22068184 Prabhas Shrestha.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:79790181

Submission Date

Jan 22, 2025, 5:29 AM GMT+5:45

Download Date

Jan 22, 2025, 5:31 AM GMT+5:45

File Name

22068184 Prabhas Shrestha.docx

File Size

29.5 KB

26 Pages

4,458 Words

25,392 Characters

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **56** Not Cited or Quoted 13%
  Matches with neither in-text citation nor quotation marks
- **9** Missing Quotations 2%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8%	Internet sources
4%	Publications
13%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **56** Not Cited or Quoted 13%
  Matches with neither in-text citation nor quotation marks
- **9** Missing Quotations 2%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
4% Publications
13% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	Napier University on 2023-08-16	1%
2	Internet	pbsiddhartha.ac.in	<1%
3	Submitted works	University of Westminster on 2024-11-11	<1%
4	Internet	www.coursera.org	<1%
5	Internet	www.coursehero.com	<1%
6	Submitted works	University of Westminster on 2024-04-30	<1%
7	Submitted works	Asia Pacific University College of Technology and Innovation (UCTI) on 2024-12-10	<1%
8	Submitted works	Napier University on 2024-08-13	<1%
9	Internet	climbtheladder.com	<1%
10	Internet	gist.github.com	<1%

11 Submitted works

Kingston University on 2024-01-09 <1%

12 Publication

Ton Duc Thang University <1%

13 Internet

www.analyticsvidhya.com <1%

14 Submitted works

Roehampton University on 2024-05-01 <1%

15 Submitted works

Trinity College Dublin on 2024-05-06 <1%

16 Submitted works

University of Hertfordshire on 2024-08-29 <1%

17 Internet

machinelearninghd.com <1%

18 Submitted works

Liverpool John Moores University on 2024-06-17 <1%

19 Submitted works

University of Ulster on 2024-12-04 <1%

20 Internet

codefinity.com <1%

21 Submitted works

islingtoncollege on 2025-01-16 <1%

22 Internet

srinipratapgiri.medium.com <1%

23 Submitted works

University of Westminster on 2024-01-10 <1%

24 Internet

d1wqtxts1xzle7.cloudfront.net <1%

25 Submitted works

Nexford Learning Solutions on 2024-02-09 <1%

26 Submitted works

Sheffield Hallam University on 2023-04-26 <1%

27 Publication

Charu C. Aggarwal, Chandan K. Reddy. "Data Clustering - Algorithms and Applicat... <1%

28 Submitted works

North East Scotland College on 2023-09-19 <1%

29 Submitted works

University of Arizona Global Campus (Canvas LTI 1.1) on 2025-01-13 <1%

30 Internet

interviewprep.org <1%

31 Internet

sageuniversity.edu.in <1%

32 Submitted works

Brunel University on 2024-09-11 <1%

33 Submitted works

Nottingham Trent University on 2024-08-16 <1%

34 Submitted works

University of Southampton on 2020-09-18 <1%

35 Internet

medium.com <1%

36 Internet

www.mcehassan.ac.in <1%

37 Publication

Arabghalizi, Tahereh. "Data-Driven Predictive Modeling and Multi Stakeholder Re... <1%

38 Submitted works

Edith Cowan University on 2024-10-28 <1%

39 Submitted works

University of Southampton on 2024-09-10 <1%

40 Internet

doria.fi <1%

41 Submitted works

islingtoncollege on 2025-01-16 <1%

42 Internet

open-innovation-projects.org <1%

43 Internet

www.zbw.eu <1%

44 Publication

Singh, Gurvinder. "Real-Time Quantum Computing Anomaly Detection Model on ... <1%

45 Submitted works

University of Hertfordshire on 2025-01-06 <1%

46 Submitted works

Fakultet elektrotehnike i računarstva / Faculty of Electrical Engineering and Com... <1%

