Recipes Project

Name(s): Caleb Li

Website Link: https://catliba.github.io/recipes/

In [48]:
```
import pandas as pd
import numpy as np
import ast
from pathlib import Path
import matplotlib.pyplot as plt
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score, roc_auc_score, classification_report, confusion_matrix
)
from sklearn.preprocessing import QuantileTransformer
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler

import plotly.express as px
import plotly.io as pio
pio.renderers.default = "browser"
pd.options.plotting.backend = 'plotly'

# from dsc80_utils import * # Feel free to uncomment and use this.
```

Step 1: Introduction

In [49]:
```
interactions = pd.read_csv('interactions.csv')
interactions
```

Out[49]:

	user_id	recipe_id	date	rating	review
0	1293707	40893	2011-12-21	5	So simple, so delicious! Great for chilly fall...
1	126440	85009	2010-02-27	5	I made the Mexican topping and took it to bunk...
2	57222	85009	2011-10-01	5	Made the cheddar bacon topping, adding a sprin...

3	124416	120345	2011-08-06	0	Just an observation, so I will not rate. I fo...
4	2000192946	120345	2015-05-10	2	This recipe was OVERLY too sweet. I would sta...
...	...	...	...	...	...
731922	2002357020	82303	2018-12-05	5	Delicious quick thick chocolate sauce with ing...
731923	583662	386618	2009-09-29	5	These were so delicious! My husband and I tru...
731924	157126	78003	2008-06-23	5	WOW! Sometimes I don't take the time to rate ...
731925	53932	78003	2009-01-11	4	Very good! I used regular port as well. The ...
731926	2001868099	78003	2017-12-18	5	I am so glad I googled and found this here. Th...

731927 rows × 5 columns

In [50]:
```
interactions[interactions['recipe_id'] == '85009']
```

Out[50]:

	user_id	recipe_id	date	rating	review

In [51]:
```
recipes = pd.read_csv('RAW_recipes.csv')
recipes
```

Out[51]:

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_steps	steps	description	ingredi
0	1 brownies in the world best ever	333281	40	985201	2008-10-27	['60-minutes-or-less', 'time-to-make', 'course...	[138.4, 10.0, 50.0, 3.0, 3.0, 19.0, 6.0]	10	['heat the oven to 350f and arrange the rack i...	these are the most; chocolatey, moist, rich, d...	['bitters choco 'unsa butte
1	1 in canada chocolate chip cookies	453467	45	1848091	2011-04-11	['60-minutes-or-less', 'time-to-make', 'cuisin...	[595.1, 46.0, 211.0, 22.0, 13.0, 51.0, 26.0]	12	['pre-heat oven the 350 degrees f', 'in a mixi...	this is the recipe that we use at my school ca...	['w su 'br sugar', ' 'mar
2	412 broccoli casserole	306168	40	50969	2008-05-30	['60-minutes-or-less', 'time-to-make', 'course...	[194.8, 20.0, 6.0, 32.0, 22.0, 36.0, 3.0]	6	['preheat oven to 350 degrees', 'spray a 2 qua...	since there are already 411 recipes for brocco...	['fr bro cuts', 'cr of chi s

