DS3000 Project – Recipe Evaluation Predictor

Allison Li, David Zambrano, Jasmine Wong, Yuezhu Lang

Have you ever found a recipe that seems like it could be good but you're hesitant to make it because it has no reviews? Our model is created to solve that problem!

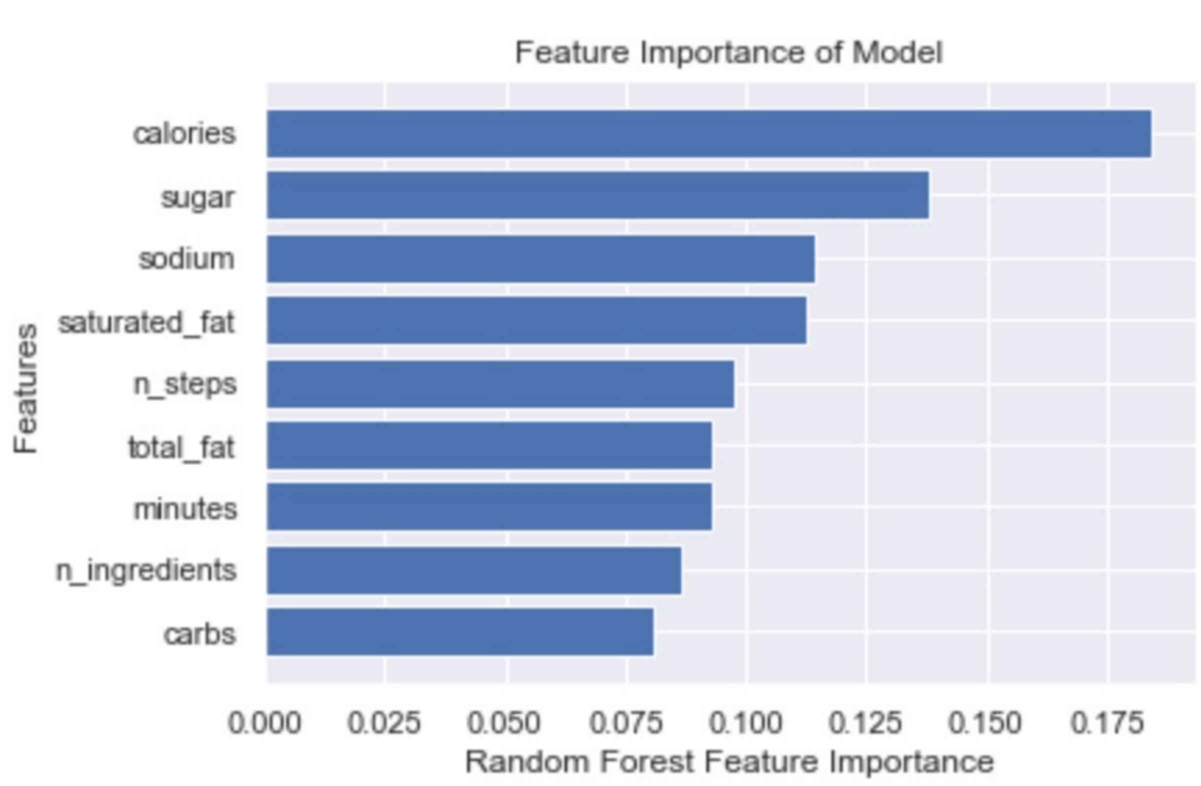
We created a predictive model that allows both online recipe/meal prep vendors, such as Blue Apron and Hello Fresh, and the general public to predict the feedback of a dish based on its recipe. This model will allow the users to easily find the best-rated recipes with their preferences.

Motivation and Objective

The objective of this project is to create a model that could predict the rating of a recipe based on its characteristics.

Specifically, we are basing our predictions on these variables:

- The **number of ingredients** required of the recipe
- The complexity of the recipes, determined by the **number of steps** needed.
- The **quantitative nutritious values** of the recipe, such as calorie count, sugar level, fat amount, carb amount, etc.
- The **estimated time** needed to complete the cooking process



A bar graph indicating the importance and correlation between the predictors and the target

Related work

Kaggle has previous food recommender engines

- Exploring TripAdvisor UK Restaurant reviews
  - A sample of user restaurant reviews on TripAdvisor UK, with a particular focus on London
- Zomato Recommendation system
  - Analysis of the demography of the location for 12,000+ international restaurants in Bengaluru

SuperCook

- a recipe generator that considers the ingredients users have at home

Yum-Me

- A personalized meal recommender system that enables a food preference profiling procedure through a quiz, and projects the learned profile into the domain of nutritionally appropriate food options for the user.

