Exploring Predictive Character of Main Courses in Indian Cuisine

Vanessa Tandiman

Introduction and Data Overview

One of the results of increased globalization, immigration, and tourism is the exchange of culture from various places, including food. Indian food, known for the extensive use of spices, is one example since it's easily found in many parts of the world. *Chicken tikka masala* has become Britain's national dish, and many international flights offer Indian dishes on their menu selections (Mangalassary 2016). This project will address a particular question about Indian food: What factor(s) characterize the main courses in Indian cuisine? We use a data set of Indian dishes consisting of main courses, desserts, and snacks, and the logistic regression model to model a dish's probability being the main course.

The data set has 179 Indian dishes, with four recorded variables: *prep time*, *cook time*, *flavor* (sweet or spicy), and the *course category* (dessert, main course, or snack).[1] We do not know if this is a random sample of all the Indian dishes, and it is reasonable to assume that these observations are independent. Figure 1 visualizes the recorded variables for 179 Indian dishes in the data set.

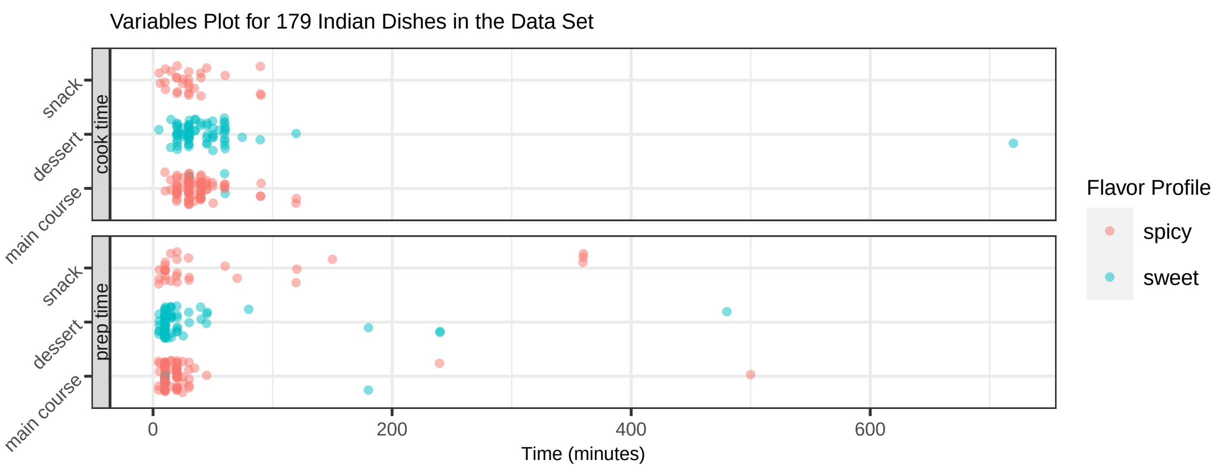

Figure 1: A jittered plot of cook time and prep time for the data. Most of the dishes have <100 minutes of cook and prep time, but a few dishes have very long prep and cook time, indicating a considerable variation in the data. Most of the main courses are spicy, all desserts are sweet, and all snacks are spicy.

Table 1 shows more information of the data's summary statistics. Flavor seems to be a characterizing variable in the course categories: 96.3% of the main courses are spicy, all desserts are sweet, and all snacks are spicy. The medians for both cook and prep time are about the same for all courses. A few dishes have very long prep and cook time, indicating considerable variation in the data, as seen in the relatively large standard deviations, especially for prep time. Main course's standard deviation is the smallest for both prep and cook time, which means that the values are closer to the mean than the other course categories.

