From Pantry to Plate: Predicting Recipe Properties from Ingredients and Instructions

A Data-Driven Study Using the Extended Recipes Dataset

Dylan OBrien
School of Data Science
Wentworth Institute of Technology
Boston, MA USA
obriend7@wit.edu

ABSTRACT

This project investigates how the text of ingredient lists and short cooking instructions can be used to explain and predict key properties of recipes and support pantry-aware recommendation. Using the publicly available *Extended Recipes Dataset: 64K Dishes*, the study links ingredients and directions to four main outcome dimensions: taste profile (e.g. sweet, savory, spicy, sour, umami), perceived difficulty (easy, medium, hard), approximate preparation and cooking time, and cuisine category. Exploratory data analysis summarizes the distribution of tastes, difficulty levels, structural complexity, and cuisines, and highlights which features appear most informative. Supervised machine learning models based on TF–IDF representations of ingredient and instruction text, combined with simple numeric features, are then trained to predict taste, difficulty, and cuisine, with performance evaluated using accuracy, macro-averaged F1, and confusion matrices. Finally, a pantry-based recommender treats recipes and user-specified ingredients in a shared ingredient feature space, ranking recipes by coverage and overlap while surfacing predicted time, difficulty, and taste. The resulting pipeline provides an interpretable case study in text-driven recipe modeling and a practical tool for deciding what to cook with available ingredients.

KEYWORDS

Recipe recommendation; natural language processing; supervised learning; text mining; food computing; taste prediction; difficulty prediction; cuisine classification; pantry-based recommender.

1 Introduction

Online recipe platforms make it easy to search for dishes by title, ingredient, or cuisine, yet many home cooks still face a familiar problem: they look into the pantry or refrigerator and remain unsure what can be made with the ingredients on hand. Standard search tools rarely integrate ingredient availability with preferences for taste, preparation time, and difficulty level, and they provide limited guidance about whether a given recipe is beginner-friendly or likely to be time-consuming. This gap motivates the need for data-driven methods that can learn patterns from existing recipes and better connect free-text descriptions to the outcomes that matter to home cooks.

Recent work in food computing and recipe recommendation has demonstrated that ingredient lists and instructions can be converted into numerical features suitable for machine learning models. Text mining techniques such as tokenization and term frequency–inverse document frequency (TF–IDF) representations allow models to capture which ingredients and verbs are most characteristic of particular tastes, cooking styles, or cuisines. At the same time, structured attributes such as the number of ingredients, number of steps, and estimated preparation time provide compact summaries of recipe complexity and effort.

This project uses the *Extended Recipes Dataset: 64K Dishes* to study how ingredients, instructions, and basic structural features jointly relate to four key aspects of recipes: taste profile, difficulty, cooking time, and cuisine. The analysis is organized around five guiding questions: (1) which ingredients best predict a recipe's taste profile, (2) which features are most associated with shorter or longer preparation times, (3) which factors explain difficulty level, (4) whether cuisine type can be predicted from text, and (5) how recipes can be ranked for a given pantry while respecting taste, time, and difficulty constraints. The work combines exploratory data analysis with supervised learning models and concludes with a pantry-based recommender that ranks recipes in an ingredient feature space.

The contributions are threefold. First, the project provides a structured exploratory analysis of a large recipe corpus, characterizing distributions over tastes, difficulty levels, structural complexity, and cuisines. Second, the study builds and evaluates interpretable models for taste prediction, difficulty classification (including both text-only and text-plus-structure variants), and cuisine prediction from ingredient and instruction text. Third, the work demonstrates a practical pantry-based recommender that uses cosine similarity and simple ranking objectives to map between a user's ingredient list and concrete recipe suggestions, while overlaying predicted time, difficulty, and taste to support decision-making.

2 Data

This section describes the dataset used in the project, including its source, main fields, and the key cleaning steps required to make it suitable for exploratory analysis, prediction tasks, and recommendation. The underlying data is rich in both text and structure, with titles, descriptions, ingredients, directions, estimated times, and various derived attributes.

Each row in the dataset corresponds to a single recipe and includes several groups of columns. Core text fields include the recipe title, description, raw ingredient list, raw directions, and cleaned text fields for ingredients and instructions. These fields provide the textual basis for TF–IDF representations and token-level analysis.

2.1 Source of dataset

The analysis is based on the "*Extended Recipes Dataset: 64K Dishes*," a publicly available collection of approximately 64,000 recipes. The dataset extends an earlier recipe corpus by adding richer annotations, including taste labels, cuisine tags, dietary indicators, and health-related scores, alongside the original titles, descriptions, ingredients, and directions. For the purposes of this project, the dataset is stored locally as a single CSV file and loaded into a pandas DataFrame for cleaning, exploration, modeling, and recommendation.

2.2 Characters of the datasets

Structural and time-related fields include num_ingredients, num_steps, estimated preparation time (est_prep_time_min), estimated cooking time (est_cook_time_min), and derived totals. These columns summarize recipe complexity and effort, and are used both in exploratory plots and as numeric features in difficulty models.

Taste and cuisine fields provide taste-related labels such as primary_taste and secondary_taste, along with multi-label representations of taste; cuisine labels stored in fields such as cuisine_list; and related metadata such as course type. These columns provide target labels for the taste and cuisine prediction tasks.

