Eatalyze: Using Machine Learning to Classify and Analyze Food Nutrition

Elisabeth Kollrack
Department of Mathematics and Statistics
University of Vermont
Burlington, VT 05405
ekollrac@uvm.edu

Ellie Sheppard
Department of Mathematics and Statistics
University of Vermont
Burlington, VT 05405
eesheppa@uvm.edu

Aaron Wilson
Department of Electrical Engineering
University of Vermont
Burlington, VT 05405
awilso23@uvm.edu

Abstract

The original objective of this project was to classify food as healthy or unhealthy from an image. However, we quickly found that this goal was beyond what we were able to do given time restraints. The project was split into two components: using machine learning models to predict healthy or not healthy based on nutritional values, and an image classifier to identify the food. The models developed include linear regression, XGBoost, and a convolutional neural network. The linear regression model had a high accuracy when predicting calorie count. XGBoost also had high accuracy when predicting healthy/unhealthy. The image classifier performs well in identifying simple food items from a generic image, such as a hot dog on a plain plate. Future work should focus on pairing these two approaches together to first identify the food item in an image, then give the user a decision on whether it is healthy or not.

1 Introduction and Problem Definition

For our project, we want to make a model that takes a photo of a a food item and classifies it. We also want to know what macro nutrients contribute to this classification. However, as stated in the abstract, the image classification portion was simplified to just classifying what the image was, instead of labeling it as healthy or unhealthy. The motivation for this project is that some people are not familiar with what food is a healthy choice or not, and could help those looking to eat well make choices that are better for their health. Machine learning is a good tool to solve this problem because it would give people an option for opinions on their diet without judgment from a real person.

2 Related Work and Bibliography

The first study aimed to look at how machine learning can be used to figure out how many calories are in food and how healthy they are. The researchers used multiple linear regression, random forests, and neural networks to predict energy content and classify the health risks. We will not be classifying health risks, which is part of the "Next Steps" section, but we will use linear regression to predict calorie content. [1]

The second study is a review of the different methods of image based food recognition. “A total of 159 studies were screened in this systematic review of IBFRS... among the implemented techniques, CNNs outperform all other approaches on the PAFDs with a large volume of data, since the richness of these datasets provides adequate training resources for such algorithms”. This effectively states that the CNN method of machine learning is the best approach when considering the size of the dataset. When the dataset is very large and has lots of entries, the CNN’s outperform other models due to the level of detail this method can pick up. This is yet another study that found CNN’s to be the choice of model for food nutrition. [2]

For the third study, an app was developed to facilitate taking pictures of the food, then a convolutional neural network was applied to the photo to find key ingredients and items. Specifically, Inception-v3, Inception-v4, and MobileNetV2 convolutional neural networks were used for the image classification. It seems as though many studies in this specific regard to food nutrition use CNN’s as their machine learning method of choice. [3]

3 Model and Training Algorithm

Linear Regression

Before using image classification, we decided to run a linear regression model to try to determine what factors impact the amount of calories in our food. We created a correlation matrix and found that carbohydrates, water, and fat have the most impact on calories. From there, water was taken out due to multicollinearity concerns. The final model we trained used total fat (a combination of monosaturated, polysaturated, and saturated fats) and carbohydrate to predict calories.

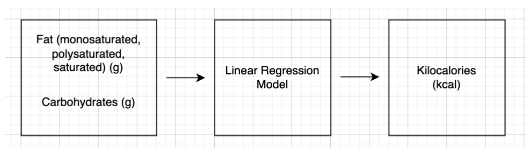

Figure 1: Linear Regression Diagram

$$\widehat{\text{calories}} = 63.68 + 3.36 \cdot \text{Fat} + 9.30 \cdot \text{Carbs} \tag{1}$$

XGBoost

We decided to use a tree-based model to predict whether a food was healthy or not. To do this, a column was added to the data set and marked as healthy if the food had less than 250 kilocalories, less than 300 milligrams of sodium, and less than 50 milligrams of cholesterol. To ensure that there was not too much data leakage (using only calorie values to predict a category that uses calories), we used protein, sugar, and fat ratios. They are calculated by taking protein, sugar, or fat and dividing it by kilocalories. This was done because these ratios are important to keep in mind when determining the nutritional value of a food.