[1] https://raw.githubusercontent.com/stat408/final_exam/master/indian.csv

Table 1: Summary Statistics of the Indian Food

Summary Statistics	Main Course (81 dishes)	Dessert (70 dishes)	Snack (28 dishes)
Prep Time (min)			
min	5	5	5
median	10	10	20
max	500	480	360
mean +- sd	26.36 ± 61.99	31.17 ± 69.86	68.04 ± 109.70
Cook Time (min)			
min	10	5	5
median	30	30	30
max	120	720	90
mean +- sd	38.09 ± 20.50	48.50 ± 83.65	32.71 ± 23.74
Flavor (dishes)			
spicy	78 (96.30%)	0 (0.00%)	28 (100.00%)
sweet	3 (3.70%)	70 (100.00%)	0 (0.00%)

The few dishes with really long prep and cook time might raise suspicion whether there had been some error. Simple internet searches verify that those seemingly unusual values are reasonable. The one dessert that takes 720 minutes to cook, *Shrikand*, is a curd based dessert that takes ±8 hours to rest the mixed ingredients, followed by another 2 or 3 hours of cooling to yield the creamy texture.[2] The spicy main course dish that takes 500 minutes to prepare, *Pindi chana*, requires soaking chickpeas for ±8 hours,[3] and another sweet dessert, *Misti doi*, takes 480 minutes of prep time because it takes ±8 hours to allow for fermentation.[4] As represented by our data, there is considerable variation in Indian dishes' prep and cook time.

Statistical Procedures

For more meaningful model coefficients, and since most time values are multiples of 5 minutes, we center the prep time and cook time at their respective averages, and then divide the centered values by 5. A scaled 5-minute difference in cook time and prep time seems more reasonable to evaluate than a 1-minute difference.

Let $y = 1$ indicate an Indian main course, and $y = 0$ indicate otherwise (either a snack or a dessert). We will model $Pr(y = 1)$. After assessing *prep time*, *cook time*, and *flavor* individually as single predictors for $Pr(y = 1)$, only *flavor* has coefficients with small enough uncertainties to have meaningful predictive information. This agrees with our earlier observation that flavor seems to be a characterizing factor. *Prep time* and *cook time* have large uncertainties as single predictors, but we still consider models that use these two variables as additional predictors to *flavor* (including possible interactions).

We then perform the 'leave one out (loo)' cross-validation to find the better predictive model. Since a few data points have considerably large cook and prep time, there is a concern about the large pareto_k. For a more credible model comparison, we perform k-fold cross validation. Table 2 displays the result for four models with the highest elpd scores, indicating more reasonable predictive models. The model with *flavor* and *cook time* (no interaction) has the highest elpd score. We will continue using this model, however, the other models listed are also reasonable since the standard error of difference is relatively large.

Table 2: Top Four Models from the K-fold Cross Validation Result

Predictors	elpd_diff	se_diff
Flavor and Cook Time	0.000000	0.000000
Flavor and Prep Time	-1.057995	5.168667
Flavor	-1.385728	1.835771
Flavor, Prep Time, and Cook Time (with interaction)	-3.154440	7.552417

[2] https://hebbarskitchen.com/shrikhand-recipe-shrikand-sweet/
[3] https://www.indianfoodforever.com/vegetables/pindi-chana.html
[4] https://hebbarskitchen.com/mishti-doi-recipe-bengali-sweet-curd/

We use a Bayesian logistic regression approach with weakly informative prior (default) in our model. The model to fit the data has the following components:
- the linear component $y^* = \beta_0 + \beta_1 x_{flavor\_sweet=1} + \beta_2 x_{cook\_time}$, where y^* is an intermediary variable in the model, and can take value from $-\infty$ to ∞.
- the logit function as the link function, mapping the output y^* to the probabilistic scale of (0,1).

$$Pr(y=1) = logit^{-1}(y^*) = \frac{exp(y^*)}{1+exp(y^*)}$$

- the binomial distribution family as the probability model.

$$Y|(X_{flavor\_sweet} = x_{flavor\_sweet}, X_{cook\_time} = x_{cook\_time}) \sim Binom(1, Pr(y=1)) = Binom(1, logit^{-1}(y^*))$$

