Predicting Attractiveness and Automating Online Dating

Abraham Yepremian
California Polytechnic University, Pomona – CS 4990 Fall 2018

Abstract

Relationships are important to everyone. For some, there is an absence of time in their lives to find mates through regular means. Online dating is becoming increasingly more popular as the internet grows. Tinder alone has 50 million users. This application allows you to swipe right on a person indicating that you like them, or swipe left indicating you would like to move on to the next person. However, online dating takes time and effort. To save on time, a bot can be created to automate online dating. Unfortunately, there was no good solution for a bot like this on Tinder. If all you did was like everyone, then the Tinder algorithms would rank you lowly and you would receive minimal matches. Fortunately, machine learning is here to save the day. Machine learning can be used to learn what matches you like. Alternatively, machine learning can be used to predict the attractiveness of a person given an image of their face. Using a machine learning model that can predict attractiveness, a bot can be made to accurately like people for you based on attractiveness.

Introduction

The problem is that online dating is too time consuming and tedious. The solution is to use machine learning to predict attractiveness of possible mates and to use this prediction to decide on how to move forward with a possible match. This action can be repeated by a bot. I used Tinder as the dating platform for this bot because there is an API available to use in Python called Pynder.

Originally, I was going to create a convolutional neural network with data that I was planning to collect myself. However, I found the SCUT-FBP5500 dataset which contains 5500 images of faces ranked by attractiveness from 1 to 5 by volunteers [1]. I found a paper that trained this data using label distributed learning and found better results for the data [2]. Upon finding this, I wanted to create my own implementation of the model. I found a trained model with this data that I could use from Qi Di's GitHub repository detailing how to predict beauty with this model [3].

After making the model work to predict attractiveness on images alone, I began to work on creating a bot to extract images from nearby users on Tinder and rate their attractiveness. If they passed a certain attractiveness threshold, they would be liked. This was found to be accurate and effective in increasing my matches.

Method

Keras was used to train the data and develop a convolutional neural network (CNN) that can be used to predict attractiveness. On top of the given SCUT-FBP500 dataset, transfer learning was used with the ResNet50 neural network. The model was trained using label distribution learning based on ResNet fine tuning [2]. The Pearson correlation coefficient achieved is 0.95, which is higher than the 0.8997 coefficient from the original publishers of the dataset. A better accuracy was achieved due to the label distribution learning and ResNet fine tuning [2]. The model used was trained by data scientist Qi Di [3]. My code was adapted entirely from Qi Di's code for beauty prediction [3].

Link to model used
Link to my full code

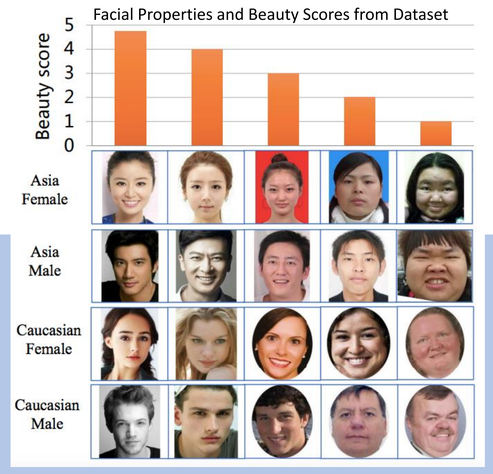
For the Tinder bot implementation, only the main profile picture is considered. The bot will detect a face in the image. The face is cropped out of the image and resized to 224 by 224 pixels. The image is then fed into the convolutional neural network where it will be predicted with a score of 1 through 5. Then the score is scaled to be out of 10. Since the model has a tendency to be a hard grader, I set my threshold for liking the potential mate to be 6/10. If the attractiveness threshold is met, then the bot will like the user. Else, the bot will dislike the user and move on to the next candidate. This bot can be left on indefinitely if you have Tinder Plus, which grants unlimited swipes .

Data

The dataset used for the training of the model was the SCUT-FBP500 dataset from the South China University of Technology [1].

