Data Mining Project Log (Baseball: Strike/Ball)

4/17/2021

* Data Cleaning
  + Get rid of unnecessary fields:
    - Unique ID fields.
    - Fields with same value throughout data set.
  + Move predicted attribute to last position.
  + Set defined prediction values with vlookup in excel: Strike or Ball:
    - If the ball was called strike, it was swung at, fouled, or hit 🡪 Then I will want to predict strike for this project.
    - If called ball, hit by pitch, or ball in dirt 🡪 Then ball.
  + Set up files I might need:
    - Excel file
    - CSV file
    - Arff file
  + Use Weka Preprocess for files I might need
    - Numeric
    - Discretized version (bins = 4; FindNumBins = True; rest are defaults)
    - Remove missing (too many missing values to remove I think, not applicable)
    - Fully Discretized Replace missing values (default settings)

4/18/21

\*\*\*\*\*\*\*\*Experiment 1 was redone in Experiment 2 due to more Data cleaning needing to be done\*\*\*\*

* Experiment 1 (Find a potential baseline model)
  + Using 10 fold cross-validation for all models
    - Chance for predicted attribute is 50%
    - Numeric ZeroR (63.6364 %; default settings)
      * Predicted Strike
    - Discretized ZeroR (63.6364 %; default settings)
      * Predicted Strike
    - Numeric OneR (68.1818 %; default settings, 6 MinBucketSize)
      * Uses px
    - Discretized OneR (72.3485 %; default settings, 6 MinBucketSize)
      * Uses pz

Test Accuracy Table For Experiment 1:

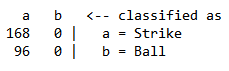
|  |  |  |
| --- | --- | --- |
|  | Numeric | Discretized |
| OneR | 68.1818% | 72.3485% |
| ZeroR | 63.6364% | 63.6364% |

* + ZeroR will be my baseline for comparisions, as I am afraid OneR is getting too high of an accuracy rate to be a baseline model.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4/21/2021

* Experiment 2: More data cleaning, Redo file set up, and Redo baselines models. (Reworking Experiment 1)
  + Models were using fields that I did not know what they necessarily meant such as pz and px. More data cleaning needs to be done so I know what all the attributes are and I can fully understand the models. That is the difference between Experiment 1 and 2
  + Reworked Files in Second Arff files folder:
    - Numeric
    - Discretized version (bins = 4; FindNumBins = True; rest are defaults)
    - Fully Discretized Replace missing (default settings)
  + Baseline Redo: Using 10 fold cross-validation for all models
    - Chance for predicted attribute is 50%
    - Numeric ZeroR
      * 63.6364% test accuracy
      * Default settings
      * Mean absolute error: 0.4632
      * Predicting majority class which is Strike
      * Much higher accuracy than pure chance
    - Discretized ZeroR
      * 63.6364 % test accuracy
      * Default settings



* + - * Predicting majority class which is Strike
      * Much higher accuracy than pure chance
    - Numeric OneR
      * 70.0758 % test accuracy
      * Default settings with 6 MinBucketSize
      * Mean absolute error: 0.2992
      * Uses zone attribute to make its predictions
      * Accuracy is even higher than ZeroR, this may be too high of a baseline
    - Discretized OneR
      * 74.6212 % test accuracy
      * default settings with 6 MinBucketSize



* + - * Uses zone attribute to make its predictions
      * Highest accuracy seen of the potential baselines. I am afraid using OneR as a baseline for comparison may be too optimistic, so therefore ZeroR will be my baseline for the project.

Test Accuracy Table For Experiment 2:

|  |  |  |
| --- | --- | --- |
|  | Numeric | Discretized |
| OneR | 70.0758 % | 74.6212 % |
| ZeroR | 63.6364% | 63.6364% |

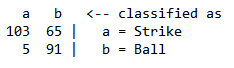
* + ZeroR will be my baseline for comparisions, as I am afraid OneR is getting too high of an accuracy rate to be a baseline model.
  + I am very pleased with this experiment, as the algorithms that I believe are the simplest are doing very well. This gives me optimism for the algorithms to follow because we have a good baseline in ZeroR and OneR did performed very well in terms of test accuracy.