β_0 is associated with the predicted probability of being a main course for a spicy Indian dish with average cook time. β_1 is associated with the predicted difference in odds ratio of being a main course between a sweet Indian dish and a spicy one that have the same cook time. β_2 is associated with the predicted difference in odds ratio of being a main course as the cook time differs by 5 minutes from average, for two dishes of the same flavor profile. $x_{flavor\_sweet=1}$ is the indicator variable for 'flavor_profile', where $x_{flavor\_sweet=1} = 1$ when the dish is sweet and $x_{flavor\_sweet=1} = 0$ when the dish is spicy. x_{cook_time} is the scaled cook time of the dish (in minutes), that is, centered at the average cook time and divided by 5.

Results and Discussion

The fitted model yields:

```
##                     Median MAD_SD
## (Intercept)          1.028  0.214
## flavor_profilesweet -4.252  0.650
## cook_time_sca        0.003  0.024
```

$$Pr(y=1) = logit^{-1}(1.028 - 4.252x_{flavor\_sweet=1} + 0.003x_{cook\_time\_scaled})$$

A spicy Indian dish with average cook time is predicted to have $Pr(y=1) = 0.74$, and with uncertainty, the interval of this probability is between 0.69 and 0.78. The odds ratio of being a main course between a sweet Indian dish and a spicy one that have the same cook time is predicted to decrease by about $e^{(-4.252)} = 0.014$. With uncertainty, this value could be between 0.007 and 0.027. The probability of a sweet Indian dish with average cook time is predicted to be about 0.04. Flavor makes a big difference in the predicted $Pr(y=1)$. The odds ratio of being a main course as the cook time differs by 5 minutes from average, for two dishes of the same flavor, is predicted to differ by about 1. With uncertainty, this value could be between 0.98 and 1.03. The odds ratio could either decrease or increase when cook times increases by 5 minutes from average. Figure 2(a) shows the fitted model and the data set. For each flavor, the plot shows $Pr(y=1)$ slightly rises as the cook time increases.

A posterior predictive check shows the fitted model closely predict the main course proportion in the data. Figure 2(b) shows the histogram of the predicted main course proportion in the data set. The mean (blue line) is close to the actual proportion in the data (red dashed line). We also generate prediction of $Pr(y=1)$ for ten randomly selected inputs from the data set, see Figure 2(c). For the most part, the model correctly assign higher $Pr(y=1)$ to main dishes, but it also assigns high $Pr(y=1)$ to a spicy snack, *Attu*. This is because the model relies a lot on the flavor profile, with spicy being closely associated with main course. Therefore, it is likely to incorrectly predict a spicy snack as a main course.

The binned residual plots shows that most binned residuals lie within the theoretical 95% error bounds on the plot against cook time. When plotted against the predicted $(Pr(y=1))$, low values of $(Pr(y=1))$ have more binned residuals that are barely inside the error bounds. The fitted model seems to potentially over predict and under predict $Pr(y=1)$ for low probability values, which is associated with sweet dishes.

