198:543 Final Project Report

Alexander Louie (Group 100)

December 18, 2018

1 Project Goals & Interest

1.1 Goals

The superficial goal of this project is to build an application that recommends food items to a user based on nutritional requirements (calories, restrictions, allergies) and preferences (gaining weight/losing weight). Users would first input their nutritional information and the application would return up to 30 dishes that the specific user is most likely to enjoy based on the average American's taste buds. From a more technical perspective, the ultimate goal of this project is to discover the relationship between certain food traits (calories, protein, sugar, etc.) and our perception of liking that particular food item.

This will be achieved by leveraging a specific data set containing vast amounts of data on different food items and their nutritional qualities. By using both linear regression and logistic regression classification throughout our analysis, this project will aim to see how deep learning can be applied to the food industry, to provide users with a neat tool to find customized food recommendations targeted towards their health needs and goals, and to establish an unknown relationship between how food qualities affect the overall American taste.

An example of usage: *User states that he or she wants a 450 calorie meal, requires kosher food, is trying to lose weight, and is allergic to peanuts and tree nuts. Application performs analysis and returns 16 food items (450 calories or less, kosher, peanut/tree nut free) along with corresponding projected ratings based on the average American taste preferences.*

1.2 Interest

The obvious main point of interest is customized food recommendations. Recommendation services already exist in society (Yelp and Zomato), but are limited to solely restaurant recommendations instead of specific food items. Nutritionists, health fanatics, hospital patients, and pretty much everyone else in society needs to control what is in their diet and always face the problem of, 'what can I eat?'. For example, diabetics often struggle with exploring diverse and delicious food options as their diets are limited and they are required to stick to a very strict regimen. This leads to a very monotonous diet that consists of very similar dishes every week. With this application, we attempt to solve that restriction and open up eaters to new possibilities.

However, the main point of interest lies within determining if there is a relationship or correlation between certain food traits and how we perceive food to be ”tasty”. We can discover this by examining how different levels of calories, protein, fat, sodium, sugar, etc. affect how we enjoy food and how we choose our food choices. As taste is very subjective, my initial hypothesis is that there is a vague, if any, relationship between food traits and enjoyment. However, if we are able to establish a relationship between these two data points or classify food into being ’good’ or ’bad’ based on traits, then we can effectively solve the issues described above and provide people with food that will meet their needs and will most likely enjoy.

In terms of technological interest, this problem is also pretty interesting because it observes the working relationship between linear regression and logistic regression classification. As this data set contains mostly numerical information, I was able to adapt my application to see how both regressions perform against each other and how they work together. This technology also has great potential in the restaurant industry and the promising AR field.

2 Data Sources, Data Format, & Data Preprocessing

2.1 Data Collection and Sources

My main data source is from Kaggle. Kaggle is a public data form that offers public access to popular data sets. I discovered these data sets from Kaggle via Google’s data set search engine. The data is reliable, relevant, and satisfactory for this project as Kaggle is a pretty reputable source. Moreover, the data set received quite a few ”up votes”, which indicates that other people found it to be useful and accurate for their projects as well. Calorie information is acceptable for this specific task as long as it’s within a certain range of accuracy. There were some minor errors regarding certain ingredients that affected the accuracy of certain allergy filtering (no ’fish’ in an ’ahi tuna sandwich’). However, for the sake of this project, the data was mostly complete and therefore satisfactory. Below are the main data sets that I consulted:

- Recipes and Calorie Information: `https://www.kaggle.com/hugodarwood/epirecipes`
- Ingredient Descriptions (only used as reference): `https://www.kaggle.com/openfoodfacts/world-food-facts/version/5`
- Google Data Set Search Engine: `https://toolbox.google.com/datasetsearch`

2.2 Data Format

The food data set comes in a .csv file. I accessed this data and maintained it throughout the project using Python lists, matrices, data frames, and torches. The .csv food titles were also not all unicode formatted, which I fixed within my Flask app.

2.3 Data Preprocessing

The .csv file required a bit of preprocessing. Aside from transforming the .csv data into Python data frames, I opted to turn columns like 'protein', 'fat', and 'sodium', into ratios to calories rather than keep it as its raw form. For example, instead of a pie having 3000 protein units, I transformed the column to .67 protein units per calorie. Due to the nature of food and having the nutritional facts being dependent on portion sizes, I felt that the ratios were more representative of a 'high protein' or 'low sodium' meal. This was accomplished through lambda functions, while the data was in data frame format. Additionally, for logistic regression classification, I created an additional binary column that split up the data into two classes: 'liked' and 'disliked'. This was used to determine which class the data was aligned with and was determined based off of ratings, requirement alignment, etc.

