PA026 - Project report

Jonáš Tichý (540490)

June 23, 2023

1 Project topic introduction

This projects topic is called `Food preference classifier` for my picky eater girlfriend Magda. I chose this topic because my girlfriend is very particular about the foods she likes and it is a real challenge to prepare her a meal that she likes. Thus, I decided to find out if a machine learning algorithm can solve this rather difficult task and help me in the food making process.

The first and most difficult part of the process is the dataset creation described in section 2. Classifier tuning and experimentation is described later in section 4. In sections 6 and 7 the web interface for user interaction with the classifier and run instructions are presented respectively.

2 Dataset creation

Creation of a useful dataset was a challenging and tedious manual task. To provide as much information as possible for the classifier, the dataset had to contain individual ingredients for a given meal along with a preference numerical classification. The most difficult part of the process was to manually create ingredient lists for individual meals. Also, the amount of meals in the dataset should be as high as possible in order to fit with sufficient classification accuracy. In the end the final dataset comprised of `192` meals that were manually created and annotated. The ingredients in the meals are represented as boolean values indicating presence/absence in a given meal and the classification is numeric in the scale `1-3` where `1` is the worst and `3` is the best.

To enhance the dataset, ChatGPT was also experimented with to create food combinations along with ingredient lists. However, a lot of manual editing was needed due to various hallucination problems and sometimes inaccurate food ingredient decomposition. While ChatGPT provided some creative and interesting combinations, its tendency to occasionally include non-existent or improbable ingredients meant that each suggested meal had to be carefully reviewed and corrected, which meant that it did not provide a significant and meaningful increase in productivity in the end.

3 Dataset characteristics

The final dataset consists of 192 records with total of 196 features. Each record represents a particular food and each feature represents a particular ingredient (or a meta feature discussed below) that is either present or absent in a given meal (thus all features are represented as boolean values of either `true` or `false`).

Additionally, the dataset has been enriched with 5 additional meta features holding information about the overall meal characteristics. These features are called `sweet`, `contains_fruit`, `fried`, `contains_meat`, and `vegetable_pieces`. These self explanatory meta features provide additional information to the classifier to base it's predictions on. These meta features were hand picked in a way to best represent important food characteristics for the target subject's food preferences. The classifier can then work with these features to incorporate them into the decision process. All of these additional meta features are also of boolean values and are present for all foods in the dataset.

The dataset has slight class imbalance, but not overly skewed one, the overall class distribution is 70 samples for class 3, 66 samples for class 2 and 56 samples for class 1. The train-test split was stratified in order to preserve the original class distribution.

4 Classifier experiments

Since this project is concerned with classification, numerous classifiers were tested, mainly from the `scikit-learn` library, such as `RandomForest`, `SVM`, and `HistGradientBoostingClassifier`. However, testing results from cross validation had shown that the classifier has very insignificant impact on the overall classification performance and the

performance is closely tied to the dataset quality and size. In the end, the `AutoGluon`[1] framework was chosen to find the optimal ensamble of classifiers for this particular task since it offered the best overall performance and was quite easy to use.
Final parameters for training were by using the `best_quality` preset and training for 10 minutes on M1 MacBook Pro. The best quality classifier achieved F1 score of `0.53` on the provided test partition of the dataset. Dataset stratification was used during the train-test split to ensure similar distributions in training and testing datasets.

5 Testing results for different methods

In this section, we present the evaluation metrics for three different classifiers: `RandomForest`, `MLP`, and the `AutoGluon` ensemble. The evaluation metrics used are precision, recall, and F1 score. These metrics were computed on the test partition of the dataset using stratification to ensure similar distributions in training and testing datasets.

The table below summarizes the performance of each classifier:

Classifier	Precision	Recall	F1 Score
RandomForest	0.5300	0.5128	0.5169
MLP	0.5175	0.5128	0.5169
AutoGluon Ensemble	**0.5312**	**0.5384**	**0.5342**

Table 1: Comparison of Precision, Recall, and F1 Score for Different Classifiers

These results indicate that while the choice of classifier does have some impact on performance, the overall differences are relatively modest. The `AutoGluon` ensemble provided the best balance between precision and recall, leading to the highest F1 score among the tested methods.

6 Web interface

The web interface was created in order to enable easy interaction with the trained classifier. The frontend was implemented in the framework `Svelte` with the use of the `Skeleteon UI` library. User can select ingredients that are in the dish he wants to know the numerical rating for. The web interface communicates with the backend to show the result to the user.

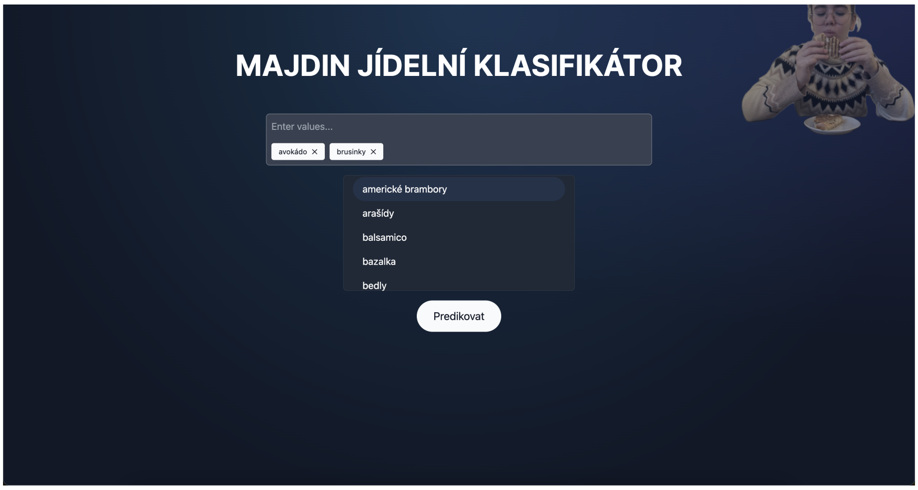

Figure 1: Main page overview with input to select ingredients.

7 Run instructions

The whole project can be run in Docker containers via `docker compose` script in the root of the project. Please follow `README.md` instructions located in the root of the project to see which commands to execute in order to get the whole project running. Since a backend server is needed, it was not practical to host this project on a public domain due to costs and complexity of deployment. Thus I opted for Dockerized environment instead.

Figure 2: Score screen.

8 Future work

The results have shown that by adding more high quality labeled data the performance steeply increases. In the future I would like to experiment with language model processing of full text internet recipes of particular foods to bypass the problematic dataset construction step that is time consuming and prone to manual errors, perhaps by using tools such as [3]. Also, additional information such as confidence or SHAP explanations [2] to help the user better understand the influence of individual features.

References

[1] Nick Erickson et al. *AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data*. 2020. arXiv: 2003.06505.

[2] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[3] Nikhil Suwalka et al. "Food Genie, Recipe Search Algorithm Using Web Scraping". In: Aug. 2023, pp. 1–6. DOI: 10.1109/ASIANCON58793.2023.10270597.