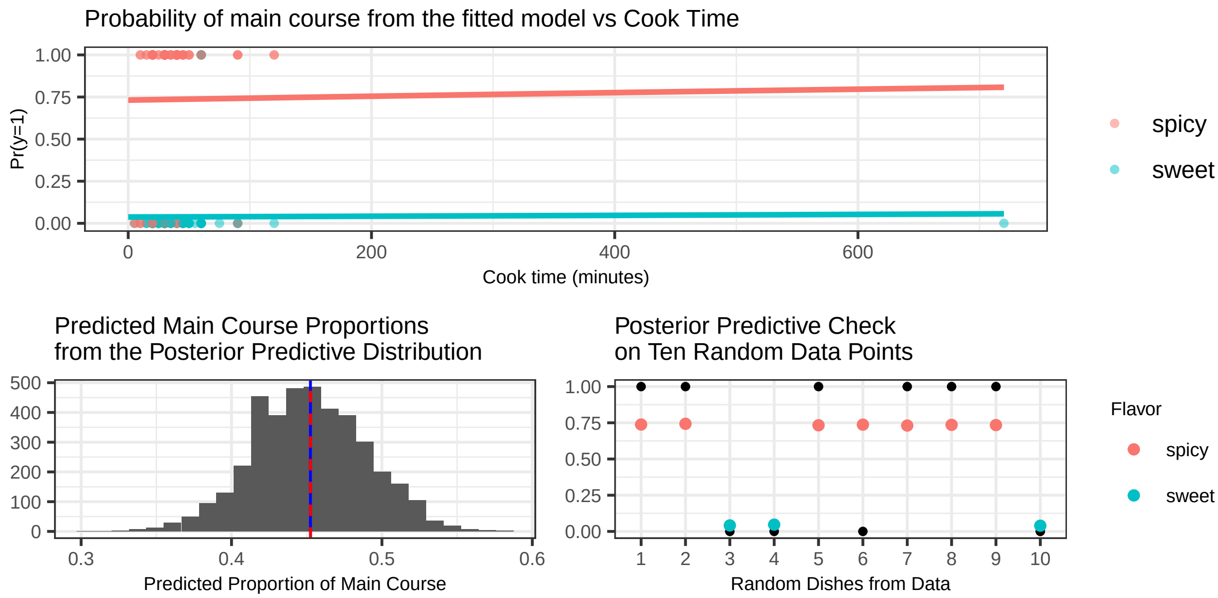

Figure 2: (a)Fitted model for main course probability. Black dots are data points. The model assigns high probability to spicy dish, and low probability to sweet dish. This probability rises slightly as the cook time increases.(b) Histogram of main course proportion in the data set from posterior predictive check. The mean of this distribution (blue line) closely approximates the actual proportion (dashed red line). (c) Posterior predictive check on ten random dishes in the data set, which mostly correct in assigning high probability to main courses. But it also predicts a spicy snack to be a main course (Dish number 6).

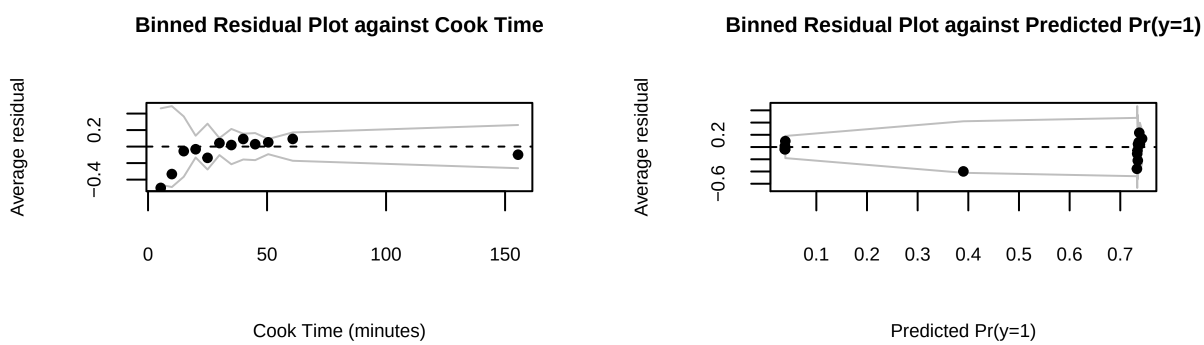

Figure 3: Binned residual plot for the fitted model, plotted versus the continuous predictor ‘cook time’, and plotted versus the predicted probability from the fitted model. The dotted line indicates the theoretical 95% error bounds if the model were true. The second plot shows more residuals on the borderline of the error bounds when Pr(y=1) is low.

