University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

Ingredient Analysis using Natural Language Processing and Custom Named Entity Recognition and Machine Learning Techniques

Atharva Atul Joshi
2311708

Supervisor: Dr. Na You

September 18, 2024
Colchester

CHAPTER 1

Abstract

Today, the stores like Aldi and Tesco are filled with numerous options of food items to select from. There are numerous ingredients present including the cheapest and the costliest products. Upon closer observation, it is pretty evident that a lot of food items are filled with harmful preservatives, unhealthy oils and a plethora of additives.Various studies show the harmful effect of such ingredients on our health. A Grade Based System to classify the food products already exists in European Countries and there is a need to establish the same in UK.

The primary goal of the project is to create awareness amongst the consumers regarding the harmful ingredients present in the food items that we consume on daily basis. **The paper generates Custom Named Entity Recognition, an integral component of Natural Language processing (NLP) to bifurcate food products into 5 different Grades**.This is done using the spaCy library, which is a very important library that is used in NLP. Named Entity Recognition (NER) is a very important aspect of NLP and thus forms the basis of our project as we exploit the limitations of NER by using CNER. For creating the custom entities we have used NER Annotation tool and made use of DocBin for training the CNER.

Lastly, **the paper incorporates the use of various Machine Learning Algorithms for training and testing the created dataset**. Of the various algorithms used, XGBoost shows a significant result with promising accuracy after hyperparameter tuning.

Contents

1 Abstract **2**

2 Introduction **9**

2.1 Motivation 9

2.2 Aim and Objective 10

2.3 Sections Overview 11

3 Literature Review **13**

3.1 Research related to Nutrition and Food Additives 13

3.2 Use of AI, NLP in Food Label Analysis 14

4 Data Collection, Preprocessing and Basic Terminologies **16**

4.1 Data Collection 16

4.2 Text Extraction and Preprocessing 17

4.3 Basic Terminologies 18

4.3.1 Customized Named Entity Recognition 18

4.3.2 Natural Language Processing 18

4.3.3 Named Entity Recognition 19

4.3.4 Tf-idf vectorizer 19

4.3.5 Custom Named Entity Recognition for Food Data 20

4.3.6 Entity generation and annotation 21

4.3.7 Training the data 21

4.3.8 Rules for Bifurcation 22

5 Food Data Analysis **24**

5.1 Category 1: Chips, Nuts and Popcorn 24

5.1.1 Entities for Category 1 24

5.1.2 Training data for Category 1 24

5.1.3 Rules for Category 1 (Chips, Nuts and Popcorn) 25

5.1.4 Category 1: Ingredient Analysis 25

5.2 Category 2: Biscuits and Crackers 28

5.2.1 Entities for Category 2: Biscuits and Crackers 28

5.2.2 Training data for Category 2 29

5.2.3 Rules for Category 2 (Biscuits and Crackers) 29

5.2.4 Category 2: Ingredient Analysis 30

5.3 Category 3: Sauces and Salads 33

5.3.1 Entities for Category 3: Sauces and Salads 33

5.4 Training data for Category 3 33

5.4.1 Rules for Category 3 (Sauces and Salads) 33

5.4.2 Category 3: Ingredient Analysis 33

5.5 Holistic Ingredient Analysis 37

6 Methodology **40**

6.1 Model Training and Building 40

6.2 Feature Engineering 40

6.3 Data Partitioning 40

6.4 Creating Pipeline 41

6.5 Model Building 41

6.5.1 K-Nearest Neighbors 41

6.5.2 Multinomial Naive Bayes 43

6.5.3 Support Vector Classifier 44
6.5.4 Random Forest Classifier 45
6.5.5 XGBoost Classifier 46

7 Model Evaluation **48**
7.1 Confusion Matrix 48
7.2 Model Performance 53
7.3 Discussion 55

8 Conclusion **56**
8.1 Model Training Inference 56
8.2 Limitations 56
8.3 Future Scope 57

List of Figures

2.1 Flow of work 10

4.1 Flowchart for text extraction and processing 17

4.2 Flowchart of OCR Engine [18] 18

4.3 TF-IDF Vectorizer visual concept (Inspired by Let's Data Science blog) 20

4.4 NER Annotater Website [14] 21

4.5 Training Pipeline for CNER using DocBin and spacy [14] 22

5.1 Category 1 26

5.2 Pie chart for category 1 27

5.3 Custom Entities for Category 1 28

5.4 Category 2 30

5.5 Custom Entity distribution in percentage for Category 2 31

5.6 Custom Entities generated for Category 2 32

5.7 Category 3 34

5.8 Pie Chart Custom Entity distribution in percentage for Category 3 35

5.9 Bar plot for Custom Entities for Category 3 36

5.10 General Count 37

5.11 Overall Pie distribution 38

5.12 Grade-wise distribution of the custom entities 39

6.1 Model Training Pipeline [1] 41

6.2 Number of neighbors vs Accuracy 42

6.3 Working of SVM Hyperplane in 2D [21] 44

7.1 Confusion Matrix for KNN 49
7.2 Confusion Matrix for Multinomial Naive Bayes 50
7.3 Confusion Matrix for Support Vector Classifier 51
7.4 Confusion Matrix for Random Forest Classifier 52
7.5 Confusion Matrix for XGBoost Classifier 53
7.6 The Best Working Algorithm for Food Data (Sourced via EDUCBA) 54

List of Tables

6.1 Classification Report for KNN after Hyperparameter Tuning 43

6.2 Classification Report for Multinomial Naive Bayes after Hyperparameter Tuning 44

6.3 Classification Report for Support Vector Classifier after Hyperparameter Tuning 45

6.4 Classification Report for XGBoost Classifier after Hyperparameter Tuning . . 46

6.5 Classification Report for Support Vector Classifier after Hyperparameter Tuning 47

7.1 Accuracy Comparison of the algorithms used 54

CHAPTER 2

Introduction

2.1 Motivation

In India, the "Label Padhega India" (India Will Read the Labels) movement was recently launched to raise awareness about food labels and the incorrect marketing of packaged foods. The movement demonstrated how several well-known corporations misled locals by utilising labels such as "Fresh" items. This led the consumers to assume that the food packet was genuinely fresh, and they would purchase it. The caveat was that such labels were always accompanied by an asterisk, indicating that the label was employed solely for marketing purposes and did not indicate any actual significance. This emphasises the need of reading food labels when shopping.

Due to the abundance of options available in UK retailers such as Tesco, Aldi, and Sainsbury's for every type of food item, we find ourselves in a state of paralysis by analysis. We can't figure out what's best for us, so we're locked in a never-ending cycle of making decisions, and because we don't know how the food label will affect us, we end up making poor choices and purchasing the wrong food. This is supported by the research by Anisha et al., who indicate that when the cardinality of assortments increases, consumers are more likely to fall back on their default alternatives.[17]. So we can assume that in a store like Tesco, where there are so many different things to choose from, individuals will just pick the cheapest or most visually appealing option available because they are stuck in decision paralysis.

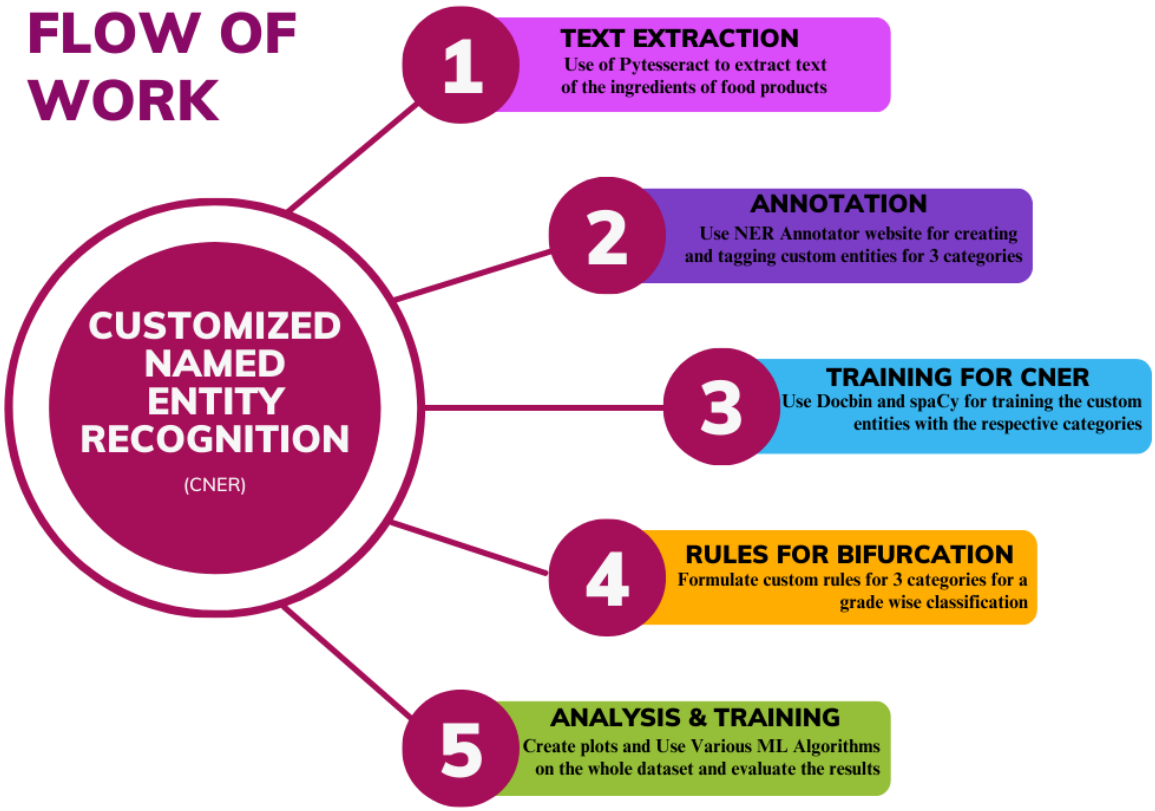

Figure 2.1: Flow of work

2.2 Aim and Objective

The following paper aims on harnessing the powers of Natural Language Processing,Python programming, Machine Learning and Deep learning techniques to bifurcate the packaged foods based on their ingredients. We start by using Google Pytesseract to extract the ingredients from a photo or screenshot of the food packet. The food products from three different categories are used for analysis. The categories are as follows: 1. Chips, crisps and nuts 2. Biscuits and bakery 3. Sauces and Salads

The initial batch of ingredients are then preprocessed and used for training our Custom Named Entity Recognizer (NER) using SpaCy library. We further use this custom NER for analysing the ingredients of every product that would be fit using the Custom NER. We would be defining the entities that we want the ingredients to be categorized into. Now the project is branched into two parts.

Part 1: The ingredients will be tokenized and then they will be analysed using TF-IDF

vectorizer to get a visualization of which ingredients and entity is frequently used. Part 2: The ingredients will be converted to a dataframe and then we will be using Python if-else logic and an in-depth research of the ingredients' effect and usage to classify the ingredients associated with to define some membership rules to categorize the the ingredients in a row into 4 distinct categories of edibility level.

2.3 Sections Overview

Both the parts together will constitute this project as whole, where in the part 1, focuses on analysis and deriving insights from the custom entities generated for the three categories and part 2 will focus on using python programming and Deep Learning for the classification of the food products.

In Chapter 3, we will review the literature associated with AI in food ingredients' classification. We will also examine the papers where NLP was used in the same use case. We will also be reviewing the domain-specific papers, related to nutrition and how various ingredients are added in any product which are completely unnecessary and harmful for our bodies.

In Chapter 4, We will discuss the 3 categories that we have selected to collect the data. We will also discuss about the text extraction using Pytesseract and furthermore, we will see a brief overview of the pre-processing done on the extracted text and why is that necessary. We will then dive deeper into the field of Named Entity Recognition, which is a part of Natural Language Processing. We will see how it is customized for our specific use case and how our data will be trained on the same. We will explore the use of DocBin package which will be used for the training.

In Chapter 5, we will perform Data Analysis on the ingredients we have . We will see the entities created by the Customized Named Entity Recognition (CNER) and discuss the python logic used to bifurcate the ingredients of one particular food item into the 5 grades (Grades A,B,C,D,E). There will be another section by the name "Undetermined". We will talk about the analysis that can be done on the ingredients of all the 3 categories. This will contain some visualizations and insights which can help the data scientist and the consumer to derive some patterns for strategizing their next actions.

Finally in Chapter 6, which is the Methodology, in which we will be training and building

our model using various classification algorithms. We are currently using 4 Classification algorithms, namely, K-Nearest Neighbour Classifier, Multinomial Naive Bayes Classifier, Support Vector Classifier, Random Forest Classifier and Extreme Gradient Boosting Classifier.

Chapter 7 will be about the results. We will also evaluate the results using the evaluation metric for classification problem. We will see the Confusion matrices for all the algorithms

In Chapter 8, we will conclude our project by mentioning what we have learnt from it, the limitations we faced and the scope for future work.

CHAPTER 3

Literature Review

We are going to investigate some associated research in two sections: The first segment will include a review of publications that emphasise the significance and consequences of several of the components we will use. These will include studies conducted all throughout the world on the science of nutrition. The second segment will go into the application of AI and NLP in food data classification. It will also include some earlier work in the field of custom named entity recognition.

3.1 Research related to Nutrition and Food Additives

Nutritional scientist Michael Moss, in his exploration of junk food, investigates how food corporations deliberately manipulate sugar, salt, and fat to activate the brain's 'bliss point,' inciting consumers to desire increased consumption. He draws attention to the fact that products like chips sourced from major grocery retailers, such as Tesco and Aldi, are extensively processed and infused with oil, sugar, salt, and fat to enhance their flavour and appeal. This emphasis on flavour frequently results in consumers disregarding the detrimental ingredients, leading Moss to reproach these corporations for their indirect contribution to the public health predicament [16].

Moss further elaborates on these concepts in his publication [15], where he examines the duplicitous strategies employed by food corporations to portray their products as nutritious. He contends that visually appealing packaging and deceptive health assertions engender a spurious perception of nutritional worth, leading consumers to opt for unhealthy choices under the pretense of healthfulness. These revelations, coupled with nutrition centric podcasts, motivated the decision to investigate this subject and utilize Natural Language Processing

(NLP) and Artificial Intelligence (AI) to construct a comprehensive ingredient classification model.