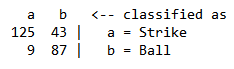
4/22/2021

* Experiment 3: Starting off with Rules
  + - Numeric OneR
      * 70.0758 % test accuracy
      * Default settings with 6 MinBucketSize
      * Mean absolute error: 0.2992
      * Uses zone attribute to make its predictions. This model uses high zone numbers to make its predictions instead of utilizing the whole zone.
      * Model does fairly well, by using one attribute we were able to predict correctly at a 70% rate.
    - Discretized OneR
      * 74.6212 % test accuracy
      * default settings with 6 MinBucketSize



* + - * Uses zone attribute to make its predictions as well but utilizes more zone numbers since the values are bucketed.
      * Better accuracy than numeric model. My theory for this is because the discretized version has values bucketed and forces OneR to use a wider range of zone values.
    - Discretized Missing Replaced OneR
      * 73.4848 % test accuracy
      * default settings with 6 MinBucketSize



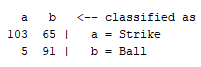
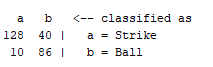
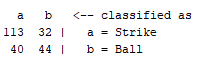
* + - * Uses zone attribute to make its predictions. Basically the same model as the first discretized version, but slightly lower accuracy because some missing values were replaced with artificial data.
    - Numeric JRip
      * 80.303 % test accuracy
      * Default settings
      * Mean absolute error: 0.2796
      * Only Zone and Last Pitch at bat attributes used
      * Very simple rule set for JRip, but the accuracy is very high. Interesting to see that we can get 80% correct on test instances using these 2 attributes. What the model is telling me is that if the pitch is high in the strike zone and it is not the last pitch in the at bat, then ball. Otherwise predict strike. This makes sense to me because high pitches have a good chance of sailing above the strike zone. Also if it is not the last pitch in the at bat, then pitchers have more pitches to utilize so they are not pressured to throw a strike.
    - Discretized JRip
      * 81.4394 % test accuracy
      * Default settings
      * 
      * Model is very similar to numeric model, because it uses the same 2 attributes (zone and last pitch in at bat indicator) to make its predictions. The rule set is basically the same, but the difference is that the cutoffs in the buckets for the discretized dataset allow more data to be in the rule set and therefore we have more of a chance to fine tune to predict correctly. I believe the simplicity in the rule set is a benefit here, because a strike does not need to rely on a ton of complexities in the data.
    - Discretized Missing Replaced JRip
      * 80.303 % test accuracy
      * Default settings
      * 
      * Model is the exact same as the Discretized with missing values, but artificial data replaced all missing values. I believe this is the reason for the slight drop in accuracy. The data was just skewed slightly in a way were we miss-classified some records.
    - Discretized Missing Replaced Prism
      * 61.3636 % test accuracy
      * Default settings
      * 
      * Discretized with Missing’s Replaced is the only applicable dataset for Prism
      * This Prism model is fairly complex. It is using many attributes throughout the rule set including fields such as pitchers and batters. The first few rules make total sense to me as the human in the loop. The rules without batter and pitcher names make sense in their predictions as the human in the loop as well. Some examples of predictions include if a pitch height is not too high and not too low, then predict strike; If a pitch is a knuckle curveball within the first 3 innings then predict ball, and many others that make sense as pitch outcomes. These two are very likely outcomes for the inputs given for a given pitch. Where the model starts to fall off, is when it tries in to get granular in the player data. Prism is trying to tie in a specific batter and pitcher for whether a pitch is a strike or not. This could be good information to know, but the accuracy is not as high as some other models. This tells me a lot about my data, in the fact that we do not need very complex models to come out with a high accuracy for predictions, but instead may be better suited with a simpler model. Prism on this dataset did worse than our baseline, making it not a very reliable model in terms of this project.

Test Accuracy Table For Experiment 3: Rules

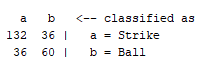
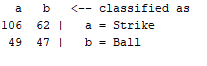
|  |  |  |  |
| --- | --- | --- | --- |
|  | Numeric | Discretized | Dicretized Missing removed |
| OneR | 70.0758 % | 74.6212 % | 73.4848 % |
| JRip | 80.303 % | 81.4394 % | 80.303 % |
| Prism | N/A | N/A | 61.3636 % |

* + I am mostly pleased with this experiment. I am disappointed with the outcome of Prism, but I expected some of the more complex algorithms to overcomplicate the data. However, JRip has the best accuracies I have seen so far across the board. I am excited from the insights I was able to draw from OneR and JRip, and Prism gave me some valuable information even for having a low accuracy.
  + I Next want to try Trees because the models between trees and rules are somewhat similar in the fact the trees have a rule for a given node in the tree. I think it will be useful compare the models and accuracy between trees and rules.

