Predicting the Success of Baseball Pitchers

Yilmaz Machine Learning 1 - 2022-2023

Wilson Chen and Bradley Cao

Contents

1 Introduction

1.1 Overview & Rationale

Major League Baseball (MLB) has become more and more statstics-driven over the years. Coaches use these statistics to evaluate player performance and make informed decisions about players, for example in draft reports. Contracts in MLB are often worth millions of dollars, so having statistics that measure player performance is extremely important. We wanted to see just how accurate predictions with these statistics are. For our project, we decided to look at a dataset on baseball pitchers. This dataset with 547 instances and 33 attributes contained information on different statistics recorded for pitchers. It came from the 2021 regular season, and only contained pitchers who had faced at least 100 batters. We wanted to see if we could use these statistics to predict whether or not a pitcher had a winning record.

1.2 Data Set Information

We got our original dataset from https://www.baseballsavant.mlb.com. It contained 33 attributes:

- **last_name**
- **first_name**
- **player_id**
- **year**
- **p_game**
- **p_total_hits**
- **p_home_run**
- **p_strikeout**
- **p_walk**
- **p_k_percent**
- **p_bb_percent**
- **batting_avg**
- **slg_percent**
- **on_base_percent**
- **on_base_plus_slg**
- **p_earned_run**

- **p_win**
- **p_loss**
- **p_era**
- **p_rbi**
- **p_called_strike**
- **p_unearned_run**
- **exit_velocity_avg**
- **launch_angle_avg**
- **sweet_spot_percent**
- **barrel_batted_rate**
- **hard_hit_percent**
- **meatball_percent**
- **pitch_hand**
- **fastball_avg_speed**
- **n_fastball_formatted**
- **offspeed_avg_speed**
- **n_offspeed_formatted**

These statistics were collected by MLB or MLBStatCast, a partner company that specializes in collecting data from MLB games.

2 Preprocessing

2.1 Missing and Redundant Values

We first started by removing redundant values, such as OPS. The OPS attribute is just the sum of the On Base Percentage (**OBP**) and Slugging (**SLG**), so we can safely remove it without losing any information. Furthermore, there were missing values for the attributes Offspeed% and Offspeed Average MPH. We decided to remove these attributes altogether, as these values are missing because not all pitchers have an offspeed pitch; filling this data with an average would not be representative, and removing these attributes helps with dimensionality reduction. We also removed all attributes that did not relate to pitcher's performance (**name**, **year**, **player ID**, etc.). We can safely remove these attributes since they provide zero information.

2.2 Class Labels

Since our data did not come with class labels, we had to create them ourselves. Using the **p_win** and **p_loss**, we created our own column, titled **winning** to keep track of whether or not the pitcher had a winning record. For pitchers that had an even record, with the same number of wins and losses, we classified them as having a winning record. We then removed the attributes **p_win** and **p_loss** because the information they give us is given by our new class variable.

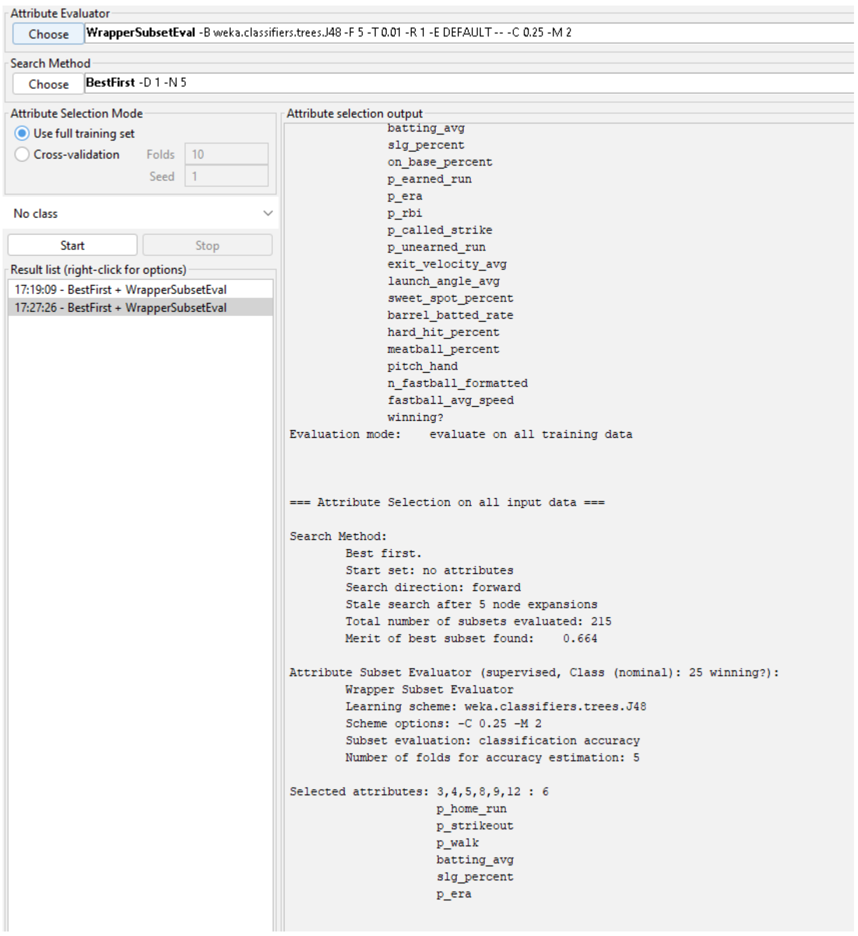
2.3 Dimensionality Reduction

2.3.1 Self Selection

The first method we used was self selection. We picked 6 attributes that we believed, with our knowledge of baseball, would be the best indicators of a pitcher having a winning record.

2.3.2 WrapperSubsetEval

WrapperSubsetEval is the first of the WEKA built-in attribute selection methods. It uses a learning scheme to evaluate attribute sets, and uses cross validation to estimate its accuracy for each set.



Attribute Evaluator: **WrapperSubsetEval** -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R 1 -E DEFAULT -- -C 0.25 -M 2

Search Method: **BestFirst** -D 1 -N 5

Attribute Selection Mode: [✓ Use full training set] Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)
- 17:19:09 - BestFirst + WrapperSubsetEval
- 17:27:26 - BestFirst + WrapperSubsetEval

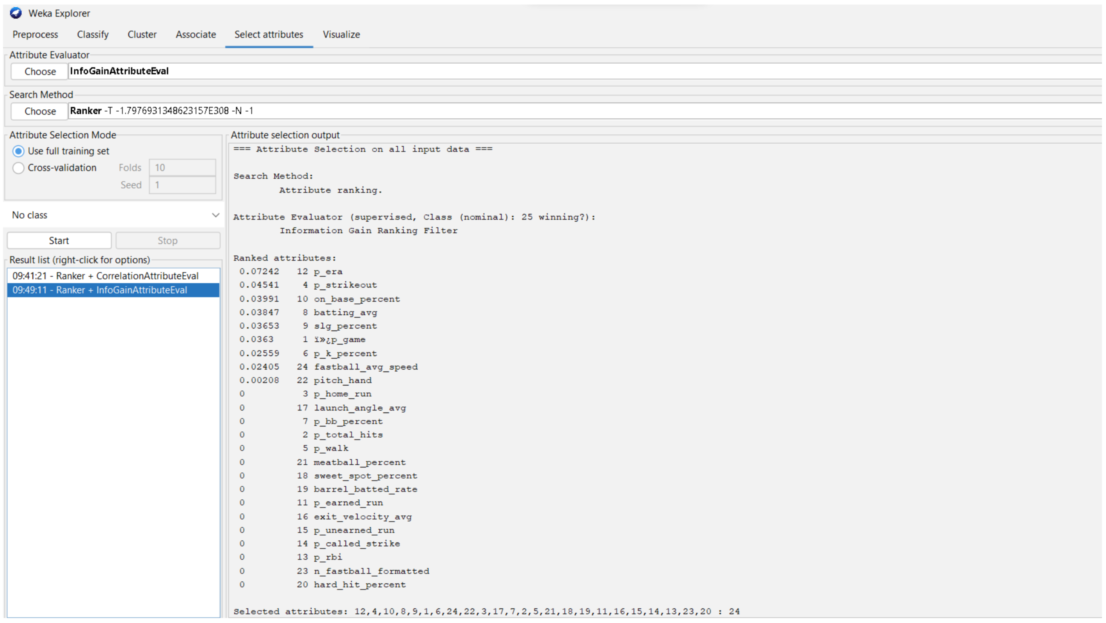
Attribute selection output

```
              batting_avg
              slg_percent
              on_base_percent
              p_earned_run
              p_era
              p_rbi
              p_called_strike
              p_unearned_run
              exit_velocity_avg
              launch_angle_avg
              sweet_spot_percent
              barrel_batted_rate
              hard_hit_percent
              meatball_percent
              pitch_hand
              n_fastball_formatted
              fastball_avg_speed
              winning?
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 215
        Merit of best subset found:    0.664

Attribute Subset Evaluator (supervised, Class (nominal): 25 winning?):
        Wrapper Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Subset evaluation: classification accuracy
        Number of folds for accuracy estimation: 5

Selected attributes: 3,4,5,8,9,12 : 6
                     p_home_run
                     p_strikeout
                     p_walk
                     batting_avg
                     slg_percent
                     p_era
```

Figure 1: WEKA screenshot showing the results of WrapperSubsetEval

2.3.3 InfoGainAttributeEval

InfoGainAttributeEval is the second WEKA built-in attribute selection method. It measures the information gain of each attribute relative to the class variable and gives us the ones with the most information gain, which are the most predictive of the class variable.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: Use full training set; Cross-validation, Folds 10, Seed 1

No class

Result list (right-click for options)
09:41:21 - Ranker + CorrelationAttributeEval
09:49:11 - Ranker + InfoGainAttributeEval

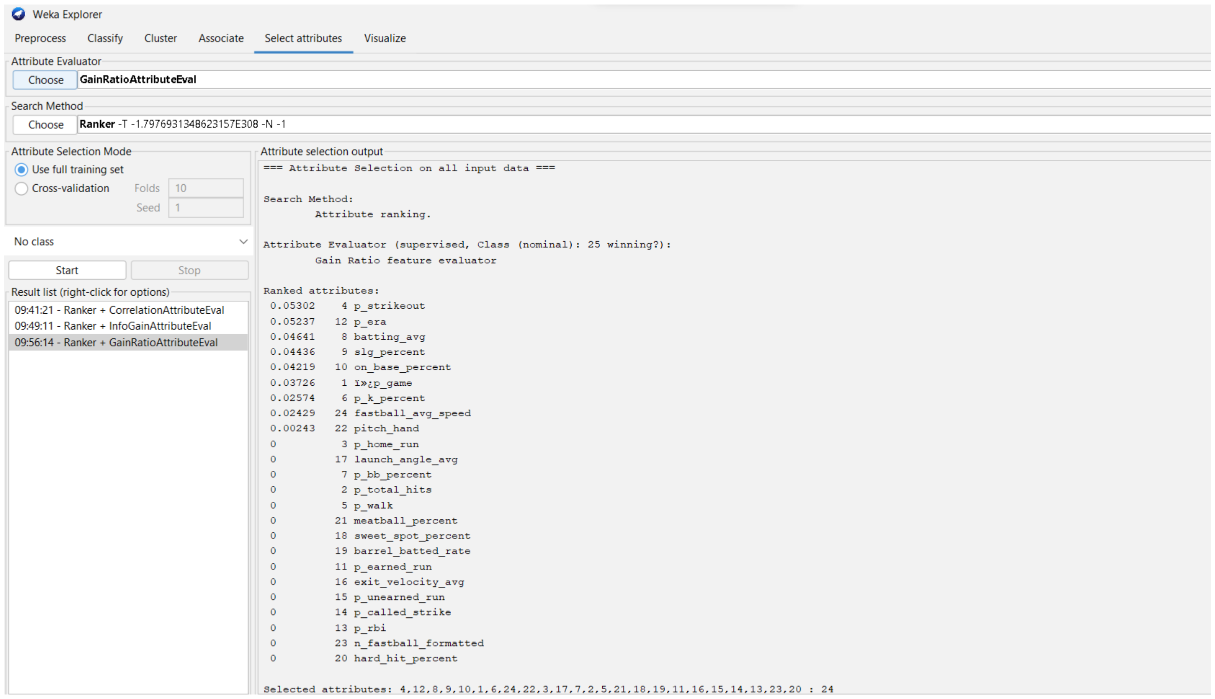
Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 25 winning?):
        Information Gain Ranking Filter

Ranked attributes:
 0.07242   12 p_era
 0.04541    4 p_strikeout
 0.03991   10 on_base_percent
 0.03847    8 batting_avg
 0.03653    9 slg_percent
 0.0363     1 ï»¿p_game
 0.02559    6 p_k_percent
 0.02405   24 fastball_avg_speed
 0.00208   22 pitch_hand
 0          3 p_home_run
 0         17 launch_angle_avg
 0          7 p_bb_percent
 0          2 p_total_hits
 0          5 p_walk
 0         21 meatball_percent
 0         18 sweet_spot_percent
 0         19 barrel_batted_rate
 0         11 p_earned_run
 0         16 exit_velocity_avg
 0         15 p_unearned_run
 0         14 p_called_strike
 0         13 p_rbi
 0         23 n_fastball_formatted
 0         20 hard_hit_percent

Selected attributes: 12,4,10,8,9,1,6,24,22,3,17,7,2,5,21,18,19,11,16,15,14,13,23,20 : 24
```

Figure 2: WEKA screenshot showing the results of InfoGainAttributeEval

2.3.4 GainRatioAttributeEval

GainRatioAttributeEval is the third WEKA built-in attribute selection method. This method is similar to the previous, but instead measures the gain ratio for each attribute relative to the class variable. The gain ratio is the ratio of information gain to intrinsic information.



Attribute Evaluator: GainRatioAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: Use full training set; Cross-validation Folds 10 Seed 1

Result list (right-click for options)
09:41:21 - Ranker + CorrelationAttributeEval
09:49:11 - Ranker + InfoGainAttributeEval
09:56:14 - Ranker + GainRatioAttributeEval

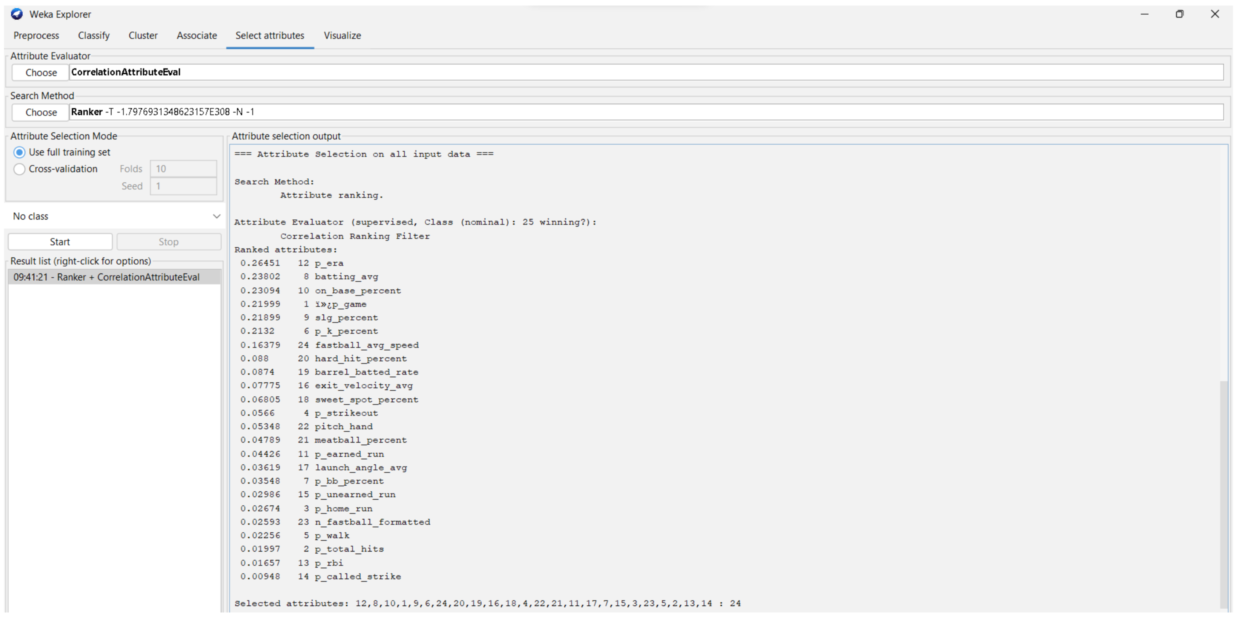
Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 25 winning?):
        Gain Ratio feature evaluator

Ranked attributes:
 0.05302    4 p_strikeout
 0.05237   12 p_era
 0.04641    8 batting_avg
 0.04436    9 slg_percent
 0.04219   10 on_base_percent
 0.03726    1 ï»¿p_game
 0.02574    6 p_k_percent
 0.02429   24 fastball_avg_speed
 0.00243   22 pitch_hand
 0          3 p_home_run
 0         17 launch_angle_avg
 0          7 p_bb_percent
 0          2 p_total_hits
 0          5 p_walk
 0         21 meatball_percent
 0         18 sweet_spot_percent
 0         19 barrel_batted_rate
 0         11 p_earned_run
 0         16 exit_velocity_avg
 0         15 p_unearned_run
 0         14 p_called_strike
 0         13 p_rbi
 0         23 n_fastball_formatted
 0         20 hard_hit_percent

Selected attributes: 4,12,8,9,10,1,6,24,22,3,17,7,2,5,21,18,19,11,16,15,14,13,23,20 : 24
```

Figure 3: WEKA screenshot showing the results of GainRatioAttributeEval

2.3.5 CorrelationAttributeEval

CorrelationAttributeEval is the final WEKA built-in attribute selection method we used. This method measures the Perason's correlation between an attribute and the class variable. The Pearson's correlation measures the strength of a linear relationship between two variables, and can range from -1 to 1.



Attribute Evaluator: CorrelationAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: [✓ Use full training set] Cross-validation Folds 10 Seed 1

Result list (right-click for options): 09:41:21 - Ranker + CorrelationAttributeEval

Attribute selection output

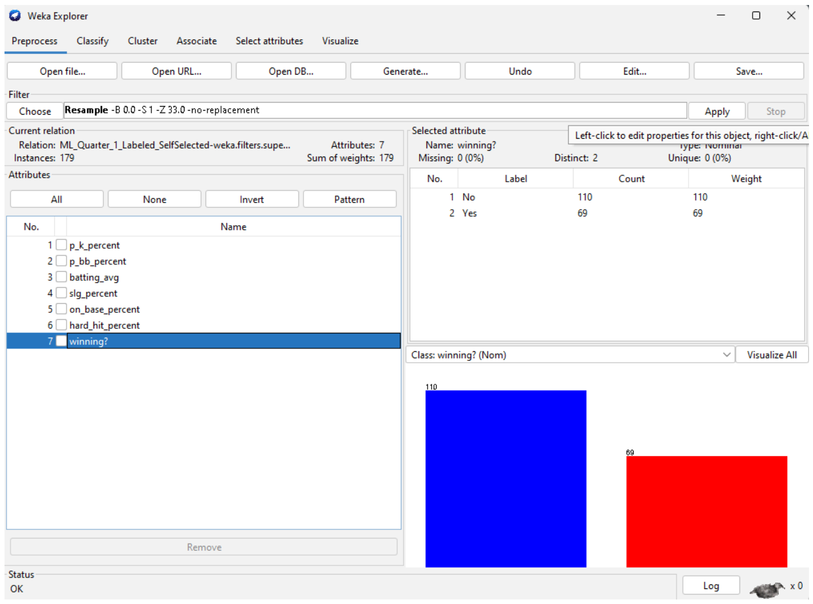
```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 25 winning?):
        Correlation Ranking Filter
Ranked attributes:
 0.26451   12 p_era
 0.23802    8 batting_avg
 0.23094   10 on_base_percent
 0.21999    1 ï»¿p_game
 0.21899    9 slg_percent
 0.2132     6 p_k_percent
 0.16379   24 fastball_avg_speed
 0.088     20 hard_hit_percent
 0.0874    19 barrel_batted_rate
 0.07775   16 exit_velocity_avg
 0.06805   18 sweet_spot_percent
 0.0566     4 p_strikeout
 0.05348   22 pitch_hand
 0.04789   21 meatball_percent
 0.04426   11 p_earned_run
 0.03619   17 launch_angle_avg
 0.03548    7 p_bb_percent
 0.02986   15 p_unearned_run
 0.02674    3 p_home_run
 0.02593   23 n_fastball_formatted
 0.02256    5 p_walk
 0.01997    2 p_total_hits
 0.01657   13 p_rbi
 0.00948   14 p_called_strike

Selected attributes: 12,8,10,1,9,6,24,20,19,16,18,4,22,21,11,17,7,15,3,23,5,2,13,14 : 24
```

Figure 4: WEKA screenshot showing the results of CorrelationAttributeEval

3 Splitting into Testing and Training

For each one of these attribute selection methods, we removed all attributes except the ones selected and the class varaible. We then split the instances into a testing set and a training set using WEKA. For this, we used WEKA's supervised filter **Resample**, which we configured to produce a random subset of the dataset without replacement. We ran it once, with the size of the output dataset to be 33% of the whole to get the testing sets, then used the **InvertSelection** feature to get the training sets. The training sets had 368 instances, with 226 classified as **yes** and 142 classified as **no**, while the testing sets had 179 instances, with 110 classified as **yes** and 69 classified as **no**.