The Health Influencer on Social Media Revant Himatsingka, an alumnus of Wharton MBA program, initiated the "Label Padhega India" campaign, advocating for transparency in food labelling practices. He critiques corporations for promoting unhealthy products as healthful, imploring them to embrace truthful labelling methodologies. His endeavours have prompted certain companies, such as Pepsi-Co, to diminish detrimental components like palm oil in their products, thereby marginally enhancing their health profile [2].

These investigations have established the groundwork for this research by illuminating the deceptive tactics utilized by the food industry. The forthcoming section scrutinizes consumer behaviour within supermarket environments, particularly how a deficiency in awareness influences sub-optimal purchasing choices. Angela Bearth et al. [3] examine consumer perceptions regarding food additives, uncovering pervasive misconceptions. Numerous consumers erroneously classify commonplace ingredients such as sugar and salt as additives, failing to acknowledge the wider and often more hazardous substances present.

The inquiry of Professor Erik Millstone into food additive regulations within the UK offers additional insights into labelling conventions [13]. He elucidates the misleading distinctions between terms such as "flavoured" and "flavour," noting that items labelled as "bacon-flavoured" frequently lack actual bacon, instead containing artificial flavourings. Millstone posits that while these additives confer no substantial benefit to consumers, they significantly serve the commercial objectives of the food industry.

Collectively, these studies exemplify how the food industry manipulates consumer choices towards unhealthy products through misleading practices. The subsequent section will explore the utilization of AI in the analysis and classification of food labels..

3.2 Use of AI, NLP in Food Label Analysis

In the research conducted by Guanlan Hu et al. [9], a dataset referred to as FLIP (Food Label Information and Price) from the University of Toronto was utilized to derive a nutrition quality score alongside the corresponding classification of food categories employing a modified pretrained sentence-Bidirectional Encoder Representations (BERT). This investigation

has effectively harnessed cutting-edge technology known as Transformers, which facilitates highly precise classification for textual data. The authors have incorporated Natural Language Processing (NLP), Machine Learning, and food labels for the categorization process. They have achieved optimal results utilizing the XGBoost Classifier Algorithm. This finding signifies that the XGBoost Classifier demonstrates commendable efficacy in text classification, thus prompting our decision to employ the XGBoost Classifier.

The paper authored by Y Zhang et al. [24] discusses the amalgamation of Deep Learning within Food Label Classification. It describes the various food datasets that are available worldwide, including the single-class food dataset, multi-class food data, ChineseNET, and FruitNET. Consequently, there exists a remarkable diversity of food data types. The document also elucidates several feature engineering methodologies that may be applied, such as SMOTE, which is employed for class balancing, alongside various Machine Learning algorithms that have been utilized. Thus, the paper references an array of methodologies employed in food label classification.

A related study was performed by Peihua Ma et al. [12], wherein four distinct Deep Learning Models were employed, specifically, Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), Graph Neural Network (GNN), and Convolutional Neural Network (CNN). They utilized the USDA Branded Food Product Database and implemented feature engineering techniques including the TF-IDF vectorizer, one-hot encoding, and the application of SMOTE for dataset balancing. The paper also concludes that the aforementioned deep learning algorithms have exhibited significantly superior performance compared to traditional Machine Learning Algorithms, with the Multi-Layer Perceptron being identified as the most effective model. This outcome is noteworthy as it underscores the advantageous suitability of Deep Learning Algorithms in comparison to Machine Learning Algorithms.

The article by Jing Li et al. [11] provides insights into the history of Named Entity Recognition (NER). The 6th Message Understanding Conference (MUC6) marked the inaugural instance of NER being employed to categorize textual data into "people," "organization," and "geographical location." NER constitutes a critical component of NLP, as it differentiates one type of word from another. The document designates NER as a fundamental pillar in the preprocessing of textual data for various applications, including Information Retrieval, question answering, and machine translation.

CHAPTER 4

Data Collection, Preprocessing and Basic Terminologies

4.1 Data Collection

We will collect 500 photos in total to create a dataset, which will include three categories. The first category is of **Chips, Nuts and Popcorn**, containing 200 images. The second category is **Biscuits and Crackers**, which also contain of 200 images. The third category is **Sauces and Salads**, containing 100 images. These 3 categories consist of all assorted food products which public usually looks forward to as snacking items. Consumers are usually conscious of the main 3 meals, which are breakfast, lunch, dinner. So there is a high chance that consumers will make a proper healthy choice. The tendency to turn towards one of the above-mentioned categories is the in-between meals by picking packets with appealing packaging, ignoring the nutritional values. This finding highlights the significance of reading product labels, especially ingredient lists, in order to make well-informed choices. The choice of these particular food categories for our investigation was influenced by this realisation. All the images of food products were sourced from official websites of Tesco UK and Aldi UK.

4.2 Text Extraction and Preprocessing

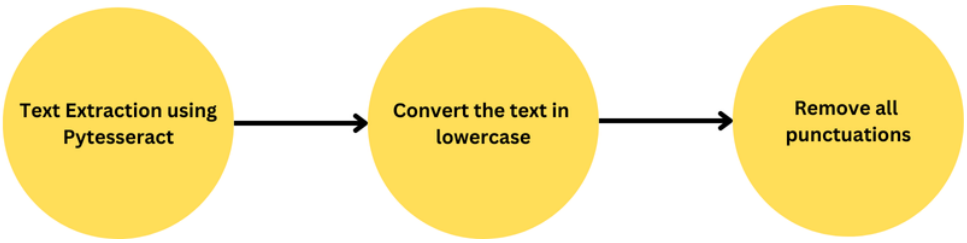

Figure 4.1: Flowchart for text extraction and processing

We will be using Google Colab for writing the code for this project. It is essential that the google drive containing all the images and other base files is mounted on colab.

After importing all the necessary libraries and mounting the google drive, we need to make sure that the extracted text will be in a standard format without any special characters and in lowercase. So we will be creating a function called **pre-processing** which contains the code for bring the extracted text in a normal lowercase format without any special characters, with all the ingredients separated by a comma. We will be using Python Regex (Regular Expression) and string methods for the pre-processing. Once the function is created, in the next cell we will be using Google Pytesseract for the text extraction of the category-1 images.

Pytesseract is an Optical Character Recognition Engine whose origins date back to 1984 as PhD Project funded by the Hewlett and Package. It was later acquired by Google in 2005 and now comes under Google [18] The paper by Saoji et al. [18] discusses in detail how the OCR works and what steps it follows. We can view it in the following image from the same paper

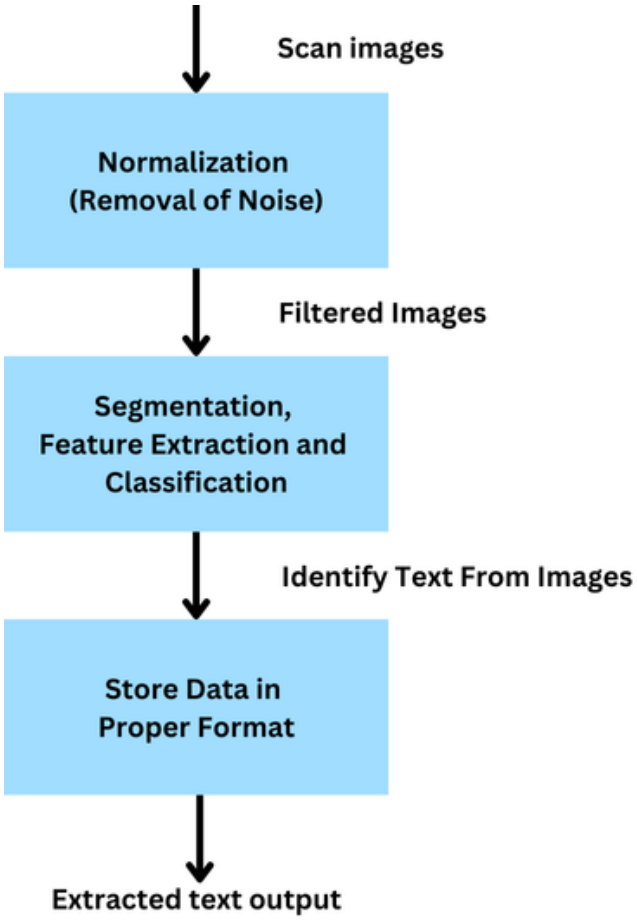

Figure 4.2: Flowchart of OCR Engine [18]

4.3 Basic Terminologies

4.3.1 Customized Named Entity Recognition

Before diving deep into Customized Named Entity Recognition, it is essential to understand the concept of Named Entity Recognition, what is natural language processing and why is it needed.

4.3.2 Natural Language Processing

When humans read a sentence, say, " A quick brown fox jumps over a lazy dog", we automatically interpret the meaning in our head by segregating the words according to different parts of speech and by picturing the scenario in our heads. Humans have the innate ability to naturally process the linguistic and semantic data, according to their comprehension. So

Natural Language Processing is a part of Artificial Intelligence, which refers to a technique that has some computational abilities to analyse the text at par with the humans using various methods like Stemming, Lemmatization, Vectorization, Parts of Speech Tagging and many more. NLP comes with its limitations which are mentioned in this paper [8]. Nevertheless, NLP can still perform analysis at word level, and still bring out various insights. NLP can be thought of like a very unbiased technique to read and analyse text data, as even though the level of processing by human is far more superior, it comes its own limitation. The paper by Chowdhary et al. [8] mentions these limitations. When a human reads something, they will associate it with some relevant experience that happened in the past, some other form of sensory experience which makes our processing biased. NLP consists of various techniques that we can use for text analysis. We will be taking a look at each one of them in this chapter.

4.3.3 Named Entity Recognition

It is one of the NLP techniques that is used for the purpose of text pre-processing. Named Entity Recognition (NER) finds out which entity the words in that sentence or paragraph belong. The paper by Sharnagat et al. mentions the three types of entities present [19], they are as follows:

1. ENAMEX: Person, organisation location

2. TIMEX: Date, time

3. NUMEX: Money, percentage and quantity

We will be using the **spacy** library for creating our own custom entities recognition.

4.3.4 Tf-idf vectorizer

Mathematically, tf-idf vectorizer is the logarithm of the ratio of the total number if documents to the number of documents containing a certain word. In simple words, it shows the importance a particular word when compared with all the words in the documents.

TF- Term Frequency, is the number of times a word appears in one document.

IDF- Inverse Document Frequency, is the number of times a word appears in the corpus of documents. It is given by the following equation,

$$\text{TF-IDF}(t, d, n) = \text{TF}(t, d) \times \text{IDF}(t, n)$$

We will be using Tf-idf vectorizer in the model training stage along with machine learning and deep learning algorithms.

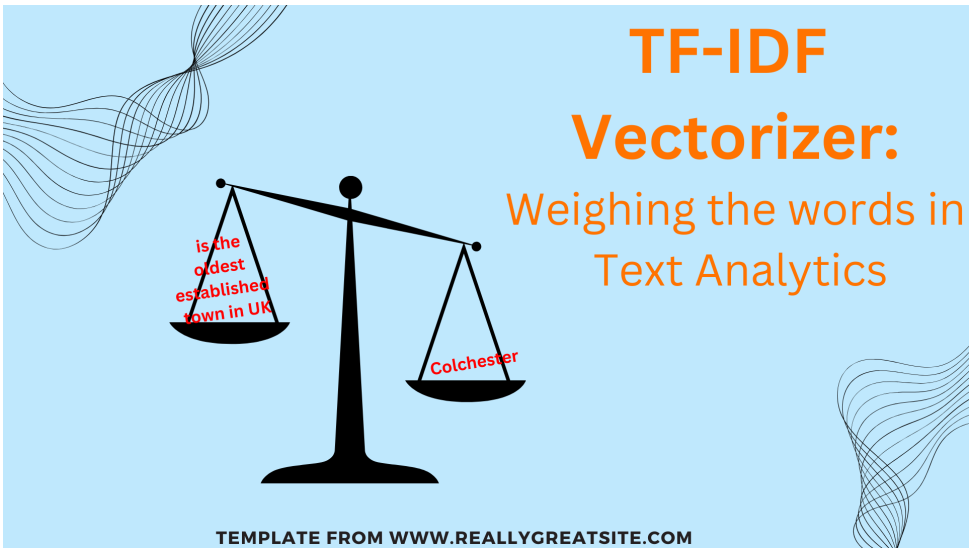

Figure 4.3: TF-IDF Vectorizer visual concept (Inspired by Let's Data Science blog)

4.3.5 Custom Named Entity Recognition for Food Data

One of the limitations of the spacy library is that it contains only a few entities which are in-built in the library. But there is also an option to customize the entities according to our use case and our needs. This is what we are calling Custom Named Entity Recognition (CNER). We will creating a CNER using the spacy library as it gives false positives and false negatives while classifying. This is important to make our model more robust [20]. Now, we will take a look into the steps taken to create a CNER for the three categories. A point to note is, we will be following the same steps for creating CNERs for three categories. There is a need to create 3 different CNERs for the three categories as the ingredients in the 3 categories will be different. For example, the ingredients of chips should be analysed separately from the ingredients of a sauce. There will be different conditions when it comes to the selection of ingredients for the three categories. Okay, let's understand the general steps of creating a CNER.

4.3.6 Entity generation and annotation

We take the help of an annotator website to tag an ingredient to it's corresponding entities. So, the annotator website gives us an option to generate some entities that we want to use to label our dataset. In our case, entities can be along the lines of "Safe oil", "Harmful Oil", "Harmful Additive" and so on. We will be creating custom entities for each category. Once that is done, we will be providing an assorted list of ingredients for tagging. We will tag the ingredients with their corresponding entities. The website will then annotate the ingredients and give us an output file in the form of a dictionary. The annotator website that we will be using is: https://agateteam.org/spacynerannotate/ and following is the screenshot of the interface.