There were also 4,188 rows with at least one NaN value or at least one missing value. I transformed all missing, corrupt, or NaN values to NaN, and removed them from the data frame. Due to the nature of food, I did not feel it was appropriate to try to recreate this missing data, as it would cause inconsistencies with the rest of the data set when classifying. Moreover, for linear regression and classification, I shuffled each training data frame and truncated the linear and logistic training set to 3000 and 2250 rows respectively to avoid over-fitting. I also eliminated all unneeded columns that did not relate to allergies, ingredients, or ratings. This included information like alcohol content, holiday relation, etc.

Lastly, the original ratings column consisted of only 5 different ratings. There was not much variation and this caused issues when performing linear regression and classification as the labels were not clearly distinct. This reduced accuracy, and as a result, I opted to apply bias weights to original ratings based on the user's preferences and allergies. For example, people trying to gain weight and allergic to peanuts would probably agree on taste with other people trying to gain weight and allergic to peanuts. On the other hand, someone who is trying to lose weight and loves peanuts would have their rating credibility slightly reduced as he or she is not as likely to have similar taste as the user. This is an assumption of the project and helped create more distinct ratings and a more accurate model.

3 Data Content & Analysis

3.1 Data Content

"Recipes and Calorie Information" is a Kaggle data set (.csv) that contains 20,052 rows of distinct foods. Each food has over 680 columns of other miscellaneous data, in which I used 20. Shape is 20,052 x 680. The primary columns that I will be using are name, review, calories, protein, fat, and sodium. There were some missing data points for some nutritional columns, which were solved and handled as described above.

"Ingredient Calories" is a Kaggle data set (.tsv) that contains rows of 300,000 ingredients and basic foods. Each ingredient or food has 100 columns that go more in-depth into nutritional facts than "Recipes and Calorie Information", but are less relevant as there are not as many full meals. For example, this set provides information on the calories for just syrup rather than pancakes and syrup as a whole meal. This was used for consultation and is not contributing to my source code.

3.2 Solution & Analysis

To solve this problem, I first asked the user for information regarding how many calories they want and what restrictions/allergies they have. This is an obvious step in determining the data set we wish to train on and the data set we wish to test on.

After collecting user input and separating training and output data sets, we establish our linear regression class and perform linear regression. Our independent variable is projected rating and our dependent variables are protein, fat, sodium, and sugar. This allows us to create projected ratings for the food items in our output data set. In terms of time performance, linear regression was rather quick as it normally takes under 2 seconds. It uses stochastic gradient descent and mean squared error. Learning rate is 0.01 and there were 250 epochs.

Independently of the linear regression, the application then proceeds to perform logistic regression classification. The labels are the lambda function-determined labels of 'liked' and 'disliked' discussed earlier in the preprocessing section. The features are protein, fat, sodium, and sugar. It is important to note that the model is trained with data that represents the general American taste and is mutually exclusive from the 30 data items that will be classified using the model. After finishing training, we can apply the model to our output data set and see if they will probably be liked or disliked. We filter out the food items that are likely to be disliked by the user according to general American taste preferences. In terms of time performance, logistic regression takes the longest as it takes less than 8 seconds. It uses stochastic gradient descent and cross entropy loss. Learning rate is 0.0001 and there were 30 epochs.

We can now take the suggested food items that were classified as 'good' from the initial 30 items and append the ratings that we obtained from the linear classifier. After sorting the data frame by linear regression rating, we now have a list of classified 'good' food suggestions, ordered by predicted rating, that fit the user's nutritional requirements. Due to the demands of user needs, each time the user submits a request, a new linear regression and logistic regression classification model is created.

The technologies involved are: Python (NumPy, Pandas, etc.), Flask, Zeppelin, and PyTorch.