3.1 SelfSelection



Filter: **Resample** -B 0.0 -S 1 -Z 33.0 -no-replacement

Current relation
Relation: ML_Quarter_1_Labeled_SelfSelected-weka.filters.supe... Attributes: 7
Instances: 179 Sum of weights: 179

Attributes

No.	Name
1	p_k_percent
2	p_bb_percent
3	batting_avg
4	slg_percent
5	on_base_percent
6	hard_hit_percent
7	winning?

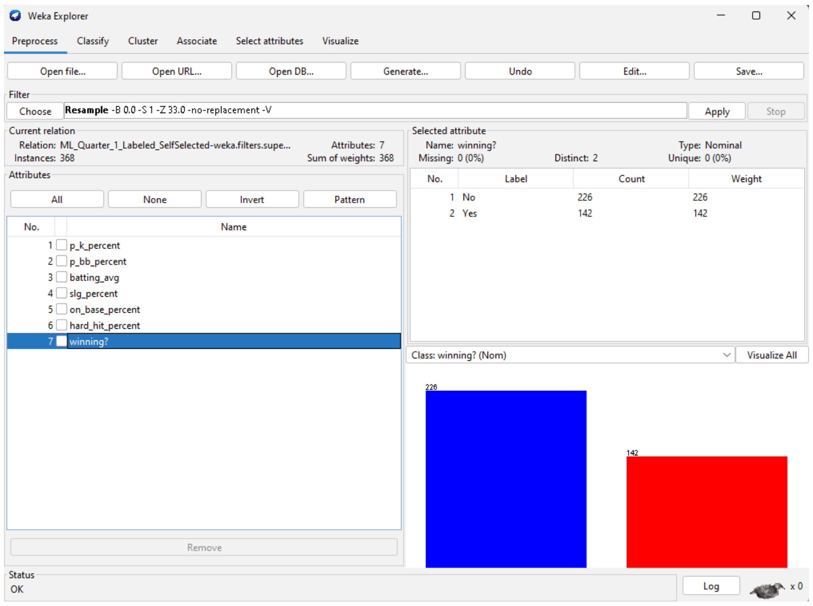
Selected attribute
Name: winning? Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	No	110	110
2	Yes	69	69

Class: winning? (Nom)

Status
OK

Figure 5: SelfSelection Test Set



Filter: **Resample** -B 0.0 -S 1 -Z 33.0 -no-replacement -V

Current relation
Relation: ML_Quarter_1_Labeled_SelfSelected-weka.filters.supe... Attributes: 7
Instances: 368 Sum of weights: 368

Attributes

No.	Name
1	p_k_percent
2	p_bb_percent
3	batting_avg
4	slg_percent
5	on_base_percent
6	hard_hit_percent
7	winning?

Selected attribute
Name: winning? Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

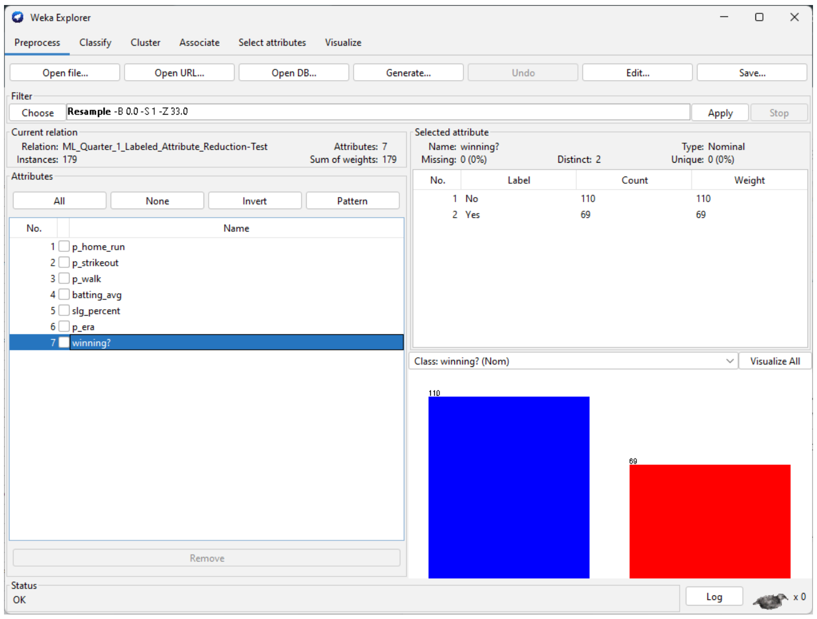
No.	Label	Count	Weight
1	No	226	226
2	Yes	142	142

Class: winning? (Nom)

Status
OK

Figure 6: SelfSelection Train Set

3.2 WrapperSubsetEval



Filter
Resample -B 0.0 -S 1 -Z 33.0

Current relation
Relation: ML_Quarter_1_Labeled_Attribute_Reduction-Test   Attributes: 7
Instances: 179   Sum of weights: 179

Selected attribute
Name: winning?   Type: Nominal
Missing: 0 (0%)   Distinct: 2   Unique: 0 (0%)

No.	Label	Count	Weight
1	No	110	110
2	Yes	69	69

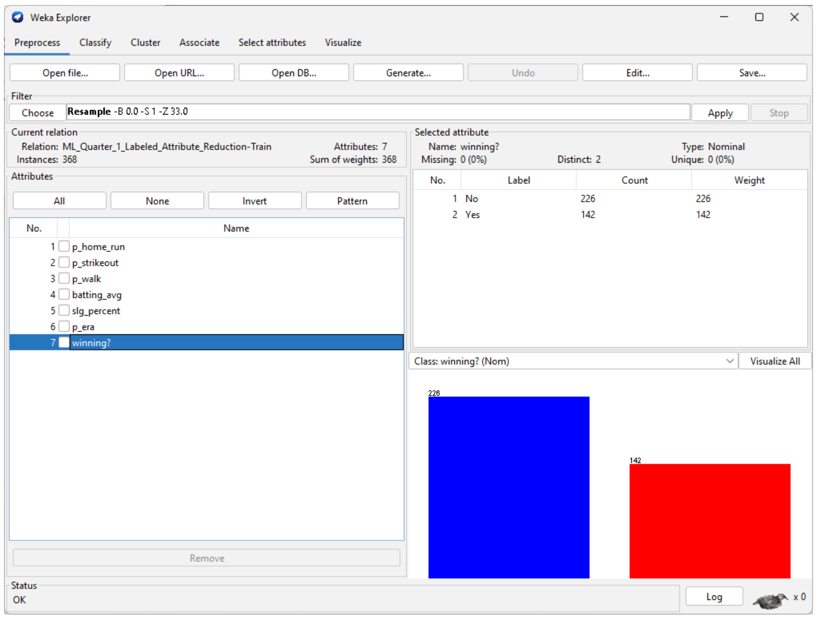
Attributes

No.	Name
1	p_home_run
2	p_strikeout
3	p_walk
4	batting_avg
5	slg_percent
6	p_era
7	winning?

Class: winning? (Nom)

Status
OK

Figure 7: WrapperSubsetEval Test Set



Filter
Resample -B 0.0 -S 1 -Z 33.0

Current relation
Relation: ML_Quarter_1_Labeled_Attribute_Reduction-Train   Attributes: 7
Instances: 368   Sum of weights: 368

Selected attribute
Name: winning?   Type: Nominal
Missing: 0 (0%)   Distinct: 2   Unique: 0 (0%)

No.	Label	Count	Weight
1	No	226	226
2	Yes	142	142

Attributes

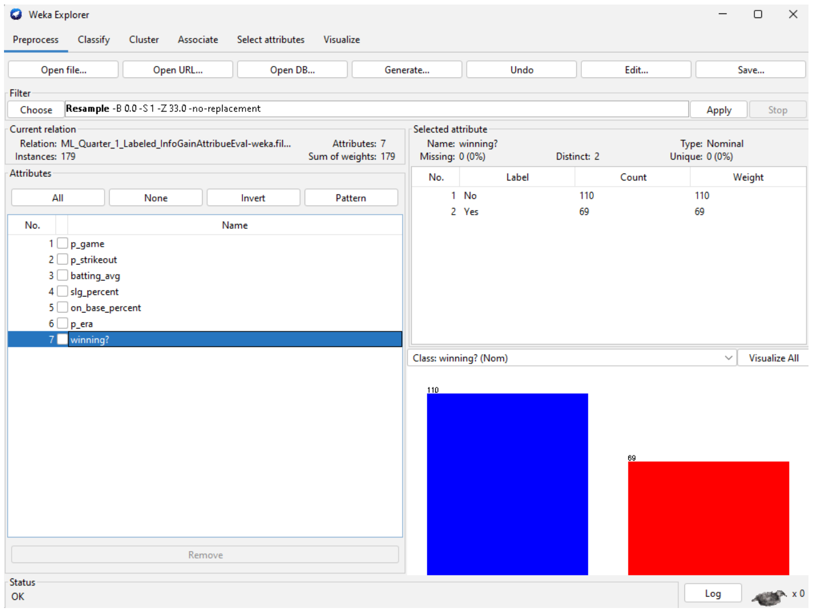
No.	Name
1	p_home_run
2	p_strikeout
3	p_walk
4	batting_avg
5	slg_percent
6	p_era
7	winning?

Class: winning? (Nom)

Status
OK

Figure 8: WrapperSubsetEval Train Set

3.3 InfoGainAttributeEval



Filter
Resample -B 0.0 -S 1 -Z 33.0 -no-replacement

Current relation
Relation: ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.fil... Attributes: 7
Instances: 179 Sum of weights: 179

Selected attribute
Name: winning? Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	No	110	110
2	Yes	69	69

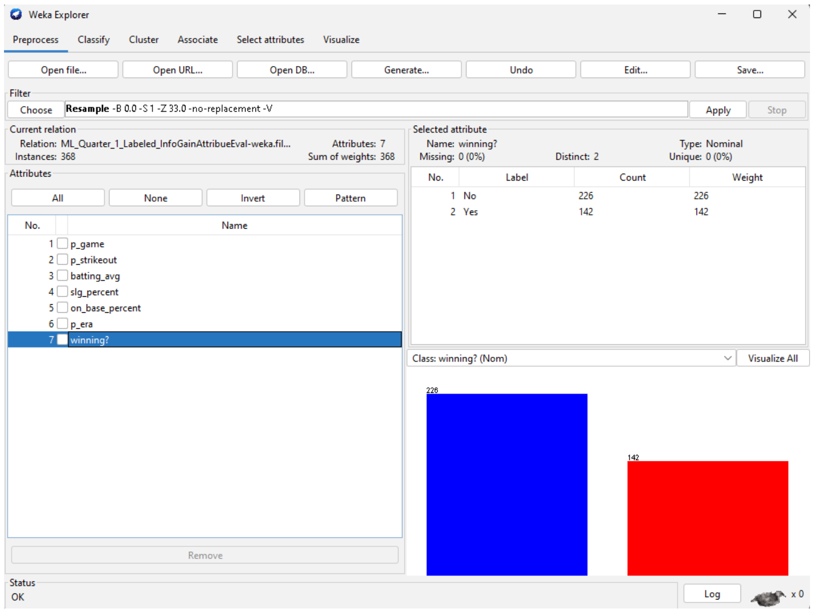
Attributes

No.	Name
1	p_game
2	p_strikeout
3	batting_avg
4	slg_percent
5	on_base_percent
6	p_era
7	winning?

Class: winning? (Nom)

Status
OK

Figure 9: InfoGainAttributeEval Test Set



Filter
Resample -B 0.0 -S 1 -Z 33.0 -no-replacement -V

Current relation
Relation: ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.fil... Attributes: 7
Instances: 368 Sum of weights: 368

Selected attribute
Name: winning? Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	No	226	226
2	Yes	142	142

Attributes

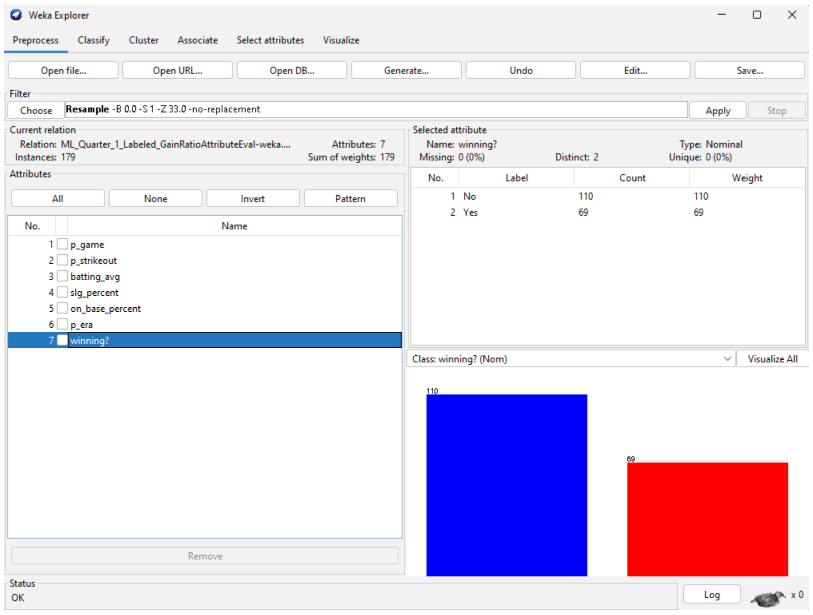
No.	Name
1	p_game
2	p_strikeout
3	batting_avg
4	slg_percent
5	on_base_percent
6	p_era
7	winning?

Class: winning? (Nom)

Status
OK

Figure 10: InfoGainAttributeEval Train Set

3.4 GainRatioAttributeEval



Filter: **Resample** -B 0.0 -S 1 -Z 33.0 -no-replacement

Current relation
Relation: ML_Quarter_1_Labeled_GainRatioAttributeEval-weka.... Attributes: 7
Instances: 179 Sum of weights: 179

Selected attribute
Name: winning? Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	No	110	110
2	Yes	69	69

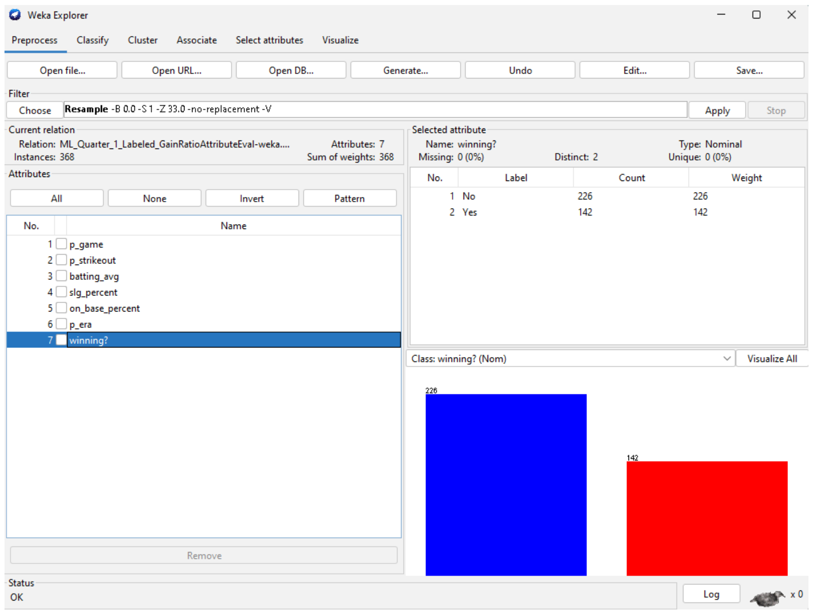
Attributes

No.	Name
1	p_game
2	p_strikeout
3	batting_avg
4	slg_percent
5	on_base_percent
6	p_era
7	winning?

Class: winning? (Nom)

Status
OK

Figure 11: GainRatioAttributeEval Test Set



Filter: **Resample** -B 0.0 -S 1 -Z 33.0 -no-replacement -V

Current relation
Relation: ML_Quarter_1_Labeled_GainRatioAttributeEval-weka.... Attributes: 7
Instances: 368 Sum of weights: 368

Selected attribute
Name: winning? Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	No	226	226
2	Yes	142	142

Attributes

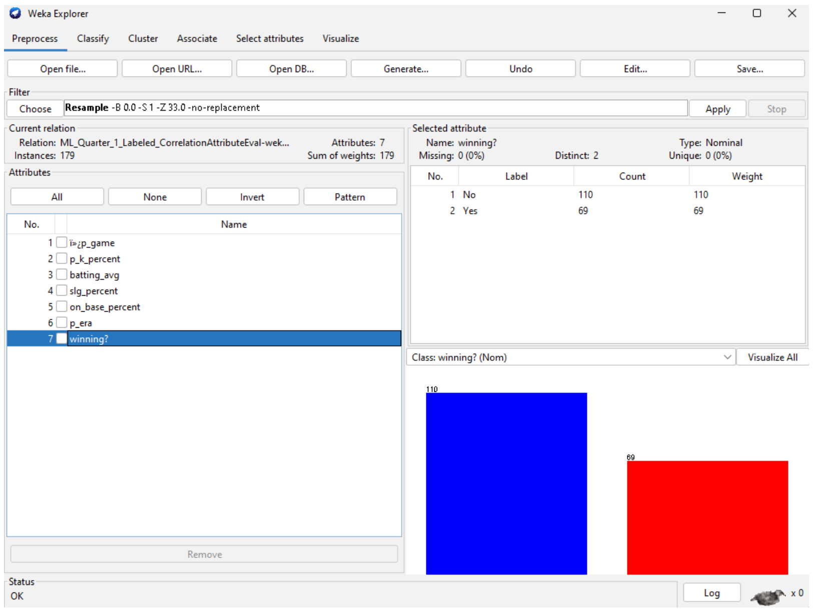
No.	Name
1	p_game
2	p_strikeout
3	batting_avg
4	slg_percent
5	on_base_percent
6	p_era
7	winning?

Class: winning? (Nom)

Status
OK

Figure 12: GainRatioAttributeEval Train Set

3.5 CorrelationAttributeEval



Filter

Resample -B 0.0 -S 1 -Z 33.0 -no-replacement

Current relation

Relation: ML_Quarter_1_Labeled_CorrelationAttributeEval-wek... Attributes: 7

Instances: 179 Sum of weights: 179

Attributes

No.	Name
1	ï»¿p_game
2	p_k_percent
3	batting_avg
4	slg_percent
5	on_base_percent
6	p_era
7	winning?

Selected attribute

Name: winning? Type: Nominal

Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

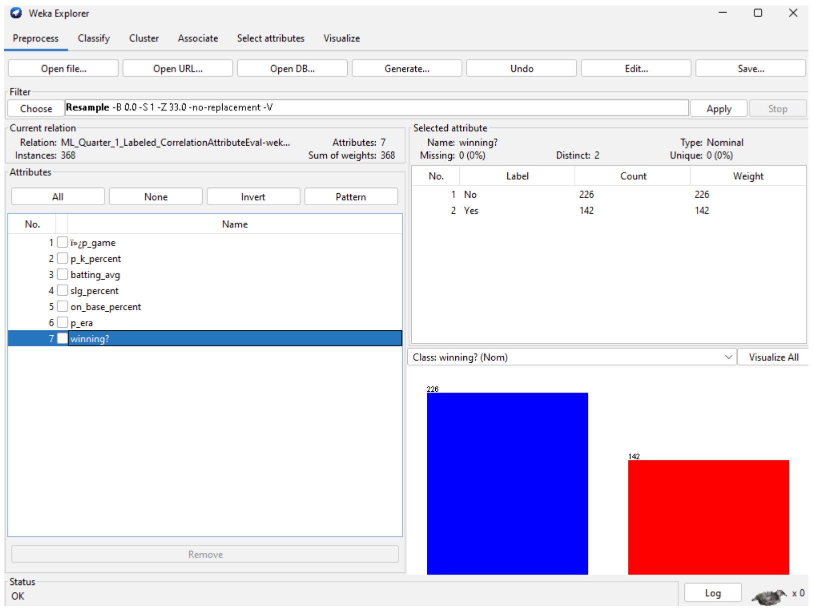
No.	Label	Count	Weight
1	No	110	110
2	Yes	69	69

Class: winning? (Nom)

Status

OK

Figure 13: CorrelationAttributeEval Test Set



Filter

Resample -B 0.0 -S 1 -Z 33.0 -no-replacement -V

Current relation

Relation: ML_Quarter_1_Labeled_CorrelationAttributeEval-wek... Attributes: 7

Instances: 368 Sum of weights: 368

Attributes

No.	Name
1	ï»¿p_game
2	p_k_percent
3	batting_avg
4	slg_percent
5	on_base_percent
6	p_era
7	winning?

