 237 (2024) 121507

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



MsGEN: Measuring generalization of nutrient value prediction across different recipe datasets

Gordana Ispirova [a,*], Tome Eftimov [a], Sašo Džeroski [b], Barbara Koroušić Seljak [a]

[a] Computer Systems Department, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[b] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Generalization
ML pipeline
Predictive modeling
Nutrient prediction
Recipe datasets

ABSTRACT

In this study, we estimate the generalization of the performance of previously proposed predictive models for nutrient value prediction across different recipe datasets. For this purpose, we introduce a quantitative indicator that determines the level of generalization of using the developed predictive model for new unseen data not presented in the training process. On a predefined corpus of recipe embeddings from six publicly available recipe datasets (i.e., projecting them in the same meta-feature vector space), we train predictive models on one of the six recipe datasets and test the models on the rest of the datasets. In parallel, we define and calculate generalizability indexes which are numbers that indicate how generalizable a predictive model is i.e., how well will a predictive model learned on one dataset perform on another one not involved in the training. The evaluation results prove the validity of these indexes — their relation with the accuracy of the predictions. Further, we define three sampling techniques for selecting representative data instances that will cover all parts from the feature space uniformly (involving data from all datasets) and further will improve the generalization of a predictive model. We train predictive models with these generalized datasets and test them on instances from the six recipe datasets that are not selected and included in the generalized datasets. The results from the evaluation of these predictive models show improvement compared to the results from the predictive models trained on one recipe dataset and tested on the others separately.

1. Introduction

In the recent decade, Artificial Intelligence (AI) and Machine Learning (ML) are actively leaning towards data-centric solutions — which is the systematical engineering of data required for building an AI/ML system. By definition, to build an AI/ML system, a selected problem needs to be defined, relevant and quality data obtained, a suitable algorithm chosen, and finally, the system trained and tested on the data. Until recently, most efforts have been put into the improvement of algorithms, while the process of obtaining and preparing data has mostly been understood as a no-brainer and a very simple task. However, in an age where data is at the core of every decision-making process, a shift from model- and architecture-focused AI/ML to data-focused one is more than obvious (Ng, 2022). Considering the recent literature (Strickland, 2022), the data-centric movement is slowly but surely moving the research focus towards the improvement of the data in terms of quality.

In this study, we focus on the generalization of an approach for nutrient prediction, explained in detail in our previous studies (Ispirova, Eftimov, & Koroušić Seljak, 2020, 2021; Ispirova, Eftimov, & Seljak, 2022), by doing landscape analysis to define the level of generalization between different datasets and improving the quality of data involved in the learning process. Most common in practice, predictive models are learned in a supervised fashion by using ML and are usually evaluated within the same dataset by using techniques (e.g., cross-fold validation) to test the robustness of the results. In our case, the term generalization refers to a predictive model's ability to react to new data, i.e. the model's ability to adapt properly to previously unseen data. Nowadays, generalization is crucial since the world is dynamic and changing, and having a model that is learned from static data cannot always provide good predictive results for new unseen data instances. Key techniques of achieving generalization lie in the meta-level, such as transfer learning, continual learning, and meta-learning. These are methods that aim to leverage prior knowledge or experience from related tasks or domains to improve the performance or adaptation of a model on a new task or domain. Chen, Shui, and Marchand (2021) provide a data-dependent bound for meta-learning. In another

* Corresponding author.
E-mail addresses: gordana.ispirova@ijs.si (G. Ispirova), tome.eftimov@ijs.si (T. Eftimov), saso.dzeroski@ijs.si (S. Džeroski), barbara.korousir@ijs.si (B.K. Seljak).

https://doi.org/10.1016/j.eswa.2023.121507
Received 17 February 2023; Received in revised form 18 August 2023; Accepted 6 September 2023
Available online 16 September 2023

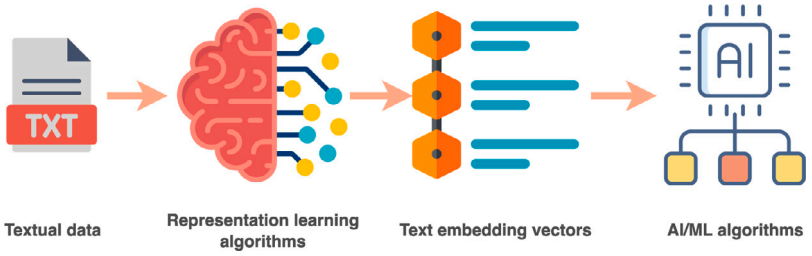


Fig. 1. Flowchart of a classical ML pipeline.

study, Guiroy, Pal, Mordido, and Chandar (2022) propose an activation-based early-stopping for meta-learning. While in Yao et al. (2021) introduce a task augmentation method for meta-learning. These papers show that meta-level techniques improve generalization in changing environments.

Our contribution: Therefore, to explore the generalizability of a predictive model we explore how a predictive model trained on a single dataset performs on other datasets. Using six recipe datasets and their recipe embeddings (i.e., representations learned from the textual descriptions of the ingredients combined with their quantities) from the predefined corpora presented in Ispirova et al. (2022), we train predictive models for predicting five nutrient values: *fat*, *protein*, *saturated fat*, *sugars*, and *sodium*. The predictive models are trained separately for each one of the datasets and then tested on the rest of the five recipe datasets. As generalization in ML is highly related to the distribution of the training data (Chung, Haas, Upfal, & Kraska, 2018; Miller et al., 2021), we introduce a generalizability index that defines the level of generalizability of a predictive model, i.e., indicates how well will the transferring of a predictive model learned on one dataset perform on another dataset. This index takes into account the similarity between the distributions of the data instances in the feature space (i.e., the distributions of the recipe embeddings of the instances across the defined clusters in the feature space) from the training data (one dataset) and the data that consists of new unseen data instances (another dataset). To test the validity of the generalizability indexes, predictive models are learned on each of the six recipe datasets for each of the five nutrients and then the models are tested on the rest of the datasets, separately. From the evaluation results, we prove that the defined generalizability index is consistent with the results. Consistency in this sense means – the higher the generalizability index, the higher the predictive accuracies.

2. Related work

A classical supervised learning prediction task involves several steps (presented in Fig. 1): (i) pre-processing data, (ii) feature extraction, (iii) training a predictive model, and (iv) evaluating the performance of the model. When dealing with textual data the second part is learning textual representations or known as text embeddings that define the features used as input data to train an ML model.

After obtaining the textual embeddings for our data, we can train predictive models that provide good solutions when the model is tweaked for the problem in hand, but the problem is that they are often over-fitted and cannot be generalized. The main issue is related to the representativeness of the training data (Binol et al., 2020; Cenikj et al., 2022; Eftimov et al., 2022; Huan et al., 2021; Killamsetty, Sivasubramanian, Ramakrishnan, & Iyer, 2021) and opens a lot of questions related to the selection of the data involved in the training process to increase the generalization of a predictive model for new unseen instances.

The work presented in Huan et al. (2021), Killamsetty et al. (2021) is in the direction of active learning, where the analysis of selecting the instances is task-oriented (e.g., classification and regression) and depends on the ML model performance. In this case, the selection of the instances is performed with some sampling technique that can also involve clustering to improve the performance of the ML model. In Huan et al. (2021), a classical multidimensional scaling technique has been used to select images that are used to train deep convolutional neural networks (CNNs) to create an efficient framework to identify rosacea lesions. Recently, a few similar studies have been published focusing on selecting diverse enough data instances that lead to robust and reproducible statistical outcomes in benchmarking studies (Cenikj et al., 2022; Eftimov et al., 2022). In Eftimov et al. (2022), a new pipeline for landscape analysis of time-series machine learning datasets has been proposed. It allows us to select a diverse portfolio of benchmark datasets by analyzing the distributions of the time-series data instance meta-representations that are coming from different data sets and the selection leads to a reproducible statistical outcome. This kind of generalization is also mentioned in a study for selecting the representative benchmark set for single-objective optimization algorithms (Cenikj et al., 2022). However, in many domains, and many types of data, as well as the Food and Nutrition domain, and recipe data, this problem, is open, and not dealt with.

Key measures and strategies used to assess and enhance the generalizability of machine learning models are: cross-validation, holdout validation, data augmentation, regularization, hyper-parameter tuning, transfer learning, ensemble methods, domain adaptation, outlier detection and handling, bias and fairness analysis and adequate data representation.

In Maleki et al. (2022) the authors discuss three methodological pitfalls that can lead to over-fitting and poor generalization of machine learning models. The first pitfall is using a small training dataset. The second pitfall is using a biased training dataset. The third pitfall is using a non-representative training dataset. The paper proposes a set of quantitative metrics for evaluating the generalizability of machine learning models:

- Holdout error: This is the error rate of the model on a held-out test dataset. A high holdout error indicates that the model is not generalizing well to new data.
- Variance: This is a measure of how much the model's predictions vary across different training datasets. A high variance indicates that the model is over-fitting to the training data.
- Bias: This is a measure of how much the model's predictions are biased towards the training data. A high bias indicates that the model is not capturing the true distribution of the data.

These metrics can be used to identify and avoid the three methodological pitfalls.