	....	.				['time-to-	[878.3, 63.0,		['freheat the	why a millionaire	['bu
4	2000 meatloaf	475785	90	2202916	2012-03-06	['time-to-make', 'course', 'main-ingredient', ...	[267.0, 30.0, 12.0, 12.0, 29.0, 48.0, 2.0]	17	['pan fry bacon , and set aside on a paper tow...	ready, set, cook! special edition contest entr...	['mea mixt 'unsm bacon', '
...	...	...	...	...	...	...	...	...	...	...	
83777	zydeco soup	486161	60	227978	2012-08-29	['ham', '60-minutes-or-less', 'time-to-make', ...	[415.2, 26.0, 34.0, 26.0, 44.0, 21.0, 15.0]	7	['heat oil in a 4-quart dutch oven', 'add cele...	this is a delicious soup that i originally fou...	['cel 'on 'green s pep '
83778	zydeco spice mix	493372	5	1500678	2013-01-09	['15-minutes-or-less', 'time-to-make', 'course...	[14.8, 0.0, 2.0, 58.0, 1.0, 0.0, 1.0]	1	['mix all ingredients together thoroughly']	this spice mix will make your taste buds dance!	['pap 'salt', 'g pow 'onion
83779	zydeco ya ya deviled eggs	308080	40	37779	2008-06-07	['60-minutes-or-less', 'time-to-make', 'course...	[59.2, 6.0, 2.0, 3.0, 6.0, 5.0, 0.0]	7	['in a bowl , combine the mashed yolks and may...	deviled eggs, cajun-style	['h coc e 'mayonna 'dijon m
83780	cookies by design cookies on a stick	298512	29	506822	2008-04-15	['30-minutes-or-less', 'time-to-make', 'course...	[188.0, 11.0, 57.0, 11.0, 7.0, 21.0, 9.0]	9	['place melted butter in a large mixing bowl a...	i've heard of the 'cookies by design' company,...	['bu 'eagle b conde milk', 'li
83781	cookies by design sugar shortbread cookies	298509	20	506822	2008-04-15	['30-minutes-or-less', 'time-to-make', 'course...	[174.9, 14.0, 33.0, 4.0, 4.0, 11.0, 6.0]	5	['whip sugar and shortening in a large bowl , ...	i've heard of the 'cookies by design' company,...	['granul su 'shorten 'eggs'

83782 rows × 12 columns

In [52]:
```
recipes.columns
```

Out[52]:
```
Index(['name', 'id', 'minutes', 'contributor_id', 'submitted', 'tags',
       'nutrition', 'n_steps', 'steps', 'description', 'ingredients',
       'n_ingredients'],
      dtype='object')
```

Column	Description
`name`	Recipe name
`id`	Recipe ID
`minutes`	Minutes to prepare recipe
`contributor_id`	User ID who submitted this recipe
`submitted`	Date recipe was submitted
`tags`	Food.com tags for recipe
`nutrition`	Nutrition information in the form `[calories (#), total fat (PDV), sugar (PDV), sodium (PDV), protein (PDV), saturated fat (PDV), carbohydrates (PDV)]`; **PDV** stands for "percentage of daily value."
`n_steps`	Number of steps in recipe
`steps`	Text for recipe steps, in order
`description`	User-provided description

In [53]:
```
# Left merge the recipes and interactions datasets together.
merged = recipes.merge(interactions, left_on='id', right_on='recipe_id', how='left')

# In the merged dataset, fill all ratings of 0 with np.nan
merged["rating"] = merged["rating"].replace(0, np.nan)

# Find the average rating per recipe, as a Series.
average_ratings = merged.groupby('id')['rating'].mean()

# Add this Series containing the average rating
recipes = recipes.merge(average_ratings.rename('avg_rating'), on='id', how='left')
```

Step 2: Data Cleaning and Exploratory Data Analysis

Clean the data appropriately. For instance, you may need to replace data that should be missing with NaN or create new columns out of given ones (e.g. compute distances, scale data, or get time information from time stamps).

Data Quality Checks:

- Scope: Do the data match your understanding of the population?
  - My data runs from 2008 to 2018, so there is a 7 year gap. It is pretty outdated but this is the dataset I am given. Perhaps in the future I choose a more relevant dataset. However, given that we are just trying to find trends in healthy and beloved, our dataset is fine.
- Measurements and values: Are the values reasonable?
  - Yes, we see correlation between the data that follows reasoning.
- Relationships: Are related features in agreement?
  - Yes.
- Analysis: Which features might be useful in a future analysis?
  - For my current question, nutrition and ratings will be my key features/labels. However, ingredients can be identified as well. Tags may provide useful info.

In [54]:
```
# Tag into array
def parse_tags(tag):
    desc = ast.literal_eval(tag)
    return desc
recipes["tags"] = recipes["tags"].apply(parse_tags)
```

From a quick scan, I've grouped some healthy tags

In [55]:
```
healthy_tags = ["vegan", "vegetarian", "gluten-free",
                "healthy", "healthy-2", "high-calcium",
                "high-fiber", 'high-in-something-diabetic-friendly', 'high-protein']
```

