



组 别 本科生
题 目 B
队 号 123



2020 年数学建模竞赛

基于机器学习的菜谱差异化模型和数字菜谱设计

姓 名 及 学 号	朱 英 豪 18373722
姓 名 及 学 号	任 昌 禹 18373718
姓 名 及 学 号	李 逸 涵 18373716
队伍联系电话	15026559349
队伍联系邮箱	zhuyinghao1999@msn.cn



摘要

人工智能技术的快速发展给烹饪带来了新的方式，本文将为一款烹饪机器人设计核心功能，建立全世界范围内多种代表性菜系的菜品模型。模型包括菜系中菜品主食材、调味料配比、烹调方法的特征。进一步，利用建立好的模型为烹饪机器人制定一份数字化菜谱，让机器人按照数字化菜品的程序即可完成菜品。

我们搜索到世界范围内的二十个菜系的菜品配料表和烹饪方法，分成食材、调味料配比、烹饪方法三个子模型，每个子模型依次对数据进行单词清洗，去除单复数和时态，TF-IDF 加权和 Logistic 回归算法建模，得到子模型的特征信息。最后使用神经网络中的反向传播算法，综合三个子模型的特征，增加若干 softmax 隐含层，使用 PyTorch 进行训练，刻画出不同菜系菜品的模型化差异。

为建立数字化菜谱，结合已经建立好的模型，使用神经网络算法实现当用户输入菜系、主要食材、口味偏好信息时，程序会为用户推荐出最为合适的食材搭配，调味料配比和烹饪方法，为用户制定独属于自己的美食计划。最后用 yaml 语言进行描述让机器人可基于以上菜谱，做对应的操作。

本模型考虑较为全面，算法科学严谨，其建模方法也可以推广至其他的定制化的服务和设计。

关键词：烹饪机器人 Logistic 回归 神经网络 反向传播



Abstract

The rapid development of artificial intelligence technology has brought a new way of cooking. This article will design the core functions of a cooking robot and establish a model of dishes of a variety of representative cuisines around the world. The model includes the characteristics of the main food ingredients, seasoning ratios, and cooking methods in the cuisine. Further, use the established model to formulate a digital recipe for the cooking robot, and let the robot complete the dishes according to the procedure of digital dishes.

We have searched the list of ingredients and cooking methods of twenty cuisines in the world, divided into three sub-models of ingredients, seasoning ratios, and cooking methods. Each sub-model sequentially cleans the data word, removes the plural and tense, TF-IDF weighted and Logistic regression algorithm modeling, get the sub-model feature information. Finally, the back propagation algorithm in the neural network is used to synthesize the features of the three sub-models, add a number of softmax hidden layers, and use PyTorch for training to characterize the model differences of dishes in different cuisines.

In order to establish a digital recipe, combined with the established model, the neural network algorithm is used to realize that when the user enters the cuisine, main ingredients, and taste preference information, the program will recommend the most suitable ingredients, seasonings, and cooking methods for the user. Make your own gourmet plan for users. Finally, the description in yaml language allows the robot to do the corresponding operation based on the above recipe.

This model is more comprehensive and the algorithm is scientifically rigorous. Its modeling method can also be extended to other customized services and designs.

Keywords: Cooking robot Logistic regression Neural network Back propagation



队伍声明

我代表参赛队伍全体队员声明，本论文及其研究工作是由队伍成员独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。



目录

1 介绍	6
1.1 问题重述	6
1.2 解决方案概述	6
2 符号定义	7
3 总体假设	7
4 模型的建立与求解	7
4.1 原料差异模型	7
4.2 调味料配比模型	10
4.3 口味烹调模型	13
4.4 综合模型	14
4.5 数字菜谱设计	15
5 模型评价	17
5.1 优势	17
5.2 不足	18
参考文献	19
附录	20



1 介绍

1.1 问题重述

我们经常想象着足不出户就能尝尽世间美食，但又很少能有时间和精力学习烹饪方法下厨制作。我们可以在网络上便捷地找到很多菜肴的配料表和烹饪步骤，但常常担心是不是不合自己的口味。伴随着人工智能的飞速发展和广泛应用，智能烹饪机器人也将进入厨房，给人们带来全世界的美食，甚至可以依据不同人的口味创造美食。这样一款可有创造力——根据不同人的口味定制化需求的智能烹饪机器人将在未来得到青睐。

作为制造智能烹饪机器人的前置工作，我们在本文中需要设计、解决以下几个问题：

- 根据世界范围内不同菜系菜品的原材料组成、复合调味料配比以及特定口味烹调过程建立模型并分析不同菜系菜品的模型化差异。
- 进一步利用建立好的模型，制定一份参数化、公式化的数字菜谱。为机器人的后续工作打下基础。

1.2 解决方案概述

对于问题一：建立菜系差异模型，我们搜集了世界上二十个菜系的菜品的配料表，食材以及调味料的配比，制作方法。对于每一个菜系，我们统计出各个配料的使用频率，编写程序提取各菜系配料特征，烹饪方法的特征，利用机器学习算法，找到三者之间的联系，最终建立起综合原材料、调味料配比、口味烹饪的模型。

对于问题二：使用建立好的模型，编写程序，设计了一种菜谱表述语言，使得计算机可以非常方便地根据该数字菜谱，了解指定的食材及做法，并根据数字菜谱的指导，完整且精确地实施烹饪步骤。在数字菜谱中，包含了主食材和调味料的配比以及烹饪的方法和顺序。



2 符号定义

- y_1 : 对各菜系食材的建模结果
- y_2 : 对各菜系调味料配比的建模结果
- y_3 : 对各菜系烹饪方法的建模结果
- I_i : 第 i 种食材
- F_i : 第 i 种调味料
- C_i : 第 i 种烹饪方法

3 总体假设

- 假设烧菜是线性、序列化的流程，即不存在同时放原料和调料的情况，每个步骤均为独立的一项，一个接着一个做。
- 假设对食材的准备、处理由人来做，烹饪机器人仅负责整个烧菜过程。
- 同一菜品的不同烹饪手法、不同口味视为不同的制作模式。
- 原料、调味料、烹调手段三个维度线性决定菜品状态。
- 文中用到的数据集可以代表现实世界上的各菜系和菜品。