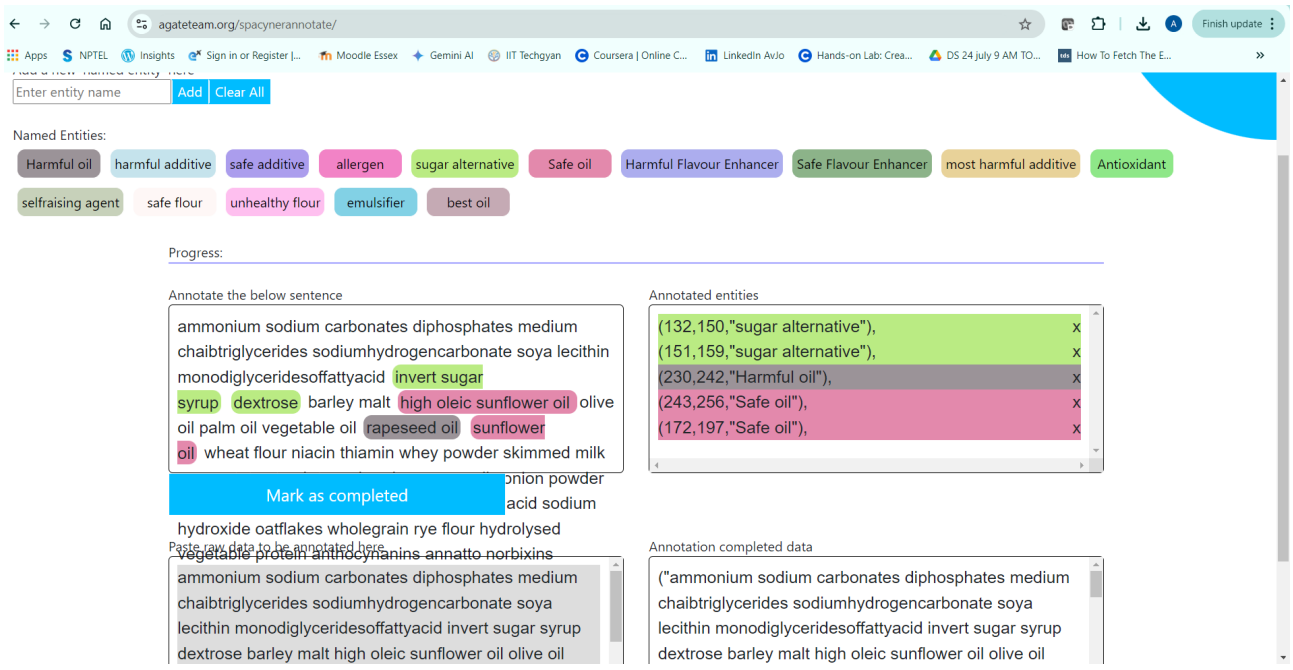

Figure 4.4: NER Annotater Website [14]

4.3.7 Training the data

The annotated ingredient-entity pair as given in the bottom right in the above screenshot is stored as train data. We will then make use of DocBin package to convert the train data into the spacy format [14]. This tutorial gives a detailed explanation of how the conversion and further training is done. The following 5 steps are an overview of what is mentioned in the tutorial to train a spaCy NER pipeline, according to our train data:

1. The preparation of training data, examples, and their labels

2. Transforming data into the spacy format

3. Producing the configuration file needed to train the model

4. The necessary parameters being added to the configuration file

5. Execute the Instruction

A model is trained in spaCy using an iterative procedure where the gradient of the loss is calculated by comparing the model's predictions to the labels (ground truth). Next, using the back-propagation process, the model weights are updated based on the gradient of the loss. The gradients indicate how much the weight values should be altered to make the model's predictions more similar or closer to the specified labels over time[14].

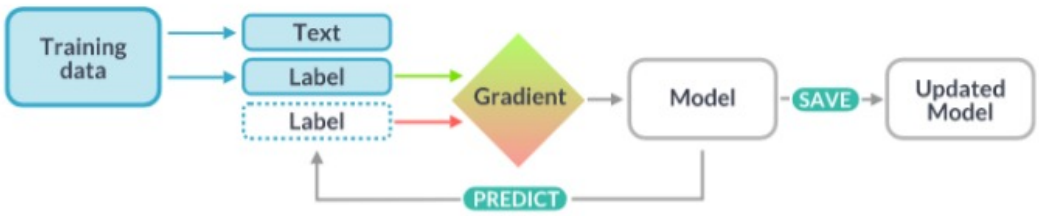

Figure 4.5: Training Pipeline for CNER using DocBin and spacy [14]

4.3.8 Rules for Bifurcation

These are a set of rules built using pandas dataframe and if-else logic. These rules enable classifying ingredients of any of the three category into one of the 4 Food Grades, namely,

1. Grade A Healthy

2. Grade B Healthy

3. Grade C Healthy

4. Grade D Healthy

5. Grade E Healthy

Grade A being the healthiest edible product and Grade E being unhealthiest edible product. Such Grading is important as consumers can't easily identify the use of an ingredient except for the popular few ingredients. It is therefore necessary for grading the food products, so

that the consumers tend to prefer Grade A, B products over Grade C,D and E. A general thumb rule for identifying Grade C, Grade D and Grade E products from all the categories is that these products will contain around 15-20 ingredients, or even more. Similarly, Grade A and Grade B products will contain fewer ingredients, with good oil or no oil at times. Depending on the category. Some data which cannot be graded due to lack of training will be classified into "Undetermined" class.

CHAPTER 5

Food Data Analysis

We will perform the analysis on the food data that we have and also on the custom entities that we will generate when we train using spaCy. This will be done on all the three categories. We will also see rules of bifurcation for all the three categories.

5.1 Category 1: Chips, Nuts and Popcorn

As the whole focus of the project is to analyse the ingredients of products that consumer endear, the first category will include the ingredients of Chips, Nuts and Popcorn.

5.1.1 Entities for Category 1

Following are the custom entities created for Category 1: [Harmful Oil, Safe Oil, Harmful Additives, Safe Additive, Most Harmful Additive, Sugar Alternative, Harmful Flavour Enhancer, Safe Flavour Enhancer, Antioxidant]

5.1.2 Training data for Category 1

The following ingredients were used training and building the CNER Model for category 1,

[palm oil, vegetable oil, rapeseed oil, xanthan gum, guar gum, sodium diacetate, maltodextrin, monosodium glutamate,malic acid,citric acid, acidity regulator, disodium ribonucleotide, potassium chloride, wheat flour, niacin, thiamin, gluocose syrup, emulsifier, sunflower oil, gram flour, maize, wholewheat flour, garlic powder, turmeric powder, paprika powder, mango, rice flour, seasoning]

5.1.3 Rules for Category 1 (Chips, Nuts and Popcorn)

MonoSodium Glutamate is an harmful flavour enhancing additive, which is used in Chips to give the Umami taste. It comes under the category of GRAS (Generally Recognized as Safe) but according to this paper by Songul et al [4], there are various studies that show the opposing negative effects of MSG but there is still no prohibition of any sorts on it's usage. As a result of which, we decided to take MSG as the **Most Harmful Additive** and its presence in any food product along with the count of entity **Harmful Additive** more than 6, renders that product as Grade E Healthy, which is the unhealthiest food product. If the product contains MSG, but contains less than 6 harmful additives, then that product gets classified as Grade D healthy

Now, as we had mentioned earlier, as food product will be classified as Grade A Healthy, if it contains less than 5 ingredients. The number 5 can be played around with, but it generally holds true as any food product with less than 5 ingredients will have minimal or no addition of harmful additives, harmful flavour enhancers. It might contain harmful oil, but then again, it is void of other ingredients. Many times a safe oil is used when the number of ingredients are less than 5 as the manufacturer has specially made such product to be healthy.

Now, we are left two intermediate grades, Grade B and Grade C. The rules for these two are based on the type of oil (Safe, Harmful) and the type of the additives (Safe, Harmful). If a product contains any of the safe oils (Sunflower, Olive) and if the number of safe additives are more than the number of harmful additives, it will be classifies as Grade B Healthy. It is an extension of Grade A Healthy Rule as the number of ingredients are increased in this case.

On the other hand, if none of the safe oils are used but if we have the number of safe additives more than harmful additives, then the product will get classified as Grade C Healthy.

5.1.4 Category 1: Ingredient Analysis

Scatter-plot for two principal components

We will be using PCA technique for dimensional reduction for the TF-IDF vectorizer. TF-IDF vectorizer converts the text data into numerical vectors which can then be uses for further training.

