AN ILLUSTRATED, HANDS-ON GUIDE

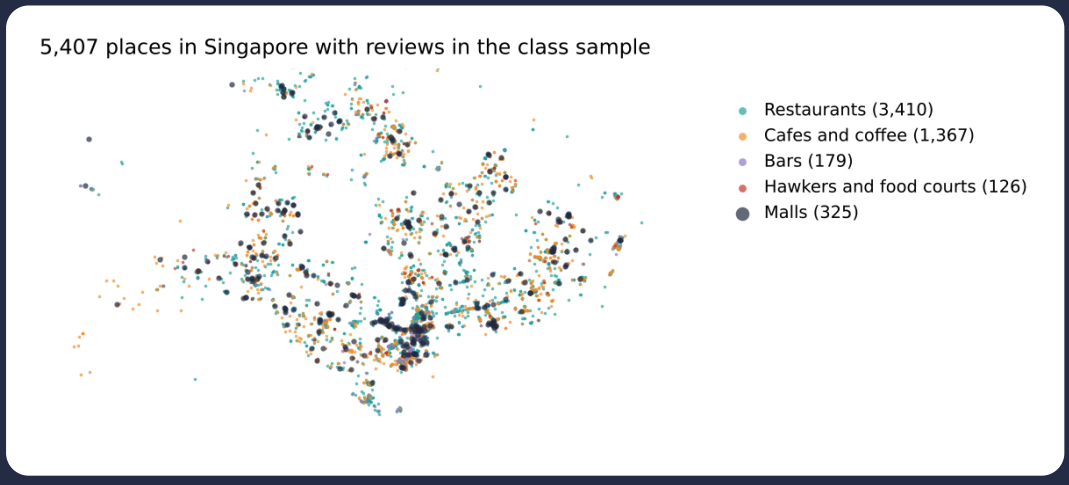
Reading the City from Reviews

A friendly guide to turning what people write about places into data, with real Singapore reviews

40,000 real Google Maps reviews of 5,407 places, six labs in Python, word lists, a text model, a comfort lens, maps and a small recommender.

Eight short chapters · Six labs · Three challenges · Cheat sheet

Prompted and directed by Mahmoud A.
Written and illustrated by Claude (Anthropic)



Prepared for the 2026 CCS: Urban AI + Agent Workshop · Harbin Institute of Technology · November 2026

How to Use This Guide

Read it in order: every chapter ends with something to try.

Every day, people write millions of short reviews of cafes, malls, hawker centres and parks. Most of these reviews talk about food and service, but many also say something about the place itself: **too hot, too loud, too crowded, cosy, quiet, spacious**. This guide turns that text into data about the city, using real Singapore reviews.

YOU WILL BE ABLE TO

- Load and explore a real review dataset.
- Find the words that go with good and bad places.
- Train a text model and read its errors.
- Build a comfort lens from word lists, and check it.
- Map comfort signals across a city.
- Rank places by comfort, not by stars.

WHAT YOU NEED

- Python 3 with `pandas`, `scikit-learn`, `matplotlib`.
- `reviews_lab.py` (or the notebook) and the two data files `sg_places.csv`, `sg_reviews.csv` from your instructor.
- About 10 seconds of computer time.

CONTENTS

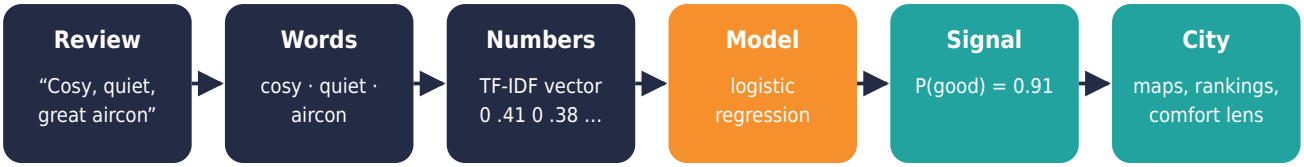
1. What reviews tell us about cities
2. The data · Lab 1
3. Words that matter · Lab 2
4. From text to a score · Lab 3
5. A comfort lens · Lab 4
6. Where and when · Lab 5
7. Recommend comfortable places · Lab 6
8. Beyond words, and toolbox

About the data. Real public Google Maps reviews of places in Singapore, collected in 2024–2025 for research. The class sample has 40,000 reviews (at most 12 per place), with no reviewer names. Use it for learning only, and do not republish it: the reviews belong to their authors and to Google's terms. Every number in this guide comes from running `reviews_lab.py` on this sample.