Dietary and health fields include Boolean indicators such as is_vegetarian and is_gluten_free, along with broader dietary profiles and health-flag fields. Although not the primary focus of this project, these fields help contextualize recipe types and could be incorporated into extended analyses.

Before modeling, the dataset undergoes several cleaning steps to improve consistency and reduce redundancy. Exact duplicate recipes are removed so that repeated entries of the same title and content do not bias counts or model training. Ingredient strings are normalized into canonical ingredients by removing quantities, common measurement units, fractional forms (including both 1/4 and ¼), and stray characters, and by collapsing obvious variants into a shared base form. The primary_taste and secondary_taste columns are combined into a multi-label taste representation, treating both as equally informative where available and handling cases where the secondary taste is recorded as "none." Basic checks are performed for missing values in key fields, and recipes lacking essential text or labels for a given task are excluded from that specific analysis.

Overall, the dataset offers a rich combination of textual and structured information. After cleaning, most recipes include sufficient information about ingredients, steps, tastes, difficulty, time, and cuisine to support the sequence of exploratory and predictive tasks carried out in this project.

3 Methodology

The modeling approach consists of four main stages: exploratory data analysis, text and feature preparation, supervised learning for taste, difficulty, and cuisine, and a pantry-based recommendation component. Throughout, the emphasis is placed on models that are expressive enough to capture meaningful patterns while remaining interpretable and computationally manageable.

The project is framed as a set of related prediction and ranking problems driven by the five guiding questions. For taste prediction, a multi-label classification task is defined in which each recipe can simultaneously exhibit multiple tastes derived from the combined primary and secondary taste labels. For difficulty prediction, multi-class classifiers distinguish between easy, medium, and hard recipes; both a text-only model and an augmented model that incorporates structural features (such as number of ingredients, number of steps, and time estimates) are developed and compared. For cuisine prediction, a multi-class classifier is trained to distinguish among the most frequent cuisine labels using combined ingredient and instruction text. In addition, correlations between structural features and preparation time are examined to better understand what drives shorter versus longer prep times.

Because the primary signals are ingredient lists and directions, text features are constructed using standard natural language processing techniques. Ingredient and instruction text is tokenized, normalized, and transformed into TF–IDF vectors that encode which words are most distinctive for particular tastes, difficulty levels, or cuisines. For structural models, numeric features such as num_ingredients, num_steps, est_prep_time_min, and est_cook_time_min are concatenated with text-based features to form joint representations. For the pantry-focused tasks, recipes are also represented in a purely ingredient-based vector space using multi-hot or TF–IDF encodings of canonical ingredients.

For the classification tasks, regularized linear models—particularly multinomial Logistic Regression—form the baseline due to their simplicity, efficiency, and interpretability. Tree-based ensemble models, such as random forests or gradient boosting, are considered where non-linear relationships are suspected and feature importance measures are useful. Models are trained and evaluated using standard train–validation–test splits, with metrics including overall accuracy and macro-averaged F1-score; confusion matrices are used to analyze error patterns, especially for difficulty and cuisine. For multi-label taste prediction, appropriate multi-label evaluation metrics and per-label F1-scores are reported.

3.1 Data Cleaning and Canonical Ingredient Representation

The raw Extended Recipes CSV contains duplicate entries, noisy ingredient strings, and occasionally missing labels. As a first step, exact duplicate recipes were removed based on a combination of title and core content fields (ingredients and directions), so that repeated entries of the same dish did not inflate counts or bias model training. Rows missing essential text or labels for a given task (for example, lacking both ingredient text and taste labels) were excluded from that specific analysis but retained for other tasks where they remained usable.

Ingredient fields were then normalized into a canonical ingredient representation. This process removed quantities (e.g., 1 cup, 2 tbsp), measurement units (teaspoon, cup, ounces), and common fractional forms (including both 1/4 and ¼), as well as leftover dashes or stray characters that appeared from parsing. Simple heuristic rules were used to collapse closely related variants (such as salt, salt to taste, 1/2 tsp salt) into a single canonical token. At the same time, meaningful multi-word ingredients (e.g., olive oil, garlic powder, soy sauce, yukon gold potatoes) were preserved where possible to avoid losing important distinctions. This canonical ingredient vocabulary underlies both the ingredient-only models and the pantry-based recommender.

3.2 Text Feature Construction

Because the primary signals in this project come from ingredient lists and directions, textual features were constructed using standard natural language processing steps. Ingredient and instruction text was tokenized, normalized, and transformed into TF–IDF representations that encode the relative importance of tokens across the corpus.

For each supervised task, a subset of these text sources was used. For taste prediction, TF–IDF features derived mainly from ingredient text were used, since tastes are strongly driven by ingredient composition. For difficulty prediction, TF–IDF features from directions (to capture technique verbs and complexity cues) were employed, optionally concatenated with ingredient text. For cuisine prediction, TF–IDF features from both ingredients and instructions were combined to capture both flavor building blocks and cooking styles.

Sparse TF–IDF matrices were used directly in linear models, avoiding the need for dense embedding layers. For models that combined text with numeric features, standardized structural variables (such as number of ingredients and steps) were horizontally concatenated with the TF–IDF features to form joint feature vectors.