Selected attribute

Name: winning? Type: Nominal

Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	No	226	226
2	Yes	142	142

Class: winning? (Nom)

Status

OK

Figure 14: CorrelationAttributeEval Train Set

4 Classification

For each method of attribute reduction, we ran 4 classification methods using WEKA. These methods included:

- **J48**
- **LogisticFunction**
- **NaiveBayes**
- **OneR**

We compared each of these methods to determine which one worked the best for each attribute reduction method. They were trained using supplied training sets we created earlier. We then tested each classification method on each of the testing sets that we created, running 20 classifications in total.

4.1 Types of Classification Methods

4.1.1 J48

J48 classification is WEKA's implementation of the C4.5 algorithm, where the J stands for Java. It builds a decision tree based on information gain. At each node of the decision tree, it splits the attributes into subsets, with the split being decided by the difference in entropy. **J48** is seen as an improvement over the C4.5 algorithm because it accounts for missing values, allows for pruning, continuous value ranges, and many other extra features.

4.1.2 LogisticFunction

LogisticFunction is most commonly used when the class variable is binary, which, in our case, it is. It uses the sigmoid function,

$$S(x) = \frac{1}{1+e^{-x}}.$$

Initial weights are modified using the error, and the coefficients are improved through each instance in the test set. We then can use these coefficients to make predictions. A decision boundary is set, like $y = 0.5$, and everything above the decision boundary is rounded to a decision of 1 and everything below the decision boundary is rounded to a decision of 0. These decisions are mapped to the class variable.

4.1.3 NaiveBayes

NaiveBayes is based on Bayes'theorem, or

$$\mathbf{P}(A|B) = \frac{\mathbf{P}(B|A)\mathbf{P}(A)}{\mathbf{P}(B)}.$$

It assumes that each attribute is independent of the rest, making the train time much faster. This makes it a good baseline to compare other classification methods to. It repeatedly applies Bayes'theorem with these naive assumptions.

4.1.4 OneR

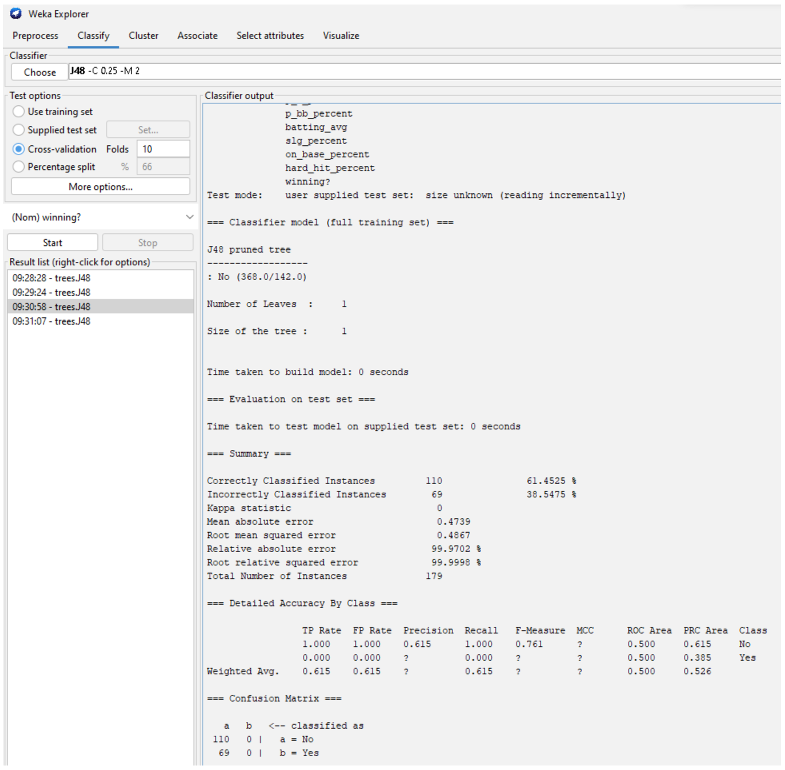
OneR, also known as **One Class** classification, looks at just one attribute to make a prediction about the instance. It uses the training set to determine which attribute is the best predictor of the class variable, determined with accuracy. It is an extremely lightweight classification algorithm, sacrificing overall accuracy for speed. Like **NaiveBayes**, it can provide a baseline for other predictors.

4.2 Methods

We ran each of the 4 classification methods in WEKA on each of our resulting datasets from our 5 attribute selection processes, giving us 20 classifications run total. We then analyzed the results from each attribute selection method individually, then together as a large group.

4.3 SelfSelected

4.3.1 J48 - SelfSelected



Classifier: J48 -C 0.25 -M 2

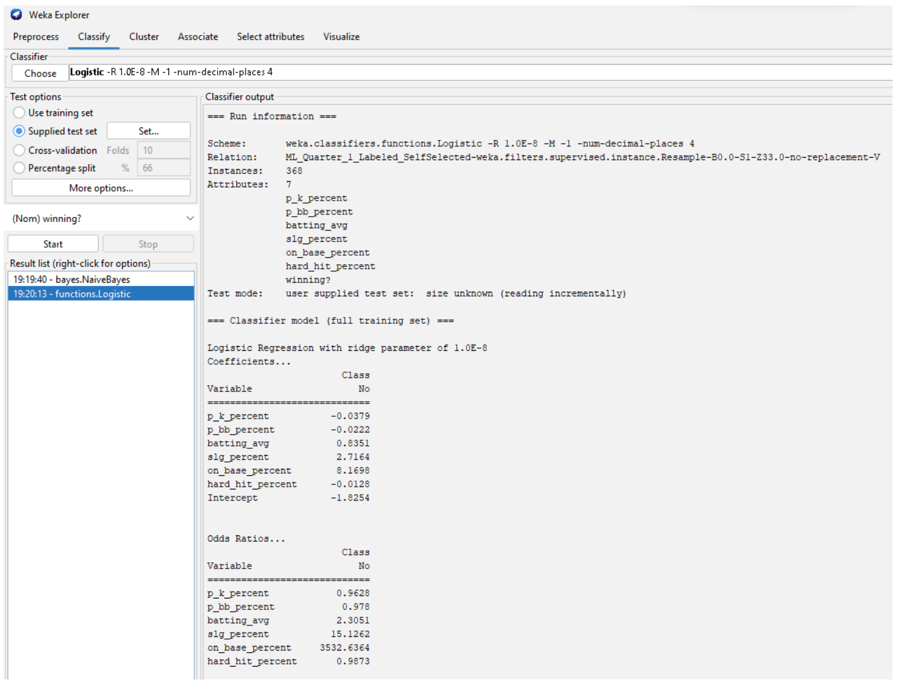
Test options: Cross-validation, Folds 10; Percentage split % 66

(Nom) winning?

```
              p_bb_percent
              batting_avg
              slg_percent
              on_base_percent
              hard_hit_percent
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

J48 pruned tree
------------------
: No (368.0/142.0)

Number of Leaves  :     1

Size of the tree :      1


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         110               61.4525 %
Incorrectly Classified Instances        69               38.5475 %
Kappa statistic                          0
Mean absolute error                      0.4739
Root mean squared error                  0.4867
Relative absolute error                 99.9702 %
Root relative squared error             99.9998 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.615      1.000    0.761      ?        0.500     0.615     No
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.385     Yes
Weighted Avg.    0.615    0.615    ?          0.615    ?          ?        0.500     0.526

=== Confusion Matrix ===

   a   b   <-- classified as
 110   0 |   a = No
  69   0 |   b = Yes
```

Figure 15: WEKA J48 Classification Results on SelfSelected Attributes

4.3.2 LogisticFunction - SelfSelected

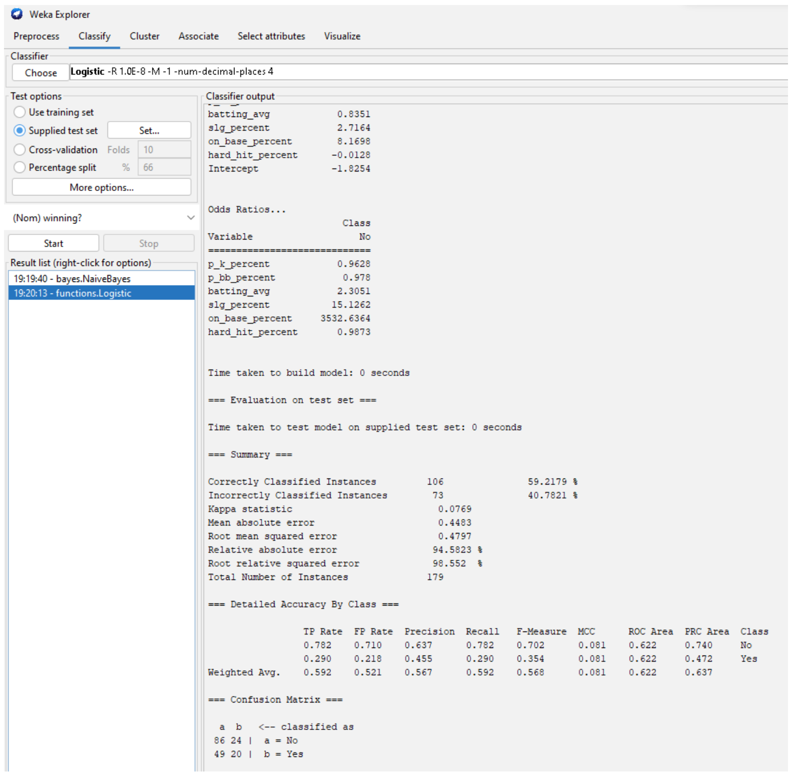


Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

19:19:40 - bayes.NaiveBayes
19:20:13 - functions.Logistic

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     ML_Quarter_1_Labeled_SelfSelected-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_k_percent
              p_bb_percent
              batting_avg
              slg_percent
              on_base_percent
              hard_hit_percent
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                        Class
Variable                   No
=============================
p_k_percent           -0.0379
p_bb_percent          -0.0222
batting_avg            0.8351
slg_percent            2.7164
on_base_percent        8.1698
hard_hit_percent      -0.0128
Intercept             -1.8254


Odds Ratios...
                        Class
Variable                   No
=============================
p_k_percent            0.9628
p_bb_percent            0.978
batting_avg            2.3051
slg_percent           15.1262
on_base_percent     3532.6364
hard_hit_percent       0.9873
```

Figure 16: WEKA LogisticFunction Model on SelfSelected Attributes



Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Supplied test set [✓]; Folds 10; % 66

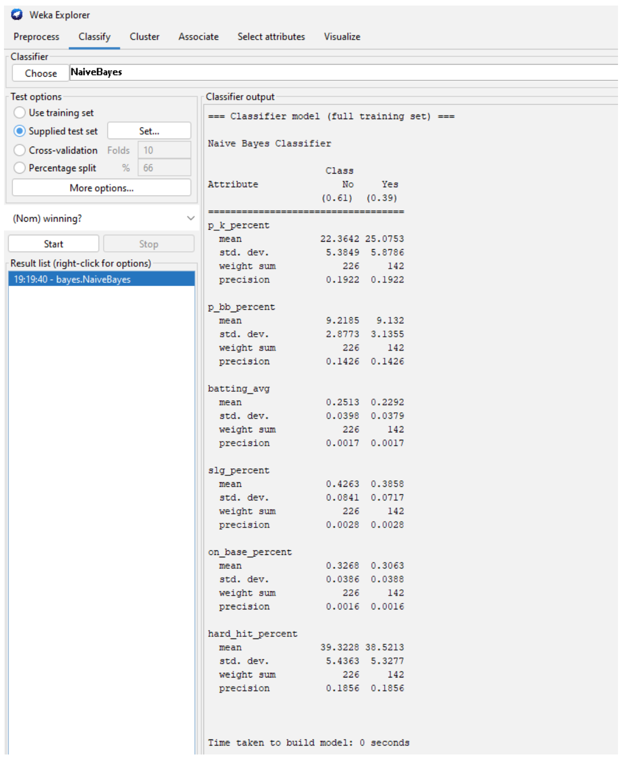
(Nom) winning?

19:19:40 - bayes.NaiveBayes
19:20:13 - functions.Logistic

```
batting_avg          0.8351
slg_percent          2.7164
on_base_percent      8.1698
hard_hit_percent    -0.0128
Intercept           -1.8254


Odds Ratios...
                      Class
Variable                 No
=============================
p_k_percent          0.9628
p_bb_percent          0.978
batting_avg          2.3051
slg_percent         15.1262
on_base_percent   3532.6364
hard_hit_percent     0.9873


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         106               59.2179 %
Incorrectly Classified Instances        73               40.7821 %
Kappa statistic                          0.0769
Mean absolute error                      0.4483
Root mean squared error                  0.4797
Relative absolute error                 94.5823 %
Root relative squared error             98.552  %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.782    0.710    0.637      0.782    0.702      0.081    0.622     0.740     No
                 0.290    0.218    0.455      0.290    0.354      0.081    0.622     0.472     Yes
Weighted Avg.    0.592    0.521    0.567      0.592    0.568      0.081    0.622     0.637

=== Confusion Matrix ===

  a  b   <-- classified as
 86 24 |  a = No
 49 20 |  b = Yes
```

Figure 17: WEKA LogisticFunction Summary Results on SelfSelected Attributes

4.3.3 NaiveBayes - SelfSelected



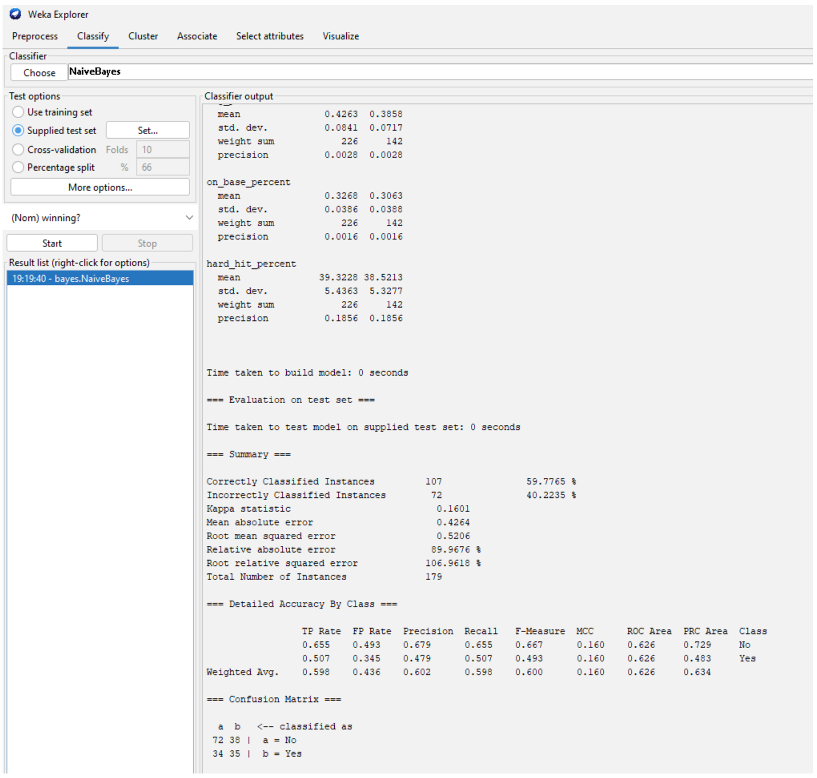
Classifier: NaiveBayes

Supplied test set [✓]

(Nom) winning?

```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                          Class
Attribute                    No      Yes
                         (0.61)   (0.39)
===================================
p_k_percent
  mean                  22.3642 25.0753
  std. dev.              5.3849  5.8786
  weight sum                226      142
  precision              0.1922  0.1922

p_bb_percent
  mean                   9.2185    9.132
  std. dev.              2.8773  3.1355
  weight sum                226      142
  precision              0.1426  0.1426

batting_avg
  mean                   0.2513  0.2292
  std. dev.              0.0398  0.0379
  weight sum                226      142
  precision              0.0017  0.0017

slg_percent
  mean                   0.4263  0.3858
  std. dev.              0.0841  0.0717
  weight sum                226      142
  precision              0.0028  0.0028

on_base_percent
  mean                   0.3268  0.3063
  std. dev.              0.0386  0.0388
  weight sum                226      142
  precision              0.0016  0.0016

hard_hit_percent
  mean                  39.3228 38.5213
  std. dev.              5.4363  5.3277
  weight sum                226      142
  precision              0.1856  0.1856



Time taken to build model: 0 seconds
```

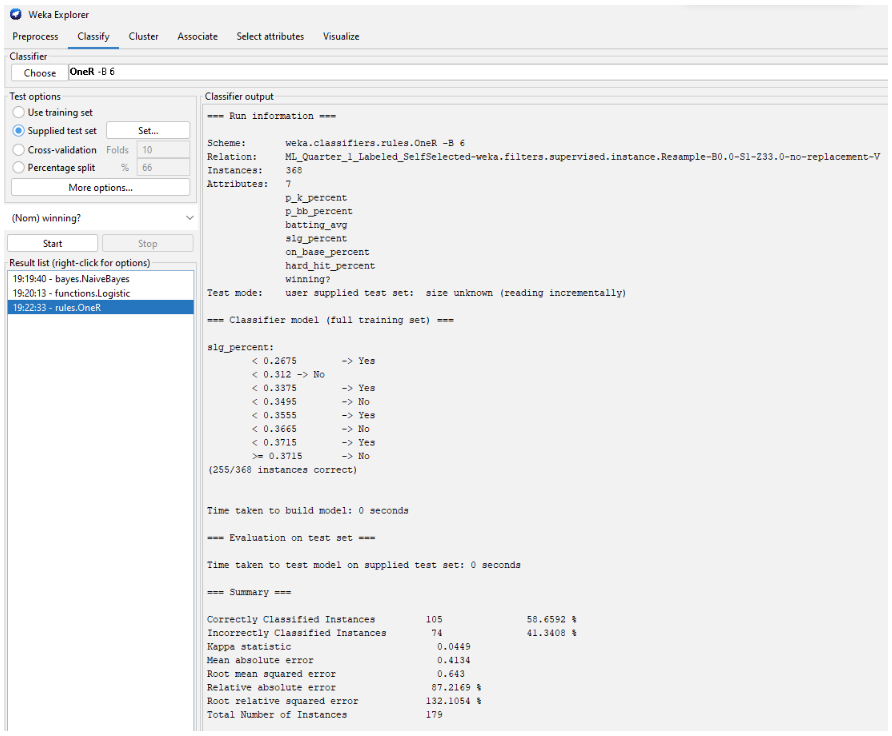
Figure 18: WEKA NaiveBayes Model on SelfSelected Attributes



```
  mean              0.4263  0.3858
  std. dev.         0.0841  0.0717
  weight sum           226     142
  precision         0.0028  0.0028

on_base_percent
  mean              0.3268  0.3063
  std. dev.         0.0386  0.0388
  weight sum           226     142
  precision         0.0016  0.0016

hard_hit_percent
  mean             39.3228 38.5213
  std. dev.         5.4363  5.3277
  weight sum           226     142
  precision         0.1856  0.1856



Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         107               59.7765 %
Incorrectly Classified Instances        72               40.2235 %
Kappa statistic                          0.1601
Mean absolute error                      0.4264
Root mean squared error                  0.5206
Relative absolute error                 89.9676 %
Root relative squared error            106.9618 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.655    0.493    0.679      0.655    0.667      0.160    0.626     0.729     No
                 0.507    0.345    0.479      0.507    0.493      0.160    0.626     0.483     Yes
Weighted Avg.    0.598    0.436    0.602      0.598    0.600      0.160    0.626     0.634