1 What Reviews Tell Us about Cities

Millions of free, unprompted, located opinions.

Urban researchers used to learn how people feel about places through surveys: slow, expensive, and small. Reviews are different. Nobody asked these people to write; they wrote because something was worth saying. Each review comes with a place, a time and often a score, which makes it a **located opinion**.



Reviews are good at	Reviews are bad at
Scale: thousands of places, many years	Who writes: younger, online, often tourists
Detail: words people actually use	Topic: mostly food and service, rarely comfort
Sub-scores: food, service, atmosphere	Time: dates are often “a year ago”
Comparing types of places	Truth: some reviews are fake or paid

IN ONE LINE

A review is a short, located opinion. Many reviews together become a sensor for how places feel, as long as you remember who is holding the sensor.

DISCUSS

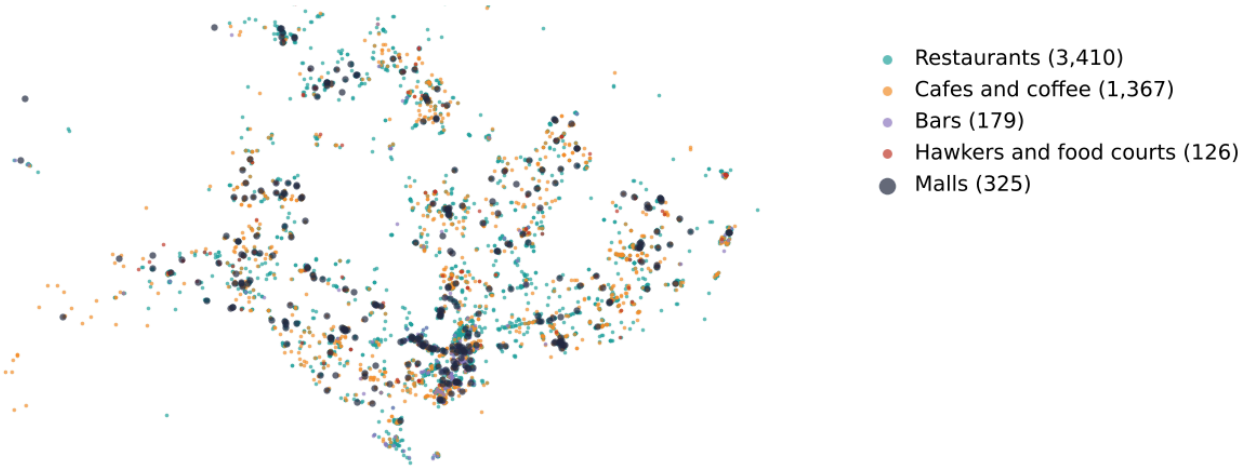
Think of a place near you. What would its reviews say about heat, noise or crowding? Would the people who write reviews be the same people who use the place most?

2 The Data

Lab 1 · 5,407 places, 40,000 reviews.

There are two tables. **Places** has a name, a category, a Google star rating and coordinates. **Reviews** has the text, a date, and up to three sub-scores that the reviewer chose to give: food, service and atmosphere. A few reviews also have a noise label and a price per person.

