Capstone Project With WISEcode

Adarsh Chandra Sekhar, Sanjana Deswal, Shikha Soman, Shivani Narahari, Vennam Prahasith

Project Objective

- The primary purpose of our project is the categorization of various foods into broad categories.
- We started of with a goal of making a machine learning algorithm which would determine whether certain dietary data from a given food could properly identify the meal from a wide variety of categories. E.g. – cereals, ice cream, breads etc.
- We intended to have 3 machine Learning algorithms for the scope of this project, namely
- Support Vector Machine
- Random Forest
- Naïve Bayes

Tools Used

- Google Colab Pro
- Anaconda
- Jupyter Notebook
- Python

Project Milestones

Data gathering, cleaning and splitting.

Implementation of the three ML algorithms.

Adding additional features such as weights.

Testing the algorithms.

Completing necessary documentation.

Major Challenges

- Initially we were struggling to work with the data set due to their size.
- Our initial plan was to do the project on google colab so we could all simultaneously collaborate and work together, but we ran into ram issues due to the size of the data even after purchasing colab pro.
- Implementation of SVM was not successful. Naïve Beyes was discarded due to its low performance

Ingredient_table

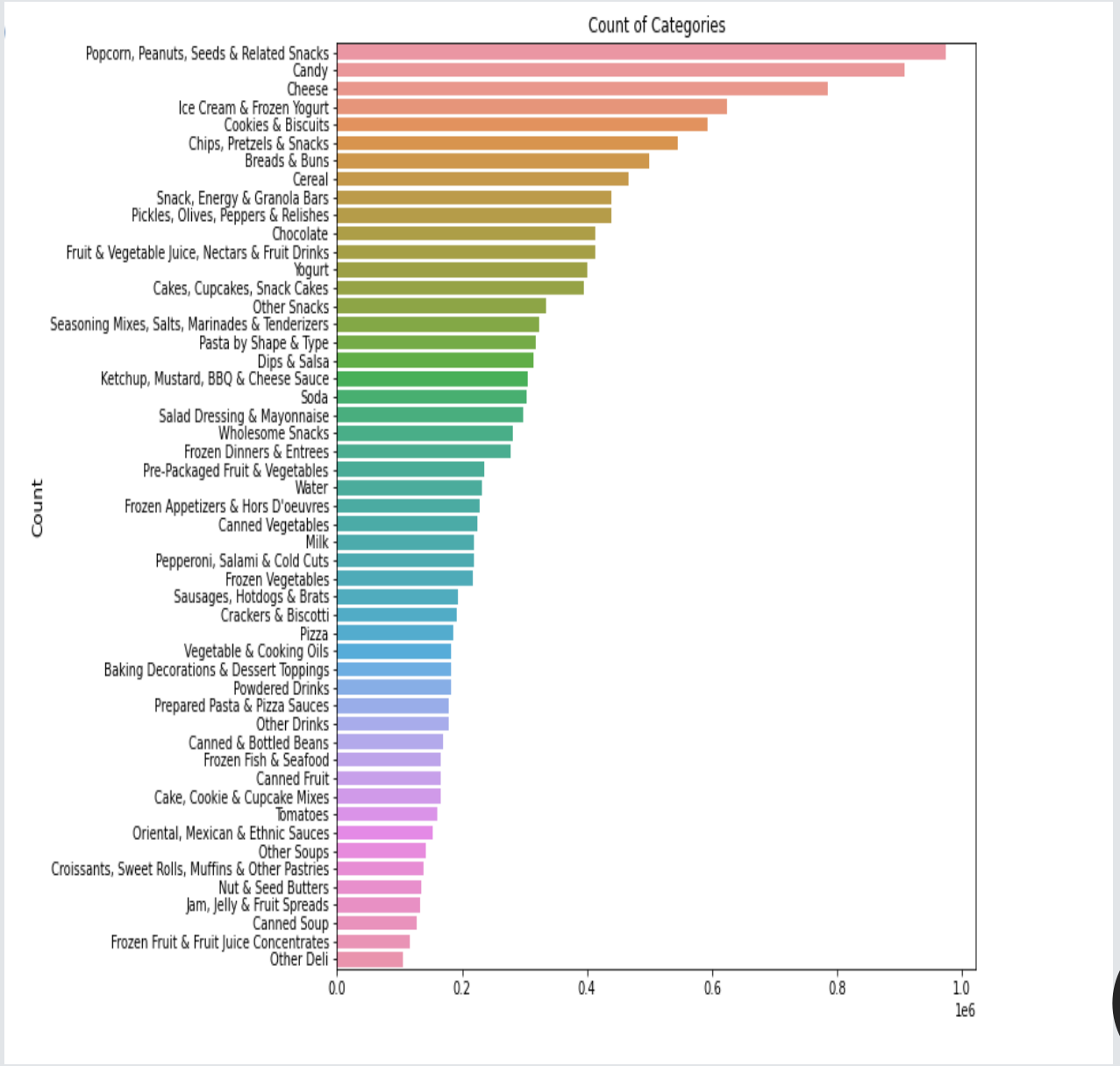
	fdc_id	brand_owner	brand_name	serving_size	serving_size_unit	branded_food_category	ingredient_list	id	nutrient_id	amount
0	1105904	Richardson Oilseed Products (US) Limited	NaN	15	ml	Oils Edible	VEGETABLE OIL	NaN	NaN	NaN
1	1105905	CAMPBELL SOUP COMPANY	NaN	240	ml	Herbs/Spices/Extracts	BEEF STOCK MIREPOIX SALT NATURAL FLAVOR YEAST ...	NaN	NaN	NaN
2	1105906	CAMPBELL SOUP COMPANY	NaN	440	g	Prepared Soups	CLAM STOCK POTATOES CLAMS CREAM VEGETABLE OIL ...	NaN	NaN	NaN
3	1105907	CAMPBELL SOUP COMPANY	NaN	440	g	Prepared Soups	WATER CREAM BROCCOLI CELERY VEGETABLE OIL MODI...	NaN	NaN	NaN
4	1105908	CAMPBELL SOUP COMPANY	NaN	240	ml	Herbs/Spices/Extracts	CHICKEN STOCK YEAST EXTRACT DEHYDRATED CHICKEN...	NaN	NaN	NaN