In [56]:
```
# Want to make the recipes with these tags as healthy
def contains_healthy(cell):
    if not isinstance(cell, list):
        return False
    cell_lower = [t.lower() for t in cell]
    return any(ht in cell_lower or ht in " ".join(cell_lower) for ht in healthy_tags)

recipes["is_healthy"] = recipes["tags"].apply(contains_healthy).astype(int)
```

In [57]:
```
recipes['submitted'].min() , recipes["submitted"].max()
```

Out[57]:
```
('2008-01-01', '2018-12-04')
```

In [58]:
```
# Note that most ratings are positive and received full marks
merged["rating"].describe()
```

Out[58]:
```
count    219393.000000
mean          4.679867
std           0.710471
min           1.000000
25%           5.000000
50%           5.000000
75%           5.000000
max           5.000000
Name: rating, dtype: float64
```

In [59]:
```
# do the same with the nutrition column
def parse_nutrition(value):
    items = ast.literal_eval(value)
    if isinstance(items, (list, tuple)) and len(items) == 7:
        return [float(x) for x in items]
    return value

recipes["nutrition"] = recipes["nutrition"].apply(parse_nutrition)
```

In [60]:
```
nutrition_col = ['calories', 'total fat', 'sugar', 'sodium', 'protein', 'saturated fat', 'carbohydrates']
recipes[nutrition_col] = pd.DataFrame(
    recipes["nutrition"].to_list(),
    index=recipes.index,
)
recipes
```

Out[60]:

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_steps	steps	description	...	n_in
0	1 brownies in the world best ever	333281	40	985201	2008-10-27	[60-minutes-or-less, time-to-make, course, mai...	[138.4, 10.0, 50.0, 3.0, 3.0, 19.0, 6.0]	10	['heat the oven to 350f and arrange the rack i...	these are the most; chocolatey, moist, rich, d...	...	
	1 in					[60-	[595.1, 46.0,		['pre-heat	this is the		

1	canada chocolate chip cookies	453467	45	1848091	2011-04-11	minutes-or-less, time-to-make, cuisine, pr...	211.0, 22.0, 13.0, 51.0, 26.0]	12	oven the 350 degrees f', 'in a mixi...	recipe that we use at my school ca...	...
2	412 broccoli casserole	306168	40	50969	2008-05-30	[60-minutes-or-less, time-to-make, course, mai...	[194.8, 20.0, 6.0, 32.0, 22.0, 36.0, 3.0]	6	['preheat oven to 350 degrees', 'spray a 2 qua...	since there are already 411 recipes for brocco...	...
3	millionaire pound cake	286009	120	461724	2008-02-12	[time-to-make, course, cuisine, preparation, o...	[878.3, 63.0, 326.0, 13.0, 20.0, 123.0, 39.0]	7	['freheat the oven to 300 degrees', 'grease a ...	why a millionaire pound cake? because it's su...	...
4	2000 meatloaf	475785	90	2202916	2012-03-06	[time-to-make, course, main-ingredient, prepar...	[267.0, 30.0, 12.0, 12.0, 29.0, 48.0, 2.0]	17	['pan fry bacon , and set aside on a paper tow...	ready, set, cook! special edition contest entr...	...
...	...	...	...	...	...	...	...	...	...	...	...
83777	zydeco soup	486161	60	227978	2012-08-29	[ham, 60-minutes-or-less, time-to-make, course...	[415.2, 26.0, 34.0, 26.0, 44.0, 21.0, 15.0]	7	['heat oil in a 4-quart dutch oven', 'add cele...	this is a delicious soup that i originally fou...	...
83778	zydeco spice mix	493372	5	1500678	2013-01-09	[15-minutes-or-less, time-to-make, course, pre...	[14.8, 0.0, 2.0, 58.0, 1.0, 0.0, 1.0]	1	['mix all ingredients together thoroughly']	this spice mix will make your taste buds dance!	...
83779	zydeco ya ya deviled eggs	308080	40	37779	2008-06-07	[60-minutes-or-less, time-to-make, course, mai...	[59.2, 6.0, 2.0, 3.0, 6.0, 5.0, 0.0]	7	['in a bowl , combine the mashed yolks and may...	deviled eggs, cajun-style	...

