Part 1 code instruction

1. **How to use the code.**

   First, download the machine.pickle and part1_client_program.py

   Second, put these two files into one folder.

   Third, under this folder, create a folder named exam_data. Under exam_data, create a folder called part_1. Under part_1, create a folder called sample_new_data.

   Fourth, put the file containing 10,000 programmers' reviews into the folder sample_new_data.

   Fifth, open part1_client_program, and change the file_name in line 16 to your file name—the file you put into the sample_new_data.

   Sixth, run the part1_client_program.py. The prediction results with the original data will be saved in the folder sample_new_data, named <your_file_name>_with_prediction.csv.

2. **Why do I choose Naïve Bayesian model among Random Forest model, Logistic Regression model, and Naïve Bayesian model?**

   In part1_compare.py, I tried the Random Forest, Logistic Regression, and Naïve Bayesian models. I will choose the Naïve Bayesian model.

   The reasons are listed below.

   I use KFold to test the trained models. The Random Forest model has the lowest average accuracy score for the test data among the three models. So, I will not choose the Random Forest model.

   The Logistic Regression model has a higher average accuracy score for the test data than the Naïve Bayesian model. However, the Logistic Regression model's average accuracy score for the train data is one, while the Naïve Bayesian model is less than one. This means the Logistic Regression model has an overfitting problem.

   To alleviate this problem, I use penalized logistic regression. I choose the L1 regularization technique, which is called Lasso Regression. The train data's average accuracy score is slightly higher than the Naïve Bayesian model in this new Logistic Regression model. But its test data's average accuracy score is lower than the Naïve Bayesian model.

   So, the Naïve Bayesian model is the best one among these three.

3. **Why do some reviews are misclassified?**

   In part1_misclassified.py, I find 15 reviews have been misclassified. In most cases, the 1-star reviews are misclassified as 2-star reviews. Only one 1-star review is misclassified as a 3-star review, and one 2-star review is misclassified as a 1-star review. So, the model cannot classify 1-star and 2-star reviews very well.

   From my perspective, this is because the Naïve Bayesian has a Conditional Independence Assumption. The model will consider each word's probability in a text but won't consider the phases' probability.

   For example, in the misclassified reviews, almost all the 1-star reviews mentioned some positive words and very euphemistic negative words. It is difficult to realize these reviews are conveying negative emotions if only considering the single word's probability in different stars' reviews. If we separate these negative phases into single euphemistic

negative words, these words won't show up frequently in 1-star reviews. So, it is understandable that these reviews are misclassified. But if we combine these single words, it is easy to realize that these reviews are negative.
The limitation of the Naïve Bayesian model is that it cannot consider phases.

Part 2 code instruction

1. **How to use the code.**
   First, download the machine_part2.pickle, machine_part2_cnn_image.pickle and part3_client_program.py
   Second, put these three files into one folder.
   Third, under this folder, create a folder named exam_data. Under exam_data, create a folder called part_2. Under part_2, create a folder called sample_new_data.
   Fourth, put the file containing 10,000 programmers' profile picture types into the folder sample_new_data. Put the folder that contains all programmers' profile images into the folder sample_new_data.
   Fifth, open part2_run_prediction, and change the file_name in line 26 to your file name—the file you put into the sample_new_data. Change the folder_name in line 24 to your folder name – the folder you put into the sample_new_data.
   Sixth, run the part2_run_prediction.py. The prediction results with the original data will be saved in the folder sample_new_data, named <your_file_name>_data_prediction.csv.

2. **Why do I choose machine_part2.pickle and the performance of the model?**
   In part2_profile_predict.py, I tried four different models, including the Naïve Bayesian model, the random forest model, the logistic model, the SVM model. All models give me the same average accuracy rate for train data and test data. That is because the profile picture types only have one word. The information is insufficient which limit the models' ability to capture the underlying pattern in the data, resulting in similar predictions across different models.
   According to part1, I believe the Naïve Bayesian model have better prediction to the text, so I choose the Naïve Bayesian model in part 2 to do the prediction as well.

3. **Why do I choose machine_part2_cnn_image.pickle?**
   In part2_profile_predict.py. I tried different neural networks model. I changed the optimizer, the batch size, the epochs, and the dropout. I also tried different layers and nodes.
   In the end, according to the average accuracy score of the train data and test data in these models, I choose the one with adam optimizer, 128 batch sizes, 30 epochs, and 0.3 dropout, because it has the highest average accuracy sore of the test data.

Part 3 code instruction

1. **How to use the code.**
   First, download the machine_part3.pickle, machine_part3_cnn_image.pickle and part3_client_program.py

Second, put these three files into one folder.
Third, under this folder, create a folder named exam_data. Under exam_data, create a folder called part_3. Under part_3, create a folder called sample_new_data.
Fourth, put the file containing 10,000 programmers' reviews and profile picture types into the folder sample_new_data. Put the folder that contains all programmers' profile images into the folder sample_new_data.
Fifth, open part3_client_program, and change the file_name in line 26 to your file name—the file you put into the sample_new_data. Change the folder_name in line 24 to your folder name – the folder you put into the sample_new_data.
Sixth, run the part3_client_program.py. The prediction results with the original data will be saved in the folder sample_new_data, named <your_file_name>_data_prediction.csv.

2. **Why do I use this method?**
   I choose to combine the review text and the profile picture types into one column, then use Naïve Bayesian model to predict the programmers' stars.
   In part3_private_program, I have tried three methods. The first one is the one I choose. The second one is to train the model separately for different profile picture types. The third one is to predict the quality of the programmer using the profile picture only and supplement the prediction with the one I have done in part 1 using review text.
   Among these models, the first one can get the highest average accuracy rate in the test data. The average accuracy rates in the train data of these models are similar with each other.
   Thus, I choose the first model to do the prediction.

3. **Limitation of the method**
   Combining the review text and the profile picture types into one column probably will lose some contextual information. The profile and review texts have distinct meanings. However, after combining these two columns, the effect of profile will be diluted which may lead to the loss of the specific details.
   The second limitation is that the Naïve Bayesian has a Conditional Independence Assumption. The model will consider each word's probability in a text but won't consider the phases' probability.
   These limitations can affect the performance of the Naïve Bayesian model.