Solved_Food_Allergies

October 28, 2024

1 Solving My Dog's Food Allergies

In this project, I am working to resolve the food allergies of Phoebe, my 8-year-old Airedale Terrier. She has been struggling with allergies for the past year, but I finally found a brand with ingredients that suit her. I can relate to her experience, as I grew up with food allergies myself, being allergic to coconut oil and a variety of fruits. Fortunately, my allergens consisted of single ingredients, unlike the more complex products that will be discussed in this documentation.

Diagnosing food allergies in dogs is more challenging due to the long and complex ingredient lists in pet foods. To simplify this process, we will leverage *Python's* `set` function for efficient comparison.

1.0.1 Project Overview:

- Using Prompt Engineering Methods: To generate a dictionary of dog foods and their ingredients.
- Comparing Problematic and Safe Foods: Applying the `difference` method to identify allergens in foods that triggered reactions.
- Extracting Relevant Allergens: Using various Python techniques to isolate key allergens.
- Creating a Scanning Function: Building a function that scans a list of allergens against ingredient lists to flag potential triggers.

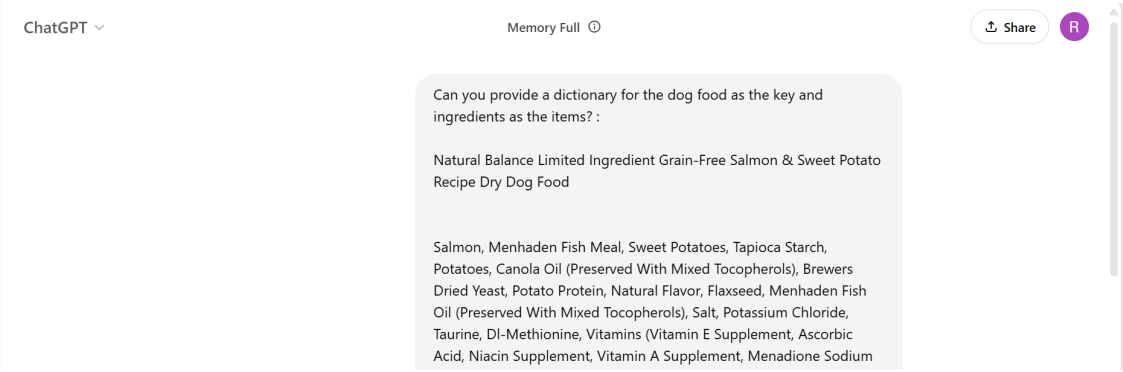
1.1 Prompting Via OpenAI

My account is limited in length outputs I can print, but for those with access to extended tokens feel free to use this code below.

Note: I will be using a simpler and more cost-effective approach, which is to ask ChatGPT on the website, `chatgpt.com`, to generate the dog food ingredients in dictionary format.

[ ]:
```
import openai

#####  Replace with your actual API key
openai.api_key = "your_openai_api_key"

def ask_openai(prompt, model="gpt-4", max_tokens=100):
    """
    Function to send a prompt to OpenAI API and return the response.

    Args:
    - prompt (str): The question or input for the model.
```

```
    - model (str): The OpenAI model to use ('gpt-3.5-turbo' or 'gpt-4').
    - max_tokens (int): The maximum number of tokens in the response.

    Returns:
    - str: The response from the model.
    """
    response = openai.ChatCompletion.create(
        model=model,
        messages=[{"role": "user", "content": prompt}],
        max_tokens=max_tokens,
        temperature=0.0  # Adjust creativity level (0.0 = focused, 1.0 = very creative)
    )
    return response.choices[0].message["content"]

#####  Prompting

prompt = "prompt"
response = ask_openai(prompt)
print(response)
```