In a separate study, Zhou et al. (2020) introduced an innovative approach to segmenting manipulated images. The method commences by generating a pool of candidate segments through random sampling of the image. Subsequently, these candidate segments undergo a refinement process involving iterative merging and splitting, resulting in a set of finely segmented regions. Notably, this method demonstrates exceptional accuracy in segmenting manipulated images, even when the manipulations are subtle. The effectiveness of the technique lies in its ability to rank candidate segments based on their likelihood of

manipulation, with the highest-ranked candidates undergoing rigorous refinement until a certain level of segment homogeneity is achieved.

In this study, we are focusing on generalization of ML predictive models using domain adaptation, presented in previous studies (Ispirova et al., 2021, 2022), and adequate data representation using a quantitative indicator which we call generalizability index, presented in this study. It is important to emphasize that our focus is not on addressing the issue of over-fitting, which can be a part of the modeling approach. Instead, we aim to highlight that a model trained on one dataset can be applicable to a new dataset if their distributions in the feature space are similar.

3. Landscape analysis to define the level of generalization between different datasets

Next, we are going to explain the steps required to define the generalizability level of a predictive model to new unseen data instances that are not included in the training data. The general pipeline for the methodology — Measuring Generalization of Nutrient Value Prediction across Different Recipe Datasets (MsGEN), consists of the following steps:

1. Select a representation learning (RL) method and generate representations for the data instances from all datasets, treating them as one dataset in order to bring them to the same shared meta-feature vector space.
2. Select a dimensionality reduction technique and reduce the representations of the instances to a lower number of dimensions with enough explainable variance.
3. Select a clustering algorithm and cluster the reduced representations of the instances in order to identify groups of similar instances across all datasets based on their meta-representations.
4. Calculate how many instances from each dataset are there in each cluster.
5. Define **generalizability index**, which is an indicator of how generalizable a predictive model is i.e. how a predictive model trained on one dataset will perform on the dataset in question.

To define the generalizability index let us assume that there are n datasets available. D_i represents the ith dataset from the n datasets. R_i is the representation of the D_i dataset using the learned meta-representation further reduced with a dimensionality reduction technique. R is the dataset with the learned meta-representations of the data instances from all n datasets (i.e., all R_i datasets are merged together), again, reduced with a dimensionality reduction technique to a fixed number of dimensions.

Having the shared meta-representation space, the R dataset is further clustered into k clusters by applying a clustering algorithm. Next, the generalizability index between each pair of datasets (D_i, D_j), where D_i can be assumed as a dataset used for training the ML model and D_j as the dataset used for testing, is defined as:

$$G_{ij} = 1 - \sum_{c=0}^{c=k} \left| \left(\frac{\#Instances_{c_i}}{\#Instances_i} - \frac{\#Instances_{c_j}}{\#Instances_j} \right) \right|. \tag{1}$$

Here, $Instances_{c_i}$ and $Instances_{c_j}$ are the number of instances that belong to the cth cluster for each dataset, D_i and D_j, respectively. $\#Instances_i$ and $\#Instances_j$ denote the overall number of instances present in the datasets D_i and D_j, respectively. The idea behind the generalizability index is the similarity between the number of instances that are distributed across the clusters. The formulation of the index is a symmetrical function, which means that $G_{ij} = G_{ji}$. To define all possible learning scenarios (i.e., if more datasets are available), no matter which dataset is used for training and which for testing, we can define a generalizability matrix such as:

$$\begin{array}{c} \begin{matrix} D_1 & \dots & D_i & \dots & D_n \end{matrix} \\ \begin{bmatrix} G_{11} & \dots & G_{1i} & \dots & G_{1n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ G_{i1} & \dots & G_{ii} & \dots & G_{in} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ G_{n1} & \dots & G_{ni} & \dots & G_{nn} \end{bmatrix} \end{array} \begin{matrix} D_1 \\ \vdots \\ \vdots \\ D_i \\ \vdots \\ \vdots \\ D_n \end{matrix} \tag{2}$$

We need to point out here that because of the symmetric property of the index definition, it is enough to calculate the upper- or lower-triangular part of this matrix.

4. Results

Next, we are going to provide the details about the data involved in this study, the experimental setup — including the RL method for generating the embeddings, and all hyper-parameters used for the predictive models. Finally, we are going to present the results of the generalization of the predictive models, followed by a discussion.

4.1. Data

We have selected six publicly available recipe datasets. More details about each of the datasets are presented below:

1. Recipe1M (Marin et al., 2019) – it contains 51,500 recipes and the following data for each: recipe title (short textual description of the recipe), structured list of ingredients, recipe instruction, nutrient content of ingredients (quantity in grams of fat, protein, saturates, sodium, and sugar per 100 grams of the ingredient for each ingredient), quantity of each ingredient, units of measurement per each ingredient (household measurement system), weight in grams per each ingredient, nutrient content (quantity in grams of fat, protein, salt, saturates, and sugars per 100 grams of the recipe), and FSA traffic light labels per 100 grams.
2. Indian recipes (Indian Recipes Dataset, 2020, 2022) – it contains 6871 recipes and the following data for each: recipe URL, continuous raw text with ingredients, quantities with units of measurement in Hindi, and recipe instruction.
3. Epicurious (Epicurious Recipes Dataset, 2017; Epicurious website, 2022) – it contains 20,103 recipes and the following details for each: recipe title, recipe URL, continuous raw text with ingredients, quantities with units of measurement, calorie content, and nutritional values for protein, fat, and sodium.
4. Salad recipes (Salad Recipes Dataset, 2017) – it contains 82,243 recipes and the following details for each: recipe title, recipe instruction, continuous raw text with ingredients, quantities with units of measurement.
5. Yummly28k (Anon, 2022; Herranz, 2017; Min et al., 2016) – contains 27,639 recipes and the following data for each: recipe title, raw continuous text with ingredients, detailed nutrient information — nutrient values for 94 nutrients.
6. RecipeBox (Lee, 2020) – contains 39,802 recipes and the following data for each: recipe title, continuous raw text with ingredients, quantities with units of measurement, and recipe instructions.

4.2. Evaluation pipeline

To evaluate the generalizability of a predictive model, i.e., the correlation between the generalizability index and the predictive model's performance, we used the MsGEN pipeline presented in Fig. 2.

The pipeline consists of several steps:

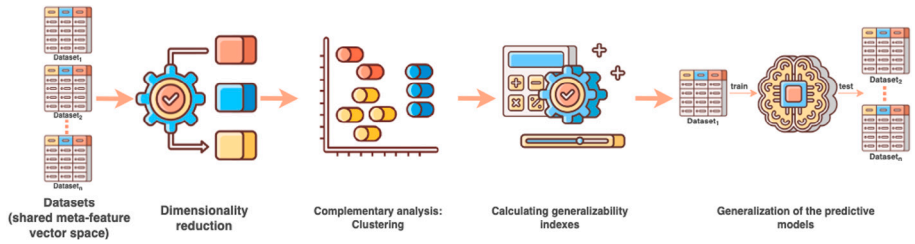


Fig. 2. Flowchart of the methodology.

- Pre-process the datasets and normalize them to the same format. We are not going to provide details here. For readers that are interested in this step, please check (Ispirova et al., 2022).
- Calculate the embeddings/vector representations for the data instances and bring them to the same meta-feature vector space. For this study, we did not perform this step from scratch, but we have utilized a predefined published corpus of ingredient embeddings presented in Ispirova et al. (2022). We need to mention here that the predefined corpus of ingredient embeddings is available in different formats depending on the textual RL method used (e.g., Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013); GloVe (Pennington, Socher, & Manning, 2014); Doc2Vec (Devlin, Chang, Lee, & Toutanova, 2018; Le & Mikolov, 2014)) as well as its hyper-parameters. Further, we used the ingredient embeddings to generate the embeddings/representations for the recipes in each dataset. For each recipe, its embedding is calculated by fusing the embeddings of its ingredients with a domain-specific heuristic defined in Ispirova et al. (2021).
- Perform predictive modeling by training predictive models on a single dataset and test the models on all the remaining datasets. In our case, we trained single-target regression models for each of the five nutrients: fat, protein, saturated fat, sugars, and sodium. The models were trained using different regression algorithms such as Linear, Lasso, Ridge, ElasticNet, Decision Tree, Random Forest, and Multilayer Perceptron Neural Network regression. Different regression models using the above-mentioned methods were trained for all the variants of the embeddings that exist in the predefined corpus. In each learning scenario that is a combination of embeddings learned by a particular RL method and one of the above-mentioned regression methods, hyper-parameter tuning is performed for the selected regression model by applying GridSearchCV and RandomizedSearchCV from the *scikit-learn* library in Python (Pedregosa et al., 2011). Finally, for each scenario, we select the best hyper-parameters, train the model on a single dataset, and further evaluate that model on the other five datasets.
- The regression models were evaluated by calculating a domain-specific accuracy. For each nutrient prediction of each data instance, we calculate if the error (the difference between the actual and the predicted nutrient value) is in the tolerance level for the specific nutrient. These tolerance levels are defined by the European Commission Health and Consumers Directorate General in 2012 (European commission health and consumers directorate-general, 2012), with the aim of providing advised recommendations for the calculation of the acceptable differences between quantities of nutrients on the label declarations of food products and the ones established in Regulation EU 1169/2011. This allows us to calculate how accurate our predictions are. More details about the tolerance levels and the domain-specific accuracy measure are presented in Ispirova et al. (2020, 2021).
- Select the best-performing RL method based on the domain-specific accuracy across all regression models and all datasets. For this purpose, we designate the settings of the embedding algorithms that yielded the top five accuracies for each dataset and each target and select one that appears most frequently. In total, with the above-explained combinations of learning scenarios, we ended up with 3660 models trained: 1440 Word2Vec, 720 GloVe, 1440 Doc2Vec, and 60 BERT models. The chosen datasets of embeddings are the embeddings generated with the Wor2Vec embedding algorithm and the following parameters: dimension 100, sliding window 3, architecture $CBOW$, and merging heuristic $average$.
- Perform dimensionality reduction of the selected embeddings. We selected the Principal Component Analysis (PCA (Wold, Esbensen, & Geladi, 1987)) dimensionality reduction technique as it is one of the most commonly used ones. PCA tries to preserve the global properties (eigenvectors with high variance) while it may lose low-variance deviations between neighbors. We used the first three puritanical components to project the original embedding. The explained variance was around 80%. In addition, we can easily visualize the data in 3D.
- Perform clustering of the reduced embeddings. In this step, we use the k-means (MacQueen, 1967) algorithm.
- Determine the percentage of data instances from each dataset in each cluster.
- Calculate the generalizability indices to analyze the generalizability of the predictive models between different datasets.

