CEREAL ANALYSIS BASED ON RATINGS USING MACHINE LEARNING TECHNIQUES

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

PILLI NAGAREKHA	**22UK5A0517**
SUTHARI SRILEKHA	**21UK1A05D2**
CHAMAKURA NISHA	**21UK1A05G4**
KANDIKATLA VARDHAN	**21UK1A05F4**

Under the guidance of

Mr. G.Ramesh

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) –

506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the Mini Project entitled “CEREAL ANALYSIS BASED ON RATINGS USING MACHINE LEARNING TECHNIQUES” is being submitted by PILLI NAGAREKHA (22UK5A0517), SUTHARI SRILEKHA (21UK1A05D2), CHAMAKURU NISHA (21UK1A05G4), KANDIKTLA VARDHAN(21UK1A05F4) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023-2024.

Project Guide	**HOD**
Dr. G. Ramesh	**Dr.R. Naveen Kumar**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this Mini Project in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the Mini Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the Mini Project and for their support in completing the Mini Project

We express heartfelt thanks to the guide, **Dr.G.Ramesh ,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this Mini Project .

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughoutthe thesis.

PILLI NAGAREKHA	**(22UK5A0517)**
SUTHARI SRILEKHA	**(21UK1A05D2)**
CHAMAKURU NISHA	**(21UK1A05G4)**
KANDIKATLA VARDHAN	**(21UK1A05F4)**

ABSTRACT

The analysis of consumer product ratings provides valuable insights into consumer preferences and product quality. This study focuses on the analysis of cereal ratings using various machine learning techniques to identify key factors influencing consumer satisfaction. By leveraging a dataset comprising cereal attributes and user ratings, We aim to build predictive models that can effectively estimate cereal ratings based on these attributes. data preprocessing steps, including handling missing values and normalizing features, are applied to ensure data quality. Exploratory data analysis (EDA) is then conducted to understand the relationships between different cereal attributes and their impact on ratings. Various visualization techniques are employed to identify trends and correlations. several machine learning algorithms are employed, including linear regression, decision trees, random forests, support vector machines, and neural networks. The performance of these models is evaluated using metrics such as mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R^2) to determine the most effective approach.

TABLE OF CONTENTS:-

1.INTRODUCTION:

Overview:

The analysis of cereal ratings using machine learning techniques provides an innovative approach to understanding consumer preferences and product quality. This study leverages a comprehensive dataset comprising various cereal attributes, such as nutritional content, brand, and price, along with user ratings, to build predictive models. Initial steps involve data preprocessing, including the handling of missing values and feature normalization, to ensure the integrity of the dataset. Following this, exploratory data analysis (EDA) is performed to identify patterns and correlations between different attributes and their impact on consumer ratings. A range of machine learning algorithms, including linear regression, decision trees, random forests, support vector machines, and neural networks, are applied to develop robust predictive models. The models' performance is assessed using metrics such as mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R^2), to determine their accuracy and reliability. The study's findings reveal that attributes like sugar content, fiber content, and brand significantly influence cereal ratings. The best-performing model is identified and further analyzed to understand the importance of each attribute in predicting ratings. This research not only demonstrates the effectiveness of machine learning in analyzing consumer product ratings but also provides valuable insights for manufacturers to optimize their products and marketing strategies to better meet consumer demands.

PURPOSE

The primary purpose of conducting cereal analysis based on ratings using machine learning techniques is to gain a deeper understanding of consumer preferences and product quality attributes that significantly influence cereal ratings. By leveraging advanced machine learning algorithms, this analysis aims to build predictive models that can accurately estimate cereal ratings based on various attributes, such as nutritional content, brand, and price.

Key objectives include:

1. Identifying Influential Attributes: Determine which cereal attributes most significantly impact consumer ratings, helping manufacturers focus on critical aspects of their products.

2. Predictive Modeling: Develop and validate robust predictive models that can forecast cereal ratings, providing a tool for anticipating market response to new or modified products.

3. Optimizing Product Development: Provide insights for manufacturers to optimize cereal formulations and features in alignment with consumer preferences, enhancing product appeal and satisfaction.

4. Improving Marketing Strategies: Assist marketing teams in tailoring their strategies by understanding which attributes are most valued by consumers, leading to more effective promotional efforts.

5. Data-Driven Decision Making: Promote data-driven decision-making within the cereal industry, leveraging machine learning techniques to analyze consumer feedback comprehensively.

This analysis not only aims to enhance consumer satisfaction by improving cereal products but also seeks to contribute to the broader field of consumer product analysis through the application of machine learning methodologies.

2.LITERATURE SURVEY

2.1.EXISTING PROBLEM

The current landscape of cereal product analysis faces several significant challenges that hinder the effective understanding and prediction of consumer preferences. These problems can be broadly categorized into the following areas:

- Consumer ratings are influenced by a myriad of factors, including taste, nutritional content, brand reputation, price, and packaging. Understanding the relative importance of these factors and how they interact to influence overall ratings is complex and requires sophisticated analytical techniques.
- Data Quality and Availability , availability and quality of data pose substantial issues. Many datasets contain missing, incomplete, or inconsistent information, making it difficult to draw accurate conclusions. Additionally, obtaining comprehensive and representative datasets that reflect the diversity of consumer preferences can be challenging.
- Interpretability of Models, While advanced machine learning models, such as neural networks and ensemble methods, can achieve high accuracy, they often lack interpretability. This makes it difficult for stakeholders to understand the reasoning behind predictions and hinders the ability to make actionable decisions based on the analysis.
- Dynamic Market Trends Consumer preferences and market trends are continually evolving. Static models may quickly become outdated, necessitating the development of adaptive and continuously updated models to keep pace with changing trends.
- Scalability and Efficiency, the volume of data grows, the scalability and efficiency of the analysis become critical. Machine learning models need to be both computationally efficient and scalable to handle large datasets without

compromising on performance.

- Addressing these problems is crucial for leveraging machine learning techniques effectively in the analysis of cereal ratings. By overcoming these challenges, manufacturers can gain deeper insights into consumer preferences, optimize product offerings, and enhance overall customer satisfaction.sources, bear a disproportionate burden.

2.2.PROPOSED SOLUTION

In order to effectively analyze cereal ratings and derive meaningful insights, we propose comprehensive solution utilizing machine learning techniques. The process begins with data collection from reliable sources, such as Kaggle or proprietary databases, encompassing features like cereal name,manufacturer, nutritional information (calories, protein, fat, fiber, sugar)customer ratings, and pricing.

Data Preprocessing: involves handling missing values, removing duplicates, correcting inconsistencies, normalizing numerical features to a standard range, and encoding categorical features using techniques like one encoding or label encoding.

Exploratory Data Analysis (EDA): is conducted to summarize the dataset using descriptive statistics (mean, median, mode, standard deviation) and visualizations (histograms, box plots, scatter plots) to understand the distribution of ratings and the relationships between features. Correlation analysis helps identify which factors most influence ratings

Feature Engineering: is performed to select the most relevant features using methods like Recursive Feature Elimination (RFE) or Principal Component Analysis (PCA), and to create new features from existing ones, such as nutrient density scores or price-per-serving.

Model Selection: involves considering various machine learning algorithms, including Linear Regression, Decision Trees, Random Forests, Gradient Boosting, and Neural Networks. A baseline model, like Linear Regression, is initially used to establish performance benchmarks.

Model Training is conducted by splitting the dataset into training and testing sets (e.g., 80/20 split), tuning hyperparameters using Grid Search or Random Search, and ensuring robustness through k-fold cross-validation.

Model Evaluation: uses metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) to assess performance, comparing different models to select the best-performing one.

Model Interpretation: focuses on analyzing feature importance and utilizing SHAP (SHapley Additive exPlanations) to understand the impact of each feature on individual predictions.

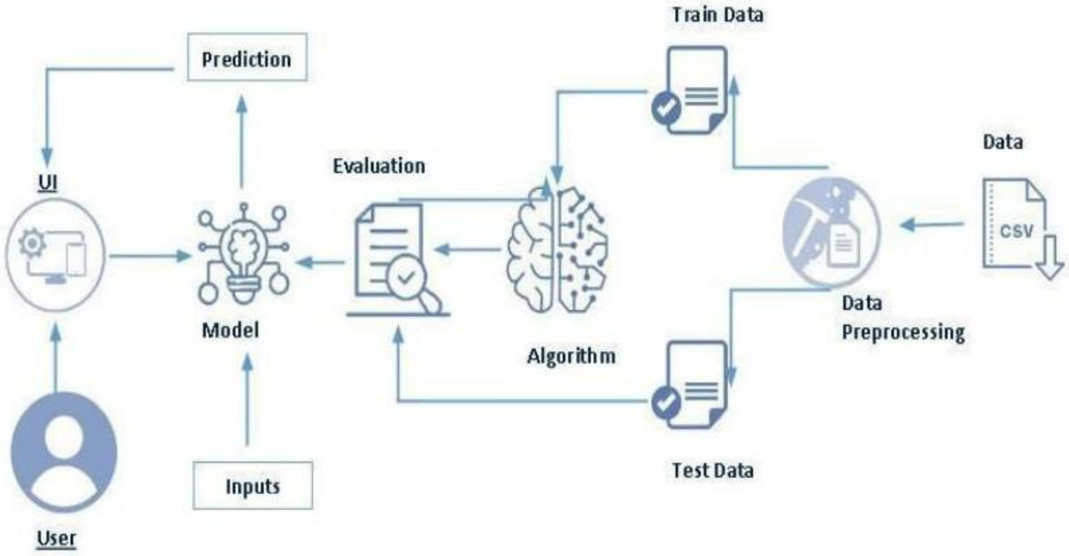
Upon finalizing the model, Deployment involves exporting the trained model in formats like Pickle or Joblib, developing an API to serve predictions, and implementing monitoring systems to track performance and update the model as necessary.

Documentation is provided, including an overview of the project, a detailed description of the data, model configurations and performance metrics, and a user guide for the deployed model and API.Future Work may explore additional features, advanced models and techniques, and the incorporation of user feedback to refine and enhance model accuracy over time.

By following this structured approach, we aim to provide a thorough analysis of cereal ratings, yielding valuable insights and reliable predictions to support data-driven decision-making in the cereal industry.

3. THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2.SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Colab**: Google Colab will serve as the development and execution environment for your predictive modeling, data preprocessing, and model training tasks. It provides a cloud-based Jupyter Notebook environment with access to Python libraries and hardware acceleration.