Introduction

4 Artificial Intelligence (AI)

Artificial intelligence (AI) is the study and creation of computer systems capable of performing tasks that traditionally require human intelligence, such as recognizing speech, making decisions, and identifying patterns. It is a vast field that encompasses
24 technologies such as natural language processing (NLP), deep learning, and machine learning. Despite the fact that these systems have greatly improved contemporary technology, there is much disagreement about whether they are indeed artificial intelligence or just highly developed forms of machine learning. General artificial intelligence (GAI) is a future stage where robots may think, learn, and adapt at a level comparable to that of humans. Critics contend that present AI systems are specialized and lack the comprehensive reasoning, adaptability, and creativity associated with General artificial intelligence (GAI) (Coursera Staff, 2024)

Machine Learning (ML)

40 The goal of machine learning, a branch of artificial intelligence (AI), is learning from experience to achieve explicit programming of computers, enabling them to emulate intelligent human behaviour in solving problems. According to MIT's CSAIL's Boris Katz, the goal of artificial intelligence is to make computers that can understand conversational language, recognize visual scenes, and carry out physical operations. Machine learning, according to 1950s AI pioneer Arthur Samuel, is the ability of a
8 computer to learn without the need for explicit programming. MIT Sloan's Mikey Shulman explained how machine learning is ideal for complex tasks such as face recognition by humans, which may be hard to program and instruct a computer to

implement, because it allows computers to learn and adapt through experience, compared to traditional programming, which relies on precise directions, like following a recipe. (Brown, 2021)

About The Project

The chosen problem domain for this project focuses on recipe recommendation, which provides the recipe based on the ingredients the user provides. In today's world, many student/people migrate to foreign countries and live on their own, struggling with what to eat and how to cook. With their busy schedules and limited time, cooking on their own can become very hectic. So, a recipe recommendation system can help solve such issues by providing the user with recipe from the ingredients that are available at their home, based on their preference. With the help of this project food wastage can be reduce by suggesting food that are available. For example, during the COVID lockdown people had no idea on what to eat and what to prepare with limited ingredients. In such cases, this recipe recommendation will be very helpful. To make this project successful, K-Nearest Neighbors (k-NN) k-means Clustering and Logistic Regression will be integrated to find out which algorithm gives the most accurate recommendation.

Background

Research On the Issue

The COVID-19 pandemic has significantly impacted eating habits, with a noticeable shift toward more home cooking as restaurants closed and stay-at-home orders were enforced. A report from The Hartman Group highlighted that in spring 2020, 88% of all eating and drinking occasions occurred at home, a 12% increase from 2019, while dining out drastically decreased. As people found themselves cooking more, many struggled with meal planning and deciding what to cook, often with limited ingredients. This challenge, compounded by the increased time spent cooking at home, led to a rise in the demand for easy-to-prepare meals and recipe inspiration. In response, a recipe recommendation system that suggests meals based on available ingredients at home can help mitigate food waste and simplify meal planning for individuals, especially in times when finding inspiration for meals can become difficult. By leveraging such a system, users can make use of ingredients they already have, thus reducing the pressure of figuring out what to cook while minimizing food wastage. (Wiklund, 2020)

Algorithms and Tools

The Algorithms used in this project

are:

K-Nearest Neighbors (k-NN)

K-Nearest Neighbors is a supervised machine learning algorithms that is simple and instance-based which is used for classification and regression tasks. In this project, it helps in finding the recipe similar to those already preferred by the user.

K-Means Clustering

K-Means Clustering is an unsupervised machine learning algorithm which is used to divide a dataset into distinct groups based on the similarities of between data points. In this project, it helps to group recipes based on similar ingredients.

Logistic Regression

Logistic Regression is a supervised machine learning algorithm used to classify data into categories. In this project, it helps predict the probability of the set of ingredients matching a particular recipe.

The Programming tools used in this project are:

Jupyter Notebook

Jupiter Notebook is a web-based interactive computing platform. This is where all the codes of the project will be initiated.

NumPy

NumPy is a core Python library for scientific computing, offering powerful support for multidimensional arrays and a wide range of mathematical, logical, and statistical operations. (NumPy, 2024)

Scikit-learn

Scikit-learn is a Python library that offers a wide range of both supervised and unsupervised learning algorithms. It is built on top of popular libraries like NumPy, pandas, and Matplotlib, making it easy to integrate with other data science tools. Scikit-learn simplifies machine learning tasks such as classification, regression, clustering, and model evaluation. (codeacademy Team, 2024)

Pandas

Pandas is an open-source data analysis and manipulation tool, built on Python, known for its speed, flexibility, and ease of use. (pandas.pydata, 2024)

Advantages, Drawbacks and Issues

K-Nearest Neighbors (k-NN)

Advantages

It is east to implement and very simple.