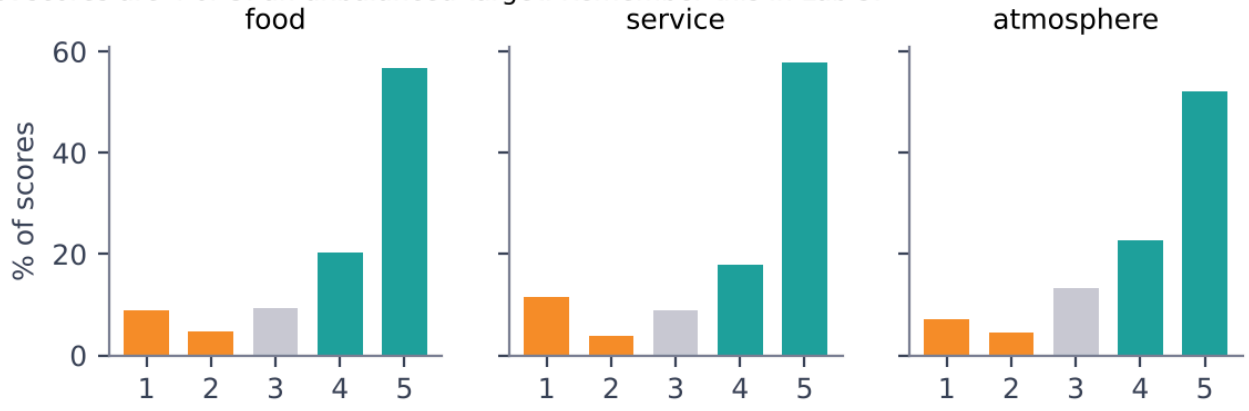
5,407 places in Singapore with reviews in the class sample




```
places  = pd.read_csv('sg_places.csv')                # name, category, rating, lat, lon
reviews = pd.read_csv('sg_reviews.csv', parse_dates=['date'])
df = reviews.merge(places, on='place_id')             # one row per review, with its place
df.atmosphere.value_counts()                          # sub-scores 1..5, or empty
```

```
places: 5407 | reviews: 40000 | from 2012-05 to 2025-05
top categories: {'Cafe': 5409, 'Restaurant': 4530, 'Coffee shop': 4309, 'Shopping mall': 2319,
'Chinese restaurant': 2054, 'Indian restaurant': 1351}
median review length: 199 characters
food       scores given in 52% of reviews | mean 4.11 | share of 1-2 stars 14%
service    scores given in 53% of reviews | mean 4.07 | share of 1-2 stars 15%
atmosphere scores given in 52% of reviews | mean 4.08 | share of 1-2 stars 12%
example: Had community gathering on the second floor of the restaurant and later dinner with
friends on the restaurant. It was a superb place for hangout. Tried the Tom  ...
```

Most scores are 4 or 5: an unbalanced target. Remember this in Lab 3.



3 Words That Matter

Lab 2 · which words separate good places from bad ones?

Before training any model, look at the words. Split reviews into two groups, here atmosphere = 5 and atmosphere ≤ 2, and ask which words are much more common in one group than in the other. The **log-odds** score does exactly that; adding 1 to every count stops rare words from exploding.

```
def log_odds(texts_a, texts_b, min_count=30):
    cv = CountVectorizer(stop_words='english', min_df=min_count)
    X = cv.fit_transform(pd.concat([texts_a, texts_b])) > 0       # does the review use the word?
    a = X[:len(texts_a)].sum(0) + 1;  b = X[len(texts_a):].sum(0) + 1
    return np.log(a / len(texts_a)) - np.log(b / len(texts_b))    # > 0: typical of group a
```

```
more common when atmosphere = 5  : refreshing, amazing, beautifully, balanced, cozy, impeccable,
loved, inviting, beautiful, perfect, incredible, wonderful
more common when atmosphere <= 2 : disgusting, lousy, rudely, terrible, horrible, refund,
unacceptable, pathetic, worst, uncomfortable, ridiculous, rude
```

The “bad atmosphere” side is full of words about **people** (rude, rudely, refund), not about the room. Reviewers often use the atmosphere score to punish a bad experience in general. This is the first lesson about review data: **a label means what reviewers make it mean**, not what the platform intended.

TRY THIS · EXERCISE 1

1. Compare `Hawker Stall` reviews with `Shopping mall` reviews instead. Which words come out? Are they about the place or the food?
2. Change `min_count` from 30 to 5. What kind of words appear, and why are they less trustworthy?