4 模型的建立与求解

4.1 原料差异模型

菜系差异由原材料组成、复合调味料配比以及特定口味烹调过程来决定。

首先考察原材料组成对菜系差异的影响：

由 Kaggle 中的数据集 What's Cooking，其中含世界上 20 个主要菜系各种菜的原料组成。统计各个菜系中各种原料出现的次数。

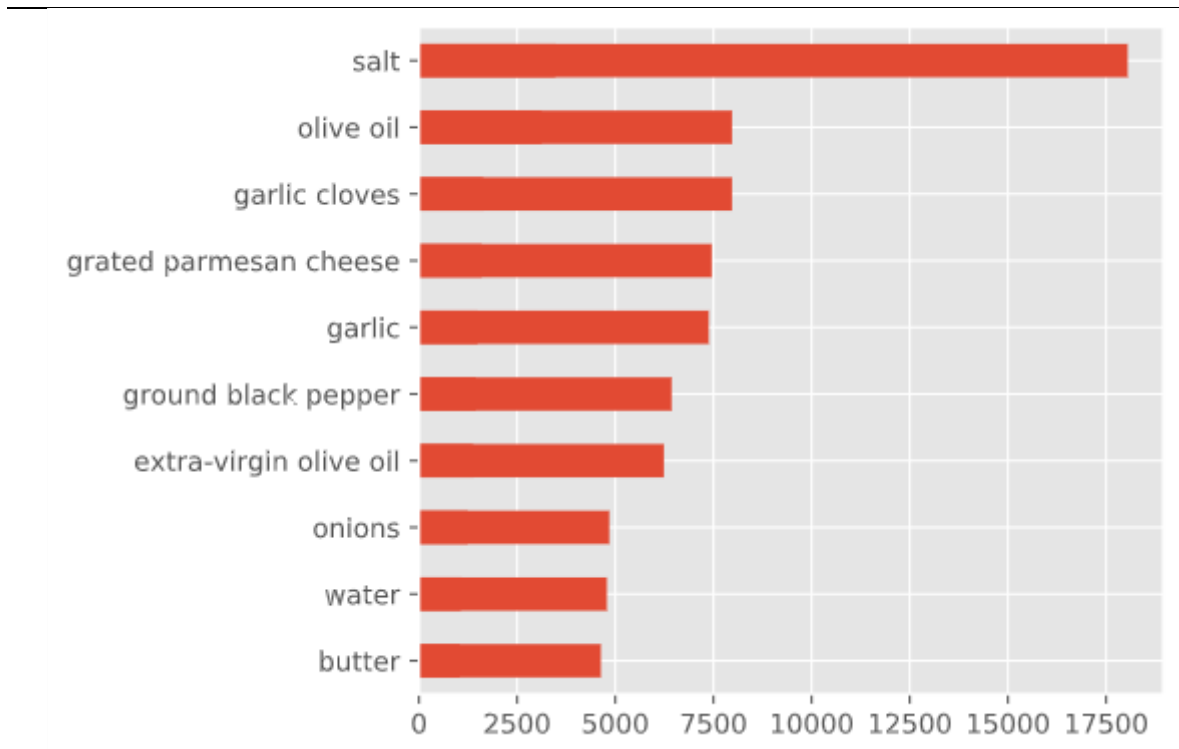


图 1：世界各菜系主要材料的使用频率

以上为经 Python 数据处理后，在所有约 4 万道菜的数据中，使用最频繁的前 10 种佐料。

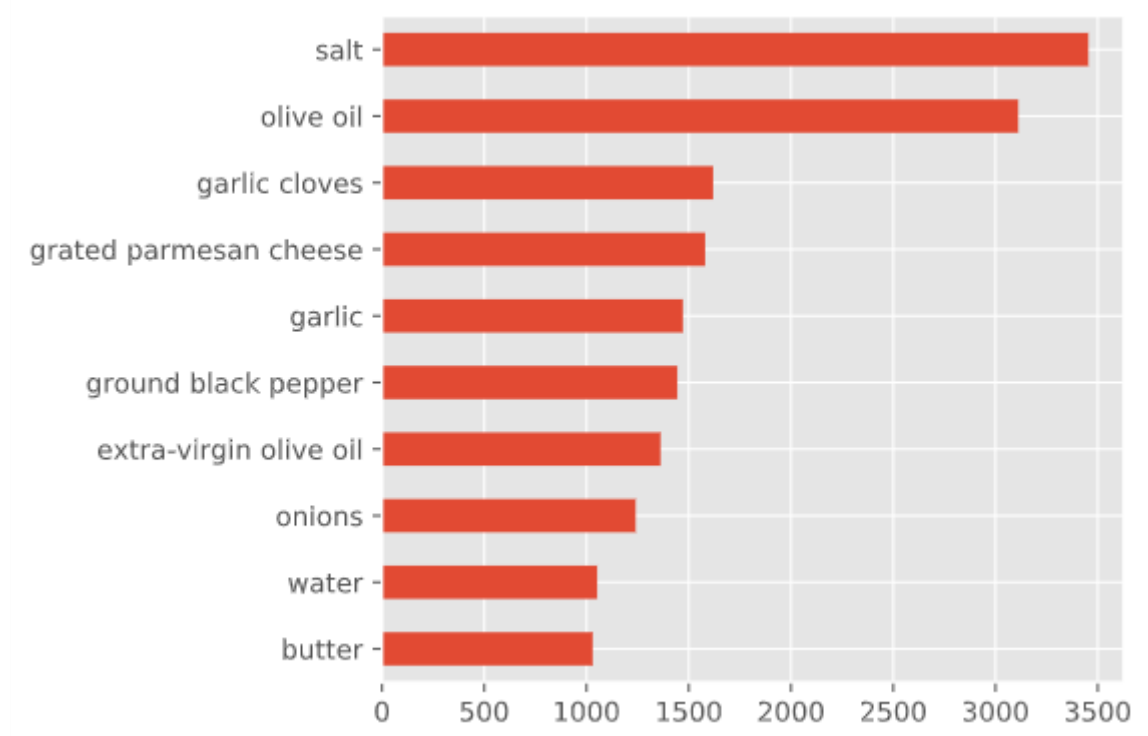


图 2：意大利菜中最常见的 10 个佐料



于是，我们可构建出如下模型：

提取各菜系的配料特征，我们将菜品的配料转换成数值特征向量。考虑到绝大多数菜中都包含 salt、water、sugar、butter 等，采用独热编码的方法提取的向量将不能很好的对菜系作出区分。

我们将考虑按照配料出现的次数对配料进行 TF-IDF 算法的加权：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$$idf_i = \log \frac{|D|}{|\{j: t_i \in d_j\}|}$$

$$tfidf_{i,j} = tf_{i,j} \times idf_i$$

即配料在所有菜系菜品中出现次数越多，则该配料的区分性就越低，所占权重就越低，从而减弱食盐、糖等基础调味料对通过对各菜系食材特征提取的影响。

使用 Logistic 回归算法训练模型，得到各菜系食材的特征，将数据集划分为训练集和测试集，将测试集带入用训练集训练得到的模型，得到较高的成功率，可以评价模型建立的较为合理。其具体步骤为，借助 Sigmoid 函数，假设函数为：

$$h(\vec{x}) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{-\vec{\theta}^{(k)T} \vec{x}}}$$

