Predicting the Success of Baseball Pitchers

Yilmaz Machine Learning 1 - 2022-2023

Wilson Chen and Bradley Cao

1 Introduction

1.1 Overview

For our project, we decided to look at a dataset on baseball pitchers. This dataset with 547 instances and 31 attributes contained information on different statistics recorded for pitchers. 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

We got our original dataset from REPLACE. It contained 31 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**
- **n_fastball_formatted**
- **fastball_avg_speed**
- **n_offspeed_formatted**

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

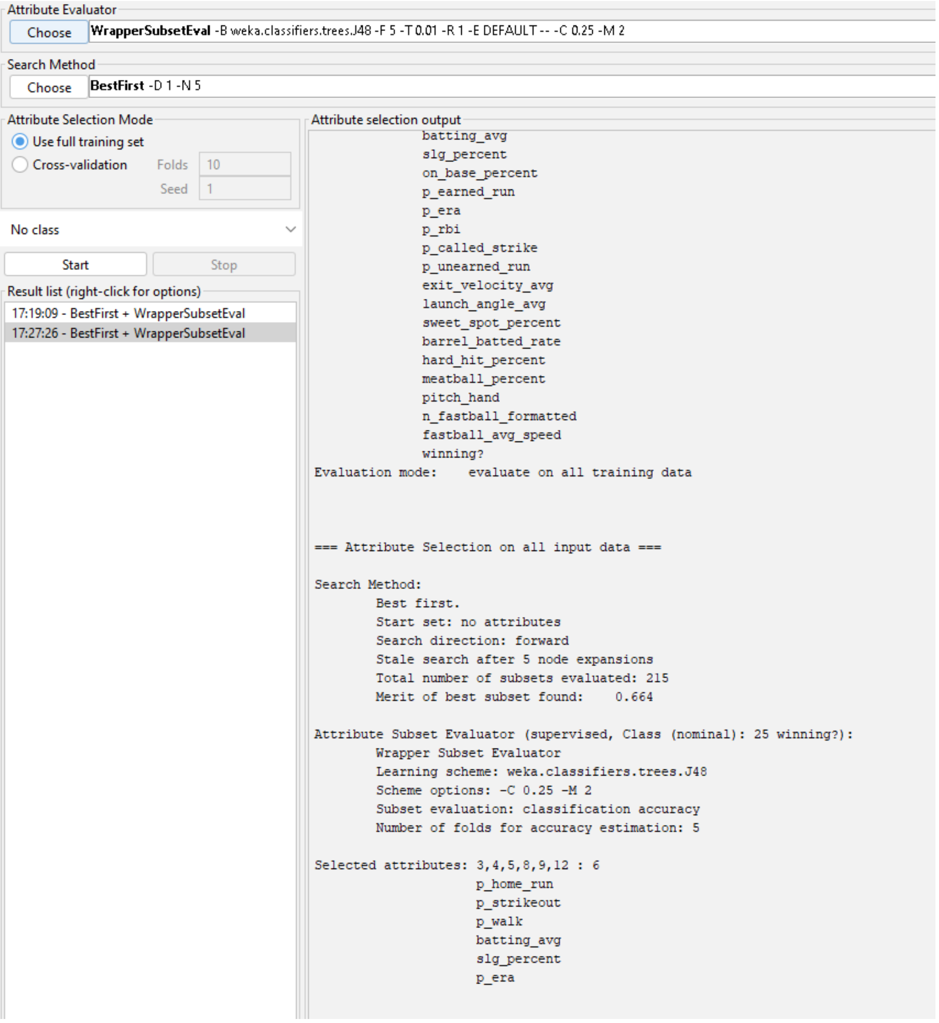
2 Preprocessing

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.).

2.2 Dimensionality Reduction

We used WEKA's **WrapperSubsetEval** technique with search method **BestFit** to perform further dimensionality reduction. As shown in the screenshot below, the attibutes that were kept were **p_home_run** (number of home runs given up), **p_strikeout** (number of strikeouts), **p_walk** (number of walks), **batting_avg** (batting average of opposing batters), **slg_percent** (slugging percent of opposing batters), and **p_era** (earned run average).