=== Confusion Matrix ===

  a  b   <-- classified as
 72 38 |  a = No
 34 35 |  b = Yes
```

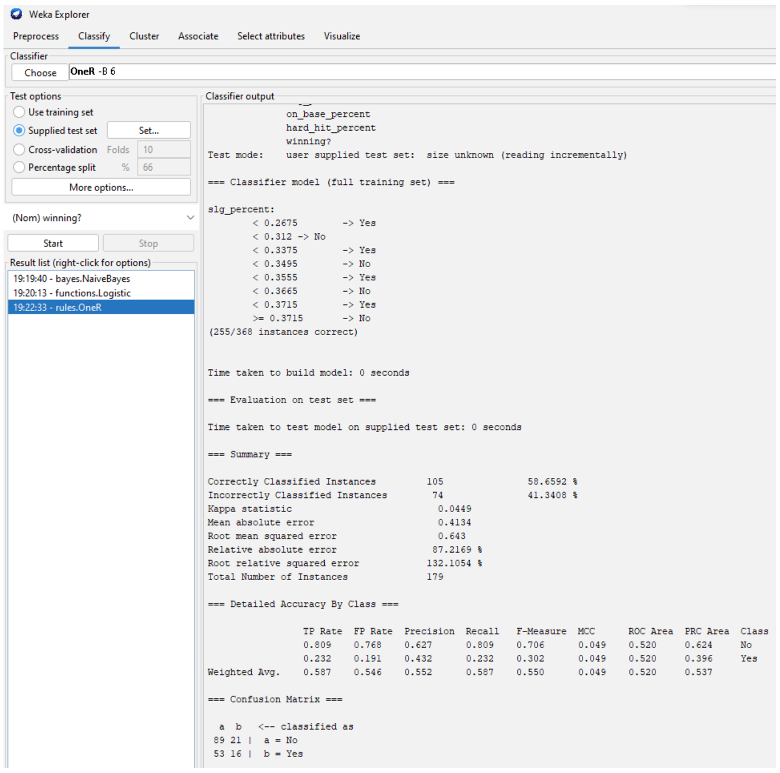
Figure 19: WEKA NaiveBayes Summary Results on SelfSelected Attributes

4.3.4 OneR - SelfSelected



```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     ML_Quarter_1_Labeled_SelfSelected-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_k_percent
              p_bb_percent
              batting_avg
              slg_percent
              on_base_percent
              hard_hit_percent
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

slg_percent:
	< 0.2675	-> Yes
	< 0.312 -> No
	< 0.3375	-> Yes
	< 0.3495	-> No
	< 0.3555	-> Yes
	< 0.3665	-> No
	< 0.3715	-> Yes
	>= 0.3715	-> No
(255/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         105               58.6592 %
Incorrectly Classified Instances        74               41.3408 %
Kappa statistic                          0.0449
Mean absolute error                      0.4134
Root mean squared error                  0.643
Relative absolute error                 87.2169 %
Root relative squared error            132.1054 %
Total Number of Instances              179
```

Figure 20: WEKA OneR Model on SelfSelected Attributes



```
              on_base_percent
              hard_hit_percent
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

slg_percent:
        < 0.2675        -> Yes
        < 0.312 -> No
        < 0.3375        -> Yes
        < 0.3495        -> No
        < 0.3555        -> Yes
        < 0.3665        -> No
        < 0.3715        -> Yes
        >= 0.3715       -> No
(255/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         105               58.6592 %
Incorrectly Classified Instances        74               41.3408 %
Kappa statistic                          0.0449
Mean absolute error                      0.4134
Root mean squared error                  0.643
Relative absolute error                 87.2169 %
Root relative squared error            132.1054 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.809    0.768    0.627      0.809    0.706      0.049    0.520     0.624     No
                 0.232    0.191    0.432      0.232    0.302      0.049    0.520     0.396     Yes
Weighted Avg.    0.587    0.546    0.552      0.587    0.550      0.049    0.520     0.537

=== Confusion Matrix ===

  a  b   <-- classified as
 89 21 |  a = No
 53 16 |  b = Yes
```

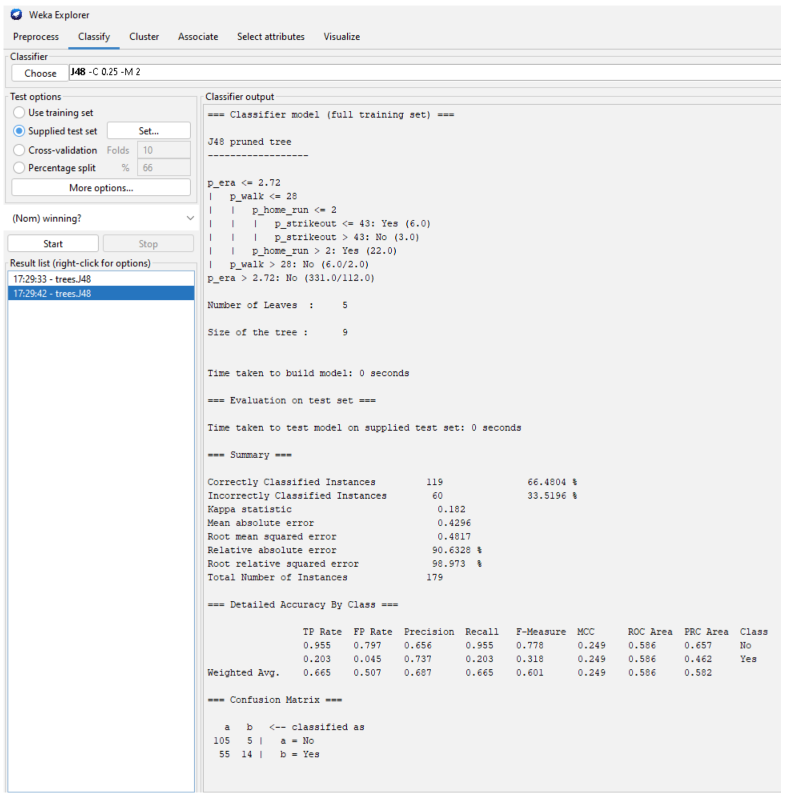
Figure 21: WEKA OneR Summary Results on SelfSelected Attributes

4.3.5 Analysis

We can see that overall **J48** performed with the highest accuracy. However, one thing to note is that there are no true negatives for **J48** classification.

4.4 WrapperSubsetEval

4.4.1 J48 - WrapperSubsetEval

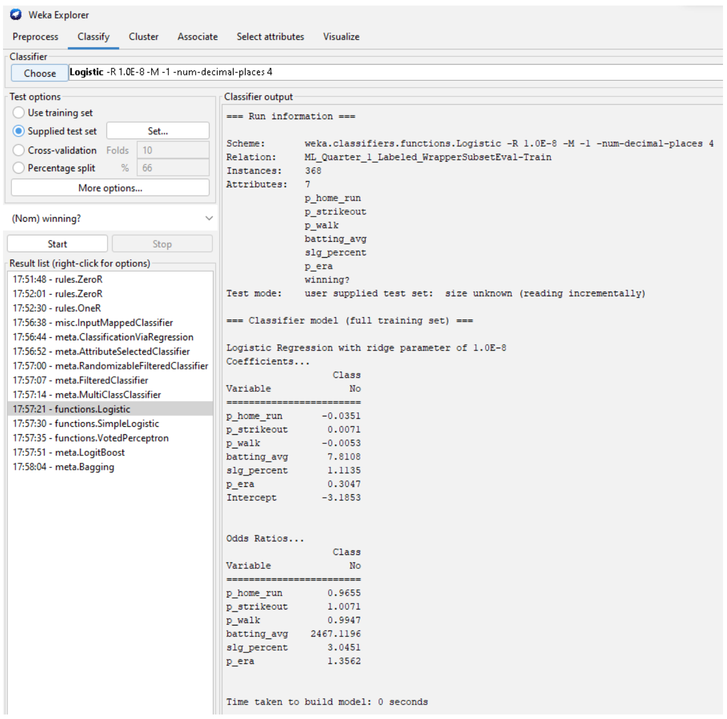


J48 -C 0.25 -M 2

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

p_era <= 2.72
|   p_walk <= 28
|   |   p_home_run <= 2
|   |   |   p_strikeout <= 43: Yes (6.0)
|   |   |   p_strikeout > 43: No (3.0)
|   |   p_home_run > 2: Yes (22.0)
|   p_walk > 28: No (6.0/2.0)
p_era > 2.72: No (331.0/112.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         119               66.4804 %
Incorrectly Classified Instances        60               33.5196 %
Kappa statistic                          0.182
Mean absolute error                      0.4296
Root mean squared error                  0.4817
Relative absolute error                 90.6328 %
Root relative squared error             98.973  %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.955    0.797    0.656      0.955    0.778      0.249    0.586     0.657     No
                 0.203    0.045    0.737      0.203    0.318      0.249    0.586     0.462     Yes
Weighted Avg.    0.665    0.507    0.687      0.665    0.601      0.249    0.586     0.582

=== Confusion Matrix ===

   a   b   <-- classified as
 105   5 |   a = No
  55  14 |   b = Yes
```

Figure 22: WEKA J48 Classification Results on WrapperSubsetEval Attributes

4.4.2 LogisticFunction - WrapperSubsetEval



Classifier: Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

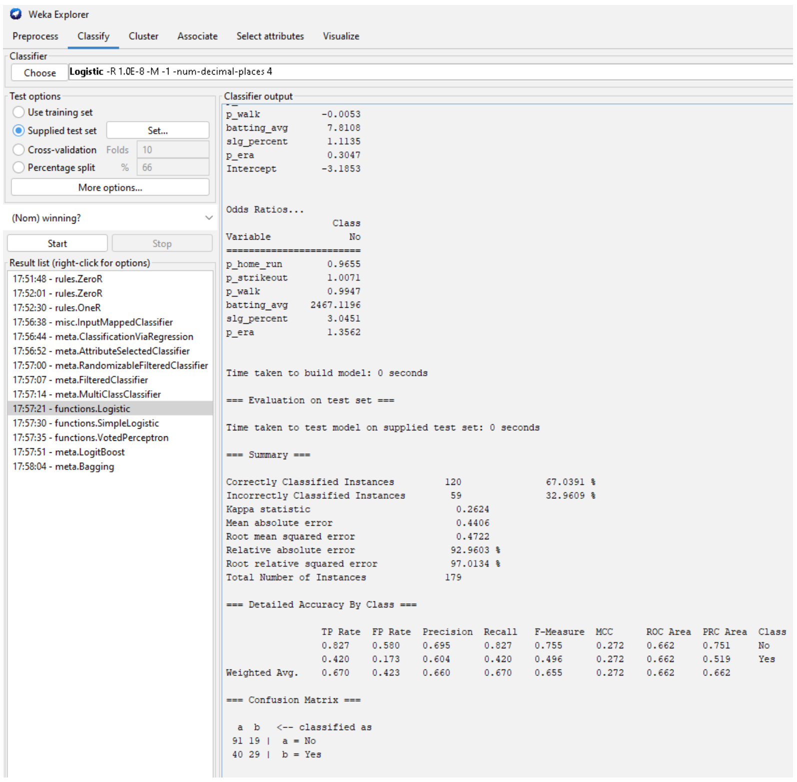
Test options: ( ) Use training set [✓ Supplied test set] ( ) Cross-validation Folds 10 ( ) Percentage split % 66

(Nom) winning?

Result list (right-click for options)
- 17:51:48 - rules.ZeroR
- 17:52:01 - rules.ZeroR
- 17:52:30 - rules.OneR
- 17:56:38 - misc.InputMappedClassifier
- 17:56:44 - meta.ClassificationViaRegression
- 17:56:52 - meta.AttributeSelectedClassifier
- 17:57:00 - meta.RandomizableFilteredClassifier
- 17:57:07 - meta.FilteredClassifier
- 17:57:14 - meta.MultiClassClassifier
- 17:57:21 - functions.Logistic
- 17:57:30 - functions.SimpleLogistic
- 17:57:35 - functions.VotedPerceptron
- 17:57:51 - meta.LogitBoost
- 17:58:04 - meta.Bagging

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     ML_Quarter_1_Labeled_WrapperSubsetEval-Train
Instances:    368
Attributes:   7
              p_home_run
              p_strikeout
              p_walk
              batting_avg
              slg_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                       Class
Variable                  No
========================
p_home_run       -0.0351
p_strikeout       0.0071
p_walk           -0.0053
batting_avg       7.8108
slg_percent       1.1135
p_era             0.3047
Intercept        -3.1853


Odds Ratios...
                       Class
Variable                  No
========================
p_home_run        0.9655
p_strikeout       1.0071
p_walk            0.9947
batting_avg    2467.1196
slg_percent       3.0451
p_era             1.3562


Time taken to build model: 0 seconds
```

Figure 23: WEKA LogisticFunction Model on WrapperSubsetEval Attributes



Classifier: Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options: Supplied test set

(Nom) winning?

Result list (right-click for options)

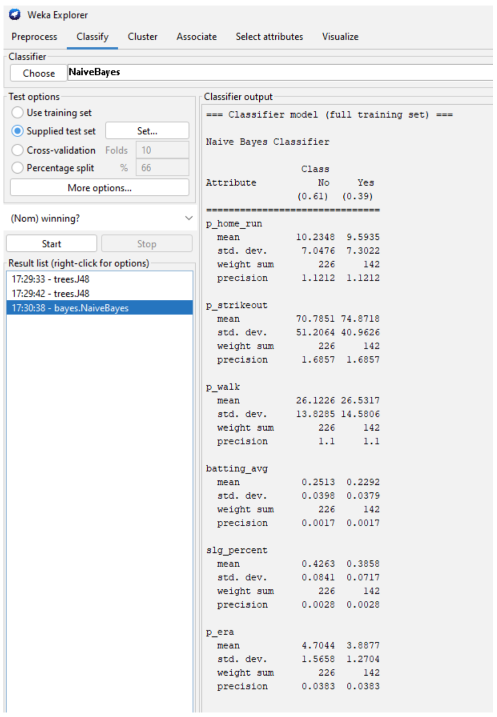
```
17:51:48 - rules.ZeroR
17:52:01 - rules.ZeroR
17:52:30 - rules.OneR
17:56:38 - misc.InputMappedClassifier
17:56:44 - meta.ClassificationViaRegression
17:56:52 - meta.AttributeSelectedClassifier
17:57:00 - meta.RandomizableFilteredClassifier
17:57:07 - meta.FilteredClassifier
17:57:14 - meta.MultiClassClassifier
17:57:21 - functions.Logistic
17:57:30 - functions.SimpleLogistic
17:57:35 - functions.VotedPerceptron
17:57:51 - meta.LogitBoost
17:58:04 - meta.Bagging
```

Classifier output

```
p_walk          -0.0053
batting_avg      7.8108
slg_percent      1.1135
p_era            0.3047
Intercept       -3.1853


Odds Ratios...
                  Class
Variable             No
========================
p_home_run       0.9655
p_strikeout      1.0071
p_walk           0.9947
batting_avg   2467.1196
slg_percent      3.0451
p_era            1.3562


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         120               67.0391 %
Incorrectly Classified Instances        59               32.9609 %
Kappa statistic                          0.2624
Mean absolute error                      0.4406
Root mean squared error                  0.4722
Relative absolute error                 92.9603 %
Root relative squared error             97.0134 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.827    0.580    0.695      0.827    0.755      0.272    0.662     0.751     No
                 0.420    0.173    0.604      0.420    0.496      0.272    0.662     0.519     Yes
Weighted Avg.    0.670    0.423    0.660      0.670    0.655      0.272    0.662     0.662

=== Confusion Matrix ===

  a  b   <-- classified as
 91 19 |  a = No
 40 29 |  b = Yes
```

Figure 24: WEKA LogisticFunction Summary Results on WrapperSubsetEval Attributes

4.4.3 NaiveBayes - WrapperSubsetEval



Classifier: NaiveBayes

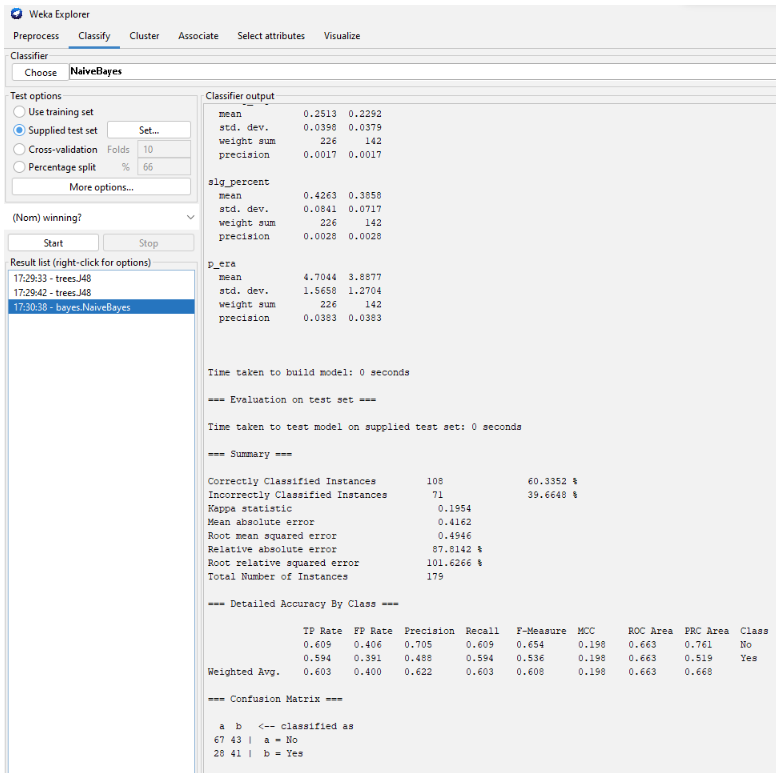
Test options: Supplied test set; Folds 10; % 66

(Nom) winning?

17:29:33 - trees.J48
17:29:42 - trees.J48
17:30:38 - bayes.NaiveBayes

```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                  Class
Attribute            No     Yes
                 (0.61)  (0.39)
================================
p_home_run
  mean           10.2348  9.5935
  std. dev.       7.0476  7.3022
  weight sum         226     142
  precision       1.1212  1.1212

p_strikeout
  mean           70.7851 74.8718
  std. dev.      51.2064 40.9626
  weight sum         226     142
  precision       1.6857  1.6857

p_walk
  mean           26.1226 26.5317
  std. dev.      13.8285 14.5806
  weight sum         226     142
  precision          1.1     1.1

batting_avg
  mean            0.2513  0.2292
  std. dev.       0.0398  0.0379
  weight sum         226     142
  precision       0.0017  0.0017

slg_percent
  mean            0.4263  0.3858
  std. dev.       0.0841  0.0717
  weight sum         226     142
  precision       0.0028  0.0028

p_era
  mean            4.7044  3.8877
  std. dev.       1.5658  1.2704
  weight sum         226     142
  precision       0.0383  0.0383
```

Figure 25: WEKA NaiveBayes Model on WrapperSubsetEval Attributes



Classifier: NaiveBayes

Test options: Use training set [✓ Supplied test set] Cross-validation (Folds 10) Percentage split (% 66)

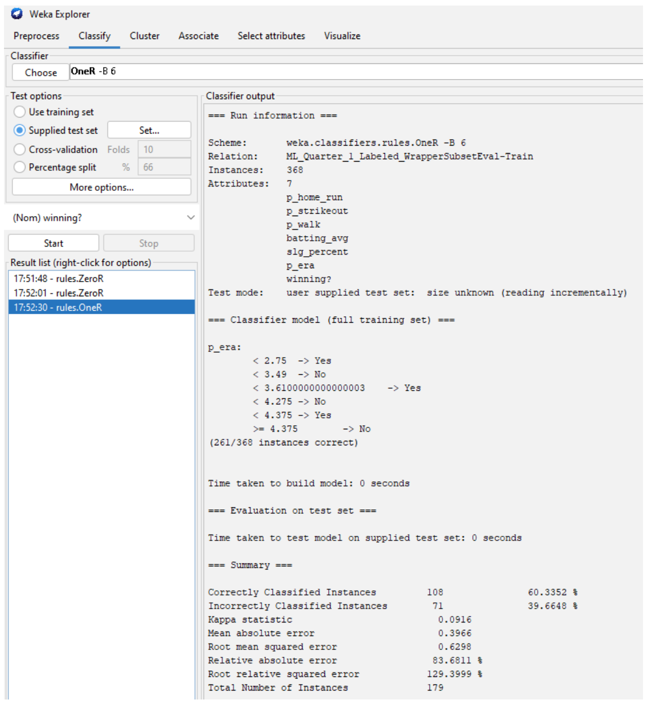
(Nom) winning?

17:29:33 - trees.J48
17:29:42 - trees.J48
17:30:38 - bayes.NaiveBayes

```
  mean           0.2513  0.2292
  std. dev.      0.0398  0.0379
  weight sum        226     142
  precision      0.0017  0.0017

slg_percent
  mean           0.4263  0.3858
  std. dev.      0.0841  0.0717
  weight sum        226     142
  precision      0.0028  0.0028

p_era
  mean           4.7044  3.8877
  std. dev.      1.5658  1.2704
  weight sum        226     142
  precision      0.0383  0.0383



Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         108               60.3352 %
Incorrectly Classified Instances        71               39.6648 %
Kappa statistic                          0.1954
Mean absolute error                      0.4162
Root mean squared error                  0.4946
Relative absolute error                 87.8142 %
Root relative squared error            101.6266 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.609    0.406    0.705      0.609    0.654      0.198    0.663     0.761     No
                 0.594    0.391    0.488      0.594    0.536      0.198    0.663     0.519     Yes
Weighted Avg.    0.603    0.400    0.622      0.603    0.608      0.198    0.663     0.668

