**Allergen Identifier Model**

Ouk Chan Vathanak

Machine Learning with Python

**1. Brief Summary**

This project was assigned to me by my Machine Learning lecturer. It is aimed at making a basic machine learning model. This model that I designed aims to be able to identify active allergens in foods and from that, it should be able to classify whether or not new foods contain allergens. This may help people who suffer from such allergies by informing them whether a food they plan to consume contains harmful ingredients.

**2. Societal Impact of the Project**

This project has potential to positively affect peoples’ lives as its capability to predict allergens that are active in food, though rusty at first, could grow and evolve into a more accurate and reliable model later on. That could help people all around the world avoid having nasty reactions to allergens in foods they eat.

**3. Research Questions**

1. What is the problem that people with allergic reactions meet often?

A: I believe that the problem they often run into is accidentally consuming food that has an allergen that they are allergic to without knowing it.

2. Why should we make a model to determine allergens in foods?

A: We should make such a model so that we can easily identify if a certain food contains any allergens or not, so that people who are allergic to it can avoid them.

3. How can we determine the allergens in foods efficiently?

A: We can train the model on a dataset of foods and their allergens so that the model can accurately identify allergens when a new food comes in.

**4. Approach to the Questions**

In addressing these questions, my approach was:

* Go through the dataset to determine the presence of imbalance
* Pre-process the data to avoid issues like overfitting and underfitting
* Identified the problem as classification problem and tested four possible models, then picking the best one
* Analyze the models’ performances and relationships, then derive useful insights that could help further improve the model

**5. Contributions**

My contributions in this project included:

* Designing and debugging the code
* Training the model to identify allergens
* Create data visualizations
* Test the accuracy of the model
* Contemplate how this model could improve and potentially help people with allergic reactions

**6. About the dataset (Details, Visuals, etc.)**

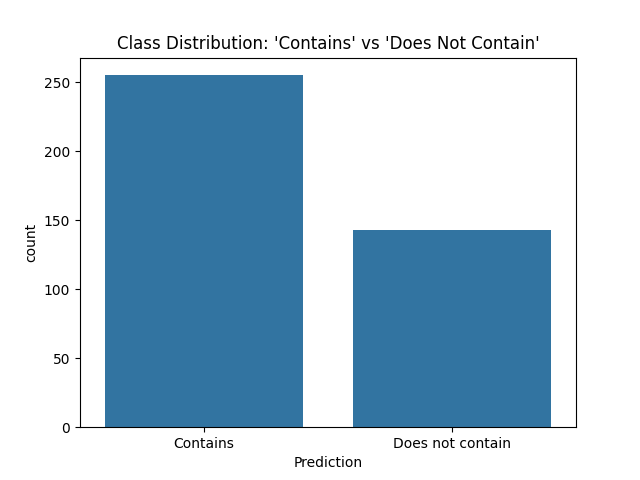
The [dataset](https://www.kaggle.com/datasets/uom190346a/food-ingredients-and-allergens/data) was obtained from Kaggle. It is a collection of information regarding present allergens in various food items. The dataset has up to 400 records, with each representing a specific food item and its associated allergen (if any).

Dataset Columns:

* Food Product: The name of the food item
* Main Ingredient: The primary ingredient of the food item
* Sweetener: The type or presence of a sweetener used in the food item
* Fat/Oil: The type or presence of fat or oil used in the food item
* Seasoning: The spices or seasonings added for flavor enhancement
* Allergens: The allergen(s) associated with the food item
* Prediction: Label for the food item based on its ingredients and allergens

Visualizations:

Figure 1: Shows the distribution of the Prediction column



As seen above, there is class imbalance present. This issue is handled later in the code via synthetic samples.

Another visualization is a **correlation heatmap of encoded features** to show the correlation coefficients between all pairs of features in the dataset.

**7. Machine Learning Model Chosen and Justification for the Choice**

The problem is a classification task, as the goal is to classify whether a food item contains allergens or not. Among the models tested, **Logistic Regression** performed the best and was chosen for the final implementation.

