1. **The problem statements**

Can we predict if a horse can win in a horse racing game?

1. **Data sources**

https://www.kaggle.com/datasets/gdaley/hkracing

**races.csv**

Each line describes the condition of an individual race.

**race\_id** - unique identifier for the race

**date** - date of the race, in YYYY-MM-DD format (note that the dates given have been obscured and are not the real ones, although the durations between each race should be correct)

**venue** - a 2-character string, representing which of the 2 race courses this race took place at: ST = Shatin, HV = Happy Valley

**race\_no** - race number of the race in the day's meeting

**config** - race track configuration, mostly related to the position of the inside rail. For more details, see the [HKJC website](http://www.hkjc.com/english/racinginfo/racing_course.htm).

**surface** - a number representing the type of race track surface: 1 = dirt, 0 = turf

**distance** - distance of the race, in metres

**going** - track condition. For more details, see the [HKJC website](http://www.hkjc.com/english/racinginfo/racing_course.htm).

**horse\_ratings** - the range of horse ratings that may participate in this race

**prize** - the winning prize, in HK Dollars

**race\_class** - a number representing the class of the race

**sec\_time1** - time taken by the leader of the race to reach the end of the end of the 1st sectional point (sec)

**sec\_time2** - time taken by the leader of the race to reach the end of the 2nd sectional point (sec)

**sec\_time3** - time taken by the leader of the race to reach the end of the 3rd sectional point (sec)

**sec\_time4** - time taken by the leader of the race to reach the end of the 4th sectional point, if any (sec)

**sec\_time5** - time taken by the leader of the race to reach the end of the 5th sectional point, if any (sec)

**sec\_time6** - time taken by the leader of the race to reach the end of the fourth sectional point, if any (sec)

**sec\_time7** - time taken by the leader of the race to reach the end of the fourth sectional point, if any (sec)

**time1** - time taken by the leader of the race in the 1st section only (sec)

**time2** - time taken by the leader of the race in the 2nd section only (sec)

**time3** - time taken by the leader of the race in the 3rd section only (sec)

**time4** - time taken by the leader of the race in the 4th section only, if any (sec)

**time5** - time taken by the leader of the race in the 5th section only, if any (sec)

**time6** - time taken by the leader of the race in the 6th section only, if any (sec)

**time7** - time taken by the leader of the race in the 7th section only, if any (sec)

**place\_combination1** - placing horse no (1st)

**place\_combination2** - placing horse no (2nd)

**place\_combination3** - placing horse no (3rd)

**place\_combination4** - placing horse no (4th)

**place\_dividend1** - placing dividend paid (for place\_combination1)

**place\_dividend2** - placing dividend paid (for place\_combination2)

**place\_dividend3** - placing dividend paid (for place\_combination2)

**place\_dividend4** - placing dividend paid (for place\_combination2)

**win\_combination1** - winning horse no

**win\_dividend1** - winning dividend paid (for win\_combination1)

**win\_combination2** - joint winning horse no, if any

**win\_dividend2** - winning dividend paid (for win\_combination2, if any)

### runs.csv

Each line describes the characteristics of one horse run, in one of the races given in races.csv.

**race\_id** - unique identifier for the race

**horse\_no** - the number assigned to this horse, in the race

**horse\_id** - unique identifier for this horse

**result** - finishing position of this horse in the race

**won** - whether horse won (1) or otherwise (0)

**lengths\_behind** - finishing position, as the number of horse lengths behind the winner

**horse\_age** - current age of this horse at the time of the race

**horse\_country** - country of origin of this horse

**horse\_type** - sex of the horse, e.g. 'Gelding', 'Mare', 'Horse', 'Rig', 'Colt', 'Filly'

**horse\_rating** - rating number assigned by HKJC to this horse at the time of the race

**horse\_gear** - string representing the gear carried by the horse in the race. An explanation of the codes used may be found on the [HKJC website](http://racing.hkjc.com/racing/Info/meeting/RaceCard/english/).

**declared\_weight** - declared weight of the horse and jockey, in lbs

**actual\_weight** - actual weight carried by the horse, in lbs

**draw** - post position number of the horse in this race

**position\_sec1** - position of this horse (ranking) in section 1 of the race

**position\_sec2** - position of this horse (ranking) in section 2 of the race

**position\_sec3** - position of this horse (ranking) in section 3 of the race

**position\_sec4** - position of this horse (ranking) in section 4 of the race, if any

**position\_sec5** - position of this horse (ranking) in section 5 of the race, if any

**position\_sec6** - position of this horse (ranking) in section 6 of the race, if any

**behind\_sec1** - position of this horse (lengths behind leader) in section 1 of the race

**behind\_sec2** - position of this horse (lengths behind leader) in section 2 of the race

**behind\_sec3** - position of this horse (lengths behind leader) in section 3 of the race

**behind\_sec4** - position of this horse (lengths behind leader) in section 4 of the race, if any

**behind\_sec5** - position of this horse (lengths behind leader) in section 5 of the race, if any

**behind\_sec6** - position of this horse (lengths behind leader) in section 6 of the race, if any

**time1** - time taken by the horse to pass through the 1st section of the race (sec)

**time2** - time taken by the horse to pass through the 2nd section of the race (sec)

**time3** - time taken by the horse to pass through the 3rd section of the race (sec)

**time4** - time taken by the horse to pass through the 4th section of the race, if any (sec)

**time5** - time taken by the horse to pass through the 5th section of the race, if any (sec)

**time6** - time taken by the horse to pass through the 6th section of the race, if any (sec)

**finish\_time** - finishing time of the horse in this race (sec)

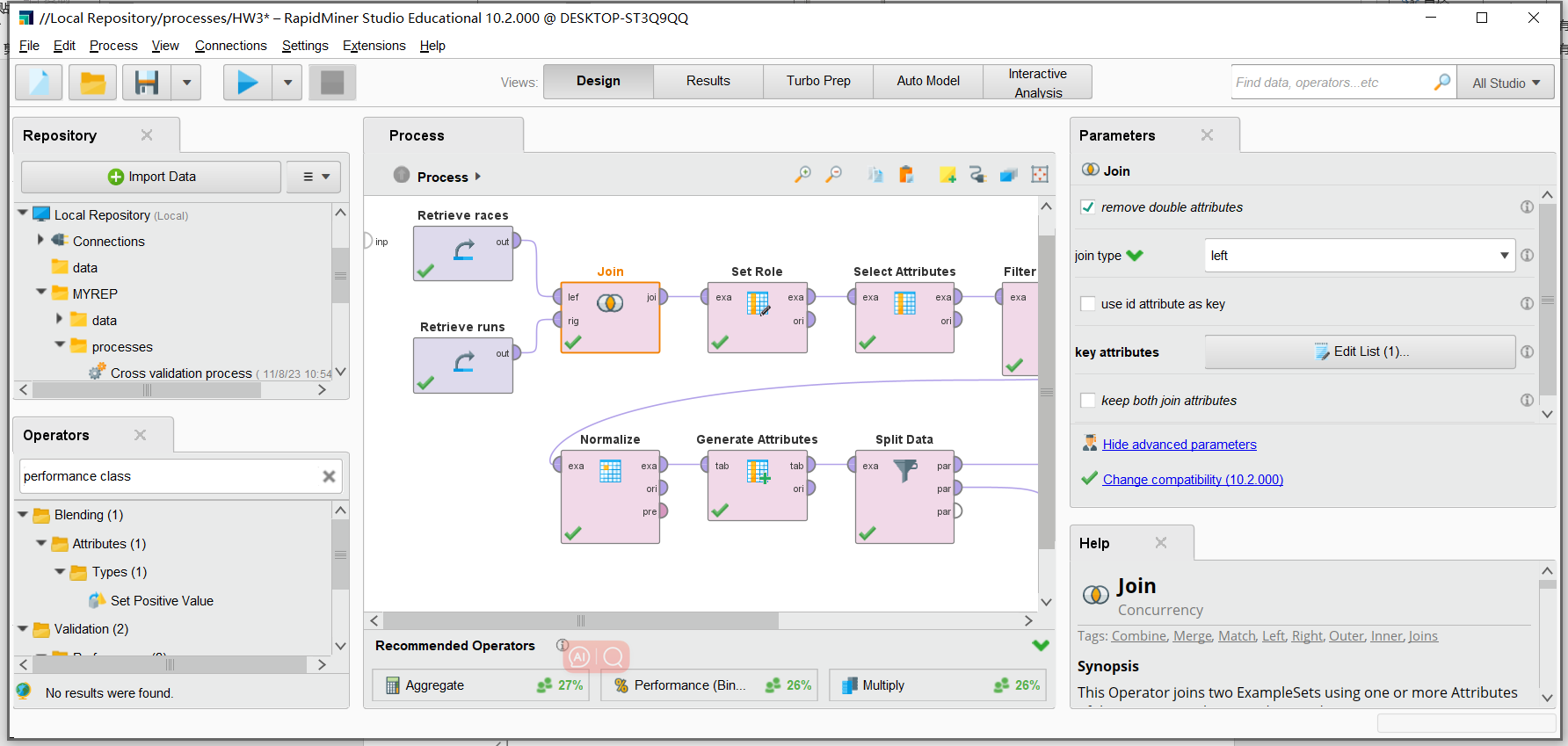
**win\_odds** - win odds for this horse at start of race