1.1.1 Chat's output:

[180]:
```
import pandas as pd
dog_food_ingredients = {
    "Natural Balance Limited Ingredient Grain-Free Salmon & Sweet Potato Recipe Dry Dog Food": [
        "Salmon", "Menhaden Fish Meal", "Sweet Potatoes", "Tapioca Starch", "Potatoes",
        "Canola Oil (Preserved With Mixed Tocopherols)", "Brewers Dried Yeast",
        "Potato Protein", "Natural Flavor", "Flaxseed",
        "Menhaden Fish Oil (Preserved With Mixed Tocopherols)", "Salt", "Potassium Chloride",
        "Taurine", "Dl-Methionine",
```

```
      "Vitamins (Vitamin E Supplement, Ascorbic Acid, Niacin Supplement,␣
↪Vitamin A Supplement, Menadione Sodium Bisulfite Complex, Thiamine␣
↪Mononitrate, D-Calcium Pantothenate, Riboflavin Supplement, Pyridoxine␣
↪Hydrochloride, Vitamin B12 Supplement, Folic Acid, Biotin)",
      "Choline Chloride",
      "Minerals (Zinc Proteinate, Zinc Sulfate, Ferrous Sulfate, Iron␣
↪Proteinate, Copper Sulfate, Copper Proteinate, Sodium Selenite, Manganese␣
↪Sulfate, Manganese Proteinate, Calcium Iodate)",
      "Dicalcium Phosphate", "Citric Acid (Preservative)",
      "Mixed Tocopherols (Preservative)", "Rosemary Extract"
  ],
  "CANIDAE Grain-Free PURE Limited Ingredient Salmon & Sweet Potato Recipe␣
↪Dry Dog Food": [
      "Salmon", "Salmon Meal", "Menhaden Fish Meal", "Lentils", "Sweet␣
↪Potatoes",
      "Garbanzo Beans", "Peas", "Canola Oil", "Potatoes", "Flaxseed",
      "Natural Flavor", "Choline Chloride", "Taurine", "Salt",
      "Mixed Tocopherols (A Preservative)", "Zinc Sulfate", "Potassium␣
↪Chloride",
      "Vitamin E Supplement", "Ferrous Sulfate", "Niacin",
      "L-Ascorbyl-2-Polyphosphate", "Calcium Pantothenate", "Manganese␣
↪Sulfate",
      "Copper Sulfate", "Vitamin A Supplement", "Sodium Selenite",
      "Riboflavin Supplement", "Thiamine Mononitrate",
      "Dried Lactobacillus Acidophilus Fermentation Product", "Biotin",
      "Vitamin B12 Supplement", "Zinc Proteinate",
      "Pyridoxine Hydrochloride (Vitamin B6 Supplement)",
      "Ethylenediamine Dihydroiodide", "Vitamin D3 Supplement", "Folic Acid",
      "Dried Lactobacillus Casei Fermentation Product",
      "Dried Lactobacillus Plantarum Fermentation Product"
  ],
  "Blue Buffalo Life Protection Formula Adult Fish & Brown Rice Recipe Dry␣
↪Dog Food": [
      "Whitefish", "Fish Meal (source of Omega 3 Fatty Acids)", "Brown Rice",␣
↪"Barley",
      "Oatmeal", "Peas", "Pea Starch",
      "Chicken Fat (preserved with Mixed Tocopherols)", "Flaxseed (source of␣
↪Omega 6 Fatty Acids)",
      "Natural Flavor", "Dried Tomato Pomace", "Dried Yeast", "Chicken Meal",
      "Calcium Carbonate", "Direct Dehydrated Alfalfa Pellets", "Salt",␣
↪"Potassium Chloride",
      "Dried Chicory Root", "Alfalfa Nutrient Concentrate", "Choline␣
↪Chloride",
      "Dicalcium Phosphate", "DL-Methionine", "L-Threonine",
      "Vitamin E Supplement (preserved with Mixed Tocopherols)", "Dried Sweet␣
↪Potatoes",
```

```
        "Carrots", "Glucosamine Hydrochloride", "Zinc Amino Acid Chelate",␣
 ↪"Zinc Sulfate",
        "Vegetable Juice for color", "Ferrous Sulfate", "Iron Amino Acid␣
 ↪Chelate",
        "Blueberries", "Cranberries", "Barley Grass", "Parsley", "Turmeric",
        "Dried Kelp", "Yucca Schidigera Extract", "Niacin (Vitamin B3)",
        "Calcium Pantothenate (Vitamin B5)", "Copper Sulfate",
        "L-Ascorbyl-2-Polyphosphate (source of Vitamin C)", "L-Lysine",
        "Biotin (Vitamin B7)", "Vitamin A Supplement", "Copper Amino Acid␣
 ↪Chelate",
        "Manganese Sulfate", "Taurine", "Manganese Amino Acid Chelate",
        "Thiamine Mononitrate (Vitamin B1)", "Riboflavin (Vitamin B2)",
        "Vitamin D3 Supplement", "Vitamin B12 Supplement",
        "Pyridoxine Hydrochloride (Vitamin B6)", "Calcium Iodate",
        "Folic Acid (Vitamin B9)", "Sodium Selenite", "Oil of Rosemary"
    ]
}
```