4 From Text to a Score

Lab 3 · a text model, and why accuracy can lie.

To give text to a model, we turn each review into numbers. **TF-IDF** gives every word (and every pair of words) a weight: high if the word is frequent in this review but rare in all reviews. A **logistic regression** then learns one weight per word and adds them up to a probability.

```
d = df[df.atmosphere.isin([1, 2, 4, 5])];  y = (d.atmosphere >= 4)
tr, te = train_test_split(d, test_size=0.25, stratify=y)
vec = TfidfVectorizer(ngram_range=(1, 2), min_df=3, sublinear_tf=True)
clf = LogisticRegression(max_iter=2000, class_weight='balanced')
clf.fit(vec.fit_transform(tr.text), tr.atmosphere >= 4)
clf.predict_proba(vec.transform(['Very stuffy and noisy, no aircon']))
```

```
train 13413 | test 4471 | always-guess-good baseline 0.86 | model accuracy 0.92 | balanced
accuracy 0.89
words pushing toward GOOD: good, nice, great, friendly, delicious, love, amazing, excellent, well,
best
words pushing toward BAD : not, no, rude, bad, poor, terrible, worst, don, waited, attitude
  P(good atmosphere) = 0.91  <- "Cosy little cafe, quiet music, great for working."
  P(good atmosphere) = 0.08  <- "Very stuffy and noisy, no aircon, tables were dirty."
  P(good atmosphere) = 0.42  <- "Food was ok but the place is cramped and hot."
```



Atmosphere: good (4–5) vs poor (1–2), 4,471 test reviews. Balanced accuracy = mean of t

86% of atmosphere scores are good, so a model that always says “good” already gets 0.86. The text model reaches 0.92, and its **balanced accuracy** (the average of how often it catches good and how often it catches bad) is 0.89. Always report the baseline and a balanced score for unbalanced targets.

```
: baseline 0.83 | accuracy 0.94 | balanced 0.92
same model for food    : baseline 0.85 | accuracy 0.93 | balanced 0.93
```

5 A Comfort Lens

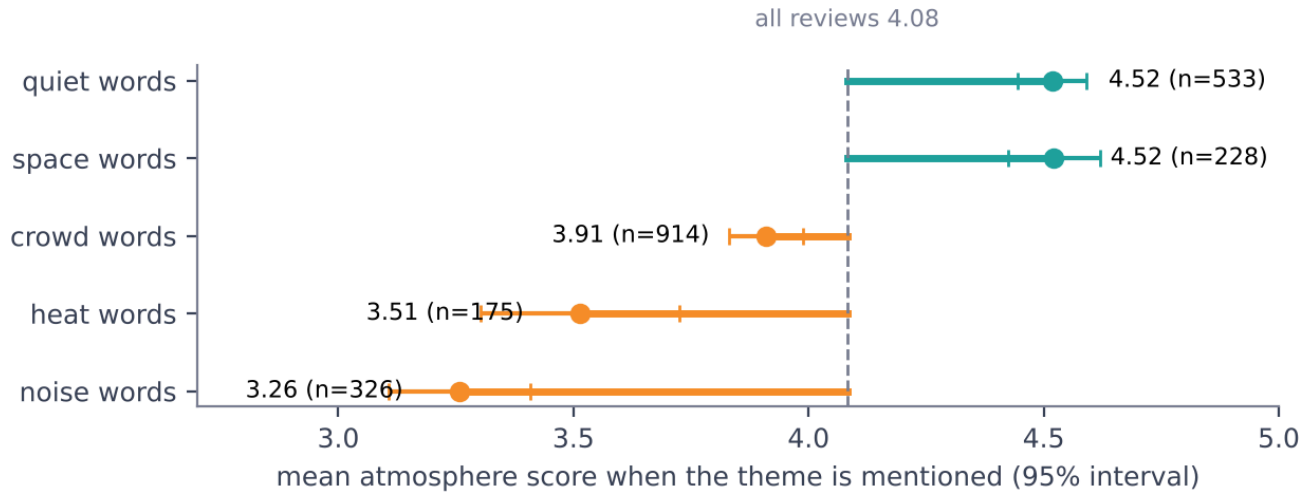
Lab 4 · word lists, and how to check them.