- We optimized the ingredient_table and created a string of Ingredients.

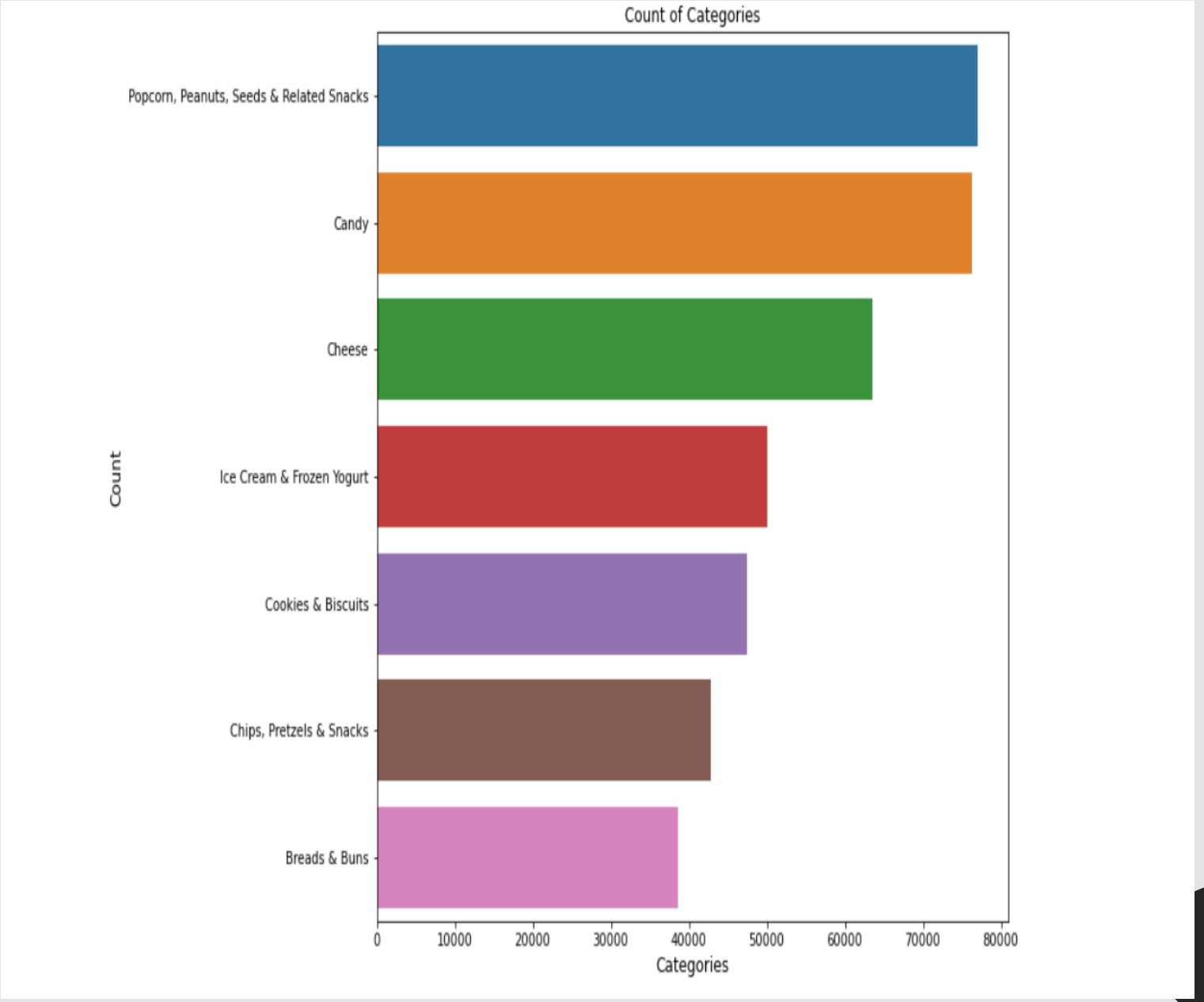
DATA VISUALISATION

Categories of Food in the Data

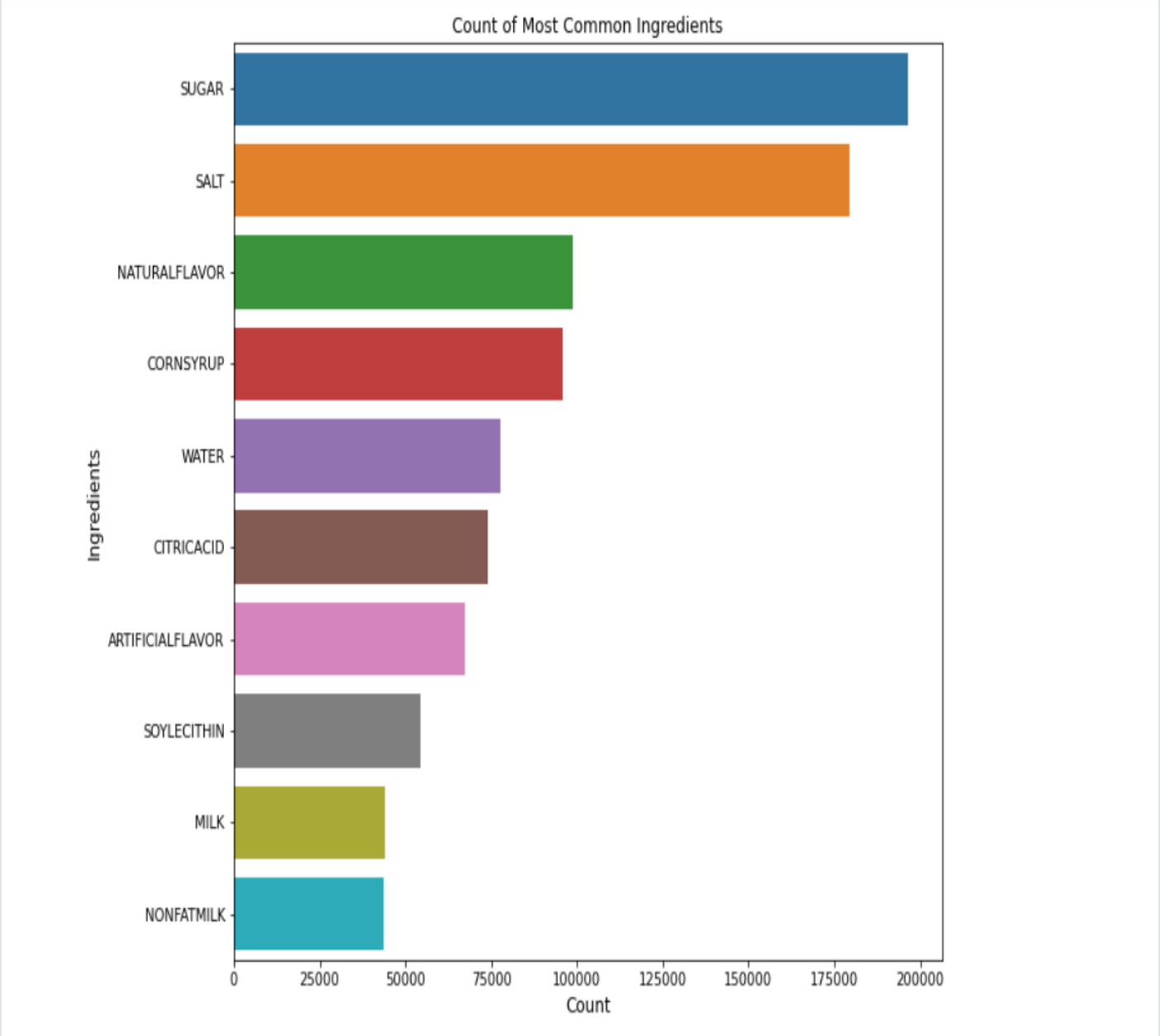
- In order to get a better idea of our data, we split it into two categories - Foods measured in grams and Foods in ML.