The research question is what factor(s) characterize the main courses in Indian cuisine. From the data set, and assuming that it is representative of the general Indian cuisine, *flavor* seems to best answer to this question. Even though we use *cook time* in our predictive model, the uncertainty for *cook time*'s coefficient is relatively large, and we acknowledge that there could still be an underlying relationship in the larger population, other than what's indicated in our model fit. A recommendation for further work is to add other predictors, such as main ingredients, number of spices used, etc.

References

Auguie, Baptiste. 2017. *GridExtra: Miscellaneous Functions for "Grid" Graphics.* https://CRAN.R-project.org/package=gridExtra.

Gelman, Andrew, Jennifer Hill, and Aki Vehtari. 2020. *Regression and Other Stories.* Cambridge University Press.

Kassambara, Alboukadel. 2020. *Ggpubr: 'Ggplot2' Based Publication Ready Plots.* https://CRAN.R-project.org/package=ggpubr.

Mangalassary, Sunil. 2016. "Indian Cuisine-the Cultural Connection." In *Indigenous Culture, Education and Globalization*, 119–34. Springer.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain Fran?ois, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.* https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2020. *Dplyr: A Grammar of Data Manipulation.* https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2020. *Knitr: A General-Purpose Package for Dynamic Report Generation in R.* https://CRAN.R-project.org/package=knitr.

List of URLs used:
https://bookdown.org/yihui/rmarkdown-cookbook/kable.html
https://dplyr.tidyverse.org/reference/arrange.html
https://stats.stackexchange.com/questions/11406/boxplot-with-respect-to-two-factors-using-ggplot2-in-r
http://www.cookbook-r.com/Graphs/Axes_(ggplot2)/
https://cran.r-project.org/web/packages/qwraps2/vignettes/summary-statistics.html#count-and-percentages
http://thatdatatho.com/2018/08/20/easily-create-descriptive-summary-statistic-tables-r-studio/
https://cran.r-project.org/web/packages/qwraps2/vignettes/summary-statistics.html#using-variable-labels
https://stackoverflow.com/questions/12626687/positioning-horizontal-boxplots-in-ggplot2
https://stat.ethz.ch/R-manual/R-devel/library/stats/html/sigma.html
https://stat.ethz.ch/pipermail/r-help/2013-July/357111.html
https://stackoverflow.com/questions/60811561/integer0-when-plotting-a-shapefile-and-points
https://r.789695.n4.nabble.com/interpreting-quot-not-defined-because-of-singularities-quot-in-lm-tt882827.html#none
https://stats.stackexchange.com/questions/13465/how-to-deal-with-an-error-such-as-coefficients-14-not-defined-because-of-singu
http://datacornering.com/dplyr-error-in-select-unused-argument/
https://bookdown.org/ndphillips/YaRrr/arranging-plots-with-parmfrow-and-layout.html
https://mc-stan.org/loo/articles/loo2-example.html

https://stackoverflow.com/questions/14622421/how-to-change-legend-title-in-ggplot
https://www.indianfoodforever.com/vegetables/pindi-chana.html
https://hebbarskitchen.com/mishti-doi-recipe-bengali-sweet-curd/
https://stackoverflow.com/questions/52297978/decrease-overal-legend-size-elements-and-text
https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf
https://mc-stan.org/rstanarm/reference/print.stanreg.html