**place\_odds** - place (finishing in 1st, 2nd or 3rd position) odds for this horse at start of race

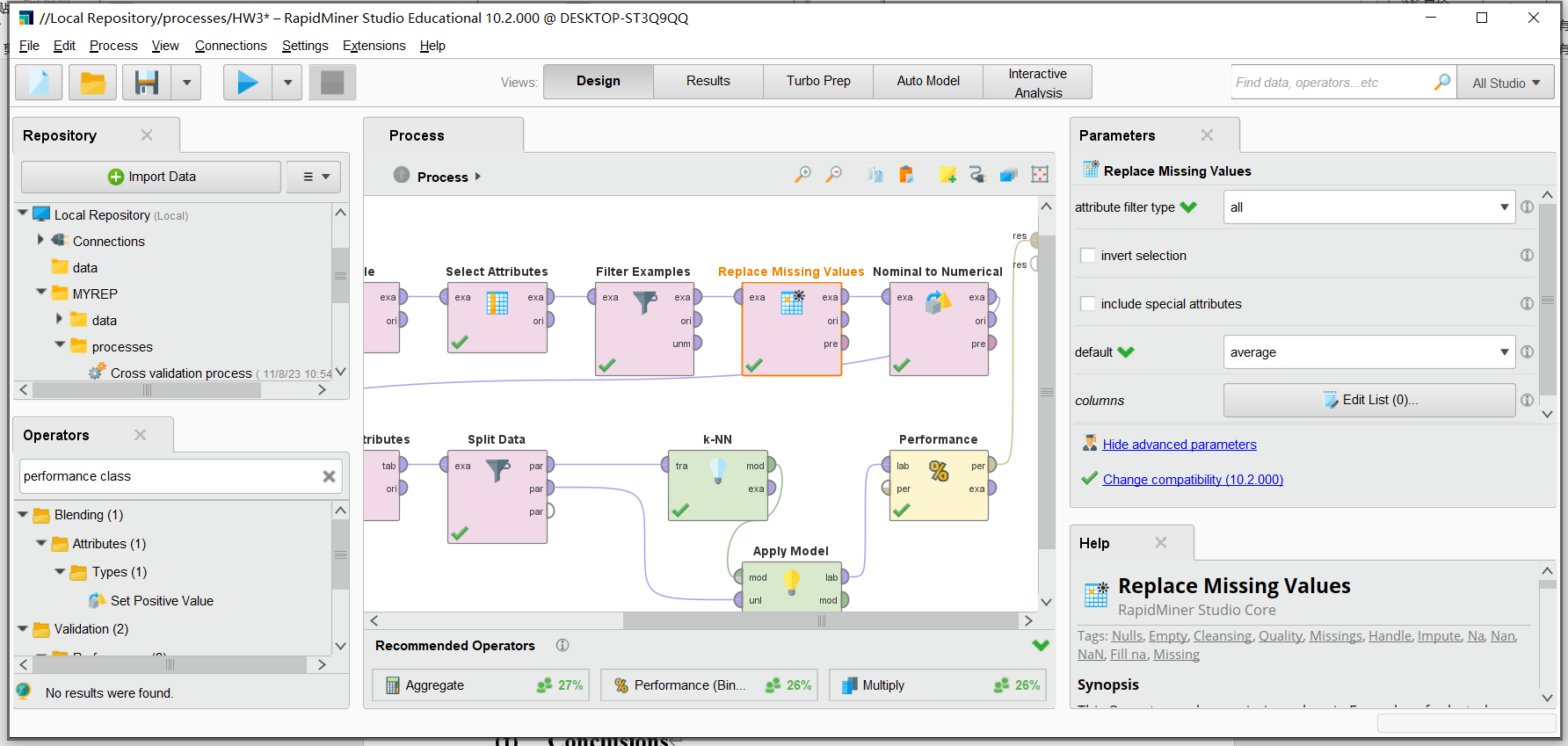
**trainer\_id** - unique identifier of the horse's trainer at the time of the race

**jockey\_id** - unique identifier of the jockey riding the horse in this race

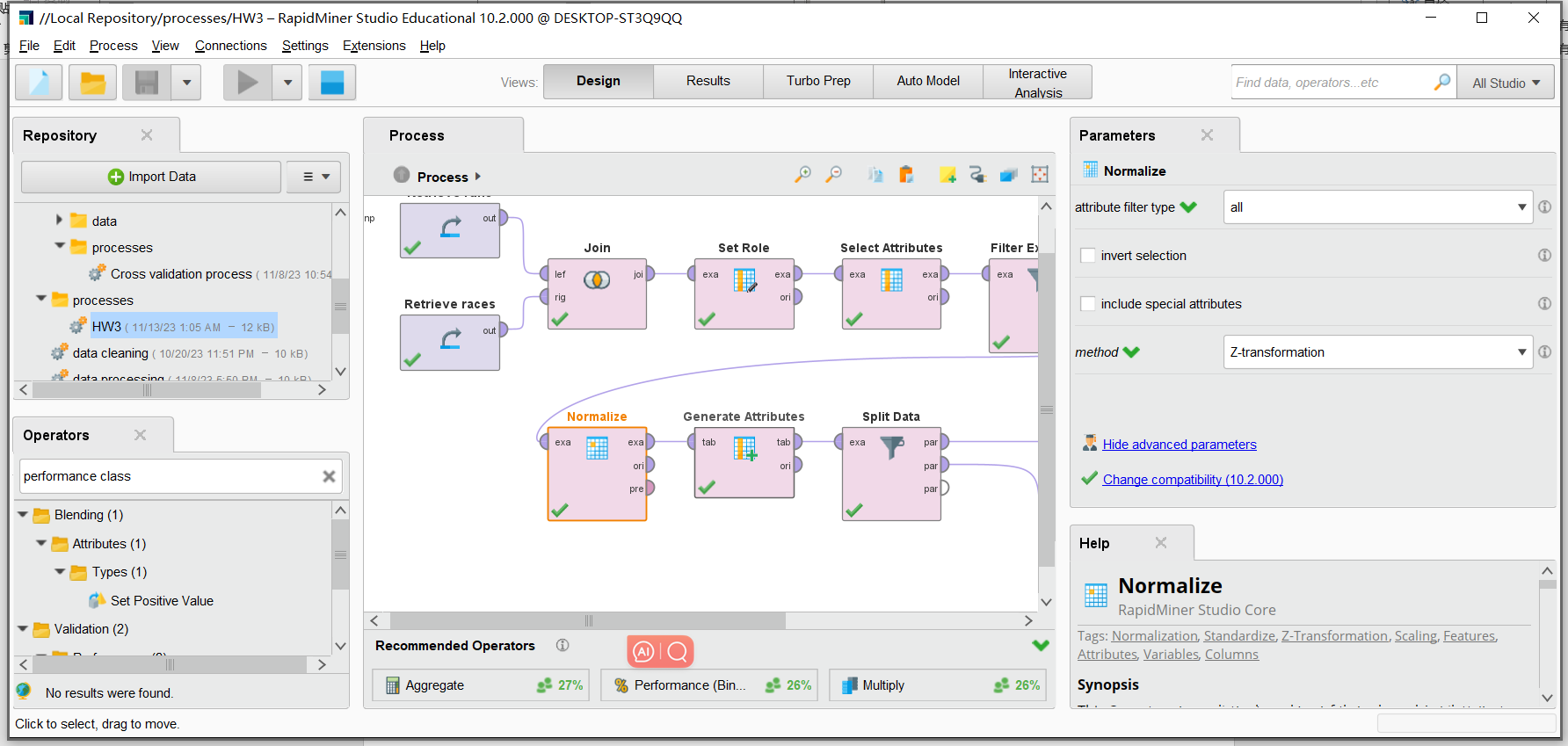
1. **Feature engineering and preprocessing done**
2. **Left join two source datasets with “race\_id” as foreign key**
3. **Set “won” as the label**
4. **Select features we want**



1. **Drop/Replace missing values with average**
2. **Convert Nominal values into Numerical values**