Methodology

We tried three different approaches

A **k-nn regression model** :

- recipe clusters may exist that share common traits

A **regression decision tree model** :

- provides a contrast to random forest

A **random forest model** :

- presence of multiple decision trees may present a better model than one decision tree

We used k-fold split to train our model

- X = number of steps, cooking time, nutrition information, number of ingredients.
- Y = predicted average rating

To evaluate the model ,we used the **mean squared error** to measure the amount of error in our model; the mean squared error allowed us to access the average squared difference between the observed and predicted values.

Next, for each model, we used **GridSearch** to tune inputs. For hyper parameter tuning, we used:

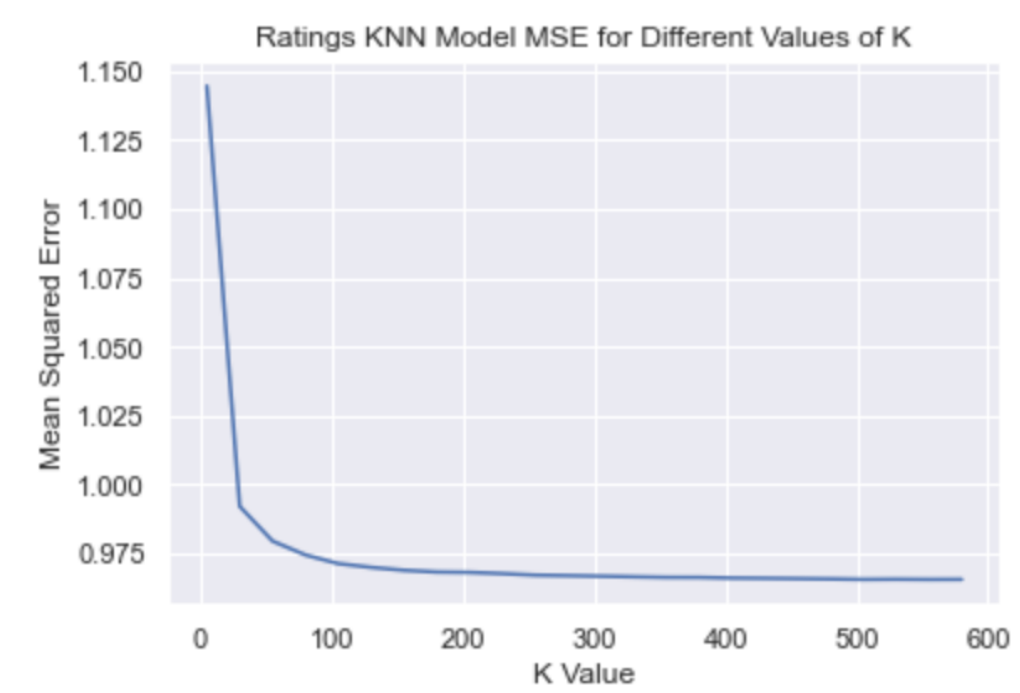
- **n_neighbors** for k-nn,
- **n_estimator** for Random Forest

Results and Evaluation

K-Nearest Neighbors Regressor:

After we compared the mean squared error from all three models, we found that **the k-nn model** was the best at predicting ratings depending on features of recipe.

Using **580** neighbors, we found the model had a mean squared error of **0.966**.



A line graph depicting the impact of varying k values on k-nn regression model's mean squared error.

Decision Tree Regressor:

We found that the decision tree model had a mean squared error of around **1.375**.

This means that our decision tree model performed worse than the KNN model, producing slightly more errors. This validates what we have learned in class, that KNN model can be better at predicting datasets than the decision tree model.

Random Forest Regression:

We originally created our random forest regression model with 100 n estimators.