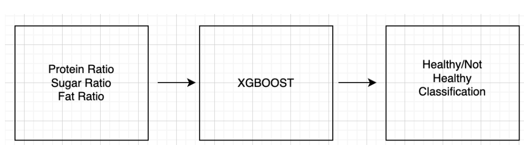

Figure 2: XGBoost Diagram

Convolutional Neural Network

There are many factors that went into the CNN, so I will only mention a few key elements. To start, the model requires a 224x224 RGB image (preprocessed so that values are between 0 and 1). This is then passed through multiple *convolutional*, *normalization*, and *max pooling* layers, before eventually being flattened and sent through the final *fully connected layers*. These *dense* layers converge to the 101 food labels. Adam was used for the optimizer, categorical cross entropy was used as the loss function, and early stoppage was implemented to stop overfitting. Epochs and general architecture were subject to change as efficiency and accuracy varied greatly.

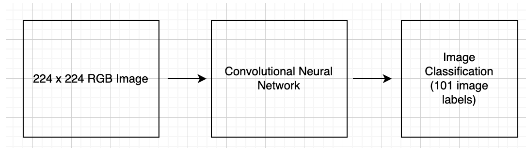

Figure 3: CNN Diagram

4 Dataset

The dataset that we are using for this first stage in the process is from kaggle.com. It includes nutritional information (carbs, fats, proteins, calories, etc.) for various categories of food items. It has 7413 cases and 18 features and around 1 MB. No augmentation was done for this dataset, however the data was normalized to get information like calories per gram. There is plenty of data for linear regression and XGBoost, as it has plenty of features and samples. A larger dataset is provided for our image classification part of the project.

For our image classification, we used this dataset from kaggle that contains images of food and their associated label. This includes things from "apple pies" to "ramen" to "donuts. There are hundreds of photos for each label, with a variety of backgrounds. This will provide very good training data to build a robust CNN. It is worthy to note that this kaggle set already has pre-partitioned train-test-split files. We did not use this in implementation, but it was a nice addition. An example of the images in this 6GB dataset can be seen in figure 4.

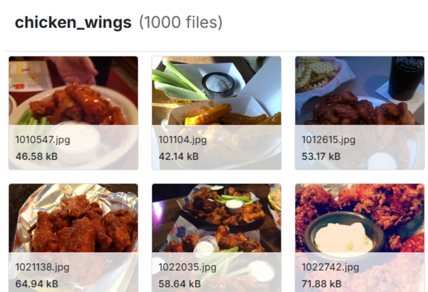

Figure 4: Example Label - Chicken Wings

5 Experimental Evaluation

5.1 Evaluation Methodology

Our hypothesis is that *"nutritional data can be used to classify whether a food item is healthy or not, and image data can be used to identify the type of food"*. To test this, we had two parts of our project: a numerical approach and an image based approach.

For the numerical approach, we used nutrition information to either predict calorie content with linear regression or classify food as healthy or unhealthy using XGBoost. To evaluate model performance, we used K-Fold cross-validation and testing on a withheld test set. These methods help assess model performance and reduce overfitting risk. We compared accuracy (classification) and RMSE (regression) for both validation techniques. Both produced similar results, so we can conclude our models generalize well to unseen data. This approach is well-suited as it ensures reliability with K-Fold cross-validation and evaluates the performance of the model using unseen data.