当测试输入 \vec{x} 时，得到正确预测值的概率为：

$$p = \{h(\vec{x})^{P(y=s_1)}(1-h(\vec{x}))^{P(y \neq s_1)}\} \cdot \{h(\vec{x})^{P(y=s_2)}(1-h(\vec{x}))^{P(y \neq s_2)}\} \cdot \dots \{h(\vec{x})^{P(y=s_K)}(1-h(\vec{x}))^{P(y \neq s_K)}\}$$

求得似然函数：

$$l(\vec{\theta}) = \prod_{i=1}^m \prod_{k=1}^K (h(\vec{x}^{(i)})^{P(y^{(i)}=s_k)}(1-h(\vec{x}^{(i)}))^{P(y^{(i)} \neq s_k)})$$

所以代价函数为：

$$J(\vec{\theta}) = - \sum_{i=1}^m \sum_{k=1}^K (P_{y^{(i)}=s_k} \ln(h(\vec{x}^{(i)})) + (1 - P_{y^{(i)}=s_k}) \ln(1 - h(\vec{x}^{(i)})))$$

接下来使用梯度下降更新：



$$\theta_j^{(t)} := \theta_j^{(t)} - \alpha \frac{\partial J(\vec{\theta})}{\partial \theta_j^{(t)}}$$

$$\begin{aligned} \frac{\partial J(\vec{\theta})}{\partial \theta_j^{(t)}} &= - \sum_{i=1}^{i=m} \sum_{k=1}^{k=K} (P_{y^{(i)}=s_k} \ln(h(\vec{x}^{(i)})) + (1 - P_{y^{(i)}=s_k}) \ln(1 - h(\vec{x}^{(i)}))) \\ &= \sum_{i=1}^{i=m} \sum_{k=1}^{k=K} (h(\vec{x}^{(i)}) - P_{y^{(i)}=s_k}) x_j \left(\frac{e^{-\vec{\theta}^{(t)T} \vec{x}}}{\sum_{u=1}^{u=K-1} e^{-\vec{\theta}^{(u)T} \vec{x}}} \right) \end{aligned}$$

至此，我们建立好了各菜系的菜品配料的差异化模型。

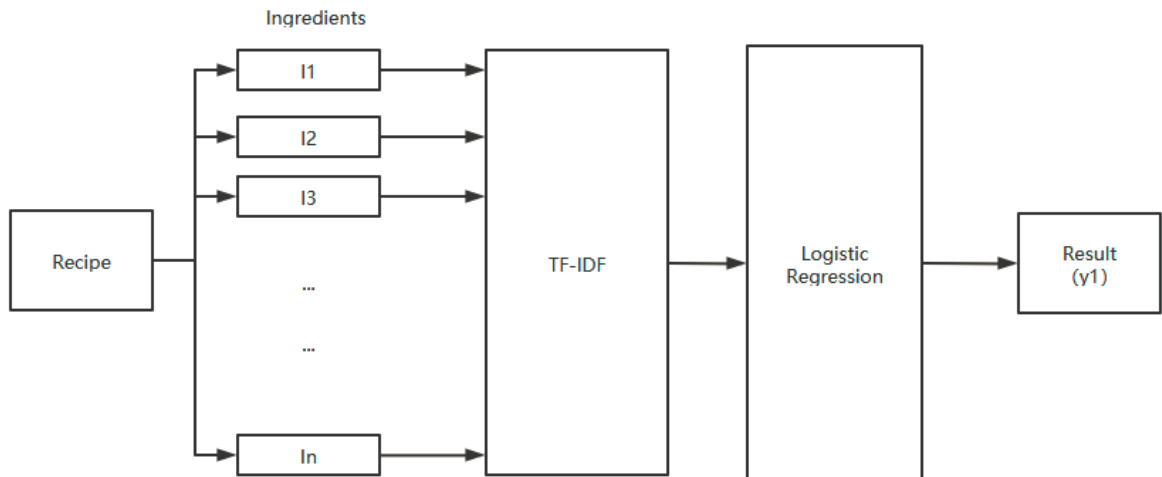


图 3：对各菜系食材特征的建模流程

4.2 调味料配比模型

对于不同菜系的菜品，由于当地饮食习惯不同，以及自然条件对烹饪的影响，当两道来自不同菜系的菜品，所选用食材大致相同时，其调味方式和产生的口味也存在较大差异。而同一菜系往往由于诞生于同一地域和文化环境下，具有更大的相似性。因此我们对不同菜系的调味料以及烹饪过程，根据菜系分别进行讨论。

考虑到菜品的多元化，同一菜系下的菜品，日常饮食和特色菜之间、不同厨师的烹制手法之间，实际上也存在较大差异。为了简化问题，便于进一步的量化，我们在每个菜系下选取当地较为典型的菜品，来代表整个菜系的特征。