4.3. Landscape analysis results

4.3.1. Dimensionality reduction

The high-dimensional vector representations of the recipes from all six recipe datasets using the selected combination of RL method and its settings, undergo a PCA dimensionality reduction and are transformed from 100 dimensions into three dimensions. The distributions of the reduced three-dimensional vectors of the recipe embeddings from the six datasets in the same vector (meta-feature) space are presented in Fig. 3, for each dataset separately. From this figure, we can see how each of the datasets is distributed in the feature space, and how different their distributions are. Fig. 4 presents the distribution of the reduced embeddings for each dataset together in the same feature space. From this figure, it is very noticeable how the Salad recipe dataset is more widely distributed compared to the rest of the datasets.

4.3.2. Clustering

After reducing the 100-dimensional embeddings to 3-dimensional embeddings, the embeddings are clustered with the k-means clustering method using the $sklearn$ library in Python (Pedregosa et al., 2011). The number of clusters is chosen using the silhouette analysis method (which assesses the quality of clustering), and comparing the average silhouette scores for various values of k – the numbers of clusters. In other words, it establishes how well each object fits within its cluster – a high average silhouette width indicates good clustering, i.e. more separated clusters. The optimal number of clusters k is the one that maximizes the average silhouette over a range of the chosen possible values (Kaufman & Rousseeuw, 1990).

In our study, we fixed the number of clusters from 3 to 12, and for each one, we calculate the average silhouette value. Fig. 5 presents the

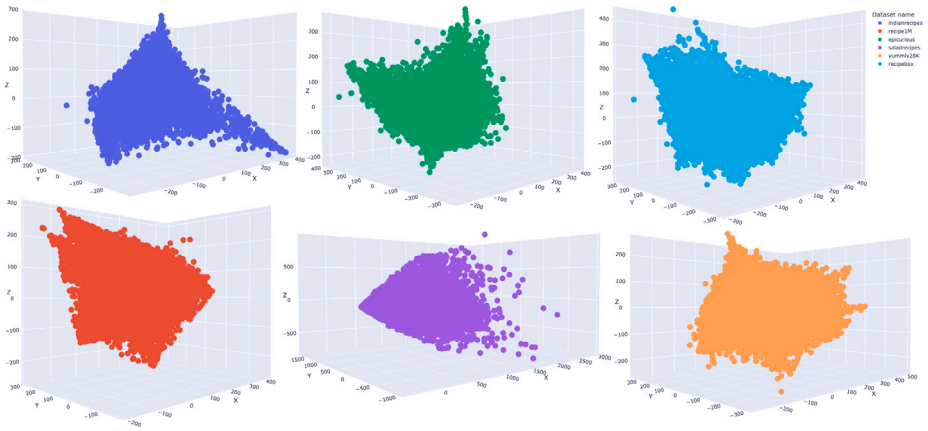


Fig. 3. Reduced recipe embeddings for all six datasets obtained with the Word2Vec algorithm (architecture: CBOW, dimension: 100, sliding window: 3 merging heuristic: average) presented separately in the same feature space.

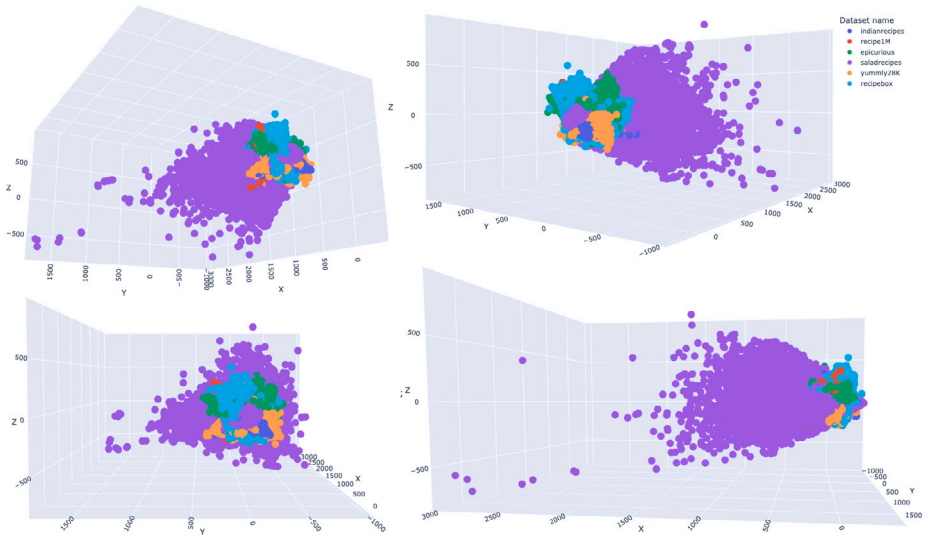


Fig. 4. Reduced recipe embeddings for all six datasets obtained with the Word2Vec algorithm (architecture: CBOW, dimension: 100, sliding window: 3 merging heuristic: average) presented together in the same feature space.

Table 1
Number of instances in each cluster.

Cluster	0	1	2	3	4	5	6	7
Number of instances	44,576	9,363	33,971	24,751	7,387	32,274	1,966	65,477

Table 2
Generalizability matrix.

	Indian recipes	Recipe1M	Epicurious	Salad recipes	Yummly28K	Recipe box
Indian recipes	1.00	0.38	0.31	0.34	0.61	0.43
Recipe1M	0.38	1.00	0.87	0.08	0.73	0.85
Epicurious	0.31	0.87	1.00	0.07	0.66	0.84
Salad recipes	0.34	0.08	0.07	1.00	0.15	0.19
Yummly28K	0.61	0.73	0.66	0.15	1.00	0.76
Recipe box	0.43	0.85	0.84	0.19	0.76	1.00

curve of the average silhouette for values of the number of clusters k from 3 to 12. From it, it follows that the maximum silhouette score is achieved for $k = 8$, which is around 0.585. This indicates that in our case the clusters will have good separation. Fig. 6 presents the clustering results.

Table 1 presents the number of instances (i.e., recipes) that belong to each cluster, while Figs. 7 and 8 present two heat maps depicting percentages of each dataset per cluster and the number of instances per cluster from each dataset respectively.

Next, we calculate the generalizability indexes between all pairs of datasets according to the distributions of each dataset across the clusters by using Eq. (1) and generate the generalizability matrix (presented in Table 2).

To analyze the correlation between the generalizability indices and the performance of the predictive models, we evaluate the predictive

Table 3
Results from the predictive models trained on each one of the recipe datasets separately and tested on the rest obtained with the Word2Vec embeddings merged with the domain heuristic. *Max* is the maximum accuracy obtained, and *Average* is the average accuracy obtained.