It is effective in finding out complex interactions among the data in the datasets.

It requires no assumptions.

Drawbacks

Its is slow for large datasets.

It requires a lot of memory.

Incorrect data can reduce accuracy.

Issues

It is hard to manage when there are lots of features (columns).

Results can be inconsistent and confusing.

Handling different data type can be difficult.

K-Means Clustering

Advantages

It is suitable for large datasets as it is simple to understand and runs faster.

It is very easy to implement and his straightforward.

It is very scalable.

Drawbacks

Its lacks consistency.

It is highly sensitive to initial selection.

It requires predefined k beforehand.

Issues

Missing data can reduce the clustering quality

It is expensive for large datasets

Its reasoning is hard to understand.

38

Logistic Regression

Advantages

It is east to implement and interpret.

It doesn’t require a lot of data

It is very efficient and quick

Drawbacks

It is not good for complex relationship between the data.

In case of a lot of columns it may perform poorly.

Extreme values can affect the data.

Issues

Requires a lot of data for good performance.

Requires scaled features for better performance.

Requires different features for better performance.

Datasets

The data used in this project was researched and acquired through Hugging Face website, renowned for having a large collection of datasets. This website consists of over 32000 recipes, having a wide array of food items from snacks to main dishes categorized in different type of dishes like appetizers, main course, desserts and beverages.

The dataset is properly detailed with the name of the dish, the creators name, a link from where the recipe was integrated in the database, description about the recipe, ingredients and the preparation steps of the recipe along with their nutritional values like carbohydrates, proteins, fats, calories etc. This dataset helps understand the complexity and nutrition of each recipe, making it easy to be researched upon. This data structure is ideal for training machine learning models as it consists of a large amount of data. Additionally, each recipe has metadata such as ratings, the number of reviews and the URL of the image offering the insight about the user's preference and engagement making it easier to predict what recipe will be preferred by the users Overall, the dataset is very useful and valuable for AI and ML projects, as it offers a structured and comprehensive collection of data with proper detailed columns.

Figure 1: Figure showing the used Dataset (1).

Figure 2: Figure showing the used Dataset (2).

Analysis of Existing Scenario

Existing Projects

Some projects similar to my recipe recommendation are:

ChefGpt

ChefGpt is a website/app that helps people cook. In this website the user can type in their ingredients and select which meal they want to create along with the kitchen tools the users owns or currently available. The website further asks the cooking skill of the user to provide guidance if they are beginners and also providing cooking tips.

Dish Gen

DishGen is another such website which helps people cook by simplifying the cooking steps. This website is different from ChefGpt as it just asks the user for the recipe title and generates the recipe with detailed ingredients and description of the dish with detailed cooking steps.

KitchenAid

KitchenAid is another website that offers a variety of recipes tailored for their appliances from baking bread to making homemade pasta helping users create every dish with ease.

Solution

21 Proposed solution/approach to solving the problem

The proposed solution in solving the problem is by using a M; based system that recommends recipes based on the ingredients the user has at hand. Further, by using algorithms like K-Means Clustering to group similar recipes, K-Nearest Neighbors(K-NN) to match user preference and Logistic Regression to predict the user satisfaction, the system provides a recipe suggestion system. This helps in reduction of food wastage as well as helping hand to cooking food.

Algorithms used in solving such problems:

9 Using K-Means Clustering:

K-means clustering is an unsupervised learning algorithm that groups data points into clusters based on similarities. It is one of the most used clustering techniques because it is relatively simple, efficient, and effective in partitioning a dataset into clusters. This algorithm works by repeatedly updating the assignment of each data point to the closest cluster centroid and recomputing the centroid from those points assigned to it. The process repeats until convergence or attainment of maximum iterations. K-means finds its applications in customer segmentation, image segmentation, and recommendation systems among others. (Winland, 2024)

This Algorithm organises the recipes into different groups based on how similar the ingredients are to each other. This then simplifies the process by allowing the users to identify clusters of recipes that can be made with the ingredients they have at hand. By categorizing recipe into groups, K-Means helps to make it easier for the recipe in cooking the food they have at hand and avoiding wastage of food.

