Milestone 2 Submission

Caroline Amsbary [*1] **Emily Friedman** [*1] [2] **Suhanee Singh** [*1] [3 4]

1. Milestone 1

1.1. Dataset Overview

Local county health departments conduct inspections of restaurants and other retail food establishments to ensure that employees practice safe food handling and that kitchen facilities meet required standards. The data we will be using is **Restaurant Inspection Scores in the state of Washington**. We have chosen to focus on **King County**, as this is the county where Seattle is located, yielding results from a higher population.

The dataset contains Health Department inspection results for food service establishments in King County from 2006 to the present. The information is organized by Business, Inspection Date, and Violation. Each row represents a single inspection.

Source: https://kingcounty.gov/en/dept/dph/health-safety/food-safety/search-restaurant-safety-ratings#/

1.2. Variable Description

To clean this data in preparation for analysis, we can start by creating a new table with only the relevant variables. As of right now, here are the variables provided in the dataset:

- **Name:** Name of the restaurant
- **Program Identifier:** Name of the restaurant (same as Name)
- **Inspection Date:** Date that inspection took place
- **Description:** Describes the restaurant's seating capacity and Risk Category, which is associated with the level of food safety risk (I being lowest risk, IV being highest)
- **Address:** Restaurant's address
- **City:** Restaurant's city
- **Zip Code:** Restaurant's zip code
- **Phone:** Restaurant's phone number
- **Longitude:** Restaurant's longitude
- **Latitude:** Restaurant's latitude
- **Inspection Business Name:** Name of the restaurant (same as Name)
- **Inspection Type:** What kind of inspection was performed
- **Inspection Score:** The score given by the inspector (0 is ideal; higher is worse)
- **Inspection Result:** For routine inspections, either "Satisfactory" or "Unsatisfactory"; for consultation/education inspections, either "Complete" or "Incomplete"
- **Inspection Closed Business:** "True" if the inspection caused closure; "False" if the restaurant remained open
- **Violation Type:** "Red" for high-risk violations and "Blue" for low-risk violations
- **Violation Description:** Detailed explanation of the food safety issue identified
- **Violation Points:** Discrete points (0–30) indicating severity of violation
- **Business_ID:** Unique identifier for each establishment
- **Inspection_Serial_Num:** Unique tracking number for each inspection
- **Violation_Record_ID:** Unique identifier for a specific violation
- **Grade:** Numeric code for severity: 1 = Critical, 2 = Intermediate, 3 = Core

1.3. Selected Variables for Analysis

For our analysis, we have determined that we only need the following variables:

- Name
- Inspection Date
- Description

- Inspection Type
- Inspection Score
- Inspection Result
- Inspection Closed Business
- Violation Type
- Violation Description
- Violation Points
- Grade

1.4. Data Cleaning Process

To clean the data, we will replace missing data with `NaN`, although as far as we have seen, there is no missing data. This is hard to verify just from visual inspection, since there are over 277,000 rows. Additionally, we will ensure that numerical variables are stored as `int` or `float`, which is essential for statistical summaries and plotting.

1.5. Project Goal

We aim to estimate the probability that a restaurant's next inspection in King County, Washington will include at least one **critical ("red") violation**. The model is intended to support risk-based scheduling and targeted education by identifying establishments most likely to require intervention.

2. Milestone 2: Preliminary Methods

2.1. Method Overview

Per Milestone 1, the aim of this project is to examine the restaurants in King County, Washington, in particular their previous inspection results, to predict the likelihood that the next inspection will include at least one critical violation. We plan to achieve this by using the past inspection results to explore patterns and write predictive models using some of the key methods covered in class, specifically kNN, linear models, and decision trees. Our approach will follow this general overview, all of which are discussed in more detail below:

1. Step 1: Data wrangling and cleaning
2. Step 2: EDA and visualization of cleaned data
3. Step 3: Feature engineering and selection
4. Step 4: Model training
5. Step 5: Model validation

2.2. Data Wrangling and Cleaning

Again, as mentioned in Milestone 1, we will begin by creating a new dataset that includes only the relevant variables for our analysis:

- Name
- Inspection Date
- Description (includes Risk Category)
- Inspection Type
- Inspection Score
- Inspection Result
- Inspection Closed Business
- Violation Type
- Violation Description
- Violation Points
- Grade
- Longitude
- Latitude

**Note that we added longitude and latitude to the list of relevant variables, as we decided this is something we'd like to take into consideration.*

Cleaning steps will include:

- Replacing missing data with NaN values and confirming completeness.
- Converting numeric variables such as Inspection Score, Violation Points, and Grade to numeric data types (int or float).
- Ensuring that Inspection Date is properly formatted as a datetime object.
- Removing duplicate or irrelevant records (for example, entries where inspection data are incomplete).
- Encoding categorical variables (for example, Inspection Type, Violation Type, Inspection Result) using one-hot encoding.

This cleaned dataset will serve as the foundation for exploratory analysis and model development.