Train dataset	Test dataset	Fat		Protein		Saturated fat		Sugars		Sodium	
		Max	Average	Max	Average	Max	Average	Max	Average	Max	Average
Indian recipes	Recipe1M	32.36	26.22	25.39	19.80	70.30	61.58	28.62	18.35	35.98	32.62
	Epicurious	31.53	26.22	26.06	20.76	72.72	65.31	31.85	19.63	37.69	30.77
	Salad recipes	3.32	3.02	22.72	13.53	45.81	29.05	12.54	4.89	24.25	21.57
	Yummly28K	47.01	41.16	43.76	38.43	89.95	79.01	45.52	33.64	26.28	22.27
	Recipe box	36.34	31.06	28.93	23.75	72.64	65.76	27.51	16.96	34.06	26.64
Recipe1M	Indian recipes	45.72	25.36	56.52	53.99	72.49	55.97	37.95	21.67	42.91	36.23
	Epicurious	64.44	44.44	81.28	77.49	93.83	78.47	60.38	42.63	65.93	53.41
	Salad recipes	17.32	14.21	53.83	47.41	46.26	23.38	12.44	12.10	27.90	22.49
	Yummly28K	36.62	17.60	47.55	42.04	67.51	47.20	32.06	15.33	15.61	4.78
	Recipe box	45.44	27.69	65.21	60.83	72.19	57.29	36.54	19.74	44.21	29.82
Epicurious	Indian recipes	59.83	56.07	57.81	55.29	86.52	84.92	61.38	51.69	95.83	83.99
	Recipe1M	78.29	74.53	82.67	80.48	83.91	82.63	78.54	70.68	95.76	86.87
	Salad recipes	29.24	20.47	58.83	52.69	61.54	57.23	41.68	27.79	88.98	79.53
	Yummly28K	54.67	48.08	48.36	44.49	85.16	82.67	57.59	45.00	93.81	76.05
	Recipe box	57.56	54.41	67.39	65.28	83.86	82.47	52.55	44.94	96.28	89.06
Salad recipes	Indian recipes	74.82	72.87	83.51	81.42	90.11	88.37	69.83	67.85	99.87	99.87
	Recipe1M	34.00	31.59	46.14	35.84	51.98	50.96	33.51	30.13	67.80	67.68
	Epicurious	32.05	28.30	41.99	33.02	54.56	53.69	34.42	29.65	67.93	67.85
	Yummly28K	32.02	26.79	25.81	18.74	56.23	55.32	33.13	24.78	68.03	67.29
	Recipe box	30.14	25.88	30.14	22.58	55.72	54.88	28.39	18.47	68.07	67.19
Yummly28K	Indian recipes	60.86	56.62	65.70	64.74	88.59	86.48	68.33	65.37	99.12	92.37
	Recipe1M	58.25	54.11	68.25	67.26	84.34	82.77	64.67	62.56	97.91	93.42
	Epicurious	59.56	55.23	67.50	66.66	86.02	84.90	66.36	63.93	98.05	94.62
	Salad recipes	30.42	21.30	42.36	40.21	68.40	64.10	50.35	43.95	98.38	84.88
	Recipe box	78.67	74.99	86.97	86.19	86.6	85.24	83.4	81.24	98.08	95.26
Recipe box	Indian recipes	40.28	35.87	58.55	54.78	89.20	86.25	68.99	66.42	98.99	89.71
	Recipe1M	63.39	56.87	83.81	80.53	86.33	83.86	86.62	84.38	98.12	91.19
	Epicurious	41.45	35.20	67.13	64.25	87.69	85.15	66.34	64.41	98.20	92.59
	Salad recipes	3.95	2.37	61.88	53.82	63.82	58.43	47.72	42.82	98.55	94.83
	Yummly28K	25.10	21.64	47.92	42.69	88.76	84.32	71.51	67.20	98.55	85.31

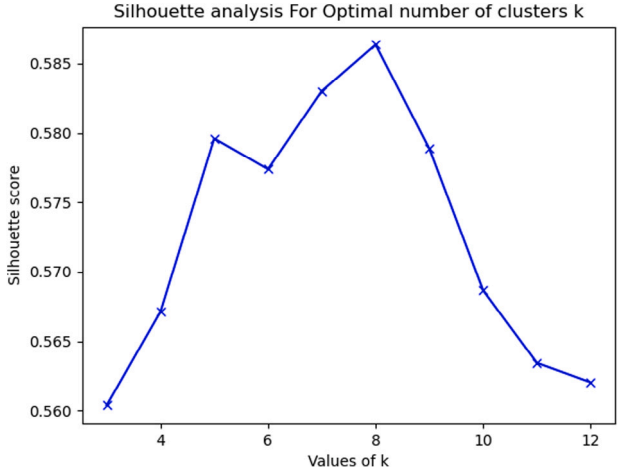


Fig. 5. Curve of the average silhouette for values of the number of clusters k from 3 to 12.

models trained on a single dataset and then evaluated on the other five datasets, the results i.e., the domain-specific accuracies for each nutrient separately are presented in Table 3. Since we are testing different regression models trained and tested with the same embeddings, in this table we reported the maximum obtained accuracy for each pair of datasets no matter which regression model lead to it, and the averaged obtained accuracy that is the mean accuracy across all regression models trained for that pair of datasets.

Using the generalizability matrix (Table 2) and the results from the evaluation of the predictive models (Table 3), several findings can be drawn and discussed:

- **Model trained on the Indian recipes dataset** – The generalizability index of the Indian recipes dataset is the highest with the Yummly28K recipes dataset. For the remaining datasets, the generalizability indexes are lower. From the results presented in Table 3 we can see that the models trained on the Indian recipes dataset achieved the highest accuracy for all nutrients when evaluating it on the Yummly28K recipe dataset. To support this finding, Fig. 9 presents the distributions of the Indian recipes dataset and the Yummly28K recipe dataset in the feature space. From it, we can see that the samples form their underlying distributions overlap in contrast to the distributions of the Indian recipes dataset and the Recipe1M, Epicurious, and Recipe box datasets (presented in Fig. 10). From Fig. 11 we can see that the pattern of the distribution of the Salad recipes dataset matches the distribution of the Indian recipes dataset but it overshadows it as the Salad recipes dataset is significantly larger (around 13 times larger) and its distribution is widely spread across the feature space, therefore a model trained on the Indian recipes dataset cannot cover the diversity of instances contained in the Salad recipes dataset.
- **Model trained on the Recipe1M dataset** – For the Recipe1M dataset, the highest generalizability index is achieved for the Epicurious recipe dataset, followed by the Recipe box dataset, and right after is the Yummly28K recipe dataset. From Fig. 12, we can see how similar the distributions in the feature space of the three datasets are, and from Fig. 13, we can observe why the Yummly28K dataset has a slightly lower generalizability index — there are two big chunks of instances from the dataset that are not covered by the distribution of the other three datasets. Comparing the number of instances of the Recipe1M dataset and the Yummly28K recipe dataset, we can see that the Recipe1M dataset is almost twice the size of the Yummly28K dataset, meaning this happens because the instances of the Yummly28K recipe dataset

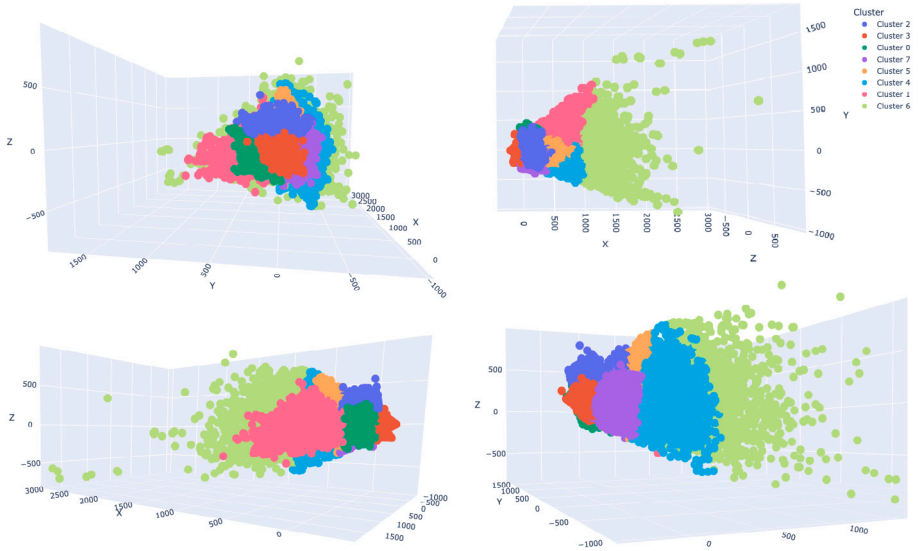


Fig. 6. Clustering into eight clusters of the reduced embedding produced with Word2Vec — architecture: CBOW, dimension: 100, sliding window: 3 merging heuristic: average.

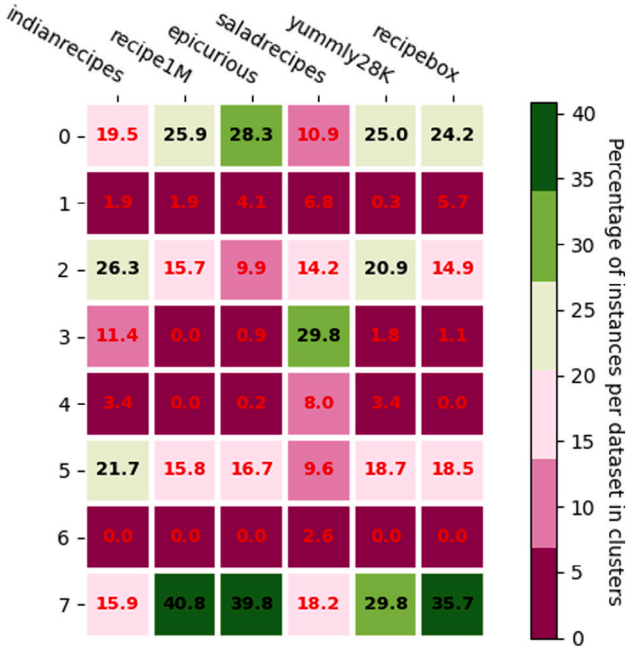


Fig. 7. Percentage of instances of each dataset per cluster.