- For food in Grams, we got high value counts for foods such as Candy, Cheese, Ice Cream, Popcorn / Peanut related snacks.
- For food in ML, we got high value counts for juices, soda, water, and milk.



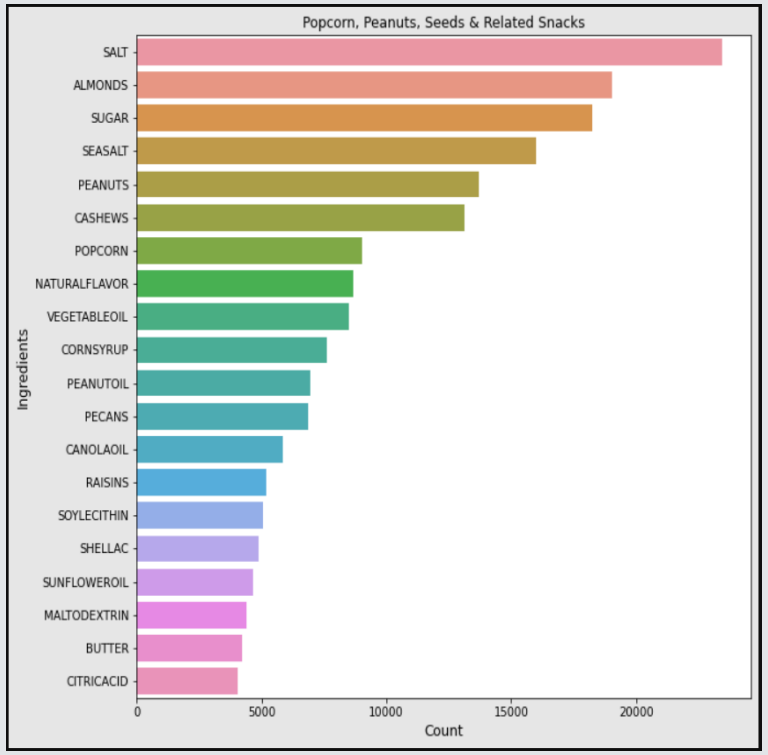
Number of samples in the Shortlist Dataframe