1. **Normalize values with Z-score**
2. **7:3 Train-Test Split**

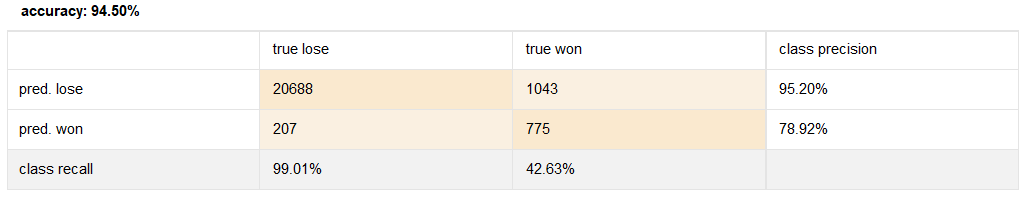


1. **Results from all the models in the form of a table**

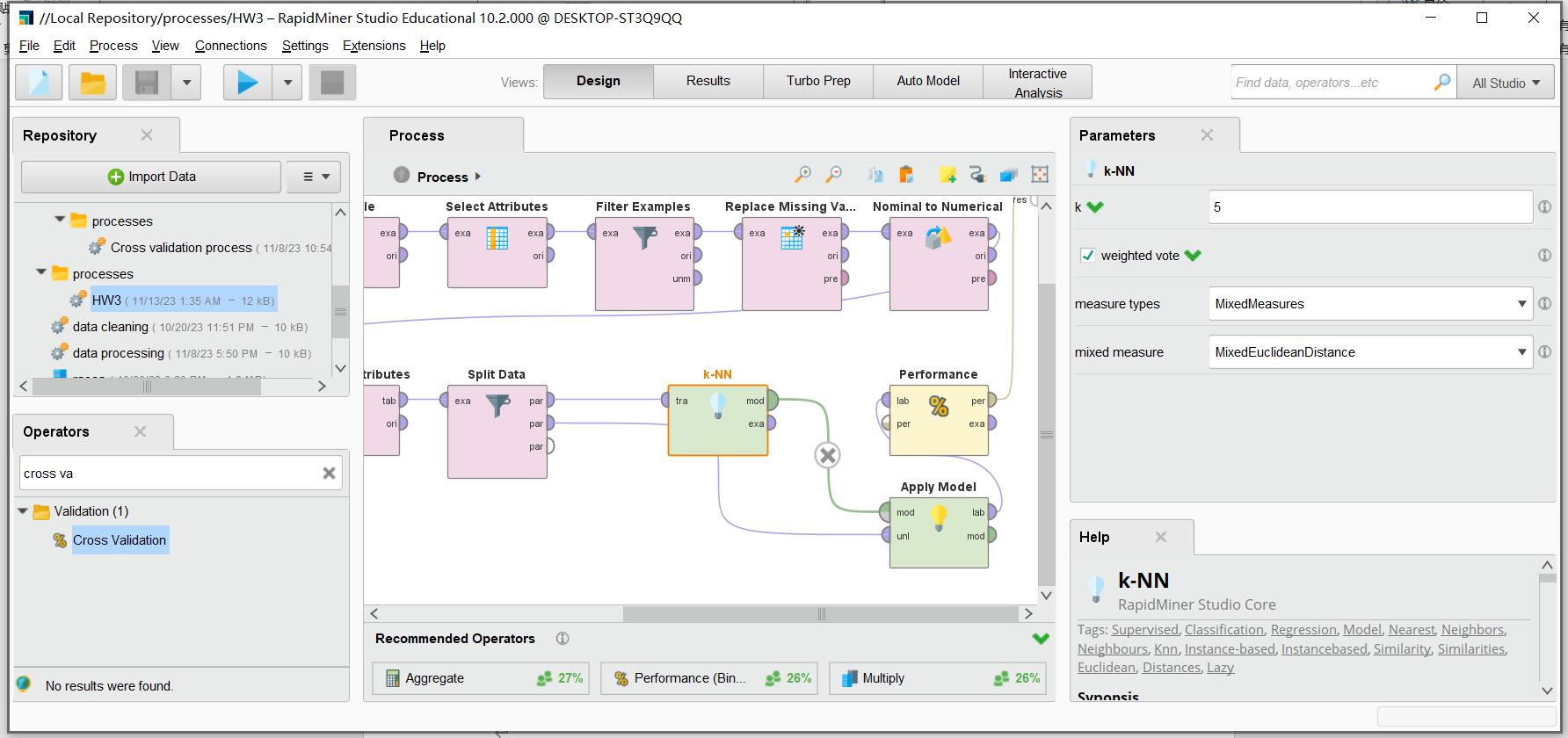
**1.KNN**

**KNN in Python from scratch:** KNN.ipynb

**KNN in RapidMiner:**



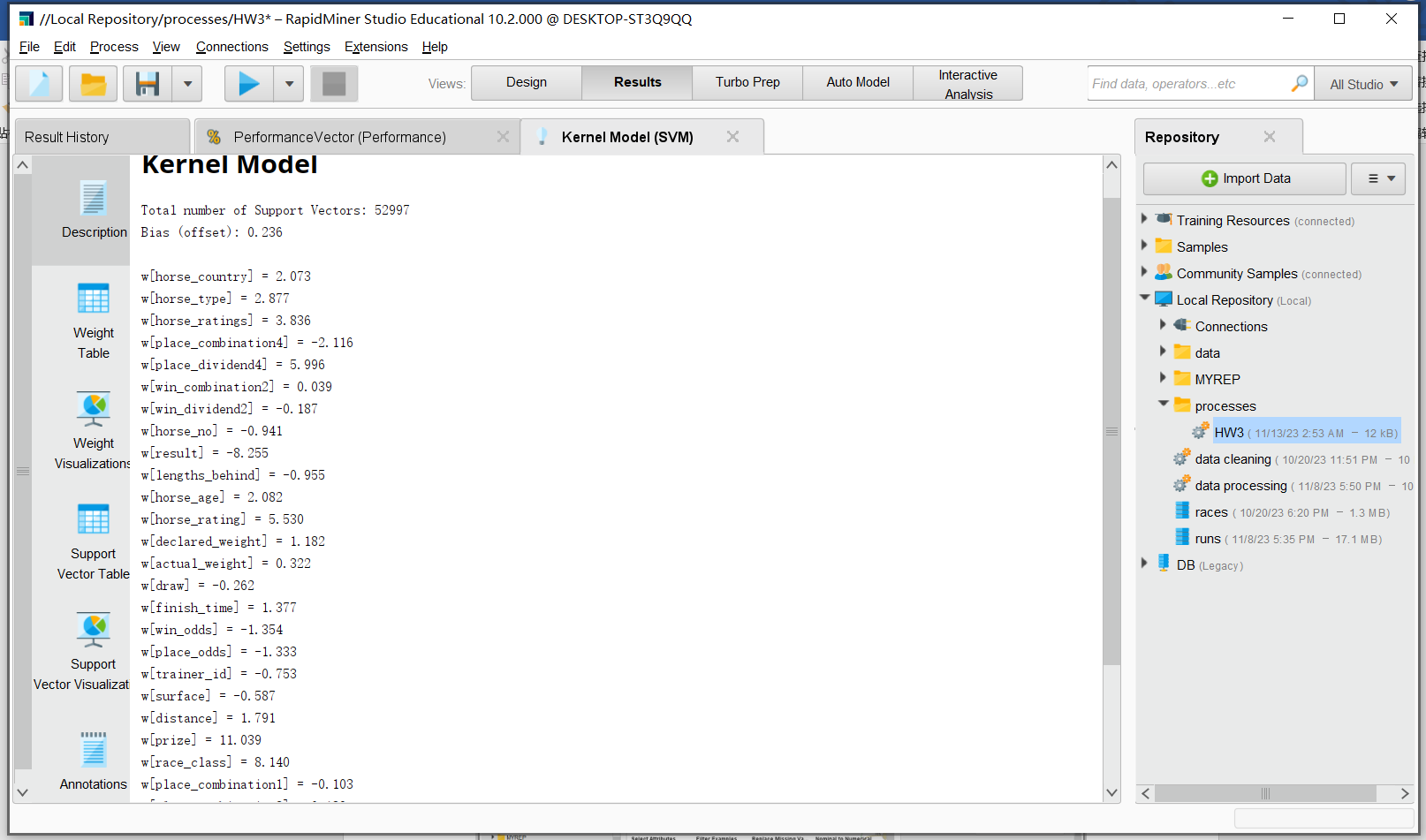
**Hyperparameters:**

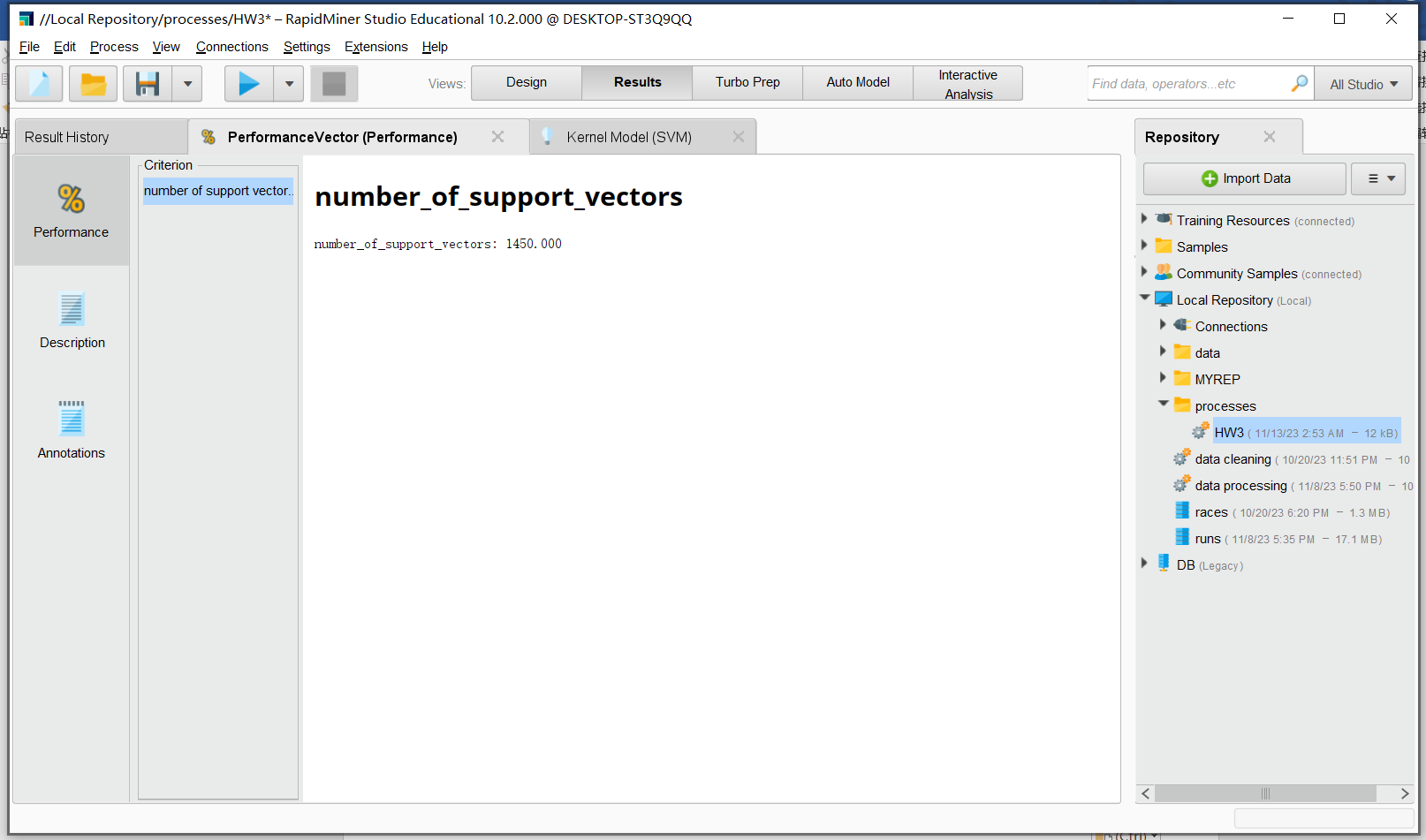