另外相比于世界上各菜系的差异，为了简化模型，此处忽略同一国家的菜系进一步划分下的差异（如中国的八大菜系），简单视为同一种菜系。



北京航空航天大学数学建模竞赛

利用以上数据集，通过与上一模型类似的机器学习方法，为每一种菜系分别刻画出调味料配比的模式，将世界上各菜系菜品进行划分。

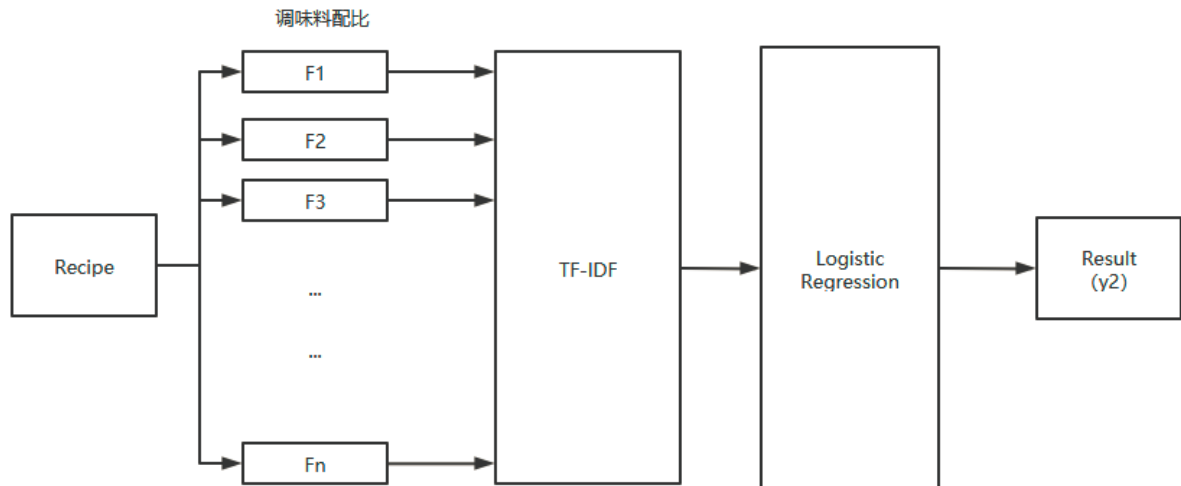


图 4：对各菜系调味料配比特征的建模流程

根据数据集的统计结果，我们绘制了饼状图，可大致刻画出各调味料的用量比例：

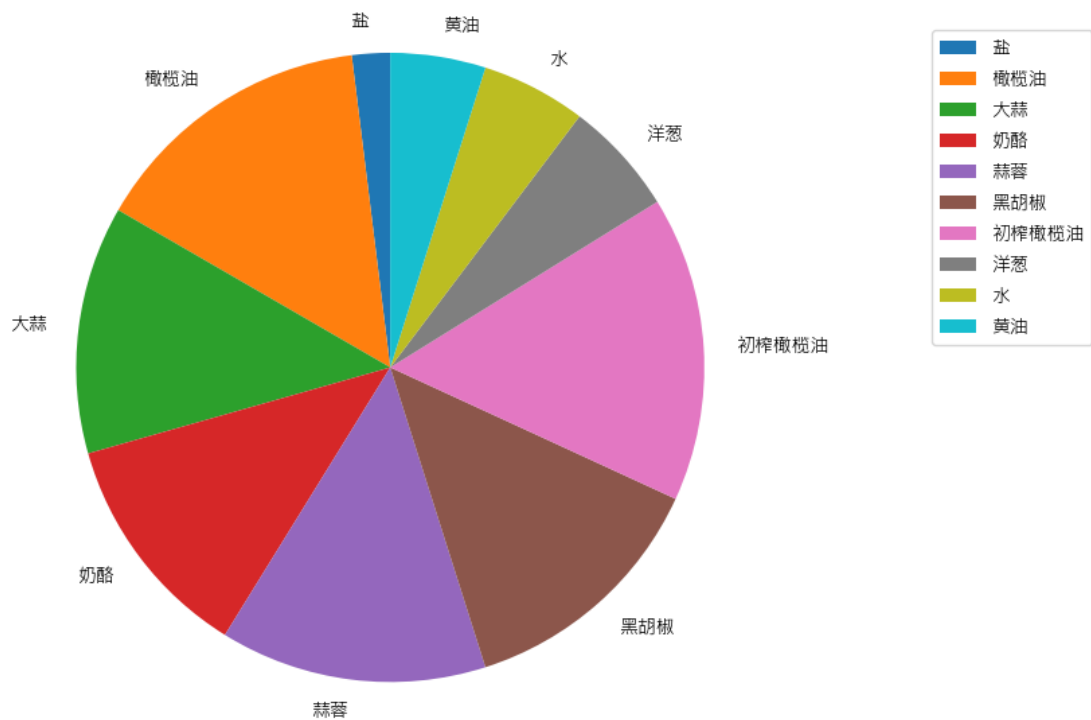


图 5：世界所有菜系调味料用量比例图



北京航空航天大学数学建模竞赛

下面对每个菜系分别展开研究，以其中较为典型的 10 个菜系为例，分析得出结论：

菜系	调味料	烹饪手法
巴西	洋葱在巴西菜中出现频繁，通常作为配菜，用量较大。其次巴西菜偏爱大蒜，蒜蓉、蒜瓣通常会作为主要调味品。酸橙作为地域特色，在甜品中与朗姆酒以不同比例混合使用，用于调味。	炖菜较为常见，食材通常切成块状处理，先与调料在油中翻炒保证充分入味，再入水炖制使汤品粘稠。
英国	很多菜品没有调味料。在蛋糕等烤制食品中，会使用大量黄油和少量盐。	制作方式分为烤箱烤制，和锅里煮两种。 由于调味料的用量较少，菜品口味清淡。一些菜直接在水中煮，菜品更具有原汁原味。
路易斯安那	洋葱比例较大。 调味料中青椒和大蒜占非常大的比例。在一些典型菜品如什锦菜中，辣椒（青椒）和鸡胸肉用量比例甚至相同。	食材多切为块状，在大火中烹调。
中国	中国菜品类丰富，因此用到的调味料种类众多。酱油，耗油，香油，甜面酱，芝麻酱，醋，大葱等调味料，是其他菜系中几乎没有的，在中国菜系中，根据不同菜品，其配比差异较大。 通过对照实际菜名和其所属地域可知，北方菜的醋含量较高；川菜的辣椒含量较高，有时配比超过主食材。盐在大多数菜品中都有出现，但除腌制食品外，一般含量较小。	中国菜制作精细，为保证调味料充分入味，需要将材料提前腌制。在原材料的处理上讲究刀法的差异。
菲律宾	菲律宾菜大多是酸，甜，咸口味调味料结合出现。 调味料中盐的比例较大，常搭配果汁用于均衡口味。	菲律宾烹饪方式炖菜和汤出现频繁。 蔬菜和水果切成小丁，或榨成果汁，保证清香味充分进入食材。



法国

法国菜口感细腻，酱料美味，拥有丰富的品类。

法国菜一般用西红柿代替番茄酱，因此番茄酱使用较少，突出了原色、原味。

法国菜糖的比例较高，其次是油、面粉、鸡蛋，这和英国菜有些相似之处。

法国菜烹调方法多样，包含烤、煎、烩、焗、铁扒、焖、蒸等。

法国菜重视沙司制作，牛肉菜配牛骨汤汁，羊肉菜配羊骨汤汁。一般熬制 8 小时以上。

希腊

希腊菜调味料较为独特，橄榄油、牛至、乳酪所占比例较高。在烧烤中，主要选取柠檬汁、盐作为主要调味料，且用量相当。

希腊菜烹饪方式较为简单，大多是烤或者油炸，之后撒上调味料。

印度

印度菜通常用大量的香辛粉、姜黄根粉（用于做咖喱）和咖喱调味，有时也放入少量茴香籽，油的用量较少。

印度菜烹饪方法简单，通常是清炒为主，其次是凉拌。但印度调料文化丰富，有些调味料往往还是药材，这使得印度菜拥有神秘独特的风味。

墨西哥

墨西哥菜偏辣，茴香、辣椒粉、大蒜用量很大，且辣椒在一些菜品中占超过一半以上。酸奶油、牛油果等独特食材，在墨西哥菜中时有出现。

由于辣椒、酸奶油的混合使用，墨西哥菜兼有酸、辣风味。墨西哥菜做法比较复杂，因而口味丰富，但由于辣椒的大量使用，口味通常以辣为主。

4.3 口味烹调模型

同样利用以上的数据集，我们选取数据集中的烹调步骤列，得到每道菜品的操作步骤的字符串。由于烹饪过程在该字符串中以动词和形容词的形式出现，我们需要从中进行筛选。

手工添加并保留与烹饪相关的关键词，如 `mix`、`dried`、`baking` 等，单词的不同时态均予以保留。使用 Python 的 `nltk` 库对单词进行清洗，筛选后，再将同义不同时态的词归为同一类。