83780	cookies by design cookies on a stick	298512	29	506822	2008-04-15	[30-minutes-or-less, time-to-make, course, pre...	[188.0, 11.0, 57.0, 11.0, 7.0, 21.0, 9.0]	9	[' place melted butter in a large mixing bowl a...	i've heard of the 'cookies by design' company,...	...
83781	cookies by design sugar shortbread cookies	298509	20	506822	2008-04-15	[30-minutes-or-less, time-to-make, course, pre...	[174.9, 14.0, 33.0, 4.0, 4.0, 11.0, 6.0]	5	['whip sugar and shortening in a large bowl , ...	i've heard of the 'cookies by design' company,...	...

83782 rows × 21 columns

In [61]:
```
recipes.columns
```

Out[61]:
```
Index(['name', 'id', 'minutes', 'contributor_id', 'submitted', 'tags',
       'nutrition', 'n_steps', 'steps', 'description', 'ingredients',
       'n_ingredients', 'avg_rating', 'is_healthy', 'calories', 'total fat',
       'sugar', 'sodium', 'protein', 'saturated fat', 'carbohydrates'],
      dtype='object')
```

In [62]:
```
# normalize macros
recipes["protein_per_100kcal"] = (
    recipes["protein"] / recipes["calories"] * 100
)
recipes["sugar_per_100kcal"]   = (
    recipes["sugar"] / recipes["calories"] * 100
)
```

In [63]:
```
# derive time-based features
recipes["submitted"] = pd.to_datetime(recipes["submitted"], errors="coerce")

merged["date"]       = pd.to_datetime(merged["date"],       errors="coerce")
merged["review_year"]     = merged["date"].dt.year
merged["review_month"]    = merged["date"].dt.month
merged["review_weekday"]  = merged["date"].dt.dayofweek

# handy recipe age feature
today = pd.Timestamp("today").normalize()
recipes["recipe_age_years"] = (today - recipes["submitted"]).dt.days / 365.25
```

In [64]:

In [64]:
```
# standardize nutrition facts
scaler        = StandardScaler()
scaled_cols   = [f"{c}_z" for c in nutrition_col]
recipes[scaled_cols] = scaler.fit_transform(recipes[nutrition_col])
```

In [65]:
```
recipes.columns
```

Out[65]:
```
Index(['name', 'id', 'minutes', 'contributor_id', 'submitted', 'tags',
       'nutrition', 'n_steps', 'steps', 'description', 'ingredients',
       'n_ingredients', 'avg_rating', 'is_healthy', 'calories', 'total fat',
       'sugar', 'sodium', 'protein', 'saturated fat', 'carbohydrates',
       'protein_per_100kcal', 'sugar_per_100kcal', 'recipe_age_years',
       'calories_z', 'total fat_z', 'sugar_z', 'sodium_z', 'protein_z',
       'saturated fat_z', 'carbohydrates_z'],
      dtype='object')
```

In [66]:
```
# look at distribution of ratings
recipes["avg_rating_floor"] = np.floor(recipes["avg_rating"])
fig = px.histogram(
    recipes.dropna(subset=["avg_rating_floor"]),
    x="avg_rating_floor",
    category_orders={"avg_rating_floor": [1, 2, 3, 4, 5]},
    labels={"avg_rating_floor": "Floored average rating", "count": "Number of recipes"},
    title="Distribution of Floored Average Recipe Ratings",
)

fig.update_xaxes(dtick=1)
fig.update_layout(bargap=0.05)
fig.show()
```

In [67]:
```
# distribution of Protein per 100 Calories
fig1 = px.histogram(
    recipes,
    x="protein_per_100kcal",
    nbins=4,
    title="Distribution of Protein per 100 Calories",
    labels={"protein_per_100kcal": "Protein per 100 Calories (g)"},
    color_discrete_sequence=["mediumturquoise"],
)
fig1.update_layout(yaxis_title="Number of Recipes")
fig1.show()
```

In [68]:
```
# look known healthy tags and their ratings

fat_col   = "total fat"   # % Daily Value
```

```
sugar_col = "sugar"

fig = px.scatter(
    recipes.dropna(subset=[fat_col, sugar_col, "is_healthy"]),
    x=sugar_col,
    y=fat_col,
    color=recipes["is_healthy"].map({1: "Healthy", 0: "Not healthy"}),
    hover_data=["name", "calories", "avg_rating"],
    labels={
        sugar_col: "Sugar (% daily value)",
        fat_col:   "Total fat (% daily value)",
        "color":   "Health flag"
    },
    title="Sugar vs. Total Fat per Serving — Healthy vs. Not-Healthy Recipes",
    opacity=0.6,
)

fig.update_layout(legend_title_text="Recipe classified as:")
fig.show()
```