4 Applications & Next Steps

4.1 Applications

In terms of practical application, this data can be used for pretty much any food purpose. The ability to classify nutritional information and to recommend food choices based on who you are as a user is quite amazing. There are a ton of use cases where restaurant patrons, diabetics, vegans, health fanatics, and anybody else who controls their diet can use this application to help not only provide diversity to their diet, but also to ensure nutritional requirements are met. For example, restaurants can use this technology to help their patrons pick different dishes on their menu that fit their nutritional goals and to introduce dishes that the patrons would normally not select. Yelp and Zomato can also apply this technology to their recommendation system to recommend restaurants based on unique user needs instead of past restaurant choices. Although the ratings and predictions will never be guaranteed to be accurate, due to the inevitable, unpredictable nature of human taste, having an application that tells us which foods we are most likely to enjoy based on both who we are and what most Americans enjoy would definitely be useful. In short, food enjoyment cannot be definitively defined by a number as it obviously varies per person, but it is nice to have a reference point on other people's opinions.

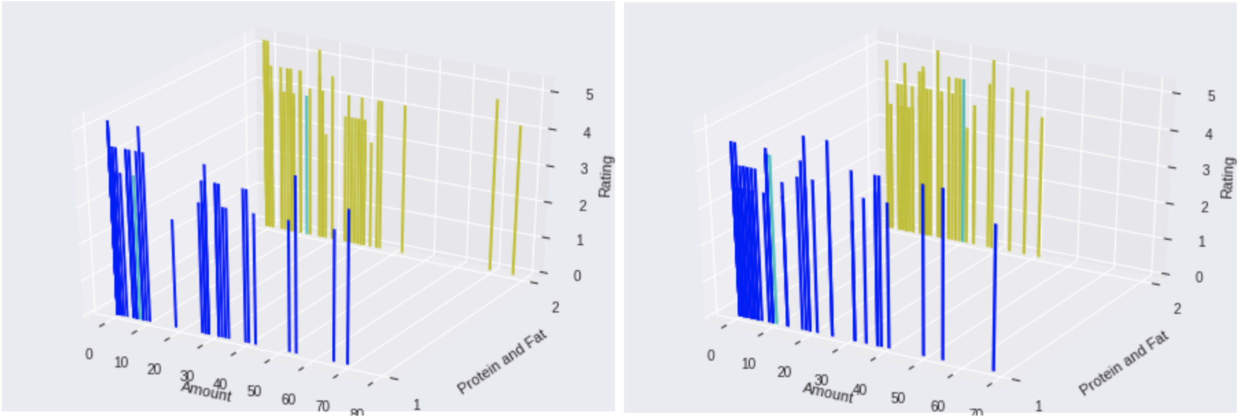
4.2 Next Steps

As for the next phases, having a more robust and more accurate data set would definitely help. As described above, there were two main issues with the data. One of them was the fact that portion sizes (how much of what) of foods obviously change how much protein or sodium is in a food item. For example, one thousand peanuts have more protein than one piece of beef, even though beef clearly has more protein per unit. I attempted to fix this through creating trait-to-calorie ratios, but having standardized information would definitely help improve the accuracy of the calculations and models. The second issue was simply inaccurate data. The data held incorrect information as it stated: 'ahi tuna sandwiches' does not contain fish. Fixing these issues is not possible without manual intervention, and I would definitely look to fix this in the next stages of the project. And as one last improvement, I would try to add more features that would better define the user, like gender, current weight, ethnicity, etc. This would require a more robust data set.

Most importantly, the biggest change I will make in the next stages is to parallelize the training processes for both linear regression and logistic regression classification. I think the biggest flaw with the application is the time it takes to retrieve results. Although the wait time is not atrocious, I would definitely like to see how GPU technology can add to the speed of the application.

To compliment the project, I included a Flask framework in which users can interact with the models in a more user-friendly manner.

