

A Machine Learning Approach To Classifying Cuteness

Anoushka Shrivastava
Carnegie Mellon University

March 25, 2021

Abstract. Computer science students have conquered many problems, but finding their ideal match is not one of them (and unfortunately algorithmic solutions tend to be lacking in this area). Nevertheless, in preparation for a time (in the far, far future) that we get stormed with hundreds of dating offers instead of Gradescope receipts in our inboxes, this paper aims to devise a machine learning model to filter through our options by classifying whether people are cute or not. This classification is life-saving since by narrowing down candidates for them, we save CS students time to focus on their personal hygiene, a topic on which an entire study of its own can (and probably should) be conducted.

I. Introduction.

Anyone intrigued by the abstract is probably a lost CS student, so instead of scaring (or scarring) you with statistics about the percentage of students who are able to find love (hint: think less than the probability of rolling a 7 on a 6-sided die), I will use the introduction as encouragement for you. If you're at the point where you need machine learning to solve your problems, your desperation probably leads you to have low standards for a partner, so you will find someone eventually. However, this is also why the model is important: since your bar is so low, when you finally do get offers, we can be sure to use the model to reject the people that you should be immediately rejecting under normal circumstances.

II. Methods.

When I finally got an idea for this paper a few hours before the deadline, the only people around at home were my mother, father, and sister. Therefore, my dataset consists of these three, and my own photos. I made many attempts to find others to take pictures of, including following my neighbors on their walks, but I was informed by their lawyer that the ethics of this approach were a bit controversial.

In an effort to produce more training data, I decided to include front, side, back, up, down, upside-down, left, right, north, south, east, west, and diagonal profiles of each family member. I also included them sitting, standing, jumping, kicking, swinging, eating, and any other pose corresponding to my first-grade vocabulary verb list of the week of September 27th. Finally, because the dataset was still not full enough, I included pictures of household objects such as my sofas and air conditioner.

I began with the intent to lead with a supervised training approach. However, after my mother learned that I had labeled the worm in our garden cuter than her, she suggested that if I wanted to keep my tuition at CMU, I best make this an unsupervised learning model. I obliged, especially since labeling my own image would probably be a conflict of interest.

Honestly, after rereading my methods on data collection, I'm not quite sure that the rest of my methods are so sound. In fact, I'll just keep the rest of this section to myself.

III. Results and Conclusions.

So, it turns out the cutest person is the kitchen spoon in our left drawer. While I am surprised that the spoon beats even my flattering diagonal profile, I still think the model's confidence of 1003.27% seems legit and makes it trustworthy.

IV. Future Work.

After my family realized that they had lost to a spoon, they informed me that they were drafting up a 10-year research project titled "How long can one college student survive on dining hall food?" with me as the test subject. With sincerely no offense to my university, I do not wish to participate in this study, and will be terminating my work on this project.

V. Acknowledgements

I would like to thank my friends Keevyu and Karen for unintentionally inspiring this project. I would also like to thank Sheryl Mathew and Preethi Krishnamoorthy for providing feedback and proofreading. Finally, I would like to thank my coffee mug, who was disappointed that the spoon was cuter than it (mug, you'll always have a special place in my heart).

VI. References

None.