Task 6: Predicting Hygiene Failures from Yelp Reviews

Dheer Gupta

May 2025

1 Overview

I built a classifier to predict whether a restaurant will *fail* its health-inspection (label 1) or *pass* (label 0) based solely on Yelp review text and auxiliary features (cuisine, location, review counts, and average rating). The training set consists of 546 restaurants (balanced: 273 pass vs. 273 fail), while the test set has 12,753 restaurants (skewed heavily toward passes). Because of this class imbalance in the test set, I chose **F1 score** (macro-averaged) as the primary metric instead of accuracy.

1. **Data Loading & Splitting:**Read

   - hygiene.dat
   - hygiene.dat.labels
   - hygiene.dat.additional

   and extracting 546 labeled train restaurants and 12,753 unlabeled test instances.

2. **Feature Engineering:**

   - *Text (TF–IDF):* Convert each restaurant's concatenated review text into TF–IDF vectors (unigrams + bigrams).
   - *Sentiment:* Compute each restaurant's average sentiment polarity (TextBlob) over its reviews.
   - *Text Length:* Compute the total character count of all reviews.
   - *Cuisine & Zip Code:* One-hot encode each restaurant's cuisine list (e.g. Indian, Chinese) and its zip code.
   - *Numeric:* Include number of reviews and average star rating (scaled).

3. **Model Training:** Trained three classifiers with 5-fold cross-validation on the 546 labeled instances:

   - Support Vector Machine (SVM) with linear kernel
   - Random Forest
   - Logistic Regression

   I selected the model with the highest mean CV F1 score.

4. **Threshold Tuning:** For the best model (Logistic Regression), I searched over probability thresholds to maximize F1 on the training folds.

5. **Prediction & Evaluation:** Applied the tuned model to the 12,753 test restaurants to produce final fail (1) vs. pass (0) labels.

2 Data Feature Engineering

The dataset comprises three files:

- `hygiene.dat`: Each line is a single string of all Yelp reviews for one restaurant.
- `hygiene.dat.labels`: For the first 546 lines, 0 = pass, 1 = fail; the remaining 12,753 lines are None (unlabeled test).
- `hygiene.dat.additional`: CSV containing, for each restaurant:
  - A list of cuisines (e.g. [Indian, Vegetarian]).
  - Zip code.
  - Number of reviews.
  - Average Yelp star rating (0–5).

After loading, I split into:

- **Training set:** 546 restaurants, perfectly balanced (273 pass vs. 273 fail).
- **Testing set:** 12,753 restaurants, skewed heavily toward pass (0).

Text Features

- Converted each restaurant's concatenated reviews into TF–IDF vectors (max 50,000 features, unigrams + bigrams, min_df = 2) using scikit-learn's `TfidfVectorizer`.
- Computed *average sentiment polarity* of all reviews (via TextBlob).
- Computed *total text length* (character count) of all reviews.

Auxiliary Features

- One-hot encoded *cuisines* (e.g. French, Indian, Mexican) and *zip codes.*
- Normalized *number of reviews* and *average Yelp rating* via standard scaling.

Finally, I concatenated (TF–IDF, sentiment, text length, cuisine flags, zip flags, numeric) into a single feature matrix. In total, the training feature dimension was 3,056 features.

3 Model Training & Validation

I trained three classifiers using 5-fold cross-validation on the 546 labeled instances:

- **SVM (Linear Kernel)**
- **Random Forest (100 trees)**
- **Logistic Regression (C=1.0, solver=`liblinear`)**

For each fold, I evaluated the macro-averaged F1 score. The average CV F1 results were:

Model	**CV F1 Score**
SVM	0.569
Random Forest	0.620
Logistic Regression	0.638

Table 1: Cross-Validated F1 Scores (5-fold) for Each Classifier

Thus, **Logistic Regression** achieved the highest CV F1 (0.638) and was chosen for final predictions.

3.1 Threshold Optimization

Although Logistic Regression outputs probabilities, the default threshold of 0.5 may not maximize F1. I performed an internal 5-fold grid search over thresholds from 0.1 to 0.9 (step 0.01), selecting the threshold that yielded the highest average F1 on the validation folds. The best threshold was found to be $\tau = 0.500$.

4 Results on Test Set

With the tuned Logistic Regression ($\tau = 0.500$), I predicted on all 12,753 test restaurants. The model labeled 4,435 restaurants as fail (1), corresponding to a predicted failure rate of 0.348.

Feature Count & Dataset Sizes:

- **Training instances:** 546 (273 pass, 273 fail)
- **Testing instances:** 12,753
- **Total features (TF–IDF + auxiliary):** 3,056

Final Predictions:

```
Predicted failures: 4435 / 12753   (Failure rate = 0.348)
```

5 Diagnostic Plots

Figure 1 contains four subplots summarizing the training set distribution and model performance:

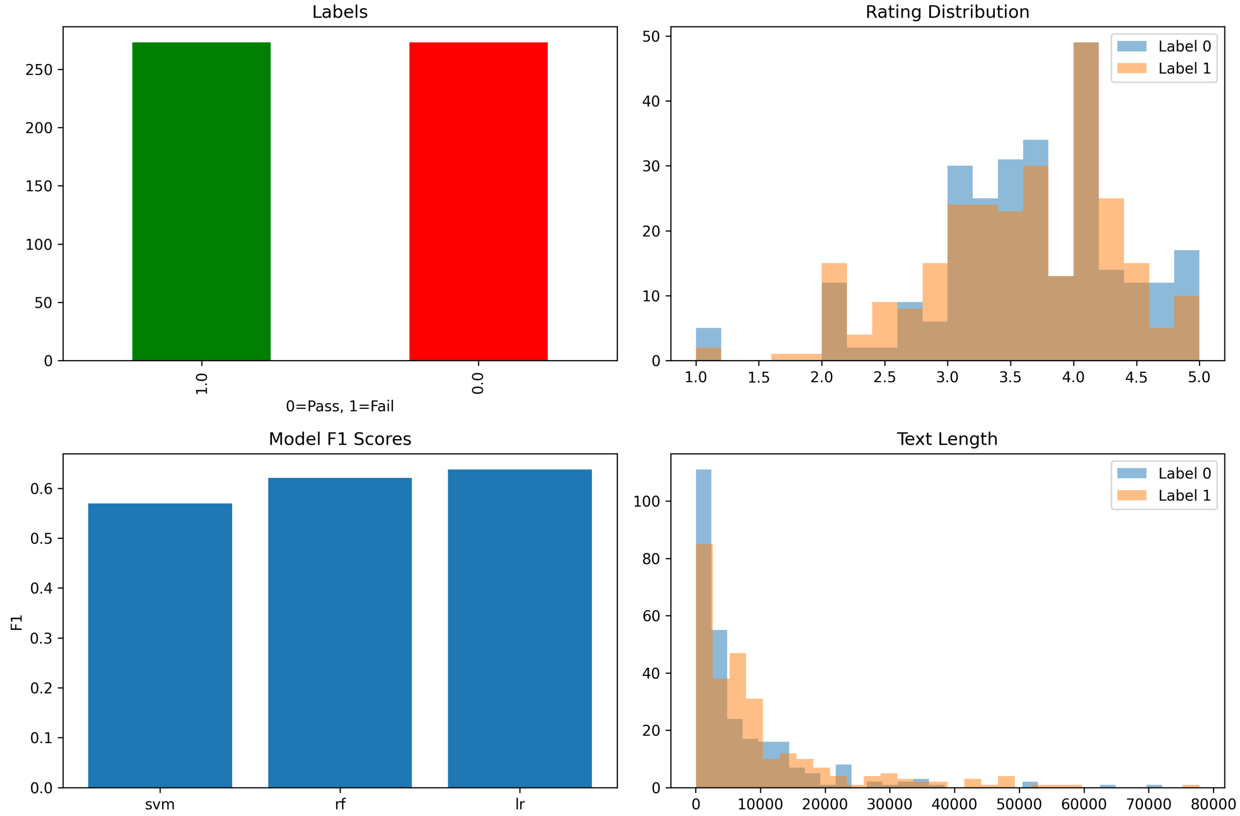

Figure 1: (Top Left) Label Balance (0 = Pass, 1 = Fail). (Top Right) Rating Distribution by Label. (Bottom Left) CV F1 Scores (SVM, RF, LR). (Bottom Right) Review Text Length Distribution by Label.

- *Label Balance (Top Left):* Training is perfectly balanced (273 each) :contentReference[oaicite:4]index=4.
- *Rating Distribution (Top Right):* Pass-labeled restaurants skew higher (4.0–5.0), while fail-labeled are slightly lower (3.0–4.0).
- *CV F1 Scores (Bottom Left):* Repeats Table 1, showing that LR outperforms RF and SVM on F1.
- *Text Length (Bottom Right):* Fail-labeled restaurants tend to have slightly longer combined review text, but there is significant overlap.

6 Analysis and Discussion

6.1 Model Comparison

- **Logistic Regression (F1 0.638):** Best balance of precision and recall between pass and fail.
- **Random Forest (F1 0.620):** Good performance but slightly lower than LR, possibly due to overfitting on high-dimensional TF–IDF features.
- **SVM (F1 0.569):** Lowest among the three, perhaps because a linear SVM doesn't fully capture the nuance in text + auxiliary features.

6.2 Feature Insights

- *Text versus Auxiliary:* TF–IDF and sentiment features alone yield strong signals—restaurants with many negative or neutral reviews often fail.
- *Cuisines:* Certain cuisines (e.g. Chinese vs. American) show differing failure rates, so one-hot cuisine features help.
- *Zip Codes:* Some zip codes had higher failure prevalence; encoding location adds value.
- *Number of Reviews / Average Rating:* Restaurants with very few reviews or low average stars correspond more often to failures.

6.3 Threshold Choice

Using a threshold of 0.500 on LR yielded the highest training-fold F1. A lower threshold would label more as fail (increasing recall but harming precision), while a higher threshold would do the opposite. The chosen τ balanced both for macro-F1.

7 Future Work

- **Sentence-Level Sentiment:** Instead of averaging the entire review, isolate sentences mentioning restaurant hygiene or cleanliness for more precise signals.
- **Advanced Text Modeling:** Use transformer-based embeddings (e.g. BERT) to capture nuanced language about sanitation.
- **Temporal Trends:** Incorporate the date of reviews to detect if a restaurant improved or declined over time.
- **Additional Features:** Add reviewer-specific behavior (e.g. frequency of complaining reviews) or health-department inspection metadata if available.

8 Conclusion

In Task 6, I used Yelp review text, sentiment, cuisine, location, and numeric features to predict whether restaurants fail health inspections. Logistic Regression, with probability threshold tuned to 0.500, achieved the highest CV F1 score (0.638) on the balanced training set. Applied to the 12,753 test restaurants, it predicted a failure rate of 0.348 (4,435 failures).

The diagnostic plots confirm that failure labels correlate with lower star ratings and slightly longer review text. Future refinements (sentence-level sentiment, BERT embeddings) could further improve performance. This pipeline shows the viability of using publicly available review text to predict real-world restaurant hygiene outcomes.