Appendix

The code used in this paper is found below.

```
knitr::opts_chunk$set(echo = FALSE, message=FALSE, warning=FALSE)
library(tidyverse)
library(rstanarm)
library(grid)
library(ggplot2)
library(gridExtra)
library(knitr)
library(dplyr)
library(qwraps2)
options(qwraps2_markup = "markdown")
library(arm)
library(knitr)
library(dplyr)
library(ggpubr)
library(kableExtra)

indian_food <- read_csv('https://raw.githubusercontent.com/stat408/final_exam/master/indian.csv') %>%
  filter(course != 'starter') %>%
  dplyr::select(-ingredients, -diet, -region) %>%
  mutate(flavor_profile = factor(flavor_profile),
         course = factor(course,
                         levels = c("main course",
                                    "dessert",
                                    "snack")))%>%
  arrange(name)



indian_mod <- tibble(name = rep(indian_food$name,2),
                     course = rep(indian_food$course,2),
                     flavor = rep(indian_food$flavor_profile,2),
                     time = c(indian_food$cook_time, indian_food$prep_time),
                     act = c(rep("cook time",nrow(indian_food)),
                             rep("prep time",nrow(indian_food))))
plot_all <- ggplot(aes(x=time, y=course,
                       color=flavor),
                   data = indian_mod) +
  geom_jitter(alpha=0.5, size=1, height = .3) +
  xlab("Time (minutes)") + xlim(0,NA) + theme_bw()+
  facet_wrap(.~act, dir = "v",
             strip.position="left") +
  guides(colour=FALSE) +
```

```r
  ggtitle("Variables Plot for 179 Indian Dishes in the Data Set")+
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size = 7),
        axis.title.x = element_text(size = 7),
        axis.text.y = element_text(size = 7, angle = 45),
        axis.title.y = element_blank(),
        strip.text.y = element_text(margin = margin(r=0.2,l =0.2)),
        plot.title = element_text(size = 8))

for_legend <- indian_mod %>% ggplot(aes(x=time,
                                        y=course,
                           color=flavor)) +
  geom_point(alpha=0.5, size=1) + scale_color_discrete(name = "Flavor Profile")+
  theme(legend.position = 'right') +
  theme(legend.title = element_text(size = 8),
               legend.text = element_text(size = 8))
my_legend <- as_ggplot(get_legend(for_legend))

grid.arrange(plot_all, my_legend, nrow=1, widths=c(7,1))


our_summary1 <-
  list("Prep Time (min)" =
       list("min"       = ~ min(prep_time),
            "median"    = ~ median(prep_time),
            "max"       = ~ max(prep_time),
            "mean +- sd" = ~ qwraps2::mean_sd(prep_time)),
       "Cook Time (min)" =
       list("min"       = ~ min(cook_time),
            "median"    = ~ median(cook_time),
            "max"       = ~ max(cook_time),
            "mean +- sd" = ~ qwraps2::mean_sd(cook_time)),
       "Flavor (dishes)" =
       list("spicy"       = ~ n_perc(flavor_profile == "spicy"),
            "sweet"       = ~ n_perc(flavor_profile == "sweet")))

by_course <- summary_table(dplyr::group_by(indian_food, course), our_summary1)
print(by_course,rtitle = "Summary Statistics",
      cnames = c("Main Course (81 dishes)",
                 "Dessert (70 dishes)", "Snack (28 dishes)"),
      caption = "Summary Statistics of the Indian Food")


indian_food <- indian_food %>%
  mutate(prep_time_sca = (prep_time-mean(prep_time))/5,
         cook_time_sca = (cook_time-mean(cook_time))/5) %>%
  mutate(main = factor(as.numeric(indian_food$course=="main course")))

# Single Predictors
set.seed(10101010)
fit_prep <- stan_glm(main~prep_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
```

```r