3.3 Taste Prediction (Multi-Label Classification)

Taste prediction was formulated as a multi-label classification problem. Each recipe can express more than one taste, so the primary_taste and secondary_taste fields were combined into a single set of taste labels per recipe. Rows where the secondary taste was recorded as none were treated as having only a single taste, and the none pseudo-label was excluded from the modeling labels.

Taste labels were binarized using a multi-label encoder, producing one binary indicator per taste category (e.g., sweet, savory/umami, spicy, sour, bitter). TF–IDF features derived from canonical ingredients formed the input feature matrix. A regularized linear classifier (Logistic Regression wrapped in a one-vs-rest multi-label strategy) was chosen for its balance of performance and interpretability; coefficients for each taste label provide direct insight into which ingredients are most strongly associated with that taste.

Models were trained on a subset of recipes with valid taste labels and evaluated on a held-out test set. Evaluation focused on overall multi-label accuracy, macro-averaged F1-score, and per-taste F1-scores, to highlight which tastes are easiest or hardest to predict. Confusion at the label level (for example, sweet vs savory) was analyzed by inspecting the top predictive tokens per taste and verifying that the model's associations aligned with culinary intuition.

3.4 Difficulty Prediction (Multi-Class, Text vs Text+Structure)

Difficulty prediction was treated as a three-class classification task with labels easy, medium, and hard. Two main model variants were constructed. The first was a text-only model that used TF–IDF representations of recipe directions (and optionally ingredients) as input features, with the goal of capturing linguistic cues of complexity such as advanced cooking techniques, multi-step instructions, and specialized equipment.

The second variant was a text-plus-structure model that combined the same TF–IDF text features with structural numeric variables, including num_ingredients, num_steps, est_prep_time_min, and est_cook_time_min. This variant allowed the model to combine language with directly observable measures of complexity and time.

In both cases, multinomial Logistic Regression with regularization was used as the primary classifier. Recipes with missing difficulty labels were dropped from the training and evaluation splits. Class imbalance was modest, but stratified splitting was used to preserve the proportion of easy, medium, and hard recipes across train and test sets.

Performance was evaluated using overall accuracy, macro-averaged F1-score, and confusion matrices normalized by true class. Comparing confusion matrices for the text-only and text-plus-structure models made it possible to quantify how much additional accuracy was gained by including numeric features and which classes benefited most. Coefficients for numeric features were also inspected to understand which structural aspects most strongly influence the predicted difficulty.

3.5 Cuisine Prediction (Multi-Class Classification)

Cuisine prediction was framed as a multi-class text classification problem. The dataset contains many possible cuisine labels, but

some occur only rarely. To build a stable classifier and yield interpretable confusion matrices, the analysis focused on the most frequent cuisines in the cleaned data, such as the top five by recipe count.

For each recipe, a single primary cuisine label was derived from the cuisine_list field by selecting the first or most prominent cuisine entry when multiple were present. Recipes without any cuisine annotation or with extremely rare cuisine types were excluded from this task.

Combined ingredient and instruction text was vectorized using TF–IDF. A multinomial Logistic Regression classifier was trained to predict the cuisine label, using a standard train–test split. Evaluation relied on overall accuracy, macro-averaged F1-score, and confusion matrices restricted to the top cuisines. Misclassifications were examined to identify patterns where cuisines overlap in ingredients and techniques.

3.6 Pantry-Based Recommendation Pipeline

The pantry-based recommendation component operationalizes the idea of what can be cooked with what is on hand using the ingredient representations developed earlier. The pipeline consists of several steps. First, a user's pantry items are entered as free-text ingredient names, matched against the canonical ingredient vocabulary, and encoded as either a multi-hot vector or a TF–IDF-weighted vector in the same feature space used for recipes.

Second, each recipe is represented by its canonical ingredient vector in the same space as the pantry. For each recipe, the number of overlapping ingredients between the pantry and the recipe is computed (overlap), and the overlap is normalized by the number of ingredients in the recipe to yield coverage, the fraction of a recipe that can be made from the pantry. Cosine similarity between the pantry vector and each recipe vector is also computed to capture overall ingredient-space similarity.

Third, two ranking modes are defined. In coverage mode, the system prioritizes recipes that are almost fully cookable with the pantry (high coverage), which is useful when minimizing missing ingredients is important. In overlap mode, the system prioritizes recipes that use as many pantry items as possible (high overlap), which is useful when the goal is to use up ingredients. Users can switch between modes and optionally constrain recipes by maximum preparation time, desired difficulty levels, and preferred tastes.

Fourth, for top-ranked recipes, previously trained taste, difficulty, and cuisine models are applied. Each recommended recipe is annotated with predicted difficulty, dominant tastes, and time summaries, providing a richer view than rankings alone. Finally, an ipywidgets-based graphical interface in Jupyter allows users to browse available canonical ingredients through a search panel, add selected ingredients to the pantry, set filters, and view recommended recipes and detailed information such as titles, descriptions, ingredients, and directions.

4 Results

This section presents empirical findings from exploratory data analysis, supervised models for taste, difficulty, and cuisine, and the pantry-based recommendation system. The results are organized around the five guiding questions outlined in the introduction.