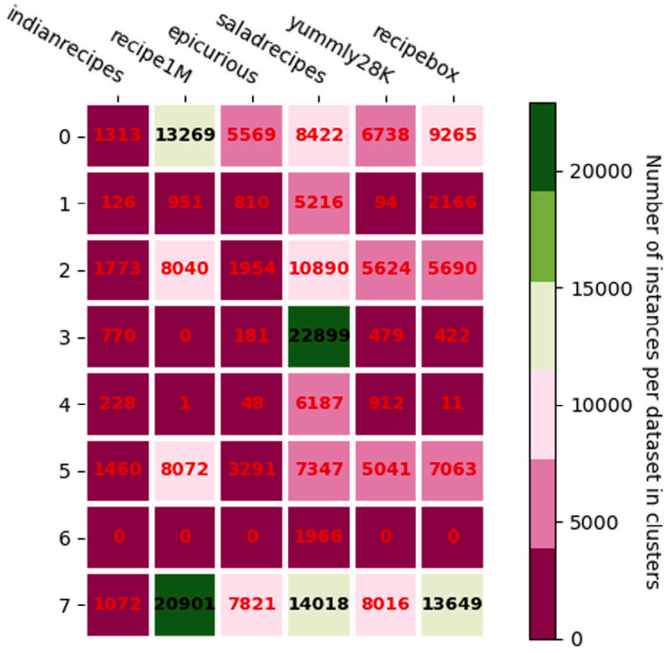


Fig. 8. Number of instances from each dataset per cluster.

are distributed more widely in the feature space, and the distribution of the instances of the Recipe1M dataset is denser in the feature space. From the results presented in Table 3 we can observe that the models trained on the Recipe1M dataset, and tested on these three datasets (Epicurious, Yummly28K, and Recipe box) yield much higher accuracies in contrast to the testing on the other two datasets (Indian recipes and Salad recipes).

- **Model trained on the Epicurious recipe dataset** – For the Epicurious recipe dataset, we can observe that the highest generalizability index is obtained for the Recipe1M dataset, and the second highest for the Recipe box dataset, which is expected from the distributions in the feature space presented in Fig. 12. The Yummly28K recipe dataset, again, has yielded a lower generalizability index (Fig. 13), which we can see also reflects the results presented in Table 3.
- **Model trained on the Salad recipes dataset** – For the Salad recipes dataset, we can see that the results from calculating the generalizability indexes are quite different. Very low values are obtained for the Recipe1M and the Epicurious datasets, while slightly higher generalizability indexes are obtained for the Yummly28K and the Recipe box datasets. The highest generalizability index is obtained for the Indian recipes dataset, and from Fig. 11, as observed previously, we can see that this happens because out of all of the datasets, the Salad recipes dataset has the most similar distribution with the Indian recipes dataset. This is also evident from the results presented in Table 3. If we observe the heat map presented in Fig. 8, we can see that in the cluster number 6, there are only instances from the Salad recipes dataset, which therefore cannot be captured from any model trained on the other five datasets (presented in Fig. 14). An interesting observation from the heat map presented in Fig. 8 is that the

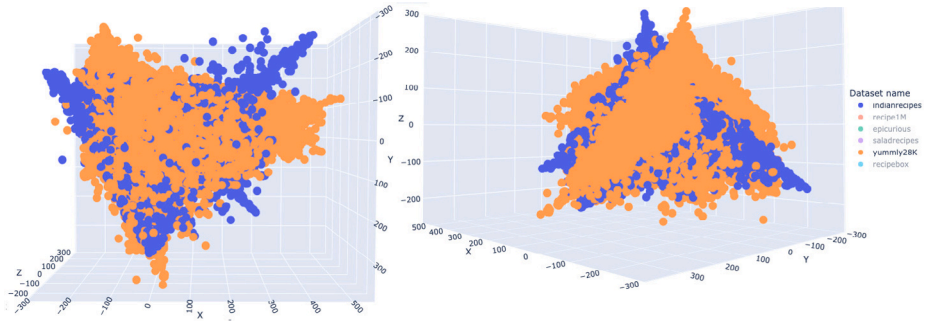


Fig. 9. Distributions in the feature space of the Indian recipes dataset and the Yummly28K dataset.

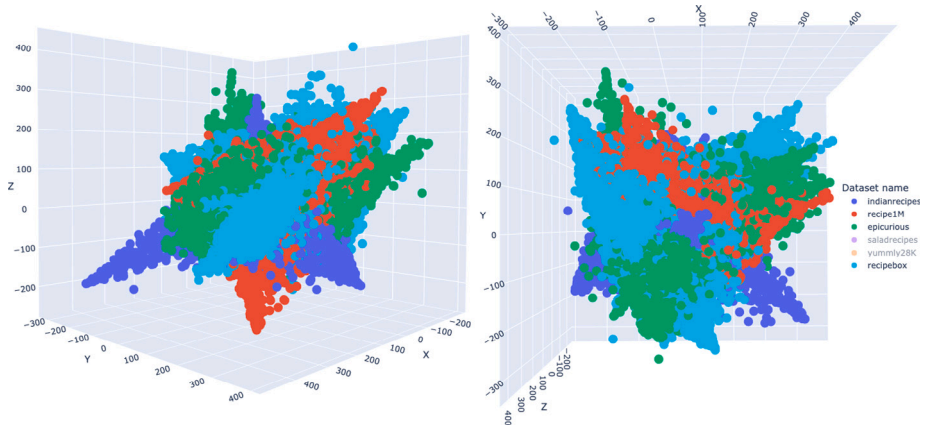


Fig. 10. Distributions in the feature space of the Indian recipes, Recipe1M, Epicurious, and Recipe box datasets.

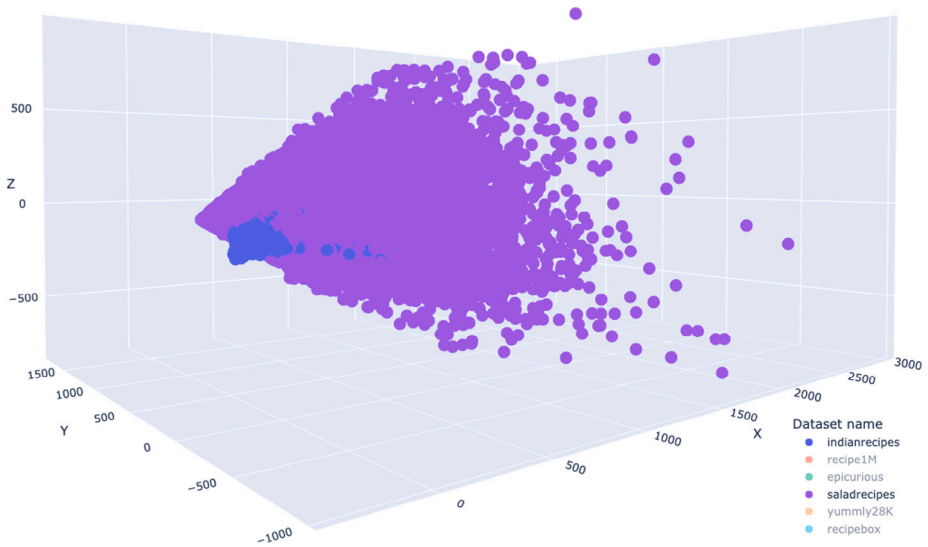


Fig. 11. Distributions in the feature space of the Indian recipes dataset and the Salad recipes dataset.

cluster number 7 is densely populated by instances from the Recipe1M, Epicurious, Yummly28K, and the Recipe box datasets (above 30% from each dataset). Whereas, from the percentage of the remaining two datasets in the same cluster, we can see that the Indian recipes and Salad recipes datasets have only 15.9% and 18.2% of their instances in the cluster number 7, respectively. These two observations show why the generalizability indexes of the Salad recipes dataset are so low for these four datasets, and the highest generalizability index is achieved for the Indian recipes dataset.

- **Model trained on the Yummly28K recipe dataset** – For the Yummly28K recipe dataset, the highest generalizability index is obtained for the Recipe box dataset, followed by the Recipe1M dataset, the Epicurious recipe dataset, and then the Indian recipes datasets. If we compare the distributions of instances in the feature space of the Yummly28K recipe dataset and those of the Recipe box, Recipe1M, Epicurious, and Indian recipes datasets, we can see that all of these four datasets have overlapping with the Yummly28K recipes dataset. The generalizability indices are sorted the same as the number of instances per dataset. Since

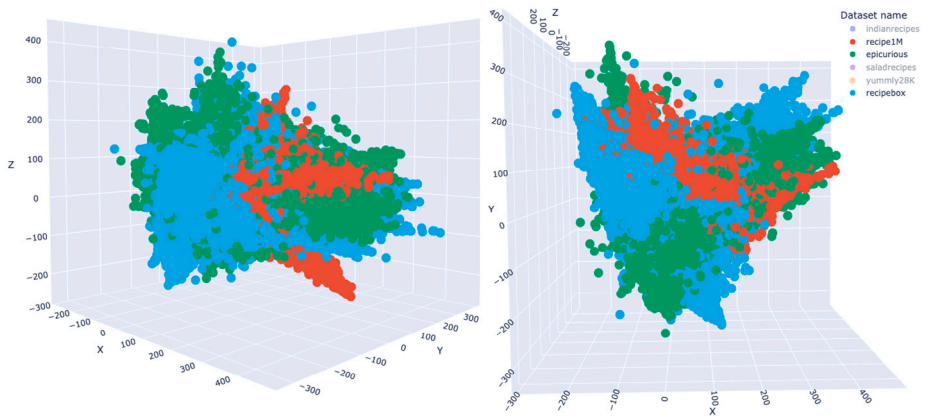


Fig. 12. Distributions in the feature space of the Recipe1M, Epicurious, and the Recipe box datasets.