In [69]:

```
# Average rating by health flag
plot_df = (
    recipes
    .dropna(subset=["avg_rating", "is_healthy"])
    .assign(health_label=lambda d: d["is_healthy"].map({1: "Healthy", 0: "Not healthy"}))
)

summary = (
    plot_df.groupby("health_label")["avg_rating"]
           .agg(["count", "mean", "std"])
           .rename(columns={"mean": "avg_rating_mean", "std": "avg_rating_sd"})
)
display(summary)

fig = px.box(
    plot_df,
    x="health_label",
    y="avg_rating",
    points="all",
    color="health_label",
    labels={
        "health_label": "",
        "avg_rating": "Average recipe rating (1–5 stars)"
    },
    title="Average Rating Distribution — Healthy vs. Not-Healthy Recipes",
)

overall_mean = plot_df["avg_rating"].mean()
fig.add_hline(
```

```
    overall_mean, line_dash="dash", line_color="gray",
    annotation_text=f"Overall mean = {overall_mean:.2f}",
    annotation_position="bottom right"
)

fig.update_layout(showlegend=False)
fig.show()
```

health_label	count	avg_rating_mean	avg_rating_sd
Healthy	28792	4.613944	0.649445
Not healthy	52381	4.631640	0.635860

Step 3: Assessment of Missingness

In [70]:
```
interactions.isnull().sum()
```

Out[70]:
```
user_id        0
recipe_id      0
date           0
rating         0
review       169
dtype: int64
```

In [71]:
```
recipes.isnull().sum()
```

Out[71]:
```
name                1
id                  0
minutes             0
contributor_id      0
submitted           0
tags                0
nutrition           0
n_steps             0
steps               0
description        70
ingredients         0
n_ingredients       0
avg_rating       2609
is_healthy          0
calories            0
total fat           0
sugar               0
```

```
sodium                    0
protein                   0
saturated fat             0
carbohydrates             0
protein_per_100kcal      26
sugar_per_100kcal        26
recipe_age_years          0
calories_z                0
total fat_z               0
sugar_z                   0
sodium_z                  0
protein_z                 0
saturated fat_z           0
carbohydrates_z           0
avg_rating_floor       2609
dtype: int64
```

In [72]:

```python
# perform permutation tests to analyze the dependency of the missingness of this column on other columns
def perm_test_diff_means(y, x, n_perm=1000, seed=42):
    rng  = np.random.default_rng(seed)
    obs  = x[y == 1].mean() - x[y == 0].mean()
    extreme = np.sum([
        abs(rng.permutation(x)[y == 1].mean() - rng.permutation(x)[y == 0].mean()) >= abs(obs)
        for _ in range(n_perm)
    ])
    return obs, extreme / n_perm
```

In [73]:

```python
# 1 = missing indicator
recipes["miss_rating"] = recipes["avg_rating"].isna().astype(int)

# Candidate predictor A  (expected DEPENDENCE)
x_age = recipes["recipe_age_years"].to_numpy()

# Candidate predictor B  (expected INDEPENDENCE)
x_protein = recipes["protein_z"].to_numpy()

y_miss = recipes["miss_rating"].to_numpy()
# perm tests
obs_age,  p_age  = perm_test_diff_means(y_miss, x_age,     n_perm=1000, seed=1)
obs_prot, p_prot = perm_test_diff_means(y_miss, x_protein, n_perm=1000, seed=1)

print(f"Recipe age change= {obs_age:+.2f} years,  p = {p_age:.4e}")
print(f"Protein change= {obs_prot:+.2f},       p = {p_prot:.4f}")
```

```
Recipe age change= -0.75 years,  p = 0.0000e+00
Protein change= +0.03,       p = 0.1720
```