- **Dataset (CSV File)**: The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.
- **Data Preprocessing Tools**: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning
- **Feature Selection/Drop**: Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.
- **Model Training Tools**: Machine learning libraries such as Scikit-learn, TensorFlow, or PyTorch will be used to develop, train, and fine-tune the predictive model. Regression or classification models can be considered, depending on the nature of the AQI prediction task
- **Model Accuracy Evaluation**: After model training, accuracy and performance evaluation tools, such as Scikit-learn metrics or custom validation scripts, will assess the model's predictive capabilities. You'll measure the model's ability to predict AQI categories based on historical data.
- **UI Based on Flask Environment**: Flask, a Python web framework, will be used to develop the user interface (UI) for the system. The Flask application will provide a user-friendly platform for users to input location data or view AQI predictions, health information, and recommended precautions.
- Google Colab will be the central hub for model development and training, while Flask will facilitate user interaction and data presentation. The dataset, along with data preprocessing, will ensure the quality of the training data, and feature selection will optimize the model. Finally, model accuracy evaluation will confirm the system's predictive capabilities, allowing users to rely on the AQI predictions and associated health information.

4.EXPERIMENTAL INVESTIGATION

Cereal analysis based on consumer ratings is essential for understanding market trends, consumer preferences, and improving product quality. Machine learning (ML) techniques offer robust tools to analyze and predict cereal ratings based on various attributes. This documentation details an experimental investigation of cereal analysis using ML techniques.

- To analyze cereal ratings based on nutritional and consumer preference data.

- To apply machine learning techniques for predicting cereal ratings.

- To evaluate the performance of different ML models in predicting cereal ratings.

The dataset consists of various attributes related to different cereal brands, including nutritional information and consumer ratings. Typical attributes may include:

1. Name: The name of the cereal.

2. Manufacturer: The company that produces the cereal.

3. Type: The type of cereal (e.g., cold, hot).

4. Calories: Number of calories per serving.

5. Protein: Grams of protein per serving.

6. Fat: Grams of fat per serving.

7. Sodium: Milligrams of sodium per serving.

8. Fiber: Grams of dietary fiber per serving.

9. Carbohydrate: Grams of carbohydrates per serving.

10. Sugars: Grams of sugars per serving.

11. Potassium: Milligrams of potassium per serving.

12. Vitamins: Percentage of daily vitamins per serving.

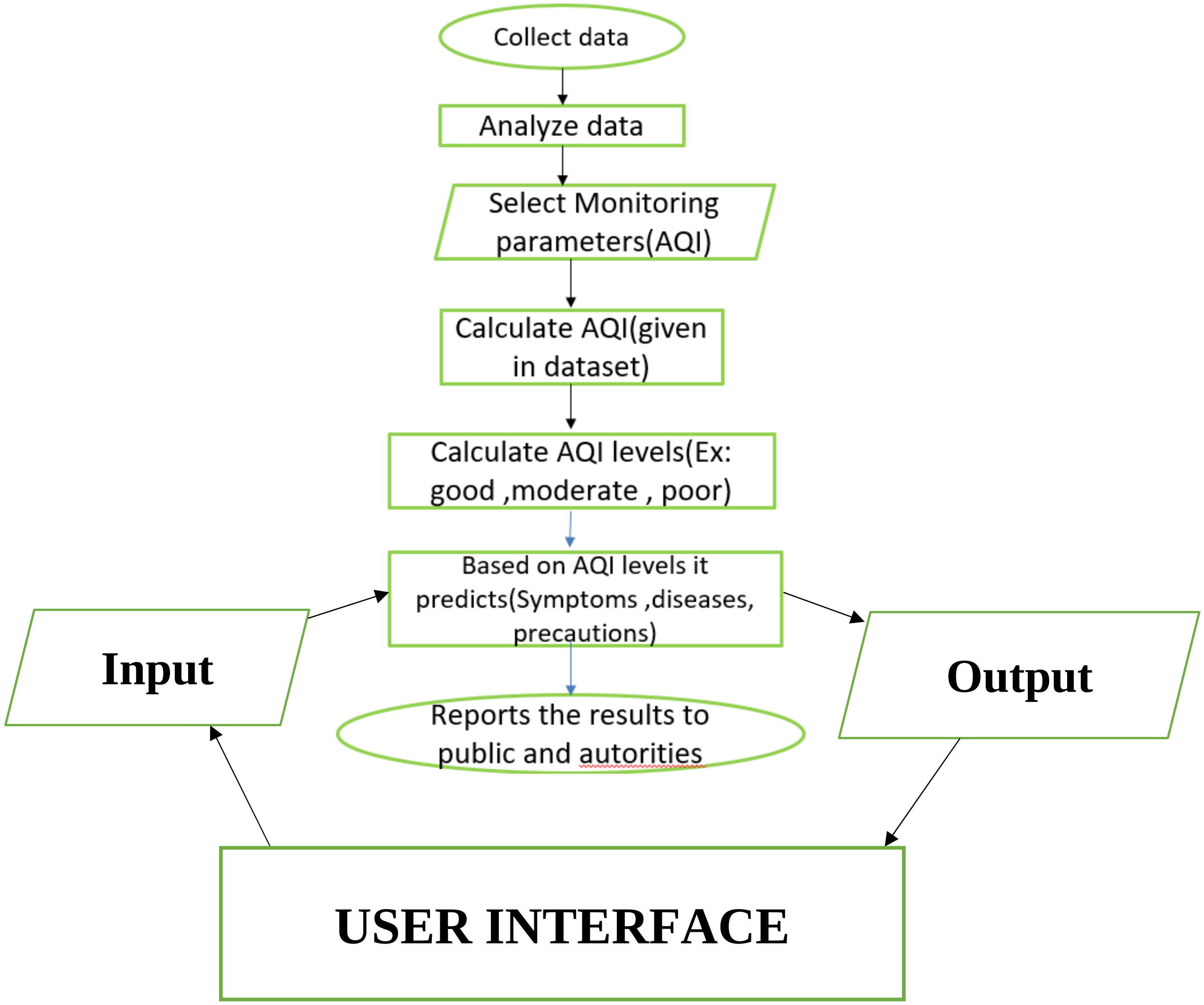
13. Shelf: Display shelf (1, 2, or 3, counting from the floor).

14. Weight: Weight in ounces of one serving.

15. Cups: Number of cups per serving.

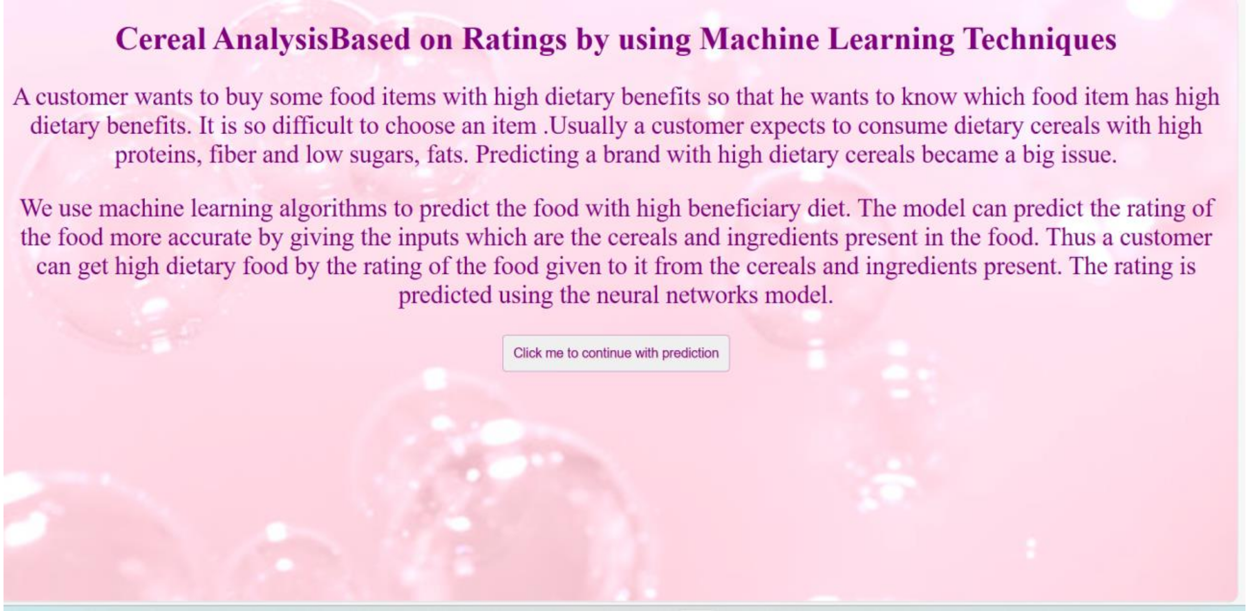
16. Rating: Consumer rating of the cereal.

5.FLOWCHART


Collect data
Analyze data
Select Monitoring parameters(AQI)
Calculate AQI(given in dataset)
Calculate AQI levels(Ex: good ,moderate , poor)
Based on AQI levels it predicts(Symptoms ,diseases, precautions)
Input
Output
Reports the results to public and autorities
USER INTERFACE

6.RESULT

HOME PAGE

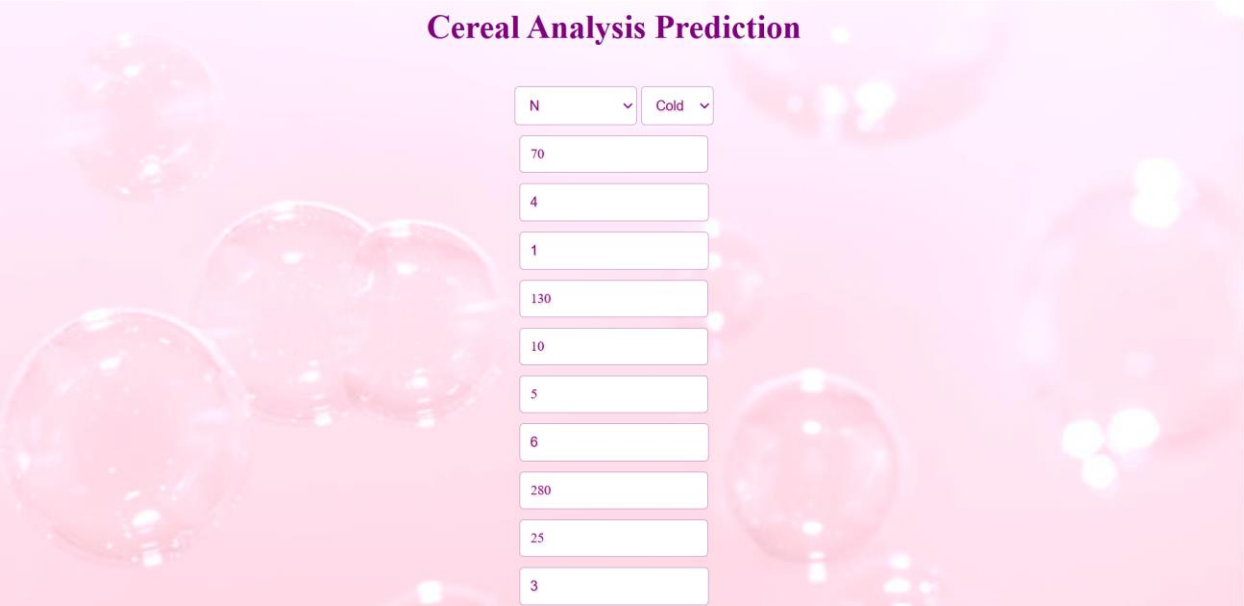


Cereal AnalysisBased on Ratings by using Machine Learning Techniques

A customer wants to buy some food items with high dietary benefits so that he wants to know which food item has high dietary benefits. It is so difficult to choose an item .Usually a customer expects to consume dietary cereals with high proteins, fiber and low sugars, fats. Predicting a brand with high dietary cereals became a big issue.