The model learns what reviewers like in general. For a city question, we want specific themes: heat, noise, crowding. A simple, transparent tool is a **word list** (a regular expression) for each theme. Word lists are easy to explain, but they must be checked.

```
COMFORT = {
  'heat':  r"\b(?:stuffy|humid|sweat\w*|hot inside|too warm|no air ?-?con\w*|poor
ventilation|...)",
  'noise': r"\b(?:noisy|loud|noise|hard to hear|deafening|blasting)",
  'quiet': r"\b(?:quiet|peaceful|calm|serene|tranquil)",
  'crowd': r"\b(?:crowded|packed|long queue|cramped|no seats|...)" }
for k, pat in COMFORT.items():
    df[k] = df.text.str.lower().str.contains(pat, regex=True)
```

```
reviews containing "hot": 1386 | of those, 89% also talk about food or drinks
   e.g. ['erved to us. the steak is not hot or warm.', ''s well air-conditioned for a hot day like
35 degrees and spaci', 'p which was tangy, spicy 🌶 & hot; to the prawn 🦐 vermicelli wh']
share of reviews per comfort theme: {'heat': '0.7%', 'cool': '1.8%', 'noise': '1.2%', 'quiet':
'2.4%', 'crowd': '4.3%', 'space': '1.1%'}
check against 790 reviews with a noise label: "noise" words appear in 6% of loud vs 0% of quieter
places; "quiet" words in 4% of quieter vs 0% of loud
```

Two checks. First, “hot” is a trap: 89% of reviews with “hot” talk about food or drinks, so it is not in the heat list. Second, compare the list with labels people gave: the noise words appear in 6% of reviews of loud places and in 0% of reviews of quieter ones. The list is **precise** (when it fires, it is right) but has **low recall** (it misses most loud places, because most reviewers do not mention noise).



When a reviewer mentions noise, the atmosphere score drops from 4.10 to 3.26; heat costs about 0.6 stars. Quiet and space add about 0.4.

L4

Comfort by Type of Place

People complain relative to what they expect.

```
Hawker Stall   heat  0.5% | aircon/cool  0.6% | crowd  5.7% | n=776
  Food court     heat  0.6% | aircon/cool  1.7% | crowd  6.1% | n=179
  Coffee shop    heat  0.8% | aircon/cool  1.5% | crowd  3.8% | n=4309
  Cafe           heat  0.9% | aircon/cool  2.5% | crowd  5.2% | n=5409
  Restaurant     heat  0.8% | aircon/cool  2.0% | crowd  3.2% | n=4530
  Shopping mall  heat  0.8% | aircon/cool  2.3% | crowd  5.6% | n=2319
```

Hawker centres are open-air and rarely air-conditioned, yet their reviews mention heat **less** than cafes and malls (0.5% against 0.8–0.9%). The full dataset of 437,619 reviews shows the same: 0.51% for hawker stalls against 0.86% for cafes and 0.96% for malls. People complain about heat when they expected to be cool. A review is a measure of the gap between **expectation and experience**, not of temperature.

WATCH OUT

Never read the share of complaints as a measure of the thing complained about. A mall with many heat complaints may be cooler than a hawker centre with none. To measure heat, use sensors or a climate model; use reviews to measure how people feel about it.

TRY THIS · EXERCISE 2

1. Add the words “fan” and “fans” to the cool list. What happens to hawker centres? Read 10 matching reviews before you decide to keep the change.
2. Write a `smell` list (smelly, smoke, oily air…). Which type of place has the most mentions?

