Name : Njinju Zilefac Fogap

R-Number : R0767037

Course : Artificial Intelligence

Group : DSPS

Paper : AI Solution on predicting the weight contributed by different food ingredients.

Section A : PROBLEM DEFINAITON

“It is health that is the real wealth not just piece of gold and silver.” – Mahatma Pollan.

As an international student in Belgium, I had gained a lot of weight since I came to Belgium because of the intake of different and new ingredient which I buy daily in Belgium. So, I am trying to build a machine learning model that would help me solve this issue. So, I have data(dummies) concerning calories contributed , fiber content and approximate weight contributed by some ingredients it would help me in control the consumption of these ingredients or food that has these ingredients in it since I can predict the weight of contribution of these ingredients. My algorithm would study patterns in these data and would give me good predictions of weights and hence I would know the kind of ingredient I can add to my food or not.

SECTION B : THINKING PROCESS AND AGLORITHMS

As required, I would be using two algorithms to solve this daily problem of mine ; the two algorithms , reason I choose the model and my thinking process will be shown below :

B1. KNN classification :

Reason why I chose the model :

- Supervised learning for predicting Weight_Contribution_Level.
- Categorical output (low, high, medium).
- KNN’s adapts to new data without training aligns with potential future ingredients additions.

Thinking process :

My data set has the following features and their data type :

Ingredient : Categorical variable representing the food ingredient, like apple , Spinach , banana , Eggs , Chicken Breast etc.

Ingredient_Code : numeric representation of each ingredient.

Calories : Numeric variable representing the calories of the food.

Taste_Rating : Numeric variable representing the taste rating of the food.

Fiber_Content : Numeric variable representing the fiber content of the food.

Is_Healthy : Numeric variable representing of if the ingredients Is healthy or not. 1 -> is healthy and 0 -> is not healthy.

Weight_Contribution_Level : Numeric variable representing the weight contribution of the food. (low , medium, and high).

First step -> cleaning data(remove header) , add to list (cleandata).Bullet proof for empty file. Method name(`cleaningdata()`)

The next step was creating a dictionary that holds each ingredient and its codes, as such when a user enters an ingredient, it checks the ingredient and uses the code that represents each ingredient uniquely to calculate distance with observed features. -> Helps to bullet proof(user inputs ingredients which are not in data).method (`creatincridentcodes()`).

From the list, I added all my features to a dictionary that holds **Weight_Contribution_Level** as a value and the other features as the key. This is done using a method (`adddatatodictionary()`).

KKN groups data into class and looks for the shortest distances with the input data to return its class. I created a method for this that takes the input features and the features in the dictionary. Loops and calculates the distance between them.(`CalculateDistance()`).

The final method call (knnclassify()) that takes two arguments, the input feature we want to predict and k value. This method does the following to do our prediction :

- Converts my input from a string to int , using codes to represent each ingredient.
- Checks if the value of k is greater than 0.
- Calculates all distance between the user input and features in the dictionary and adds the distances and the weighted **Weight_Contribution_Level** to another dictionary(`distanceandclassdictionary`).
- Sorts the new dictionary with the distances in ascending order and takes the K nearest neighbor using the value of k for the user.
- Created another dictionary that would hold the counts number of the number of class and each class.
- The final stage for finding the class with the maximum number of counts and returning the class.
- I also used the same method done in class without the k and had same result with this having k.

B2. **Multi Linear regression**.

Reason why I chose the model :

Firstly, I chose this for my project because I have multiple features which I want to quantify relationships between it and my target variable. Also, I also realized that the was no correlation between the between the features so it good and there will be no overfitting. Lastly, I also want to use both regression and classification aglorithms in my project, here the output is continuing so another reason why I choose this regression technique.

Thinking process :

For linear regression, the estimated linear equation is $\hat{y} = b_0 + b_1*x_1 + b_2*x_{2,} + e$, where e is the error between the predicted value and actual value, and it is 0 by default.

- The first step was cleaning the data set (removing headers and adding the clean data in a list).
- Then, I created different list that holds the features(x1-x5) and the target (y) from the cleaddata list. This was done with `creatingfeatures()`.
- I created a dictionary for ingredients and codes. Would help for bullet proofing and also use code when user input ingredient. -> creatincridentcodes().
- Calculate X_1^2, X_2^2, X_1y, X_2y and X_1X_2 -> achieved with methods in project.
- Next, I calculated the regression sum –> using methods and formula in code.
- Then finally I calculated the coefficient of the regression b0,b1,b2,b3. Formular method included in code.
- Then I created my predicted function that gets in user input in a string and converts it's to int[], also converts the ingredients by taking it equivalent code from the dictionary created above and give our prediction base on this output. $\hat{y} = b_0 + b_1*x_1 + b_2*x_{2,} + e$.

SECTION C : **CHALLENGES FACED**

- The first challenge I first was getting adequate data to suit my project, so I created dummies data using python.
- Getting enough resources for K nearest neighbors' solution.
- Having the exact formula to calculate the coefficient of the multi linear regression.

SECTION D : **ETHICAL ASPECTS**

- **Privacy** : since we are using data from users my application doesn't protect personal data from user if hacked, heath related data of user can be seen.
- **Bais and discrimination:** since my ML application doesn't consider all ingredients, it would be biased for certain cultures or races.
- **Accuracy and representativeness :** data collection Is very difficult, therefore having accurate predictions would be difficult.

Sources :

1. https://www.statology.org/multiple-linear-regression-by-hand/
2. https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4
3. https://www.c-sharpcorner.com/learn/learn-machine-learning-with-python/machine-learning-knn-knearest-neighbors
4. https://towardsdatascience.com/using-k-nearest-neighbours-to-predict-the-genre-of-spotify-tracks-796bbbad619f
5. https://www.colorado.edu/amath/sites/default/files/attached-files/lesson12_multregression.pdf