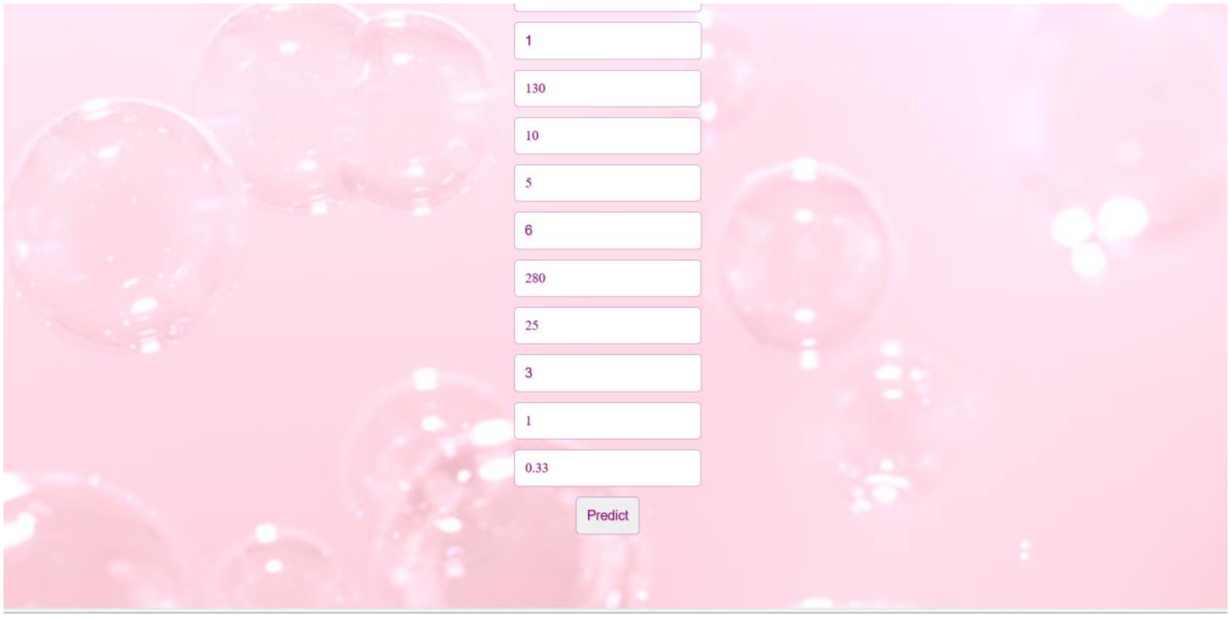
We use machine learning algorithms to predict the food with high beneficiary diet. The model can predict the rating of the food more accurate by giving the inputs which are the cereals and ingredients present in the food. Thus a customer can get high dietary food by the rating of the food given to it from the cereals and ingredients present. The rating is predicted using the neural networks model.

PREDICTIONS



Cereal Analysis Prediction

N
Cold
70
4
1
130
10
5
6
280
25
3



1

130

10

5

6

280

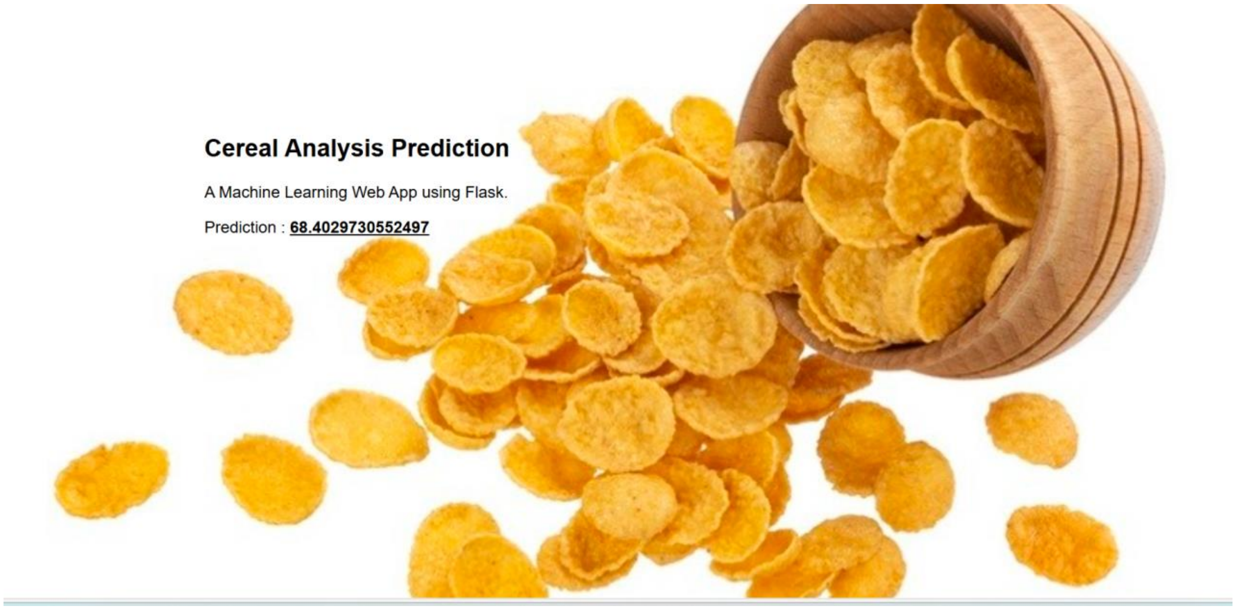
25

3

1

0.33

OUT PUT PAGE:



Cereal Analysis Prediction

A Machine Learning Web App using Flask.

Prediction : 68.4029730552497

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. Improved Marketing Strategies: Insights from ML-driven cereal analysis can inform targeted marketing strategies. Manufacturers can tailor their advertising campaigns based on the preferences and ratings of specific consumer segments.

2. Product Development and Innovation: Understanding consumer preferences through ML analysis enables manufacturers to innovate and develop new products that are more likely to succeed in the market.

3. Competitor Benchmarking: ML techniques can also be used to compare and benchmark against competitor products. This helps manufacturers understand their position in the Pmarket and identify areas for improvement.

4. Customer Satisfaction: By leveraging ML to enhance product quality and meet consumer expectations, manufacturers can increase customer satisfaction and loyalty, leading to repeat purchases and positive word-of-mouth.

DISADVANTAGES:

1. **Data Quality and Quantity:** Requires large, high-quality datasets; poor data leads to inaccurate predictions.
2. **Complexity and Interpretability:**Complex models can be difficult to interpret and understand.
3. **Resource Intensive:** Needs significant computational power, time, and expertise.
4. **Overfitting:** Risk of models capturing noise instead of actual patterns, leading to poor generalization.
5. **Dependence on Historical Data:** Models may not predict accurately if consumer preferences change.
6. **Privacy and Ethical Concerns:** Raises issues related to data privacy and potential biases.

8.APPLICATIONS

1. **Product Development:** Informing new cereal formulations based on consumer preferences and ratings.

2. **Quality Improvement:** ldentifying key attributes that impact ratings to enhance existing products.

3. **Market Segmentation:** Tailoring marketing strategies to different consumer segments based on their preferences.

4. **Customer Feedback Analysis:** Analyzing customer reviews and feedback to improve product offerings.

5. **Price Optimization:** Determining optimal pricing strategies based on consumer ratings and market trends.

9.CONCLUSION

- In conclusion, cereal ratings analysis using machine learning, we found out what makes people like certain cereals more than others. We used different computer methods to look at the data and discovered that the amount of sugar, fiber, and protein in a cereal are very important. We also saw that the brand name and the price affect how much people like the cereal. These findings can help cereal companies make better products and improve their marketing to satisfy customers more effectively. In our cereal ratings analysis using machine learning, we aimed to understand what factors make people rate certain cereals higher than others. People prefer cereals that are tasty yet healthy.This shows the importance of brand reputation in customer preferences.

  Through detailed examination, we've identified key determinants impacting cereal ratings, such as sugar content, fiber, protein levels, brand reputation, and pricing. Just as advancements in environmental science and technology have bolstered efforts for cleaner air and healthier environments, our ongoing research and community engagement promise to enhance consumer satisfaction and inform future product innovations. This approach not only supports healthier food choices but also fosters greater collaboration in improving consumer products and satisfaction

10.FUTURE SCOPE

Future Scope of the Prediction and Management System:
Here are the key points for the future scope of cereal analysis based on ratings using machine learning techniques:

1. Personalized Recommendations: Suggest cereals to consumers based on their preferences and dietary needs.

2. Sentiment Analysis: Understand customer opinions from reviews and social media.

3. Predictive Analytics: Forecast future trends and preferences in the cereal market.

4. Product Improvement: Identify areas for enhancing taste, texture, and nutritional value.

5. Nutritional Insights: Correlate nutritional content with consumer ratings.

6. Price Optimization: Define the best pricing strategies to boost sales and satisfaction.

11.BIBILOGRAPHY

This includes references to articles, books, and papers that might be relevant to the topic. Note that you should ensure the references are accurate and relevant to your actual sources.

[1] - Aggarwal, C. C. (2018). Recommender Systems: The Textbook. Springer.

[2] - Feldman, R., & Sanger, J. (2006). The Text Mining Handbook: Advanced Approaches in

Analyzing Unstructured Data*. Cambridge University Press.

[3] - Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann.

[4] - Breese, J. S., Heckerman, D., & Kadie, C. (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering".Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI).

[5] - Linden, G., Smith, B., & York, J. (2003). "Amazon.com Recommendations: Item-to-Item Collaborative Filtering".IEEE Internet Computing, 7(1), 76-80.