=== Confusion Matrix ===

  a  b   <-- classified as
 67 43 |  a = No
 28 41 |  b = Yes
```

Figure 26: WEKA NaiveBayes Summary Results on WrapperSubsetEval Attributes

4.4.4 OneR - WrapperSubsetEval



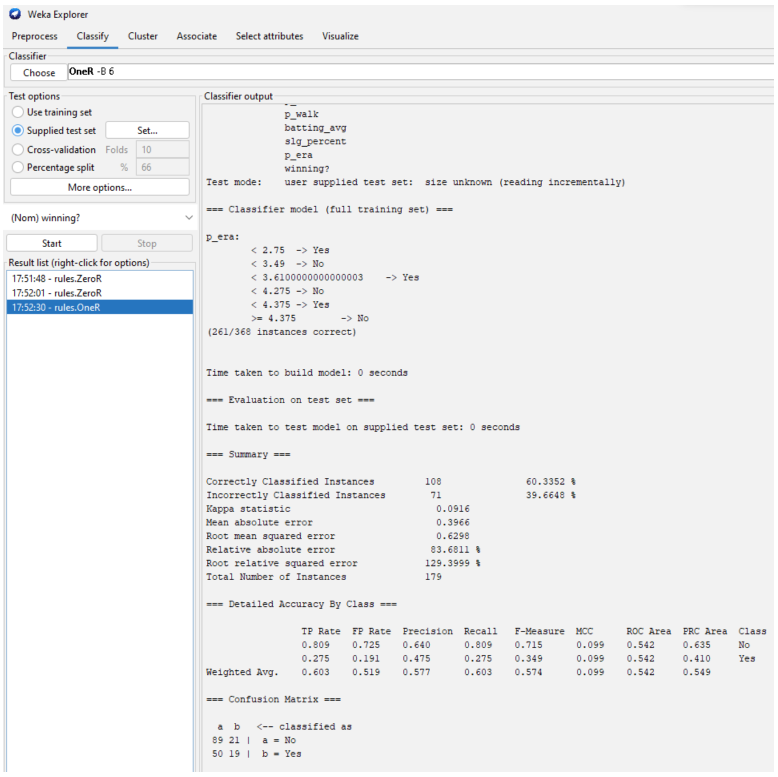
OneR -B 6

(Nom) winning?

17:51:48 - rules.ZeroR
17:52:01 - rules.ZeroR
17:52:30 - rules.OneR

```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     ML_Quarter_1_Labeled_WrapperSubsetEval-Train
Instances:    368
Attributes:   7
              p_home_run
              p_strikeout
              p_walk
              batting_avg
              slg_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
        < 2.75  -> Yes
        < 3.49  -> No
        < 3.6100000000000003    -> Yes
        < 4.275 -> No
        < 4.375 -> Yes
        >= 4.375        -> No
(261/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         108               60.3352 %
Incorrectly Classified Instances        71               39.6648 %
Kappa statistic                          0.0916
Mean absolute error                      0.3966
Root mean squared error                  0.6298
Relative absolute error                 83.6811 %
Root relative squared error            129.3999 %
Total Number of Instances              179
```

Figure 27: WEKA OneR Model on WrapperSubsetEval Attributes



OneR -B 6

( ) Use training set
(●) Supplied test set
( ) Cross-validation Folds 10
( ) Percentage split % 66

(Nom) winning?

17:51:48 - rules.ZeroR
17:52:01 - rules.ZeroR
17:52:30 - rules.OneR

Classifier output

```
            p_walk
            batting_avg
            slg_percent
            p_era
            winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
	< 2.75	-> Yes
	< 3.49	-> No
	< 3.6100000000000003	-> Yes
	< 4.275 -> No
	< 4.375 -> Yes
	>= 4.375	-> No
(261/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         108               60.3352 %
Incorrectly Classified Instances        71               39.6648 %
Kappa statistic                          0.0916
Mean absolute error                      0.3966
Root mean squared error                  0.6298
Relative absolute error                 83.6811 %
Root relative squared error            129.3999 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.809    0.725    0.640      0.809    0.715      0.099    0.542     0.635     No
                 0.275    0.191    0.475      0.275    0.349      0.099    0.542     0.410     Yes
Weighted Avg.    0.603    0.519    0.577      0.603    0.574      0.099    0.542     0.549

=== Confusion Matrix ===

  a  b   <-- classified as
 89 21 |  a = No
 50 19 |  b = Yes
```

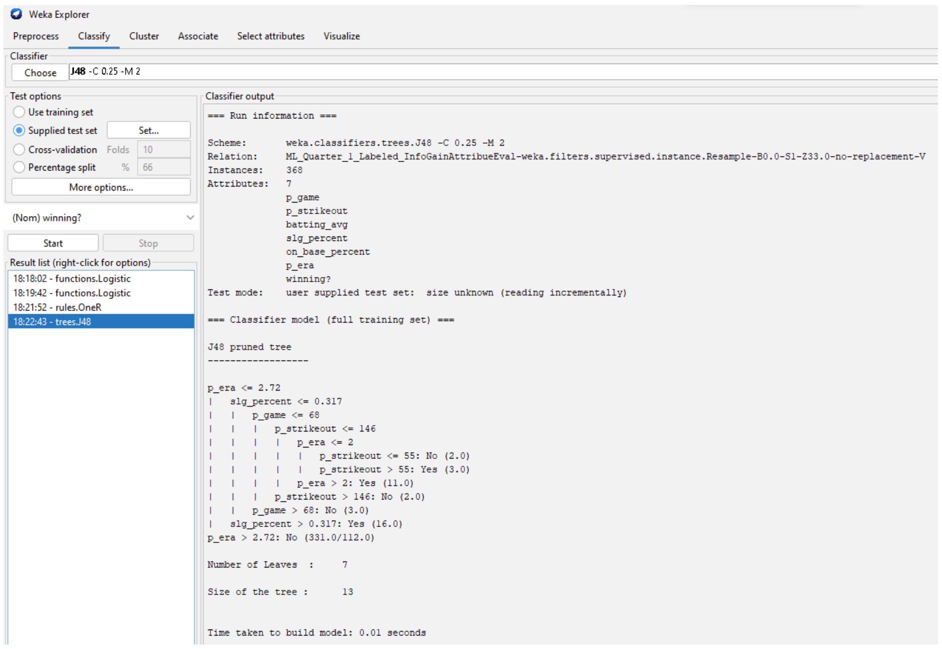
Figure 28: WEKA OneR Summary Results on WrapperSubsetEval Attributes

4.4.5 Analysis

We see that **LogisticFunction** performed best here, but **J48** was a close second with only one more instance misclassified. **J48** was a lot better at determining true positives while **LogisticFunction** was a lot better at determining true negatives.

4.5 InfoGainAttributeEval

4.5.1 J48 - InfoGainAttributeEval



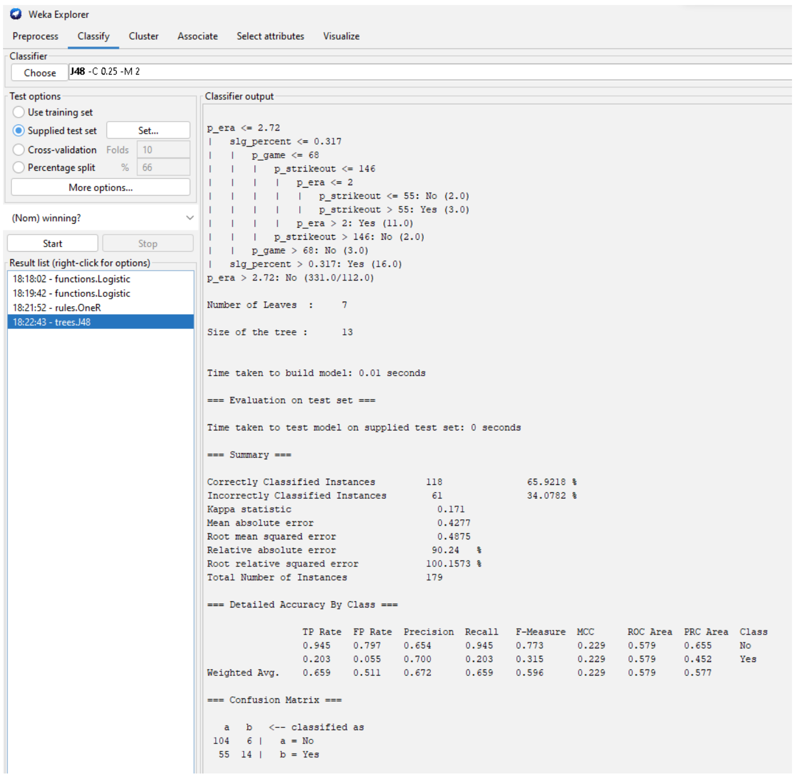
J48 -C 0.25 -M 2

Folds 10, % 66

18:18:02 - functions.Logistic
18:19:42 - functions.Logistic
18:21:52 - rules.OneR
18:22:43 - trees.J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

J48 pruned tree
------------------

p_era <= 2.72
|   slg_percent <= 0.317
|   |   p_game <= 68
|   |   |   p_strikeout <= 146
|   |   |   |   p_era <= 2
|   |   |   |   |   p_strikeout <= 55: No (2.0)
|   |   |   |   |   p_strikeout > 55: Yes (3.0)
|   |   |   |   p_era > 2: Yes (11.0)
|   |   |   p_strikeout > 146: No (2.0)
|   |   p_game > 68: No (3.0)
|   slg_percent > 0.317: Yes (16.0)
p_era > 2.72: No (331.0/112.0)

Number of Leaves  :     7

Size of the tree :      13


Time taken to build model: 0.01 seconds
```

Figure 29: WEKA J48 Model on InfoGainAttributeEval Attributes



Classifier: J48 -C 0.25 -M 2

Test options: Use training set; Supplied test set [✓]; Cross-validation Folds 10; Percentage split % 66

(Nom) winning?

Result list (right-click for options)

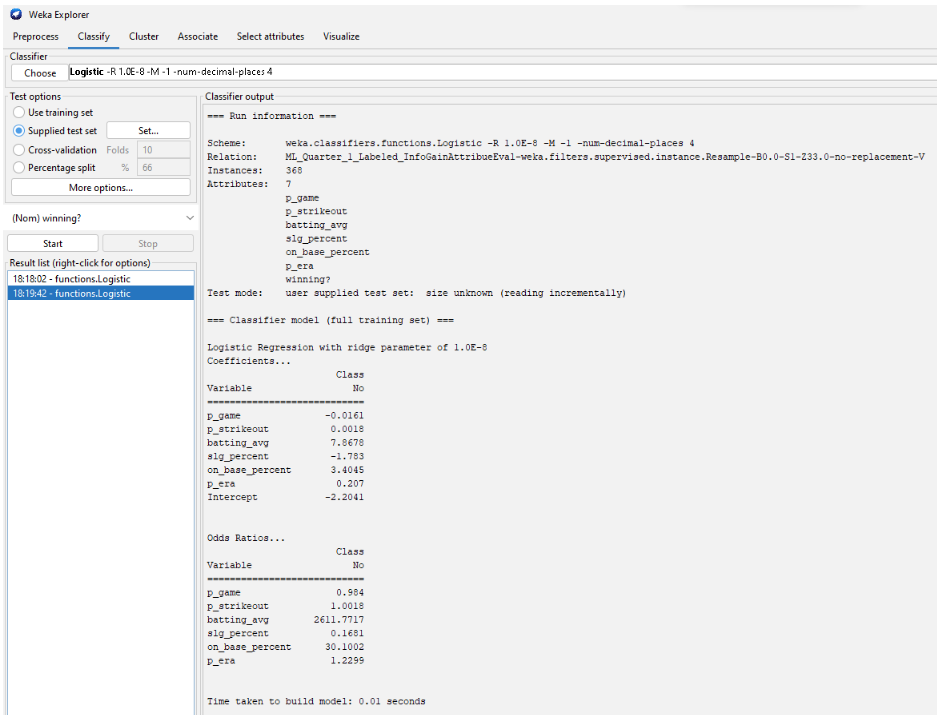
18:18:02 - functions.Logistic
18:19:42 - functions.Logistic
18:21:52 - rules.OneR
18:22:43 - trees.J48

Classifier output

```
p_era <= 2.72
|   slg_percent <= 0.317
|   |   p_game <= 68
|   |   |   p_strikeout <= 146
|   |   |   |   p_era <= 2
|   |   |   |   |   p_strikeout <= 55: No (2.0)
|   |   |   |   |   p_strikeout > 55: Yes (3.0)
|   |   |   |   p_era > 2: Yes (11.0)
|   |   |   p_strikeout > 146: No (2.0)
|   |   p_game > 68: No (3.0)
|   slg_percent > 0.317: Yes (16.0)
p_era > 2.72: No (331.0/112.0)

Number of Leaves  :     7

Size of the tree :      13


Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         118               65.9218 %
Incorrectly Classified Instances        61               34.0782 %
Kappa statistic                          0.171
Mean absolute error                      0.4277
Root mean squared error                  0.4875
Relative absolute error                 90.24   %
Root relative squared error            100.1573 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.945    0.797    0.654      0.945    0.773      0.229    0.579     0.655     No
                 0.203    0.055    0.700      0.203    0.315      0.229    0.579     0.452     Yes
Weighted Avg.    0.659    0.511    0.672      0.659    0.596      0.229    0.579     0.577

=== Confusion Matrix ===

   a   b   <-- classified as
 104   6 |   a = No
  55  14 |   b = Yes
```

Figure 30: WEKA J48 Classification Results on InfoGainAttributeEval Attributes

4.5.2 LogisticFunction - InfoGainAttributeEval



Classifier: Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options: Use training set; [✓ Supplied test set]; Cross-validation Folds 10; Percentage split % 66

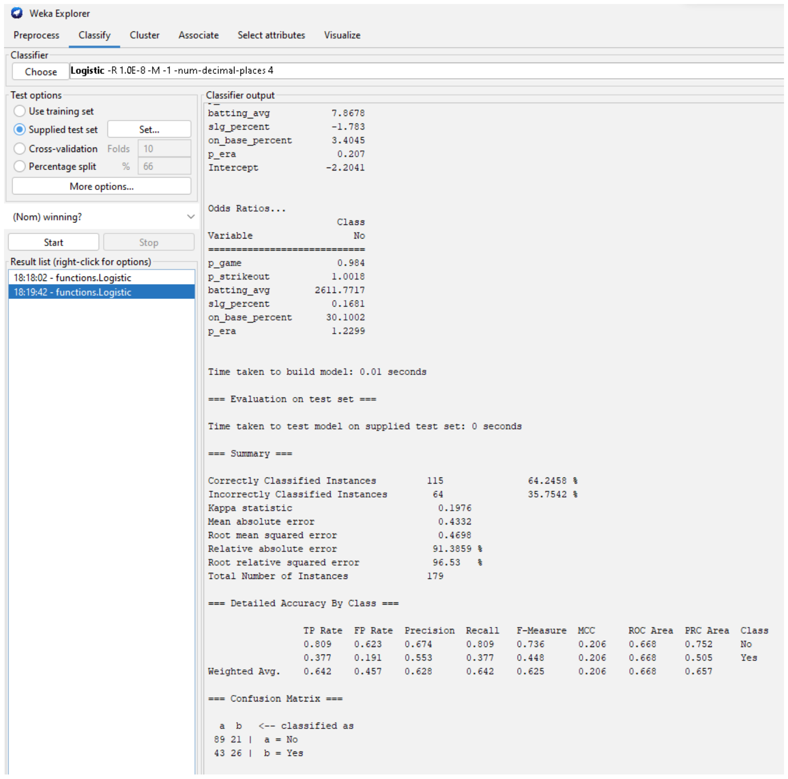
(Nom) winning?

Result list (right-click for options)
18:18:02 - functions.Logistic
18:19:42 - functions.Logistic

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                       Class
Variable                  No
============================
p_game               -0.0161
p_strikeout           0.0018
batting_avg           7.8678
slg_percent           -1.783
on_base_percent       3.4045
p_era                  0.207
Intercept            -2.2041


Odds Ratios...
                       Class
Variable                  No
============================
p_game                 0.984
p_strikeout           1.0018
batting_avg        2611.7717
slg_percent           0.1681
on_base_percent      30.1002
p_era                 1.2299


Time taken to build model: 0.01 seconds
```

Figure 31: WEKA LogisticFunction Model on InfoGainAttributeEval Attributes



Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options: Use training set; [✓ Supplied test set]; Cross-validation Folds 10; Percentage split % 66

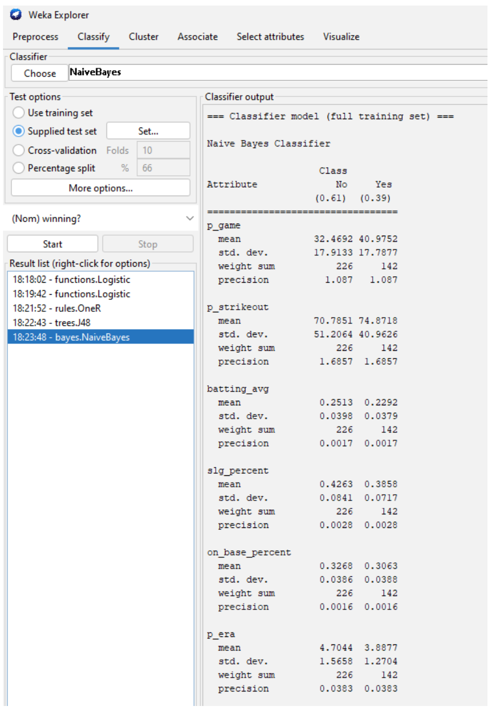
18:18:02 - functions.Logistic
18:19:42 - functions.Logistic

Classifier output

```
batting_avg           7.8678
slg_percent           -1.783
on_base_percent       3.4045
p_era                  0.207
Intercept            -2.2041


Odds Ratios...
                       Class
Variable                  No
============================
p_game                 0.984
p_strikeout           1.0018
batting_avg        2611.7717
slg_percent           0.1681
on_base_percent      30.1002
p_era                 1.2299


Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         115               64.2458 %
Incorrectly Classified Instances        64               35.7542 %
Kappa statistic                          0.1976
Mean absolute error                      0.4332
Root mean squared error                  0.4698
Relative absolute error                 91.3859 %
Root relative squared error             96.53   %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.809    0.623    0.674      0.809    0.736      0.206    0.668     0.752     No
                 0.377    0.191    0.553      0.377    0.448      0.206    0.668     0.505     Yes
Weighted Avg.    0.642    0.457    0.628      0.642    0.625      0.206    0.668     0.657

=== Confusion Matrix ===

  a  b   <-- classified as
 89 21 |  a = No
 43 26 |  b = Yes
```

Figure 32: WEKA LogisticFunction Summary Results on InfoGainAttributeEval Attributes

4.5.3 NaiveBayes - InfoGainAttributeEval



NaiveBayes

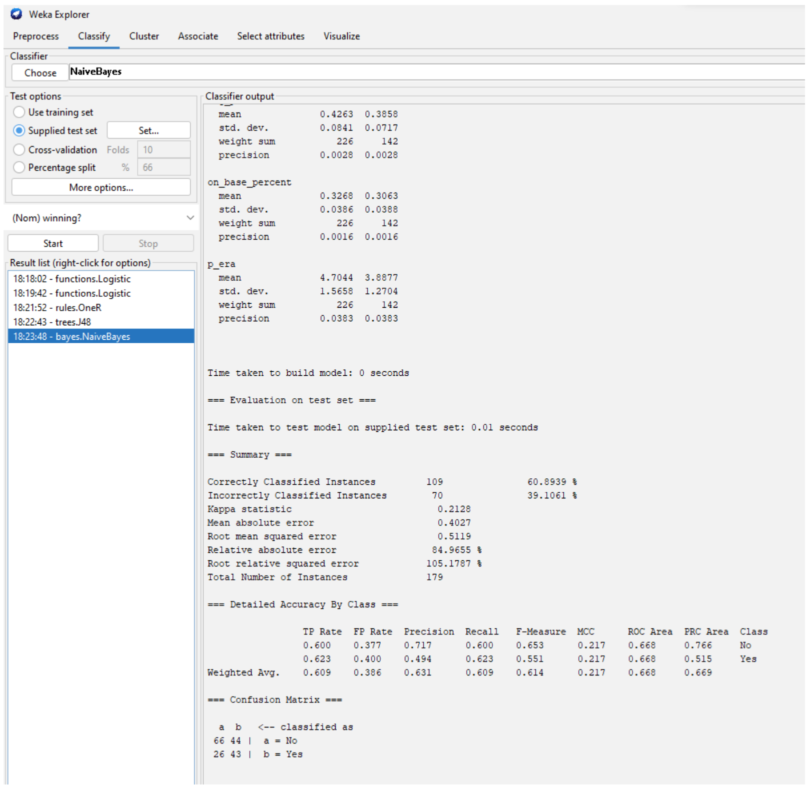
Supplied test set
Folds 10
% 66

(Nom) winning?

18:18:02 - functions.Logistic
18:19:42 - functions.Logistic
18:21:52 - rules.OneR
18:22:43 - trees.J48
18:23:48 - bayes.NaiveBayes