[182]:
```
#Converting list of ingredients into sets
natural_balance = set(dog_food_ingredients["Natural Balance Limited Ingredient␣
 ↪Grain-Free Salmon & Sweet Potato Recipe Dry Dog Food"]) #good food

candidae = set(dog_food_ingredients["CANIDAE Grain-Free PURE Limited Ingredient␣
 ↪Salmon & Sweet Potato Recipe Dry Dog Food"]) #problemic food


blue_buffalo = set(dog_food_ingredients["Blue Buffalo Life Protection Formula␣
 ↪Adult Fish & Brown Rice Recipe Dry Dog Food"]) #problemic food

#Length of Ingredients
print(f'# of Natural Balance Dog Ingredients: {len(natural_balance)}')

print(f'\n # of Candidae Dog Ingredients: {len(candidae)}')

print(f'\n # of Blue Buffalo Dog Ingredients: {len(blue_buffalo)}')
```

```
# of Natural Balance Dog Ingredients: 22

 # of Candidae Dog Ingredients: 38

 # of Blue Buffalo Dog Ingredients: 59
```

2 Comparing Natural Balance with Both Allergen Dog Foods

Now that I've converted the ingredient lists into sets, I will use the `union` method to combine the two dog foods that caused reactions (Canidae and Blue Buffalo). Below, you'll see how to use both the `difference` method and a loop to identify the ingredients in allergy_food that are not present

in natural_balance.

[184]:
```
# Union of the allergen ingredients

allergy_food = candidae.union(blue_buffalo)

#Now using difference method, to return a set of elements that are in␣
 ↪allergy_food but not in natural_balance.

allergens = allergy_food.difference(natural_balance)

print(f'Length of Allergen list: {len(allergens)}')
print(f'List of Allergen: {allergens}')
```

```
Length of Allergen list: 72
List of Allergen: {'Biotin', 'Chicken Fat (preserved with Mixed Tocopherols)',
'Vegetable Juice for color', 'Flaxseed (source of Omega 6 Fatty Acids)',
'Glucosamine Hydrochloride', 'Ferrous Sulfate', 'Calcium Pantothenate (Vitamin
B5)', 'Niacin (Vitamin B3)', 'Salmon Meal', 'Cranberries', 'Manganese Amino Acid
Chelate', 'Brown Rice', 'Manganese Sulfate', 'Pyridoxine Hydrochloride (Vitamin
B6)', 'L-Ascorbyl-2-Polyphosphate', 'Thiamine Mononitrate (Vitamin B1)', 'Direct
Dehydrated Alfalfa Pellets', 'Dried Lactobacillus Acidophilus Fermentation
Product', 'Barley', 'Oatmeal', 'Zinc Proteinate', 'Canola Oil', 'Ethylenediamine
Dihydroiodide', 'Garbanzo Beans', 'Barley Grass', 'Chicken Meal', 'Folic Acid',
'Alfalfa Nutrient Concentrate', 'Riboflavin Supplement', 'Vitamin B12
Supplement', 'Vitamin A Supplement', 'Biotin (Vitamin B7)', 'Riboflavin (Vitamin
B2)', 'Mixed Tocopherols (A Preservative)', 'Sodium Selenite', 'L-Lysine',
'Carrots', 'Whitefish', 'Folic Acid (Vitamin B9)', 'Oil of Rosemary', 'Dried
Kelp', 'Lentils', 'Copper Amino Acid Chelate', 'L-Ascorbyl-2-Polyphosphate
(source of Vitamin C)', 'Zinc Sulfate', 'Dried Tomato Pomace', 'Dried Sweet
Potatoes', 'Niacin', 'Turmeric', 'Yucca Schidigera Extract', 'Zinc Amino Acid
Chelate', 'Dried Yeast', 'Peas', 'Pyridoxine Hydrochloride (Vitamin B6
Supplement)', 'Iron Amino Acid Chelate', 'Vitamin E Supplement', 'Calcium
Pantothenate', 'Dried Lactobacillus Plantarum Fermentation Product', 'Vitamin D3
Supplement', 'Pea Starch', 'Blueberries', 'Calcium Iodate', 'Fish Meal (source
of Omega 3 Fatty Acids)', 'Copper Sulfate', 'Thiamine Mononitrate', 'Dried
Chicory Root', 'DL-Methionine', 'L-Threonine', 'Parsley', 'Calcium Carbonate',
'Vitamin E Supplement (preserved with Mixed Tocopherols)', 'Dried Lactobacillus
Casei Fermentation Product'}
```