Figure 3: Figure Showing K-Means Clustering (javatpoint, 2024)

Using K-Nearest Neighbors(K-NN)

K-Nearest Neighbors is among the simplest and most widely used supervised machine learning methods. It works simply by classifying any data point based on the majority of its nearest neighbors. It memorizes all available cases and classifies new cases by their similarity to the memorized cases through the use of a distance metric, with the most common being Euclidean distance. KNN is a non-parametric algorithm, and by definition, it does not assume the fundamental distribution of data; hence, this algorithm can handle both numerical and categorical variables. This makes it versatile in application areas like pattern recognition, recommendation systems, and data pre-processing for missing values. It estimates the label or value of a new point by its closeness to labelled data points, taking 'k' closest neighbors as a source for class or regression value estimation. (GeeksforGeeks, 2024)

This algorithm develops this project by analyzing users' preference and past recipe selections. This algorithm compares the available ingredients at hand to similar past scenarios and then identifies the recipe that have satisfied such conditions previously. This helps users utilize their available ingredients ensuring they can prepare meals with ease and the meals they enjoy.

Figure 4: Figure Showing K-Nearest Neighbors (GeeksforGeeks, 2024)

Using Logistic Regression

Logistic regression is a supervised machine learning algorithm used for binary classification tasks, where it predicts the probability of an outcome falling into one of two classes. It works by analyzing the relationship between independent variables and a binary dependent variable, using the logistic function (sigmoid curve) to map predictions between 0 and 1. This algorithm is widely used for tasks like medical diagnosis, spam detection, and predicting outcomes such as success/failure or yes/no. It is easy to implement, interprets results well, and provides insights into how independent variables impact the prediction, but it works best with large datasets and assumes a linear relationship between predictors and the log-odds of the outcome. (Kanade, 2022)

This algorithm is used to predict the likelihood that a user will enjoy the recipe based on the available ingredients at hand. This is done by calculating the probability of a match between the user preference and the recipe. This helps to increase the user satisfaction with the effective use of the ingredients available and avoiding food wastage.

Figure 5: Figure Showing Logistic Regression (Kanade, 2022)

Pseudocode

K-Means Clustering Pseudocode

START

IMPORT the required libraires

IMPOER the .CSV file (DATASET)

// Pre-process the data

DROP unnecessary columns

DROP NaN values

SELECT relevant features

//Define K

13 SET the number of clusters “k”

INITIALIZE k random centroids

//Continue the clustering until it stops

REPEAT

ASSIGN each recipe to the nearest centroid

UPDATE centroid

//displaying result

DISPLAY clusters of similar recipes

ASSIGN users; ingredients to the nearest

cluster

DISPLAY the recipes assigned

END

K-Nearest Neighbors(K-NN) Pseudocode

START

IMPORT the required libraires

IMPOER the .CSV file (DATASET)

// Pre-process the data

DROP unnecessary columns

DROP NaN values

SELECT relevant features

SELECT target variable

//Define K

SET the number of neighbors k

//Iterate the user input

CALCULATE the distance between input ingredient and

recipe

DISPLAY recipe based on the distance

SELECT the top ‘k’

//checking values

DO

SELECT the class with highest occurrence

DISPLAY the predicted class

//Displaying results

CALCULATE the mean of the target from “k”

DISPLAY the recipes

END

Logistic Regression Pseudocode

START

IMPORT the required libraires

IMPOER the .CSV file (DATASET)

// Pre-process the data

DROP unnecessary columns

DROP NaN values

SELECT relevant features

SELECT target variable

//Define the logistic regression

SET logistic regression model

SPLIT data into train and test

TRAIN the model

//Prediction and evaluation

USE the trained model to predict the target value for

testing data

DO

//Displaying results

DISPLAY predicted outcomes

DISPLAY the recipes

END

Flowchart

K-Means Clustering Flowchart

Figure 6: Figure showing K-Means Clustering flowchart

K-Nearest Neighbors(K-NN) Flowchart

Figure 7: Figure showing K-Nearest Neighbors(K-NN) Flowchart

Logistic Regression Flowchart

Figure 8: Figure showing Logistic Regression Flowchart

Block Diagram

Figure 9: Figure showing block diagram

Development

Step 1: Import Necessary Too

Figure 10: Figure showing K-Means Clustering Importing Libraries.