**Pros:** The K-Nearest Neighbors (KNN) algorithm is beneficial for its ease of understanding and implementation, and it can be very accurate if the dataset has enough representative data points.

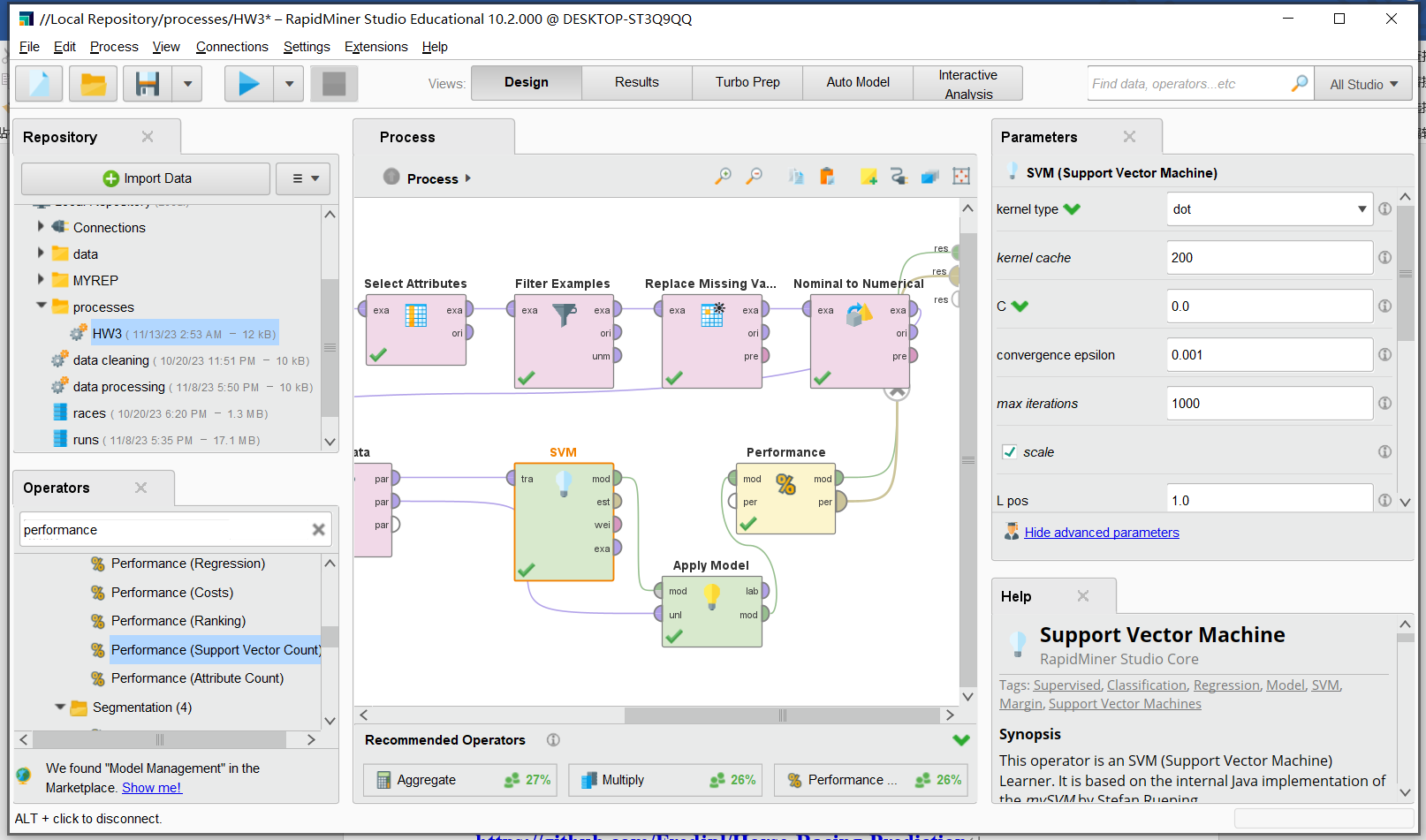
**Cons:** However, it suffers from high computation time during the testing phase and is not suitable for large datasets or datasets with many dimensions due to the 'curse of dimensionality'.