[186]:
```
#I noticed there are similar ingredients so I will use a loop to further filter

# Create a list to hold allergens that have similar starting names
similar_allergens = set()

# Iterate over the ingredients in allergy_ingredients
for allergen in allergens:
    # Check if any ingredient in natural_balance starts with the allergen
```

```
    for food in natural_balance:
        if food.startswith(allergen):
            similar_allergens.add(allergen)

# Print the similar allergens found
print(similar_allergens)
```

```
{'Canola Oil'}
```

3 Comparing Only Candidae

[190]:
```
#Comparing Just Canidae and Natural Balance to Shorten the List of 71 allergens

candidae_allergens = candidae.difference(natural_balance)

#Removing Canola Oil since we know it is also in Natural Balance
candidae_allergens.remove('Canola Oil')

print(f'Length of Allergen list: {len(candidae_allergens)}')

for allergy in candidae_allergens:
    print(f"\n{allergy}") #Now only 27 allergens
```

```
Length of Allergen list: 27

Biotin

Ferrous Sulfate

Salmon Meal

Manganese Sulfate

L-Ascorbyl-2-Polyphosphate

Dried Lactobacillus Acidophilus Fermentation Product

Zinc Proteinate

Ethylenediamine Dihydroiodide

Garbanzo Beans

Folic Acid

Riboflavin Supplement
```

```
Vitamin B12 Supplement

Vitamin A Supplement

Mixed Tocopherols (A Preservative)

Sodium Selenite

Lentils

Zinc Sulfate

Niacin

Pyridoxine Hydrochloride (Vitamin B6 Supplement)

Vitamin E Supplement

Peas

Calcium Pantothenate

Dried Lactobacillus Plantarum Fermentation Product

Vitamin D3 Supplement

Copper Sulfate

Thiamine Mononitrate

Dried Lactobacillus Casei Fermentation Product
```

[198]:
```
# Further filtering candidae_allergens list
#Vitamin and Minearals are also Natural Balance but not individualized
vitamin_mineral = [
    "Vitamin E Supplement", "Ascorbic Acid", "Niacin Supplement", "Vitamin A␣
 ↪Supplement",
    "Menadione Sodium Bisulfite Complex", "Thiamine Mononitrate", "D-Calcium␣
 ↪Pantothenate",
    "Riboflavin Supplement", "Pyridoxine Hydrochloride", "Vitamin B12␣
 ↪Supplement",
    "Folic Acid", "Biotin", "Choline Chloride", "Zinc Proteinate", "Zinc␣
 ↪Sulfate",
    "Ferrous Sulfate", "Iron Proteinate", "Copper Sulfate", "Copper Proteinate",
    "Sodium Selenite", "Manganese Sulfate", "Manganese Proteinate", "Calcium␣
 ↪Iodate"
] #Itemized Vitamin and Mineral ingredient in Natural_Balance
```

```
# Set to store similar allergens
similar_ingredients = set()

for allergen in candidae_allergens:
    # Check if any ingredient in vitamin_mineral starts with the allergen
    for food in vitamin_mineral:
        if allergen.startswith(food):
            similar_ingredients.add(allergen)

# Print similar allergens found
print(f"Length of Similar Ingredients: {len(similar_ingredients)}")
print(f"\nSimilar Allergens: {similar_ingredients}")
```

```
Length of Similar Ingredients: 14

Similar Allergens: {'Biotin', 'Zinc Proteinate', 'Ferrous Sulfate', 'Copper
Sulfate', 'Thiamine Mononitrate', 'Folic Acid', 'Riboflavin Supplement',
'Vitamin B12 Supplement', 'Manganese Sulfate', 'Zinc Sulfate', 'Pyridoxine
Hydrochloride (Vitamin B6 Supplement)', 'Vitamin A Supplement', 'Vitamin E
Supplement', 'Sodium Selenite'}
```