对于每一种菜系，分别统计烹调词汇的出现频率。



根据其出现的频率、烹调词汇的不同，使用与先前模型类似的多项 Logistic 回归，对含烹调步骤的数据集进行处理。

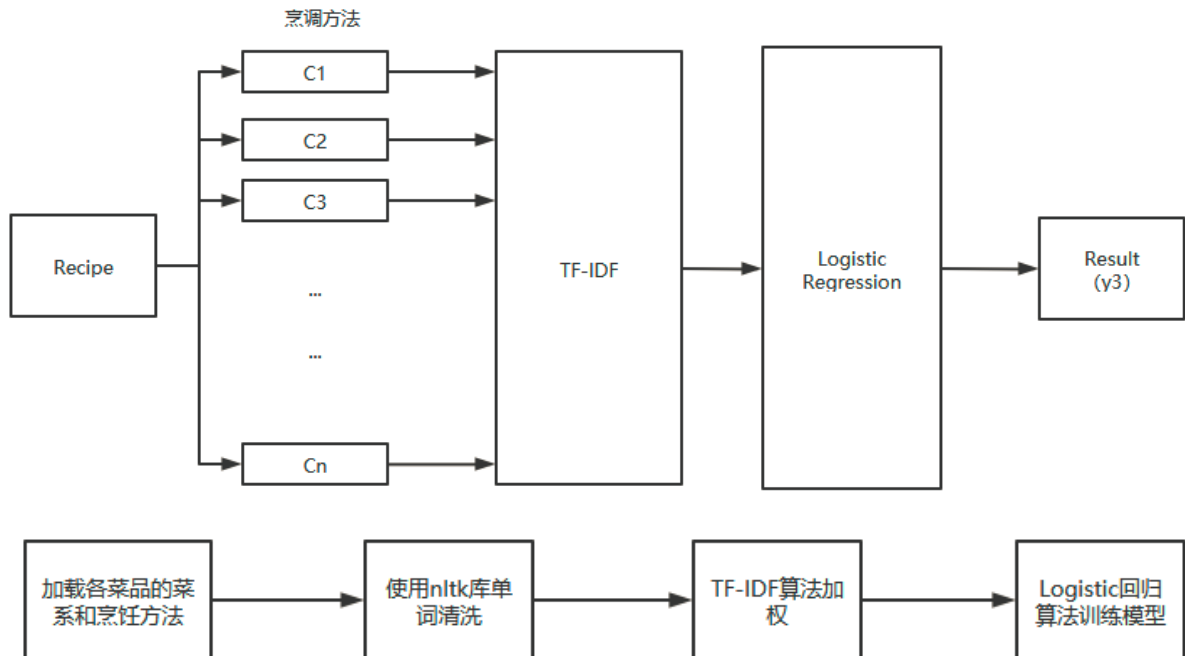


图 6: 对各菜系烹饪方法的建模流程

4.4 综合模型

以上三个模型均通过机器学习的方式，从各自的不同维度，即原料、调味料、口味烹调三个方面对给定菜谱进行分析，可预测得到该菜谱属于哪一种菜系。

然而，对于同一个菜谱的分析不一定总能返回出同一个菜系结果。因此，需要对以上三个模型的预测结果进行综合。

我们考虑使用神经网络中的反向传播算法。将上述三个模型的输出向量作为该综合模型的输入向量，增加若干 Softmax 隐含层，使用 PyTorch 进行训练。

$$y = w_1y_1 + w_2y_2 + w_3y_3 + b$$

对训练集进行训练，最终得到 w_1 、 w_2 、 w_3 的值。

至此即可刻画出不同菜系菜品的模型化差异。



以下为完整模型设计的架构图：

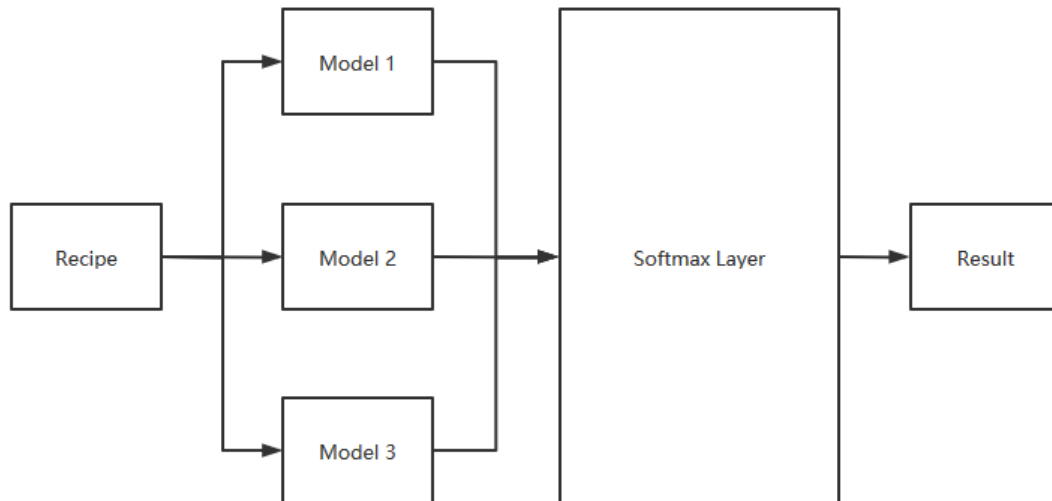


图 7：综合各维度预测菜系特征的建模流程

4.5 数字菜谱设计

对于一款烹饪机器人，其做菜时需要严格按照数字化的菜谱来进行做菜。做菜时，各种原料放的多少、放什么原料、在什么时候放什么原料等都需要以确定的方式在菜谱中给出。在此，我们设计了一中数字菜谱生成算法，以明确、量化的指令控制机器人针对用户需求进行烹饪。

在上文的模型中，由于数据集大小有限，难以刻画世界各种菜系菜品的全貌。因此，对于数据集的处理，我们采取了为每种菜系生成唯一的固定模式，其原材料、调味料配比、烹制过程更像一种大致的画像，难以代表具体的某种菜品、某个口味的实际烹调方式。

对于更加丰富的数据集，我们可以换一种处理方式：在每种菜系下训练出不同的模式，比如对于中国菜的八大菜系、家常菜和宴席风味菜、同一菜品的多种烹饪方法，分别得出不同的训练结果。

