Authors: Patrick O'Connell (oconne16) & Jahnvi Patel (jpate201)

1. **What is your data and task (including data preprocessing steps)?**
   Dataset for this project is acquired from *Young People Survey* from *kaggle.com*. The main task of the project is to predict a persons' empathy as either "very empathetic" or "not very empathetic". The dataset provided in the .csv files is either numerical or categorical. The numerical values were sorted based on 0 being the least and 5 being the most. While, the categorical data was sorted more delicately based on specific keywords (i.e. "never smoked" = 0, or "doctorate degree" = 5). The empty fields were filled with the total average of the column. After loading and cleaning up the data, we normalized the data so all the categories can have the same scale for a fair comparison between them. The people rated their empathy rate higher than 3 are marked as "very empathetic" while the rest are marked as "not very empathetic".

2. **What ML solution did you choose and, most importantly, why was this an appropriate choice?**
   From the ML solutions we learned in class, we have chosen to implement K-Nearest Neighbors, Decision Tree, Gradient Descent using Loss functions (Hinge, Logistic, Squared) and Random Forest. We chose KNN due to its ability to pattern match, text mining and natural language processing. Decision Trees are great for classification and regression. The Loss functions are useful for mapping an event or values of variables onto a real number, which intuitively represents the "cost" associated with the event. Random Forest (RF) is an ensemble method, in which its classifiers provide higher accuracy and prevent overfitting (when more trees are present). The main principle of RF is that a group of "weak learners" can come together to form a "strong learner". In other words, each classifier, individually, is a "weak learner" while all the classifiers taken together create a "strong learner".

   We used these algorithms to evaluate which algorithm would run the best against the baseline. Ultimately, Random Forest gave us the best resulting test accuracy of 79% (approximately 15% better than the baseline)

3. **How did you choose to evaluate success, including baselines, experimental setup (e.g., % train/dev/test), metrics?**
   In order to set up the experiments, the dataset is separated into training, validation and testing dataset. These were chosen based on rule-of-thumb (80-10-10) discussed in class . Given 1010 responses, 810 were chosen as training data, 100 were chosen as validation data and the remaining 100 responses were set aside as test data. Success of the algorithm is based on its performance compared to the baseline collected. If the algorithm performs better than the baseline accuracy then it is marked as a success. The baseline is computed by using the training dataset marked with "very empathetic" and "not very empathetic" responses. Next, by using this training models most frequent empathetic state for out guessing strategy on the testing data, we obtain an accuracy of about 64%.

4. **What software did you use and why did you choose it?**

   (a) matplotlib.pyplot: To visualize plots

   (b) sklearn.tree: Classify decision tree-based models for classification and regression.

   (c) sklearn.metrics: To compute score functions, performance metrics and pairwise metrics and distance computations.

5. **What are the results?**

Algorithm	Test Accuracy
Baseline	63.63636363636364
Sqaured Loss	68.68686868686869
Logistic Loss	68.68686868686869
Hinge Loss	70.70707070707071
Decision Tree	70.70707070707071
KNN	72.72727272727273
Random Forest	78.78787878787878

Figure 1: Testing Accuracy of "empathetic" vs "not very empathetic"

6. **Show some examples from the development data that your approach got correct and some it got wrong: if you were to try to fix the ones it got wrong, what would you do?**