Further Filtering They were 27 allergens present in the candidae_allergens set before similar allergens were filtered. Now there are

[212]:
```
filtered_allergens = candidae_allergens.difference(similar_ingredients)

print(f'Length of Filtered Allergen list: {len(filtered_allergens)}')

for allergen in filtered_allergens:
    print(f'\n{allergen}')  # List the remaining allergens
```

```
Length of Filtered Allergen list: 13

Vitamin D3 Supplement

Ethylenediamine Dihydroiodide

Lentils

Garbanzo Beans

Salmon Meal

Niacin

L-Ascorbyl-2-Polyphosphate
```

```
Dried Lactobacillus Plantarum Fermentation Product

Peas

Dried Lactobacillus Acidophilus Fermentation Product

Mixed Tocopherols (A Preservative)

Calcium Pantothenate

Dried Lactobacillus Casei Fermentation Product
```

[214]:
```python
#Still have quite a long list: will search common dog allergies to filter further
legumes = [
    "Lentils",
    "Chickpeas",
    "Black Beans",
    "Pinto Beans",
    "Kidney Beans",
    "Navy Beans",
    "Great Northern Beans",
    "Garbanzo Beans",
    "Adzuki Beans",
    "Mung Beans",
    "Peas",
    "Soybeans",
    "Green Beans",
    "Fava Beans",
    "Split Peas",
    "Pigeon Peas",
    "Borlotti Beans"
]

count = 0

for legume in legumes:
    if legume in filtered_allergens:
        count += 1
        print(legume)
print(f'Number of Legumes in Allergen List: {count}')
```

```
Lentils
Garbanzo Beans
Peas
Number of Legumes in Allergen List: 3
```

[216]:
```
# Initialize a counter for fermentation products found
fermentation_count = 0

# Loop through each allergen in the filtered_allergens list
for allergen in filtered_allergens:
    # Check if "Fermentation Product" is in the allergen
    if "Fermentation Product" in allergen:
        fermentation_count += 1  # Increment the counter if found
        print(allergen)

print(f'\nNumber of Fermentation Products in Allergen List:␣
 ↪{fermentation_count}')
```

```
Dried Lactobacillus Plantarum Fermentation Product
Dried Lactobacillus Acidophilus Fermentation Product
Dried Lactobacillus Casei Fermentation Product

Number of Fermentation Products in Allergen List: 3
```

3.0.1 Does the Blue Buffalo Also Have These Ingredients? How many?

[218]:
```
count = 0

for legume in legumes:
    if legume in blue_buffalo:
        count += 1
        print(legume)
print(f'\nNumber of Legumes in Allergen List: {count}')


# Initialize a counter for fermentation products found
fermentation_count = 0

# Loop through each allergen in the filtered_allergens list
for allergen in blue_buffalo:
    # Check if "Fermentation Product" is in the allergen
    if "Fermentation Product" in allergen:
        fermentation_count += 1  # Increment the counter if found

# Print the total count of fermentation products found
print(f'Number of Fermentation Products in Allergen List: {fermentation_count}')
```

```
Peas

Number of Legumes in Allergen List: 1
Number of Fermentation Products in Allergen List: 0
```

Peas are mutually present in both dog foods. However, are there are other allergens to consider in

the Blue Buffalo ingredients.

[222]:
```
#Filtering Similar Ingredients from Natural Balance
#like I did with the did with the Candidae ingredients
blue_buffalo_allergens = blue_buffalo.difference(natural_balance)

similar_ingredients = [
    "Canola Oil", "Vitamin E Supplement", "Ascorbic Acid", "Niacin Supplement",␣
 ↪"Vitamin A Supplement",
    "Menadione Sodium Bisulfite Complex", "Thiamine Mononitrate", "D-Calcium␣
 ↪Pantothenate",
    "Riboflavin Supplement", "Pyridoxine Hydrochloride", "Vitamin B12␣
 ↪Supplement",
    "Folic Acid", "Biotin", "Choline Chloride", "Zinc Proteinate", "Zinc␣
 ↪Sulfate",
    "Ferrous Sulfate", "Iron Proteinate", "Copper Sulfate", "Copper Proteinate",
    "Sodium Selenite", "Manganese Sulfate", "Manganese Proteinate", "Calcium␣
 ↪Iodate"
]

allergens = set()
for allergen in blue_buffalo_allergens:
    if allergen not in similar_ingredients:
        allergens.add(allergen)

print(f'\n Length of List: {len(allergens)}')

for allergen in allergens:
    print(f'\n{allergen}')
```

```
 Length of List: 45

Chicken Fat (preserved with Mixed Tocopherols)

Vegetable Juice for color

Flaxseed (source of Omega 6 Fatty Acids)

Glucosamine Hydrochloride

Calcium Pantothenate (Vitamin B5)

Niacin (Vitamin B3)

Cranberries
```