设想这样一款机器人：当用户输入菜系、主要食材、口味偏好时，机器会根据训练得到的模型，在菜系中选择与主要食材最为匹配的食材搭配，再根据用户的口味偏



好，适当调整调味料配比，匹配合适的烹饪模式，组合生成一份数字指令，操控机器人进行实际操作。其整体流程如下图所示：

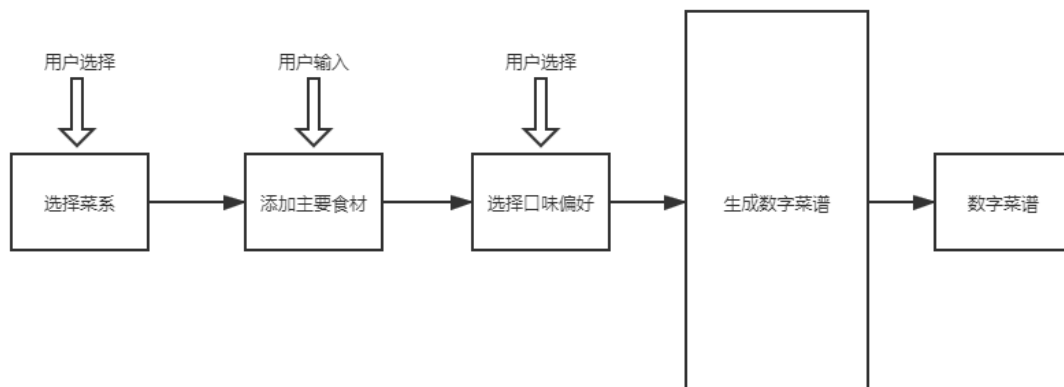


图 8：根据用户需求生成数字菜谱流程

下以蛋炒饭为例展示数字化菜谱的生成过程：

当用户选择菜系为“中式”，主要食材有“鸡蛋，米饭，胡萝卜”，口味偏好为“普通”机器会在中式菜系的模式中，匹配到主要食材组成与“鸡蛋，米饭，胡萝卜”最为接近的“米饭 500g，鸡蛋 1 个，胡萝卜 100g”，由于口味为“普通”，匹配到的调味料为“葱 10g，姜 10g，蒜 5g，油 100g，生抽 10g，老抽 5g，糖 10g，盐 5g”、烹饪方式为“炒”。

生成的菜谱以自然语言描述为：

原料：

- 剩米饭1碗
- 鸡蛋1个
- 葱1根
- 姜2片
- 蒜2瓣
- 胡萝卜1/3根

步骤 1：葱姜蒜切末，胡萝卜切成小丁。（食材的准备由人来做）

步骤 2：少量油炒葱姜蒜，再放胡萝卜碎翻炒几下。

步骤 3：倒入剩米饭炒散，再加入两勺生抽，一勺老抽，半茶匙白糖，半茶匙食盐，翻炒均匀。



步骤 4: 把米饭扒拉在一边, 加少量油倒入鸡蛋液, 快速炒散。

步骤 5: 把鸡蛋和米饭炒在一起即可, 最后可撒点葱花。

计算机程序生成对应的机器指令, 此处以 `yaml` 语言进行描述:

以下操作仅为菜谱参考示例, 不代表真实可行的做法

```
name: 蛋炒饭
ingredients:
  - 剩米饭: 500
  - 鸡蛋: 1
  - 葱: 10
  - 姜: 10
  - 蒜: 5
  - 胡萝卜: 100
operations:
  - 00:00: open
  - 00:01: add 葱, 姜, 蒜
  - 00:02: add 胡萝卜
  - 00:03: 翻炒
  - 00:04: add 剩米饭
  - ...
  - 00:10: 翻炒
  - 00:11: 撒 葱
  - 00:12: close
```

5 模型评价

5.1 优势

- 从食材、调味料配比、口味和烹调方法多方向对各菜系进行建模, 考虑较为全面, 模型要素齐全。
- 使用 TF-IDF 算法加权, Logistic 回归算法建模, 神经网络反向传播算法进行多模组整合, 数据分析可信度、准确度较高。
- 相比于实际菜谱的经验化操作, 生成的数字化菜谱对于原材料的用量、调味料配比更加明确, 易于控制和推广。
- 使用 `yaml` 编程语言设计数字化菜谱用于指导机器人操作, 简洁易懂, 且贴合实际要求。



5.2 不足

- 由于部分菜系菜品种类繁多，菜系内部也有诸多分类，每种分类又有特定的食材和烹调方法，本模型基于一个菜系整体做回归分析，缺少更加细分的分类。
- 仅基于菜系中较为典型的菜做调味料配比模型，通用性、全面性稍显不足，有待做更多的数据分析。
- 生成数字化菜谱的方法目前缺乏足够的数据集支持，仅给出一种思路。



参考文献

- [1] https://blog.csdn.net/wang_yunpeng/article/details/103401016
- [2] <https://www.kaggle.com/c/whats-cooking/data>
- [3] <https://www.cnblogs.com/paulonetwo/p/9932029.html>
- [4] 贺德富,谢龙.基于遗传算法的团餐食谱生成算法[J].计算机与数字工程,2018,46(07):1288-1292.



附录

机器学习的具体算法:

```
# 导入依赖库
import json
import codecs
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# 加载数据集
train_filename='train.json'
train_content = pd.read_json(codecs.open(train_filename, mode='r', encoding='utf-8'))

test_filename = 'test.json'
test_content = pd.read_json(codecs.open(test_filename, mode='r', encoding='utf-8'))

categories=np.unique(train_content['cuisine'])
print("一共包含 {} 种菜品, 分别是:\n{}".format(len(categories),categories))

### 将特征与目标变量分别赋值
train_ingredients = train_content['ingredients']
train_targets = train_content['cuisine']

## 统计意大利菜系中佐料出现次数, 并赋值到italian_ingredients字典中
list_italian = train_content.loc[train_content['cuisine'].isin(['italian'])]['ingredients'].reset_index(drop=True)
n = []
for j in range(len(list_italian)):
    n += list_italian[j]
italian_ingredients = pd.Series(n).value_counts().to_dict()

plt.style.use('ggplot')
fig = pd.DataFrame(italian_ingredients, index=[0]).transpose()[0].sort_values(ascending=False, inplace=False)[:10].plot(kind='barh')
fig.invert_yaxis()
fig = fig.get_figure()
fig.tight_layout()

import re
```