**2. SVM**





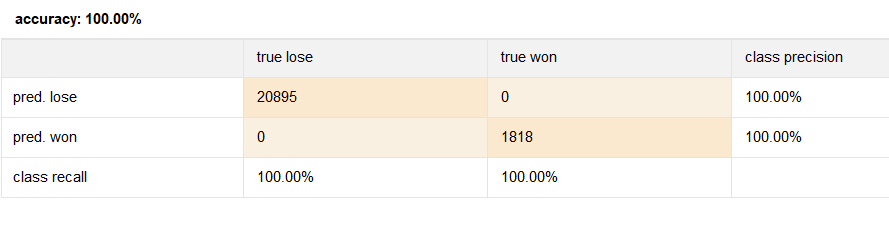
**Hyperparameters:**



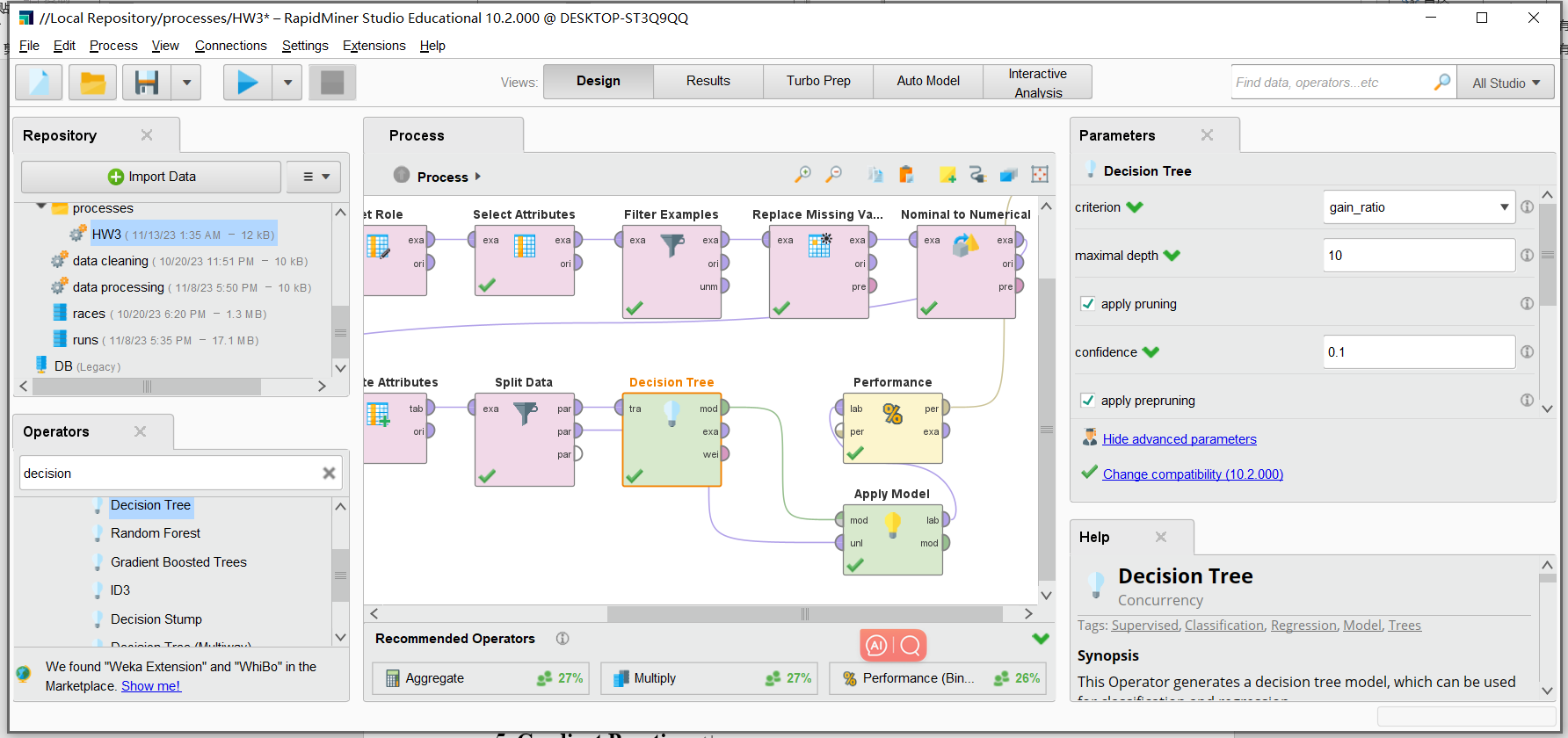
**Pros:** Support Vector Machines (SVM) are effective in high-dimensional spaces and in cases where the number of dimensions exceeds the number of samples, making them powerful for complex classification problems.

**Cons:** However, they can be memory-intensive, difficult to tune due to the importance of choosing the right kernel, and their results are less interpretable than simpler models.

**3. Decision Tree**



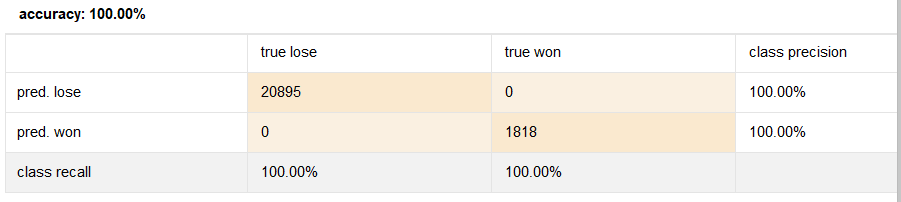
**Hyperparameters:**



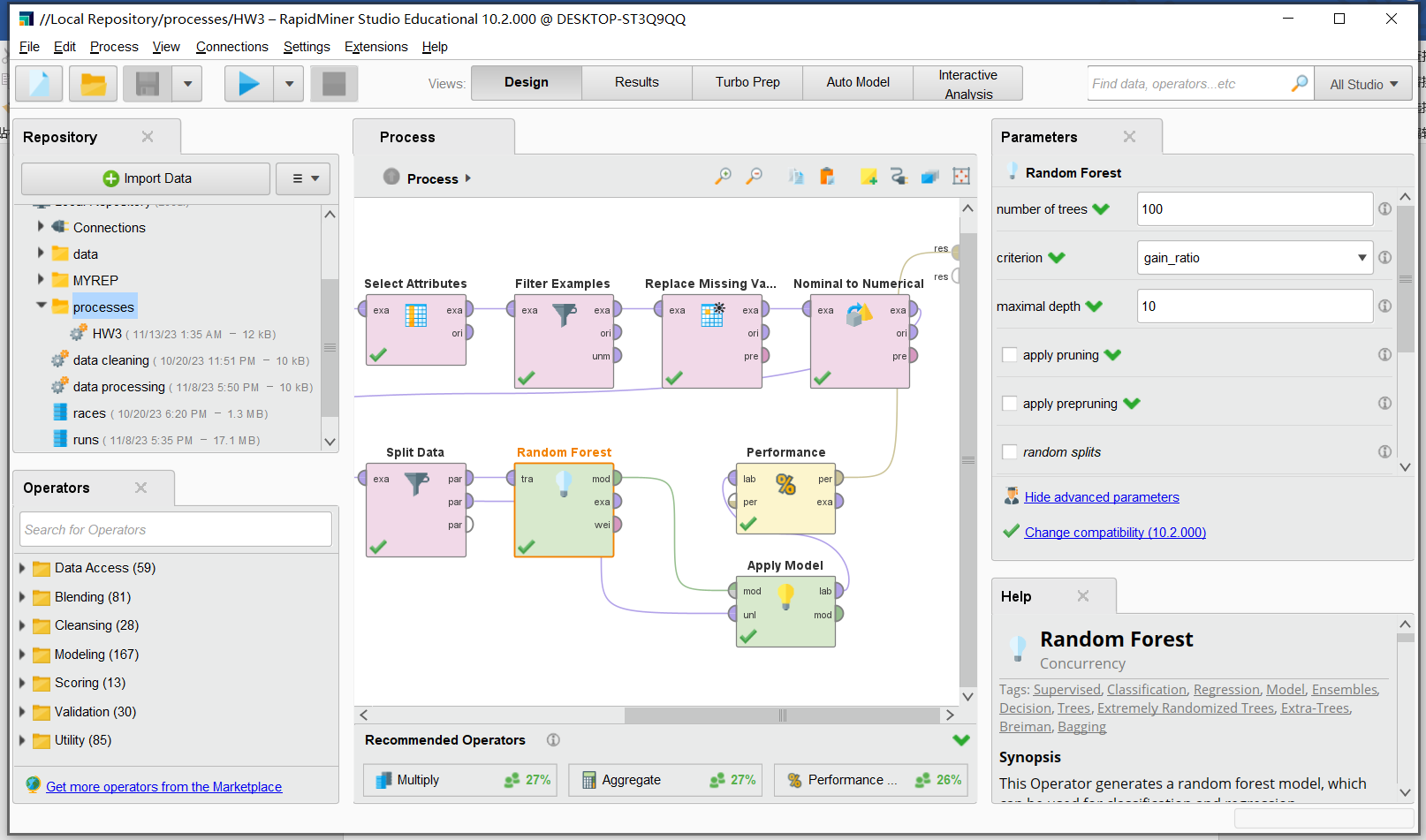
**Pros:** Decision Trees are straightforward to understand and interpret, and they require little data preparation compared to other algorithms.

**Cons:** However, they can create overly complex trees that do not generalize well from the training data, known as overfitting, and can be biased if some classes dominate.