In this step, the necessary tools are imported for data processing, machine learning and evaluation.

Step 2: Importing the datasets

Figure 11: Figure showing importing the data set.

Step 3: Cleansing the data

Figure 12: Figure showing cleansing of data.

This step involves selecting the columns that are necessary from the dataset to prepare data. Rows with missing values (NaN) are removed from the data to ensure clean and complete data.

Step 4: Pre-processing the data

Figure 13: Figure showing data pre-processing.

A Count Vectorizer is initialized with English stop words removed to focus on meaningful terms. Then, the vectorizer transforms the cleaned ingredients text into a sparse matrix of token counts, where each row represents a recipe and each column corresponds to a unique word.

Logistic Regression

Step 1: Prepare Dara for Logistic Regression

Figure 14: Figure showing prepare Data for Logistic Regression.

This step prepares the dataset for training a Logistic Regression model. First, a binary target column high_rating is created, categorizing recipes with a rating of 4.5 or higher as 1 and others as 0. The ingredients column is vectorized, converting text into numerical vectors. The dataset is then split into training and testing sets. This setup ensures the model has structured data for training and evaluation.

Step 2: Train Logistic Regression Model

Figure 15: Figure showing training Logistic Regression Model.

This step initializes a Logistic Regression model with a maximum of 1000 iterations and a fixed random state for reproducibility. The model is then trained using the training dataset to predict whether a recipe is highly rated based on its features. This forms the basis for making accurate predictions on new data.

Step 3: Calculate and Filter Ingredient Matches

Figure 16: Figure showing filtering Recipes by Ingredient Match.

This code calculates the match score between the user provided and recipe ingredients, storing it in ingredient_match_score. Recipes with a score greater than 0 are filtered into a new dataset, ensuring only relevant recipes based on the user input are included.

Step 4: Partial Ingredient Matching and Filtering

Figure 17: Figure showing partial Ingredient Matching and filtering.

This step calculates a partial ingredient match score by checking if each user provided ingredient is present as a substring in the recipe ingredients. The match score is computed as the ratio of matched ingredients to the total user ingredients. This score is applied to all recipes and stored in ingredient_match_score.

Step 5: Generate and Rank Recipe Recommendations

Figure 18: Figure showing generation and ranking of rank recipe.

This step predicts enjoyment likelihood for filtered recipes using Logistic Regression. Recipe are ranked by a combined score of ingredient match and enjoyment likelihood. The top 5 recommendations are displayed.

Step 6: Evaluation of Model Performance

Figure 19: Figure showing evaluation of model performance.

This step evaluated the performance of logistic regression using metrics: accuracy, precision, recall and F1-score. Prediction is made on the test dataset and each metric is calculated to assess the model's ability to classify data correctly and handle imbalances.

Step 7: Hyperparameter Tuning for logistic regression.

Figure 20: Figure showing Hyperparameter Tuning for Logistic Regression.

This step performs the hyperparameter tuning for a logistic regression model using GridSearchCv. A parameter grid is defined for regularization strength, solver options, penalty type and maximum iterations. The model is trained with 5-fold cross-validation, evaluating accuracy for each parameter's combination.

Step 8: Visualization of Model Performance Metrics

Figure 21: Figure showing Visualization of Model Performance Metrics

K-Nearest Neighbors (K-NN)

Step 1: Feature Preparation for K-Nearest Neighbor

Figure 22: Figure showing feature preparation for K-Nearest Neighbor

This step prepares features for a K-Nearest Neighbor recommendation system. Ingredient vectors are converted into an array, and numerical features like rating are scaled using StandardScalar. These processed features are combined into a single feature matrix using, which serves as the input for the KNN model.