In [74]:

```
# 1 = missing
y_miss = recipes["avg_rating"].isna().astype(int).to_numpy()
x_age  = recipes["recipe_age_years"].to_numpy()

# observed difference in means (missing - observed)
obs = x_age[y_miss == 1].mean() - x_age[y_miss == 0].mean()

# permutation distribution
n_perm = 1000
rng     = np.random.default_rng(42)
diffs   = [
    rng.permutation(x_age)[y_miss == 1].mean() -
    rng.permutation(x_age)[y_miss == 0].mean()
    for _ in range(n_perm)
]
```

In [75]:

```
# plot
fig = px.histogram(
    pd.DataFrame({"Δ": diffs}),
    x="Δ",
    nbins=60,
    histnorm="probability",
)
fig.add_vline(x=obs, line_color="red")
fig.update_layout(xaxis_title="Permuted (years)", yaxis_title="Probability")
fig.show()
```

Step 4: Hypothesis Testing

In [76]:

```
#   H0: Healthy and Not-healthy recipes have the same mean rating.
#   H1: Healthy recipes have a HIGHER mean rating than Not-healthy recipes.

df = recipes.dropna(subset=["avg_rating", "is_healthy"]).copy()
df["is_healthy"] = df["is_healthy"].astype(int)     # 1 = healthy, 0 = not

healthy      = df.loc[df["is_healthy"] == 1, "avg_rating"]
not_healthy  = df.loc[df["is_healthy"] == 0, "avg_rating"]
obs_diff     = healthy.mean() - not_healthy.mean()
n_repetitions = 1000
differences   = []

for _ in range(n_repetitions):
    with_shuffled = df.assign(
        Shuffled_Rating = np.random.permutation(df["avg_rating"])
    )
    group_means = (
```

```
        with_shuffled
        .groupby("is_healthy")["Shuffled_Rating"]
        .mean()
    )
    diff = group_means.loc[1] - group_means.loc[0]
    differences.append(diff)
```

In [77]:
```
mean_ratings = (
    df.groupby("is_healthy")["avg_rating"]
      .mean()
)
observed_difference = mean_ratings[1] - mean_ratings[0]
observed_difference
```

Out[77]:
```
np.float64(-0.017696141422482548)
```

In [78]:
```
# empirical distribution of the mean differences
fig = px.histogram(
    pd.DataFrame(differences),
    x=0,
    nbins=50,
    histnorm="probability",
    title=(
        "Empirical Distribution of Mean-Rating Differences<br>"
        "(Healthy – Not-healthy)"
    ),
)

fig.add_vline(x=observed_difference, line_color="red")
fig.update_layout(xaxis_title="Permuted", yaxis_title="Probability",
                  margin=dict(t=60))
fig
```

In [79]:
```
p_val = (np.sum(np.array(differences) >= obs_diff) + 1) / (len(differences) + 1)
p_val
```

Out[79]:
```
np.float64(1.0)
```

Step 5: Framing a Prediction Problem

In [80]:
```
# I want to predict whether a food recipe is healthy/not healthy given its nutritional facts.
```

Step 6: Baseline Model

In [93]:
```
recipes.columns
```

Out[93]:
```
Index(['name', 'id', 'minutes', 'contributor_id', 'submitted', 'tags',
       'nutrition', 'n_steps', 'steps', 'description', 'ingredients',
       'n_ingredients', 'avg_rating', 'is_healthy', 'calories', 'total fat',
       'sugar', 'sodium', 'protein', 'saturated fat', 'carbohydrates',
       'protein_per_100kcal', 'sugar_per_100kcal', 'recipe_age_years',
       'calories_z', 'total fat_z', 'sugar_z', 'sodium_z', 'protein_z',
       'saturated fat_z', 'carbohydrates_z', 'avg_rating_floor',
       'miss_rating'],
      dtype='object')
```

In [94]:
```
# set train test
nutrition_cols = [
    'calories_z', 'total fat_z',
    'sugar_z', 'sodium_z', 'protein_z', 'saturated fat_z',
    'carbohydrates_z'
]
target_col = "is_healthy"

df = recipes.dropna(subset=nutrition_cols + [target_col]).copy()
X = df[nutrition_cols]
y = df[target_col].astype(int)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42, stratify=y
)
```

In [95]:
```
# LogisticRegression model
pipe = Pipeline([
    ("clf", LogisticRegression(
        solver="lbfgs",
        max_iter=1000,
        class_weight="balanced"
    )),
])
```