```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                     Class
Attribute               No     Yes
                    (0.61)  (0.39)
==================================
p_game
  mean             32.4692 40.9752
  std. dev.        17.9133 17.7877
  weight sum           226     142
  precision          1.087   1.087

p_strikeout
  mean             70.7851 74.8718
  std. dev.        51.2064 40.9626
  weight sum           226     142
  precision         1.6857  1.6857

batting_avg
  mean              0.2513  0.2292
  std. dev.         0.0398  0.0379
  weight sum           226     142
  precision         0.0017  0.0017

slg_percent
  mean              0.4263  0.3858
  std. dev.         0.0841  0.0717
  weight sum           226     142
  precision         0.0028  0.0028

on_base_percent
  mean              0.3268  0.3063
  std. dev.         0.0386  0.0388
  weight sum           226     142
  precision         0.0016  0.0016

p_era
  mean              4.7044  3.8877
  std. dev.         1.5658  1.2704
  weight sum           226     142
  precision         0.0383  0.0383
```

Figure 33: WEKA NaiveBayes Model on InfoGainAttributeEval Attributes



Classifier: NaiveBayes

Test options: [✓ Supplied test set] Folds 10 % 66

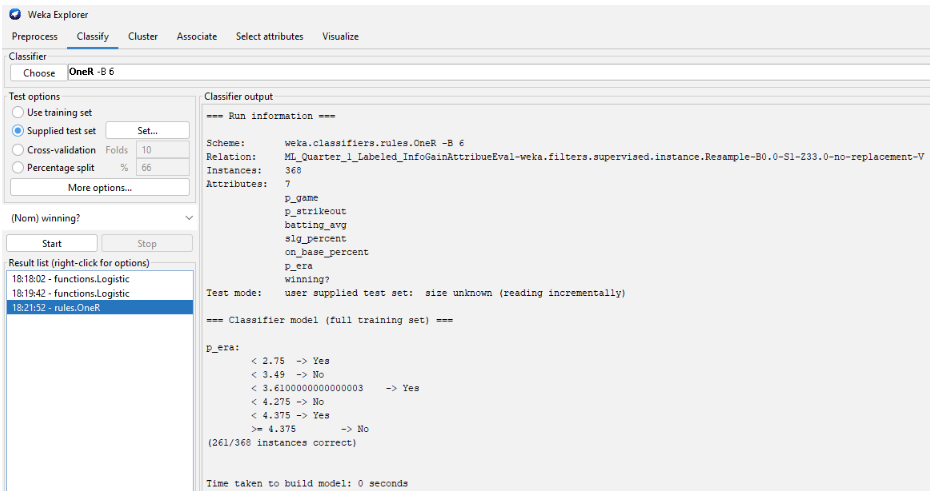
(Nom) winning?

```
18:18:02 - functions.Logistic
18:19:42 - functions.Logistic
18:21:52 - rules.OneR
18:22:43 - trees.J48
18:23:48 - bayes.NaiveBayes
```

```
  mean              0.4263  0.3858
  std. dev.         0.0841  0.0717
  weight sum           226     142
  precision         0.0028  0.0028

on_base_percent
  mean              0.3268  0.3063
  std. dev.         0.0386  0.0388
  weight sum           226     142
  precision         0.0016  0.0016

p_era
  mean              4.7044  3.8877
  std. dev.         1.5658  1.2704
  weight sum           226     142
  precision         0.0383  0.0383



Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         109               60.8939 %
Incorrectly Classified Instances        70               39.1061 %
Kappa statistic                          0.2128
Mean absolute error                      0.4027
Root mean squared error                  0.5119
Relative absolute error                 84.9655 %
Root relative squared error            105.1787 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.600    0.377    0.717      0.600    0.653      0.217    0.668     0.766     No
                 0.623    0.400    0.494      0.623    0.551      0.217    0.668     0.515     Yes
Weighted Avg.    0.609    0.386    0.631      0.609    0.614      0.217    0.668     0.669

=== Confusion Matrix ===

  a  b   <-- classified as
 66 44 |  a = No
 26 43 |  b = Yes
```

Figure 34: WEKA NaiveBayes Summary Results on InfoGainAttributeEval Attributes

4.5.4 OneR - InfoGainAttributeEval



Classifier: OneR -B 6

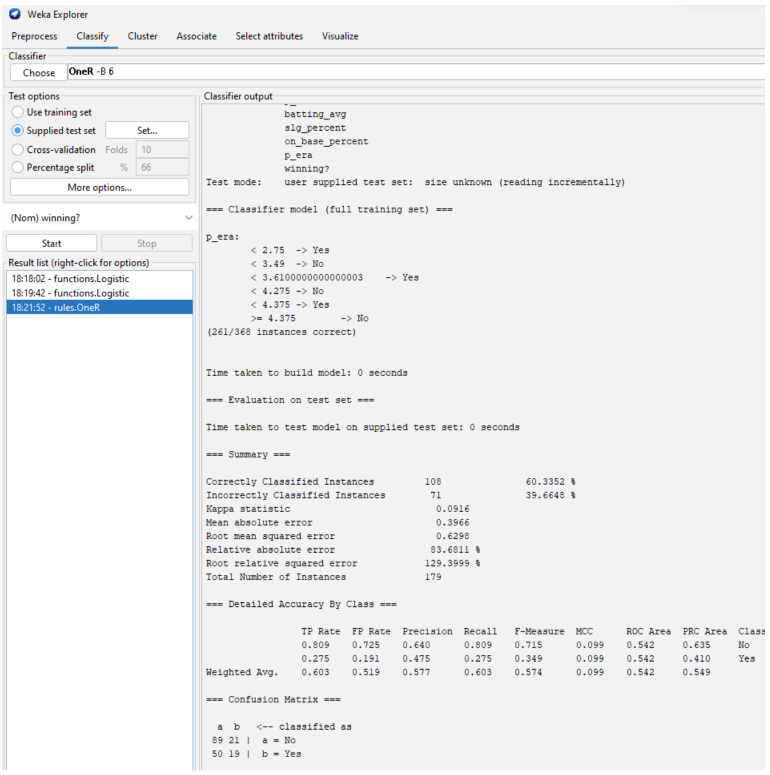
Test options:
- Use training set
- [✓ Supplied test set]
- Cross-validation Folds 10
- Percentage split % 66

(Nom) winning?

Result list:
- 18:18:02 - functions.Logistic
- 18:19:42 - functions.Logistic
- 18:21:52 - rules.OneR

```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
	< 2.75	-> Yes
	< 3.49	-> No
	< 3.6100000000000003	-> Yes
	< 4.275	-> No
	< 4.375	-> Yes
	>= 4.375	-> No
(261/368 instances correct)


Time taken to build model: 0 seconds
```

Figure 35: WEKA OneR Model on InfoGainAttributeEval Attributes



OneR -B 6

Supplied test set
Folds 10
% 66

(Nom) winning?

18:18:02 - functions.Logistic
18:19:42 - functions.Logistic
18:21:52 - rules.OneR

```
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
        < 2.75  -> Yes
        < 3.49  -> No
        < 3.6100000000000003    -> Yes
        < 4.275 -> No
        < 4.375 -> Yes
        >= 4.375        -> No
(261/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         108               60.3352 %
Incorrectly Classified Instances        71               39.6648 %
Kappa statistic                          0.0916
Mean absolute error                      0.3966
Root mean squared error                  0.6298
Relative absolute error                 83.6811 %
Root relative squared error            129.3999 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.809    0.725    0.640      0.809    0.715      0.099    0.542     0.635     No
                 0.275    0.191    0.475      0.275    0.349      0.099    0.542     0.410     Yes
Weighted Avg.    0.603    0.519    0.577      0.603    0.574      0.099    0.542     0.549

=== Confusion Matrix ===

  a  b   <-- classified as
 89 21 |  a = No
 50 19 |  b = Yes
```

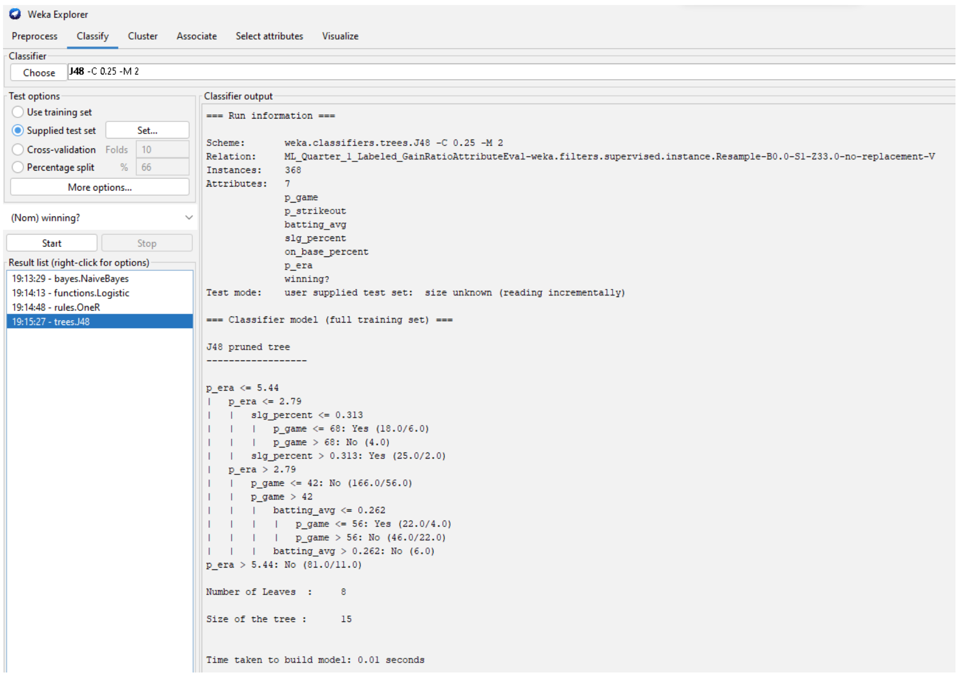
Figure 36: WEKA OneR Summary Results on InfoGainAttributeEval Attributes

4.5.5 Analysis

J48 performed the best with this attribute selection method, with **LogisticFunction** being a close second. Once again, **J48** was a lot better at categorizing true positives while **LogisticFunction** was a lot better at categorizing true negatives.

4.6 GainRatioAttributeEval

4.6.1 J48 - GainRatioAttributeEval



Classifier: J48 -C 0.25 -M 2

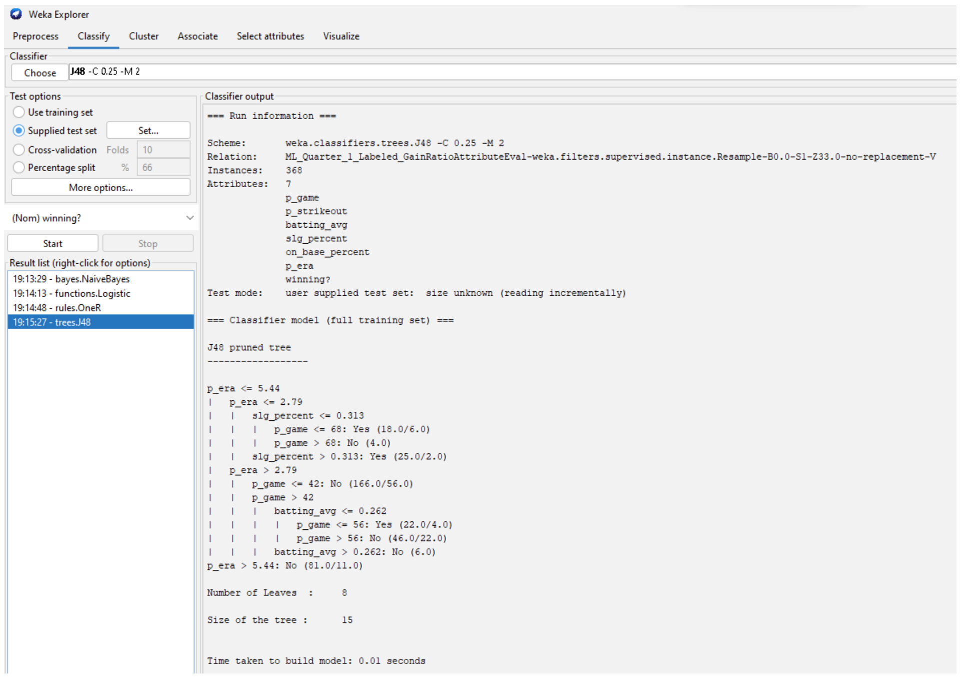
Test options: Use training set [✓ Supplied test set] Cross-validation Folds 10 Percentage split % 66

(Nom) winning?

Result list (right-click for options):
19:13:29 - bayes.NaiveBayes
19:14:13 - functions.Logistic
19:14:48 - rules.OneR
19:15:27 - trees.J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     ML_Quarter_1_Labeled_GainRatioAttributeEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

J48 pruned tree
------------------

p_era <= 5.44
|   p_era <= 2.79
|   |   slg_percent <= 0.313
|   |   |   p_game <= 68: Yes (18.0/6.0)
|   |   |   p_game > 68: No (4.0)
|   |   slg_percent > 0.313: Yes (25.0/2.0)
|   p_era > 2.79
|   |   p_game <= 42: No (166.0/56.0)
|   |   p_game > 42
|   |   |   batting_avg <= 0.262
|   |   |   |   p_game <= 56: Yes (22.0/4.0)
|   |   |   |   p_game > 56: No (46.0/22.0)
|   |   |   batting_avg > 0.262: No (6.0)
p_era > 5.44: No (81.0/11.0)

Number of Leaves  :     8

Size of the tree :      15


Time taken to build model: 0.01 seconds
```

Figure 37: WEKA J48 Model on GainRatioAttributeEval Attributes



Classifier: J48 -C 0.25 -M 2

Test options: Use training set [✓ Supplied test set] Cross-validation Folds 10 Percentage split % 66

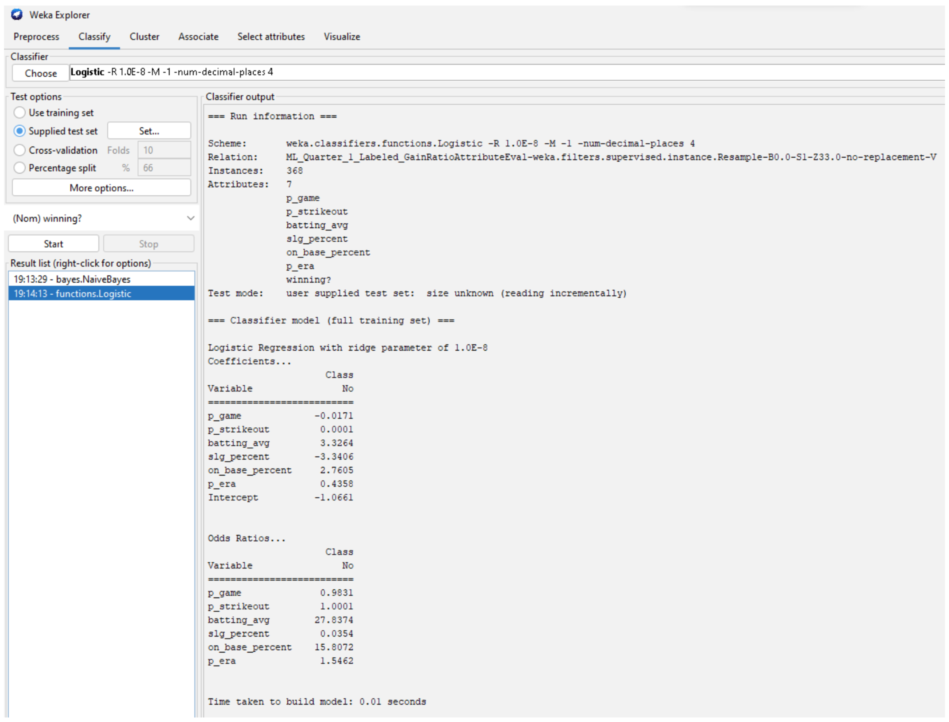
(Nom) winning?

Result list (right-click for options)
19:13:29 - bayes.NaiveBayes
19:14:13 - functions.Logistic
19:14:48 - rules.OneR
19:15:27 - trees.J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     ML_Quarter_1_Labeled_GainRatioAttributeEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

J48 pruned tree
------------------

p_era <= 5.44
|   p_era <= 2.79
|   |   slg_percent <= 0.313
|   |   |   p_game <= 68: Yes (18.0/6.0)
|   |   |   p_game > 68: No (4.0)
|   |   slg_percent > 0.313: Yes (25.0/2.0)
|   p_era > 2.79
|   |   p_game <= 42: No (166.0/56.0)
|   |   p_game > 42
|   |   |   batting_avg <= 0.262
|   |   |   |   p_game <= 56: Yes (22.0/4.0)
|   |   |   |   p_game > 56: No (46.0/22.0)
|   |   |   batting_avg > 0.262: No (6.0)
p_era > 5.44: No (81.0/11.0)

Number of Leaves  :     8

Size of the tree :      15


Time taken to build model: 0.01 seconds
```

Figure 38: WEKA J48 Classification Results on GainRatioAttributeEval Attributes

4.6.2 LogisticFunction - GainRatioAttributeEval

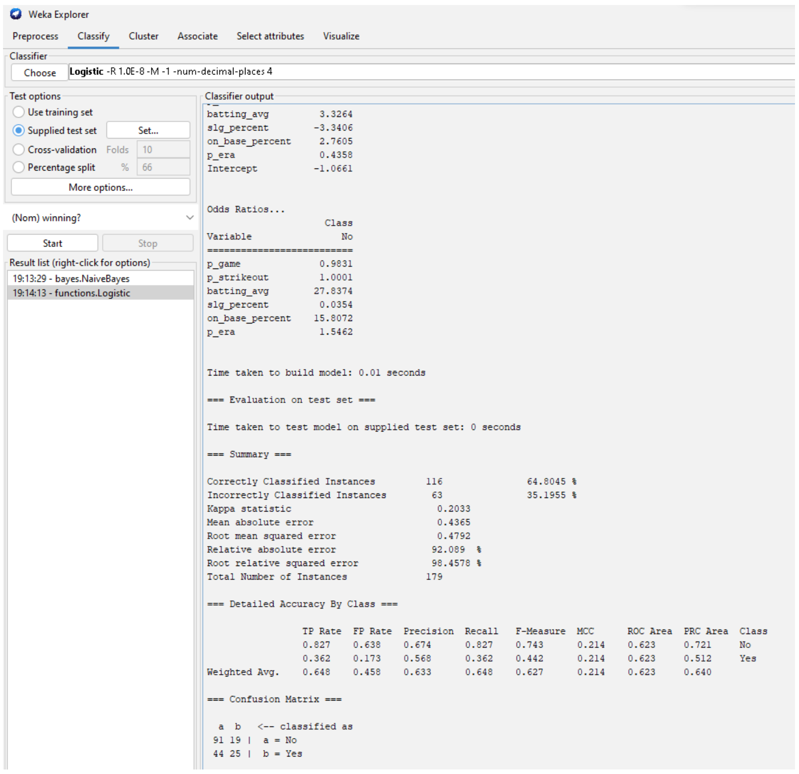


Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Use training set [✓ Supplied test set] Cross-validation Folds 10 Percentage split % 66

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     ML_Quarter_1_Labeled_GainRatioAttributeEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                     Class
Variable                No
==========================
p_game             -0.0171
p_strikeout         0.0001
batting_avg         3.3264
slg_percent        -3.3406
on_base_percent     2.7605
p_era               0.4358
Intercept          -1.0661


Odds Ratios...
                     Class
Variable                No
==========================
p_game              0.9831
p_strikeout         1.0001
batting_avg        27.8374
slg_percent         0.0354
on_base_percent    15.8072
p_era               1.5462


Time taken to build model: 0.01 seconds
```

Figure 39: WEKA LogisticFunction Model on GainRatioAttributeEval Attributes


Classifier: Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options: Supplied test set

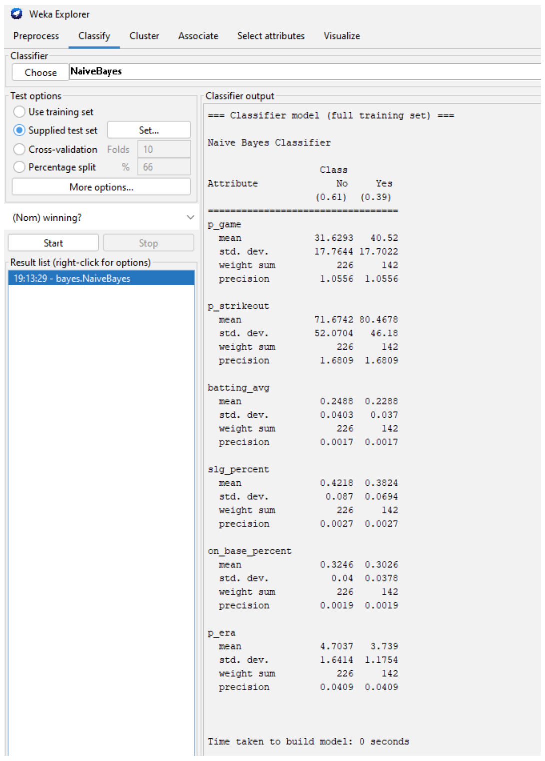
(Nom) winning?

