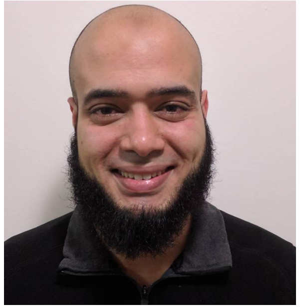
Machine Learning

Evaluation Metrics 4

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



© 2023 All rights reserved.
Please do not reproduce or redistribute this work without permission from the author

Little Background for sake of deeper understanding

Question

- A) Suppose a car drives 1 mile at 30 mph and 1 mile at 60 mph. What is average speed over the whole two miles?
- B) Suppose a population grows at 5% one year and 10% the next. What is average population growth rate over the whole two years?

- You shouldn’t just use the arithmetic mean!
- For A use harmonic mean and for B use geometric mean. Any other metrics will produce wrong answers!
- For reference: answer, answer, tutorial

Measures of Central Tendency

- We can aggregate N numbers in different ways. We naturally use arithmetic mean, however it may not be a suitable one!

Arithmetic mean	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1 + \cdots + x_n)$
Geometric mean	$\sqrt[n]{\prod_{i=1}^{n} x_i} = \sqrt[n]{x_1 \cdot x_2 \cdots x_n}$
Harmonic mean	$\frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}}$

Img src

Measures of Central Tendency

- Observe relationships

$$\log\left(\left(\prod_{i=1}^{n} x_i\right)^{1/n}\right) = \frac{1}{n}\sum_{i=1}^{n}\log x_i,$$

for $x_1, \ldots, x_n > 0.$

If a and b are postive numbers, then

$$\textbf{Arithmetic Mean (AM)} = \frac{a+b}{2}$$

$$\textbf{Geometric Mean (GM)} = \sqrt{ab}$$

$$\textbf{Harmonic Mean (HM)} = \frac{2ab}{a+b} = \frac{(GM)^2}{AM}$$

Measures of Central Tendency

- One major criteria in comparing the different means (arithmetic, geometric, harmonic) is their **sensitivity** to **outliers**
  - Outliers: quantity (extreme big or small values) - relationship (additive or multiplicative)
- **Arithmetic Mean** is for data that is **symmetric** and has **no extreme** large values (no additive outliers)
  - Salary avg in a company [50k, 95k, 7,000,000]/3 ⇒ misleading
  - Dominated by the **largest numbers**
- Both **Geometric Mean** and **Harmonic Mean** are less sensitive to extreme (large) values but sensitive to extreme small values (all values > 0)
  - Dominated by the **smallest numbers**
  - **Geometric Mean**: Useful for **multiplicative** effects (e.g. **exponential** growth)
  - **Harmonic Mean**: Useful for **averaging rates (**km per hour**)** or **ratios** (precision and recall are ratios)

Big Picture of Summary Statistics

- Measures of **Central** Tendency
  - Arithmetic Mean, Median (not sensitive to extreme values but drops data), Mode
  - Harmonic (rates or ratios) and Geometric (percentages, rates, or exponential growth factors)
- Measures of **Spread**
  - Range, Variance, Std
  - Interquartile Range: The range within which the middle 50% of your data falls: Q3-Q1
- Measures of **Shape**
  - Skewness (asymmetry)
  - Kurtosis (tailedness of the distribution)
- Measures of **Relationship**
  - Correlation & Covariance

Back to Metrics

F1-Score

- Relying on a sole metric often leads to misinterpretation
  - Using multiple metrics is useful
  - However, sometimes a singular score is necessary, demanding an combination of other metrics
- F1 score provides a **balance** between precision and recall, offering a **single metric** to evaluate the overall performance of a binary classification model
  - It can be used for imbalanced datasets

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-Score Intuition

The harmonic mean H of the positive real numbers $x_1, x_2, \ldots, x_n$ is defined to be

$$H(x_1, x_2, \ldots, x_n) = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}.$$

- The F1 score uses the **harmonic mean** of precision and recall rather than the **arithmetic mean**
- Think in their comparison!