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

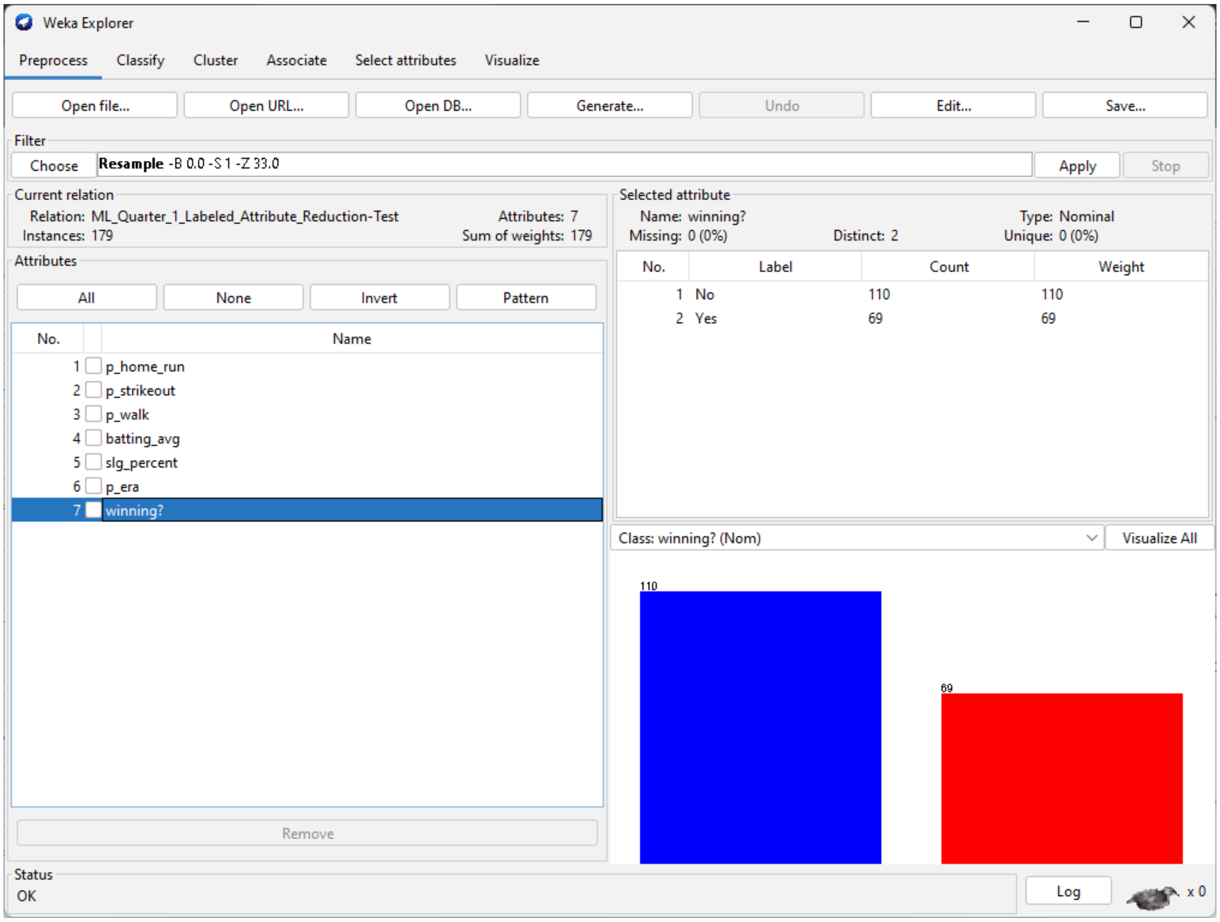
Attribute selection output

```
              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 Dimensionality Reduction results

3 Splitting into Testing and Training

After reducing dimensionality to these 7 attributes, we used WEKA to split our data into training and testing sets. We used WEKA's supervised method **Resample**, with sample size percent of 33% and **noReplacement** set to **True**. We ran it twice, once with **invertSelection** set to **False** to obtain our testing set. The second time, we ran it with **invertSelection** set to **True** to obtain our training set.



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

Attributes

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

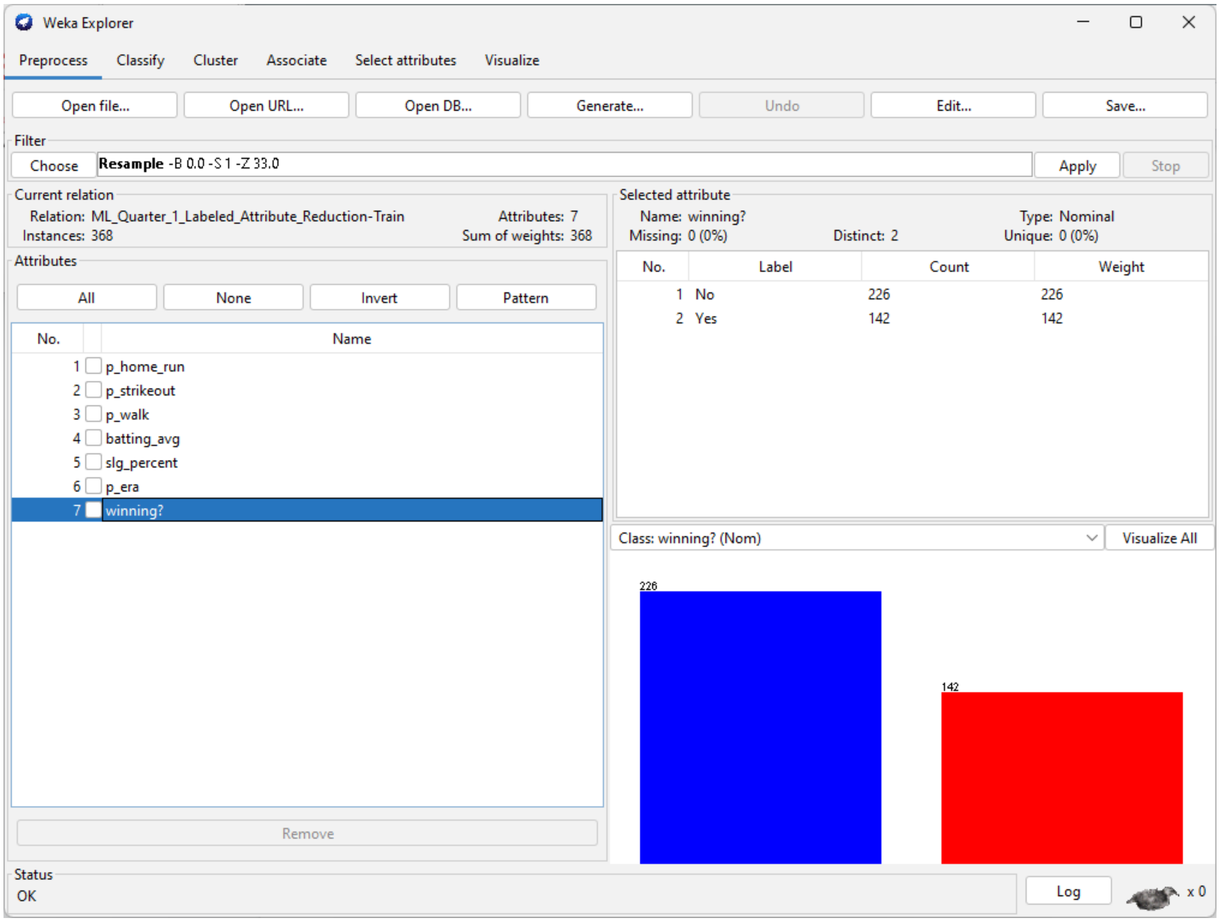
Selected attribute
Name: winning?
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)

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

Class: winning? (Nom)

Status
OK

Figure 2: WEKA Screenshot showing the Testing Set



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

Attributes

No.		Name
1	□	p_home_run
2	□	p_strikeout
3	□	p_walk
4	□	batting_avg
5	□	slg_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 3: WEKA Screenshot showing the Training Set

The class distribution of the testing and training sets can be seen in Figures 2 and 3. As seen by the class distributions, the WEKA **Resample** method has maintained the non-uniform class distribution in the training and testing sets. This means that the

4 Classification

5 Discussion

6 Conclusion

7 Sources

We got our data from https://www.mlb.com and used WEKA to preprocess and do classification work.