4.1 Exploratory Analysis of Taste, Difficulty, and Structural Complexity

Initial exploratory analysis summarized how tastes, difficulty levels, and structural features are distributed across the cleaned recipe corpus. Combining the primary and secondary taste annotations into a single multi-label representation revealed that a substantial fraction of recipes carry savory/umami or sweet labels, while other tastes such as spicy, sour, and bitter appear less frequently but are still well represented. This imbalance suggests that models will see more training examples for certain tastes, which is reflected in later classification performance.

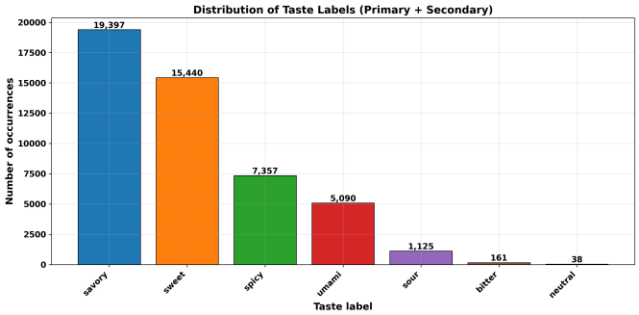

Figure 1: Distribution of combined taste labels across the corpus

Difficulty labels show that most recipes are annotated as easy or medium, with hard recipes forming a smaller but non-trivial subset. This distribution supports the use of multi-class classification while highlighting the importance of macro-averaged metrics to avoid over-emphasizing the majority classes.

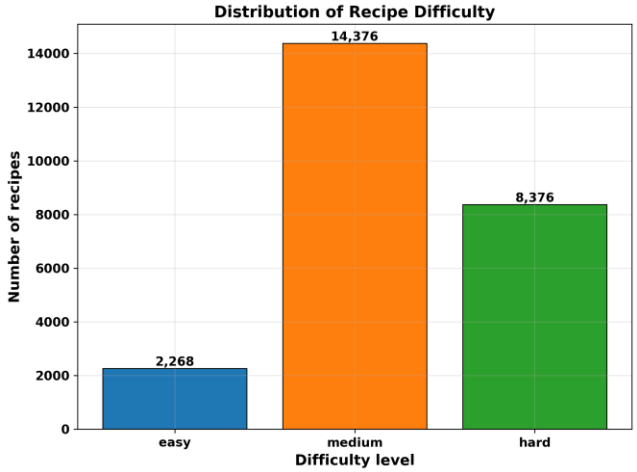

Figure 2: Distribution of difficulty labels (easy/medium/hard)

Structural summaries of num_ingredients, num_steps, and time estimates show a heavy-tailed pattern: many recipes are relatively simple, with modest ingredient lists and short preparation times, while a smaller set of recipes are much more complex, with long ingredient lists, many steps, and extended total time. Plots of prep

and cook time highlight that prep times tend to be shorter and more tightly clustered than cook times, but both exhibit long right tails.

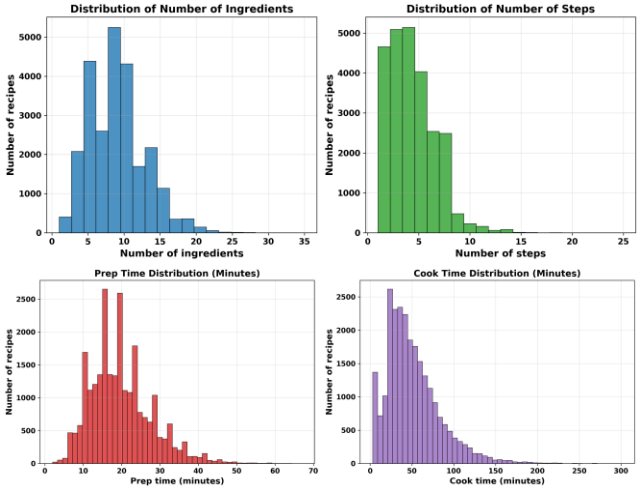

Figure 3: Distributions of number of ingredients, number of steps, preparation time, and cooking time

A heatmap of taste versus difficulty further indicates that some tastes are more common in simple recipes (for example, sweet dishes and many savory everyday meals), while others are more evenly spread or skewed toward more complex dishes. This motivates later models that link taste and difficulty to both ingredients and structure.

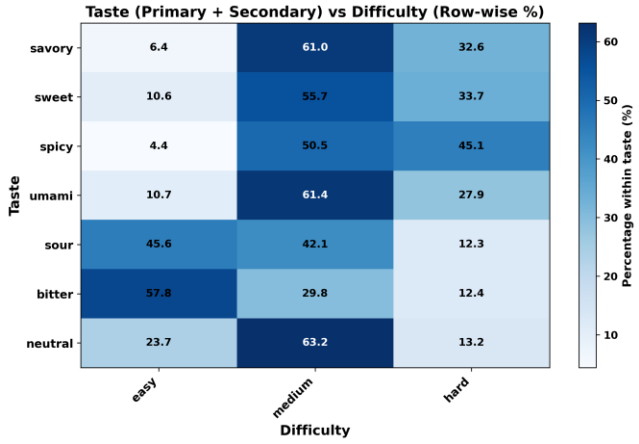

Figure 4: Heatmap of combined taste labels versus difficulty level

4.2 Taste Prediction from Ingredients

Taste prediction was framed as a multi-label classification problem using TF–IDF representations of canonical ingredients as input. Logistic Regression with a one-vs-rest strategy achieved reasonably strong performance across the major taste labels, with macro-averaged F1-scores indicating that the model can reliably distinguish between sweet, savory/umami, and spicy profiles, and somewhat lower but still informative performance on less frequent tastes such as sour and bitter.