Result list (right-click for options)
19:13:29 - bayes.NaiveBayes
19:14:13 - functions.Logistic

```
batting_avg         3.3264
slg_percent        -3.3406
on_base_percent     2.7605
p_era               0.4358
Intercept          -1.0661


Odds Ratios...
                     Class
Variable                No
==========================
p_game              0.9831
p_strikeout         1.0001
batting_avg        27.8374
slg_percent         0.0354
on_base_percent    15.8072
p_era               1.5462


Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         116               64.8045 %
Incorrectly Classified Instances        63               35.1955 %
Kappa statistic                          0.2033
Mean absolute error                      0.4365
Root mean squared error                  0.4792
Relative absolute error                 92.089  %
Root relative squared error             98.4578 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.827    0.638    0.674      0.827    0.743      0.214    0.623     0.721     No
                 0.362    0.173    0.568      0.362    0.442      0.214    0.623     0.512     Yes
Weighted Avg.    0.648    0.458    0.633      0.648    0.627      0.214    0.623     0.640

=== Confusion Matrix ===

  a  b   <-- classified as
 91 19 |  a = No
 44 25 |  b = Yes
```

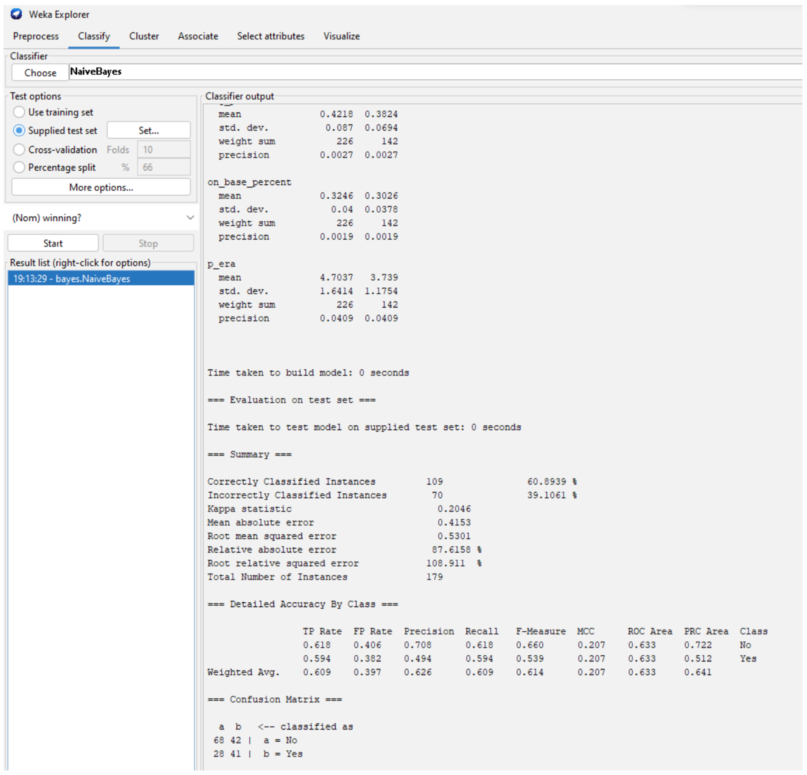
Figure 40: WEKA LogisticFunction Summary Results on GainRatioAttributeEval Attributes

4.6.3 NaiveBayes - GainRatioAttributeEval



```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                     Class
Attribute               No     Yes
                    (0.61)  (0.39)
==================================
p_game
  mean             31.6293   40.52
  std. dev.        17.7644 17.7022
  weight sum           226     142
  precision         1.0556  1.0556

p_strikeout
  mean             71.6742 80.4678
  std. dev.        52.0704   46.18
  weight sum           226     142
  precision         1.6809  1.6809

batting_avg
  mean              0.2488  0.2288
  std. dev.         0.0403   0.037
  weight sum           226     142
  precision         0.0017  0.0017

slg_percent
  mean              0.4218  0.3824
  std. dev.          0.087  0.0694
  weight sum           226     142
  precision         0.0027  0.0027

on_base_percent
  mean              0.3246  0.3026
  std. dev.           0.04  0.0378
  weight sum           226     142
  precision         0.0019  0.0019

p_era
  mean              4.7037   3.739
  std. dev.         1.6414  1.1754
  weight sum           226     142
  precision         0.0409  0.0409



Time taken to build model: 0 seconds
```

Figure 41: WEKA NaiveBayes Model on GainRatioAttributeEval Attributes



Classifier: NaiveBayes

Test options: Supplied test set; Folds 10; % 66

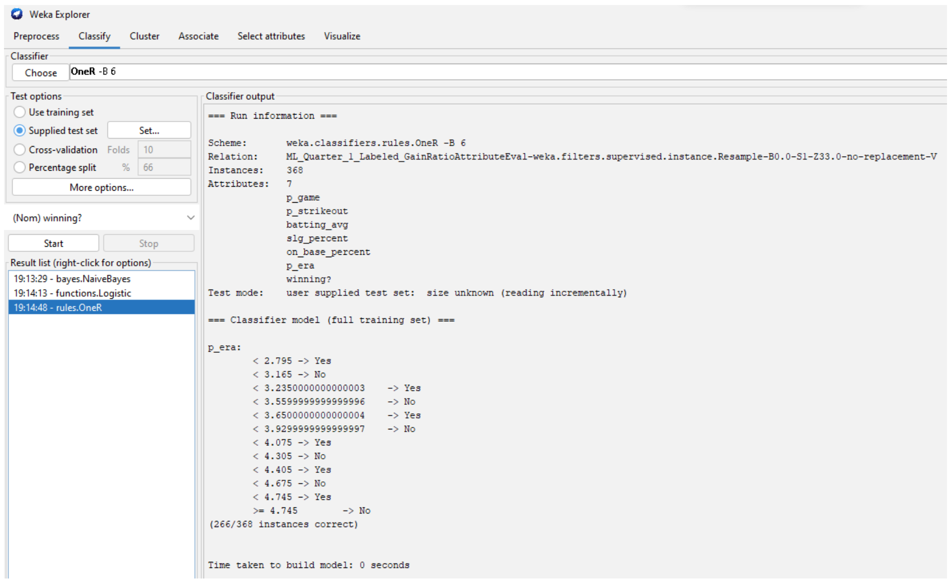
(Nom) winning?

19:13:29 - bayes.NaiveBayes

```
  mean             0.4218  0.3824
  std. dev.         0.087  0.0694
  weight sum          226     142
  precision        0.0027  0.0027

on_base_percent
  mean             0.3246  0.3026
  std. dev.          0.04  0.0378
  weight sum          226     142
  precision        0.0019  0.0019

p_era
  mean             4.7037   3.739
  std. dev.        1.6414  1.1754
  weight sum          226     142
  precision        0.0409  0.0409



Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         109               60.8939 %
Incorrectly Classified Instances        70               39.1061 %
Kappa statistic                          0.2046
Mean absolute error                      0.4153
Root mean squared error                  0.5301
Relative absolute error                 87.6158 %
Root relative squared error            108.911  %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.618    0.406    0.708      0.618    0.660      0.207    0.633     0.722     No
                 0.594    0.382    0.494      0.594    0.539      0.207    0.633     0.512     Yes
Weighted Avg.    0.609    0.397    0.626      0.609    0.614      0.207    0.633     0.641

=== Confusion Matrix ===

  a  b   <-- classified as
 68 42 |  a = No
 28 41 |  b = Yes
```

Figure 42: WEKA NaiveBayes Summary Results on GainRatioAttributeEval Attributes

4.6.4 OneR - GainRatioAttributeEval



Classifier: OneR -B 6

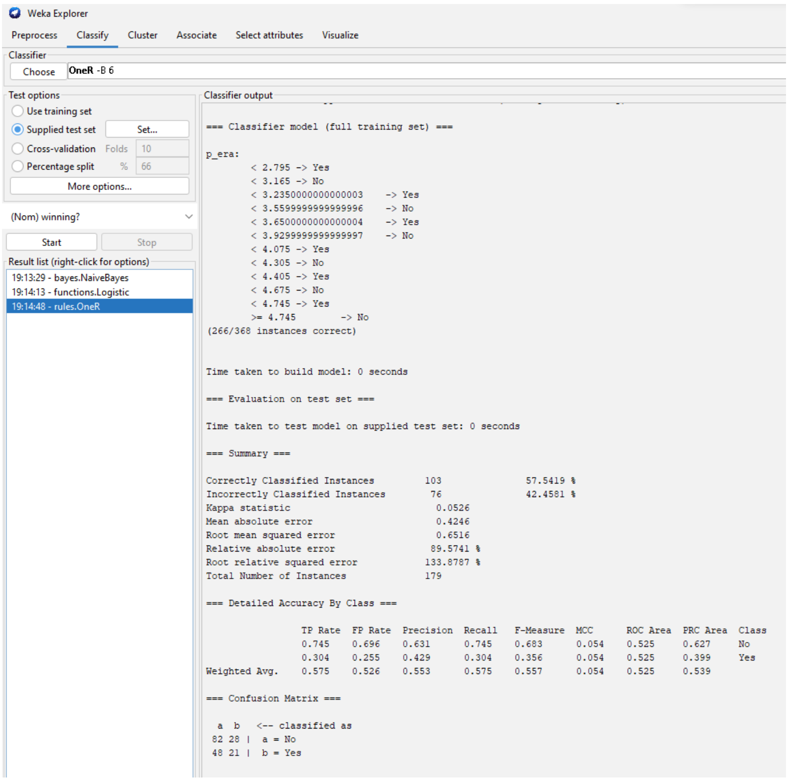
Test options: Supplied test set; Cross-validation Folds 10; Percentage split % 66

(Nom) winning?

Result list (right-click for options):
19:13:29 - bayes.NaiveBayes
19:14:13 - functions.Logistic
19:14:48 - rules.OneR

```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     ML_Quarter_1_Labeled_GainRatioAttributeEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
	< 2.795 -> Yes
	< 3.165 -> No
	< 3.2350000000000003	-> Yes
	< 3.5599999999999996	-> No
	< 3.6500000000000004	-> Yes
	< 3.9299999999999997	-> No
	< 4.075 -> Yes
	< 4.305 -> No
	< 4.405 -> Yes
	< 4.675 -> No
	< 4.745 -> Yes
	>= 4.745	-> No
(266/368 instances correct)


Time taken to build model: 0 seconds
```

Figure 43: WEKA OneR Model on GainRatioAttributeEval Attributes



Classifier: OneR -B 6

Test options: Use training set [✓ Supplied test set] Cross-validation (Folds 10) Percentage split (% 66)

(Nom) winning?

```
=== Classifier model (full training set) ===

p_era:
        < 2.795 -> Yes
        < 3.165 -> No
        < 3.2350000000000003    -> Yes
        < 3.5599999999999996    -> No
        < 3.6500000000000004    -> Yes
        < 3.9299999999999997    -> No
        < 4.075 -> Yes
        < 4.305 -> No
        < 4.405 -> Yes
        < 4.675 -> No
        < 4.745 -> Yes
        >= 4.745        -> No
(266/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         103               57.5419 %
Incorrectly Classified Instances        76               42.4581 %
Kappa statistic                          0.0526
Mean absolute error                      0.4246
Root mean squared error                  0.6516
Relative absolute error                 89.5741 %
Root relative squared error            133.8787 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.745    0.696    0.631      0.745    0.683      0.054    0.525     0.627     No
                 0.304    0.255    0.429      0.304    0.356      0.054    0.525     0.399     Yes
Weighted Avg.    0.575    0.526    0.553      0.575    0.557      0.054    0.525     0.539

=== Confusion Matrix ===

  a  b   <-- classified as
 82 28 |  a = No
 48 21 |  b = Yes
```

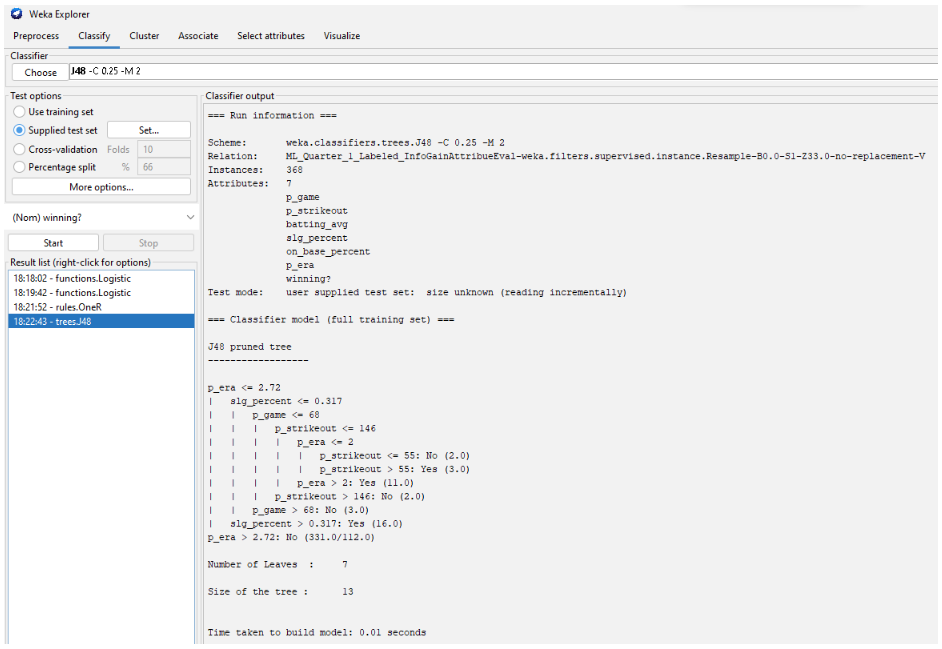
Figure 44: WEKA OneR Summary Results on GainRatioAttributeEval Attributes

4.6.5 Analysis

J48 performed the best with this attribute selection method, with **LogisticFunction** being a close second. Much like in previous cases, **J48** was a lot better at categorizing true positives while **LogisticFunction** was a lot better at categorizing true negatives.

4.7 InfoGainAttributeEval

4.7.1 J48 - InfoGainAttributeEval



Classifier: J48 -C 0.25 -M 2

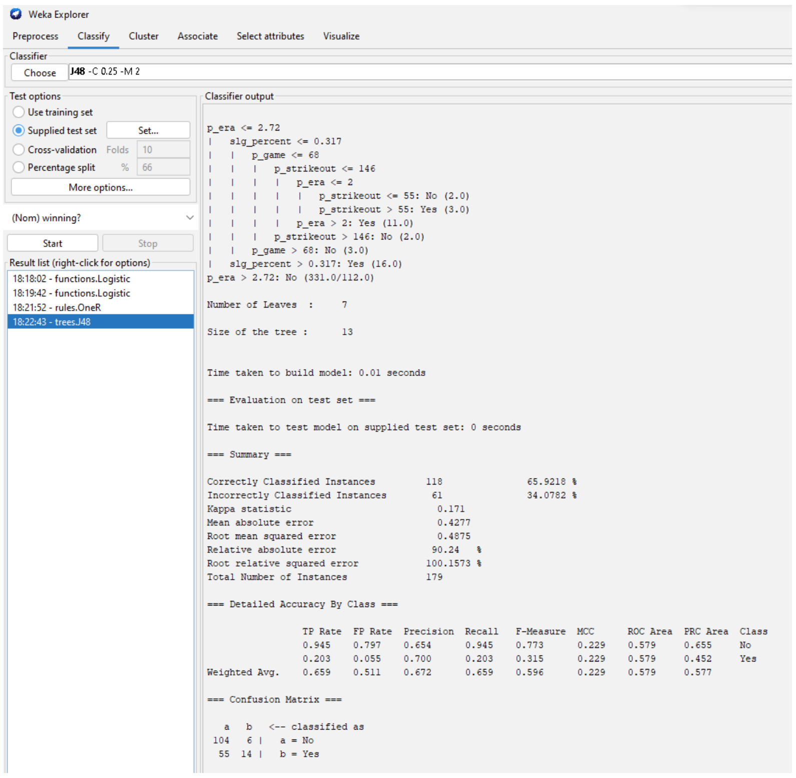
Test options: Use training set [✓ Supplied test set] Cross-validation Folds 10 Percentage split % 66

(Nom) winning?

Result list (right-click for options):
- 18:18:02 - functions.Logistic
- 18:19:42 - functions.Logistic
- 18:21:52 - rules.OneR
- 18:22:43 - trees.J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

J48 pruned tree
------------------

p_era <= 2.72
|   slg_percent <= 0.317
|   |   p_game <= 68
|   |   |   p_strikeout <= 146
|   |   |   |   p_era <= 2
|   |   |   |   |   p_strikeout <= 55: No (2.0)
|   |   |   |   |   p_strikeout > 55: Yes (3.0)
|   |   |   |   p_era > 2: Yes (11.0)
|   |   |   p_strikeout > 146: No (2.0)
|   |   p_game > 68: No (3.0)
|   slg_percent > 0.317: Yes (16.0)
p_era > 2.72: No (331.0/112.0)

Number of Leaves  :     7

Size of the tree :      13


Time taken to build model: 0.01 seconds
```

Figure 45: WEKA J48 Model on InfoGainAttributeEval Attributes

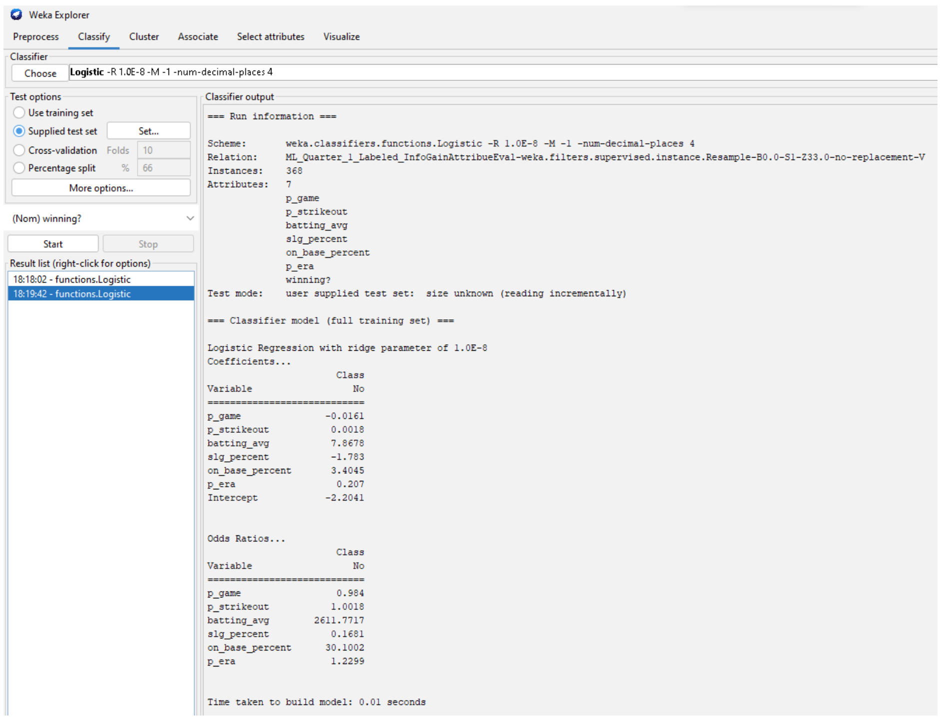


```
Classifier: J48 -C 0.25 -M 2

Test options
( ) Use training set
(•) Supplied test set
( ) Cross-validation   Folds 10
( ) Percentage split   %  66

(Nom) winning?

