Jessica Lin & Leelabrindavanan Karunakaran

jl52746 & lk6788

Final Project Report

a) How did your team go about tackling this problem?

Neither of us have done a Kaggle competition before so it was a bit difficult to start and we had a lot of trial and error. In addition to that, we tried to keep the process as simple as possible and used resources like sklearn. One day in class, Nikhil suggested trying decision trees for the final project, so we began there and tried many different algorithms. We started off working individually and then if either of us came across a method that would increase our score, we would let the other person know. We used cross validation using a 30/70 split to try to get as high of an AUC as possible. At a later class date, Ethan gave a presentation in class about Kaggle competitions and some useful tips and tricks that we were able to implement as well.

b) Which methods/algorithms did you try?

We tried a variety of algorithms and used cross validation to try to get as high of an AUC as possible. Some algorithms include looking at different models such as decision trees, random forests, ADABoost, logistic regression, and others individually as well as combining some of the models using voting classification and experimenting with the weights of each model. We tried different combinations of these algorithms as well as several different methods of combining the models. Then, we implemented One-Hot Encoding to see if it made a difference. We tried to

keep our methods and algorithms as simple and basic as possible. We also looked into removing different features and seeing if that made a difference.

c) What is your final methodology? Walk through it in detail, starting from data pre-processing. Explain all the machine learning algorithm(s) you used as well as the parameters you chose. Also discuss any external tools or libraries that you used.

First we parsed and stored the data using pandas DataFrame by .read_csv(). We deemed the last two features ("ROLE_FAMILY" and "ROLE_CODE") as not useful, so we ignored the last two columns of the data. Then, we used one-hot encoding through sklearn's preprocessing. Following that, we created the following models: linear using logistic regression with C=3, random forest model using randomForestClassifier(), boost model using AdaBoostClassifier() with n_estrimators = 100, and a vote model that combined the above models with equal weights. The final outputs are the voting model's predicted probabilities.

Submission	Files	Public Score	Private Score
Post-Deadline: Fri, 06 May 2016 17:17:49 Edit description	vote2.csv	0.89048	0.88985