Brown Rice

Manganese Amino Acid Chelate

Pyridoxine Hydrochloride (Vitamin B6)

Thiamine Mononitrate (Vitamin B1)

Direct Dehydrated Alfalfa Pellets

Barley

Oatmeal

Barley Grass

Chicken Meal

Alfalfa Nutrient Concentrate

Biotin (Vitamin B7)

Riboflavin (Vitamin B2)

L-Lysine

Carrots

Whitefish

Folic Acid (Vitamin B9)

Oil of Rosemary

Dried Kelp

Copper Amino Acid Chelate

L-Ascorbyl-2-Polyphosphate (source of Vitamin C)

Dried Tomato Pomace

Zinc Amino Acid Chelate

Dried Sweet Potatoes

Iron Amino Acid Chelate

Turmeric

Yucca Schidigera Extract

Dried Yeast

Peas

Vitamin D3 Supplement

Pea Starch

Blueberries

Fish Meal (source of Omega 3 Fatty Acids)

Dried Chicory Root

DL-Methionine

L-Threonine

Parsley

Calcium Carbonate

Vitamin E Supplement (preserved with Mixed Tocopherols)

[230]:
```
#Now extracting possible allergens from Blue Buffalo Ingredients
grains = [
    "Wheat", "Oatmeal", "Barley", "Rye", "Corn", "Rice",
    "Quinoa", "Millet", "Sorghum", "Bulgur", "Spelt",
    "Farro", "Amaranth", "Teff", "Buckwheat", "Couscous"
]

legumes = [
    "Lentils", "Chickpeas", "Black Beans", "Pinto Beans", "Kidney Beans",
    "Navy Beans", "Great Northern Beans", "Garbanzo Beans", "Adzuki Beans",
    "Mung Beans", "Peas", "Soybeans", "Green Beans", "Fava Beans",
    "Split Peas", "Pigeon Peas", "Borlotti Beans"
]

filter_allergens = set()

# Iterate over Blue Buffalo allergens
for allergen in allergens:
    # Check if allergen contains any grain
```

```
    for grain in grains:
        if grain in allergen:
            filter_allergens.add(allergen)
            break
    # Check if allergen contains any legume
    for legume in legumes:
        if legume in allergen:
            filter_allergens.add(allergen)
            break
    # Special check for 'Ferment Product'
    if 'Ferment Product' in allergen:
        filter_allergens.add(allergen)

# Display the filtered allergens
print(filter_allergens)

update_allergen = allergens.difference(filter_allergens)
len(update_allergen)
```

```
{'Barley Grass', 'Brown Rice', 'Peas', 'Barley', 'Oatmeal'}
```

[230]: 40

The *Blue Buffalo* food contains a lot of grains, *Peas*, and no fermentation products. However, this dog food is completely different from the Natural Balance's, 40 different compared to 14 for *Candidae*, this so may not be not be great reference.

3.1 Function: Compare Future Foods to Allergen List

[149]:
```
def scan_allergens(allergen_list, ingredient_lists):
    """
    Scans ingredient lists for potential allergens.

    Args:
    - allergen_list (list): A list of allergens to check against.
    - ingredient_lists (list of list): A list of ingredient lists, each␣
 ↪representing a product.

    Returns:
    - set: A set of flagged ingredients that match any of the allergens.
    """
    flagged_ingredients = set()  # Set to store unique flagged ingredients

    # Iterate over each ingredient list (product)
    for ingredients in ingredient_lists:
        # Check each allergen against the ingredients
        for allergen in allergen_list:
```

```
            if allergen in ingredients:
                flagged_ingredients.add(allergen)  # Add allergen to flagged␣
 ↪set if found

    return flagged_ingredients
```