2.3. Exploratory Data Analysis and Visualization

We will use EDA techniques to understand the distribution and discover which relationships, trends, or anomalies exist in the data. Specifically, we plan to:

- Plot the distribution of Inspection Scores and Violation Points to assess their skewness and range.
- Examine how Risk Category (from Description) relates to the frequency of red violations.
- Use bar charts to show the proportion of inspection results ("Satisfactory" vs. "Unsatisfactory") by restaurant type.
- Explore geographic patterns using longitude and latitude to see whether certain areas of King County experience more frequent red violations.

These visualizations will help identify potential predictive features and highlight patterns that could inform our model choices.

2.4. Feature Engineering

In this step, we will perform feature engineering to simplify the model, reduce overfitting, and speed up our computations. From the raw inspection records, we will generate higher-level features such as:

- Average past inspection score: captures a restaurant's overall performance trend
- Number of previous inspections: reflects inspection frequency, which may correlate with compliance consistency
- Proportion of inspections with red violations: indicates the severity or recurrence of health risks
- Average violation points per inspection: provides a measure of violation intensity
- Time since last inspection: represents temporal risk, as longer gaps may be associated with increased likelihood of violations

These features will help capture each restaurant's inspection history and risk level, providing more in-depth and relevant information for our predictive models.

POTENTIAL CODE for feature engineering:

```
import pandas as pd

# Assume df is the raw inspection dataset
df['Inspection Date'] = pd.to_datetime(df['
    ↪ Inspection Date'])

# Sort by date to ensure correct temporal
    ↪ ordering
df = df.sort_values(['Name', 'Inspection
    ↪ Date'])

# Group by restaurant name
features = df.groupby('Name').apply(lambda
    ↪ x: pd.Series({
    'avg_past_score': x['Inspection Score'
        ↪ ].mean(),
    'num_prev_inspections': x['Inspection
        ↪ Date'].nunique(),
    'prop_red_violations': (x['Violation
        ↪ Type']
        .str.contains('red', case=False, na
            ↪ =False)).mean(),
    'avg_violation_points': x['Violation
        ↪ Points'].mean(),
    'time_since_last_insp': (x['Inspection
        ↪ Date'].max()
        - x['Inspection Date'].min()).days
})).reset_index()
```

2.5. Modeling Approach

In order to predict the likelihood of future red violations, we will set up a binary classification problem, where the response variable Y, is defined as:

$$Y = \begin{cases} 1, & \text{if the next inspection contains} \geq 1 \text{ violation} \\ 0, & \text{otherwise} \end{cases}$$

We plan to train and compare the following models:

- Linear Models (Logistic Regression): To serve as a simple and easily interpretable baseline for predicting the probability of a red violation. Logistic regression will allow us to examine the weight of each feature and identify the most significant predictors.
- k-Nearest Neighbors (k-NN): To predict a restaurant's risk by comparing it to other restaurants that are most similar in the data. In other words, it looks at the "k" closest restaurants and uses their inspection outcomes to make a prediction. We will try different values of k to see how changing the number of neighbors affects the model's accuracy and recall.
- Decision Trees: Predict outcomes by splitting the data into branches based on different features, like inspection type or risk category. This will show how certain combinations of factors can lead to red violations. Decision trees are also easy to understand because they create a clear visual diagram that shows how decisions are made step by step.

Additionally, we will use unsupervised clustering (such as k-means) as an exploratory step to see whether restaurants naturally form groups with distinct inspection patterns (like low-risk vs. high-risk clusters).

2.6. Model Training and Validation

We will split the data into training (70%) and testing (30%) groups. To avoid data leakage, all inspections from the same restaurant will be kept in only one of these sets. For each model, we will:

- Use cross-validation on the training data to confirm the best settings (for example, the number of neighbors in k-NN or the maximum depth in a decision tree)
- Test how well the model performs on the test set using accuracy, precision, recall, F1-score, and ROC-AUC

F1-score = balance between precision and recall

ROC-AUC = Receiver Operating Characteristic and Area Under Curve, tell us how well the model separates the two possible outcomes

After this, we will compare all models to find which one provides the best balance between accuracy and ease of interpretation.

2.7. Preliminary Validation Plan

Some ways we will visualize performance is through confusion matrices to show how many inspections were correctly and incorrectly classified. Another way is through feature importance plots (for decision trees and linear models) to interpret which variables contribute most strongly to predictions. We also hope to test the models on data from different time periods (e.g., pre-2018 vs. post-2018) to evaluate their stability over time.

2.8. Future Work

Looking forward to the Final Report we plan to refine our models by:

- Incorporating temporal trends (seasonality or time since last inspection).
- Testing models that combine multiple techniques.
- Building visual dashboards to display predicted risk levels by restaurant or neighborhood.

2.9. Summary

Our preliminary methodology integrates the data wrangling, EDA, visualization, and many other methods introduced in class. By combining interpretable models such as logistic regression with flexible approaches like k-NN and decision trees, we aim to identify reliable predictors of critical violations and support food safety efforts in King County, Washington.