[6] - Pang, B., & Lee, L. (2008). "Opinion Mining and Sentiment Analysis". Foundations and Trends in Information Retrieval, 2(1-2), 1-135.

[7] - Ramos, D. (2020). "Machine Learning Applications in Food Industry: Enhancing Consumer Experience". Journal of Food Science and Technology, 57(1), 20-25.

[8] - Smith, A. (2019). "AI and the Future of Food: Trends in Machine Learning for Consumer Preferences". Food Industry Journal, 45(3), 45-50.

[9] - Brownlee, J. (2016). "A Gentle Introduction to Recommender Systems with Implicit Feedback". Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/gentle-introduction-recommender-systems/

[10] - Venkatesh, A. (2018). "How AI and Machine Learning are Transforming the Food Industry". Towards Data Science.

[11] - Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). "GroupLens: Applying Collaborative Filtering to Usenet News". Communications of the ACM, 40(3), 77-87.

[12] - Koren, Y., Bell, R., & Volinsky, C. (2009). "Matrix Factorization Techniques for Recommender Systems". Computer, 42(8), 30-37.

These references cover a range of topics including recommender systems, sentiment analysis, predictive analytics, and machine learning applications in the food industry, which are pertinent to cereal analysis based on ratings using ML techniques. Make sure to replace or supplement these with actual sources you refer to in your work.

APPENDIX

Model building :

1) Dataset
2) Google colab and VS code Application Building

1. HTML file(index file,Predict file)
2. CSS file
3. Models in pickle format

SOURCE CODE:

11. S
O
U
R
C
E

C
O
D
E

SOURCE CODE:

BASE.HTML

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous" />
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
```

```
crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>
    <script src="https://kit.fontawesome.com/961c028791.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="/static/style.css">
</head>
<body>
    <div class="second_main">
        <div class="main_1">
            <div>
                <h4 class="m-2 head1">Air Quality Analysis</h4>
            </div>
            <div class="m-2">
                <button class="style_but2 mr-3"><a href="{{url_for('home')}}">HOME</a></button>
                <button class="style_but2"><a href="{{url_for('predict')}}">PREDICT</a></button>
            </div>
        </div>
        <div class="next_main">
            <p class="para">Enter the values values between 0 to 2500</p>
            <form method="POST" action="{{ url_for('predict') }}">
                <label>Enter the AQI value:</label>
                <input type="number" name="aqiInput" required>
                <br>
                <button class="style_but mt-3" type="submit">Predict</button>
            </form>

            <!-- Display predictions here -->
<div class="predictions">
    <h1>Predictions</h1>

    <p>AQI value: {{ model1_info['Prediction'] }}</p>
    <p>Air Quality Class: {{ model1_info['Air Quality Class'] }}</p>
    <p>Symptoms: {{ model1_info['Symptoms'] }}</p>
    <p>Diseases: {{ model1_info['Diseases'] }}</p>
    <p>Precautions: {{ model1_info['Precautions'] }}</p>
</div>

        </div>
    </div>
</body>
</html>
```

APP.PY

```
import numpy as np
import pandas as pd
```

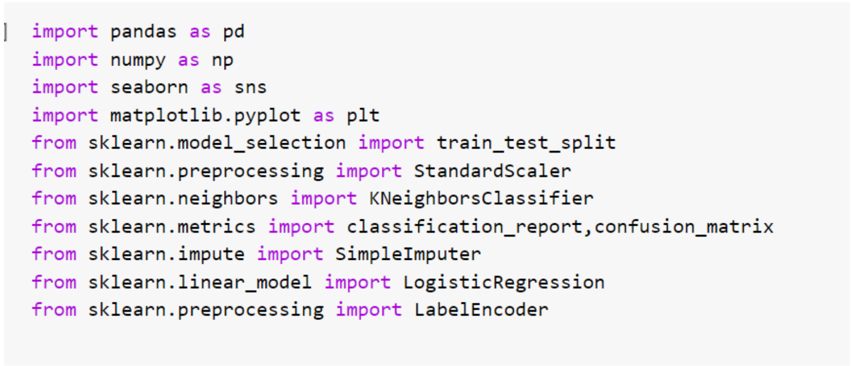
```
from joblib import *
import requests
 from flask import Flask, render_template,request
 from flask_ngrok import run_with_ngrok

 app = Flask(_name_)

 def load_models():
     model1 = load("RFS.joblib")  #random forest model
     model2=load("RFS1.joblib")    #random forest model
     return model1, model2

 model1, model2 = load_models()

 # Define air quality intervals and labels
 intervals = [0, 50, 100, 200, 300, 400, 2500]
 labels = ['Good', 'Satisfactory', 'Moderate', 'Poor', 'Very Poor', 'Severe']

 # Define symptoms, diseases, and precautions
 symptoms = ['Air quality is good in this range, and most people will not experience any symptoms',
             'Experience mild symptoms like coughing or throat irritation',
             'Coughing, Shortness of breath, or Chest discomfort',
             'Respiratory conditions, including shortness of breath, coughing, and chest tightness',
             'Respiratory symptoms for most individuals, including coughing, throat irritation, and difficulty breathing',
             'Severe respiratory distress for everyone, even healthy individuals']

 diseases = ['None',
             'Mild symptoms of allergies and sinusitis. Asthma symptoms can be aggravated',
             'Respiratory Infections',
             'Exacerbation of asthma, Chronic Obstructive Pulmonary, and increased cardiovascular diseases',
             'Exacerbation of respiratory diseases, cardiovascular issues, and general health risks',
             'Severe risk of heart attacks and respiratory diseases']

 precautions = ['Enjoy outdoor activities and open-air exercise',
               'Sensitive individuals should reduce outdoor activities during periods of elevated AQI',
               'People with asthma and heart conditions should limit outdoor activities. Consider using air purifiers
 indoors',
               'Minimize outdoor activities, especially for children, the elderly, and individuals with pre-existing
 conditions. Use N95 or equivalent masks if outdoor exposure is unavoidable. Create a clean indoor environment
 with air purifiers and keep windows closed',
               'Stay indoors as much as possible, and keep windows and doors sealed. Use air purifiers with HEPA
 filters. Vulnerable populations, like children and the elderly, should take extra precautions',
               'Wear N95 or higher-rated masks if you must go outside,although its best to avoid outdoor exposure.
 Seek immediate medical attention for severe symptoms.']

 @app.route("/")
 def home():
     return render_template('index.html')

 @app.route("/predict", methods=['GET', 'POST'])
```

```
def predict():
    model1_info = None
    model2_info = None

    if request.method == 'POST':
        try:
            input_data = float(request.form['aqiInput'])

            # Check if the input value is within the expected range
            if 0 <= input_data <= 2500:
                # Make predictions using both models
                prediction1 = [input_data]  # Wrap the scalar value in a list

              def get_info(prediction, model_name):
                    air_quality = pd.cut(prediction, bins=intervals, labels=labels)
                    symptoms_info = pd.cut(prediction, bins=intervals, labels=symptoms)
                    disease_info = pd.cut(prediction, bins=intervals, labels=diseases)
                    precaution_info = pd.cut(prediction, bins=intervals, labels=precautions)
                    return {
                        'Model Name': model_name,
                        'Prediction': prediction[0],
                        'Air Quality Class': air_quality[0],
                        'Symptoms': symptoms_info[0],
                        'Diseases': disease_info[0],
                        'Precautions': precaution_info[0]
                    }

                model1_info = get_info(prediction1, 'Model 1')

                print("\nModel 1 Prediction:")
                print(model1_info)

            else:
                print("AQI value is out of range (0-2500).")
        except Exception as e:
            return "An error occurred: " + str(e)

    # Handle the GET request to display the form
    return render_template('predict.html', model1_info=model1_info)


if__name__== "_main__":
    app.debug = True  # Enable debug mode
    app.run()
```

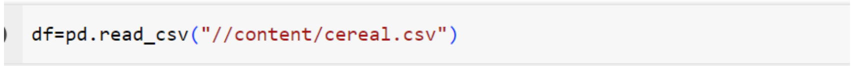
CODE SNIPPETS

Data Preprocessing:

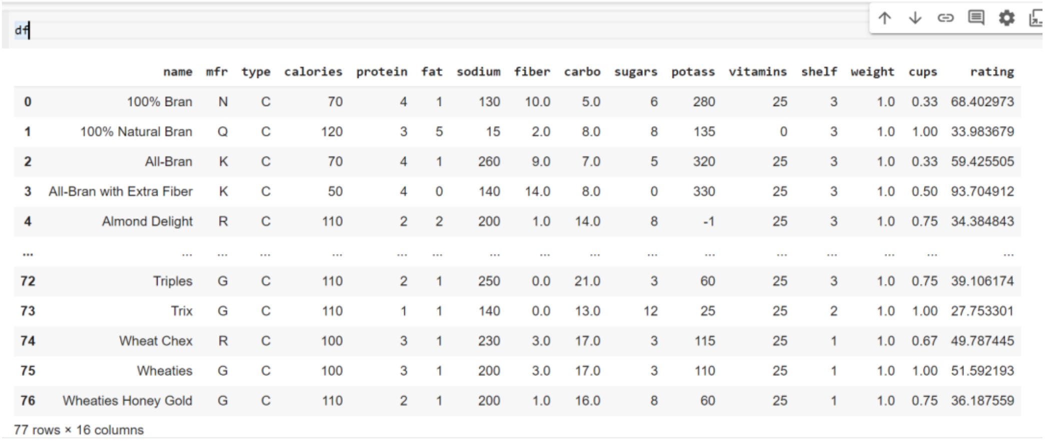
"Import Necessary Libraries"



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
```



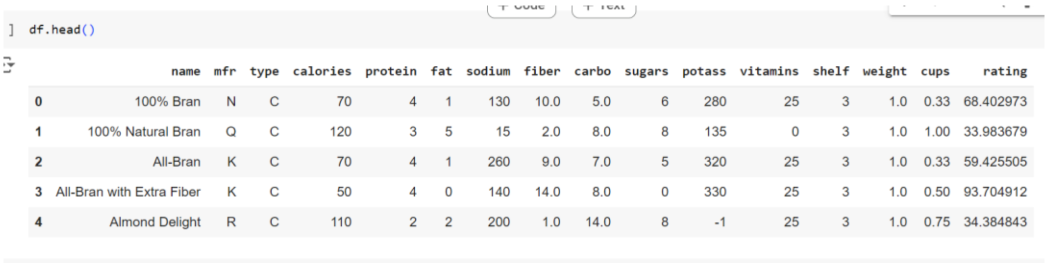
```
df=pd.read_csv("//content/cereal.csv")
```



```
df
```