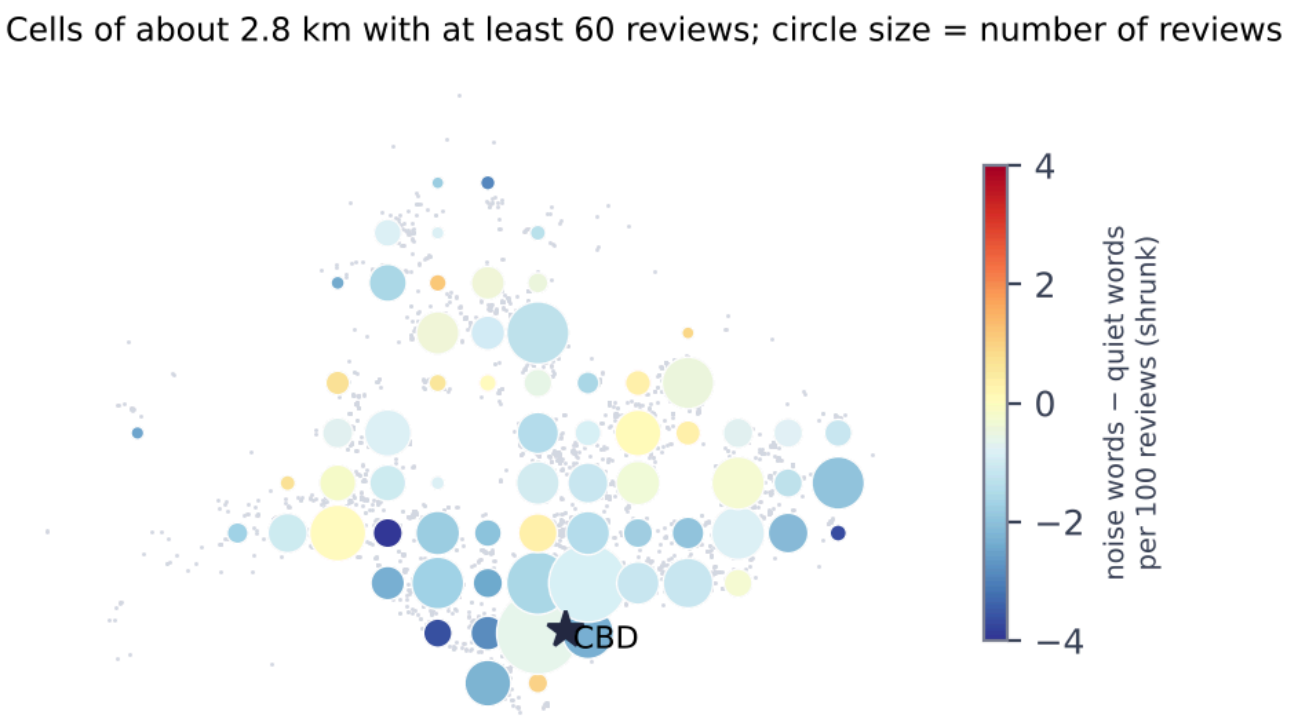
CHALLENGE A · A BETTER HEAT DETECTOR

Label 200 reviews that mention heat-related words by hand (real heat complaint or not). Train a small model on your labels and compare its precision and recall with the word list.

6 Where and When

Lab 5 · maps, rings, and a trap in the dates.

Each review sits at a place with coordinates, so every theme can be mapped. Below, the city is cut into cells of about 2.8 km; each cell shows noise words minus quiet words per 100 reviews, shrunk toward zero when a cell has few reviews.




```
heat complaints by month (%): 1:0.5 2:0.6 3:1.0 4:1.0 5:0.7 6:0.5 7:0.3 8:0.7 9:0.9 10:0.9 11:0.7
12:0.7
by distance from the CBD:
          heat  noise  crowd  quiet      n
ring
0-3 km     0.7    1.8    4.5    3.0   6544
3-6 km     0.7    1.4    4.0    3.1   5021
6-10 km    0.8    1.3    5.0    2.6   7923
10+ km     0.7    1.0    4.1    2.0  20504
```

Noise words fall steadily with distance from the CBD (1.8% to 1.0%), and quiet words are also lower far out, because suburban reviews talk more about food and less about the place.

WATCH OUT · THE DATE TRAP

94% of dates fall on the 28th of a month. Google shows “a year ago” or “3 months ago”, and the scraper turned that into a date. Month-by-month patterns are therefore unreliable: this sample peaks in March–April, while the full data peaks in June and September. Small signals plus fuzzy dates means: **do not publish a seasonal finding from this data**.