[235]:
```
#legumes, grains, fermentation product
possible_allergens = ['Peas', 'Garbanzo Beans', 'Lentils', 'Barley Grass',␣
 ↪'Brown Rice', 'Peas', 'Barley', 'Oatmeal','Fermentation Product']

#Dog Food example: https://www.chewy.com/victor-select-lamb-meal-sweet-potato/
 ↪dp/129302

victor = [
    "Lamb Meal", "Peas", "Sweet Potato", "Chicken Fat (preserved with mixed␣
 ↪Tocopherols)",
    "Garbanzo Beans", "Blood Meal Conventionally Dried", "Yeast Culture",
    "Menhaden Fish Meal (source of DHA-Docosahexaenoic Acid)", "Dehydrated␣
 ↪Alfalfa Meal",
    "Natural Flavor", "Potassium Chloride", "Carrot Powder", "Tomato Pomace",
    "Salt", "Organic Dried Seaweed Meal", "Taurine", "Choline Chloride",
    "Calcium Stearate", "Zinc Methionine Complex", "Vitamin E Supplement",
    "DL-Methionine", "Iron Amino Acid Complex", "Hydrolyzed Yeast",
    "Manganese Amino Acid Complex", "Silicon Dioxide", "L-Carnitine",
    "Selenium Yeast", "Brewers Dried Yeast", "Dried Enterococcus Faecium␣
 ↪Fermentation Product",
    "Dried Lactobacillus Acidophilus Fermentation Product", "Dried Aspergillus␣
 ↪Oryzae Fermentation Extract",
    "Dried Trichoderma Longibrachiatum Fermentation Extract", "Dried Bacillus␣
 ↪Subtilis Fermentation Extract",
    "Copper Sulfate", "Niacin Supplement", "Vitamin B12 Supplement", "Vitamin A␣
 ↪Supplement",
    "D-Calcium Pantothenate", "Thiamine Mononitrate", "Biotin", "Yucca␣
 ↪Schidigera Extract",
    "Calcium Carbonate", "Riboflavin Supplement", "Calcium Iodate", "Pyridoxine␣
 ↪Hydrochloride",
    "Vitamin D3 Supplement", "Tetrasodium Pyrophosphate", "Vegetable Oil",␣
 ↪"Rosemary Extract",
    "Green Tea Extract", "Spearmint Extract", "Inulin", "Lecithin", "Folic Acid"
]

#Using Function
scan_allergens(possible_allergens, victor)
```

[235]:
```
{'Fermentation Product', 'Garbanzo Beans', 'Peas'}
```

3.1.1 Summary

Despite all the Dog Foods claiming fish as on the main ingredient, they vary in ingredients.

[158]:
```
#Length of Ingredients
print(f'Number of Natural Balance Dog Ingredients: {len(natural_balance)}')

print(f'\nNumber of Candidae Dog Ingredients: {len(candidae)}')

print(f'\nNumber of Blue Buffalo Dog Ingredients: {len(blue_buffalo)}')
```

```
Number of Natural Balance Dog Ingredients: 22

Number of Candidae Dog Ingredients: 38

Number of Blue Buffalo Dog Ingredients: 59
```

The common allergen between the two Dog Foods were Peas. I can confidently say that this is a possible allergy and it may be best to avoid the group of legumes altogether. However, grains and fermentation products were not mutually present in both Candidae and Blue Buffalo so this may require more dog food example to confirm.

4 Conclusion

I have successfully identified a food that worked best for my dog, `Natural Balance`. Through a thorough process of elimination, I narrowed down the specific allergens that caused her reactions. It appears that *Legumes and Fermented Products* were the most prevalent in the `Candidae's` ingredients. While Natural Balance contains *Brewers Dried Yeast*, which also plays a role in fermentation, it's important to note that this is a yeast whereas Fermentation Products involve probiotic bacteria. This investigation has significantly improved my understanding of my dog's allergies, although there is always potential for further exploration.

A common allergen identified in `Blue Buffalo` was the *Pea* ingredient, but the role of Grain and Fermented Products requires additional confirmation.

Moreover, I encountered challenges related to similar ingredients presented in various formats. Variations in ingredient names and labeling can complicate allergen identification, highlighting a common issue in food labeling where the same ingredient may be listed in different forms.

Ultimately, this journey has not only deepened my understanding of my dog's dietary needs but also emphasized the importance of vigilance in monitoring her health. I look forward to continuing this project, refining my approach, and ensuring my dog thrives on a diet that best supports her well-being.