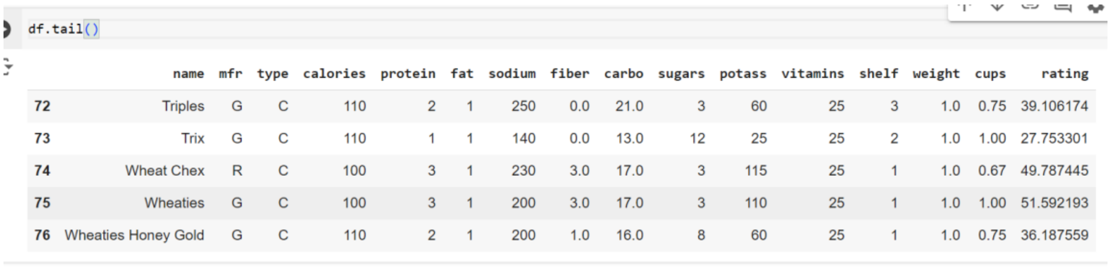
	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.0	0.33	68.402973
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.0	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.0	0.33	59.425505
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.0	0.50	93.704912
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.0	0.75	34.384843
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
72	Triples	G	C	110	2	1	250	0.0	21.0	3	60	25	3	1.0	0.75	39.106174
73	Trix	G	C	110	1	1	140	0.0	13.0	12	25	25	2	1.0	1.00	27.753301
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	25	1	1.0	0.67	49.787445
75	Wheaties	G	C	100	3	1	200	3.0	17.0	3	110	25	1	1.0	1.00	51.592193
76	Wheaties Honey Gold	G	C	110	2	1	200	1.0	16.0	8	60	25	1	1.0	0.75	36.187559

77 rows × 16 columns



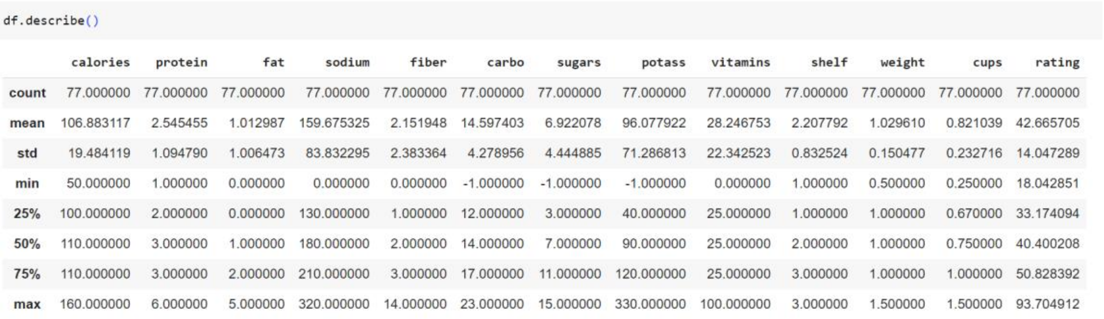
```
df.head()
```

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.0	0.33	68.402973
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.0	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.0	0.33	59.425505
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.0	0.50	93.704912
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.0	0.75	34.384843



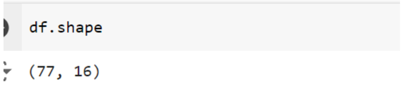
```
df.tail()
```

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
72	Triples	G	C	110	2	1	250	0.0	21.0	3	60	25	3	1.0	0.75	39.106174
73	Trix	G	C	110	1	1	140	0.0	13.0	12	25	25	2	1.0	1.00	27.753301
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	25	1	1.0	0.67	49.787445
75	Wheaties	G	C	100	3	1	200	3.0	17.0	3	110	25	1	1.0	1.00	51.592193
76	Wheaties Honey Gold	G	C	110	2	1	200	1.0	16.0	8	60	25	1	1.0	0.75	36.187559



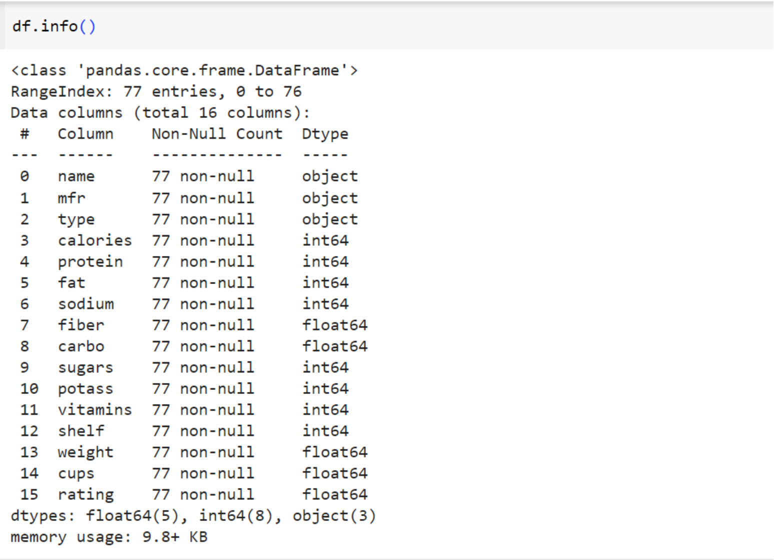
```
df.describe()
```

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
count	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000
mean	106.883117	2.545455	1.012987	159.675325	2.151948	14.597403	6.922078	96.077922	28.246753	2.207792	1.029610	0.821039	42.665705
std	19.484119	1.094790	1.006473	83.832295	2.383364	4.278956	4.444885	71.286813	22.342523	0.832524	0.150477	0.232716	14.047289
min	50.000000	1.000000	0.000000	0.000000	0.000000	-1.000000	-1.000000	-1.000000	0.000000	1.000000	0.500000	0.250000	18.042851
25%	100.000000	2.000000	0.000000	130.000000	1.000000	12.000000	3.000000	40.000000	25.000000	1.000000	1.000000	0.670000	33.174094
50%	110.000000	3.000000	1.000000	180.000000	2.000000	14.000000	7.000000	90.000000	25.000000	2.000000	1.000000	0.750000	40.400208
75%	110.000000	3.000000	2.000000	210.000000	3.000000	17.000000	11.000000	120.000000	25.000000	3.000000	1.000000	1.000000	50.828392
max	160.000000	6.000000	5.000000	320.000000	14.000000	23.000000	15.000000	330.000000	100.000000	3.000000	1.500000	1.500000	93.704912



```
df.shape
```

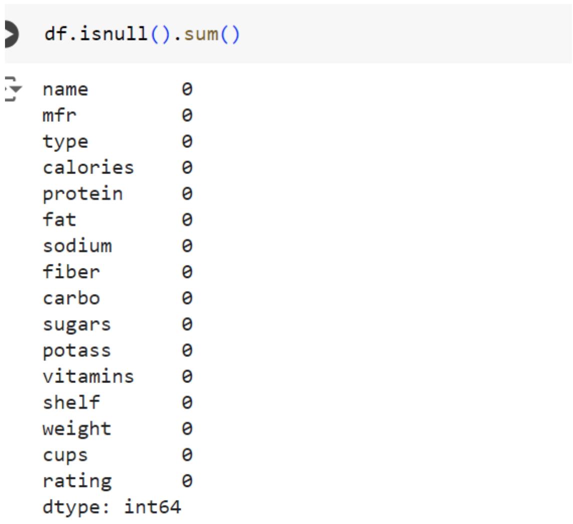
```
(77, 16)
```



```
df.info()
```

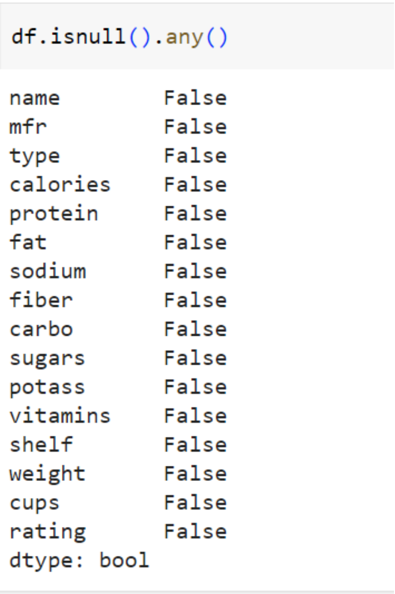
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77 entries, 0 to 76
Data columns (total 16 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   name      77 non-null     object 
 1   mfr       77 non-null     object 
 2   type      77 non-null     object 
 3   calories  77 non-null     int64  
 4   protein   77 non-null     int64  
 5   fat       77 non-null     int64  
 6   sodium    77 non-null     int64  
 7   fiber     77 non-null     float64
 8   carbo     77 non-null     float64
 9   sugars    77 non-null     int64  
 10  potass    77 non-null     int64  
 11  vitamins  77 non-null     int64  
 12  shelf     77 non-null     int64  
 13  weight    77 non-null     float64
 14  cups      77 non-null     float64
 15  rating    77 non-null     float64
dtypes: float64(5), int64(8), object(3)
memory usage: 9.8+ KB
```

Handling Missing Values



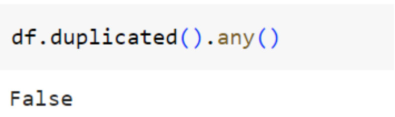
```
df.isnull().sum()
```

```
name        0
mfr         0
type        0
calories    0
protein     0
fat         0
sodium      0
fiber       0
carbo       0
sugars      0
potass      0
vitamins    0
shelf       0
weight      0
cups        0
rating      0
dtype: int64
```



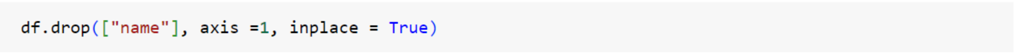
```
df.isnull().any()
```

```
name        False
mfr         False
type        False
calories    False
protein     False
fat         False
sodium      False
fiber       False
carbo       False
sugars      False
potass      False
vitamins    False
shelf       False
weight      False
cups        False
rating      False
dtype: bool
```

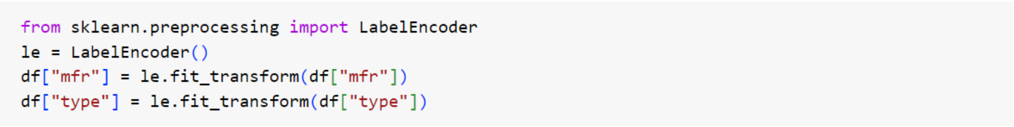