fit_cook <- stan_glm(main~cook_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_flav <- stan_glm(main~flavor_profile,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)

#2 predictors
set.seed(10101010)
fit_flav_cook <- stan_glm(main~flavor_profile+cook_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_flav_cook_int <- stan_glm(main~flavor_profile*cook_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_flav_prep <- stan_glm(main~flavor_profile+prep_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_flav_prep_int <- stan_glm(main~flavor_profile*prep_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
#3 predictors
fit_all <- stan_glm(main~flavor_profile+prep_time_sca+cook_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_all_int <- stan_glm(main~flavor_profile*prep_time_sca*cook_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_all_int_prep <- stan_glm(main~flavor_profile*prep_time_sca+cook_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)
fit_all_int_cook <- stan_glm(main~flavor_profile*cook_time_sca+prep_time_sca,
                     family = binomial(link = "logit"),
                     data = indian_food, refresh=0)



# Cross Validation
#loo
loo_flav <- loo(fit_flav)
loo_flav_cook <- loo(fit_flav_cook)
loo_flav_cook_int <- loo(fit_flav_cook_int)
loo_flav_prep <- loo(fit_flav_prep)
loo_flav_prep_int <- loo(fit_flav_prep_int)
loo_all <- loo(fit_all)
loo_int <- loo(fit_all_int)
loo_all_int_prep <- loo(fit_all_int_prep)
loo_all_int_cook <- loo(fit_all_int_cook)
loo_compare(loo_flav,loo_flav_cook, loo_flav_cook_int,
            loo_flav_prep,loo_flav_prep_int,loo_all,loo_int,
            loo_all_int_prep, loo_all_int_cook)
```

```r
#k-folds
set.seed(10101010)
k_flav <- kfold(fit_flav, K=5)
k_flav_cook <- kfold(fit_flav_cook, K=5)
k_flav_cook_int <- kfold(fit_flav_cook_int, K=5)
k_flav_prep <- kfold(fit_flav_prep, K=5)
k_flav_prep_int <- kfold(fit_flav_prep_int, K=5)
k_all <- kfold(fit_all, K=5)
k_int <- kfold(fit_all_int, K=5)
k_all_int_prep <- kfold(fit_all_int_prep, K=5)
k_all_int_cook <- kfold(fit_all_int_cook, K=5)
kfoldcv <- as.data.frame(loo_compare(k_flav, k_flav_cook, k_flav_cook_int,
            k_flav_prep,k_flav_prep_int,
            k_all,k_int))
#display
display <- cbind(Predictors = c("Flavor and Cook Time", "Flavor and Prep Time",
                                "Flavor",
                   "Flavor, Prep Time, and Cook Time (with interaction)"),
                 kfoldcv[1:4,1:2])
knitr::kable(display, "pipe",
             row.names = F,
             #align = "cc",
             caption = "Top Four Models from the K-fold Cross Validation Result")


invlogit(coef(fit_flav_cook)["(Intercept)"])
invlogit(coef(fit_flav_cook)["(Intercept)"]-fit_flav_cook$ses["(Intercept)"])
invlogit(coef(fit_flav_cook)["(Intercept)"]+fit_flav_cook$ses["(Intercept)"])
exp(coef(fit_flav_cook)["flavor_profilesweet"])
exp(coef(fit_flav_cook)["flavor_profilesweet"]-fit_flav_cook$ses["flavor_profilesweet"])
exp(coef(fit_flav_cook)["flavor_profilesweet"]+fit_flav_cook$ses["flavor_profilesweet"])
invlogit(coef(fit_flav_cook)["(Intercept)"]+coef(fit_flav_cook)["flavor_profilesweet"])
exp(coef(fit_flav_cook)["cook_time_sca"])
exp(coef(fit_flav_cook)["cook_time_sca"]-fit_flav_cook$ses["cook_time_sca"])
exp(coef(fit_flav_cook)["cook_time_sca"]+fit_flav_cook$ses["cook_time_sca"])


set.seed(10101010)
post_pred_all <- posterior_predict(fit_flav_cook, data=indian_food) %>%
  apply(1,mean)

post_all <- tibble(mean=post_pred_all) %>%
  ggplot(aes(x=mean))+geom_histogram(bins = 25)+theme_bw()+
  xlab("Predicted Proportion of Main Course")+
  geom_vline(aes(xintercept=mean(post_pred_all)), color="blue")+
