# AI Stuff

Abdullah Al Mahmud

December 1, 2025

# Contents

# Chapter 1

# Stuffs You Should Know

## 1.1 Probability

If A is an event, then the probability of A is

$$P(A) = \frac{Favorable\ Outcomes}{Total\ Outcomes} \tag{1.1}$$

The range of probability is $[0, 1]$.

## 1.2 Odds

The ratio of favorable outcomes to unfavorable outcomes. For event A

$$\begin{aligned} Odds(A) &= \frac{Favorable\ Outcomes}{Unfavorable\ Outcomes} \\ Odds(A) &= \frac{P(A)}{1 - P(A)} \end{aligned} \tag{1.2}$$

The range of odds is $[0, \infty)$

**Why use Log-Odds?**

Linear models (like $y = mx + b$) naturally output values in the range $(-\infty, \infty)$. Probability is bounded to $[0, 1]$. To bridge this gap, we use the logit function to map probability to the

real number line.

$$Logit(P) = \ln\left(\frac{P}{1-P}\right) = \ln\left(Odds\right) \tag{1.3}$$

Let's take an example, the odds of my cricket team winning is 1 to 6. That means if they win 1 game they lose 6 games. So, odds of winning $Odd = \frac{1}{6} = 0.167$. Now if the odds of my team winning is 6 to 1 (win 6, lose 1) then odds of winning is $Odd = \frac{6}{1} = 6$.

One interesting thing here is if my team is very bad and keeps losing then the odds of winning keeps getting close to 0. For example let's look at some odds where team is winning less. $\frac{Win}{Loose} = \frac{1}{10} = 0.1, \frac{1}{100} = 0.01, \frac{1}{1000} = 0.001, \{range[0,1]\}$

One the other hand, if my team is very good and keep winning then here are the odds. $\frac{Win}{Loose} = \frac{10}{1} = 10, \frac{100}{1} = 100\{range[1,\infty)\}$

This asymmetry makes it difficult to map these numbers to number lines or to compare odds for winning and against winning. This is where Logit function comes in handy. Why? Here is the same examples given above but with Log.

$\frac{Win}{Loose} = \log\frac{1}{10} = -1, \log\frac{1}{100} = -2, \log\frac{1}{1000} = -3$

$\frac{Win}{Loose} = \log\frac{10}{1} = 1, \log\frac{100}{1} = 2, \log\frac{1000}{1} = 3$

We get similar relation with ln as $ln$ is just $\log_e x$.

Log odds create a normal distribution making it useful for solving classification problems.

Reference Video: Odds and Log Odds

## 1.3 Sigmoid Function

On the other hand, we have Sigmoid function to convert odds to probability.

$$P = \frac{e^{\ln\left(Odds\right)}}{1 + e^{\ln\left(Odds\right)}} \tag{1.4}$$

## 1.4 Chi Square Test

Chi-square evaluates how strongly each categorical feature correlates with the target in classification pipelines. High $\chi^2$ = higher predictive value. Low $\chi^2$ = noisy or irrelevant feature. Chi-square gives each feature a score, a single number that measures how strongly

that feature is associated with the target. Higher score means stronger association. let's take an example. Consider the following data.

| gender | age group | customer rating |
|---|---|---|
| male | 18-30 | 5 |
| male | 31-40 | 4 |
| female | 40+ | 3 |

Table 1.1: Chi Square Example Dataset

We will find out chi square value using scikit learn.

```
from sklearn.feature_selection import chi2

# X = features, Y = target
chi_scores, p_values = chi2(X, Y)
```

We get following result for this dataset:

| feature | chi square score | p values |
|---|---|---|
| gender | 0.70 | 0.70 |
| age group | 1.95 | 0.38 |
| customer rating | 22145.69 | 0.00 |

Table 1.2: Chi Square Example Dataset

If one feature strongly aligns with a specific sentiment (positive/negative), the observed and expected frequencies differ massively, producing a huge $\chi^2$. Gender, age group none of them seem to influence whether a person leaves positive/negative sentiment. If the target distribution is similar across categories (e.g., males and females leave similar sentiments), chi-square stays small. $\chi^2$ can be NaN when a feature has only 1 unique value, there were missing values, the feature was non-numeric.

## 1.5 TF IDF

TF - Term Frequency. IDF - Inverse Document Frequency. The term document here means one individual text instance in the dataset. Is is the smalles unit on which TF IDF is calculated. A document can be one or multiple sentences. The definition depends entirely on the structure of the dataset. TF is calculated inside a single document. IDF is calculated

across all documents. If a dataset has a column named review with 1000 rows then each row is one document, all 1000 rows together are document corpus. Let's take an example. Following is a column of a dataset.

| serial | review |
|---|---|
| 1 | food is great |
| 2 | great service great ambiance |
| 3 | food was cold |

Table 1.3: TF IDF Example

For a word $w$ in document $d$

$$TF(w,d) = \frac{number\ of\ times\ w\ appears\ in\ d}{total\ number\ of\ words\ in\ d} \tag{1.5}$$

In the first row, the word 'great' appears 1 time and total words in the row is 3. So, $TF(w,d) = \frac{1}{3}$.

For a word $w$

$$IDF(w) = \log(\frac{N+1}{df(w)+1}) + 1 \tag{1.6}$$

$N$ = total number of documents. $df(w)$ = number of documents that contain the word. the $+1$ adjustment prevent division by zero and zero logs.

The word 'great' appears in 2 documents and total number of documents in the above dataset is 3. So, $IDF(great) = log(\frac{3+1}{2+1}) + 1$.

$$TF - IDF(w,d) = TF(w,d) \cdot IDF(w) \tag{1.7}$$

## 1.6 Need to Study following Methods

### 1.6.1 Feature Selection

- Feature is categorical, target is categorical

    1. Mutual Information (MI)
    2. Cramer's V
- Feature is numeric, target is categorical
    1. ANOVA
    2. Point-Biserial Correlation
    3. Kruskal–Wallis H Test (non-parametric ANOVA)
- Feature is categorical, target is numeric
    1. Correlation ratio
- Feature is numeric, target is numeric
    1. Pearson correlation
    2. Spearman rank correlation
    3. F-test / Univariate regression coefficients
    4. Lasso (L1 regularization)

### 1.6.2 Text Encoding

- TF IDF
- Bag of Words
- Embeddings (BERT)

# Bibliography