Result list (right-click for options)
18:18:02 - functions.Logistic
18:19:42 - functions.Logistic
18:21:52 - rules.OneR
18:22:43 - trees.J48
```

```
Classifier output

p_era <= 2.72
|   slg_percent <= 0.317
|   |   p_game <= 68
|   |   |   p_strikeout <= 146
|   |   |   |   p_era <= 2
|   |   |   |   |   p_strikeout <= 55: No (2.0)
|   |   |   |   |   p_strikeout > 55: Yes (3.0)
|   |   |   |   p_era > 2: Yes (11.0)
|   |   |   p_strikeout > 146: No (2.0)
|   |   p_game > 68: No (3.0)
|   slg_percent > 0.317: Yes (16.0)
p_era > 2.72: No (331.0/112.0)

Number of Leaves  :     7

Size of the tree :      13


Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         118               65.9218 %
Incorrectly Classified Instances        61               34.0782 %
Kappa statistic                          0.171
Mean absolute error                      0.4277
Root mean squared error                  0.4875
Relative absolute error                 90.24   %
Root relative squared error            100.1573 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.945    0.797    0.654      0.945    0.773      0.229    0.579     0.655     No
                 0.203    0.055    0.700      0.203    0.315      0.229    0.579     0.452     Yes
Weighted Avg.    0.659    0.511    0.672      0.659    0.596      0.229    0.579     0.577

=== Confusion Matrix ===

   a   b   <-- classified as
 104   6 |   a = No
  55  14 |   b = Yes
```

Figure 46: WEKA J48 Clasifcation Results on InfoGainAttributeEval Attributes

4.7.2 LogisticFunction - InfoGainAttributeEval



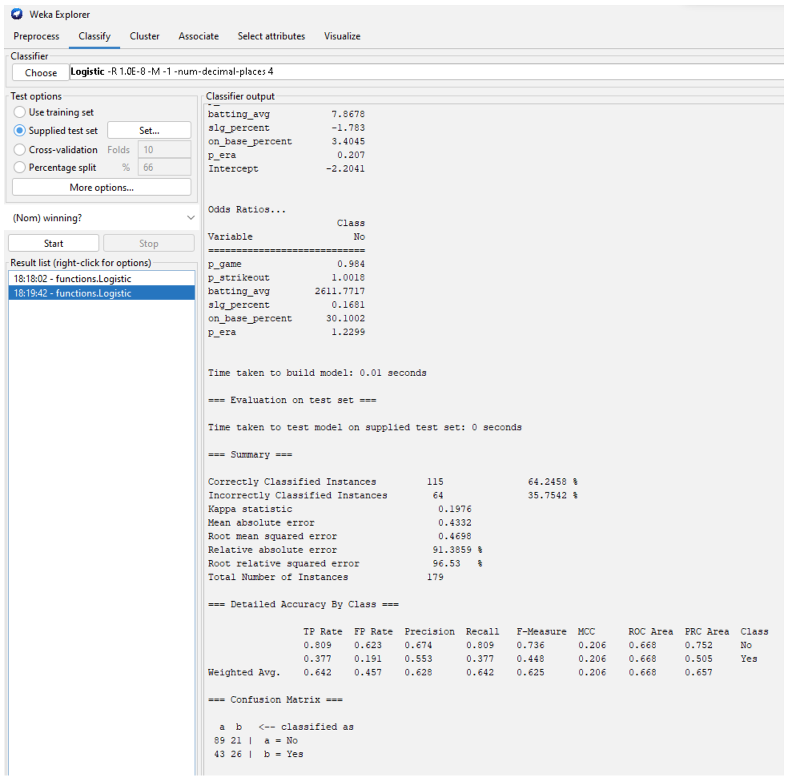
Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

(Nom) winning?

18:18:02 - functions.Logistic
18:19:42 - functions.Logistic

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     ML_Quarter_1_Labeled_InfoGainAttribueEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_strikeout
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                       Class
Variable                  No
============================
p_game               -0.0161
p_strikeout           0.0018
batting_avg           7.8678
slg_percent           -1.783
on_base_percent       3.4045
p_era                  0.207
Intercept            -2.2041


Odds Ratios...
                       Class
Variable                  No
============================
p_game                 0.984
p_strikeout           1.0018
batting_avg        2611.7717
slg_percent           0.1681
on_base_percent      30.1002
p_era                 1.2299


Time taken to build model: 0.01 seconds
```

Figure 47: WEKA LogisticFunction Model on InfoGainAttributeEval Attributes



Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Folds 10

% 66

(Nom) winning?

18:18:02 - functions.Logistic
18:19:42 - functions.Logistic

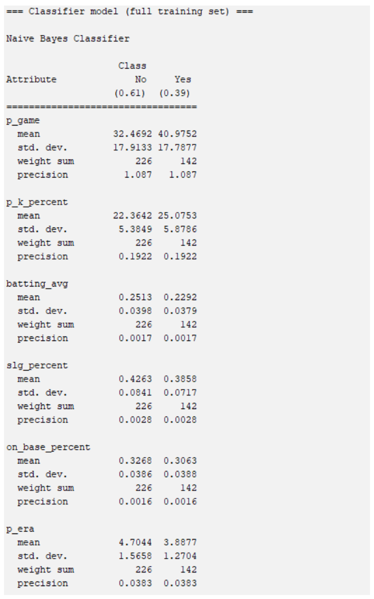
```
batting_avg          7.8678
slg_percent          -1.783
on_base_percent      3.4045
p_era                 0.207
Intercept           -2.2041


Odds Ratios...
                      Class
Variable                 No
============================
p_game                0.984
p_strikeout          1.0018
batting_avg       2611.7717
slg_percent          0.1681
on_base_percent     30.1002
p_era                1.2299


Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         115               64.2458 %
Incorrectly Classified Instances        64               35.7542 %
Kappa statistic                          0.1976
Mean absolute error                      0.4332
Root mean squared error                  0.4698
Relative absolute error                 91.3859 %
Root relative squared error             96.53   %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.809    0.623    0.674      0.809    0.736      0.206    0.668     0.752     No
                 0.377    0.191    0.553      0.377    0.448      0.206    0.668     0.505     Yes
Weighted Avg.    0.642    0.457    0.628      0.642    0.625      0.206    0.668     0.657

=== Confusion Matrix ===

  a  b   <-- classified as
 89 21 |  a = No
 43 26 |  b = Yes
```

Figure 48: WEKA LogisticFunction Summary Results on InfoGainAttributeEval Attributes

4.7.3 Analysis

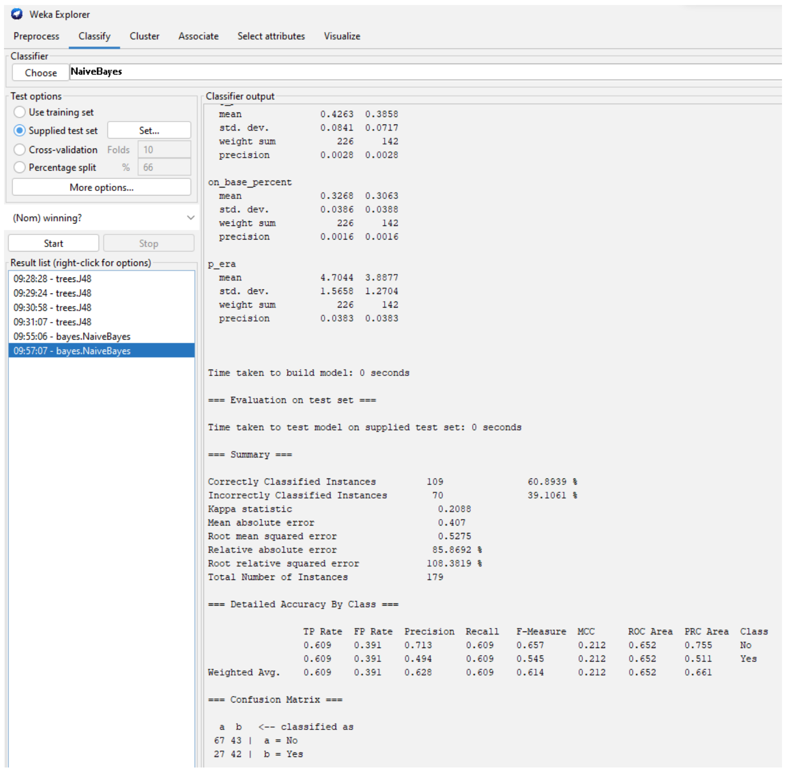
J48 performed the best with this attribute selection method, with **LogisticFunction** being a close second. **J48** was a lot better at categorizing true positives while **LogisticFunction** was a lot better at categorizing true negatives.

4.7.4 NaiveBayes - CorrelationAttributeEval



```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                    Class
Attribute              No     Yes
                   (0.61)  (0.39)
==================================
p_game
  mean             32.4692 40.9752
  std. dev.        17.9133 17.7877
  weight sum           226     142
  precision          1.087   1.087

p_k_percent
  mean             22.3642 25.0753
  std. dev.         5.3849  5.8786
  weight sum           226     142
  precision         0.1922  0.1922

batting_avg
  mean              0.2513  0.2292
  std. dev.         0.0398  0.0379
  weight sum           226     142
  precision         0.0017  0.0017

slg_percent
  mean              0.4263  0.3858
  std. dev.         0.0841  0.0717
  weight sum           226     142
  precision         0.0028  0.0028

on_base_percent
  mean              0.3268  0.3063
  std. dev.         0.0386  0.0388
  weight sum           226     142
  precision         0.0016  0.0016

p_era
  mean              4.7044  3.8877
  std. dev.         1.5658  1.2704
  weight sum           226     142
  precision         0.0383  0.0383
```

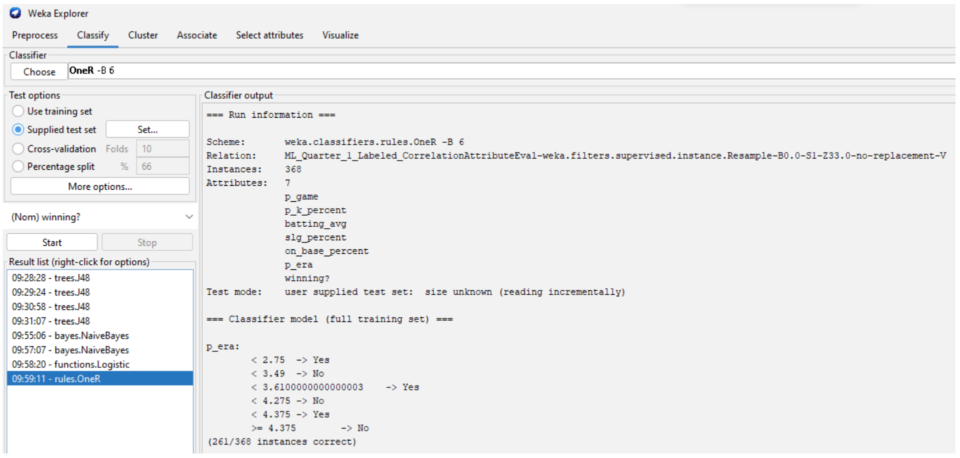
Figure 49: WEKA NaiveBayes Model on CorrelationAttributeEval Attributes



```
  mean            0.4263  0.3858
  std. dev.       0.0841  0.0717
  weight sum         226     142
  precision       0.0028  0.0028

on_base_percent
  mean            0.3268  0.3063
  std. dev.       0.0386  0.0388
  weight sum         226     142
  precision       0.0016  0.0016

p_era
  mean            4.7044  3.8877
  std. dev.       1.5658  1.2704
  weight sum         226     142
  precision       0.0383  0.0383



Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         109               60.8939 %
Incorrectly Classified Instances        70               39.1061 %
Kappa statistic                          0.2088
Mean absolute error                      0.407
Root mean squared error                  0.5275
Relative absolute error                 85.8692 %
Root relative squared error            108.3819 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.609    0.391    0.713      0.609    0.657      0.212    0.652     0.755     No
                 0.609    0.391    0.494      0.609    0.545      0.212    0.652     0.511     Yes
Weighted Avg.    0.609    0.391    0.628      0.609    0.614      0.212    0.652     0.661

=== Confusion Matrix ===

  a  b   <-- classified as
 67 43 |  a = No
 27 42 |  b = Yes
```

Figure 50: WEKA NaiveBayes Summary Results on CorrelationAttributeEval Attributes

4.7.5 OneR - CorrelationAttributeEval



OneR -B 6

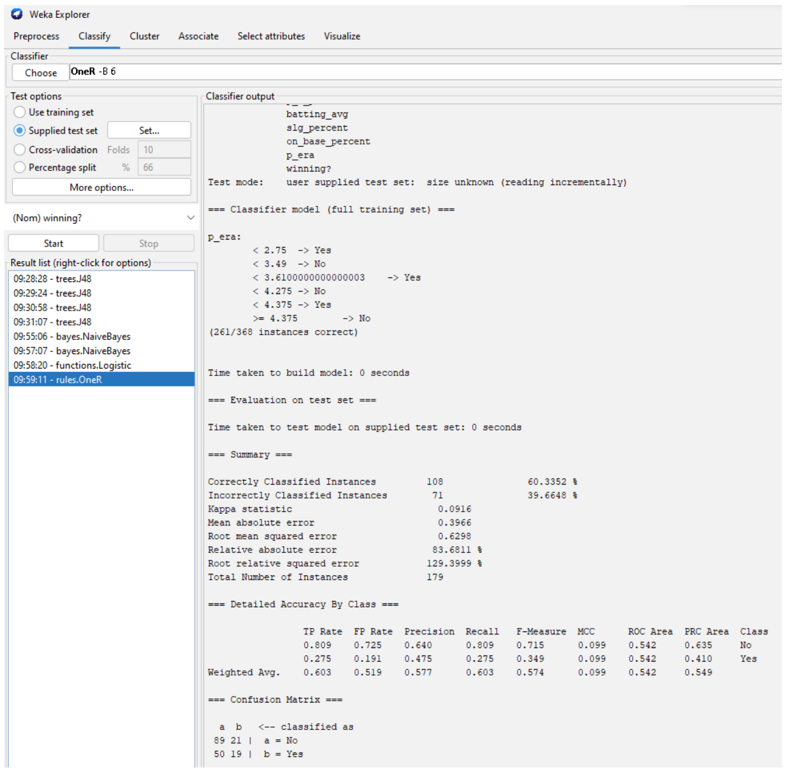
Supplied test set

(Nom) winning?

```
09:28:28 - trees.J48
09:29:24 - trees.J48
09:30:58 - trees.J48
09:31:07 - trees.J48
09:55:06 - bayes.NaiveBayes
09:57:07 - bayes.NaiveBayes
09:58:20 - functions.Logistic
09:59:11 - rules.OneR
```

```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     ML_Quarter_1_Labeled_CorrelationAttributeEval-weka.filters.supervised.instance.Resample-B0.0-S1-Z33.0-no-replacement-V
Instances:    368
Attributes:   7
              p_game
              p_k_percent
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
        < 2.75  -> Yes
        < 3.49  -> No
        < 3.6100000000000003    -> Yes
        < 4.275 -> No
        < 4.375 -> Yes
        >= 4.375        -> No
(261/368 instances correct)
```

Figure 51: WEKA OneR Model on CorrelationAttributeEval Attributes



```
              batting_avg
              slg_percent
              on_base_percent
              p_era
              winning?
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

p_era:
        < 2.75  -> Yes
        < 3.49  -> No
        < 3.6100000000000003    -> Yes
        < 4.275 -> No
        < 4.375 -> Yes
        >= 4.375        -> No
(261/368 instances correct)


Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         108               60.3352 %
Incorrectly Classified Instances        71               39.6648 %
Kappa statistic                          0.0916
Mean absolute error                      0.3966
Root mean squared error                  0.6298
Relative absolute error                 83.6811 %
Root relative squared error            129.3999 %
Total Number of Instances              179

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.809    0.725    0.640      0.809    0.715      0.099    0.542     0.635     No
                 0.275    0.191    0.475      0.275    0.349      0.099    0.542     0.410     Yes
Weighted Avg.    0.603    0.519    0.577      0.603    0.574      0.099    0.542     0.549

=== Confusion Matrix ===

  a  b   <-- classified as
 89 21 |  a = No
 50 19 |  b = Yes
```

Figure 52: WEKA OneR Summary Results on CorrelationAttributeEval Attributes

4.7.6 Analysis

J48 performed the best with this attribute selection method, with **LogisticFunction** being a close second. **J48** was a lot better at categorizing true positives while **LogisticFunction** was a lot better at categorizing true negatives. This time, **LogisticFunction** performed a bit worse than normal compared to **J48**.

5 Conclusion

Looking at the data given to us by WEKA, we can see that the best combination of attribute selection and classification method is **WrapperSubsetEval** and **LogisticFunction**. This method classified with an accuracy of 67.0391%. However, with just one more instance misclassified, **WraperSubsetEval** and **J48** was a close second. Overall, the best performing models used the **WrapperSubsetEval** attribute selection method. This is probably due to the fact that it is one of the more complex attribute selection methods, and therefore is more accurate.

There are many reasons why **WrapperSubsetEval** and **LogisticFunction** are better in our case over **J48**. The former had the highest true positive rate (0.670), the lowest false positive rate (0.423), and the highest F-measure (0.655). These three facts combined make **WrapperSubsetEval** and **LogisticFunction** the undeniable best attribute selection method and classification method combination.

Overall, the accuracy of the classifier models was low, with the highest at just over 67%. This was surprising because we expected these statistics to be better predictors of performance, but the results make sense given the random nature of baseball and other external factors, like scouting reports and weather that could skew the statistics in some way.

6 Discussion

Throughout the course of this project, we learned a lot about data preprocessing and classification. We learned that well-chosen attributes are essential to having an accurate model, as evidenced by the fact that the attributes that we selected because we thought were the most predictive of pitcher performance were actually not the best attributes, and that greatly decreased the accuracy of any classifier algorithm, and even was the worst performing attribute selection method based on the results of the classification methods. This is a classic case of garbage in, garbage out.

We also learned that sometimes you cannot just compare the accuracy to determine which of two models is better. There needs to be some sort of justification. In our case, the difference between **LogisticFunction** and **J48** for the **WrapperSubsetEval** was almost negligible, but several factors pushed us to determine that **LogisticFunction** was better.

Finally, we learned that more is better. It is always better to test out more combinations of classifier models and attribute selection methods. While some are there to provide a baseline estimate, like **OneR** and **NaiveBayes**, they help provide us with insights if the more complex models go wrong and perform worse than the lightweight ones. Through testing of more combinations, we are more likely to find better performing models. While

J48 performed the best on most attribute selection methods, it did not perform the best overall, and it would be wrong for us to assume so. If we had not tried out **LogisticFunction** and **WrapperSubsetEval**, we may not have gotten the same level of performance.
The work was split evenly between both members during this project, with each person doing half of each step. We collaborated on the best way to handle missing values, and ended up deciding on just deleting rows we found were not needed. We each did three attribute reduction methods, collaborating on the **SelfSelected** method. We then ran classifier models on the two attribute reduction methods that we each did, and took screenshots of our work. Progress was updated through a git repository, so we always had access to each other's work. Finally, we collaborated on the writeup and the presentation.

7 Sources

We got our data from https://www.baseballsavant.mlb.com and used WEKA to preprocess and do classification work.
Datasets for this project can be found at https://drive.google.com/drive/folders/1sgNZJn71GUput_BZugSGtm8ybw6_6p0P?usp=sharing. This link includes both the original datasets as well as the modified datasets.
Research on models was done on both the official WEKA sourceforge site and through https://www.medium.com. The links to sources used, as well as the original dataset, are in the Bibliography on the next page.
WEKA (Waikato Environment for Knowledge Analysis), is a collection of machine learning and data mining algorithms and tasks. It was developed at the University of Waikato, New Zealand.

8 Bibliography

https://baseballsavant.mlb.com/leaderboard/custom?year=2021&type=pitcher&filter=
&sort=2&sortDir=asc&min=100&selections=p_game,p_total_hits,p_home_run,p_strikeout,
p_walk,p_k_percent,p_bb_percent,batting_avg,slg_percent,on_base_percent,on_base_plus_slg,
p_earned_run,p_win,p_loss,p_era,p_rbi,p_called_strike,p_unearned_run,exit_velocity_avg,
launch_angle_avg,sweet_spot_percent,barrel_batted_rate,hard_hit_percent,meatball_percent,
pitch_hand,n_fastball_formatted,fastball_avg_speed,n_offspeed_formatted,offspeed_avg_
speed,&chart=false&x=p_total_hits&y=p_total_hits&r=no&chartType=beeswarm

https://weka.sourceforge.io/doc.dev/weka/attributeSelection/WrapperSubsetEval.html

https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html

https://weka.sourceforge.io/doc.dev/weka/attributeSelection/GainRatioAttributeEval.html

https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CorrelationAttributeEval.
html

https://weka.sourceforge.io/doc.dev/weka/filters/supervised/instance/Resample.html

https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e

https://medium.com/swlh/lets-talk-logistic-regression-4b2072ad7b4e

https://medium.com/x8-the-ai-community/a-simple-introduction-to-naive-bayes-23538a0395a

https://medium.com/squad-engineering/one-class-classification-for-images-with-deep-features-69182fb4c9