  geom_vline(aes(xintercept=sum(indian_food$course=="main course")/nrow(indian_food)),
             color="red", linetype="dashed") +
  ggtitle("Predicted Main Course Proportions \nfrom the Posterior Predictive Distribution") +
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size = 7),
        axis.title.x = element_text(size = 7),
        axis.text.y = element_text(size = 7),
        axis.title.y = element_blank(),
```

```
        strip.text.y = element_text(margin = margin(r=0.2,l =0.2)),
        plot.title = element_text(size = 9))

set.seed(10101)
uji <- sample(nrow(indian_food), 10)
x_uji <- indian_food[uji,] %>%
  dplyr::select(-prep_time, -cook_time)
predict_uji <- posterior_predict(fit_flav_cook, newdata = x_uji)
uji_result <- tibble(Number = factor(seq(1:10)),
                     Course = as.numeric(x_uji$course=="main course"),
                     Model = colMeans(predict_uji),
                     Flavor = x_uji$flavor_profile,
                     Cook = x_uji$cook_time_sca)
post_rand <- uji_result %>% ggplot(aes(x=Number, y=Course))+
  geom_point(size=1)+xlab("Random Dishes from Data")+ylab("Pr(y=1)")+
  geom_point(inherit.aes = F, aes(x=Number, y=Model, color=Flavor)) +
  #theme(axis.text.x = element_text(size = 8, angle = 45))+
  theme_bw()+
  ggtitle("Result of Posterior Predictive Check \non the Data's Ten Random Dishes")+
  theme(strip.text = element_text(size=7),
        axis.text.x = element_text(size = 7),
        axis.title.x = element_text(size = 7),
        axis.text.y = element_text(size = 7),
        axis.title.y = element_blank(),
        legend.title = element_text(size = 7),
        legend.text = element_text(size = 7),
        strip.text.y = element_text(margin = margin(r=0.2,l =0.2)),
        plot.title = element_text(size = 9))



beta <- coef(fit_flav_cook)
model_fit_plot <- ggplot(indian_food, aes(x=cook_time, y=as.numeric(main)-1,
                         color=flavor_profile))+
  theme_bw()+geom_point(alpha=0.5, size=1)+
  geom_line(inherit.aes = F,
          data=tibble(temp = seq(0,720, by = .5),
                      y = invlogit(beta[1] + beta[3]*(temp-mean(indian_food$cook_time))/5)),
          aes(x=temp, y=y), color = "#F8766D", lwd=1)+
  geom_line(inherit.aes = F,
          data=tibble(temp = seq(0,720, by = .5),
                      y = invlogit(beta[1] + beta[2]+beta[3]*(temp-mean(indian_food$cook_time))/5)),
          aes(x=temp, y=y), color = "#00BFC4", lwd=1) +
  ylab("Probability of main course") + xlab("Cook time (minutes)") +
  ggtitle("Probability of main course from the fitted model vs Cook Time")+
  theme(plot.title = element_text(size=9),
        #legend.title = element_text(size=7),
        axis.title.x = element_text(size=7),
        axis.text.x = element_text(size = 7),
        axis.title.y = element_text(size = 7),
        axis.text.y = element_text(size = 7),
        legend.title = element_blank())
```

```r
grid.arrange(model_fit_plot,
             arrangeGrob(post_all, post_rand,
                         nrow=1, widths=c(4,5)),
             nrow=2, heights=c(2,2))


par(mfrow=c(1,2))
resid_cook <- binnedplot(indian_food$cook_time,resid(fit_flav_cook),
           xlab = 'Cook Time (minutes)',
           nclass=40, main =  "Binned Residual Plot against Cook Time",
           ps=15,
           cex.main=0.8,cex.pts=0.6, cex.lab=0.6, cex.axis=0.6)
resid_predicted <- binnedplot(predict(fit_flav_cook,type = 'response'),
           resid(fit_flav_prep_int),
           xlab = 'Predicted Pr(y=1)',
           nclass=40,
           ps=15,
           main = "Binned Residual Plot against Predicted Probability",
           cex.main=0.8,cex.pts=0.6, cex.lab=0.6, cex.axis=0.6)
```