Step 2: Identify Recipes for Ingredients

Figure 23: Figure showing identification of recipes.

This step processes the user-provided ingredients by combining them into a single input string, vectorizing the input using the trained vectorizer, and finding the nearest using the trained KNN model. The kneighbors function retrieves the closet recipes based on consine similarity.

Step 3: Train KNN for Matching

Figure 24: Figure showing Training of KNN.

8 In this step, the Nearest Neighbors model is initialized with 5 neighbors and cosine similarity as the distance metric. This model is then trained using the ingredient vectors to identify recipes similar to user-provided ingredients. This setup enables the KNN algorithm to perform efficient ingredient-based similarity searches.

Step 4: Display Top Similar Recipes

Figure 25: Figure showing the results.

This step retrieves the top 5 recipes closest to the user-provided ingredients using the indices from the KNN model. It calculates the similarity scores by converting distances to similarity scores are displayed showcasing the best matches for the user’s input ingredients.

23 Step 5: Scatter Plot: Rating vs Review Count

Figure 26: Figure showing scatter plot for rating vs review count.

The scatter plot visualizes the relationship between product ratings and the number of reviews. Each point represents a product, with its rating on the x-axis and its review count on the y-axis. The plot highlights that higher-rated products tend accumulate significantly more reviews, indicating a potential correlation between popularity and positive feedback

Step 6: Evaluation of Recommendation System

Figure 27: Figure showing evaluation of recommendation system.

This code evaluates a recommendation system by calculating precision, recall, and F1-score based on predicted similarity scores and ground truth labels. A threshold of 0.5 is used to classify predictions as relevant or irrelevant Precision measures the accuracy of relevant recommendations, recall assesses the model's ability to capture all relevant items, and the F-1 score provides a balanced evaluation of both metrics. The results show a precision of 0.6, recall of 1.0 and an F1-score calculated as the harmonic mean of precision and recall.

K-Means Clustering

Step 1: Applying K-Means Clustering

Figure 28: Figure showing applying K-Means Clustering.

In this step, the K-Means clustering algorithm is applied to group recipes based on their ingredients. The number of clusters is set to 5, defining how many groups the data will be divided into. A K-Means model is initialized with these parameters, including random_state=42 for reproducibility and n_init =10 to run the algorithm multiple times for optimal results.

Step 2: Transforming User Ingredients to Vector.

Figure 29: Figure showing transforming User Ingredients to Vector.

This step processes the user's input ingredients and converts them into a vector format for compatibility with the clustering model. The list user ingredients contain the ingredients entered by the user. These ingredients are combined into a single string using to ensure proper formatting for vectorization. The combined string is then transformed into a numerical vector using the same Count Vectorizer that was applied to the dataset earlier. This resulting user_vector can now be used to compare the user's input with the clustered data.

Step 3: Recipe Clustering and Similarity Scoring

Figure 30: Figure showing Recipe Clustering and Similarity Scoring..

This step involves calculating and sorting recipes based on similarity to user- provided ingredients while leveraging clustering for better organization. The user's input is first vectorized, and the K-Means model predicts the most relevant cluster. Cosine similarity is then calculated between the user input and all recipe ingredient vectors to determine how closely each recipe matches the input. Then the recipes are ranked in descending order of similarity scores, and the top 10 most relevant recipes are displayed.

Step 4: Evaluating K-Means Clustering Performance

Figure 31: Figure showing evaluating K-Means Clustering Performance.

This step calculates the Silhouette Score to assess the quality of clustering. The silhouette_score function takes the ingredient vectors and their corresponding cluster labels to measure how well data point are assigned to their clusters. A higher Silhouette Score indicates that the clusters are well-defined and distinct. The result is printed with two decimal precisions, showing the score for the current K-Means clustering configuration.

Step 5: Hyperparameter Tuning for KNN.

Figure 32: Figure showing Hyperparameter Tuning for KNN.