**4. Random Forest**



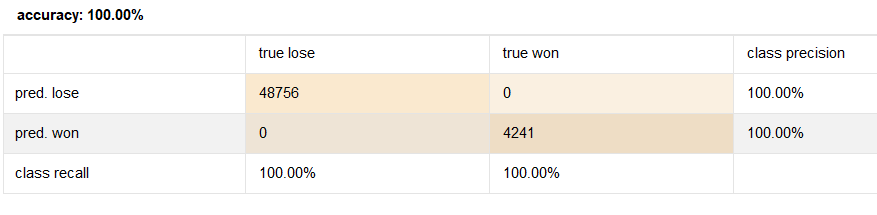
**Hyperparameters:**



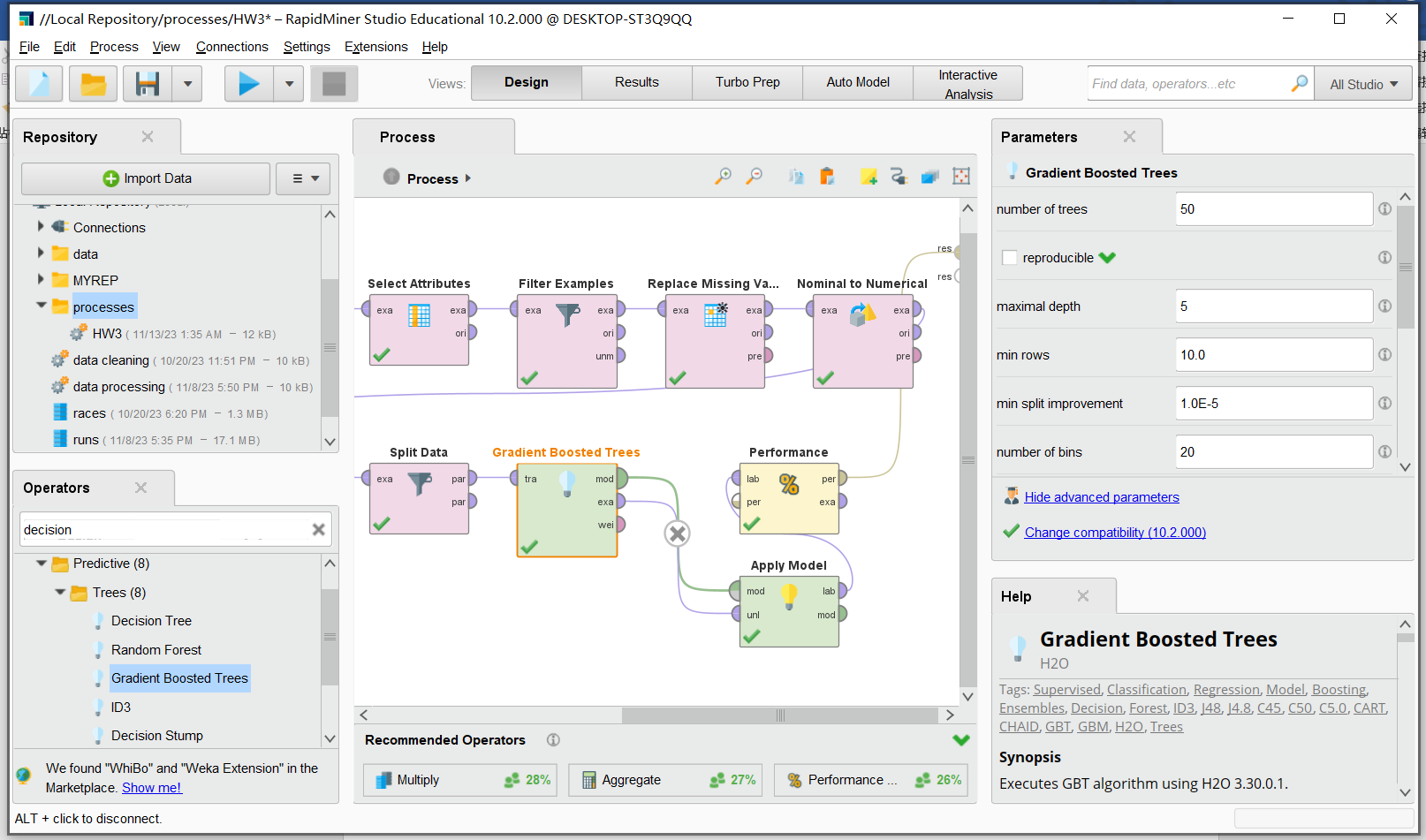
**Pros:** Random Forest is a versatile and powerful classification method that generally performs well on a wide range of problems, offering benefits such as handling large data sets with higher dimensionality and providing estimates of feature importance.

**Cons:** However, it can be computationally expensive to train, may overfit on some noise or unbalanced datasets, and the model outcomes are less interpretable compared to decision trees.