7 Recommend Comfortable Places

Lab 6 · rank by comfort, not by stars.

The recommendation guide of this series predicted ratings. Here we build a simpler, transparent recommender: a **comfort score** per place from the comfort lens. Places with few reviews are shrunk toward zero, as if each had five extra neutral reviews.

```
g = df.groupby('place_id').agg(n=('text','size'), quiet=('quiet','sum'), cool=('cool','sum'),
                               noise=('noise','sum'), heat=('heat','sum'))
k = 5                                  # as if every place had 5 extra neutral reviews
g['comfort'] = (g.quiet + g.cool - g.noise - g.heat) / (g.n + k)
```

```
top 5 cafes by comfort words (min 6 reviews in the sample):
                           name  n  comfort  atm  rating
Jing Si Books & Cafe (Yishun) 静思  7     0.42 5.00     4.7
                  Pulse and Pour  9     0.36 5.00     4.9
                      F&B Outlet 10     0.33 3.67     3.9
        Suzuki Cafe and Roastery  9     0.29 4.56     4.6
      The Coffee Bean & Tea Leaf  9     0.21 3.67     3.8
top 5 cafes by Google star rating:
                           name  n  comfort  atm  rating
Gaming On The Table Safra Yishun 11     0.00 5.00     5.0
                  Kin Joe Coffee  9     0.00 5.00     5.0
         Game Over Gaming & Cafe  9     0.00 5.00     5.0
           ASTRA, a coffee popup  8     0.08 4.75     5.0
            Coffeesmith(Chijmes)  8     0.00 5.00     5.0
correlation between comfort score and star rating across 1089 cafes: 0.16
```

The two top-5 lists share no cafe, and across 1,089 cafes the comfort score and the star rating correlate only 0.16. Stars mostly measure food and service; a person who needs a quiet, cool place to work needs a different ranking. This is the case for **multi-criteria recommendation**: ask what matters to the person, then rank by that.

CHALLENGE B · A PERSONAL RANKING

Ask three classmates to weight quiet, cool, space and price from 0 to 3. Build each person’s top 5 cafes. How different are the lists? Which places appear for everybody?

8 Beyond Words

Bigger models, pictures of streets, and ethics.

Better text models. Word lists and TF-IDF are fast and explainable. Sentence embeddings and large language models understand negation (“not noisy at all”) and context (“hot chocolate”), and can tag themes with no hand-made lists. They cost more, can be confidently wrong, and must be checked against labels in exactly the way you did in Lab 4.

Pictures of the street. Reviews say how a place feels; street view images show what the street looks like: trees, sky, buildings, cars. Open tools such as **ZenSVI** download street view images from open platforms like Mapillary, and segment them into green, sky and building shares. Datasets such as **Global Streetscapes** already provide millions of labelled images, including Singapore. Joining the two answers questions like: *do places on greener streets get fewer heat complaints?*

> **CHALLENGE C · REVIEWS MEET STREETS**
>
> For 50 places, find the nearest street view image in Global Streetscapes. Compare the green share of the image with the heat and quiet scores of the place. What else could explain what you see?

Ethics checklist

- **Terms of service.** Reviews belong to their authors and the platform. Use them for research; do not republish raw text.
- **Privacy.** Drop names and profile links. Short quotes can be searched; paraphrase in papers.
- **Who is missing.** Older people, low-income residents and migrant workers rarely write reviews, but they use hawker centres and streets the most.
- **Fake reviews.** Look for bursts of five-star reviews with similar text.

Pitfalls and Glossary

Read this before you use reviews on a real project.

Pitfalls checklist

- **Unbalanced labels.** Most scores are 4–5. Report a baseline and balanced accuracy.
- **Labels mean what users make them mean.** “Atmosphere” is used to punish rude staff.
- **Ambiguous words.** “Hot” is mostly food. Read examples before trusting a list.
- **Low recall.** Most people never mention noise; absence of complaints is not evidence of comfort.
- **Expectation, not measurement.** Complaints measure the gap between expectation and experience.
- **Fuzzy dates.** Relative dates (“a year ago”) blur any seasonal pattern.
- **Few reviews per place.** Shrink scores toward the average before ranking.
- **Sampling.** We kept at most 12 reviews per place so big malls do not dominate. Say how you sampled.