北京航空航天大学数学建模竞赛

```
from nltk.stem import WordNetLemmatizer
import numpy as np
def text_clean(ingredients):
    #去除单词的标点符号, 只保留 a..z A...Z 的单词字符
    ingredients= np.array(ingredients).tolist()
    print("菜品佐料: \n{}".format(ingredients[9]))
    ingredients=[re.sub('[^A-Za-z]', ' ', word) for word in component]for component in ingredients]
    print("去除标点符号之后的结果: \n{}".format(ingredients[9]))

    # 去除单词的单复数, 时态, 只保留单词的词干
    lemma=WordNetLemmatizer()
    ingredients=[" ".join([ " ".join([lemma.lemmatize(w) for w in words.split(
" ")]]) for words in component]) for component in ingredients]
    print("去除时态和单复数之后的结果: \n{}".format(ingredients[9]))
    return ingredients

print("\n 处理训练集...")
train_ingredients = text_clean(train_content['ingredients'])
print("\n 处理测试集...")
test_ingredients = text_clean(test_content['ingredients'])
from sklearn.feature_extraction.text import TfidfVectorizer
# 将佐料转换成特征向量

# 处理 训练集
vectorizer = TfidfVectorizer(stop_words='english', ngram_range=(1, 1),
                             analyzer='word', max_df=.57, binary=False,
                             token_pattern=r"\w+", sublinear_tf=False)
train_tfidf = vectorizer.fit_transform(train_ingredients).todense()

## 处理 测试集
test_tfidf = vectorizer.transform(test_ingredients)
train_targets=np.array(train_content['cuisine']).tolist()
train_targets[:10]

### 划分出验证集
from sklearn.model_selection import train_test_split
X_train , X_valid , y_train, y_valid = train_test_split(train_tfidf, train_targets, test_size = 0.2, random_state=0)

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
```



```
## 建立逻辑回归模型
parameters = {'C':[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
classifier = LogisticRegression()
grid = GridSearchCV(classifier, parameters)

grid = grid.fit(X_train, y_train)

from sklearn.metrics import accuracy_score ## 计算模型的准确率

valid_predict = grid.predict(X_valid)
valid_score=accuracy_score(y_valid,valid_predict)

print("验证集上的得分为: {}".format(valid_score))
```