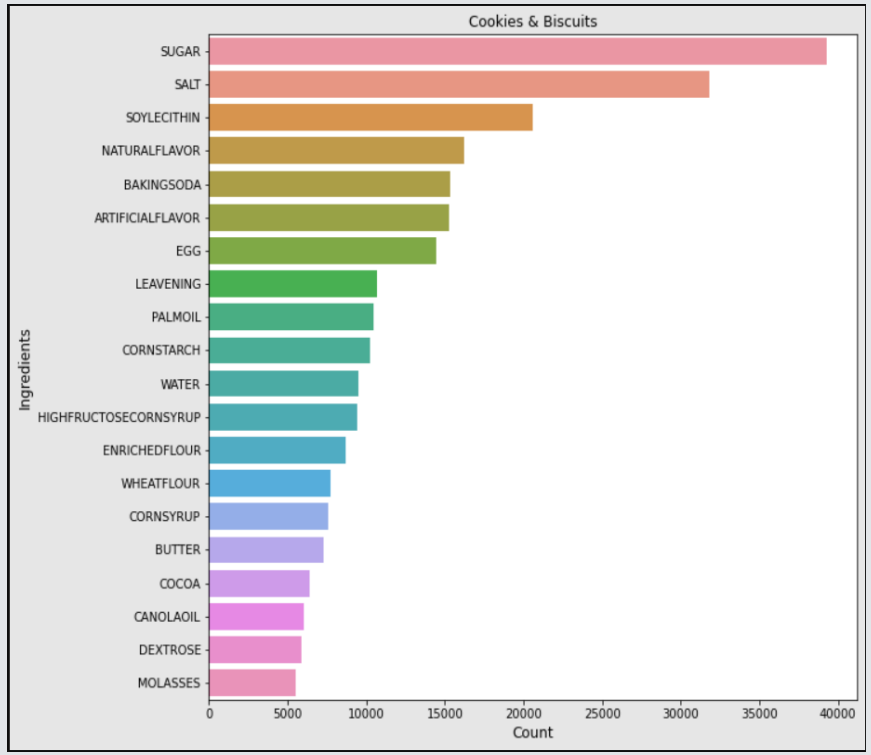


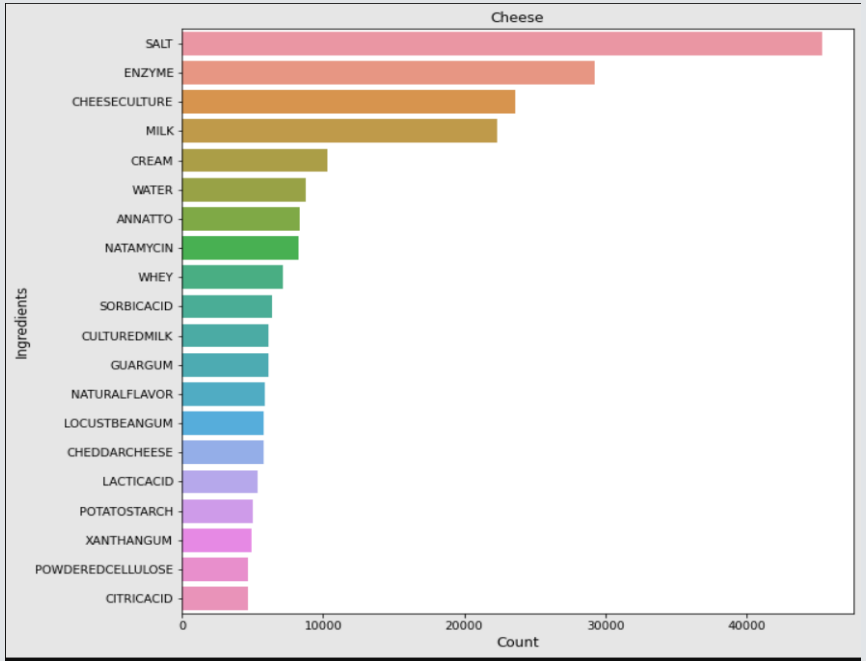
Count of the most common ingredients In the Shortlist Dataframe