Per-label F1-scores show that tastes with higher support (sweet and savory/umami) are easier to predict, while rarer tastes exhibit more variability. Inspecting the learned coefficients confirms that the model's behavior aligns with culinary intuition. For example, the sweet classifier places high positive weight on tokens such as sugar, brown sugar, honey, vanilla, and chocolate, whereas the spicy classifier emphasizes chili powder, cayenne, jalapeño, red pepper flakes, and related ingredients. The savory/umami classifier is driven by garlic, onion, butter, olive oil, cheese, soy sauce, broth, parmesan, and similar flavor bases.

Top 15 indicative tokens per taste label (multi-label One-vs-Rest model):

	taste_label	token_1	token_2	token_3	token_4	token_5
0	bitter	beer (10.96)	kale (8.52)	coffee (7.63)	vodka (7.21)	gin (4.64)
1	neutral	water (5.82)	grams (3.44)	eggs (3.03)	pressure (2.98)	polenta (2.90)
2	savory	butter (8.80)	salt (7.60)	cheese (4.82)	and salt (4.32)	oil (3.25)
3	sour	lemon (8.51)	juice (5.90)	buttermilk (5.65)	yogurt (5.58)	lemon juice (4.70)
4	spicy	seasoning (21.42)	ginger (19.82)	cayenne (12.86)	chili (11.33)	spice (10.28)
5	sweet	sugar (20.12)	wine (13.47)	sweet (11.56)	honey (9.57)	brown sugar (7.86)
6	umami	milk (7.97)	cheese (5.31)	cream (4.59)	egg (4.31)	parmesan (3.54)

Table 1: Top 15 canonical ingredient tokens per taste label (only 5 shown here for clarity)

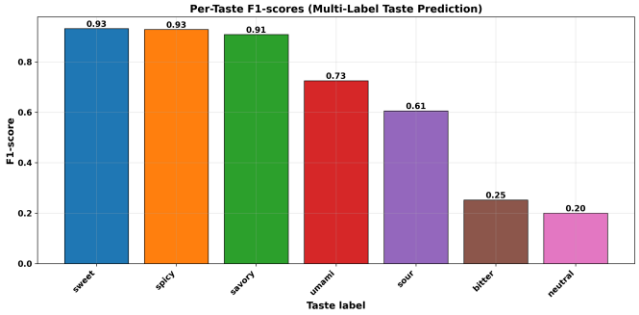

Figure 5: Per-taste F1-scores for the multi-label taste prediction model

Overall, the results indicate that canonical ingredients alone carry enough signal to recover taste labels with good fidelity when there is enough available data, and that the model's most important features provide a transparent mapping between taste categories and ingredient space.

4.3 Structural Drivers of Preparation Time

To quantify how structure relates to effort, correlations between est_prep_time_min and several predictors were examined, including num_ingredients, num_steps, and est_cook_time_min. Scatter plots and correlation bar charts show that number of steps and number of ingredients are both positively associated with longer preparation times, with the number of steps typically emerging as the strongest single structural correlate. Cook time also correlates with prep time but less strongly, suggesting that recipes requiring long stove or oven time do not always require extended hands-on preparation.

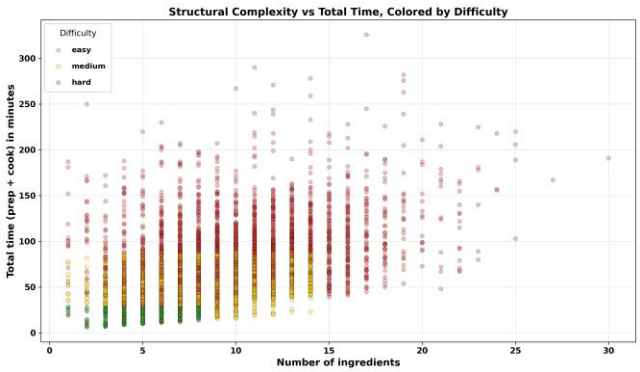

Figure 6: Structural complexity vs total estimated time for recipes, colored by difficulty

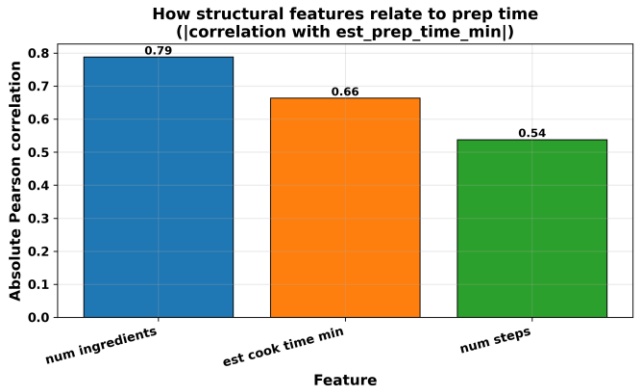

Figure 7: Correlation of preparation time with number of ingredients, number of steps, and cooking time

Boxplots of prep and cook time by difficulty level further reinforce these patterns. Easy recipes tend to have fewer steps and shorter prep times, while medium and hard recipes shift toward higher medians and wider spreads. Hard recipes in particular exhibit a broad range of preparation times, consistent with both complex showpiece dishes and multi-day or multi-component recipes.