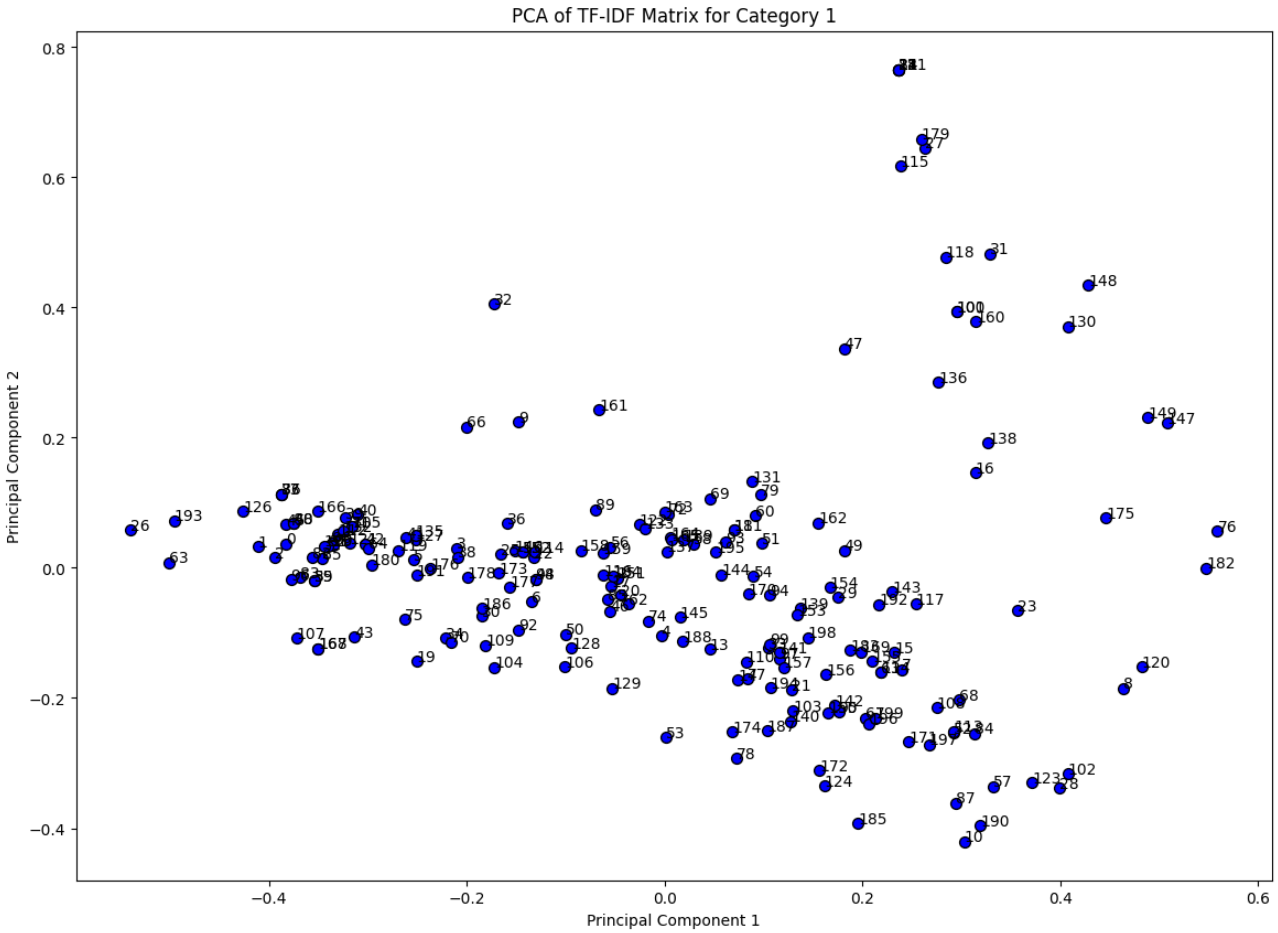

Figure 5.1: Category 1

1. The points which are clustered closer to each other indicate the similarity of ingredients present in them.

2. The indexes of the ingredients are shown in the plot instead of the plots.

3. The plot shows that Category 1 has many similar ingredients as the ingredients are close by.

Pie Chart for Custom Entities

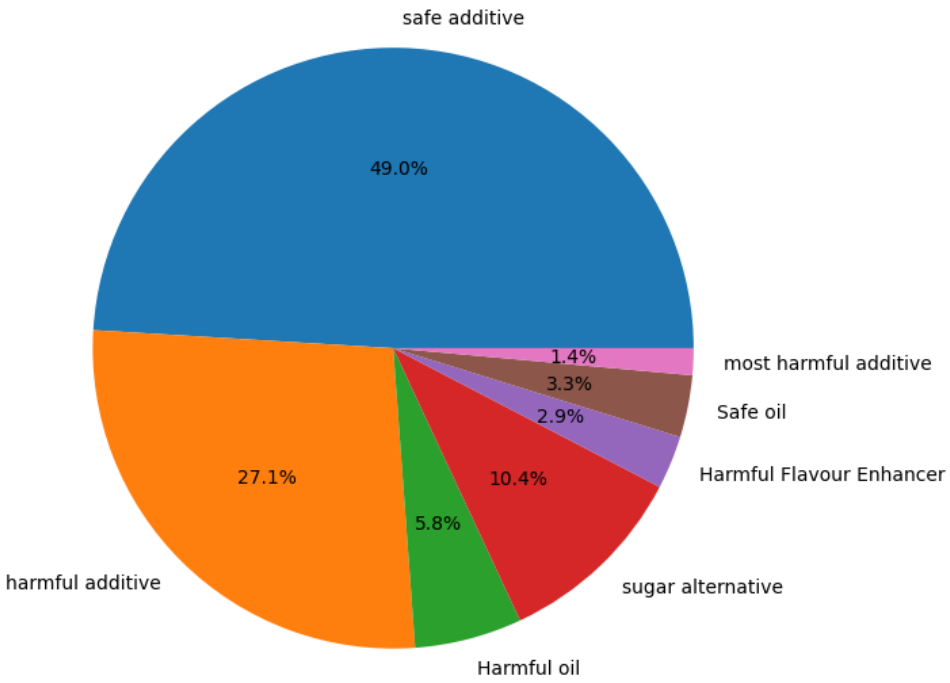

Figure 5.2: Pie chart for category 1

1. 49 percent safe additives are present

2. The cumulative share of harmful entities is slightly than safe entities.

Bar plot for custom entities

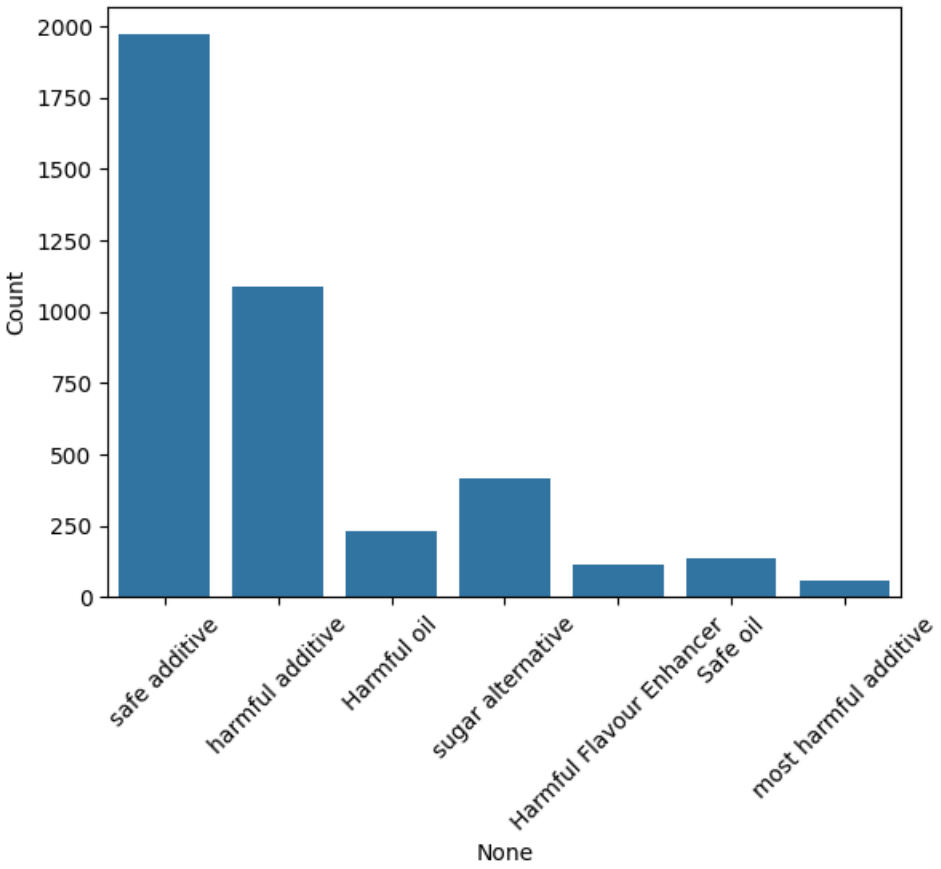

Figure 5.3: Custom Entities for Category 1

1. The bar plot shows the count of the custom entities in the Category 1.

2. Relatively the count of harmful and safe is even in Category 1

5.2 Category 2: Biscuits and Crackers

5.2.1 Entities for Category 2: Biscuits and Crackers

Following are the custom entities created for Category 2: [Harmful Oil, Safe Oil, Harmful Additives, Safe Additive, Most Harmful Additive, Sugar Alternative, Harmful Flavour Enhancer, Safe Flavour Enhancer, Antioxidant, Self raising agent, safe flour, harmful flour, emulsifier]

5.2.2 Training data for Category 2

The following ingredients were used training the CNER for category 2,

[ammonium sodium carbonates diphosphates medium chain triglycerides sodiumhydrogencarbonate soya lecithin monodiglyceridesoffattyacid invert sugar syrup dextrose barley malt high oleic sunflower oil olive oil palm oil vegetable oil rapeseed oil sunflower oil wheat flour niacin thiamin whey powder skimmed milk corn potato starch tocopherol pepper garlic onion powder chocolate cocoa acidity regulator folic malic acid sodium hydroxide oatflakes wholegrain rye flour hydrolysed vegetable protein anthocynanins annatto norbixins]

5.2.3 Rules for Category 2 (Biscuits and Crackers)

Now, in the second category, the base entities like types of oil, types of additive remain the same, what is added is the type of flour (Safe and Unhealthy), along with the entities of emulsifier, self-raising agent and antioxidant.

So, the rule for Grade A Healthy classification, is that the number of ingredients to be less than 5, but it also needs to be made from a safe flour. So what are safe flours? The ones that we get usually, are only wheat flour, which is not very great. There various healthier options like Wholewheat flour, Oats flour, etc. So inclusion of these healthier flours in the food product is a great sign of an healthy option.

The food product will be classified as Grade B Healthy, if the number of ingredients are greater than 5, if it uses a healthy flour and the oil type is a safe one.

For Grade C Healthy, the requirement is that the oil type must be a safe one even if unhealthy flour options are used.

If the flour used is unhealthy, along with the excess use of sugar alternatives (more than 3), but oil is a safe one, then that particular food product will be classified as Grade D Healthy.

Now in Grade D Healthy, suppose the oil type is also a harmful, then that makes the product the worst of all, and is therefore classifies as Grade E Healthy

5.2.4 Category 2: Ingredient Analysis

Scatter-plot for two principal components

Similar to Plot 1, we will be using PCA on the tf-idf vectorizer matrix for category 2. Here's the plot drawn against the two principal components.

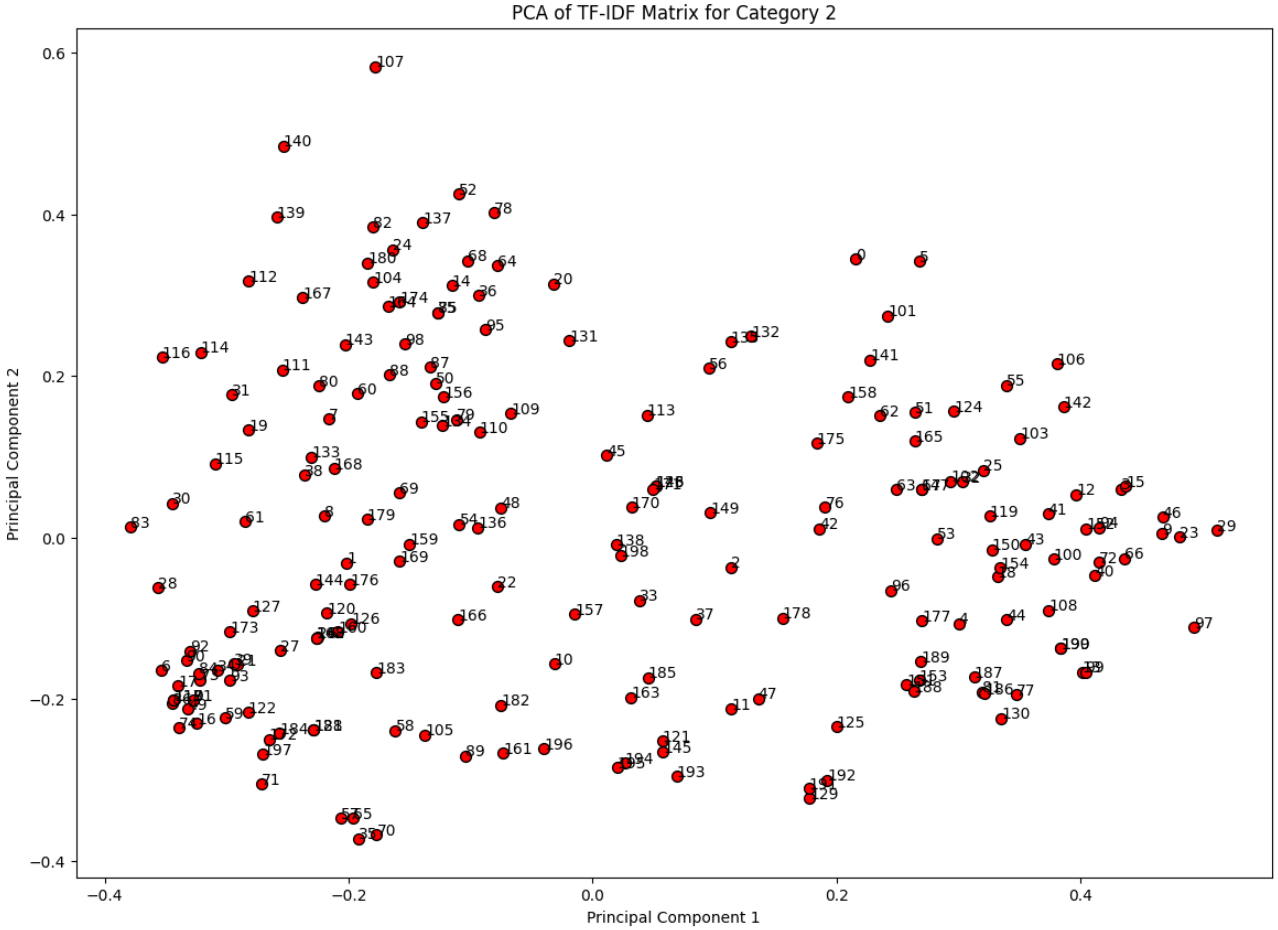

Figure 5.4: Category 2

1. The points which are clustered closer to each other indicate the similarity of ingredients present in them.

2. The indexes of the ingredients are shown in the plot instead of the plots.

3. While similarity still exists, it is scattered all over the graph and it is not as good as Category 1.

Pie Chart for Custom Entities

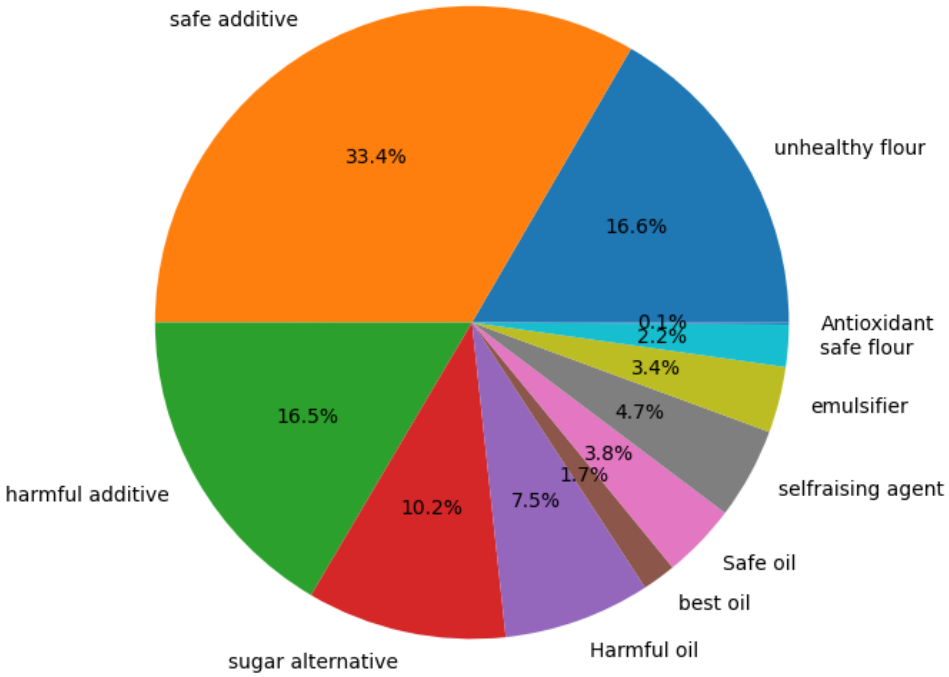

Figure 5.5: Custom Entity distribution in percentage for Category 2

1. Safe additives contribute the most with 33.4 percentage
2. Unhealthy flour contributes to 16.6 percentage
3. Harmful additives contribute with 16.5 percentage

Bar plot for custom entities

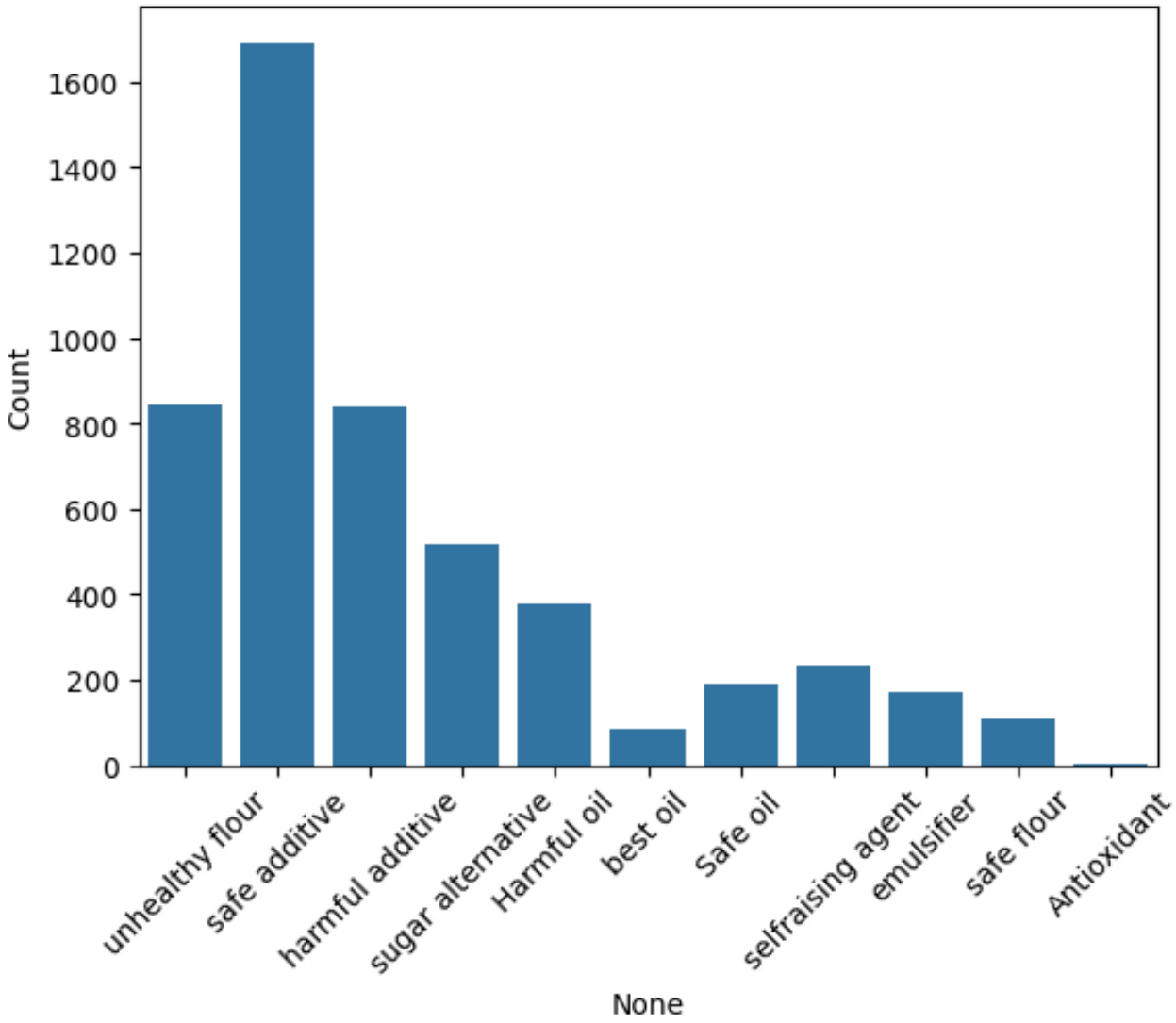

Figure 5.6: Custom Entities generated for Category 2

1. Safe additives has the highest frequency.
2. Despite safe additive being high in number, there are a lot of unhealthy entities, the cumulative total of that becomes very high.
3. The usage of unhealthy flour is higher than that of safe flour, which is not a good sign