Top 20 Ingredients in each category:







Categories List & Vectorization

```
Popcorn, Peanuts, Seeds & Related Snacks    77038
Candy                                       76226
Cheese                                      63552
Ice Cream & Frozen Yogurt                   50030
Cookies & Biscuits                          47472
Chips, Pretzels & Snacks                    42831
Breads & Buns                               38554
Name: branded_food_category, dtype: int64
```

- We created a data frame with a short list of ingredients. The number of samples in each category is shown on the left
- We utilized CountVectorizer on the ingredient_list to transform the categorical variables into an array of numbers.

Naïve Bayes Algorithm

- We initially tried implementing the NB algorithm with ingredient list as the independent variable and Branded Food Category as the dependent variable.

- However, this gave us a very low accuracy of 18%, where almost all the predictions were of one class.

- Hence, we moved onto implementing Support Vector Classifier.

Support Vector Machine

- We initially planned to work with SVM and not random forest, but we ran into some problems with SVM.
- The code did not return any errors, but it kept running for over 24hrs.
- Due to this we decided to stop working on SVM and shift to implementing Random Forest due to time constraints.

```
In [296]: print(classification_report(y_test,y_pred_unwt_test))
```

	precision	recall	f1-score	support
0	0.98	0.90	0.94	11593
1	0.90	0.87	0.89	22773
2	0.99	0.88	0.93	18979
3	0.97	0.58	0.72	12844
4	0.95	0.76	0.84	14255
5	0.99	0.79	0.88	15115
6	0.58	0.95	0.72	23152
accuracy			0.84	118711
macro avg	0.91	0.82	0.85	118711
weighted avg	0.89	0.84	0.84	118711

0: Breads & Buns, 1: Candy, 2: Cheese, 3: Chips, Pretzels, & Snacks, 4: Cookies & Biscuits, 5: Ice Cream & Frozen Yogurt, 6: Popcorn, Peanuts, Seeds & Related Snacks

Random Forest with Unweighted Ingredients

- We used the unweighted ingredient list as the only independent variable for our first trial.
- As an output for the random forest model, we got the classification report as the output. Max_depth was set at 20 which gave an accuracy of 83.54%
- We received high precision, recall and f1-scores for categories like breads & buns and Cheese.

Random Forest

A possible reason for some categories having a higher score than the others could circle back to the number of core ingredients in that product.

Having fewer base ingredients helps predict with higher accuracy.

Categories like breads and buns may generally have a base of some kind of flour.

Categories like Cheese would have core ingredients such as milk due to which we predict the scores were higher than the rest.

The Addition of weights

- Since ingredients on a product are listed in descending order of their quantity present, we have added weights to the ingredient list such that the first ingredient has the highest weight, and the last ingredient has the least.
- We cut down the ingredients of every product to the first 5 in order to make computation less expensive.
- Then, we created a new Weighted Ingredients column such that the first ingredient is repeated 5 times, the next 4, and so on.
- This would help the CountVectorizer differentiate between different quantities of each ingredient.