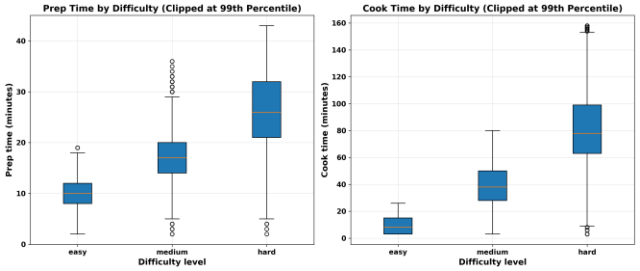

Figure 8: Boxplots of preparation and cooking time by difficulty level

These findings support the intuitive notion that structural complexity is a key driver of effort and provide a natural set of numeric features for difficulty models and for filtering recipes by time in the recommender system.

4.4 Difficulty Prediction with and without Structural Features

Two families of multi-class classifiers were trained to predict difficulty: a text-only model using TF–IDF features of directions (and optionally ingredients), and an augmented model combining these text features with structural variables (num_ingredients, num_steps, est_prep_time_min, and est_cook_time_min). Both models achieved reasonable overall accuracy and macro-averaged F1-scores, showing that difficulty labels can be learned from available data.

Confusion matrices reveal important differences between the two approaches. The text-only model correctly identifies many easy recipes but shows more confusion between medium and hard classes, likely because both can involve advanced verbs and similar cooking terminology. When structural features are added, the model becomes better calibrated: misclassifications between medium and hard are reduced, and hard recipes with long ingredient lists and numerous steps are more reliably distinguished from medium recipes.

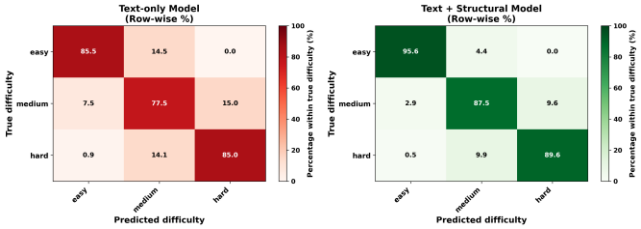

Figure 9: Normalized confusion matrices for the text-only difficulty classifier and text+structure difficulty classifier

An inspection of learned coefficients for the numeric features shows positive contributions from the number of steps, number of ingredients, and total time towards the harder difficulty classes, and negative contributions towards easy recipes. Combined with the earlier correlation analysis, this supports the conclusion that both text and structure are important for difficulty, but structural features provide a particularly strong and interpretable signal.

From a practical standpoint, these models enable identification of beginner-friendly recipes as those with short, simple instructions, few ingredients and steps, and high predicted probability of the easy class. They also help flag recipes that, despite seemingly simple titles, may be more complex due to lengthy instructions or advanced technique language.

4.5 Cuisine Prediction from Ingredients and Instructions

Cuisine prediction was framed as a multi-class classification task over the most frequent cuisine labels in the dataset. After deriving a single primary cuisine label per recipe and restricting analysis to the top cuisines by frequency, a TF–IDF plus Logistic Regression model was trained using combined ingredient and instruction text.

The resulting classifier exhibits meaningful discriminative power: overall accuracy and macro-averaged F1-scores indicate that common cuisines such as American, Italian, Indian, Mexican, and British can be distinguished from one another based on ingredient

and instruction patterns. A normalized confusion matrix focused on the top cuisines shows that most predictions fall along the diagonal, with off-diagonal entries reflecting culinary overlap. For instance, Mediterranean and Middle Eastern cuisines share many ingredients and techniques with other European or fusion categories, leading to some misclassifications.

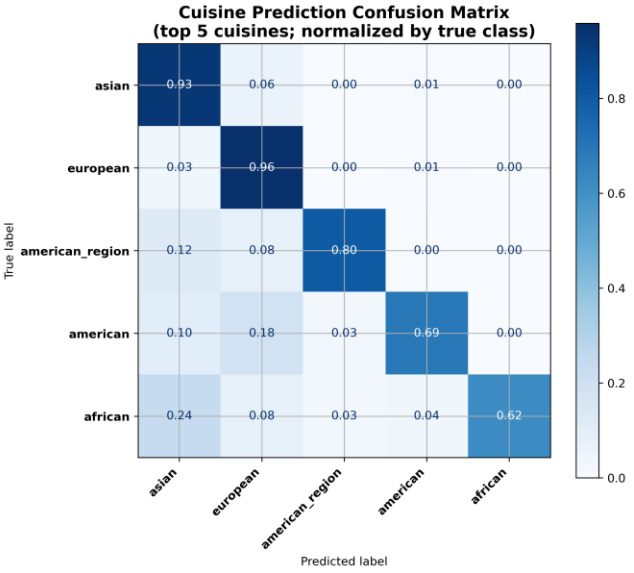

Figure 10: Normalized confusion matrix for top cuisines predicted by the cuisine model

These results suggest that cuisine carries a distinct textual and ingredient signature, but that fine-grained distinctions between closely related cuisines are challenging, especially when labels are noisy or overlapping. The cuisine model nonetheless provides useful annotations for the recommendation system and helps contextualize recipes beyond difficulty and taste.