The second part of our project is using a Convolutional Neural Network to classify the food that is within a given image frame. To evaluate image performance, we split the data into training, validation, and test sets. Within each epoch of training, a validation set is used to calculate loss metrics and prevent overfitting to the data. Finally, the withheld "test" data is used to evaluate the overall performance of the model. This gives us an accuracy score, which we can use to judge how well the particular architecture does. A live test was also performed to see how well the trained model would do with *on the spot* images, as the whole purpose of this project is its practicality.

5.2 Results

This section presents the results after running testing data through the models above.

Below is a graph representing actual calories versus predicted calories. The closer the dots are to the line $y = x$, the better predictions. Most dots fall near this line. However, the model gets less accurate and has more outliers between 500 and 900 actual calories. The accuracy of the model was assessed using the root mean square error (RMSE), which was found to be 53.01. This suggests that on average, the model's predictions deviate by approximately 53 calories from the actual values. When using 10-fold cross validation, we found an R^2 of 0.8983, which shows that 89.83% of the variability in calories can be explained by the model.

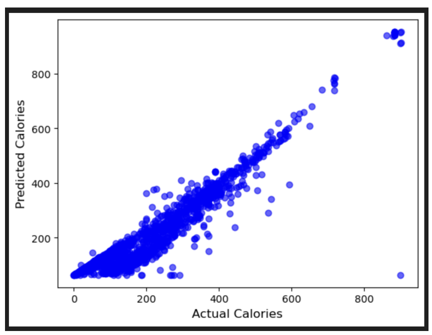

Figure 5: Linear Regression Results

In Figure 6 are the results of XGBoost. A precision of 0.87 for unhealthy foods (0) means that when the model predicts a food is unhealthy, it is usually right. A precision of 0.74 for healthy foods (1), means the model has some false positives and not everything the model predicts as healthy is actually healthy. A recall of 0.88 for unhealthy foods means that unhealthy foods are usually correctly identified. A recall of 0.72 for healthy foods means that the model misses some healthy foods. In our opinions, a false positive is worse than a false negative because we wouldn't want a user to think food is healthy when it actually is not. The overall testing accuracy is 0.83. Next steps could include tuning the model with a higher decision threshold.

```
Cross-validation accuracy: 0.8206

Test set accuracy: 0.8274
              precision    recall  f1-score   support

           0       0.87      0.88      0.87      1008
           1       0.74      0.72      0.73       475

    accuracy                           0.83      1483
   macro avg       0.80      0.80      0.80      1483
weighted avg       0.83      0.83      0.83      1483
```

Figure 6: Results from XGBoost

Figure 7 shows the results from training the CNN. We can notice that the testing set accuracy score (0.4069) is much lower than the training *validation* score (0.9155). This is to be expected, as the training set contains completely new data that the model has never seen. Testing and editing architecture was difficult when training would last for hours on end. This brings us to the first tradeoff of the CNN. The more complex we make the model, the more accurate it is, but also the longer it takes to train. Here, I needed to settle for a quicker training time to make adjustments to the model, but realistically this algorithm should run for 50+ epochs.

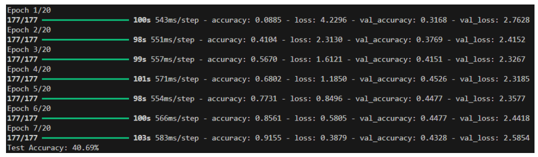

Figure 7: Results from CNN

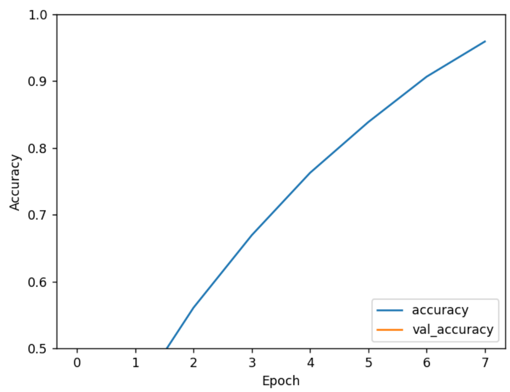

Figure 8: Epoch vs. Accuracy (validation sets)

5.3 Discussion

Our hypothesis was largely supported. In the linear regression model, we aimed to predict the caloric content of food items—a key factor in assessing healthiness. The model achieved **a low RMSE of 53**, indicating that its predictions deviated by only 53 calories on average. Given the scale of calorie values in food, this margin of error is relatively small, suggesting that the model is effective for this task.