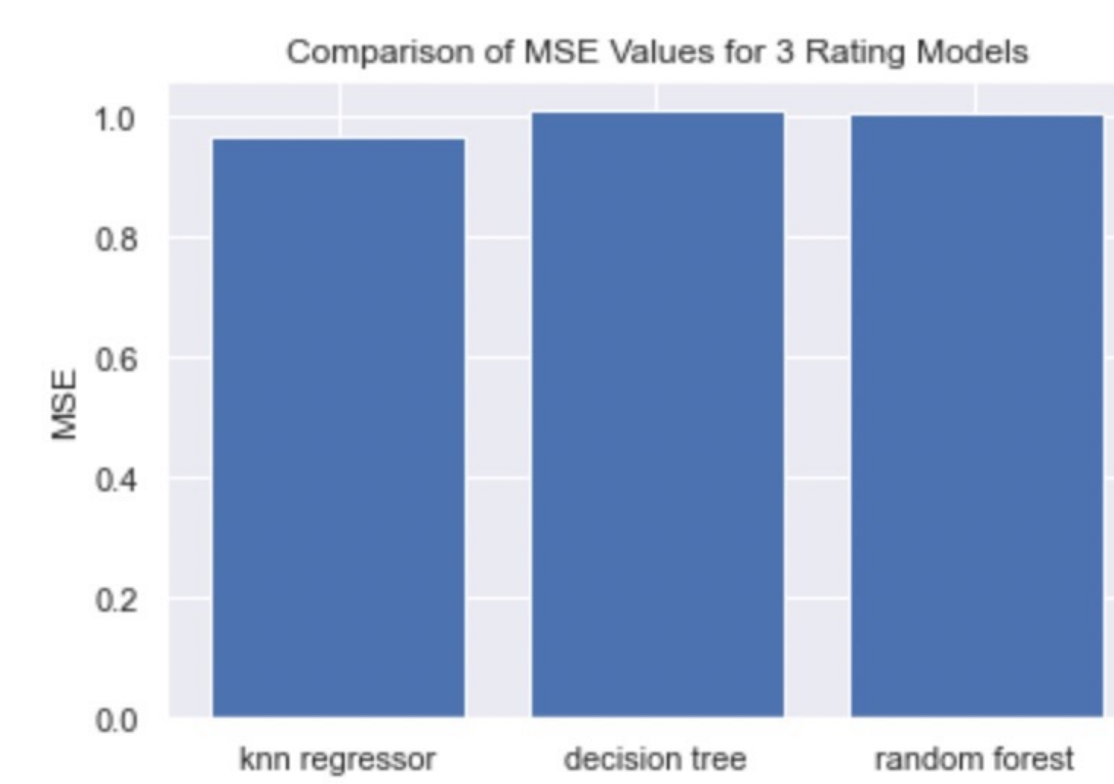
After some hyper parameter tuning, we found the n estimator value that produces the best result is **200**, as the mean squared error is **1.003**.

Conclusion and Impacts

While the mean squared error scores fluctuating around 1 suggests that the model is not extremely accurate at predicting ratings, we believe that this project has left us with two key takeaways: it has furthered our understanding about the topic of food ratings and produced a model is an asset to meal preparation vendors.

First, the model building and hyperparameter tuning revealed the difficulty of predicting human preferences. Regardless of how we tuned the ratings, the predictions still contained large mean squared errors. Thus, we must remember that the rating for a recipe is not an objective score for how good a recipe is, but the subjective opinions of a user who tested and tasted their rendition of the recipe. As a result, the reviewer's methodology is a black box that is hard to predict based only on numeric values such as calorie count or cooking time alone.

Nevertheless, our model can still be of use to large scale meal preparation vendors such as Blue Apron or Hello Fresh. The model does not perform well at predicting the exact rating of a recipe, but it does sufficiently well at differentiating successful recipes from unsuccessful recipes. These predicted ratings serve a suggestive purpose for those attempting to calculate the general public's response to the recipe. Our model could assist in the initial narrowing down of possible recipes that meal preparation vendors are considering adding to their monthly menu. Our model can also be an asset to such vendors, as it offers feedback on thousands of recipes extremely quickly, which is crucial a company's decision making.



A bar graph comparing the MSE of all three of our models.