ST154 Kaggle Final Project

March 31, 2016

1 Introduction

Link to join at Kaggle: `https://kaggle.com/join/ST154TweeterSentiment`. You need to use your Berkeley email address for this competition. Once you are there, you can go to the Data tab, and download the data. It has a description for each of the dataset. Competition will last until Friday, April 29th, 3:00pm Pacific time (10:00pm UTC time).

To make a submission, create a csv file similar to the SampleSubmission.csv file. See the Evaluation tab on Kaggle website for examples on how to generate the submission file. Performance is evaluated based on the accuracy on the test set. There is a limit of two submissions per day, per person.

You can form a team of up to 3 people. No external data usage are allowed in this competition. Also, you are not allowed to read the tweet and label them manually. You are free to use any models and create any features you like.

2 Write up

Please provide a short write up of 3 -5 pages describing a bit about the data (eda), the features (if you do any feature engineering), and the models with hyper parameter you use. For example report tables like this

Model	CV Acc	Kaggle Acc	CV AUC	CV F1-Score	Hyper-parameters	Features
Random Forest	.80	.79	.88	.88	n_tree = 100	Set 1
Logistic L1	.80	.79	.88	.88	$\lambda = 1$	Set 2

Table 1: Model Performance

The write-up should be in Latex. You can use Lyx as an easy way to write Latex. See for example the homework solution. The grade for this project will depend both on the Kaggle Private Leaderboard, and the clarity of the write up. Please upload your codes to github, and include the link to the repository at the end of the write up.

3 Suggestions

Following are some suggestions, best practices in Kaggle competition. It is not required, though you might find these tips helpful.

1. See past Kaggle winning solutions, e.g. `https://github.com/pyduan/amazonaccess` for how to organize the codes and data, or a list of them here
2. Feature engineering: use more words in vocabulary, use meta-features (length of tweet), or ignore stop words.
3. Ensemble: averaging the result of multiple models to improve the performance.
4. Hyperparameter tuning: either manually, by grid search, or randomized grid search
5. In Python, scikit-learn is a great package for general machine learning. In R, check out XGboost, H2O, liblinear, or a bigger list here.
6. Git: use git and github for collaborating between the team