**Justification for choice:** Logistic Regression was chosen primarily because it gave the highest accuracy and f1-score out of all the models tested. This showed its strong ability to make accurate predictions. Though other models may provide different benefits, the superior f1-score and accuracy puts Logistic Regression the most suitable choice for this project.

**8. Alternative Models That Could Be Used and Why They Weren’t**

In addition to Logistic Regression, the following models were also tested and evaluated:

* Random Forest: A versatile ensemble learning method used for both classification and regression tasks.
* Naïve Bayes (MultinomialNB/GaussianNB): A family of simple yet effective classification algorithms based on Bayes’ Theorem. They are known for being fast and efficient.
* Support Vector Machine (SVM) Classifier: A powerful and versatile tool for both classification and regression, well-suited for finding complex decision boundaries.

**9. Evaluation Techniques and how I used them**

The project utilized a comprehensive set of evaluation techniques, such as:

* Classification report (from sklearn.metrics): This key metric provides a summary of the precision, recall, F1-score, and support for each class (‘Contains’ and ‘Does not contain’)
* Confusion Matrix (from sklearn.metrics): A table that summarizes the performance of a classification model. It shows the number of correct and incorrect predictions made by the model, broken down by each class.
* Accuracy (from sklearn.metrics): Measures the proportion of correctly classified instances out of the total instances. It's the most intuitive metric but can be misleading for imbalanced datasets, as a model might achieve high accuracy by simply predicting the majority class all the time.
* F1-score (from sklearn.metrics): This is the harmonic mean of precision and recall. In my classification, I specified for it to get the F1-score for the ‘Contains’ class.

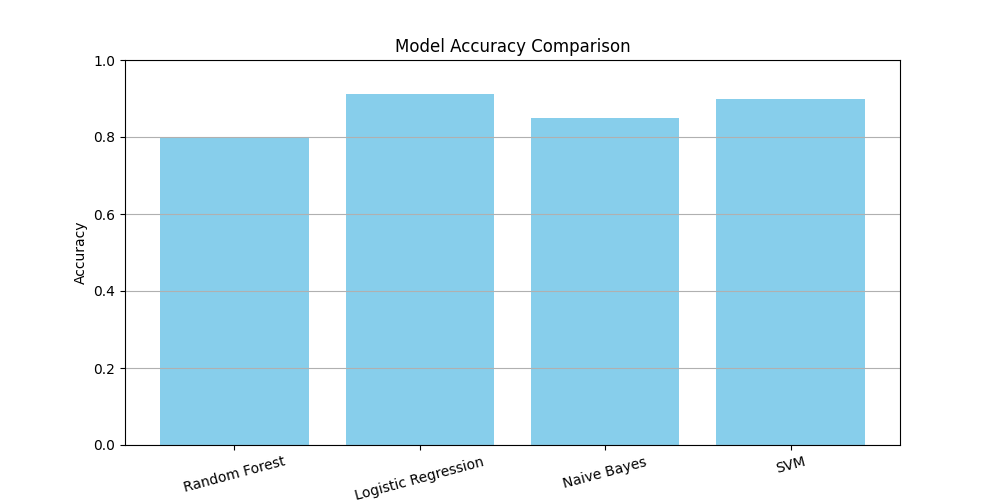
**10. Any Hyperparameter Tuning Performed, and How**

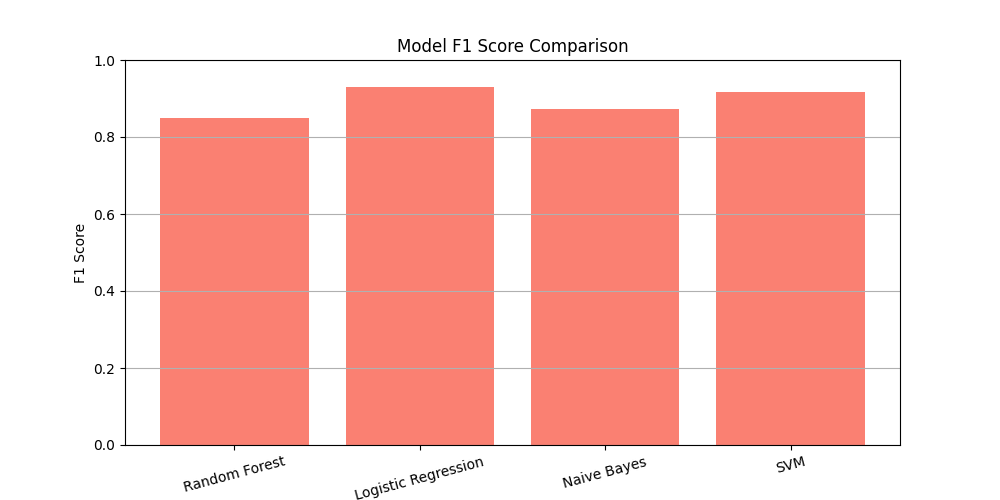
In my code, I instantiated each model with a specific parameter:

* RandomForestClassifier(random\_state=42, class\_weight=’balanced’)
* LogisticRegression(max\_iter=1000, class\_weight=’balanced’)
* GaussianNB() (default parameters)
* SVC(kernel=’linear’,probability=True, class\_weight=’balanced’)

**11. Accuracy/Performance of the model**

Testing the four models for their accuracy and F1-scores gave me the following graphs:





The results are as follows:

* Random Forest:
  + Accuracy: 0.8
  + F1 Score: 0.8490566037735849
* Logistic Regression:
  + Accuracy: 0.9125
  + F1 Score: 0.9306930693069307
* Naive Bayes:
  + Accuracy: 0.85
  + F1 Score: 0.8723404255319149
* SVM:
  + Accuracy: 0.9
  + F1 Score: 0.9166666666666666

The results clearly show that Logistic Regression is the best performing model out of all four, due to it having the highest accuracy and f1-score.

**12. How You Handled Underfitting and Overfitting**

As shown in the graph in part 6, the dataset was imbalanced and could have led to overfitting if fed to the models. This was assessed by:

* Using a train-test-split: The dataset was split into separate training (80%) and testing (20%) sets. The model was only trained on the training set then had its performance evaluated on the test set.
* Implementing SMOTE (Synthetic Minority Over-sampling Technique): This helps create synthetic samples for the minority class, which provides the model with a more diverse training dataset.

**13. Key Leanings from the Project**

Several key learnings were obtained from this project:

* Importance of data understanding and preprocessing: Thorough data inspection and proper data preparation steps are crucial in making a machine learning model.
* Addressing Class Imbalance Is Paramount: The challenge that was overfitting due to class imbalance was a major learning point. I discovered that simply having a model doesn’t guarantee good performance, especially for critical minority classes.
* Choosing the right evaluation metric: This project helped me understand that metrics like precision, recall, and f1-score are more appropriate for the minority class and that accuracy can be misleading for imbalanced datasets.
* Machine Learning is an iterative process: The journey from data loading to identifying overfitting, to implementing solutions and refining the evaluation process demonstrated the iterative nature of machine learning projects.

Overall, this project provided a practical lesson in identifying and mitigating common challenges in classification problems, particularly those involving imbalanced datasets, which are very common in real world scenarios.

**14. Potential Usefulness of your Project to Society or Industry**

This project has quite the potential usefulness to society. Its capability to identify allergens can help people who have allergies avoid the foods that activate them. This can greatly save potential spending on treatment or trips to the hospital.

**15. Any Intention to extend this project**

Regarding any possible extension of this project, I do not intend to pursue a research paper on this topic yet. While I would like to try publishing one day, it is not my immediate goal for this project. However, this project does have potential for future development into a robust application, perhaps integrated with image recognition to better identify foods and allergens.

**16. Conclusion based on your Findings**

This successfully developed and evaluated a Logistic Regression model to identify and tell whether a certain food item contains allergens or not. The analysis confirmed that the influencing factors were the main ingredients, sweetener, fat/oil, and seasoning. The model obtained an accuracy of 0.9125 and an f1-score of 0.9306930693069307, demonstrating a very accurate performance. The model provides a valuable framework for understanding the factors that determine an allergen’s presence in food items, giving insight for machine learning model builders to further advance the technology.

**Code Implementation**

Here is the implementation of the code:

<https://github.com/ItsVathanak/MLProjectDetails/blob/main/data/allergen_predictor.py>