Select Consensus Algorithm

Components

- Estimate Accuracy Model
- Weighted Mean with Time-weighted Decay

Estimate Accuracy Model

cappedUserEstimatesCount = min(100, userEstimatesCount)
wgt = sqrt(cappedUserEstimatesCount)

accyZScore ~ daysToReport + bias + (userAccyZScore * wgt) + cappedUserEstimatesCount

Variables

- **accyZScore**
- The *negative* absolute error z-score of the estimate (winsorize within the distribution of absolute errors for the release. Where absError = abs(estimate - actual). We negate the z-score to avoid confusion so that a high positive z-score indicates higher accuracy.
- **daysToReport**
- The number of days the estimate was created before the release reports.
- **bias**
- The z-score of the estimate (value) in the distribution of estimates for the release.
- **userAccyZScore**
- The prior historical average of the user's (analyst's) accuracyZScores.
- **cappedUserEstimatesCount**
- The number of prior estimates (globaly) the user has made, bounded between 0 and 100.
- **wgt** Used to scale the user's accuracy, giving more weight to users with a longer history.

Sample Coefficients

variable	as_of	intercept	coefficients

eps	1532145600000	0.127459640622283	-0.00711816012940946, 0.162417763548987, 0.123017803163136, -0.000203414215003563
eps	1530936000000	0.127402429861642	-0.00704488580938566, 0.15412156598128, 0.120382285867316, -0.00021079801801204
eps	1530763200000	0.127539151110316	-0.00704624591880856, 0.153921223788591, 0.120154101817247, -0.000211517531187198

Weighted Mean with Time-weighted Decay

Estimate Confidence

Models are generated on a monthly basis from a one year window of the estimates whose releases have reported (we still use the full user accuracy history). As new estimates are created they are transformed into a feature vector that is fed into the model to predict the **accyZScore** of that particular estimate. From the **accyZScore** each estimate is assigned a **confidence** by transforming the z-score into a value between 0.0 and 1.0 via the cumulative probability density function.

Time-weighted decay

Once the confidence of the estimate has been generated, we multiply it with a time-weighted decay factor based on the estimate's **daysToReport**.

decayedConfidence = confidence * exp(-0.45 * daysToReport)

Weighed Mean

Finally to generate the select consensus, we take all the estimates and their corresponding **decayedConfidences** and generate the weighted mean.

selectConsensus = weightedMean(estimates, decayedConfidences)