4.6 Pantry-Based Recommendation and Case Study

The pantry-based recommendation system operationalizes the earlier models and exploratory findings into a practical tool. Given a set of user-specified pantry ingredients, the system encodes the pantry and all recipes in a shared canonical ingredient space, computes overlap and coverage for each recipe, and then ranks recipes according to either a coverage-focused or overlap-focused objective.

In a coverage-focused scenario, the system surfaces recipes for which most required ingredients are already on hand, minimizing the number of missing ingredients. In an overlap-focused scenario, the system instead emphasizes recipes that consume as many pantry items as possible, which is ideal for using up ingredients. Example bar charts for a fixed test pantry show how different recipes trade off high coverage and high overlap, illustrating that some recipes are nearly fully cookable but use only a few pantry items, while others use many pantry items but still require several additional ingredients.

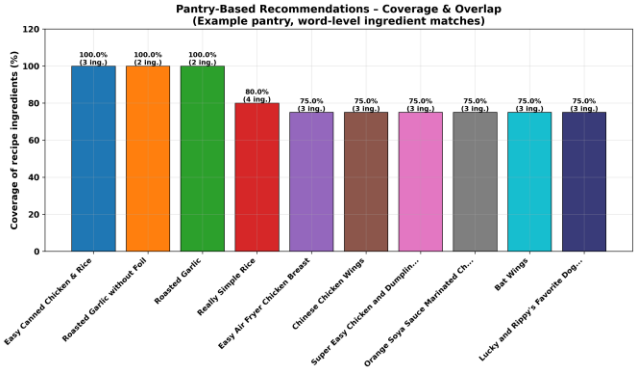

Figure 11: Coverage- and overlap-based rankings for a sample pantry

A two-dimensional PCA projection of the ingredient feature space further visualizes how the user's pantry relates to recommended recipes: the pantry vector appears as a point in ingredient space, surrounded by recipes with similar ingredient profiles. Recipes closer to the pantry in this embedding tend to have higher overlap and coverage, reinforcing the geometric interpretation of the recommender.

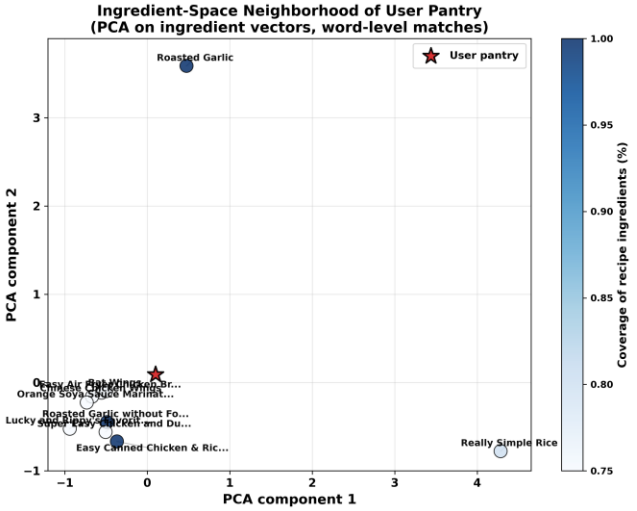

Figure 12: PCA projection of ingredient-space neighborhood showing pantry and recommended recipes

Finally, application of the trained difficulty and taste models to the top-ranked recipes allows each recommendation to be annotated with predicted difficulty and dominant tastes. Summary charts for recommended recipes indicate how the chosen ranking mode and filters (for example, difficulty, maximum prep time, taste preferences) shift the mix of suggested recipes. In combination with the interactive Jupyter GUI, these results demonstrate that an ingredient-based representation, augmented with learned taste, difficulty, and cuisine models, can support transparent, controllable recipe recommendation that answers practical questions about what to cook with available ingredients

and how challenging or time-consuming those recipes are likely to be.

5 Discussion

The results of this project show that relatively simple, interpretable models built on top of cleaned ingredient and instruction text can capture much of the structure behind taste, difficulty, time, and cuisine in a large recipe corpus. At the same time, the findings highlight important limitations of the data and methods, and suggest several directions for further work.

From the perspective of taste modeling, the multi-label classifiers confirm that taste is strongly encoded in the ingredient list. The top predictive tokens for each taste category align closely with human expectations: sugars and flavorings for sweet recipes, chilies and spice blends for spicy dishes, and savory pantry staples such as garlic, onion, and cheese for umami-heavy meals. This agreement between model coefficients and culinary intuition suggests that the canonical ingredient cleaning procedure was largely successful in recovering meaningful ingredient signals. However, the lower performance on rarer tastes also points to limitations of label coverage and highlights that some taste categories may be underrepresented or inconsistently annotated in the underlying dataset.

The structural analysis of preparation time and difficulty provides a complementary view. Correlations between the number of steps, number of ingredients, and preparation time are intuitive, but quantifying them allows more precise reasoning about what makes a recipe simple. The augmented difficulty model demonstrates that structural features significantly help distinguish medium from hard recipes, reducing confusion that the text-only model struggles to resolve. This supports the idea that difficulty is not merely a linguistic phenomenon: two recipes describing similar techniques may differ substantially in difficulty simply because one requires more components, preparation stages, or waiting periods.