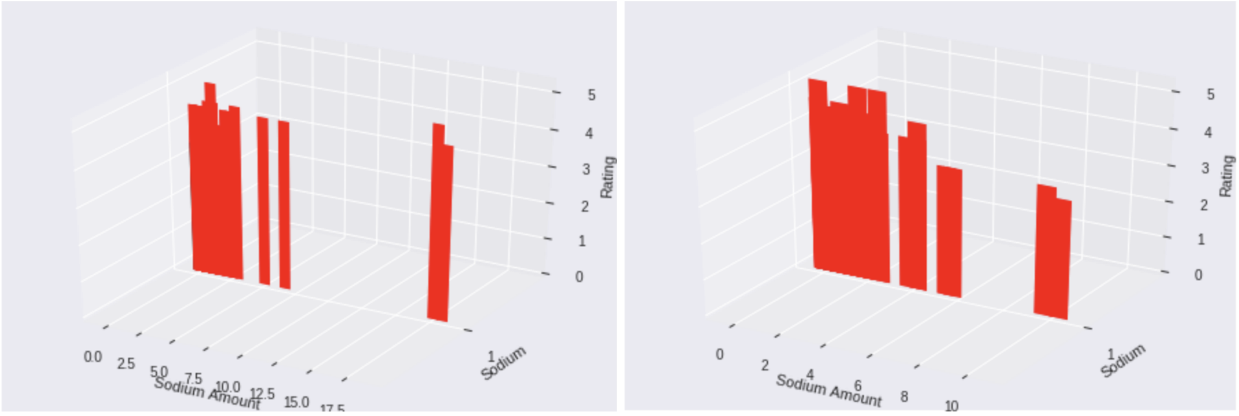
5 Visualizations & Results



(a) Low Calorie Items

(b) High Calorie Items

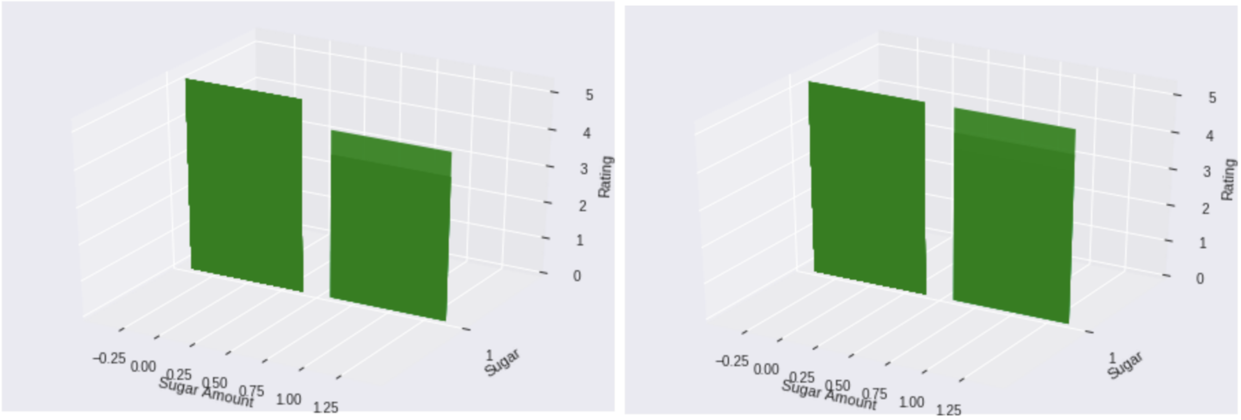
Figure 1: Protein (blue) and Fat (yellow) vs. Ratings



(a) Low Calorie Items

(b) High Calorie Items

Figure 2: Sodium (red) vs. Ratings



(a) Low Calorie Items

(b) High Calorie Items

Figure 3: Sugar (green) vs. Ratings

The above graphs display how a certain trait influences our perception of taste. Ratings are measured on a scale of 1 to 5. Each trait was split up into low calorie items and high calorie items to see how traits influence different category groups.

5.1 Figure 1

Figure 1 shows how protein and fat affect our taste for both low calorie and high calorie foods. For low calorie foods, protein does not seem to affect ratings much as the average rating did not change, whereas lower fat content items seems to have higher enjoyment ratings than higher fat content items. This makes sense as people who are eating low calorie foods (water, fruit, etc.) would typically be trying to lose weight and would not enjoy fat content. For high calorie foods, protein once again does not seem to have much of an effect, whereas fat increases slightly up to a certain amount, and then decreases after exceeding the point. This makes sense as people would enjoy up to a certain amount of fat content for high calorie foods (pie, hamburger. etc.), but would then progressively dislike it if there is too much fat in one particular item (margarine, fatty french fries).

5.2 Figure 2

Figure 2 shows how sodium affects our taste buds for low calorie and high calorie foods. For low calorie foods, sodium has no impact on enjoyment ratings. Average rating stayed the same. This leads us to believe that people don't think of sodium as an influential factor in enjoyment when consuming things like fruit and small snacks. For high calorie foods, increased sodium amounts progressively decrease enjoyment ratings. This leads us to believe that excessive amounts of sodium is disliked according to the people in the data set. This would make sense as perhaps a hamburger that is too salty will be disliked.