Link to publication
Link to dataset

The SCUT- FBP5500 dataset contains 5500 images of frontal faces with diverse properties (male/female, Asian/Caucasian, ages) and diverse labels (face landmarks, beauty scores within [1, 5], beauty score distribution). The ratings from 1 to 5 (5 most attractive) were done by 60 volunteers from the ages of 18 to 27. This data allows different computational models with different FBP paradigms. From the publication, it was found that the CNN-based ResNeXt-50 model was the best performing.




```
Abrahams-MacBook-Air:ldl+resnet abe$ python3 tinder.
Using TensorFlow backend.
2018-12-10 19:08:59.298019: I tensorflow/core/platfo
Fetching users...
tinder80.jpg Rating: 4.95
You disliked Meg
tinder81.jpg Rating: 5.74
You disliked Christine
tinder82.jpg Rating: No Faces Found
You disliked Victoria
tinder83.jpg Rating: 5.39
You disliked Lexi
tinder84.jpg Rating: 6.88
You liked sierra
tinder85.jpg Rating: 5.86
You disliked Maeve
tinder86.jpg Rating: 8.02
You liked Richiier
tinder87.jpg Rating: 5.83
You disliked Hailey
tinder88.jpg Rating: 6.64
You liked Ellie
tinder89.jpg Rating: 5.93
You disliked Sofi
tinder90.jpg Rating: 7.85
You liked Bella
```

Output from my Tinder bot

Shows my large increase in matches as a result of using the bot

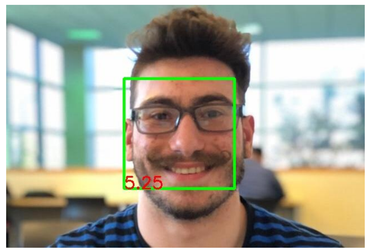


Image output for evaluation of attractiveness

Results

The training model was incredibly accurate given the dataset. The trained model used had a Pearson correlation coefficient of 0.95, which is higher than the 0.8997 that the original publishers of the dataset were able to achieve. Moreover, the tinder bot was incredibly effective. I used a Tinder boost to improve the views of my profile, set my distance to be the maximum allowed, and turned on the bot to let it run overnight. When I woke up, I was amazed to find 76 new matches! The bot had swiped on over 7200 users overnight.

One drawback of the model is that it grades harshly. It is rare for someone to score at least an 8/10. With over 1000 images that I monitored, the highest score achieved by anyone was 8.20/10.

Link to video demonstration

Conclusion

This project resulted in me creating a functional, effective bot for predicting attractiveness and automating online dating on Tinder. I am very happy with my results as it proved to be a solution to the automation of online dating.

However, there are improvements that can be made to both the training model and the bot. The model may be biased towards Chinese ideals of beauty. I presume the 60 volunteers that rated the images from the dataset are assimilated into Chinese culture. The images are all of Asian or Caucasian people. I can improve the model by adding more images to the dataset with my own ratings. This will increase the diversity of the dataset and have my model learn from me. Moreover, I can add a model to evaluate the bio of the user. I can also have my bot evaluate all images from the user rather than just the main profile picture. Using the bio and all the images, I can generate a total profile score. Adding a GUI would make this project user friendly. I can also make a chrome extension to work on the tinder web service using TensorFlow.js. Lastly, I can adapt the bot to work with other dating services.

Acknowledgement

[A1] For the model used and my starting code base, full credit goes to Qi Di's GitHub repo.
[A2] For experimenting with more advanced models for better performance, credit goes to Shu Liu, Bo Li, Yangyu Fan, Zhe Guo, and Ashok Samal for their publication.
[A3] For interacting with the Tinder API via Python, credit goes to the Pynder library.
[A4] For providing the initial inspiration to pursue this project, credit goes to Oscar Alsing and his Tinder bot series on YouTube.

References

[1] Liang, L., Lin, L., Jin, L., Xie, D., & Li, M. (2018). SCUT-FBP5500: A Diverse Benchmark Dataset for Multi-Paradigm Facial Beauty Prediction. *CoRR, abs/1801.06345*.
[2] Fan, Y., Liu, S., Li, B., Guo, Z., Samal, A., Wan, J., & Li, S.Z. (2018). Label Distribution-Based Facial Attractiveness Computation by Deep Residual Learning. *IEEE Transactions on Multimedia, 20*, 2196-2208.
[3] Di Q., Beauty Prediction, (2018), GitHub repository, https://github.com/ustcqidi/BeautyPredict