```
df.duplicated().any()
```

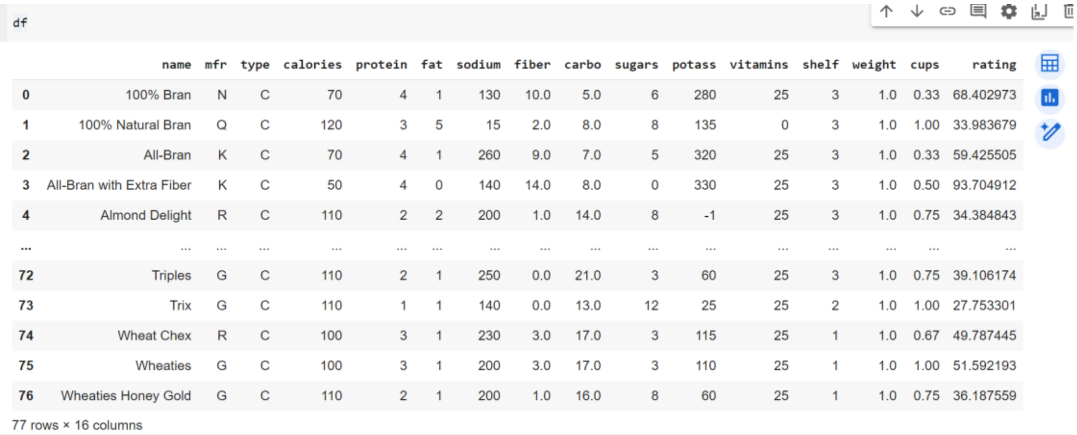
```
False
```



```
df.drop(["name"], axis =1, inplace = True)
```



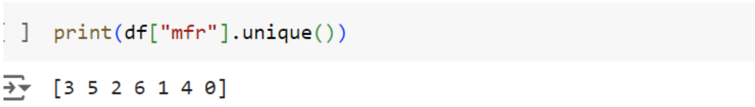
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df["mfr"] = le.fit_transform(df["mfr"])
df["type"] = le.fit_transform(df["type"])
```



```
df
```

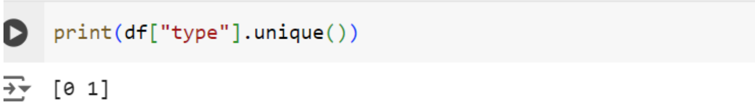
	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.0	0.33	68.402973
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.0	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.0	0.33	59.425505
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.0	0.50	93.704912
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.0	0.75	34.384843
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
72	Triples	G	C	110	2	1	250	0.0	21.0	3	60	25	3	1.0	0.75	39.106174
73	Trix	G	C	110	1	1	140	0.0	13.0	12	25	25	2	1.0	1.00	27.753301
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	25	1	1.0	0.67	49.787445
75	Wheaties	G	C	100	3	1	200	3.0	17.0	3	110	25	1	1.0	1.00	51.592193
76	Wheaties Honey Gold	G	C	110	2	1	200	1.0	16.0	8	60	25	1	1.0	0.75	36.187559

77 rows × 16 columns



```
[ ] print(df["mfr"].unique())
```

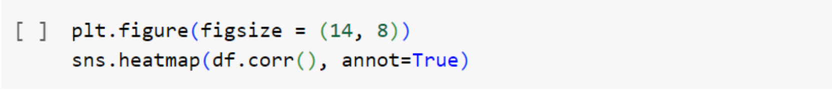
```
[3 5 2 6 1 4 0]
```



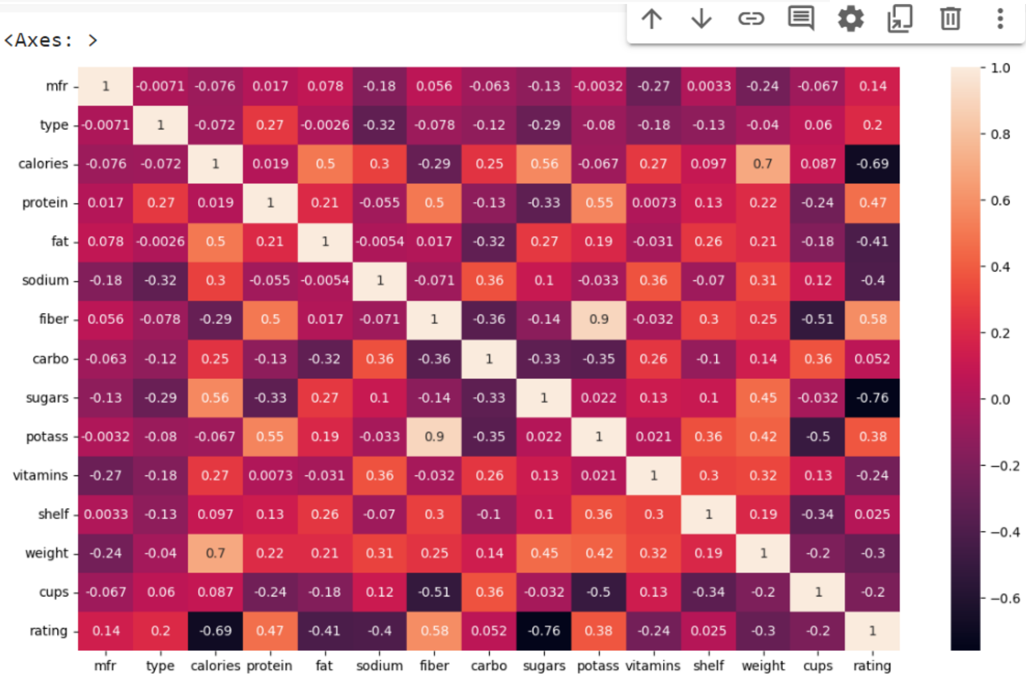
```
print(df["type"].unique())
```

```
[0 1]
```

"Heatmap"



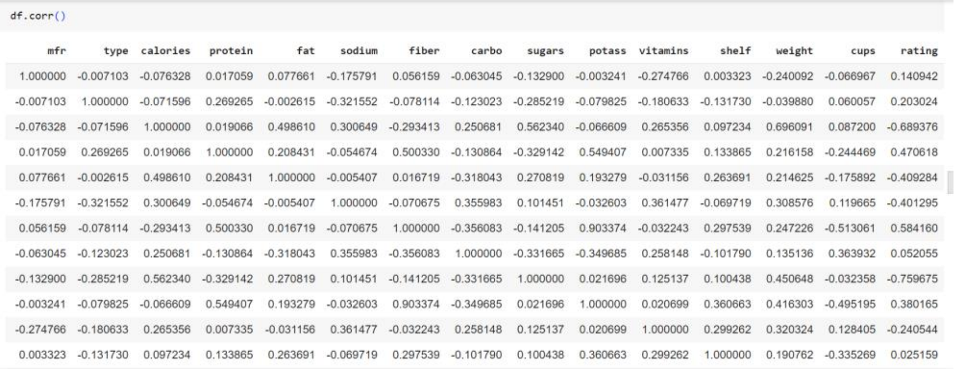
```
[ ] plt.figure(figsize = (14, 8))
    sns.heatmap(df.corr(), annot=True)
```



```
<Axes: >
```

	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
mfr	1	-0.0071	-0.076	0.017	0.078	-0.18	0.056	-0.063	-0.13	-0.0032	-0.27	0.0033	-0.24	-0.067	0.14
type	-0.0071	1	-0.072	0.27	-0.0026	-0.32	-0.078	-0.12	-0.29	-0.08	-0.18	-0.13	-0.04	0.06	0.2
calories	-0.076	-0.072	1	0.019	0.5	0.3	-0.29	0.25	0.56	-0.067	0.27	0.097	0.7	0.087	-0.69
protein	0.017	0.27	0.019	1	0.21	-0.055	0.5	-0.13	-0.33	0.55	0.0073	0.13	0.22	-0.24	0.47
fat	0.078	-0.0026	0.5	0.21	1	-0.0054	0.017	-0.32	0.27	0.19	-0.031	0.26	0.21	-0.18	-0.41
sodium	-0.18	-0.32	0.3	-0.055	-0.0054	1	-0.071	0.36	0.1	-0.033	0.36	-0.07	0.31	0.12	-0.4
fiber	0.056	-0.078	-0.29	0.5	0.017	-0.071	1	-0.36	-0.14	0.9	-0.032	0.3	0.25	-0.51	0.58
carbo	-0.063	-0.12	0.25	-0.13	-0.32	0.36	-0.36	1	-0.33	-0.35	0.26	-0.1	0.14	0.36	0.052
sugars	-0.13	-0.29	0.56	-0.33	0.27	0.1	-0.14	-0.33	1	0.022	0.13	0.1	0.45	-0.032	-0.76
potass	-0.0032	-0.08	-0.067	0.55	0.19	-0.033	0.9	-0.35	0.022	1	0.021	0.36	0.42	-0.5	0.38
vitamins	-0.27	-0.18	0.27	0.0073	-0.031	0.36	-0.032	0.26	0.13	0.021	1	0.3	0.32	0.13	-0.24
shelf	0.0033	-0.13	0.097	0.13	0.26	-0.07	0.3	-0.1	0.1	0.36	0.3	1	0.19	-0.34	0.025
weight	-0.24	-0.04	0.7	0.22	0.21	0.31	0.25	0.14	0.45	0.42	0.32	0.19	1	-0.2	-0.3
cups	-0.067	0.06	0.087	-0.24	-0.18	0.12	-0.51	0.36	-0.032	-0.5	0.13	-0.34	-0.2	1	-0.2
rating	0.14	0.2	-0.69	0.47	-0.41	-0.4	0.58	0.052	-0.76	0.38	-0.24	0.025	-0.3	-0.2	1