5.3 Category 3: Sauces and Salads

5.3.1 Entities for Category 3: Sauces and Salads

Following are the custom entities created for Category 3: [Harmful oil, harmful additive,safe additive,allergen,sugar alternative,Safe oil,Harmful Flavour Enhancer,Safe Flavour Enhancer, Antioxidant, Harmful stabilizer]

5.4 Training data for Category 3

The following ingredients were used training the CNER for category 3, [Guar gum, xanthan gum, onion powder, garlic, Sodium metabisulphite, spirit vinegar, tomatoes, puree, rapeseed oil, palm oil, egg yolk, acetic acid, sodium citrate, sulphite, steviol, glucose-fructose syrup, milk, lemon juice, starch, pectin, molasses, sugar, sunflower oil, barley malt, potassium sorbate, E361, E627, balsemic vinegar, natural flavouring, riboflavin, paprika, pepper, celery powder, mustard, sweetner, ascorbic, rosemary]

5.4.1 Rules for Category 3 (Sauces and Salads)

Similar to the first two categories, the rule for "Grade A Healthy" is that the number of ingredients should be less than 5 in that particular food item.

For Grade B Healthy, it is required that safe oil should be used and the entities labelled safe additives should be greater than the harmful additives. But if the oil is a harmful type, and the safe additives count is greater than harmful additives, then it belongs to the Grade C Healthy.

Few of the stabilizers or thickeners, like, Xanthan gum, Guar gum are absolutely unnecessary and therefore, if any food products contain these along with harmful additives count more than 3 or presence of sugar alternatives entities, then these products are classified as Grade E Healthy. If only stabilizers are present, then they are bifurcated into Grade D Healthy

5.4.2 Category 3: Ingredient Analysis

Similarly, here is the plot for category 3:

Scatter-plot for two principal components

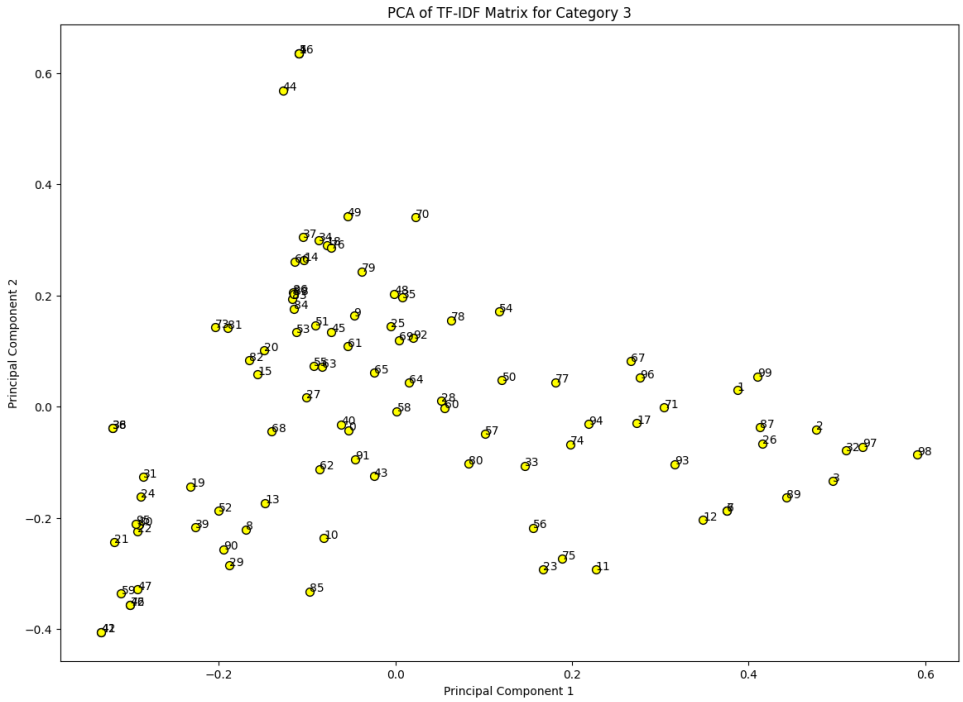

Figure 5.7: Category 3

1. The points which are clustered closer to each other indicate the similarity of ingredients present in them.

2. The indexes of the ingredients are shown in the plot instead of the plots.

3. The similarity for Category 3 is also scattered, showing a variety of ingredients present in the category.

Pie Chart for Custom Entities

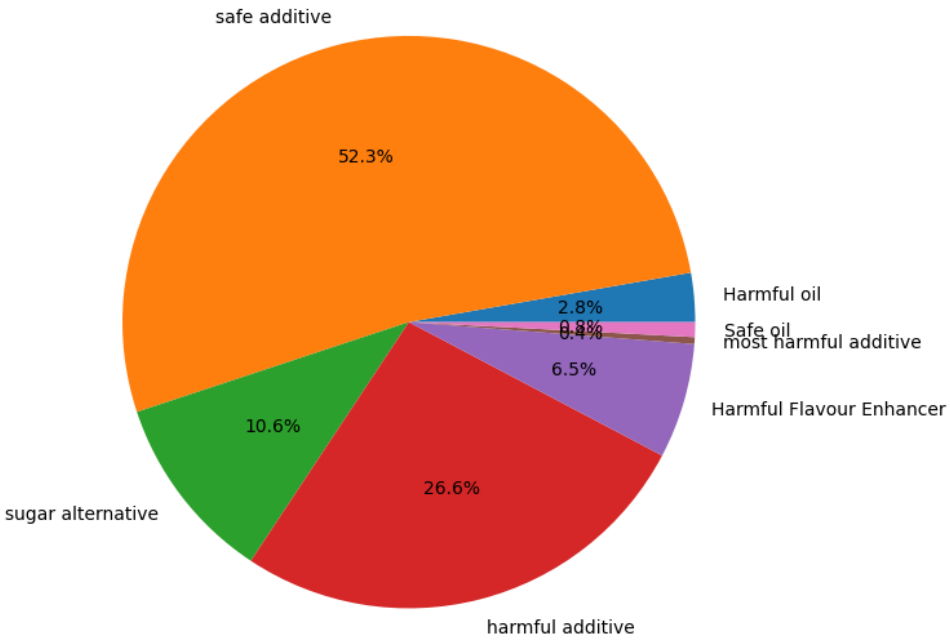

Figure 5.8: Pie Chart Custom Entity distribution in percentage for Category 3

1. Safe additives contribute the most with 52.3 percentage
2. Harmful additives contribute 26.6 percentage
3. Sugar Alternatives contribute 10.6 percentage
4. Harmful flavour enhancer contributes 6.5 percentage
5. Harmful oils contribute 2.8 percentage
6. Safe oil contributes 0.8 percentage

Bar plot for custom entities

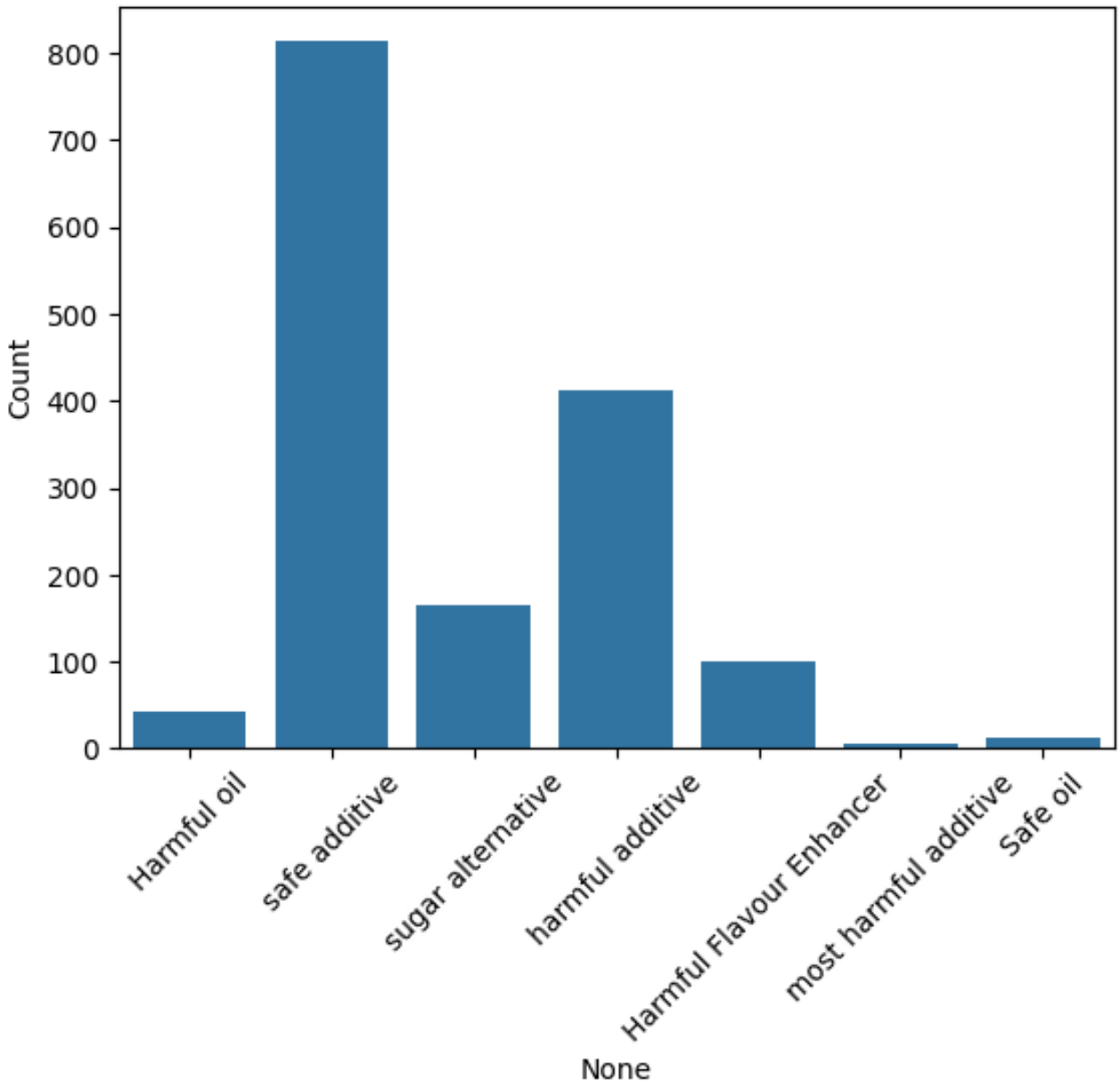

Figure 5.9: Bar plot for Custom Entities for Category 3

1. It is observed that there is a huge contribution of safe additives in the category 3.
2. Despite that there is a significant amount of harmful additives, sugar alternatives and harmful and flavour enhancers combined

5.5 Holistic Ingredient Analysis

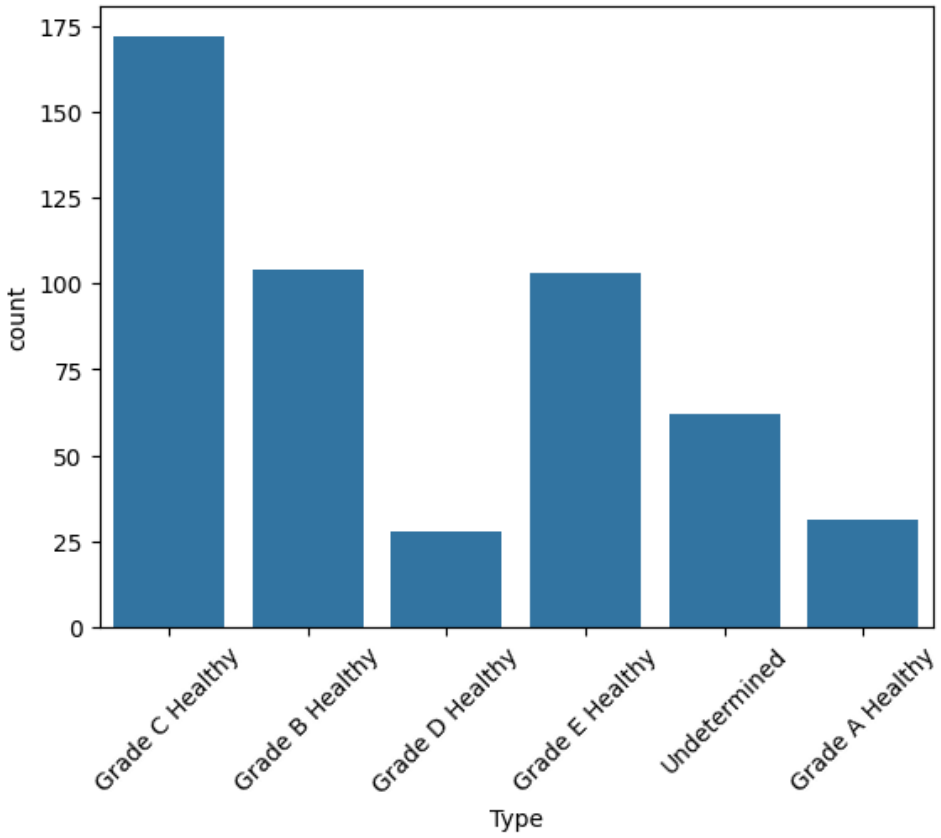

Figure 5.10: General Count

1. Most food products are classified into Grade C Healthy
2. Grade D has least number of food products
3. Very few products are classified as Grade Healthy