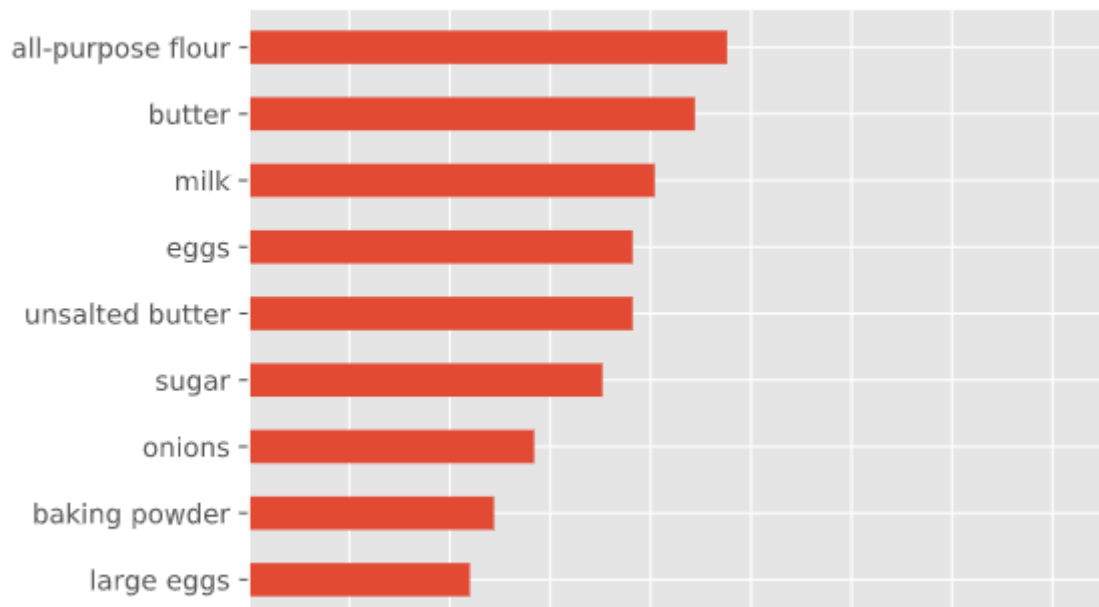


图 9：英国菜中最常见的 10 个佐料

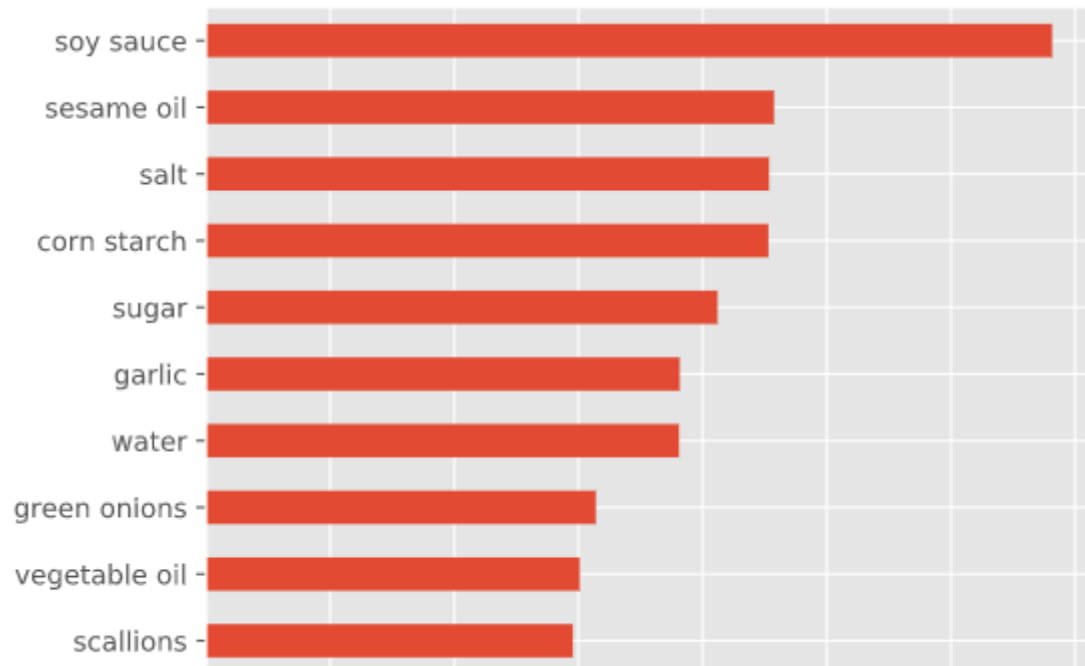


图 10: 中餐中最常见的 10 个佐料

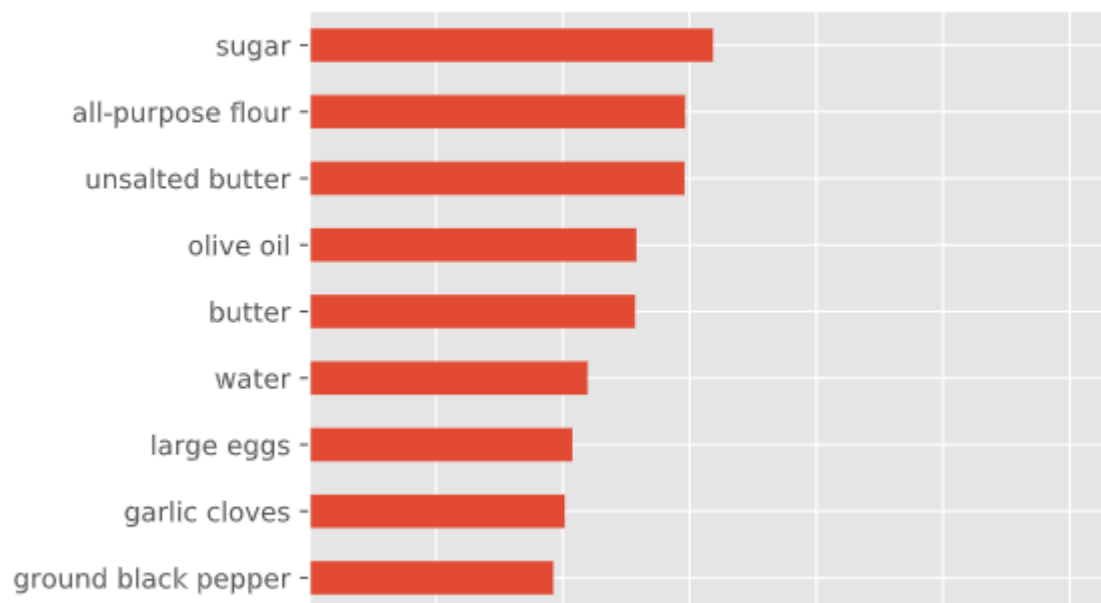


图 11: 法国菜中最常见的 10 个佐料

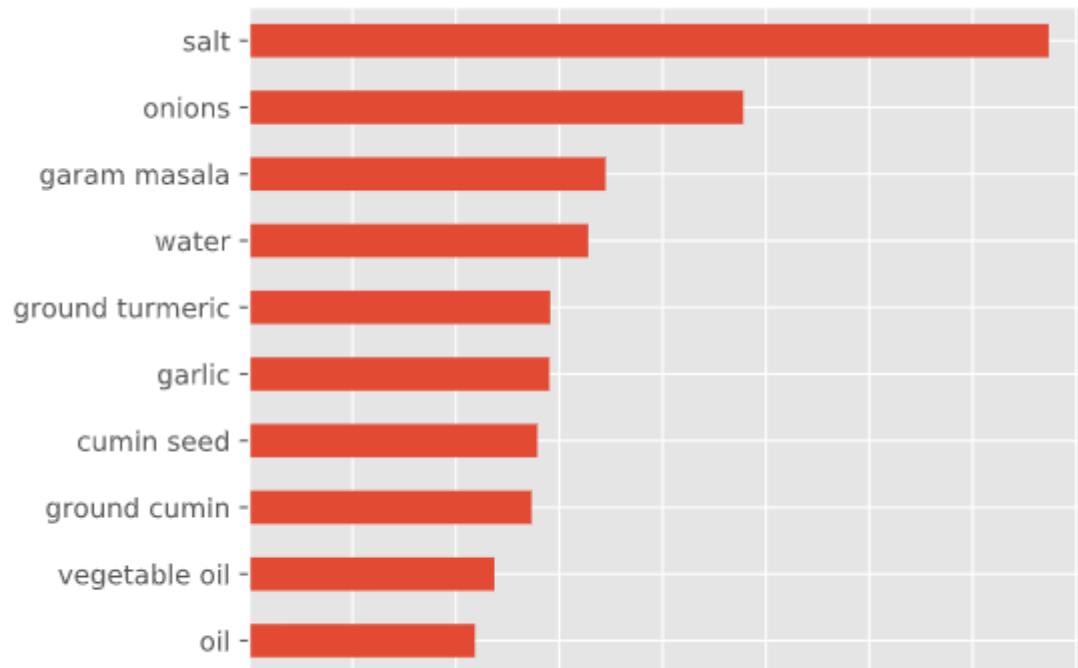


图 12: 印度菜中最常见的 10 个佐料

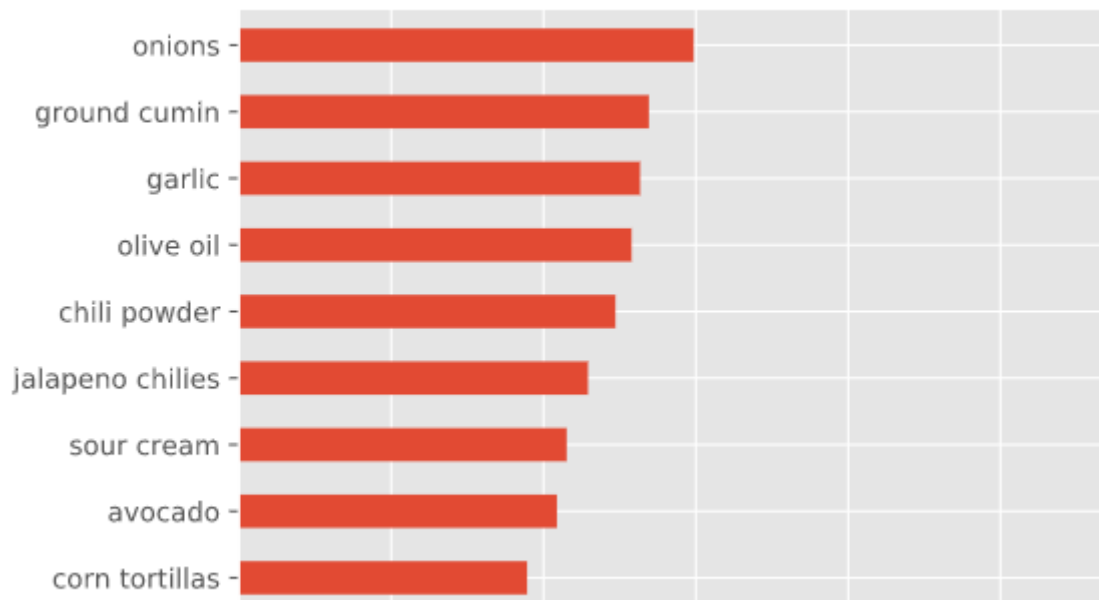


图 13: 墨西哥菜中最常见的 10 个佐料