4/24/2021

* Experiment 4: Trees
  + I decided to try the Decision Stump algorithm based on the fact that I thought It would be a good starting point for tree model driven algorithms. I thought Decision Stump would be a good starting point because it uses one attribute to form the tree model. It is a simple tree model that uses the root attribute to base its predictions from by creating as many leaf’s as needed to develop predictions based on that given root. I would compare it to OneR with rules because they both use 1 attribute for predictions.
    - Numeric Decision Stump
      * 72.3485 % test accuracy
      * Default settings
      * Mean absolute error: 0.3222
      * Here we have another model that is using the zone attribute. Zone seems to be a popular choice for the algorithms to base their predictions on. However, the zone field is not a dead give away if the pitch is a ball or strike because the algorithms are not classifying at 100%. The accuracy is comparable to OneR in the 70% range, but Decision Stump does slightly better in this case.
    - Discretized Decision Stump
      * 74.6212 % test accuracy
      * Default settings
      * 
        + A lot more predictions for ball when it was actually strike than vice versa.
      * The model is similar to the numeric model in the fact that it uses the zone attribute again. However, the buckets for the discretized data make the thresholds a bit different. For the numeric data the threshold was zone at 8.5 above ball, below stike. For discretized the threshold was if zone is in the highest range , 10.75-infinity, then ball otherwise strike. Both had missing values classified as strike. The discretized came out with a slightly higher accuracy. Again, comparing to OneR the discretized sets had virtually the same accuracy and numeric sets were very close.
    - Discretized Missing’s Replaced Decision Stump
      * 73.4848 % test accuracy
      * Default settings
      * 
      * The model is the same as the discretized version, however we are not predicting strike for missing values anymore because there are none. Instead, by looking at the confusion matrices we are predicting ball more often. This is where I believe the slight drop in accuracy is coming from. All models with Missing’s Replaced have had about a 1% drop in accuracy compared to the Discretized with missing’s so far.
    - Next, I will try J48 to look at a more complex algorithm within the tree algorithm set.
    - Numeric J48
      * 79.5455 % test accuracy
      * Default settings
      * Root mean squared error: 0.3951
      * This model is more complex than the Decision Stump model as expected. The model derived same very good insights though as the human in the loop. It using some attributes that I believe greatly impact whether a given pitch is a strike or not such as pitcher pitch type confidence, last pitch in at bat, amount of strikes, and amount of balls. The model also uses zone as the root node and in a few other spots throughout the tree. My take home message from this model is that if the zone is not too high and if the pitcher is at least 88% confident in that pitch type, then predict strike. If the zone is high, there are no strikes in the at bat, and there is not a lot of break on the ball then predict ball. If there is a lot of break on the ball with the same predecessors, then predict strike. This raises a little bit of concern on that part of the tree because pitches are harder to control with more spin on the ball so more break leading to a strike is not a great correlation. However, this does not cause enough concern to where I do not trust the model.
    - Discretized J48
      * 81.8182 % test accuracy
      * Default settings
      * 
      * This model is different from the numeric data with J48. It is a bit simplified by only using 2 attributes, zone and last pitch in at bat, as well as being a smaller overall tree. My take home message from this model is that it is saying any pitch that is not overly high, predict strike. If the pitch is high, and it is not the last pitch in the at bat, then ball. If the pitch is high and it is the last pitch in the at bat, then predict strike. Some of the model I agree with, in the fact that pitchers will tend to try to throw a strike if it is the last pitch possible in the at bat to try and get the batter out. Where I am bit questionable is that the model is not taking into account if the pitch is low. The model is still giving some valuable insight so I will not say it is unreliable, but this is a case where it is necessary to have a human in the loop to dissect these trends. The model did very well in accuracy so it is drawing relevant conclusions.
    - Discretized Missing’s Replaced J48
      * 81.0606 % test accuracy
      * Default settings
      * 
      * This model with missing’s replaced is the same as the discretized data set with J48. The difference comes in missing data being replaced which again lead the algorithm to predict ball more often which lead to a slight decrease in accuracy.
    - Discretized Missing’s Replaced ID3
      * 59.4697 % test accuracy
      * Default settings
      * 
      * Discretized with Missing’s Replaced is the only applicable dataset for ID3
      * ID3’s model is very comparable to Prism with the Discretized with missing values replaced data set. The accuracy of the model is below our baseline from ZeroR and the details of the model are complex. This is another case where the algorithm is trying to base whether a pitch is a ball or strike or not based off of batter names in a lot of cases. The model uses multiple attributes throughout such as score, inning, pitch number within at bat, balls, strikes, pitch break, pitch type, and ball spin direction. Complexity in this model is hurting the classification accuracy, because as I saw before this data can be analyzed more efficiently with simpler methods. Some of the information within the tree could be useful to know such as whether a given pitch type such as a slider will be a ball or strike when thrown against a given batter. The information is not useless, but in terms of accuracy the model is underperforming compared to the others.
* Test Accuracy Table For Experiment 4: Trees

|  |  |  |  |
| --- | --- | --- | --- |
|  | Numeric | Discretized | Dicretized Missing removed |
| Decision Stump | 72.3485 % | 74.6212 % | 73.4848 % |
| J48 | 79.5455 % | 81.8182 % | 81.0606 % |
| ID3 | N/A | N/A | 59.4697 % |

