**Experiment Details**

* What we are trying to accomplish: predict average star ratings for Pittsburgh businesses
* We want to know the accuracy or precession of our system (recall doesn’t apply here since the number of answers = number of guesses)
* Accuracy = precision = # correct guesses/# guessed
* Run 4 different experiments, each with different definition of correct guess (different goal)

**Experiments+Results**

1. Strict
   1. Goal: calculate accuracy of system where correct guess = actual star value for business
   2. Expected findings: low accuracy due to many assumptions of system(relatively simple sentiment analyzer, normalized on just user average, etc)
   3. Result: ≈20%
   4. Reaction: Honestly expected less (like 10%) due to normalizing on average user rating, and rounding of star calculations. Making these type of assumptions is not ideal for a strict evaluation. Nowhere to go but up.
2. Slightly Relaxed (0.5)
   1. Goal: calculate accuracy of system where correct guess = guess that is within 0.5 stars of actual star value for business
   2. Expected findings: higher than strict experiment. Our assumptions, normalization, rounding of sentiment predictions will not hurt us as much here.
   3. Result: ≈ 65%
   4. Reactions: Our system may not predict the exact star rating, but it does predict most ratings in very close proximity to the actual rating. Perhaps a little more effort/tuning will result in much greater strict prediction.
3. Relaxed (1)
   1. Goal: calculate accuracy of system where correct guess = guess that is within 1 star of actual star value for business
   2. Expected findings: much higher accuracy. Given a fairly large range, our system should be able to effectively guess a correct score if it is working properly.
   3. Result: ≈ 87%
   4. Reactions: Expected this type of accuracy. As stated before, if our system is working as intended, it should be able to predict a reasonably accurate score given large acceptance range.
4. Very Relaxed (1.5)
   1. Goal: calculate accuracy of system where correct guess = guess that is within 1.5 stars of actual star value for business
   2. Expected findings: near perfect accuracy. This is a bit ridiculous to consider. More of a check to see if system is working correctly.
   3. Result: ≈ 98%
   4. Reactions: Good. There are serious problems with the system if it is unable to predict most accurate scores given an absurdly large range of acceptance. This is where normalizing on user average reviews hurts system from getting even closer to 100%. If user average rating = 4.5, but actual review score is 1 due to terrible restaurant, normalization of any corresponding sentiment score will always make score inaccurate. In other words: normalizing on user average star rating will negatively effect extreme cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Experiments** | | | |
| **Type** | **Strict** | **Slightly Relaxed (0.5)** | **Relaxed (1)** | **Very Relaxed (1.5)** |
| **Accuracy** | **0.198730159** | **0.65335097** | **0.868465608** | **0.978315697** |

Observation: Predicted seems to be conservative copy of actual data (follows same pattern as actual, just closer to average rating)

Baseline Experiments:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Baseline Experiments (Rating = 3)** | | | |
| **Type** | **Strict** | **Slightly Relaxed (0.5)** | **Relaxed (1)** | **Very Relaxed (1.5)** |
| **Accuracy** | **0.689859** | **0.934391534** | **0.983068783** | **0.998236332** |