

Predicting Professor Ratings With Machine Learning

Daniel Son

Context

Students at the University of Maryland use the website PlanetTerp to leave reviews and star ratings for professors and courses they enrolled in.

Dylan Carpenter (ENGL391 - Advanced Composition)



“Such a nice person, and he really cares about his students. He is a very lenient grader, and the assignments aren't that hard...”

– Anonymous

Ting Jiang (CMSC351 - Algorithms)



“A fantastic instructor. She is extremely organized and always willing to help her students succeed. Her slides are very informative and are posted on Canvas...”

– Anonymous

The Goal

How well do students' reviews reflect the star rating they award to a professor?

With the power of machine learning, we can uncover the relationship between what students write and the score they ultimately choose

We must build a tool that takes in a review of a professor and predicts how many stars that review gave

The Data

Ting Jiang

Total reviews: 26

Average rating: 3.42

Courses taught:

- CMSC351
- CMSC417

Larry Herman

Total reviews: 336

Average rating: 3.14

Courses taught:

- CMSC106
- CMSC131
- CMSC132
- CMSC216
- CMSC250
- CMSC330

Maksym Morawski

Total reviews: 247

Average rating: 2.80

Courses taught:

- CMSC106
- CMSC250
- CMSC320
- CMSC351
- CMSC421
- DATA602

Ilchul Yoon

Total reviews: 175

Average rating: 2.26

Courses taught:

- CMSC106
- CMSC122
- CMSC131
- CMSC132
- CMSC216
- CMSC389N
- CMSC411

Allan Yashinski

Total reviews: 93

Average rating: 4.60

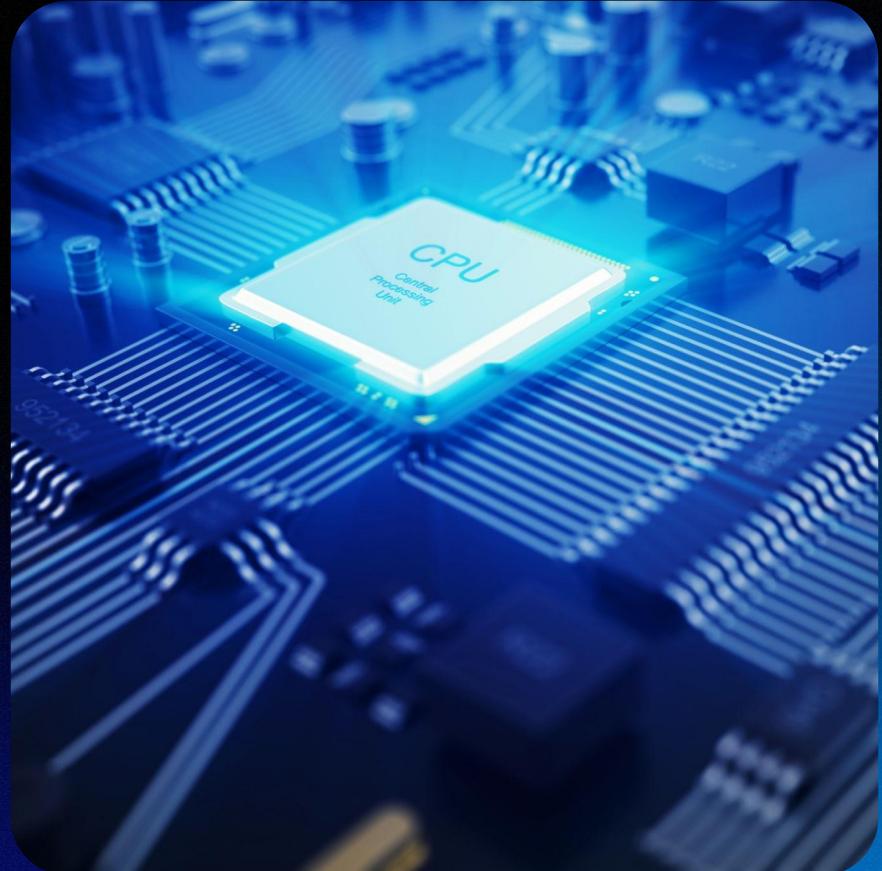
Courses taught:

- MATH240
- MATH241H
- MATH401
- MATH402
- MATH403
- MATH405
- MATH456
- MATH461
- MATH463

The ML Model

Model: PyTorch regressor

- Pre-trained DistilBert
- AdamW optimizer
- Predicts a continuous value 1-5 stars
- 5 epochs
- 80/20 test split



The Process

01 Extract and Clean Data

02 Run PyTorch Model

03 Results and Analysis

1. Extract and Clean Data

1. Obtained professor reviews from PlanetTerp, extracting key data such as
 - Paragraph review
 - Star rating (0-5)
 - Expected letter grade
2. Cleaned data by removing rows with missing data

2. Run PyTorch Model

1. Tokenized review text data and generated attention mask tensors
2. Converted token sequences into contextual embeddings representing the overall meaning of the review
3. Passed embedding and corresponding one-hot encoded expected grade into a neural network to predict star rating

3. Results and Analysis

Epoch	Training Loss
1	3.64607
2	1.11485
3	0.40607
4	0.25080
5	0.17374

Stat	Measurement
Mean Absolute Error	.59 stars
Test MSE	.70 stars
Test RMSE	.83 stars
Accuracy (within ± 0.25)	35%
Accuracy (within ± 0.5)	55%

3. Results and Analysis

- Strong grasp of overall sentiment
- Central tendency toward high ratings
- Clear sentiment cues dominate predictions
 - Reviews with explicit sentiment words (“worst,” “fantastic,” “excellent”) tend to have very small error, sometimes within ± 0.1 stars

Overall:

Most predictions are reasonably close to the true ratings, and the model clearly learns meaningful sentiment patterns from the reviews

Summary

- Collected PlanetTerp reviews for 5 UMD professors
- Fine-tuned a DistilBERT transformer in PyTorch
- Achieved statistically significant accuracy

Conclusion:

Students' reviews can be used to predict how highly they rate their professors

Follow up question:

Is there a correlation between expected grade and star rating?

55%

accuracy within ± 0.5 stars

0.59

stars away from true rating,
on average

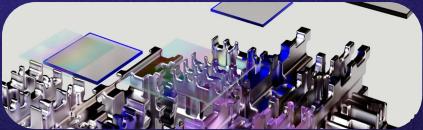
877

reviews used to train model

0.17

training loss after five epochs

Limitations



Small Dataset

With only 5 professors and 877 reviews, the model limits generalization

Lack of Stratification

Single train/test split with no stratification means metrics may be optimistic if similar reviews land in both sets

Weak Feature Handling

One-hot can mishandle missing/rare grades and learn factitious correlations

Overfitting Risk

DistilBERT is heavy for the dataset size, so results have high variance

Thank you!

Disclaimer: the code attached
may have different results