F1-Score Intuition

- Both precision and recall have to be high to get a high F1 score.
  - If either precision or recall is low, the F1 score will also be low.
  - In contrast to the arithmetic mean, where one high value can compensate for a low value
  - F1 <= AVG

```
precision=0.1, recall=0.9, f1= 0.18, avg=0.50
precision=0.2, recall=0.9, f1= 0.33, avg=0.55
precision=0.3, recall=0.9, f1= 0.45, avg=0.60
precision=0.4, recall=0.9, f1= 0.55, avg=0.65
precision=0.5, recall=0.9, f1= 0.64, avg=0.70
precision=0.6, recall=0.9, f1= 0.72, avg=0.75
precision=0.7, recall=0.9, f1= 0.79, avg=0.80
precision=0.8, recall=0.9, f1= 0.85, avg=0.85
precision=0.9, recall=0.9, f1= 0.90, avg=0.90
precision=0.1, recall=0.9, f1= 0.18, avg=0.50
precision=0.01, recall=0.9, f1= 0.02, avg=0.46
precision=0.001, recall=0.9, f1= 0.002, avg=0.45
```

F_β score

- A more general F score: recall is considered **β times** as important as precision (common values 2 and ½)
    - If our F1 score increases, it means that our model has increased performance for accuracy, recall or both.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

In terms of Type I and type II errors this becomes:

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{true positive}}{(1 + \beta^2) \cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}}.$$

F_β score

- In practice seems what is common is:
- F1-score for equal weight to precision and recall
- F2-score weighs recall higher than precision.
  - false negatives are more concerning than false positives
- F½-Score weighs precision higher than recall.
  - false positives are more concerning than false negatives.

Understanding classification_report

```
y_true = [0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1]
y_pred = [0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]

conf = confusion_matrix(y_true, y_pred)
print(conf)

tn, fp, fn, tp = conf.ravel()   # table order

print(f'tp={tp}, fn={fn}, tn={tn}, fp={fp}')

report = classification_report(y_true, y_pred)
print(report)
```

Understanding classification_report: Part 1

```
[[2 4]
 [5 3]]
tp=3, fn=5, tn=2, fp=4
              precision    recall  f1-score   support

           0       0.29      0.33      0.31         6
           1       0.43      0.38      0.40         8
```

- Focus on label 1. This is our default (1 for positive and 0 for negative)
    - So our precision is 0.43 = (3/(3+4))
- In practice, it is a good idea to complement results for the **switched** label
    - So first row means, if 0 was dealt as the positive class, then its precision is 0.29 = (2/(2+5))

Understanding classification_report: Part 2

- Now, we have 2 F-scores (2 angles). What if we want to **combine**?!
  - In multi-classifications we have: F_{cat}-score, F_{dog}-score, F_{cow}-score, F_{lion}-score
- We can simply average them, e.g. (0.31+0.40)/2 = 0.35
  - We call this one: macro average. Cons: don’t consider the dataset imbalance!

accuracy			0.36	14
macro avg	0.36	0.35	0.35	14
weighted avg	0.37	0.36	0.36	14

Understanding classification_report: Part 2

- **To consider the imbalance**, compute the weight of each class
    - # of class examples / total examples
    - W0 = 6/14 and w1 = 8/14
- 6/14 * 0.31 + 8/14 * 0.4 = 0.36
- This is called the weighted F-score
- Overall, you might need one of these 3 values

accuracy			0.36	14
macro avg	0.36	0.35	0.35	14
weighted avg	0.37	0.36	0.36	14

Micro Average

- Micro Average sum the TP, FP, FN for all the classes
  - So it has a similar sense like accuracy in aggregating across all classes
  - Now, just computer their Precision, Recall then F-Score
- Code
  - `from sklearn.metrics import f1_score`
  - `micro_f1_score = f1_score(y_test, y_pred, average='micro')`

accuracy			0.36	14
macro avg	0.36	0.35	0.35	14
weighted avg	0.37	0.36	0.36	14

Imbalanced datasets

- For every class, look individually for its F-Score first
  - Also, investigate the Confusion Matrix
- Micro-Average: Gives **equal** weight to **each instance**
  - Reflect overall performance - however biased toward the major class (more instances)
- Macro-Average: Gives **equal** weight to **each class**
  - Reflect **individual** classes performance (as if you investigated them equally)
- Cohen's Kappa: [learn in future]

Sensitivity and Specificity

- Sometimes, you may want to combine these 2 metrics
  - **Sensitivity** (true positive rate), aka recall
  - Specificity (true negative rate)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{FPR}$$

- The geometric mean (G-mean) is the root of the product of **class-wise sensitivity**.
  - Maximize the accuracy on each of the classes while keeping these accuracies balanced
  - Ability of the classifier to correctly identify **positive cases** (sensitivity)
- For binary classification G-mean is the square root of the product of the sensitivity and specificity
  - If either is low, the geometric mean will also be low
- See imblearn.metrics.geometric_mean_score

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”