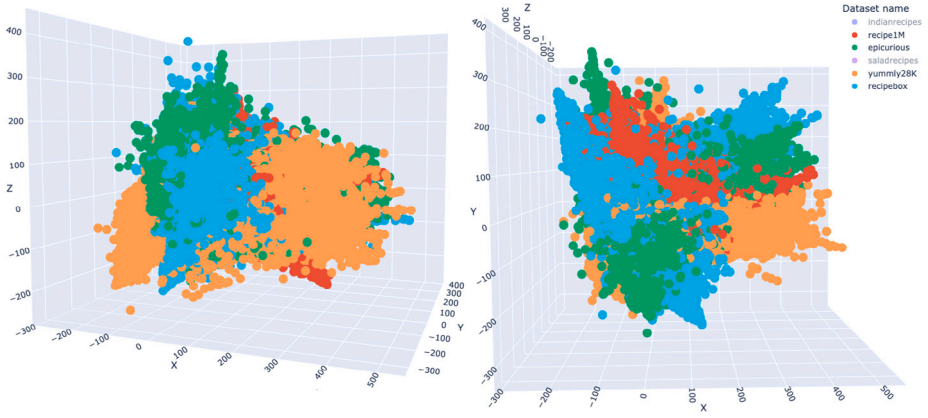


Fig. 13. Distributions in the feature space of the Recipe1M, Epicurious, Recipe box and Yummly28K datasets.

the instances from the Yummly28K recipes dataset are distributed farther apart (i.e. widely distributed) in the feature space, the generalizability index is the highest for the Recipe box dataset, because it has around 82,000 instances, therefore it is more densely distributed and the Yummly28K dataset covers more parts from the feature space and therefore more instances can be represented. Next, is the generalizability index for the Recipe1M dataset, because the Recipe1M dataset has around 55,000 instances, much more densely distributed compared to the distribution of the instances from the Yummly28K recipe dataset in the feature space. Then follow the generalizability indexes of 0.66 and 0.61 for the Epicurious recipe dataset (with around 21,000 instances) and the Indian recipes dataset (with around 6700 instances). From the presented accuracies in Table 3 we can observe how the above-made statements, i.e., how the obtained generalizability indexes correspond with the performance of the predictive models — a higher generalizability index implies a higher accuracy.

- **Model trained on the Recipe box dataset** – For the Recipe box dataset, the highest generalizability index is achieved for the Recipe1M dataset and it is very comparable with the generalizability index for the Epicurious recipe dataset. Next, we have the generalizability index for the Yummly28K recipe dataset. From the presented accuracies of the predictive models, we can observe the reliability of the generalizability indexes — the predictive models trained on the Recipe box dataset performed best when tested on the Recipe1M dataset, and second best when tested on the Epicurious recipe dataset.

Table 4 presents the Pearson correlation coefficients calculated between the generalizability indices and the performance of the ML models across different datasets for each target separately. Each row in the table corresponds to the dataset on which the model is trained, and each column corresponds to a nutrient target that is predicted by the model. For instance, when training the model on the Indian dataset and examining the correlation between the generalizability indices and the model's performance in predicting fat content on other datasets, we utilize the generalizability indices from the first row of Table 2 excluding the first value (since there is no need for a self-calculated generalizability index). Additionally, we consider the model's performance (the average accuracies) obtained for predicting fat content on the other five recipe datasets (listed in Table 3), which correspond to training on the Indian recipes dataset and testing on the other five recipe datasets. The results demonstrate high correlations (above 0.5) between the generalizability indices and the model's performance, indicating strong associations. However, there are a few exceptions where sodium as the prediction target does not exhibit such high correlations.

Table 4
Pearson correlation coefficients between the generalizability index and the average accuracies.

Dataset	Fat	Protein	Sugars	Saturated fat	Sodium
Indian	0.7126	0.9293	0.7924	0.6246	−0.4762
Recipe1M	0.6560	0.5376	0.5723	0.8026	0.2235
Epicurious	0.7926	0.5719	0.6892	0.7140	0.4935
Salad Recipes	0.8461	0.7434	0.7568	0.9207	0.8876
Yummly28K	0.9355	0.9181	0.8774	0.9385	0.9793
Recipe Box	0.7205	0.4145	0.8042	0.7711	−0.4348

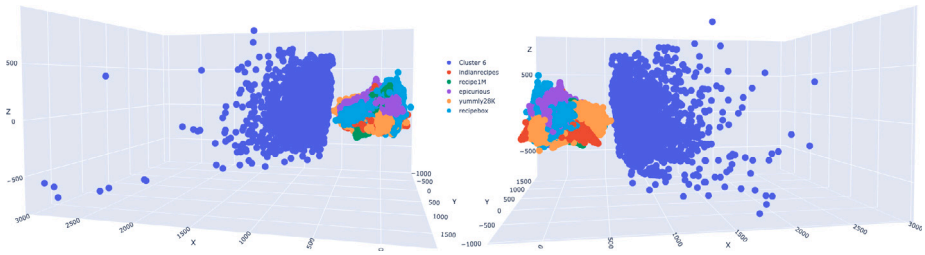


Fig. 14. Distributions in the feature space of the instances belonging to cluster number 6 and the instances from the Indian recipes, Recipe1M, Epicurious, Yummly28K, and Recipe box datasets.

Table 5
Number of instances from each cluster in the generalized training dataset selected with stratified sampling from each cluster.

Cluster	Number of instances		
	10% of the cluster	20/% of the cluster	30/% of the cluster
0	4,457	8,915	13,372
1	963	1,873	2,836
2	680	1,359	2,039
3	2,475	4,950	7,425
4	738	1,477	2,215
5	3,227	6,455	9,732
6	196	393	589
7	6,547	13,095	19,642

Table 6
Results from the evaluation on the models trained when using the generalized training dataset obtained selected with stratified sampling around the centroids of each cluster.

Generalized training dataset	Target	Average accuracy
10% closest to centroid of each cluster	Fat	71.23
	Protein	79.39
	Saturated fat	67.58
	Sugars	64.23
	Sodium	70.12
20% closest to centroid of each cluster	Fat	75.63
	Protein	83.82
	Saturated fat	73.54
	Sugars	79.84
	Sodium	76.99
30% closest to centroid of each cluster	Fat	78.76
	Protein	87.45
	Saturated fat	77.12
	Sugars	83.36
	Sodium	80.93

4.3.3. Selecting a more representative training dataset

Instead of training a model on a single dataset, in order to cover more parts from the feature space that can improve the generalization of the predictive models, we propose three different sampling techniques for selecting data instances from each cluster that can be from different datasets. The idea behind each sampling technique is presented below:

- **Stratified sampling from each cluster** – Select 10%, 20% and 30% random data instances from each cluster (the number of instances from each cluster for each percentage is presented in Table 5). These percentages are selected as more than 30% will lead to potential over-fitting of the predictive models. We repeat these selections several times, each time resulting in different samples. Then, predictive models are trained, with each of these datasets as the training dataset.
- **Stratified sampling around the centroids of each cluster** – Select 10%, 20%, and 30% of the instances in each cluster, this time closest to the centroid of that cluster. The distance between each data instance and the centroid of the cluster is calculated by calculating the cosine distance between the reduced embeddings of the instance and the centroid. Here, the sampling selects the closest 10%, 20%, or 30% data instances to the cluster centroid. As we are trying to select only the instances closest to the cluster centroid, this is a one-time procedure.
- **Uniform sampling around the centroids of each cluster** – Select the same number of data instances from each cluster. The selection is performed by selecting the instances that are in some ϵ-neighborhood to the centroid of the cluster. This sampling technique will allow us to approximate a uniform distribution over the feature space.

In the case of the first sampling technique, stratified sampling from each cluster, we train predictive models on the generated generalized datasets and evaluated them on the rest of the instances, not included in the training process. Comparing the results with the results presented in Table 3 when the training has been performed on a single dataset, we did not observe an increase, more so a slight decrease in the accuracy. For example, the average accuracy presented of predictive models for fat in a case when a single dataset is used for training is between 2.37% and 74.99%, while training predictive models on the generalized datasets, the average accuracies were up no more than 50%. We need to point out here that this kind of selection is biased towards the bigger clusters, so we are, again, having problems such that some parts from the feature space are over-represented and some are under-represented.

In the case of the second sampling technique, stratified sampling around the cluster centroid, the obtained accuracies of the predictive model trained on the generalized dataset and evaluated on the remaining instances that are not selected are presented in Table 6. The average accuracy is calculated over the obtained accuracies from models trained with all the different regression methods mentioned. We can observe that there is an increase in the average accuracies if we compare to the accuracies presented in Table 3 when the training was performed on each dataset separately and tested on the rest. We need to mention here, that even if have an improvement in the performance of the predictive modeling, we still have some bias presented in the selection since the size of the clusters indicates the size of the sample instances selected from it — the bigger the cluster the more instances selected from it.