**5. Gradient Boosting**



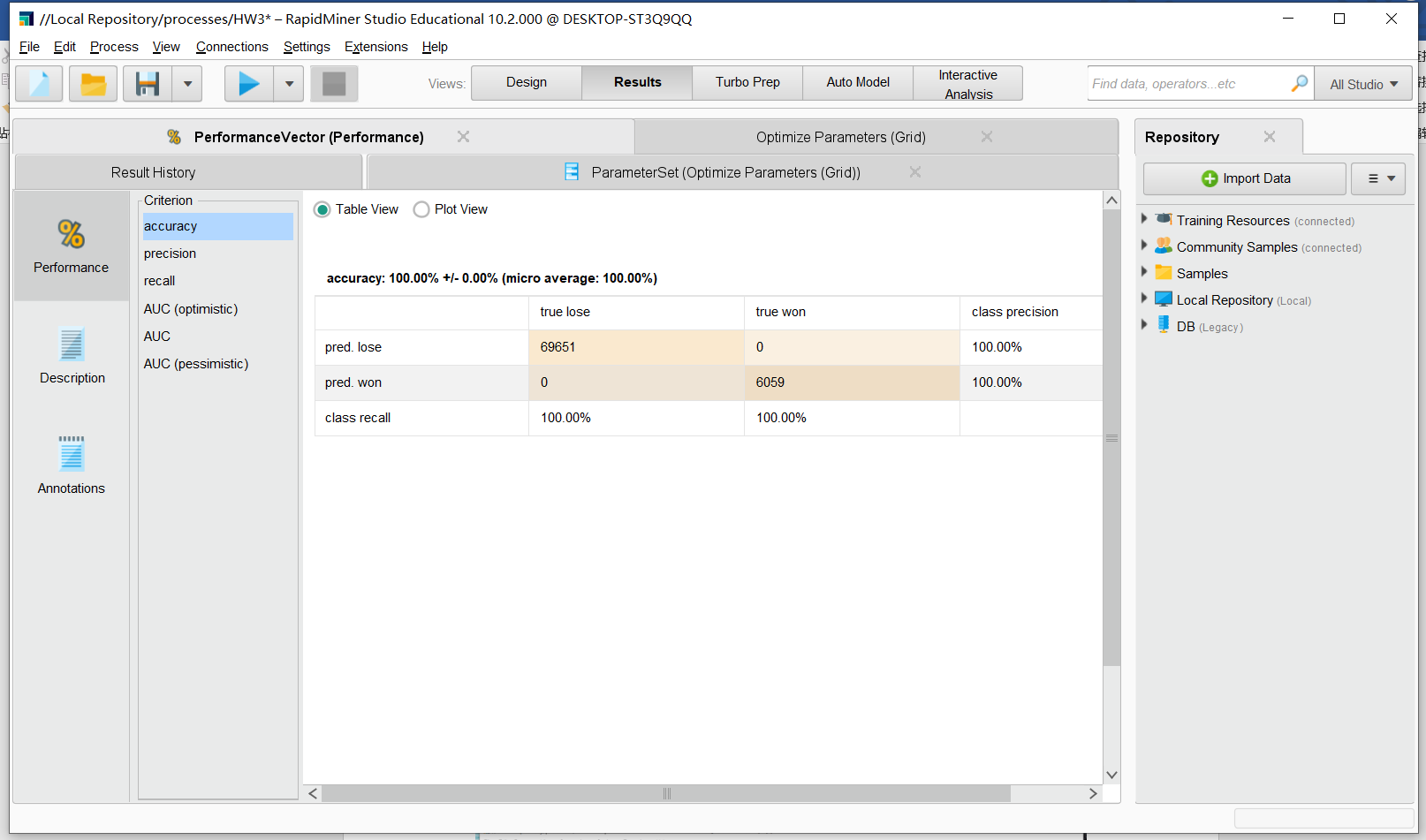
**Hyperparameters:**

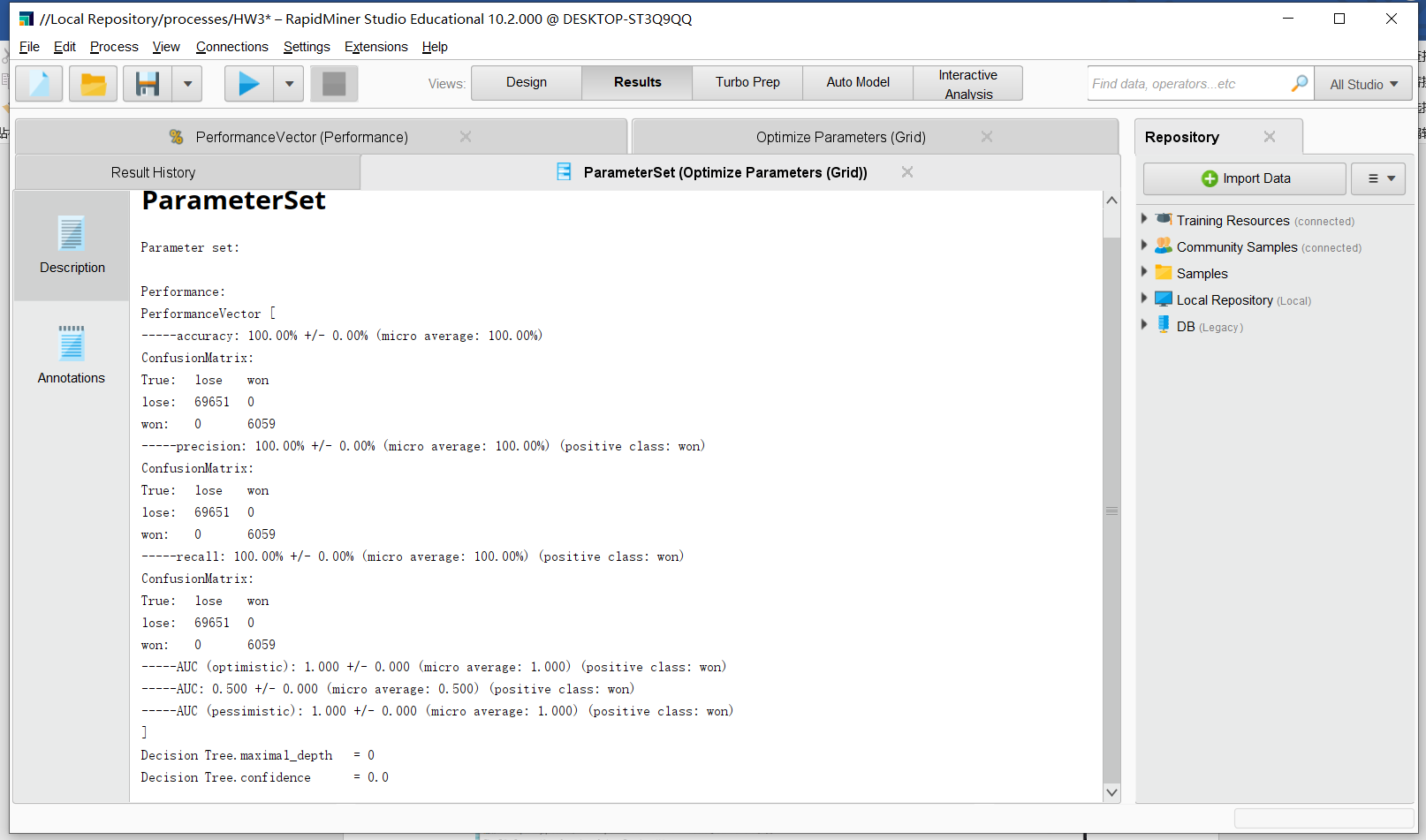


**Pros:** Gradient Boosting is a powerful ensemble technique known for its high accuracy and effectiveness in handling various types of data, as well as its flexibility in optimization.

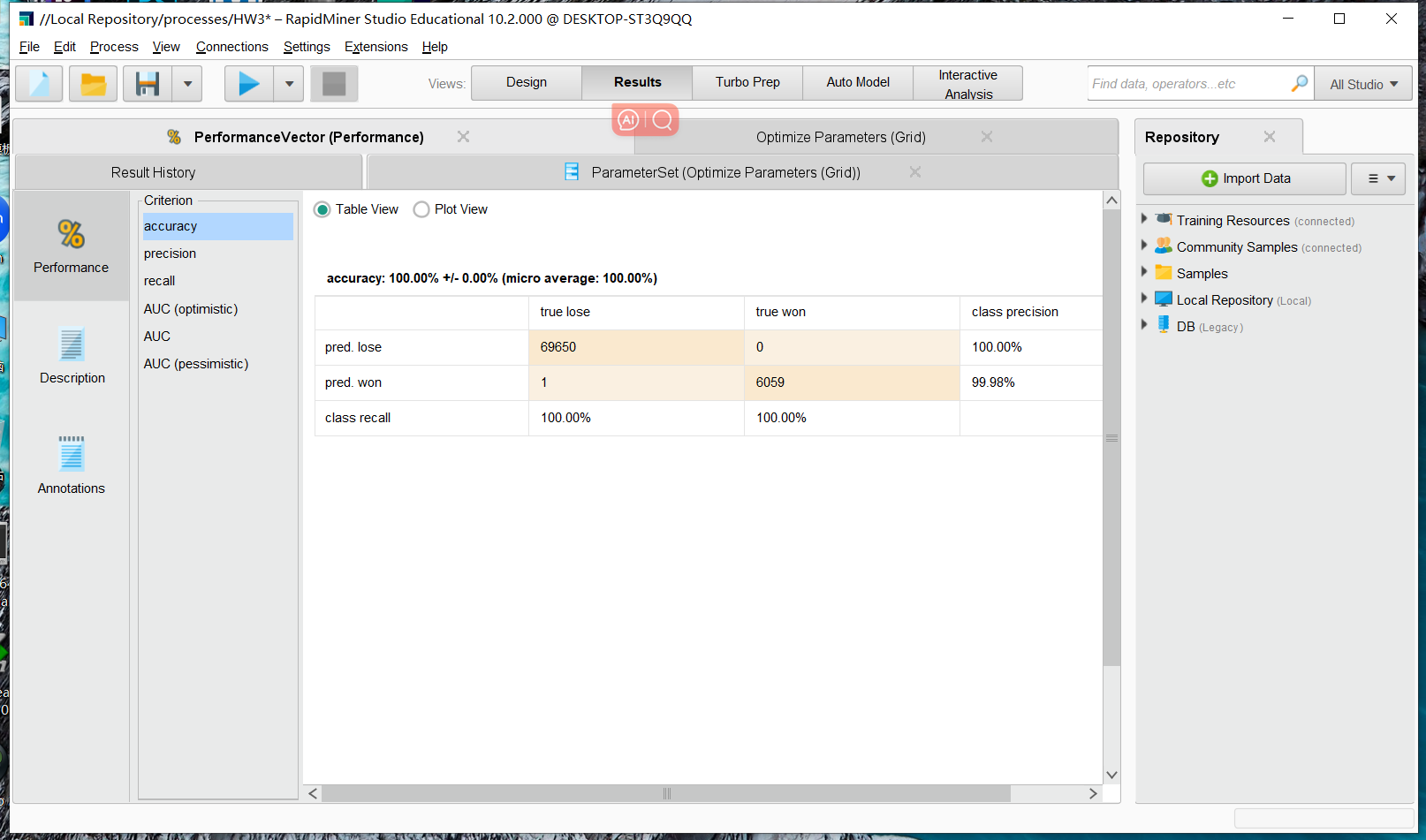
**Cons:** However, it can be prone to overfitting if not tuned properly, and the sequential nature of boosting can result in longer training times compared to algorithms that allow parallel processing.

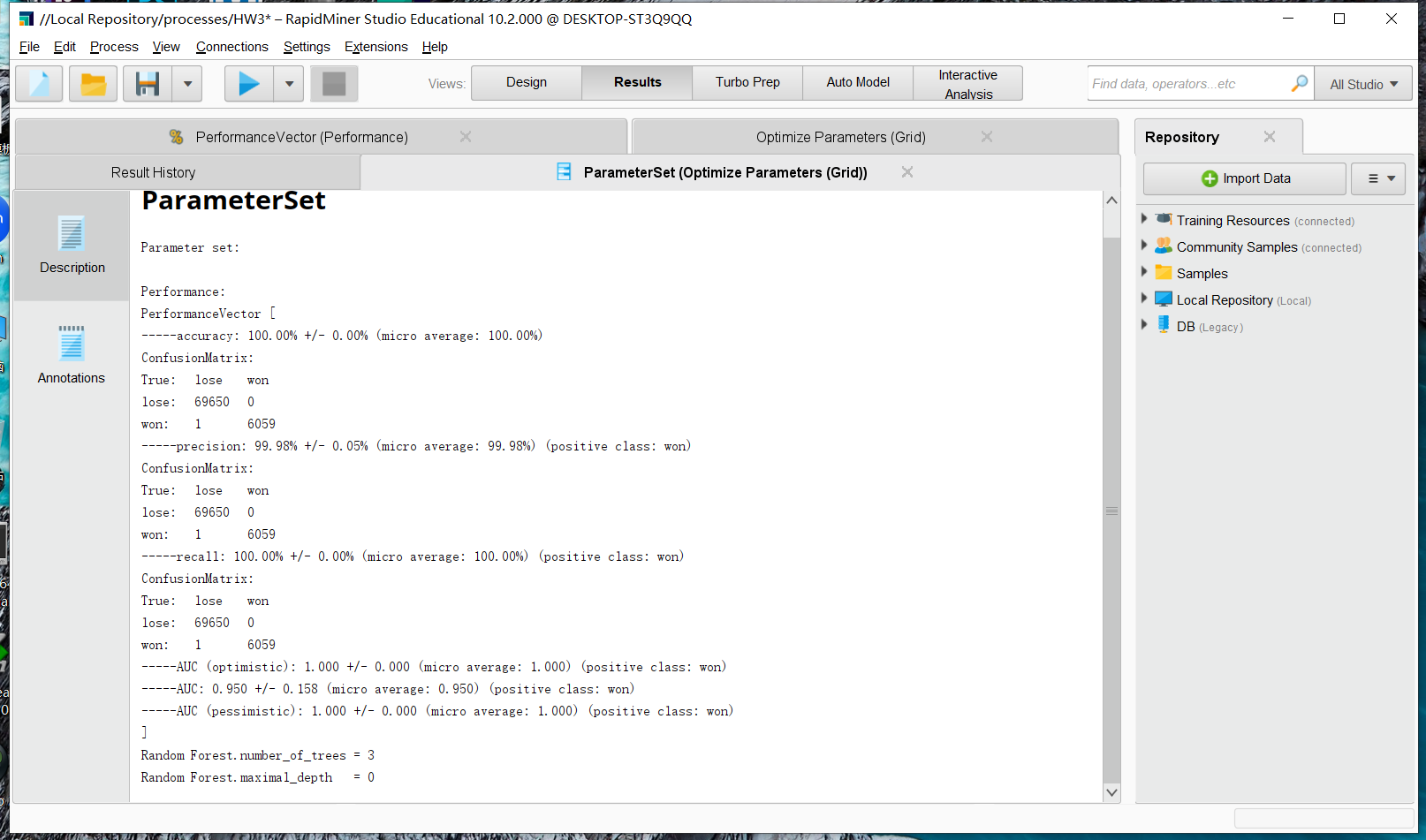
1. **Results from the hyperparameter search**
2. **Decision Tree:**