Glossary

Located opinion
An opinion tied to a place and a time.

Aspect score
A sub-score for one aspect, such as food or atmosphere.

Token
A word or piece of a word the computer counts.

Stop words
Very common words removed before counting (the, and).

Log-odds
How much more common a word is in one group than another, on a log scale.

TF-IDF
Word weight: frequent here, rare everywhere.

Logistic regression
A linear model that outputs a probability.

Baseline
The score of a trivial rule, such as always guessing the majority.

Balanced accuracy
Average of the hit rates of each class.

Precision / recall
When it fires, is it right? Of all true cases, how many did it catch?

Regular expression
A pattern that matches text.

Shrinkage
Pulling scores from few data toward the average.

One-Page Cheat Sheet

Everything in this guide on one page.

LOAD

`df = reviews.merge(places, on='place_id')`

CLEAN

Lower-case, drop very short texts, cap reviews per place, remove names.

DISTINCTIVE WORDS

$\log((a+1)/N_a) - \log((b+1)/N_b)$

TEXT TO NUMBERS

`TfidfVectorizer(ngram_range=(1,2), min_df=3)`

MODEL

`LogisticRegression(class_weight='balanced')`

REPORT

Baseline · accuracy · balanced accuracy · examples of errors.

WORD LIST

`text.str.contains(r"\b(?:noisy|loud)", regex=True)`

CHECK A LIST

Compare with labels: precision and recall. Read 20 matches.

LAB RESULTS

Atmosphere model 0.92 (baseline 0.86). Noise words: −0.84 stars. Hawker heat 0.5% vs cafe 0.9%.

MAP

Grid cells, minimum count per cell, shrink toward the city mean.

RANK

$(good - bad) / (n + k)$

NEXT

Embeddings or LLM tagging · street view images (ZenSVI) · sensors.

Further Reading

The papers and tools behind each idea.

1. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2), 1–135.
2. Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.
3. Jurafsky, D., & Martin, J. H. (draft, 2024). *Speech and Language Processing* (3rd ed.), chapters on naive Bayes, logistic regression and embeddings.
4. Monroe, B. L., Colaresi, M. P., & Quinn, K. M. (2008). Fightin' words: lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4), 372–403.
5. Hutto, C., & Gilbert, E. (2014). VADER: a parsimonious rule-based model for sentiment analysis of social media text. *ICWSM*.
6. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. *NAACL*.
7. Martí, P., Serrano-Estrada, L., & Nolasco-Cirugeda, A. (2019). Social media data: challenges, opportunities and limitations in urban studies. *Computers, Environment and Urban Systems*, 74, 161–174.
8. Biljecki, F., & Ito, K. (2021). Street view imagery in urban analytics and GIS: a review. *Landscape and Urban Planning*, 215, 104217.
9. Hou, Y., Quintana, M., Khomiakov, M., Yap, W., Ouyang, J., Ito, K., Wang, Z., Zhao, T., & Biljecki, F. (2024). Global Streetscapes: a comprehensive dataset of 10 million street-level images across 688 cities for urban science and analytics. *ISPRS Journal of Photogrammetry and Remote Sensing*. doi:10.1016/j.isprsjprs.2024.06.023
10. Ito, K., Zhu, Y., Abdelrahman, M., Liang, X., Fan, Z., Hou, Y., Zhao, T., Ma, R., Fujiwara, K., Ouyang, J., Quintana, M., & Biljecki, F. (2025). ZenSVI: an open-source software for the integrated acquisition, processing and analysis of street view imagery towards scalable urban science. *Computers, Environment and Urban Systems*, 102283.

Files · `reviews_lab.py` · `reviews_lab.ipynb` · data: `sg_places.csv`, `sg_reviews.csv` (from your instructor, for class use only)

Part of the Fundamentals 101 series · mahmoud.work