In the case of the third sampling technique, uniform sampling around each cluster centroid, we first perform empirical analysis to detect from which ϵ-neighborhood of the cluster centroid we can sample instances that can lead us to approximate uniform distribution. For this purpose, we calculate the cosine similarity of each data instance to the centroid of the cluster to which belongs. After that, for each cluster, we calculate the maximum and minimum cosine distances from each cluster's centroid to an instance belonging to the set cluster (presented in Table 7).

From this table, if we would like to make a uniform sampling from each cluster, we need to select the minimum distance (0.180) since this will allow us to select the same number of instances from each cluster, at the same time selecting all instances from the cluster with the lowest distance (i.e., in our case the cluster number 6 that is also the cluster with the lowest number of instances). This result indicates that we should select 1966 instances from each cluster and we would

Table 7
Maximum cosine distance from each cluster's centroid to an instance belonging to the same cluster.

Cluster number	0	1	2	3	4	5	6	7
Maximum cosine distance	0.422	0.186	0.507	0.184	0.244	0.0.367	0.180	0.536

Table 8
Number of instances from each cluster for $\epsilon \leq 0.186$.

Cluster number	0	1	2	3	4	5	6	7
Number of instances	38,534	9,361	29,475	24,751	7,346	28,961	1,966	47,197

end up with a training dataset of 15,728 instances, which is only 7.16% from the original corpus.

In order to have more data instances, we also select the second and the third lowest distance (0.184 and 0.186 that correspond to the cluster number 3 and the cluster number 1, respectively). In this selection, we are breaking the assumption to have the same number of instances from each cluster, since we should select 7387 (not possible to perform it from the cluster number 6) and 9361 (not possible to perform it from the cluster number 4 and the cluster number 6) from each cluster. However, we can include all instances from these clusters which also provides us guarantee that instances from the under-represented clusters will never come in the test data and we care to have a uniform distribution of the over-represented clusters.

When the selection is performed by selecting 7387 data instances from each cluster that are in the ϵ-neighborhood with $\epsilon = 0.184$ of the centroids of the clusters (meaning the selected instances have a cosine distance lower or equal to 0.184 to the centroid of the cluster they belong to). With this selection, we ended up with a dataset of 53,675 instances (this dataset includes all instances from the cluster number 6), which is 24% from the corpus with recipe embeddings.

In case when the selection is performed with 9361 instances from each cluster, we ended up with a dataset of 65,531 instances which is 29.82% of the whole corpus. This selection includes all instances from the cluster number 6 and the cluster number 4).

After determining the number of instances that should be selected from each cluster and which are in some ϵ-neighborhood to the centroids of the clusters, we repeated the sampling technique three times. When the number of instances was set to 9361, all instances from the clusters with numbers 6, 4, and 1 were selected, while the representative instances from the other clusters were selected randomly from the instances that have a cosine distance of 0.186 or lower to the centroid of the cluster to which they belong to. In each repetition of the selection, the predictive model is trained on the selected instances and evaluated on all the remaining instances from all clusters. For comparison reasons, we decided to conduct the same experiments with the three lowest maximum cosine distances: 0.180, 0.184, and 0.186. The results are presented in Table 9. We can see that there is an improvement in accuracies for all target variables for $\epsilon = 0.186$ in comparison to the previous two stratified sampling techniques of selecting the training dataset. We can also see how the accuracies improved when choosing $\epsilon = 0.186$ rather than 0.184 or 0.180, which is due to the fact of the very low samples included in the training dataset if 0.184 or 0.180 are chosen as the ϵ value. These results show that the quality and the uniform representativeness of all parts of the feature space can lead to an increase in the power of a predictive model (see Table 8).

Table 9
Results from the evaluation on the models trained when using the generalized training dataset obtained when using the instances in a defined ϵ neighborhood of the centroids from each cluster.

Generalized training dataset	Target	Average accuracy
$\epsilon = 0.180$	Fat	44.23
	Protein	55.73
	Saturated fat	53.82
	Sugars	57.31
	Sodium	60.13
$\epsilon = 0.184$	Fat	85.17
	Protein	81.34
	Saturated fat	77.22
	Sugars	82.91
	Sodium	84.34
$\epsilon = 0.186$	Fat	88.24
	Protein	96.53
	Saturated fat	83.41
	Sugars	91.32
	Sodium	93.45

5. Discussion

In this section, we provide an overview of the key findings and limitations of our research. It is important to note that the generalizability index, which measures the ability of a model to generalize, has been calculated based on a landscape analysis solely in the feature space. This analysis does not take into account the target space that the model should predict. In ML, having a comprehensive representation (a set of features) means that different combinations of feature values lead to different target values that the model should predict. During model training, the model learns from input data, which are instances represented by their feature values. Sometimes, if certain areas of the feature space are over-represented, the model may become biased towards those areas.

We begin with the assumption that if two datasets have a similar distribution in the feature space, a model trained on one dataset should be able to generalize well to the other dataset, and vice versa. To assess the similarity between datasets, we introduce a meta-representation obtained from a clustering analysis. Initially, all instances from all datasets are combined, clustered, and analyzed to determine which parts of the feature space they cover. Subsequently, a meta-representation is defined for each dataset, representing the distribution of data instances within each cluster. This enables us to compare the distribution of the datasets across the clusters in the feature space.

By comparing the generalizability index with the model's testing scores, we have identified high correlations between differences in distribution and the model's performance (as seen in Table 4).

Next, we provide points related to different steps in the proposed methodology. The selection of feature embedding, specifically textual embeddings, is an important aspect of the proposed methodology. In our previous studies, we evaluated various text representation methods, including word-level, sentence-level, and document-level embeddings, such as Word2Vec, GloVe, and BERT. Ultimately, we proposed an embedding method that combines word embeddings with a domain-specific heuristic, which yielded improved accuracy in predicting macronutrients (Ispirova et al., 2021). In this study, we use this heuristic to define the feature embeddings, ensuring that all data instances from different datasets are brought into the same feature vector space. The analysis is conducted using the embedding used for training the model. However, we acknowledge that different embedding methods can affect the model's performance. If an alternative embedding method is selected for model training, the proposed pipeline can still be utilized to estimate the model's generalizability with the

new feature embedding method. While our focus is not on analyzing the sensitivity of the pipeline to different embedding methods (which will be addressed in future work), we present a methodology that can be tested once the embedding method and modeling approach is fixed.

The generalizability index is further calculated based on a reduced embedding obtained through PCA applied to the recipe text. We opted for PCA as the dimensionality reduction technique based on its proven effectiveness for K-means clustering. This choice was also based on the fact that t-SNE primarily preserves local similarities, whereas PCA retains the global structure of the data, which is crucial for effective clustering. Our goal was to reduce the dimensions while still preserving the general structure of the data in order to perform the clustering. In our study, we ensure that the first three principal components are used, resulting in a high explained variance (approximately 80%). This indicates that the data distribution is not distorted by the dimensionality reduction process. However, for other datasets, it may be necessary to perform a pre-analysis to determine the appropriate number of principal components. The Cattell subjective scree test estimator (Cangelosi & Goriely, 2007) can be used to assist in selecting the number of principal components. Thus, as long as the feature embedding accurately represents the underlying data distribution, the generalizability index serves as a reliable measure of a model's ability to generalize to unseen data.

The clustering results obtained with K-means clustering were validated through a post hoc analysis, which demonstrated that the clusters predominantly group recipes belonging to the same food groups. This aspect has also been shown in our previous study, along with a sensitivity analysis regarding the dimension of the generated embeddings. While K-means clustering was chosen as the most commonly used approach, it is worth noting that other clustering methods like Density-based spatial clustering of applications with noise (DBSCAN) (Ester, Kriegel, Sander, Xu, et al., 1996) may yield different results and impact the values of the generalizability index. We acknowledge the importance of evaluating the index's variability under different clustering methods, and we plan to incorporate this analysis into future work to provide a more comprehensive evaluation of the generalizability index and its sensitivity to clustering and dimensionality reduction methods.

Last but not least, to demonstrate the impact of a nearly uniform distribution of training data across the feature space on the model's generalizability, we selectively chose a more representative dataset. This selection aimed to reduce over- or under-represented areas, thereby improving the predictive capabilities of the models and mitigating overfitting. In line with the principles of open science, our methodology can be applied to evaluate the generalizability index when using new, previously unseen datasets during the testing phase. By comparing the distribution of the training dataset with that of the new dataset, the index allows us to assess how well the pre-trained model performs on the new data.

6. Conclusion

One crucial aspect when working in machine learning is how to estimate if a learned predictive model will generalize its performance on new data instances not included in the training dataset. This issue is related to the quality and representatives of the data included in the training dataset. For this purpose, we propose a novel meta-approach for landscape analysis of the feature space. The approach project all data instances at the same meta-feature vector space that allows us to find which parts of the feature space are over- or under-represented. This further points out spaces from the feature space where the predictive models will be biased. We introduce a generalizability index that based on the results of the landscape analysis can provide information on how good a predictive model will be if we test it on new unseen data instances. In addition, we have also proposed a sampling technique that cares about selecting data instances such that all parts from the feature space will approximately follow a uniform distribution. This kind of selection of a training dataset improves the generalization of a predictive model.