For the XGBoost classification model, we obtained **an accuracy of 82%**, further supporting our hypothesis that nutritional ratios can help classify food as healthy or unhealthy. However, it's important to acknowledge that our definition of "healthy" was based on thresholds for calories ($<$ 250 kcal), sodium ($<$ 300 mg), and cholesterol ($<$ 50 mg). These thresholds are somewhat subjective, and different criteria could lead to different classification outcomes. Additionally, the input features—nutritional ratios—were manually selected, and with a more comprehensive dataset, we could include additional variables to improve the model's precision.

For the CNN image classification model, we obtained an **accuracy of around 40%**. These results don't quite support our hypothesis, but show strong promise toward the direction of our task. I would also like to mention, that while the CNN may get many food labels wrong, it gets very "close". Since we have translated the "labels" to numbers and use a softmax function, we oftentimes get predicted entries that are "proximally close" to the correct answer. For example, donuts are not similar to dumplings, but they are alphabetically close on the "labels" list. So, when a donuts is shown to the model, slight errors and noise can change the prediction to a dumpling (almost like gaussian noise around the prediction "donuts").

The results can be explained by the strengths of the individual algorithms: linear regression performs well on continuous prediction tasks with relatively linear relationships, while XGBoost is known for its ability to handle complex feature interactions in classification problems. The CNN uses kernels and different layering methods, which is great for image classification problems.

6 Next Steps

Next steps for this project would focus on integrating healthy or not healthy decisions based on what is identified by the image classifier in a user-friendly interface such as a phone app. This would allow our models to actually be utilized by someone to make decisions about their diet. Further work on parameters for what should and should not be considered healthy (i.e. should we cap carbs or sugar at x amount, use a calorie to fat ratio, etc) could be done, but would require someone more knowledgeable about nutrition than us. Additionally, the image classifier could be trained further to improve accuracy and add variety to the types of food it "knows". This is definitely a limitation with the machines we are running it on as it is very time consuming, but in an ideal situation such as developing it for an app someone would be able to take many more images of food and train accordingly.

7 Code

The Github repository can be found here!

8 Student Roles

- ☐ Elisabeth trained a Linear Regression model.
- ☐ Ellie has trained a XGBoost model.
- ☐ Aaron has trained a CNN
- ☐ Each member has assisted in writing this report.

9 Conclusion

We can conclude a hot dog is a hot dog! The image classifier can identify certain common foods such as a hot dog with reasonable accuracy. Nutritionally, our linear regression model identifies fat and carbohydrate content as strong predictors of calorie count, and XGBoost shows a combination of sugar, fat, and protein ratios are an effective way of determining if a food is healthy or not. Together, these results have the potential to be integrated together and develop an automated system that assesses food based on an image.

References

[1] Toby A Adjuik, Naa Adzoa A Boi-Dsane, and Bababode A Kehinde. Enhancing dietary analysis: Using machine learning for food caloric and health risk assessment. *Journal of Food Science*, 89 (11):8006–8021, 2024.

[2] Kalliopi V Dalakleidi, Marina Papadelli, Ioannis Kapolos, and Konstantinos Papadimitriou. Applying image-based food-recognition systems on dietary assessment: a systematic review. *Advances in Nutrition*, 13(6):2590–2619, 2022.

[3] Hanzhong Gao, Yanjun Liu, Jingjuan Li, and Jianwei Gao. Food nutrient extraction based on image recognition and entity extraction. pages 13–19, 2023. doi: 10.1109/WiMob58348.2023.10187783.