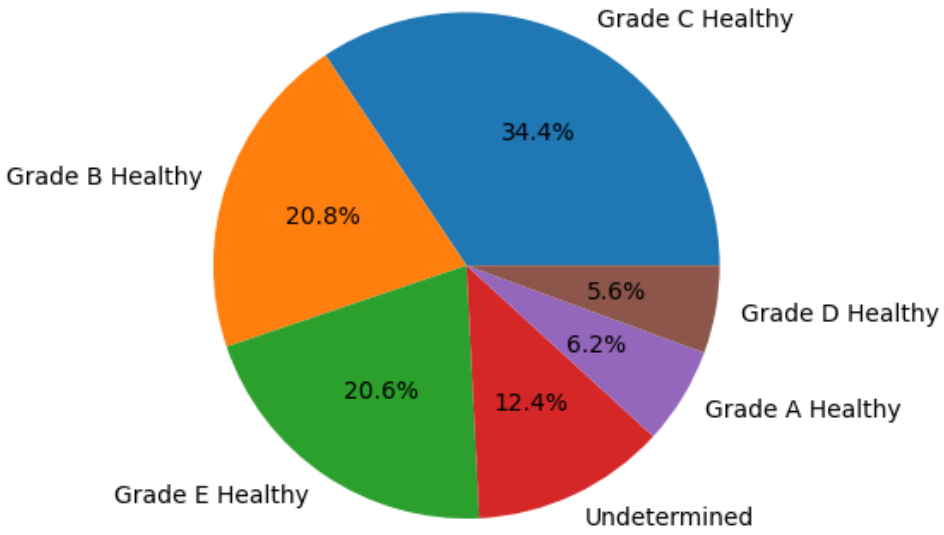

Figure 5.11: Overall Pie distribution

1. Grade C Healthy contributes 34.4 percentage
2. Grade B Healthy contributes 20.8 percentage
3. Grade E Healthy contributes 20.6 percentage
4. Grade A Healthy contributes 6.2 percentage
5. Grade D Healthy contributes 5.6 percentage
6. 12.4 percentage remains Undetermined.

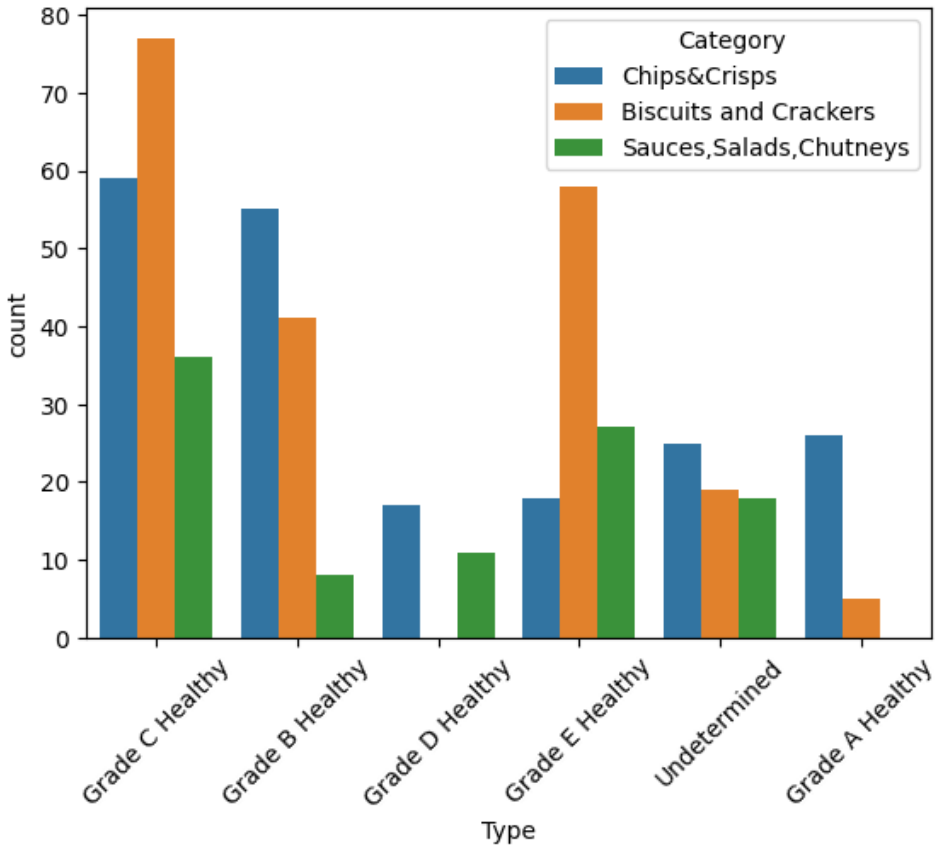

Figure 5.12: Grade-wise distribution of the custom entities

1. This count plot gives the count of Grades for products done category wise.

2. Count of Biscuits is highest in Grade C and Grade E Healthy products.

3. Grade B, Grade D, Grade A Healthy section is dominated by Chips and Crisps

CHAPTER 6

Methodology

6.1 Model Training and Building

Category 1 has 200 entries, category 2 has 200 entries and category 3 has 100 entries. So now, we have a dataset which consists of final combined dataframe of 500 rows and 3 columns. The first column is the ingredients' list that we extract with the help of tesseract. The second column consists of the name of one of the 3 categories in which that particular food product belongs. The final column consists of which grade of healthiness that food product falls in.

6.2 Feature Engineering

Feature Engineering is needed to prepare the data for model building. One of the feature engineering techniques that we will have to incorporate is label encoding for the target variable. Label encoding assigns a number to each specific task as it appears in the dataframe. As we have 6 categories in total (including the "Undetermined" category), we will get 6 numbers in place of the categories in the target column.

6.3 Data Partitioning

An efficient data split is needed, for efficient model training. We need to split the data in training and testing. It is needed so that the model learns on the train data and it can evaluate its learning on test data. We have split 75 percentage randomly into training and 25 into testing.

6.4 Creating Pipeline

We will be creating total 5 pipelines for 5 models. The pipeline will contain a Tf-idf vectorizer, which will be used on the "Ingredients" column and it will be followed by the model we would be working on.

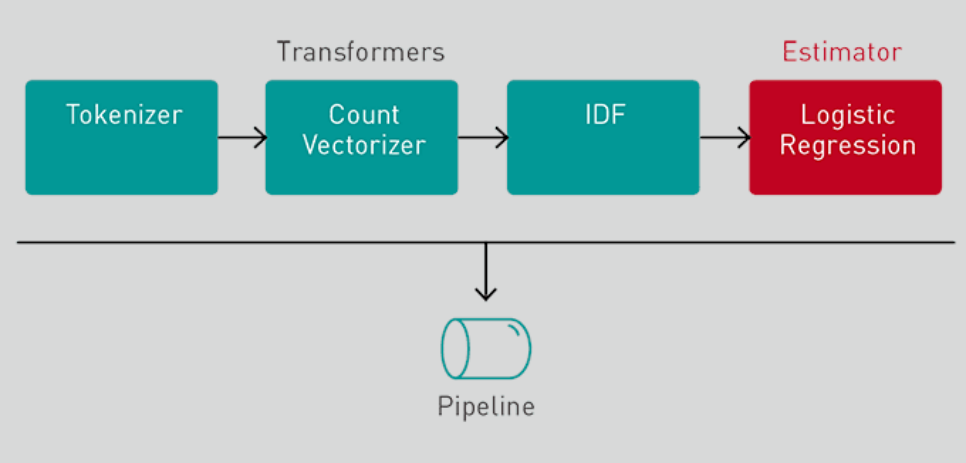

Figure 6.1: Model Training Pipeline [1]

6.5 Model Building

As mentioned in the introduction section, we will be using 5 models. Classifiers such as Naive Bayes and SVM are thought to be linear, scalable, and effective for large text corpus [6]. We will be also performing hyperparameter tuning and checking the increase and decrease of accuracy and f-1 scores of our models

6.5.1 K-Nearest Neighbors

In order to categorise an unknown document (d0), the KNN classifier uses the class labels of the k most similarity neighbours to forecast the class of the input document. The neighbours of the document are ranked among the training documents [22]. The same paper goes on to mention that the KNN can also have some limitations when it comes to handling data which is not evenly distributes. As in our case, the data in 3 categories is 200,200 and 100, it has an inherent uneven distribution present. So we get a very low accuracy and low f-1 scores for KNN Classifier.

The distance that need to be calculated in KNN can be: Minowski Distance:

$$D(a,b) = \left(\sum_{i=1}^{n} |a_i - y_i|^p \right)^{\frac{1}{p}}$$

where,

if p=1, then it becomes Manhattan Distance,

if p=2, then it becomes Manhattan Distance

KNN gives an accuracy of 0.54 after tuning it with a range of ideal number of neighbours. The ideal neighbors is 5.

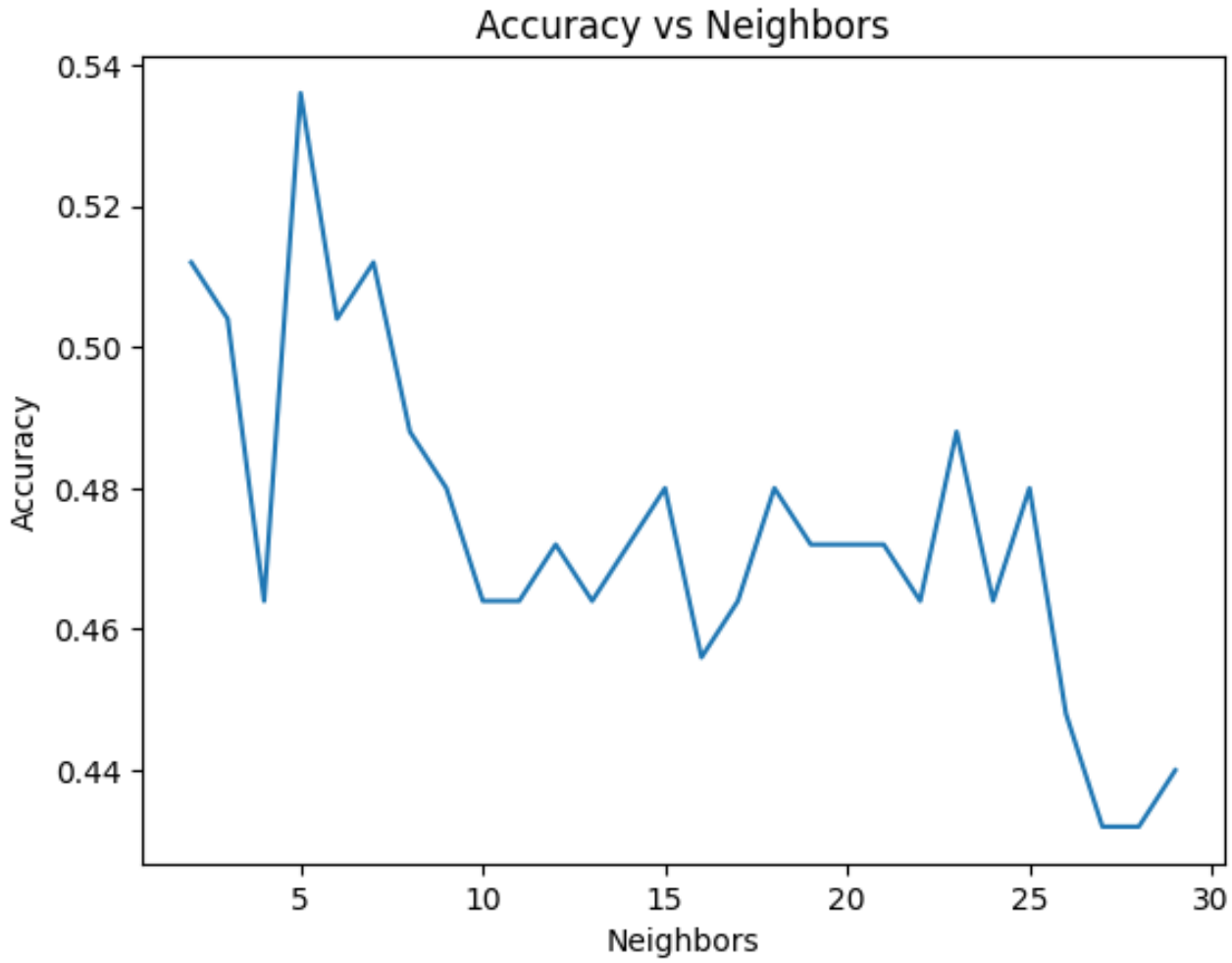

Figure 6.2: Number of neighbors vs Accuracy

Grade	Precision	Recall	F1 score
A	0.33	0.43	0.38
B	0.33	0.48	0.39
C	0.67	0.72	0.69
D	0.43	0.60	0.50
E	0.62	0.52	0.57
Undetermined	0.50	0.12	0.20
Accuracy			0.54

Table 6.1: Classification Report for KNN after Hyperparameter Tuning

6.5.2 Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic classifier used for multiclass classification. It works especially well for text categorisation since it assumes features have a multinomial distribution. Under the assumption of feature independence, the model applies the Bayes theorem to determine the probability of each class given a set of features. It uses the frequency of features in the training data to estimate class probabilities. In order to predict the class with the highest posterior probability, it computes the likelihood of observing the feature values for each class and combines it with the prior probability of that class. This method performs effectively when dealing with text data and categorical features.