More specifically, we evaluated the approach on a use case of predicting nutrient values of recipes using their description. By calculating domain-specific recipe embeddings for six different heterogeneous recipe datasets utilizing a predefined corpus of ingredient embeddings, we project them to the same meta-feature vector space. Reducing the high-dimensional vectors of the embeddings to three-dimensional vectors using PCA allowed us to visualize their distributions in the feature space and observe which of these datasets have similar distributions. Then, with the process of unsupervised ML — clustering, we separated the feature space into eight different clusters. Observing the distributions of the clusters, we could see how the datasets are distributed across clusters. Having this, we proceeded with defining a generalizability index, and a generalizability matrix for all pairs between the six datasets that indicated the generalizability of a predictive model i.e. how successful the generalization of a predictive model learned on one dataset to the other dataset not used in the training will be. Although the Salad recipes dataset is the largest and has the highest number of instances, it scored low generalizability indexes due to its distribution in the feature space and in the clusters. While the other datasets, specifically the Epicurious, Recipe1M, Yummly28K, and Recipe box, scored high amongst each other due to their similar distributions in the feature space. These two statements can be directly seen through the results from the predictive modeling and evaluation of the ML pipeline, presented in Table 3.

In addition, we introduce three sampling techniques for selecting a representative training dataset that will improve the generalization of a predictive model. The first technique involve randomly selecting a given percentage of instances from each cluster, while the second one involves the random selection of a given percentage of instances that are close to the cluster centroid. Both sampling techniques improve the generalization of a predictive model, however, the generated datasets are still biased to some parts of the feature space, since bigger clusters are represented with more instances. The third technique is randomly sampling the same number of instances from each cluster in some neighborhood to its centroid. This technique allows us uniform coverage of all parts from the feature space and improves the generalization of the predictive model.

For future work, we are planning to perform sensitivity analysis that will be involved evaluating different methods that are part of our landscape analysis, especially, testing different clustering methods to obtain the clusters of the feature space and different dimensionality reduction techniques. Another direction is to estimate the threshold of the generalizability index that can guarantee the generalization of a predictive model on new unseen instances.

CRediT authorship contribution statement

Gordana Ispirova: Conceptualization, Methodology, Data curation, Validation, Visualization, Writing – original draft. **Tome Eftimov:** Conceptualization, Methodology, Writing – reviewing & editing. **Sašo Džeroski:** Writing – reviewing & editing. **Barbara Koroušić Seljak:** Supervision, Writing – reviewing & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Barbara Korousic Seljak reports financial support was provided by Horizon Europe Excellent Science. Computer Systems Department reports financial support was provided by Public Research Agency of the Republic of Slovenia.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported by the Slovenian Research Agency (research core grant number P2-0098 and research core grant number P2-0103), and the European Union's Horizon 2020 - EXCELLENT SCIENCE - Research Infrastructures programme COMFOCUS (grant agreement ID: 101005259).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2023.121507.

References

Anon (2022). Yummly website. URL https://www.yummly.com/.

Binol, H., Niazi, M. K. K., Plotner, A., Sopkovich, J., Kaffenberger, B. H., & Gurcan, M. N. (2020). A multidimensional scaling and sample clustering to obtain a representative subset of training data for transfer learning-based rosacea lesion identification. In *Medical imaging 2020: computer-aided diagnosis, Vol. 11314* (pp. 272–278). SPIE.

Cangelosi, R., & Goriely, A. (2007). Component retention in principal component analysis with application to cDNA microarray data. *Biology Direction*, *2*(1), 1–21.

Cenikj, G., Lang, R. D., Engelbrecht, A. P., Doerr, C., Korošec, P., & Eftimov, T. (2022). SELECTOR: Selecting a representative benchmark suite for reproducible statistical comparison. arXiv preprint arXiv:2204.11527.

Chen, Q., Shui, C., & Marchand, M. (2021). Generalization bounds for meta-learning: An information-theoretic analysis. *Advances in Neural Information Processing Systems*, *34*, 25878–25890.

Chung, Y., Haas, P. J., Upfal, E., & Kraska, T. (2018). Unknown examples & machine learning model generalization. arXiv preprint arXiv:1808.08294.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Eftimov, T., Petelin, G., Cenikj, G., Kostovska, A., Ispirova, G., Korošec, P., et al. (2022). Less is more: Selecting the right benchmarking set of data for time series classification. *Expert Systems with Applications*, *198*, Article 116871.

Epicurious Recipes Dataset (2017). Epicurious recipes dataset. URL https://www.kaggle.com/datasets/hugodarwood/epirecipes?select=recipe.py.

Epicurious website (2022). URL https://www.epicurious.com/recipes-menus.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd, Vol. 96, no. 34* (pp. 226–231).

European commission health and consumers directorate-general (2012). guidance document for competent authorities for the control of compliance with eu legislation on: Regulation (EU) No 1169/2011 of the European Parliament and of the Council of 25 October 2011 on the provision of food information to consumers, amending Regulations (EC) No 1924/2006 and (EC) No 1925/2006 of the European Parliament and of the Council, and repealing Commission Directive 87/250/EEC, Council Directive 90/496/EEC, Commission Directive 1999/10/EC, Directive 2000/13/EC of the European Parliament and of the Council, Commission Directives 2002/67/EC and 2008/5/EC and Commission Regulation (EC) No 608/2004Devlin. URL https://ec.europa.eu/food/sites/food/files/safety/docs/labelling_nutrition-supplements-guidance_tolerances_1212_en.pdf.

Guiroy, S., Pal, C., Mordido, G., & Chandar, S. (2022). Improving meta-learning generalization with activation-based early-stopping. In *Conference on lifelong learning agents* (pp. 213–230). PMLR.

Herranz, L. (2017). Yummly28K recipes dataset. URL http://www.lherranz.org/datasets/.

Huan, Z., Wang, Y., He, Y., Zhang, X., Fu, C., Wu, W., et al. (2021). Learning to select instance: Simultaneous transfer learning and clustering. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 1950–1954).

Indian Recipes Dataset (2020). Indian recipe dataset. URL https://www.kaggle.com/datasets/kanishk307/6000-indian-food-recipes-dataset.

Indian Recipes Dataset (2022). Indian recipe dataset. URL https://www.archanaskitchen.com/.

Ispirova, G., Eftimov, T., & Koroušić Seljak, B. (2020). P-NUT: Predicting NUTrient Content from Short Text Descriptions. *Mathematics*, *8*(10), 1811, Publisher: Multidisciplinary Digital Publishing Institute.

Ispirova, G., Eftimov, T., & Koroušić Seljak, B. (2021). Domain Heuristic Fusion of Multi-Word Embeddings for Nutrient Value Prediction. *Mathematics*, *9*(16), 1941, Publisher: Multidisciplinary Digital Publishing Institute.

Ispirova, G., Eftimov, T., & Seljak, B. K. (2022). Predefined domain specific embeddings of food concepts and recipes: A case study on heterogeneous recipe datasets. In *2022 IEEE international conference on big data (big data)* (pp. 4074–4083). IEEE.

Kaufman, L., & Rousseeuw, P. J. (1990). *An introduction to cluster analysis*. John Wiley and Sons, Incorporated.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., & Iyer, R. (2021). Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35, no.9* (pp. 8110–8118).

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196). Beijing, China.

Lee, R. T. (2020). Recipe box dataset. URL https://eightportions.com/datasets/Recipes/.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th berkeley symp. math. statist. probability* (pp. 281–297).

Maleki, F., Ovens, K., Gupta, R., Reinhold, C., Spatz, A., & Forghani, R. (2022). Generalizability of machine learning models: Quantitative evaluation of three methodological pitfalls. *Radiology: Artificial Intelligence*, *5*(1), Article e220028.

Marin, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., et al. (2019). Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Publisher: IEEE.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119). Lake Tahoe, Nevada, USA.

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., et al. (2021). Accuracy on the line: On the strong correlation between out-of-distribution and in-distribution generalization. In *International conference on machine learning* (pp. 7721–7735). PMLR.

Min, W., Jiang, S., Sang, J., Wang, H., Liu, X., & Herranz, L. (2016). Being a supercook: Joint food attributes and multimodal content modeling for recipe retrieval and exploration. *IEEE Transactions on Multimedia*, *19*(5), 1100–1113.

Ng, A. (2022). Unbiggen ai. *IEEE Spectrum*, https://spectrum.ieee.org/amp/andrew-ng-data-centric-ai-656501647#amp_tf=from%20%251%24s&aoh=1644465158241&csi=0&referrer=https%3A%2F%2Fwww.google.com.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830, Publisher: JMLR. org.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Salad Recipes Dataset (2017). Salad recipes dataset. URL https://www.kaggle.com/datasets/snehallokesh31096/recipe.

Strickland, E. (2022). Andrew Ng, AI minimalist: The machine-learning Pioneer says small is the new big. *IEEE Spectrum*, *59*(4), 22–50.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1–3), 37–52, Publisher: Elsevier.

Yao, H., Huang, L.-K., Zhang, L., Wei, Y., Tian, L., Zou, J., et al. (2021). Improving generalization in meta-learning via task augmentation. In *International conference on machine learning* (pp. 11887–11897). PMLR.

Zhou, P., Chen, B.-C., Han, X., Najibi, M., Shrivastava, A., Lim, S.-N., et al. (2020). Generate, segment, and refine: Towards generic manipulation segmentation. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34, no. 07* (pp. 13058–13065).