5.3 Figure 3

Figure 3 shows how sugar affects our taste buds for low calorie and high calorie foods. The data represents 'sugar conscious' values, so 1 is low sugar and 0 is normal or high sugar. For low calorie items, we can see that normal or high sugar items are rated higher than low sugar items. This makes sense as things like Gatorade often are considered to be tastier than water. For high calorie items, ratings are not affected by sugar levels. Although not aligned with my hypothesis, this makes sense as for highly caloric items like pizza and ice cream, there are other flavors (savory, salty, etc.) that people enjoy that would constitute a high enjoyment rating. Sugar mostly has influence for low calorie food items.

6 Conclusion & Insights Gained

6.1 Conclusion, Insights, & What Worked Well

In terms of performance, with 250 epochs and a learning rate of .01, my linear regression model typically achieved an average loss function value of 0.7561. The loss function was reduced by each epoch and performed extremely well as expected. Along with the evidence from the graphs, this stability proves to us that coefficient weights can be created to create a model that predicts a general idea of enjoyment from food traits. This is rather fascinating as this was the end goal of the project and it is interesting to think we can guess if we will like something based on nutritional facts.

With 30 epochs, a training batch size of 20, and a learning rate of .0001, my logistic regression classifier had an average loss value of 0.7069. Although this value seems better than the linear model, the loss value fluctuated throughout the epochs and hovered back and forth around .7 throughout training. I tried to adjust all parameters of the model to avoid over fitting, but was unable to prevent fluctuation when classifying if food was 'good' or 'bad'. From this fluctuation, we can conclude that certain traits do, in fact, influence taste, but there is no way to definitely classify a food item just from its traits. After due consideration, I believe the fluctuation was due to certain outliers being present in the modelling data and ultimately creating confusion in the model. The nature of food, and especially food enjoyment, is very fickle and one differentiation can cause confusion in the model as it is deciding if it is liked or not liked.

In summary, with the performance of linear regression and logistic regression classification, we can conclude that we can definitely create coefficient weights to make a vague prediction on food rating, but cannot definitively classify food into two distinct categories. This makes sense as humans can commonly say, 'you will probably like this food because it is salty', but cannot properly state, 'this food will not be liked because it is salty'. There is a fine line between linear prediction and classification, and linear regression clearly was a better choice for this project.

In terms of student insight, this project taught me a lot about the true troubles of data science and deep learning. Looking at the data and evaluating which model is best for your objective is definitely the hardest and most important parts of deep learning. Throughout this project, I thought it would be easier to classify food enjoyment into two categories, but clearly found regression was a more suitable choice.

6.2 More Time?

- Create one model and export it through Python Pickle. Because it was a user application, I created new linear regression and logistic regression classification models for each entry. This caused variation in the results as the same query will often provide different items. However, having one consistent model will improve speed and practicality.
- Use GPU processing.
- Find another classification method. As mentioned above, there could have been a better model for me to choose. I did not have time to explore all different types of models.

- Align project with food pictures. This was not a core part of finding insight about data analysis, but would definitely be a cool addition.

7 Acknowledgements

7.1 Libraries & Code

7.1.1 Libraries

- Pandas: data manipulation
- NumPy: data manipulation
- PyTorch: linear regression + logisitic regression classification
- OS / Sys: random usage
- Flask: framework for web development
- mpltoolkits.mplot3d: 3d visualization graphs

7.1.2 Code

- Logistic Regression Example: https://www.geeksforgeeks.org/identifying-handwrit
- Linear Regression Example: https://www.geeksforgeeks.org/linear-regression-usi
- 3D Bar Graph Plotting: https://matplotlib.org/gallery/mplot3d/bars3d.html

7.2 Research Papers

Although research papers were not relevant to exactly what I was trying to do, they served as good reference points and provided context on others' thought processes.

- Diet-Right Smart Food Recommendation: https://www.researchgate.net/publication/313723698_Diet-Right_A_Smart_Food_Recommendation_System
- Food Recommendation Systems: https://www.researchgate.net/publication/320944468_Food_Recommender_Systems_Important_Contributions_Challenges_and_Future_Research_Directions
- Yelp Food Recommendation: http://cs229.stanford.edu/proj2013/SawantPai-YelpFo pdf

8 How to Use

Ensure all requirements and installations are met before running (Torch, NumPy, Pandas, MatPlotLib, Flask). Application can be used either through Flask or Zeppelin.

Flask: Locate directory with app.py and run ’flask run’. Load localhost:5000.

Zeppelin: Import JSON file into Zeppelin and load localhost:8080.

Visualization: Import Visualizations.ipynb file into Google CoLab and run. Tinker with arguments as needed to customize graphs.