The probability of a document D belonging to class c is given by:

$$P(c|D) = \frac{P(c) \prod_{i=1}^{n} P(w_i|c)^{f_i}}{P(D)}$$

where:

- $P(c|D)$ is the posterior probability of class c given document D.
- $P(c)$ is the prior probability of class c.
- $P(w_i|c)$ is the probability of word w_i given class c.
- f_i is the frequency of word w_i in document D.
- $P(D)$ is the probability of document D.

According to this paper by Shuo Xu and et al. [23], Multinomial Naive Bayes Classifier works better at handling class imbalance than KNN, for text data classification. The paper also states that Mulitnomial NB Classifier is well suited for a simple dataset. We get a slightly better accuracy and F-1 score for this algorithm than KNN.

Multinomial Naive Bayes has an accuracy of 0.5 and f-1 scores are also 0 in 2 classes and after tuning the accuracy increases to 0.54 with the f-1 scores distributed between 5 out of the 6 classes.

Grade	Precision	Recall	F1 score
A	0.33	0.14	0.20
B	0.42	0.62	0.50
C	0.60	0.77	0.67
D	0.00	0.00	0.00
E	0.70	0.55	0.62
Undetermined	0.67	0.12	0.21
Accuracy			0.54

Table 6.2: Classification Report for Multinomial Naive Bayes after Hyperparameter Tuning

6.5.3 Support Vector Classifier

An SVM classifier must be trained by identifying a hyperplane that, when used as its decision surface, has the greatest margin of separation between the positive and negative training examples[21].

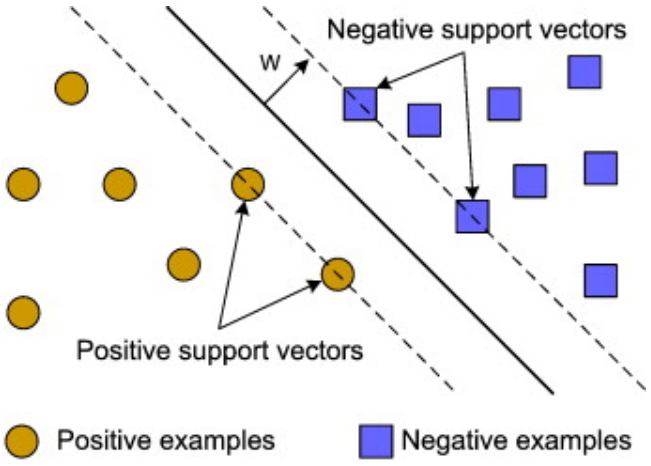

Figure 6.3: Working of SVM Hyperplane in 2D [21]

It is one of the popular algorithm when it comes to text data, so we have used it in our model building and it can be seen via the model as we get a better accuracy than Multinomial NB.

The support vector classifier gives an accuracy of 0.58 and it stays the same even after performing hyperparameter tuning. The difference is found in the f-1 scores as the results are more evenly distributed as compared to the model before tuning.

Grade	Precision	Recall	F1 score
A	0.60	0.43	0.50
B	0.44	0.57	0.50
C	0.66	0.70	0.68
D	0.17	0.20	0.18
E	0.65	0.59	0.62
Undetermined	0.55	0.38	0.44
Accuracy			0.58

Table 6.3: Classification Report for Support Vector Classifier after Hyperparameter Tuning

6.5.4 Random Forest Classifier

An ensemble of tree-structured classifiers is called a Random Forest. Each input is given the most likely class label based on a unit vote cast by each tree in the forest. In addition to being quick and noise-resistant, this effective ensemble can spot non-linear patterns in the data. Both numerical and categorical data can be handled with ease by it. One of Random Forest's main benefits is that, even when additional trees are added to the forest, it does not experience overfitting[5].

Mathematical Representation

Let $\{T_1, T_2, \ldots, T_n\}$ be the set of decision trees in the forest. For a given input x, the prediction of the i-th tree is $T_i(x)$. The final prediction $\hat{y}$ is given by:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_n(x)\}$$

Gini Impurity

The Gini impurity for a set of items with J classes is calculated as:

$$I_G(p) = 1 - \sum_{j=1}^{J} p_j^2$$

where p_j is the probability of an item being classified to class j.

Random forest performs poorly after hyperparameter tuning as the accuracy drops from 0.59 to 0.57, and not all the classes are properly distributed when we compare the f-1 scores.

Grade	Precision	Recall	F1 score
A	0.62	0.71	0.67
B	0.55	0.57	0.56
C	0.73	0.77	0.75
D	0.43	0.60	0.50
E	0.62	0.55	0.58
Undetermined	0.77	0.62	0.69
Accuracy			0.66

Table 6.4: Classification Report for XGBoost Classifier after Hyperparameter Tuning

6.5.5 XGBoost Classifier

One of the most recently developed algorithm, as it has already shown great improvements in the accuracies and f-1 scores. The objective function of XGBoost can be written as:

Mathematical Representation

$$\text{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where: - l is the loss function (e.g., mean squared error for regression). - Ω is the regularization term. - f_k represents the k-th tree in the model. - $\hat{y}_i$ is the predicted value for the i-th instance.

The regularization term Ω is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

where: - T is the number of leaves in the tree. - w_j is the weight of the j-th leaf. - γ and λ are regularization parameters.

The prediction for a given instance x_i is:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

XGBoost has some great benefits over other models as it is scalable as it can handle sparse data. It is also not sensitive to overfitting [7]. As we saw in the paper by Guanlan Hu and et al, XGBoost gives great results when it comes to food data. Our results corroborate the statement, as we get the best results using XGBoost Classifier.

The accuracy of XGBoost increases from 0.63 to 0.66 after hyperparameter tuning and even the f-1 scores are nicely distributed better than the other models between the classes.

Grade	**Precision**	**Recall**	**F1 score**
A	0.60	0.43	0.50
B	0.44	0.57	0.50
C	0.66	0.70	0.68
D	0.17	0.20	0.18
E	0.65	0.59	0.62
Undetermined	0.55	0.38	0.44
Accuracy			0.58

Table 6.5: Classification Report for Support Vector Classifier after Hyperparameter Tuning

CHAPTER 7

Model Evaluation

7.1 Confusion Matrix

Confusion Matrix will play an important part in highlighting how the test data has been classified. In a classification problem, the test data can be classified as following:

1. True Positive: Number of test cases correctly predicted as belonging to a specific class
2. True Negative: Number of test cases correctly predicted as to not belonging to a specific class.
3. False Positive: Number of test cases incorrectly predicted to belong to a specific class, but they belong to a different class
4. False Negative: Number of test cases incorrectly predicted to not belong to a specific class, but they belong to that class

If the number of False Positives and False Negatives is higher for any particular class, then that behaviour is not appreciated and needs to be altered by some optimizing techniques. We will take a look at the Confusion Matrices for all our Machine Learning Models.