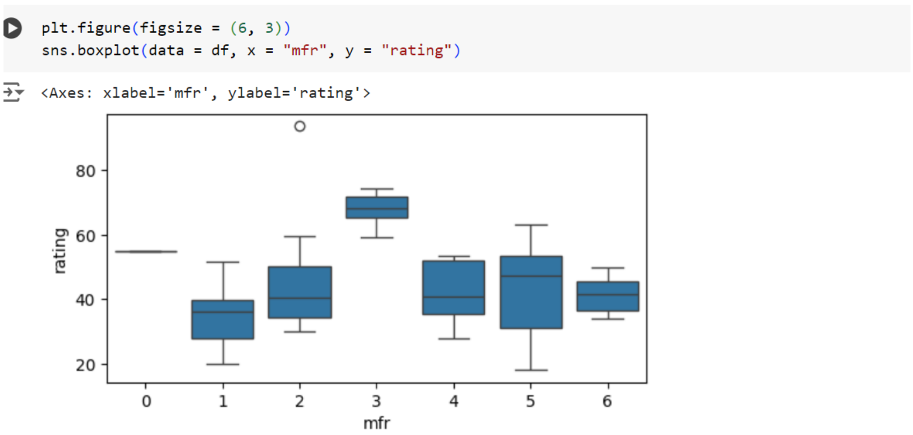
Data Visualisation:



```
df.corr()
```

mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
1.000000	-0.007103	-0.076328	0.017059	0.077661	-0.175791	0.056159	-0.063045	-0.132900	-0.003241	-0.274766	0.003323	-0.240092	-0.066967	0.140942
-0.007103	1.000000	-0.071596	0.269265	-0.002615	-0.321552	-0.078114	-0.123023	-0.285219	-0.079825	-0.180633	-0.131730	-0.039880	0.060057	0.203024
-0.076328	-0.071596	1.000000	0.019066	0.498610	0.300649	-0.293413	0.250681	0.562340	-0.066609	0.265356	0.097234	0.696091	0.087200	-0.689376
0.017059	0.269265	0.019066	1.000000	0.208431	-0.054674	0.500330	-0.130864	-0.329142	0.549407	0.007335	0.133865	0.216158	-0.244469	0.470618
0.077661	-0.002615	0.498610	0.208431	1.000000	-0.005407	0.016719	-0.318043	0.270819	0.193279	-0.031156	0.263691	0.214625	-0.175892	-0.409284
-0.175791	-0.321552	0.300649	-0.054674	-0.005407	1.000000	-0.070675	0.355983	0.101451	-0.032603	0.361477	-0.069719	0.308576	0.119665	-0.401295
0.056159	-0.078114	-0.293413	0.500330	0.016719	-0.070675	1.000000	-0.356083	-0.141205	0.903374	-0.032243	0.297539	0.247226	-0.513061	0.584160
-0.063045	-0.123023	0.250681	-0.130864	-0.318043	0.355983	-0.356083	1.000000	-0.331665	-0.349685	0.258148	-0.101790	0.135136	0.363932	0.052055
-0.132900	-0.285219	0.562340	-0.329142	0.270819	0.101451	-0.141205	-0.331665	1.000000	0.021696	0.125137	0.100438	0.450648	-0.032358	-0.759675
-0.003241	-0.079825	-0.066609	0.549407	0.193279	-0.032603	0.903374	-0.349685	0.021696	1.000000	0.020699	0.360663	0.416303	-0.495195	0.380165
-0.274766	-0.180633	0.265356	0.007335	-0.031156	0.361477	-0.032243	0.258148	0.125137	0.020699	1.000000	0.299262	0.320324	0.128405	-0.240544
0.003323	-0.131730	0.097234	0.133865	0.263691	-0.069719	0.297539	-0.101790	0.100438	0.360663	0.299262	1.000000	0.190762	-0.335269	0.025159

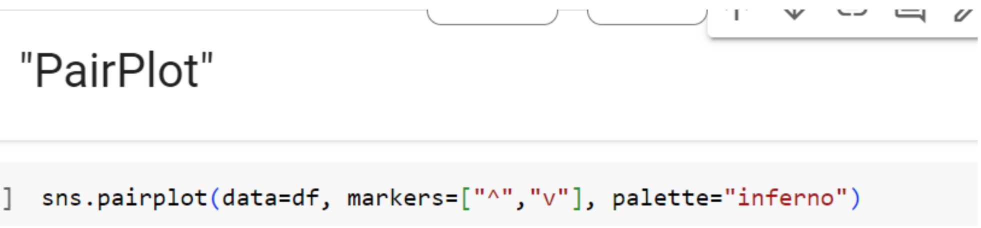
"BoxPlot"



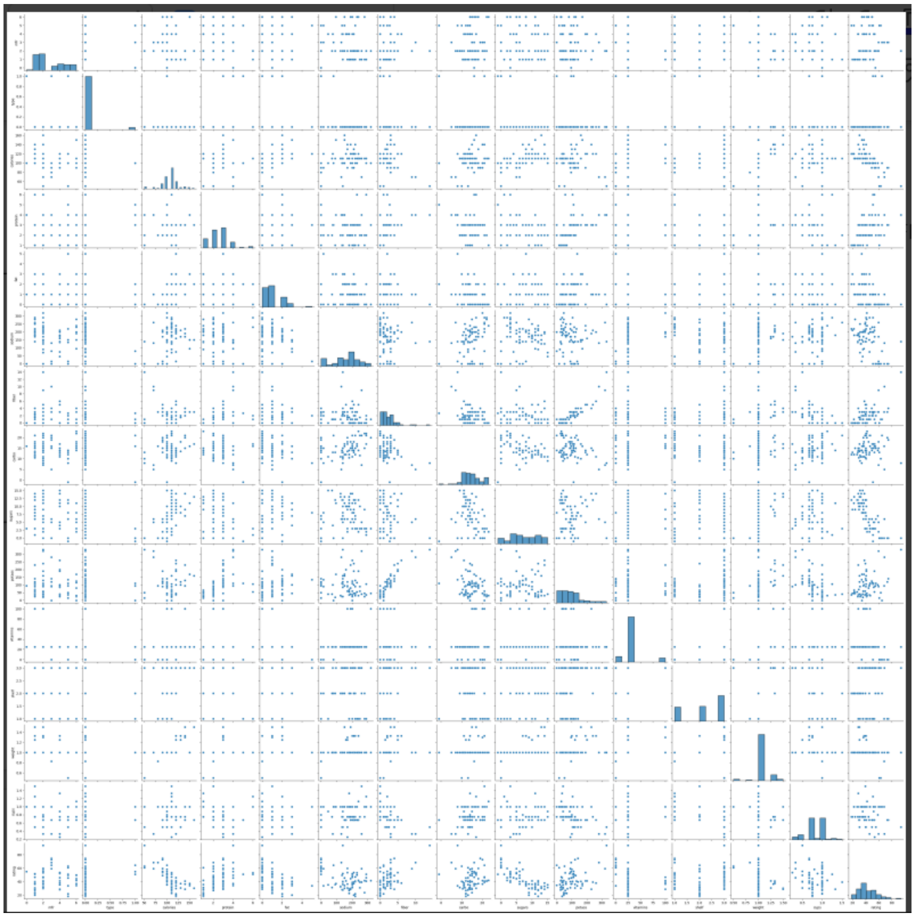
```
plt.figure(figsize = (6, 3))
sns.boxplot(data = df, x = "mfr", y = "rating")
```

```
<Axes: xlabel='mfr', ylabel='rating'>
```

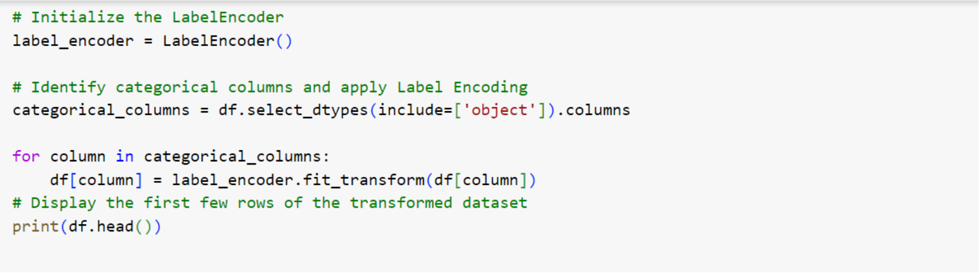
"PairPlot"



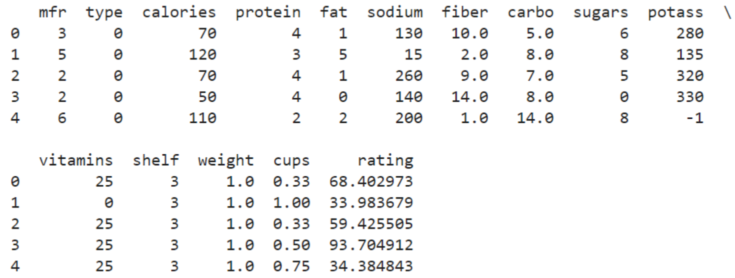
```
sns.pairplot(data=df, markers=["^","v"], palette="inferno")
```



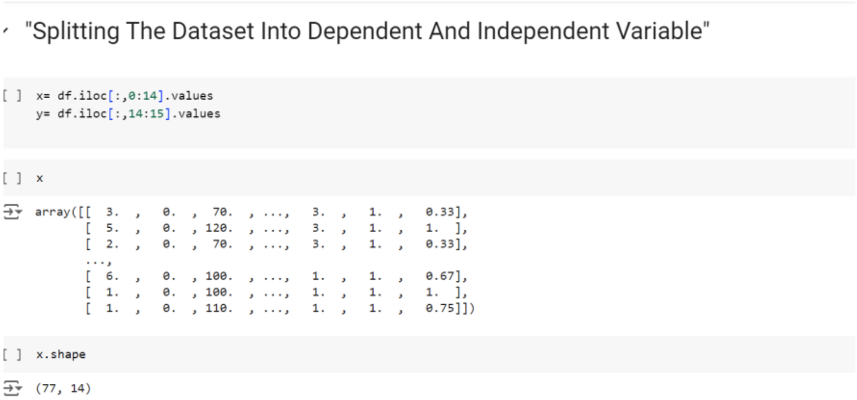
Label Encoding:



```
# Initialize the LabelEncoder
label_encoder = LabelEncoder()

# Identify categorical columns and apply Label Encoding
categorical_columns = df.select_dtypes(include=['object']).columns

for column in categorical_columns:
    df[column] = label_encoder.fit_transform(df[column])
# Display the first few rows of the transformed dataset
print(df.head())
```



```
   mfr  type  calories  protein  fat  sodium  fiber  carbo  sugars  potass  \
0    3     0        70        4    1     130   10.0    5.0       6     280   
1    5     0       120        3    5      15    2.0    8.0       8     135   
2    2     0        70        4    1     260    9.0    7.0       5     320   
3    2     0        50        4    0     140   14.0    8.0       0     330   
4    6     0       110        2    2     200    1.0   14.0       8      -1   

   vitamins  shelf  weight  cups     rating  
0        25      3     1.0  0.33  68.402973  
1         0      3     1.0  1.00  33.983679  
2        25      3     1.0  0.33  59.425505  
3        25      3     1.0  0.50  93.704912  
4        25      3     1.0  0.75  34.384843  
```



"Splitting The Dataset Into Dependent And Independent Variable"

