NotTooDeep_Learning

Ben Buzzee, Thanh Nguyen, Kellie Mclernon

Approach

- Started off with simple models on original data set
- Continued with the kernel mentioned by Martin (kernel by Tanner Carbonati)
- Attempted to improve modeling efforts with
  - Linear-based models (Lasso, Elastics net, PLS…)
  - Tree-based and deep neural net
- Combined models with the best CV errors

Results

Model	Training CV Error	Kaggle Score
PLS	.114	.1205
Lasso	.112	.1207
Deep Net (DNN)	.094	.123
PLS + XGB / 2		.1178
PLS+DNN/2		.1165
XGB + PLS + DNN / 3		.1157

- Not surprisingly, DNN tends to overfit the training data, but by averaging its predictions with PLS and XGBoost we seemed to smooth out the over-influence of outliers/noise
- Our final model resulted in 137th place on the public leaderboard (Top 7%)

Lessons Learned

- Data exploration
  - Data pre-processing (imputation, standardization, data splitting…) needs to be guided by principles that retain CV error integrity
  - Feature engineering with domain knowledge is essential
- Modelling
  - Xgboost and deep nets can be useful even with less than 1500 cases
- Model evaluation
  - Using a holdout test set in conjunction with cv error helped us get a more accurate sense of prediction error

Future Improvement

- Use optimal/theory based methods for dealing with missing data
- Use domain knowledge to create/expand features
- Ensemble models in a more refined manner instead of taking a simple average
- More parameter tuning (computationally intensive)