"How I Did It" - SeeClickFix Kaggle Contest

By Bryan Gregory & Miroslaw Horbal

- What was your background prior to entering this challenge?

  My professional background is in business intelligence and analytics/reporting and Miroslaw's background is in mathematics, so neither of us has a formal background in machine learning. However, we have both taken multiple online classes in machine learning topics, including Andrew Ng's excellent StanfordX Machine Learning course.

  We also have both competed in quite a few Kaggle competitions in the past year, steadily improving our skills and knowledge with each finish. Kaggle and its community have proven to be a goldmine for anyone interested in learning real-world machine learning techniques outside of academia.

- What made you decide to enter?

  Miroslaw and I both competed independently in the first portion of the SeeClickFix competition, a 24-hour Hackathon, and we performed well. We really enjoyed the dataset and the objective, so competing in the second portion of the contest seemed a natural progression for us both.

  In the second portion of the contest, we both competed independently until about two weeks before the contest completion. At that time, we had both been steadily climbing the leaderboard with myself hitting 1st place on Nov. 16th and Miroslaw right behind me at 4th place. However, J.A. Guerrero (the current #1 Kaggle competitor) had just joined the contest, and he and other top competitors were making steady gains on the leaderboard as well. At that time we decided it was best to hedge our bets and improve our odds of placing in the money by combining our models into a powerful ensemble.

- What preprocessing and supervised learning methods did you use?

  My approach was composed of a segmentation ensemble. This consisted of a distinct base model trained independently on each city and with remote API sourced issues trained together separately as well, for a total of five segments. Because the variables for each of the cities seemed quite distinct (a lot of variable interaction present), this data seemed ideally suited to a segmentation ensemble. One advantage of segmentation is that it greatly reduces the number of samples and the dimensionality that each base model must deal with, so I was able to utilize gradient boosted regressors (GBMs) for most of the base models.

  Miroslaw's approach was more focused on the text analysis side of the data. Using text from the issue summary and description fields, he created a tri-gram TFIDF vector from the entire training set, then

combined it with other features that we had engineered. Because of the high dimensionality, only linear models were practical for this approach, and of those he found that ridge regression performed best.

After we teamed up, our first week was spent strengthening our individual models by integrating some of the stronger features from each other's code into our own and measuring the improvement based on a combination of cross-validation score and leaderboard feedback. Our intuition was that while we did want to keep our models distinct to reduce bias within our ensemble, if a feature was proven to improve both cross-validation and leaderboard scores significantly, than its signal was powerful enough to justify inclusion in both base models.

Then our final week was spent determining the ideal weights for averaging our models' predictions together. A simple 50/50 blend made a surprisingly large gain on the leaderboard, but we then went more in-depth and performed segment based weighting for each of the models. For this we used a linear regression model to derive weights based on optimal cross-validation scores for each segment. Not surprisingly, the findings were that in some segments and targets Miroslaw's model performed better and needed to be weighted higher, while on others mine performed better. Lastly, to avoid overfitting the cross-validation test set we reduced weights to be less extreme if the linear model weighted either of our individual models too strongly.

The features used in our models were largely similar with other top competitors. We found that some of the best signals in the data came from description length, geographic location, issue tag type, issue source, summary/description text, and whether the issue was created on a weekend. From these, we created various binary and one hot encoded features, and we also used a reverse geocoding service to derive zipcodes and neighborhoods from the issue's latitude/longitude fields given in the data. This ended up paying off for us as neighborhoods in particular ended up being powerful predictors, more powerful than simply using latitude and longitude to approximate location.

- <u>What was your most important insight into the data?</u>

  Because this contest was temporal in nature, using time-series models to make future predictions, most competitors quickly realized that proper calibration of predictions was a major factor in reducing error. Even during the initial Hackathon portion of the contest, it became well known on the competition forum that one needed to apply scalars to predictions in order to optimize leaderboard scores.

  But while scaling was common knowledge, our most important insight came in applying our segmentation approach to the scalars. For example, rather than apply one optimized scalar to all predicted views for the entire test set, we applied optimized scalars for each distinct segment of the test set (the remote API sourced issues and the four cities). We then optimized the scalars using a combination of leaderboard

feedback and cross-validation scores. What we found was that each segment responded differently to scaling, so trying to apply one scalar to all issues, as many of our competitors were doing, was not optimal.

- Were you surprised by any of your insights?

  By far our biggest surprise was the effectiveness of creating an ensemble model from our individual models. Initially, before deciding to team up, we were concerned that we may see minimal gain from combining our predictions, but we quickly realized that this was not the case. Even a simple average of the two gave a huge decrease in error.

  The important lesson for us was that two distinctly developed models will yield huge gains when combined together, particularly when the two have a high degree of diversity (variance in errors) which is critical for an ensemble to perform well. We were fortunate that while our models shared many of the same features, Miroslaw's had been designed to take more advantage of the text based features and mine had been designed more to take advantage of the segments in the data.

- Which tools did you use?

  We both used the same stack of Python tools: scikit-learn, PANDAS, and NumPy. This ended up being a fortunate coincidence as it made it easy for us to share code snippets with each other.

- What have you taken away from this competition?

  First, it was a great experience collaborating together as a team. This was our first experience working on a team in a Kaggle contest, and we both agreed that it will not be the last. Working together allowed us to see the same problem from new angles and we learned many new techniques in just the two weeks we worked together. It was enlightening seeing the many creative solutions and insights that we each had developed. Our only regret is not teaming up earlier in the contest.

  Second, we both learned the power of ensembles. Prior to this contest, neither of us had utilized higher level ensembles in previous competitions, always instead focusing on improving one strong model. No longer. Going forward, ensembles will be an important and often used tool in our toolkit. In fact, we were so motivated by our results that Miroslaw and I are developing a Python module to help facilitate the creation and use of ensembles.

  Lastly, we thoroughly enjoyed participating in this contest, and we greatly appreciate SeeClickFix.com, David Eaves, and Kaggle for graciously providing us the opportunity to work with their data. Working with this data set was a blast and we had a great time learning from it and gaining new insight.

BIOS:

Bryan Gregory holds a B.S. in Information Systems from Texas A&M University and an M.B.A. with a concentration in Information Technology from Baylor University. He is a frequent Kaggler with an interest in the practical application of machine learning and business intelligence.

Miroslaw Horbal is has a B.S. in Mathematics with a specialization in Combinatorics and Optimization from the University of Waterloo. He is a self-taught machine learning enthusiast with an addiction to data science competitions.