In [96]:
```
# fit the model and see metrics
pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)
y_prob = pipe.predict_proba(X_test)[:, 1]
```

```
acc  = accuracy_score(y_test, y_pred)
auc  = roc_auc_score(y_test, y_prob)

print(acc)
```

```
0.6163991167870144
```

Step 7: Final Model

In [85]:
```
recipes
```

Out[85]:

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_steps	steps	description	...	reci
0	1 brownies in the world best ever	333281	40	985201	2008-10-27	[60-minutes-or-less, time-to-make, course, mai...	[138.4, 10.0, 50.0, 3.0, 3.0, 19.0, 6.0]	10	['heat the oven to 350f and arrange the rack i...	these are the most; chocolatey, moist, rich, d...	...	
1	1 in canada chocolate chip cookies	453467	45	1848091	2011-04-11	[60-minutes-or-less, time-to-make, cuisine, pr...	[595.1, 46.0, 211.0, 22.0, 13.0, 51.0, 26.0]	12	['pre-heat oven the 350 degrees f', 'in a mixi...	this is the recipe that we use at my school ca...	...	
2	412 broccoli casserole	306168	40	50969	2008-05-30	[60-minutes-or-less, time-to-make, course, mai...	[194.8, 20.0, 6.0, 32.0, 22.0, 36.0, 3.0]	6	['preheat oven to 350 degrees', 'spray a 2 qua...	since there are already 411 recipes for brocco...	...	
3	millionaire pound cake	286009	120	461724	2008-02-12	[time-to-make, course, cuisine, preparation, o...	[878.3, 63.0, 326.0, 13.0, 20.0, 123.0, 39.0]	7	['freheat the oven to 300 degrees', 'grease a ...	why a millionaire pound cake? because it's su...	...	
4	2000 meatloaf	475785	90	2202916	2012-03-06	[time-to-make, course, main-ingredient,	[267.0, 30.0, 12.0, 12.0, 29.0,	17	['pan fry bacon , and set aside on a paper	ready, set, cook! special edition contest	...	

						prepar...	48.0, 2.0]		tow...	entr...	
...	...	...	...	...	...	...	...	...	...	...	...
83777	zydeco soup	486161	60	227978	2012-08-29	[ham, 60-minutes-or-less, time-to-make, course...	[415.2, 26.0, 34.0, 26.0, 44.0, 21.0, 15.0]	7	['heat oil in a 4-quart dutch oven', 'add cele...	this is a delicious soup that i originally fou...	...
83778	zydeco spice mix	493372	5	1500678	2013-01-09	[15-minutes-or-less, time-to-make, course, pre...	[14.8, 0.0, 2.0, 58.0, 1.0, 0.0, 1.0]	1	['mix all ingredients together thoroughly']	this spice mix will make your taste buds dance!	...
83779	zydeco ya ya deviled eggs	308080	40	37779	2008-06-07	[60-minutes-or-less, time-to-make, course, mai...	[59.2, 6.0, 2.0, 3.0, 6.0, 5.0, 0.0]	7	['in a bowl , combine the mashed yolks and may...	deviled eggs, cajun-style	...
83780	cookies by design cookies on a stick	298512	29	506822	2008-04-15	[30-minutes-or-less, time-to-make, course, pre...	[188.0, 11.0, 57.0, 11.0, 7.0, 21.0, 9.0]	9	['place melted butter in a large mixing bowl a...	i've heard of the 'cookies by design' company,...	...
83781	cookies by design sugar shortbread cookies	298509	20	506822	2008-04-15	[30-minutes-or-less, time-to-make, course, pre...	[174.9, 14.0, 33.0, 4.0, 4.0, 11.0, 6.0]	5	['whip sugar and shortening in a large bowl , ...	i've heard of the 'cookies by design' company,...	...