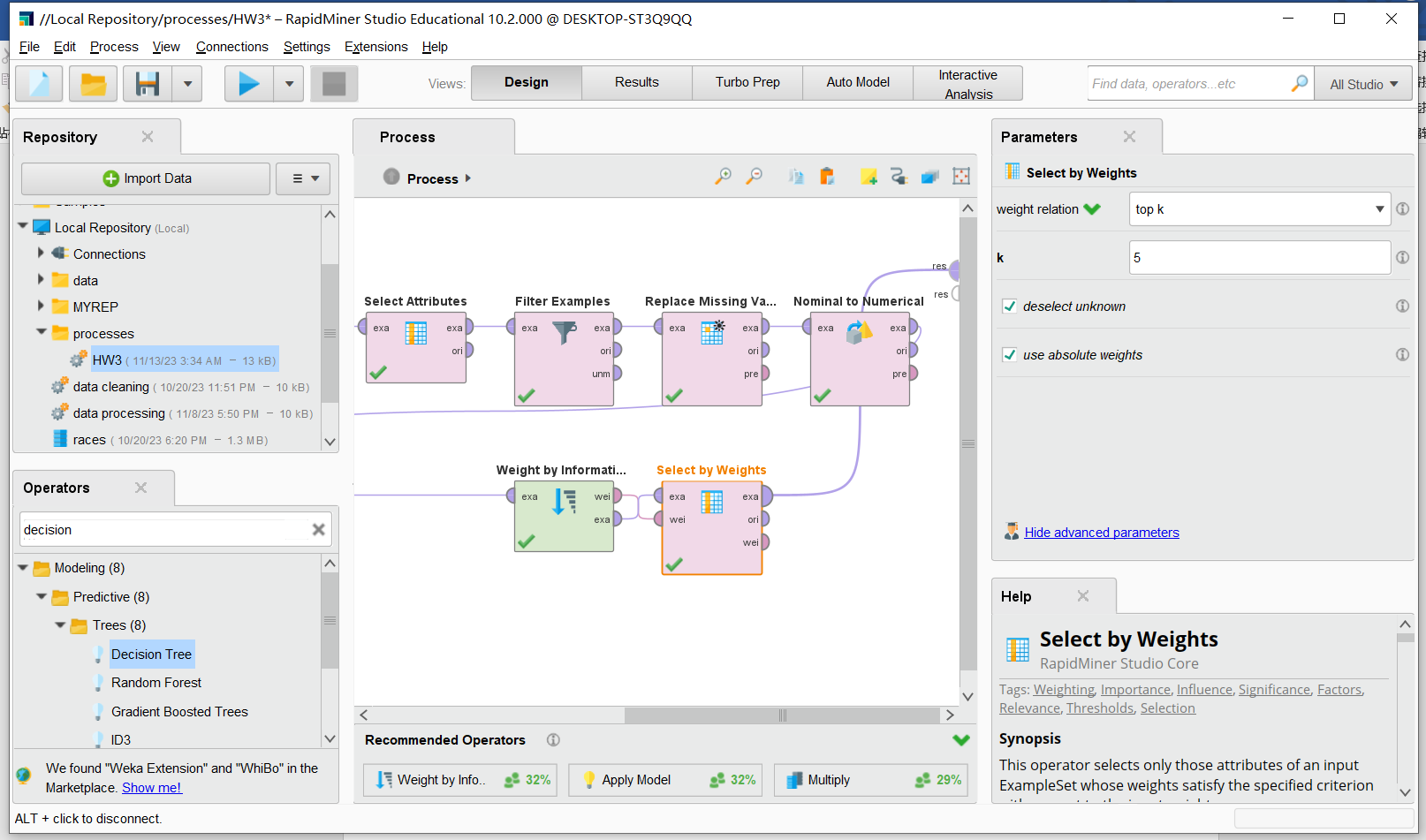
1. **Random Forest**

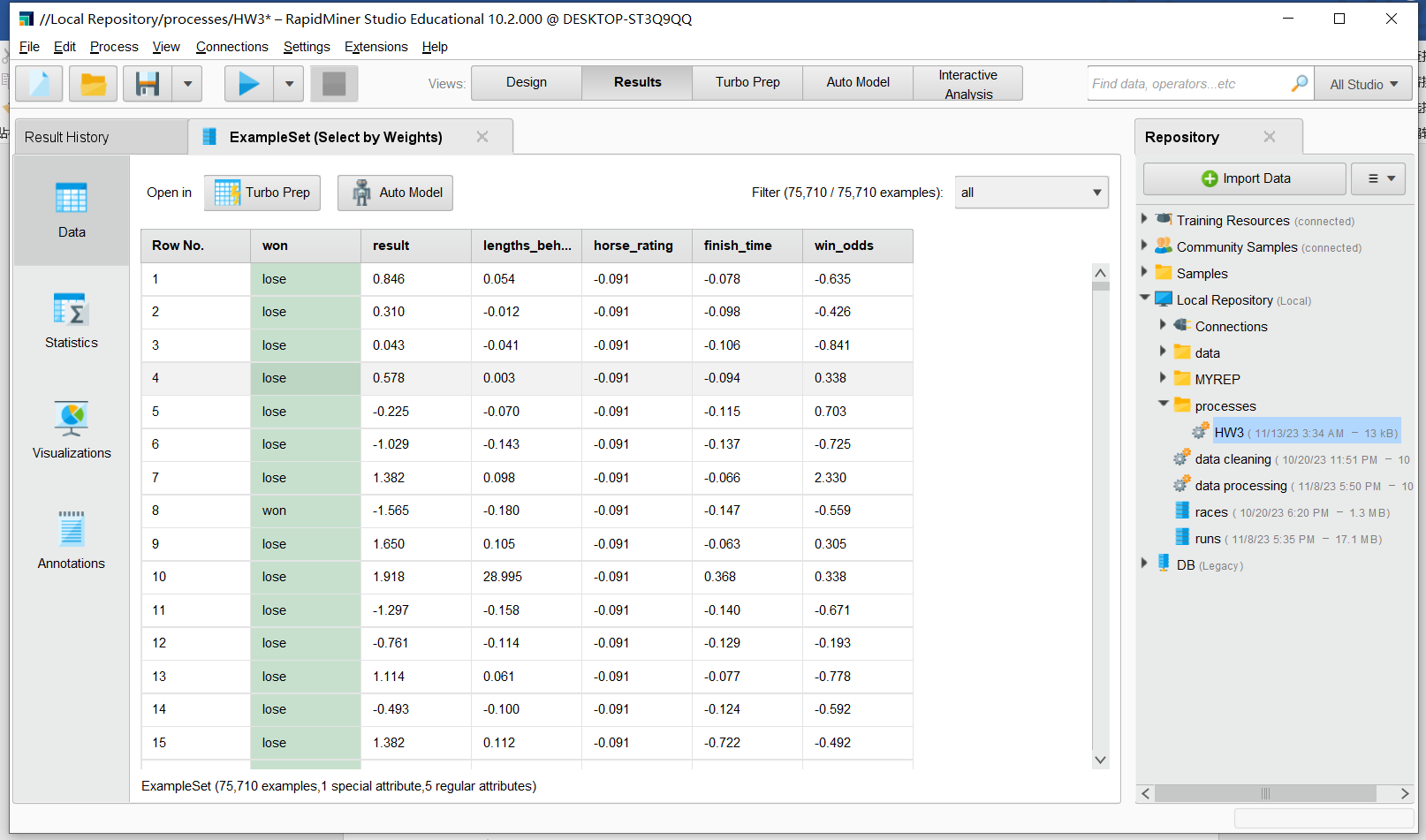




1. **Conclusions**

Due to the high accuracy of each model, I think they can be used to answer my initial question present. Here are top 5 the most important features used in the classification:





1. **Link to the GitHub repo**

[**https://github.com/Fredjpl/Horse-Racing-Prediction**](https://github.com/Fredjpl/Horse-Racing-Prediction)