This step performs hyperparameter tuning for the K-Means clustering algorithm. A parameter grid is defined to explore combinations of the number of clusters (n_clusters), initialization method (init), number of initializations (n_init), maximum iterations (max_iter), and random state. Using nested loops, the code evaluated each combination by fitting the K- Means model to the ingredient_vectors and calculating the Silhouette Score, which measures the quality of clustering

Table comparison

The project focuses on a recipe recommendation system leveraging machine learning algorithms such as K-Means Clustering, K-Nearest Neighbors (K-NN) , and Logistic Regression. The system aims to assist users in meal planning by suggesting recipes based on available ingredients. Each algorithm contributes uniquely, k-MEANS Clusters similar recipes, K-NN identifies matches based on user preferences and Logistic Regression predicts recipe enjoyment.

Among the algorithms used in the recipe recommendation project, K-means Clustering, K-NN and Logistic Regression, KNN performs the best with an F1 score of 0.75, showing a strong balance favouring Recall (1.0 but slightly lower Precision (0.6) . Logistic Regression with an F1-Score of 0.5981 and nearly balance Precision (0.5961) and Recall (0.6001), performs reasonably but falls short to KNN. K-Means, with a Silhouette Score of 0.0984, shows poorly defined clusters and is the least effective.

Criteria

Logistic Regression

K-Nearest Neighbors (K-NN)

K-Means Clustering

Purpose

Predict enjoyment likelihood for recipes

Find recipes similar to user-provided input.

Group recipes into clusters based on similarity

Accuracy

High

Moderate

Low

Scalability

Scales well to large datasets

Struggles with large dataset.

Scales moderately with appropriate clusters.

Ease of Implementation

Relatively

Simple

Moderate

Personalization

Strong

Moderate

Weak

Speen

Fast

Slower for larger datasets

Fast after clustering is complete

Best Use Case

Predicting if users will enjoy recipes.

Finding recipes similar to specific inputs

Organizing recipes into categories

Limitations

Requires labelled data for training.

Sensitive to irrelevant features.

Clusters may not be always be meaningful.

Table 1: Tables having comparison between the algorithms.

Testing

Using Milk as ingredient

KNN:

Figure 33: Figure showing result for ingredient of Milk for KNN.

K-Means:

Figure 34: Figure showing result for ingredient of Milk for K-Means.

Logistic Regression:

Figure 35: Figure showing result for ingredient of Milk for Logistic Regression.

Using Chicken as ingredient

KNN:

Figure 36: Figure showing result for ingredient of Chicken for KNN.

K-Means:

Figure 37: Figure showing result for ingredient of Chicken for K-Means.

Logistic Regression:

Figure 38: Figure showing result for ingredient of Chicken for Logistic Regression.

Using Milk as ingredient:

KNN:

Figure 39: Figure showing result for ingredient of Flour for KNN.

K-Means:

Figure 40: Figure showing result for ingredient of Flour for K-Means.

Logistic Regression:

Figure 41: Figure showing result for ingredient of Flour for Logistic Regression.

References

Conclusion

41 This project is focused on developing a recipe recommendation system with the help of Artificial Intelligence using Machine Learning algorithms like K-Means Clustering, K-Nearest Neighbors and Logistic Regression. These algorithms were used to provide accurate recommendations based on the available ingredients from the large dataset of recipes to help reduce food wastage, help people having no idea on how to cook or what to cook with the ingredients they have and during the times of pandemic like COVID-19.

The proposed solution effectively solves real-world cg=challenges such as the difficulty of meal planning and the difficulty of utilizing limited ingredients. It provides a practical and convenient way for the users to create their own preferred meal with what they already have rather than spending more amount in other ingredients, reducing the food wastage and decreases the stress of deciding what to cook even when they have limited ingredients. This project serves a s a valuable system for users looking to optimize their cooking resources making is very useful in today’s world.

Further many improvements can be included to enhance the system such as including personalized preferences such as dietary need or the users previous cooking habits for more detailed recommendations. In the future, different advanced methods can be included in the system such as collaborating filtering, for example the system can recommend recipe liked by similar users and deep learning analyzing the user's behaviours and ingredients choices. to improve the systems overall performance to offer a even wider range of recipe choices making it a very user friendly system. Further, a community-based features like reddit can be implement for the users to discuss the trending recipes or any new recipes discovered. By providing such features, the project provides a user satisfying cooking experience.