Cuisine prediction results fall somewhere in between. On the one hand, the ability to classify the most common cuisines with reasonable accuracy confirms that cuisine carries a distinct textual and ingredient signature; on the other hand, confusion between closely related or overlapping cuisines reveals both model and dataset limitations. Many recipes blend influences from multiple traditions or are labeled with broad categories, and the reduced top-cuisine setting partially sidesteps this complexity. A more nuanced model might treat cuisine as multi-label or hierarchical rather than a single flat class.

The pantry-based recommender brings these components together into a practical system. It demonstrates that a relatively lightweight ingredient representation, combined with simple overlap and coverage scores, can support a flexible notion of what to cook with what is available. The ability to switch between coverage-focused and overlap-focused ranking modes, and to overlay predicted difficulty and taste, makes the recommendations more actionable than a simple ingredient search. The interactive Jupyter GUI illustrates how data science artifacts (models, similarity measures, and visualizations) can be assembled into an end-user tool without requiring a full production web stack.

At the same time, several limitations should be acknowledged. First, the dataset is static and does not incorporate real user feedback; there is no mechanism to learn from which recipes users actually cook or enjoy. Second, all models operate on text and aggregate structural features, ignoring important aspects such as ingredient quantities, ordering of steps, and timing interactions (for example, marinating or resting periods). Third, the canonical ingredient cleaning is heuristic and may collapse distinct ingredients or fail to unify all variants, especially for nuanced products and branded items. Fourth, the use of TF–IDF and linear models, while interpretable and efficient, leaves performance gains from more expressive neural architectures on the table.

Finally, there are broader considerations around personalization and inclusivity. The current system treats all users as identical and does not account for dietary restrictions, equipment availability, or cultural preferences beyond those reflected in the dataset's cuisine labels and dietary flags. Extending the recommender to incorporate user-level profiles, nutritional constraints, and fairness-aware evaluation would be an important next step for any real-world deployment.

6 Conclusion

This project set out to explore how ingredient lists and short instructions can explain and predict key properties of recipes, and how those insights can support pantry-aware recommendation. Using the Extended Recipes Dataset as a foundation, the work combined exploratory analysis, supervised learning, and recommender design to address five guiding questions about taste, time, difficulty, cuisine, and pantry-based ranking.

The empirical results show that canonical ingredients provide a strong signal for multi-label taste prediction, that structural features such as number of steps and ingredients are closely linked to preparation time, and that difficulty labels can be predicted more reliably when both text and structure are considered together. Cuisine type is partially predictable from combined ingredient and instruction text, especially when the focus is restricted to a set of frequent cuisines. Together, these findings offer a coherent view of how textual and structural signals interact in recipe data.

Building on these models, the pantry-based recommender demonstrates a concrete application: given a list of pantry ingredients, the system maps both pantry and recipes into a shared ingredient feature space, computes overlap and coverage metrics, and ranks recipes according to user-selectable objectives. Overlaying predicted difficulty, taste, and time makes the recommendations more informative and supports use cases such as finding beginner-friendly recipes, time-constrained meals, or ways to use up specific ingredients. The accompanying Jupyter-based GUI illustrates how these components can be presented in an interactive and explorable form.

Beyond its immediate culinary application, the project serves as a compact case study in text-driven prediction and recommendation. It shows how standard tools (pandas, TF–IDF, linear models, and

simple visualizations) can be combined to answer domain-specific questions and build a working system end-to-end. Future work could extend the approach with more advanced neural text encoders, richer user modeling, integration of nutritional and cost information, and deployment of the recommender as a standalone web service or mobile application. Nevertheless, the current pipeline already demonstrates that data-driven analysis of recipes can bridge the gap between raw text and everyday decision-making about what to cook.

ACKNOWLEDGMENTS

The author thanks Dr. Wei Pang for guidance, feedback, and weekly Python coding demonstrations in DATA-6150: Data Science Foundations at Wentworth Institute of Technology, which strongly influenced the implementation style and structure of this project. The author also acknowledges Wafaa EL HUSSEINI for publishing the Extended Recipes Dataset: 64K Dishes on Kaggle, which served as the primary data source for all analyses and models in this work. Finally, appreciation is extended to the School of Computing and Data Science for providing computational resources and a supportive environment for project-based learning.

REFERENCES

[1] W. EL HUSSEINI, "Extended Recipes Dataset: 64K Dishes," Kaggle, 2025. [Online]. Available: https://www.kaggle.com/datasets/wafaaelhusseini/extended-recipes-dataset-64k-dishes

[2] W. Pang, "Weekly Coding Demos (Python)," unpublished class material, DATA-6150: Data Science Foundations, School of Computing and Data Science, Wentworth Institute of Technology, Boston, MA, USA, 2025.

[3] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A Survey on Food Computing," ACM Computing Surveys, vol. 52, no. 5, pp. 1–36, 2019.

[4] S. Chhipa, V. Berwal, T. Hirapure, and S. Banerjee, "Recipe Recommendation System Using TF-IDF," ITM Web of Conferences, vol. 44, p. 02006, 2022.

[5] A. Salvador, N. Hynes, Y. Aytar, J. Marín, F. Ofli, I. Weber, and A. Torralba, "Learning Cross-Modal Embeddings for Cooking Recipes and Food Images," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3068–3076.