* + This is another case of mixed reactions to the models. I cannot expect every model to great, but like Prism, ID3 struggled. I believe I was still able to derive valuable insights from all the models which is a plus for the experiment. Decision Stump did well and J48 did very well too.
  + Next, I want to go to Naïve Bayes, IB1, and KStar. These models will look different and most likely will not be comparable in terms of model details compared to rules and trees. Experiment 5 can be thought of in terms of trying miscellaneous models that are not necessarily in the same group of algorithms.
* Experiment 5: Naïve Bayes, IB1, and KStar
  + - Numeric Naïve Bayes
      * 73.8636 % test accuracy
      * Default settings
      * Mean absolute error: 0.2813
      * Naïve Bayes is a direct contrast to algorithms that we have seen earlier such as OneR and Decision Stump that only use 1 attribute. Naïve Bayes takes into account all attributes and assumes they are equally important to the prediction as well as being independent from each other. This model does fairly well with a classification rate of 73%.The model gives some valuable insights on both nominal attributes and numeric attributes within the dataset. An example of an insight I was able to gain from the model is that sliders where very productive in this data because it was thrown 90 times and 60 of the sliders were strikes. Another example of an insight I was able to take away is that the average strike had a break angle of about 15 where as balls had an average break angle of about 17. This makes sense because the more break a ball has on it, the less accuracy the pitcher will have on the pitch leading to more balls.
    - Discretized Naïve Bayes
      * 72.7273 % test accuracy
      * Default settings
      * 
      * There is a minor drop off in accuracy of the discretized set with Naïve Bayes compared to numeric. However, this model is producing valuable insight as well. I am able to derive some helpful knowledge about whether a given pitch will be a strike or not. One example is that if the pitch is the last pitch in the at bat, it is vastly more likely to be a strike as compared to a ball. Another example is that high pitches are more likely to be a ball than a strike because the hitters are not swinging and the umpire is declaring that it is not in the strike zone.
    - Numeric IB1
      * 53.4091 % test accuracy
      * Default settings
      * Mean absolute error: 0.4659
      * This algorithm uses the nearest neighbor to classify a given test instance. It will find the training instance closest to the test instance and base the prediction based on the training instances class. The model did not do very will which tells me that similar data for a given record does not mean it correlates to a given prediction. Instead, the data can similar metrics but be classified differently.
    - Discretized IB1
      * 57.9545 % test accuracy
      * Default settings
      * 
      * The increase in accuracy with the discretized data set tells me that the bucketing of the numeric data helped in this algorithm. The numeric data allowed for more discrepancies between similar records whereas the bucketed data allowed the algorithm to predict correctly more often.
    - Numeric KStar
      * 61.7424 % test accuracy
      * Default Settings
      * Root mean squared error: 0.5983
      * KStar using a similarity function to find similar training instances for a given test instance. KStar uses an entropy-based distance function to find similar training instances for predictions. The model does better than IB1 which only found the most similar training instance, but the accuracy is still below our baseline from ZeroR. Again, this tells me there does not need to be a lot of similarities for a given record to be classified as a certain outcome.
    - Discretized KStar
      * 59.0909 % test accuracy
      * Default settings
      * 
      * Unlike with IB1, the Discretized data set did worse with KStar. I would say this is because the algorithm is not just finding 1 similar training instance, but instead multiple similar training instances. For this algorithm the granularity within the data helps to find the most similar training instances which is why I believe there is a drop in accuracy.
* Test Accuracy Table For Experiment 5:

|  |  |  |
| --- | --- | --- |
|  | Numeric | Discretized |
| Naïve Bayes | 73.8636 % | 72.7273 % |
| IB1 | 53.4091 % | 57.9545 % |
| KStar | 61.7424 % | 59.0909 % |

* + This experiment was a surprise to me. Naïve Bayes performed well, but IB1 and KStar struggled with the data as they did not even outperform my baseline model in ZeroR. This tells me that similar data records are not interchangeable when it comes to making predictions. Similar data can produce different results which is not necessarily a good thing because it can make it slightly harder to draw conclusions.