

Machine Learning for Grading Students

Gautam Singh

CONTENTS

1	Introduction	1
2	Fitting a Gaussian Curve	1
3	K-Means Clustering	1
4	Results	2

Abstract—This document is a report which compares the grade distribution obtained by using a method based on machine learning as compared to fitting a normal curve to the scores of the students.

1 INTRODUCTION

We test the utility of the K -means algorithm in assigning grades as compared to estimating the grades using the standard normal distribution.

We consider the scores of $N = 94$ students who have taken a course in the Indian Institute of Technology, Hyderabad (IITH) as our dataset.

2 FITTING A GAUSSIAN CURVE

Since N is not very large, given the scores of each student x_i , $1 \leq i \leq N$, we can compute the population mean and population variance as

$$\mu = E[x] \quad (1)$$

$$\sigma^2 = E[(x - \mu)^2] \quad (2)$$

We assume that the scores $x \sim N(\mu, \sigma^2)$. Thus, we compute the Z-scores as

$$Z = \frac{x - \mu}{\sigma} \quad (3)$$

The grades are assigned as per Table 1.

The Python code `codes/grades_norm.py` takes the given input population dataset `marks.xlsx` and assigns grades appropriately. The grades are output to `grades.xlsx`.

Interval	Grade
$(-\infty, -3]$	F
$(-3, -2]$	D
$(-2, 1]$	C
$(-1, 0]$	B-
$(0, 1]$	B
$(1, 2]$	A-
$(2, 3]$	A
$(3, \infty)$	A+

TABLE 1: Grading Scheme.

3 K-MEANS CLUSTERING

K -Means clustering is an unsupervised classification model, which attempts to cluster unlabeled data in order to gain more structure from it.

We frame this requirement as an optimization problem. For a set of data points $\{\mathbf{x}_i\}_{i=1}^N$ and means $\{\boldsymbol{\mu}_i\}_{i=1}^K$, we define for $1 \leq n \leq N$, $1 \leq k \leq K$,

$$r_{nk} \triangleq \begin{cases} 1 & \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\| = k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Thus, we need to find points $\boldsymbol{\mu}_k$ minimizing the cost function

$$J \triangleq \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \quad (5)$$

Clearly, (5) is a quadratic function of $\boldsymbol{\mu}_k$. Differentiating with respect to $\boldsymbol{\mu}_k$ and setting the derivative to zero, we get

$$\sum_{n=1}^N 2\boldsymbol{\mu}_k r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \quad (6)$$

$$\implies \boldsymbol{\mu}_k = \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{\sum_{n=1}^N r_{nk}} \quad (7)$$

From (7), we see that the optimum is attained when $\boldsymbol{\mu}_k$ is set to the expectation of the \mathbf{x}_n with respect to r_{nk} .

Thus, the K -means algorithm is essentially an

EM algorithm, where each iteration consists of two steps.

- 1) *E Step*: Calculate the K -expected values

$$\tilde{\mu}_k \triangleq \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{\sum_{n=1}^N r_{nk}} \quad (8)$$

for $1 \leq k \leq K$.

- 2) *M Step*: Assign $\mu_k \leftarrow \tilde{\mu}_k$ for $1 \leq k \leq K$.

4 RESULTS

The grade distribution using each method is shown in Fig. 1 and Fig. 2. Based on the results,

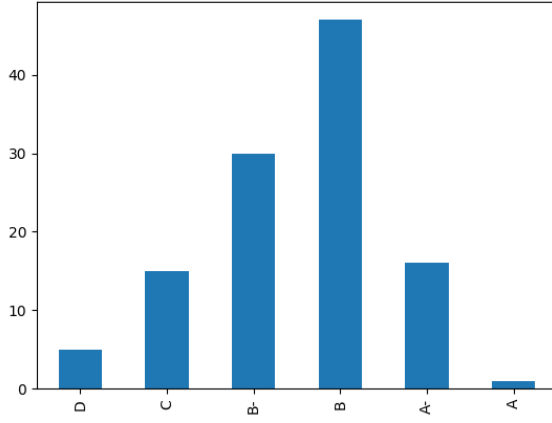


Fig. 1: Grade distribution using a Gaussian curve.

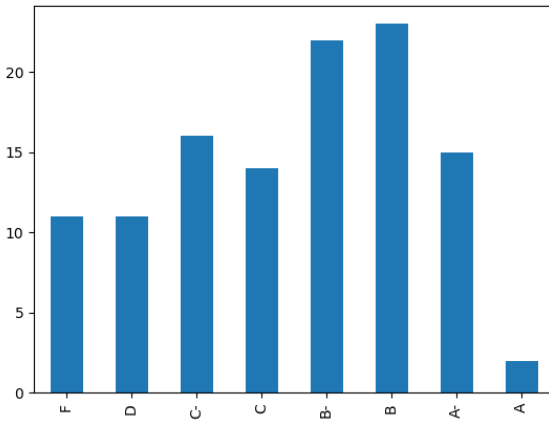


Fig. 2: Grade distribution using the K -means algorithm.

we can make the following observations:

- 1) Grading using the Gaussian distribution would lead to many students failing the course, while this is not the case using the K -means algorithm.
- 2) Using the Gaussian distribution is quite unfair, since there could be students with quite similar marks but with a difference in grade, just because they lie on either side of a predefined boundary.
- 3) The K -means algorithm allows for better decision boundaries, depending on how skewed the performance of the students is, accordingly to the difficulty of the course.
- 4) Unlike the Gaussian distribution, the K -means algorithm can be used for a fairer assignment of the grades, no matter how skewed the performance of students in a course is.