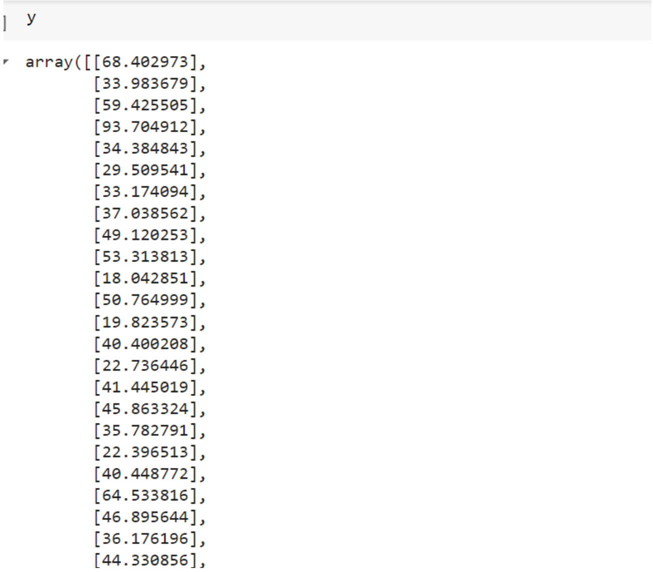
```
[ ] x= df.iloc[:,0:14].values
    y= df.iloc[:,14:15].values
```

```
[ ] x
```

```
array([[  3.  ,   0.  ,  70.  , ...,   3.  ,   1.  ,   0.33],
       [  5.  ,   0.  , 120.  , ...,   3.  ,   1.  ,   1.  ],
       [  2.  ,   0.  ,  70.  , ...,   3.  ,   1.  ,   0.33],
       ...,
       [  6.  ,   0.  , 100.  , ...,   1.  ,   1.  ,   0.67],
       [  1.  ,   0.  , 100.  , ...,   1.  ,   1.  ,   1.  ],
       [  1.  ,   0.  , 110.  , ...,   1.  ,   1.  ,   0.75]])
```

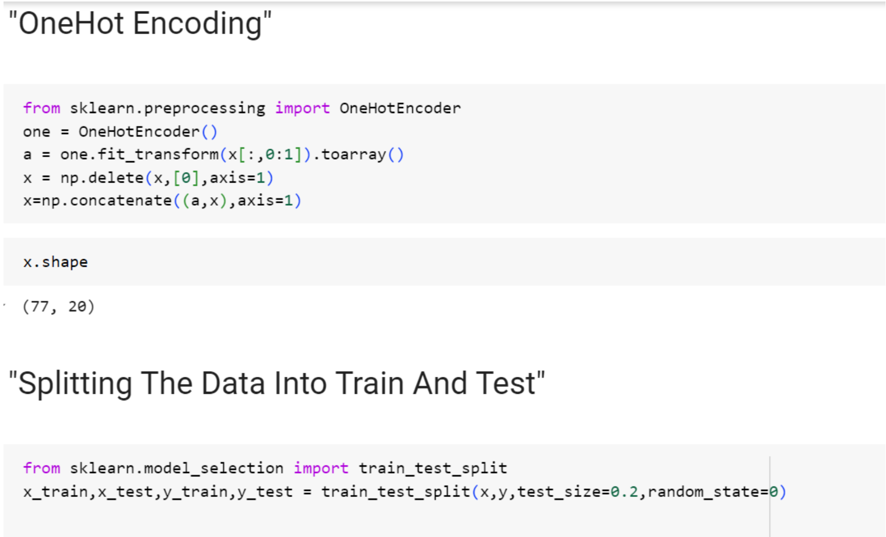
```
[ ] x.shape
```

```
(77, 14)
```



```
] y
```

```
array([[68.402973],
       [33.983679],
       [59.425505],
       [93.704912],
       [34.384843],
       [29.509541],
       [33.174094],
       [37.038562],
       [49.120253],
       [53.313813],
       [18.042851],
       [50.764999],
       [19.823573],
       [40.400208],
       [22.736446],
       [41.445019],
       [45.863324],
       [35.782791],
       [22.396513],
       [40.448772],
       [64.533816],
       [46.895644],
       [36.176196],
       [44.330856],
```



"OneHot Encoding"

```
from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
a = one.fit_transform(x[:,0:1]).toarray()
x = np.delete(x,[0],axis=1)
x=np.concatenate((a,x),axis=1)
```

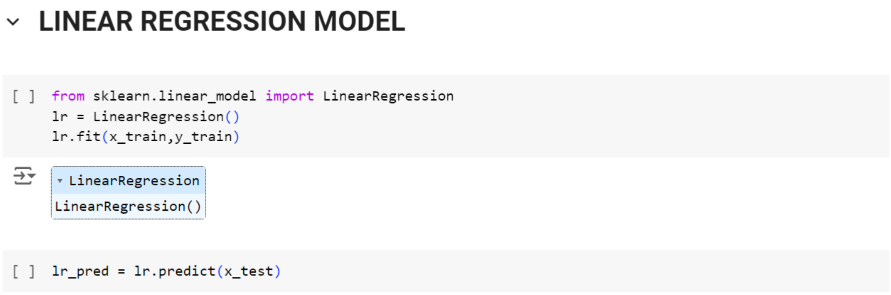
```
x.shape
```

```
(77, 20)
```

"Splitting The Data Into Train And Test"

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

MODEL BUILDING

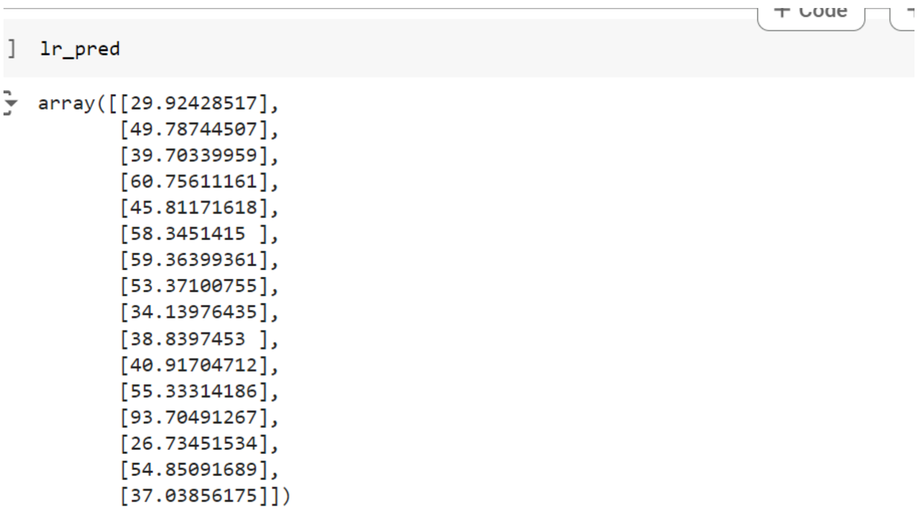


LINEAR REGRESSION MODEL

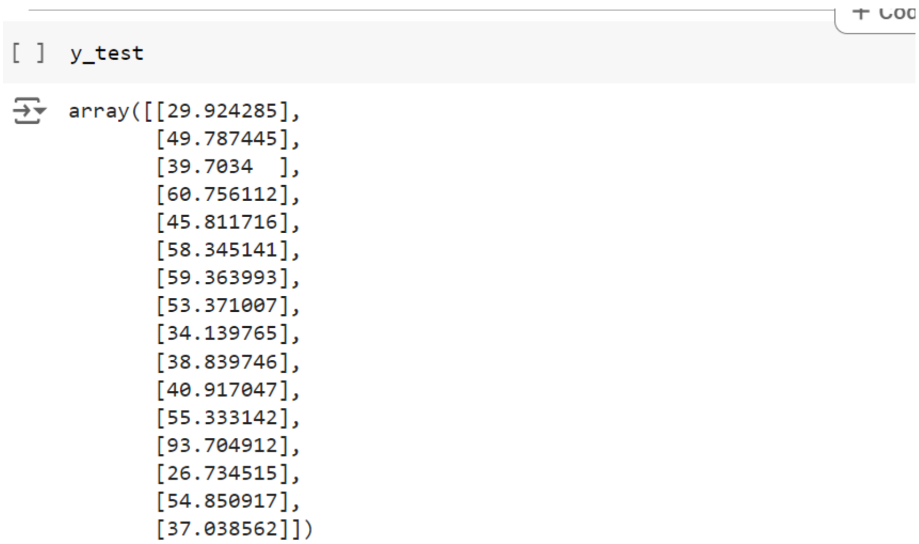
```
[ ]  from sklearn.linear_model import LinearRegression
     lr = LinearRegression()
     lr.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

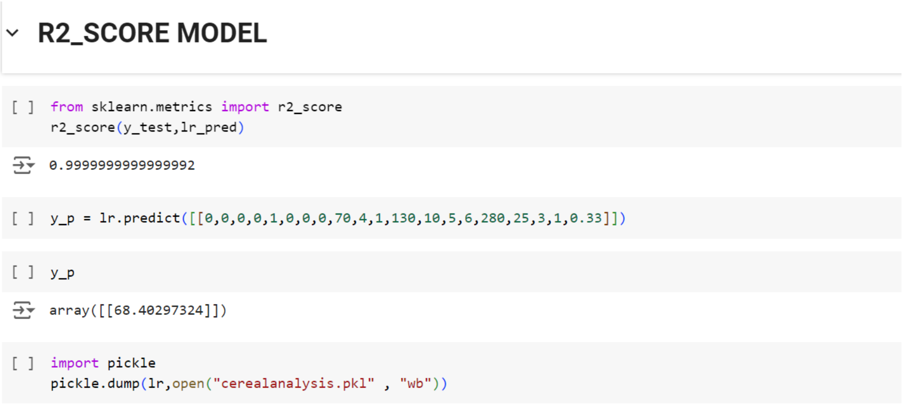
```
[ ]  lr_pred = lr.predict(x_test)
```



```
]  lr_pred

   array([[29.92428517],
          [49.78744507],
          [39.70339959],
          [60.75611161],
          [45.81171618],
          [58.3451415 ],
          [59.36399361],
          [53.37100755],
          [34.13976435],
          [38.8397453 ],
          [40.91704712],
          [55.33314186],
          [93.70491267],
          [26.73451534],
          [54.85091689],
          [37.03856175]])
```



```
[ ]  y_test

     array([[29.924285],
            [49.787445],
            [39.7034  ],
            [60.756112],
            [45.811716],
            [58.345141],
            [59.363993],
            [53.371007],
            [34.139765],
            [38.839746],
            [40.917047],
            [55.333142],
            [93.704912],
            [26.734515],
            [54.850917],
            [37.038562]])
```



R2_SCORE MODEL

```
from sklearn.metrics import r2_score
r2_score(y_test,lr_pred)
```

```
0.9999999999999992
```

```
y_p = lr.predict([[0,0,0,0,1,0,0,0,70,4,1,130,10,5,6,280,25,3,1,0.33]])
```

```
y_p
```

```
array([[68.40297324]])
```

```
import pickle
pickle.dump(lr,open("cerealanalysis.pkl" , "wb"))
```