83782 rows × 33 columns

In [86]:

```
# set train test
base_nutrition = [
    'calories_z', 'total fat_z',
    'sugar_z', 'sodium_z', 'protein_z', 'saturated fat_z',
    'carbohydrates_z'
]
```

```
# we add extra columns this time
engineered   = ["protein_per_100kcal", "sugar_per_100kcal"]
feature_cols = base_nutrition + engineered
target_col   = "is_healthy"

df = recipes.dropna(subset=feature_cols + [target_col]).copy()
X = df[feature_cols]
y = df[target_col].astype(int)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42, stratify=y
)
```

In [87]:

```
# define and train our model with GridSearchCV this time for hyperparameter tuning
gb = GradientBoostingClassifier(random_state=42)

param_grid = {
    "clf__n_estimators":   [100, 200, 400],
    "clf__learning_rate":  [0.05, 0.10, 0.20],
    "clf__max_depth":      [2, 3, 5],
    "clf__subsample":      [0.7, 1.0],
}

pipe = Pipeline(steps=[
    ("clf",  gb),
])

grid = GridSearchCV(
    pipe,
    param_grid,
    cv=2,
    scoring="roc_auc",
    n_jobs=-1,
    verbose=1,
)
```

In [88]:

```
grid.fit(X_train, y_train)

print(f"\nBest CV score (ROC AUC): {grid.best_score_:.4f}")
print("Best hyper-parameters:\n", grid.best_params_, "\n")
```

```
Fitting 2 folds for each of 54 candidates, totalling 108 fits

Best CV score (ROC AUC): 0.7509
Best hyper-parameters:
 {'clf__learning_rate': 0.05, 'clf__max_depth': 5, 'clf__n_estimators': 400, 'clf__subsample': 0.7}
```

In [91]:

```
# test and metrics
best_model = grid.best_estimator_
y_pred = best_model.predict(X_test)
y_prob = best_model.predict_proba(X_test)[:, 1]

acc  = accuracy_score(y_test, y_pred)
auc  = roc_auc_score(y_test, y_prob)

print(f"Test-set Accuracy:  {acc:.4f}")
print(f"Test-set ROC AUC:   {auc:.4f}\n")
```

```
Test-set Accuracy:  0.7192
Test-set ROC AUC:   0.7529
```

Step 8: Fairness Analysis

In [ ]:

```
# preparing data for fairness analysis
nutrition_cols = [
    'calories_z', 'total fat_z',
    'sugar_z', 'sodium_z', 'protein_z', 'saturated fat_z',
    'carbohydrates_z', "protein_per_100kcal", "sugar_per_100kcal",
]
target_col = "is_healthy"

# drop any rows with missing values in features, target, or "minutes"
df = recipes.dropna(subset=nutrition_cols + [target_col, "minutes"]).copy()
X = df[nutrition_cols]
y = df[target_col].astype(int)

# train tets split
X_train, X_test, y_train, y_test, idx_train, idx_test = train_test_split(
    X, y, df.index,
    test_size=0.20,
    random_state=42,
    stratify=y
)

# get the "minutes" values for the held-out test set
minutes_test = df.loc[idx_test, "minutes"].astype(float)

group_quick = minutes_test <= 30

# model prediction
y_true   = y_test.to_numpy()
y_pred   = best_model.predict(X_test)
```

```python
group_quick = (minutes_test <= 30).to_numpy()

# compute accuracy for the rows where mask == True
def group_accuracy(mask):
    return accuracy_score(y_true[mask], y_pred[mask])

obs_diff = group_accuracy(group_quick) - group_accuracy(~group_quick)

# permutation test
n_repetitions = 1000
differences   = []

for _ in range(n_repetitions):
    shuffled_mask = np.random.permutation(group_quick)
    diff = (
        group_accuracy(shuffled_mask) -
        group_accuracy(~shuffled_mask)
    )
    differences.append(diff)

print(differences[:10])

extreme_count = np.sum(np.abs(differences) >= abs(obs_diff))
p_value       = extreme_count / n_repetitions

print(f"\nObserved accuracy gap: {obs_diff:+.4f}")
print(f"Permutation p-value (n = {n_repetitions}): {p_value:.4f}")

# reject
```

```
[-0.0017675670393905163, -0.006592710076544215, 0.0035462073256996574, 0.0033046721272865787, 0.005236953714592096, -0.009496
693388612698, 0.0035462073256996574, 0.004512348119352527, 0.0042708129209392265, -0.008047482198133449]

Observed accuracy gap: -0.0455
Permutation p-value (n = 1000): 0.0000
```