Example:

```
#Selecting a random row
display(shortlist_df.iloc[15])
```

```
fdc_id                                                        1106131
brand_owner                                                HP Hood LLC
brand_name                                                         NaN
serving_size                                                      88.0
serving_size_unit                                                    g
branded_food_category                          Ice Cream & Frozen Yogurt
ingredient_list           [MILK, FUDGESWIRL, BROWNIES, FRUCTOSE, EGG, CO...
fiveingred_column               [MILK, FUDGESWIRL, BROWNIES, FRUCTOSE, EGG]
ingredients_withweights   milk milk milk milk milk fudgeswirl fudgeswirl...
Name: 233, dtype: object
```

```
display(shortlist_df.iloc[15,8])
```

```
'milk milk milk milk milk fudgeswirl fudgeswirl fudgeswirl fudgeswirl brownies brownies brownies fructose fructose egg'
```

```
In [299]: #Printing the classification report
          y_pred_wt = clf_test.predict(X_wt_cv_test)
          print(classification_report(y_test,y_pred_wt))
```

	precision	recall	f1-score	support
0	0.96	0.79	0.87	11593
1	0.81	0.90	0.85	22773
2	0.98	0.88	0.93	18979
3	0.97	0.66	0.79	12844
4	0.93	0.61	0.74	14255
5	0.95	0.79	0.86	15115
6	0.61	0.96	0.75	23152
accuracy			0.82	118711
macro avg	0.89	0.80	0.82	118711
weighted avg	0.86	0.82	0.82	118711

```
In [300]: # Model Accuracy, how often is the classifier correct?
          print("Accuracy:",metrics.accuracy_score(y_test, y_pred_wt))
```

```
Accuracy: 0.8219204622991972
```

Random Forest with Weighted Ingredients

- We used the weighted ingredient list as the independent variable and received an accuracy of 82%
- Again, we received high precision, recall and f1-scores for categories like breads & buns and Cheese.

0: Breads & Buns, 1: Candy, 2: Cheese, 3: Chips, Pretzels, & Snacks, 4: Cookies & Biscuits, 5: Ice Cream & Frozen Yogurt, 6: Popcorn, Peanuts, Seeds & Related Snacks

Including More Independent Variables for Classification

- Since we have a mixture of text and numerical variables in our dataset, we used the ColumnTransformer as well as the Pipeline functions from scikit-learn.

- ColumnTransformer allows different columns of the input to be transformed separately and the features generated by each transformer will be concatenated to form a single feature space. This is useful for heterogeneous or columnar data, to combine several feature extraction mechanisms or transformations into a single transformer.

In [304]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test,y_pred_ss))
```

	precision	recall	f1-score	support
0	0.95	0.87	0.91	11593
1	0.85	0.90	0.87	22773
2	0.97	0.89	0.93	18979
3	0.98	0.66	0.79	12844
4	0.94	0.65	0.77	14255
5	0.94	0.93	0.94	15115
6	0.67	0.96	0.79	23152
accuracy			0.85	118711
macro avg	0.90	0.84	0.86	118711
weighted avg	0.88	0.85	0.86	118711

In [303]:
```
#Checking Accuracy
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred_ss))
```

```
Accuracy: 0.8544869472921633
```

0: Breads & Buns, 1: Candy, 2: Cheese, 3: Chips, Pretzels, & Snacks, 4: Cookies & Biscuits, 5: Ice Cream & Frozen Yogurt, 6: Popcorn, Peanuts, Seeds & Related Snacks

Random Forest with Weighted Ingredients & Serving Size

- We used the weighted ingredient list AND serving size as the independent variables and received a higher accuracy of 85%
- This shows that adding more independent variables to the model helps in better classification.

Conclusion

- We have used the Ingredients and Branded foods datasets to build all our models out of the four datasets given to us.
- We have cleaned the merged dataset and only included the data which is necessary for building the models to achieve our main objective.
- Due to the huge amount of the data in the merged data frame, we have shortlisted 7 categories and implemented the, Naïve Bayes and Random Forest Models on the new shortlisted data frame.
- As mentioned earlier, SVM was not successful.
- For the Random Forest model, when the unweighted ingredient list was considered as the independent variable, the model gave us 83.4% accuracy and with the weighted ingredient list it was only 82%.
- But then when we have added weighted ingredient list and serving size as the independent variables, we received a higher accuracy of 85%.
- This shows that adding more independent variables to the model helps in better predictions.


The End