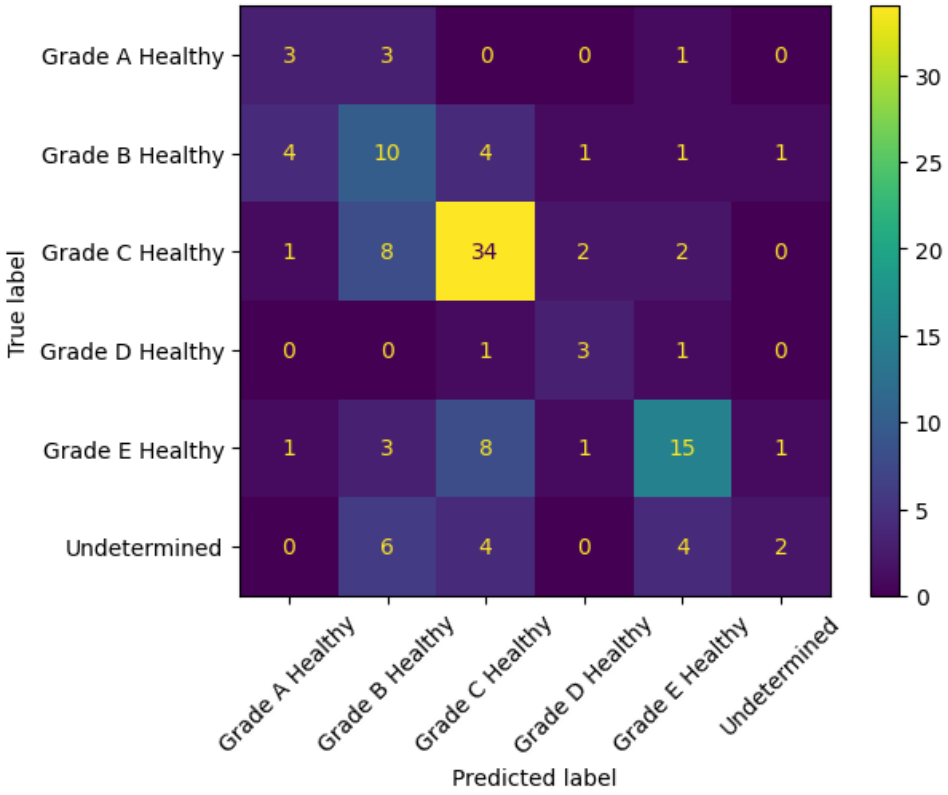

Figure 7.1: Confusion Matrix for KNN

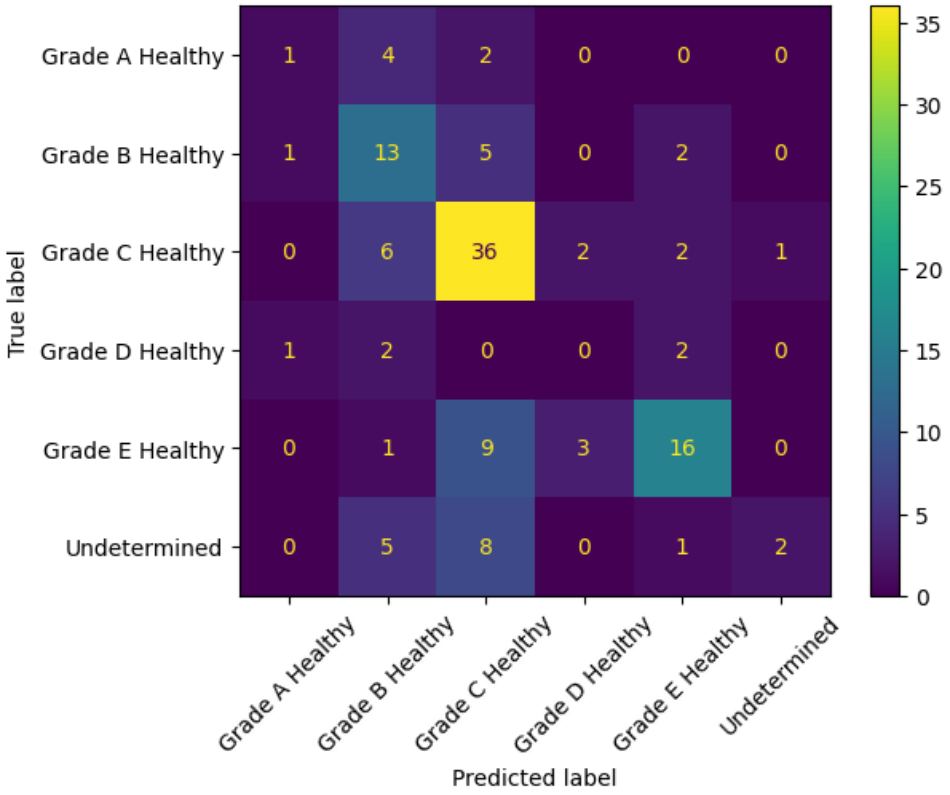

Figure 7.2: Confusion Matrix for Multinomial Naive Bayes

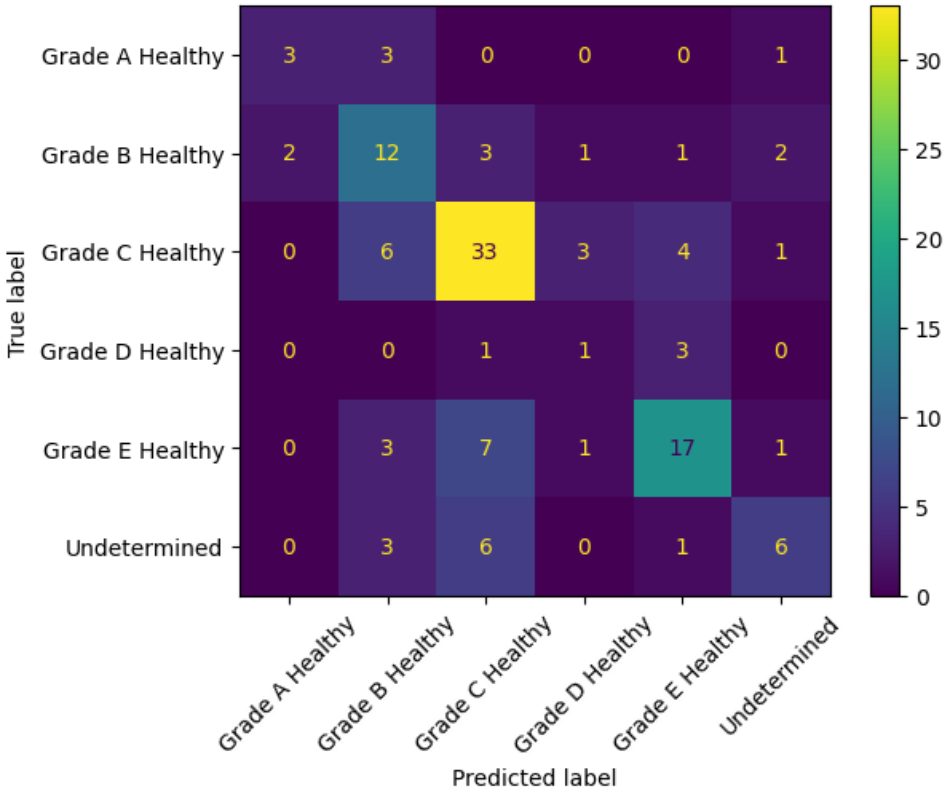

Figure 7.3: Confusion Matrix for Support Vector Classifier

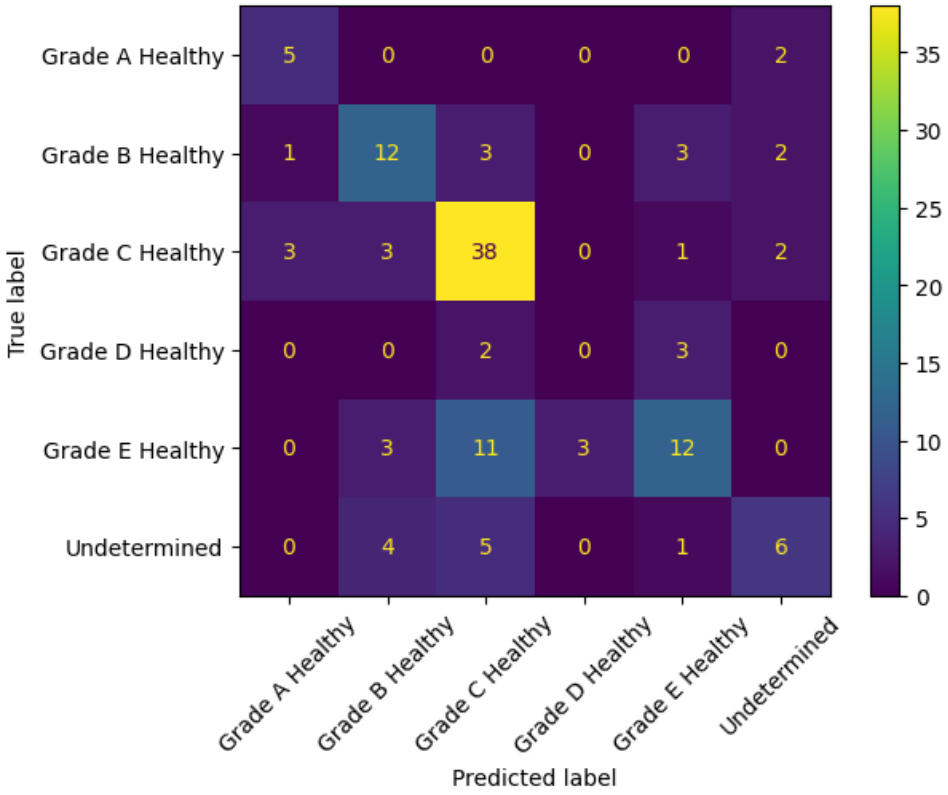

Figure 7.4: Confusion Matrix for Random Forest Classifier

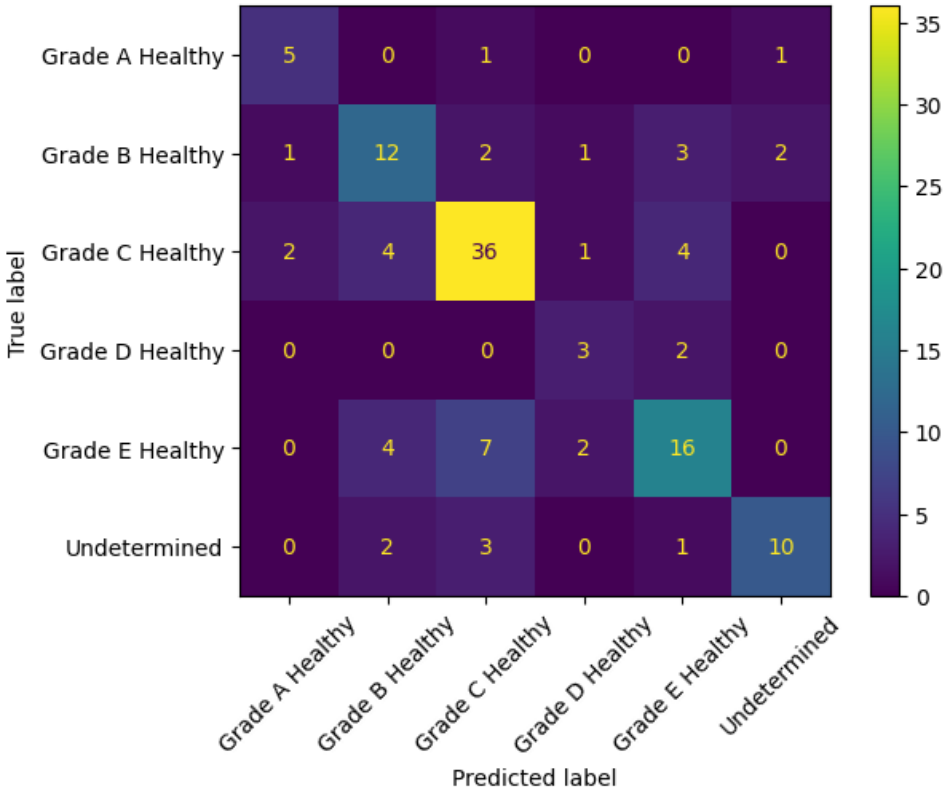

Figure 7.5: Confusion Matrix for XGBoost Classifier

1. The diagonal elements in the show the True Positive classifications for all the Grades.
2. In terms of True Positives, Random Forest Classifier is the Algorithm that has highest number of True Positives.
3. XGBoost Classifier has second most number of True Positives
4. The Off-diagonal elements represent the misclassified values

7.2 Model Performance

Following is the comparison table of all the Algorithms we used:

Out of the 5 models that were tried, XGBoost gave the best results, with the accuracy of 0.66 after hyper parameter tuning.

ML Algorithm	Accuracy
K-Nearest Neighbours	0.54
Multinomial Naive Bayes	0.54
Support Vector Classifier	0.58
Random Forest Classifier	0.57
XGBoost Classifier	0.66

Table 7.1: Accuracy Comparison of the algorithms used

Figure 7.6: The Best Working Algorithm for Food Data (Sourced via EDUCBA)

Text classification tasks using TF-IDF Vectorizer with XGBoost frequently result in good accuracy because TF-IDF efficiently captures the relevance of words in texts, converting raw text into numerical data that highlights informative phrases. Because of its strong feature selection and ability to manage non-linearity, the gradient-boosted decision tree technique XGBoost performs exceptionally well when working with high-dimensional, sparse datasets such as those generated by TF-IDF. Studies reveal that the combination of TF-IDF's feature extraction with XGBoost's ensemble technique improves performance in a range of natural language processing tasks [10].

7.3 Discussion

Tree based vs Linear Based Random Forest Classifier and XGBoost Classifier are the two tree based algorithms we have used and K-Nearest Neighbors, Multinomial Naive Bayes and Support Vector Classifier.

Tree-based algorithms like Decision Trees, Random Forests, and Gradient Boosting Machines are excellent for capturing intricate, non-linear correlations and interactions between features. They can effectively handle both numerical and categorical data and are resistant to outliers. However, they may need a substantial amount of processing power, especially for large datasets or ensemble approaches, and they can be prone to overfitting, especially with deep trees. Decision trees are interpretable, but ensemble techniques like Boosting and Random Forests are typically more intricate.

Linear methods, such as Linear Regression, Logistic Regression, and Support Vector Machines using linear kernels, are more straightforward and frequently faster to train and predict. They are extremely interpretable, with unambiguous coefficients reflecting the impact of each characteristic. These models assume linear relationships between characteristics and the target variable, which makes them ideal for issues that require such assumptions. Linear algorithms can perform well with high-dimensional data, especially when regularization techniques such as Ridge or Lasso are used to avoid overfitting. However, without extra feature engineering, they may struggle to capture non-linear patterns and interactions.

CHAPTER 8

Conclusion

The motivation behind this whole project is to create an awareness among common people regarding the plethora of harmful ingredients present in very commonly consumes food products. This is to ensure that they can make better choices, the next time they go for grocery shopping.

8.1 Model Training Inference

Despite the challenges, we were able to successfully harness the benefits of spacy library to develop our Custom Named Entity Recognition Model. We successfully did some Exploratory Data Analysis on the text data and the entity data of the various food ingredients. The results show that XGBoost Classifier (Accuracy 66%) is the best performing . From the above discussion, we can conclude, that tree based algorithms are better suited for text classification of food data. Random Forest Classifier also performs well before hyperparameter tuning. While we have also used Linear Algorithms, we are getting a comparatively lower accuracy even after hyperparameter tuning. The three linear algorithms that we have used give an accuracy less than 60 %

8.2 Limitations

Through the project our main goal was a fusion of NLP and AI in the ingredients' analysis of certain food products in the superstores in UK. SpaCy library along with DocBin package helps in training our Custom NER model and we could successfully train the CNER for all the three categories. The scope of our project was limited due to unavailability of a proper annotator website. There are various annotators which are much more efficient than what we

have used. Using those, will make our CNER model more robust and efficient. Currently, the number of ingredients we have used per category is relatively less. More the data that we can annotate, well trained our CNER model will be. Data Collection was another challenge faced during the the whole dissertation. As all the data was manually sourced from Tesco and Aldi websites and store, it became a labour intensive process. As with CNER, more the food products and their ingredients, better will be the quality of dataset.

8.3 Future Scope

A potential future scope for the project, can be adding the prices of food products into the dataframe and doing a thorough EDA from price point of view. A consumer needs a holistic view of food products, and price is one of the most important factors. So the integration of price segment will make the model training more efficient and hence, the model will be more robust Another step will be to collaborate with big companies and start-ups who are already doing similar work (TruthIN) and using their network along with Custom Named Entity Recognition to develop a model which can recommend people if they should buy a food product or not.

Bibliography

[1] Streaming Machine learning pipeline for Sentiment Analysis using Apache APIs: Kafka, Spark and Drill - Part 1 — developer.hpe.com. https://developer.hpe.com/blog/streaming-machine-learning-pipeline-for-sentiment-analysis-using-apache [Accessed 17-09-2024].

[2] &x201C;Label Padhega India" campaign aims to raise awareness on packaged foods — indiabusinesstrade.in. https://www.indiabusinesstrade.in/blogs/label-padhega-india-campaign-aims-to-raise-awareness-on-packaged-foods/ [Accessed 13-08-2024].

[3] Angela Bearth, Marie-Eve Cousin, and Michael Siegrist. The consumer's perception of artificial food additives: Influences on acceptance, risk and benefit perceptions. *Food quality and preference*, 38:14–23, 2014.

[4] Songül ÇAKMAKÇI and Mehmet Ali SALIK. Monosodium glutamate (msg) as a food additive and comments on its use. 2022.

[5] Archana Chaudhary, Savita Kolhe, and Raj Kamal. An improved random forest classifier for multi-class classification. *Information Processing in Agriculture*, 3(4):215–222, 2016.

[6] Shufeng Chen. K-nearest neighbor algorithm optimization in text categorization. In *IOP conference series: earth and environmental science*, volume 108, page 052074. IOP Publishing, 2018.

[7] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[8] KR1442 Chowdhary and KR Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020.

[9] Guanlan Hu, Mavra Ahmed, and Mary R L'Abbé. Natural language processing and machine learning approaches for food categorization and nutrition quality prediction compared with traditional methods. *The American Journal of Clinical Nutrition*, 117(3):553–563, 2023.

[10] Vipin Kumar and Basant Subba. A tfidfvectorizer and svm based sentiment analysis framework for text data corpus. In *2020 national conference on communications (NCC)*, pages 1–6. IEEE, 2020.

[11] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2022.

[12] Peihua Ma, Zhikun Zhang, Ying Li, Ning Yu, Jiping Sheng, Hande Küçük McGinty, Qin Wang, and Jaspreet KC Ahuja. Deep learning accurately predicts food categories and nutrients based on ingredient statements. *Food Chemistry*, 391:133243, 2022.

[13] Erik Millstone. Food additive regulation in the uk. *Food Policy*, 10(3):237–252, 1985.

[14] Deepak Moonat. Custom Named Entity Recognition using spaCy v3 — analyticsvidhya.com. https://www.analyticsvidhya.com/blog/2022/06/custom-named-entity-recognition-using-spacy-v3/. [Accessed 09-08-2024].

[15] Michael Moss. *Salt, sugar, fat: How the food giants hooked us*. Random House, 2013.

[16] Michael Moss. The extraordinary science of addictive junk food. In *Expanding addiction: Critical essays*, pages 127–140. Routledge, 2014.

[17] Anisha Patel, René Caldentey, and Srikanth Jagabathula. The indecisive shopper: Incorporating choice paralysis into the multinomial logit model, 2014.

[18] Saurabh Saoji, A Eqbal, and B Vidyapeeth. Text recognition and detection from images using pytesseract. *J Interdiscip Cycle Res*, 13:1674–1679, 2021.

[19] Rahul Sharnagat. Named entity recognition: A literature survey. *Center For Indian Language Technology*, pages 1–27, 2014.

[20] Hemlata Shelar, Gagandeep Kaur, Neha Heda, and Poorva Agrawal. Named entity recognition approaches and their comparison for custom ner model. *Science & Technology Libraries*, 39(3):324–337, 2020.

[21] Aixin Sun, Ee-Peng Lim, and Ying Liu. On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191–201, 2009.

[22] Songbo Tan. An effective refinement strategy for knn text classifier. *Expert Systems with Applications*, 30(2):290–298, 2006.

[23] Shuo Xu, Yan Li, and Zheng Wang. Bayesian multinomial naïve bayes classifier to text classification. In *Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11*, pages 347–352. Springer, 2017.

[24] Yudong Zhang, Lijia Deng, Hengde Zhu, Wei Wang, Zeyu Ren, Qinghua Zhou, Siyuan Lu, Shiting Sun, Ziquan Zhu, Juan Manuel Gorriz, et al. Deep learning in food category recognition. *Information Fusion*, 98:101859, 2023.