Data Mining

PROJE RAPORU

Raporu Hazırlayan:	Oğuz Kaan Satan
Ödev Yapanlar:	Oğuz Satan Kaan
Ödev Tarihi:	18.01.2021

Data Set Information

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In [Cortez and Silva, 2008], the two datasets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful (see paper source for more details).

Data Set Characteristics:	Multivariate	**Number of Instances:**	649	**Area:**	Social
Attribute Characteristics:	Integer	**Number of Attributes:**	33	**Date Donated**	2014-11-27
Associated Tasks:	Classification, Regression	**Missing Values?**	N/A	**Number of Web Hits:**	857825

Attribute Information:

Attributes for both student-mat.csv (Math course)

1) school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)

2) sex - student's sex (binary: 'F' - female or 'M' - male)

3) age - student's age (numeric: from 15 to 22)

4) address - student's home address type (binary: 'U' - urban or 'R' - rural)

5) famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)

6) Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)

7) Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)

8) Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 â€“ 5th to 9th grade, 3 â€“ secondary education or 4 â€“ higher education)

9) Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')

10) Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')

11) reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')

12) guardian - student's guardian (nominal: 'mother', 'father' or 'other')

13) traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)

14) studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)

15)failures - number of past class failures (numeric: n if 1<=n<3, else 4)

16) schoolsup - extra educational support (binary: yes or no)

17) famsup - family educational support (binary: yes or no)

18) paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)

19) activities - extra-curricular activities (binary: yes or no)

20) nursery - attended nursery school (binary: yes or no)

21) higher - wants to take higher education (binary: yes or no)

22) internet - Internet access at home (binary: yes or no)

23)romantic - with a romantic relationship (binary: yes or no)

24) famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

25) freetime - free time after school (numeric: from 1 - very low to 5 - very high)

26) goout - going out with friends (numeric: from 1 - very low to 5 - very high)

27) Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)

28) Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)

29) health - current health status (numeric: from 1 - very bad to 5 - very good)

30)absences - number of school absences (numeric: from 0 to 93)

These grades are related with the course subject,

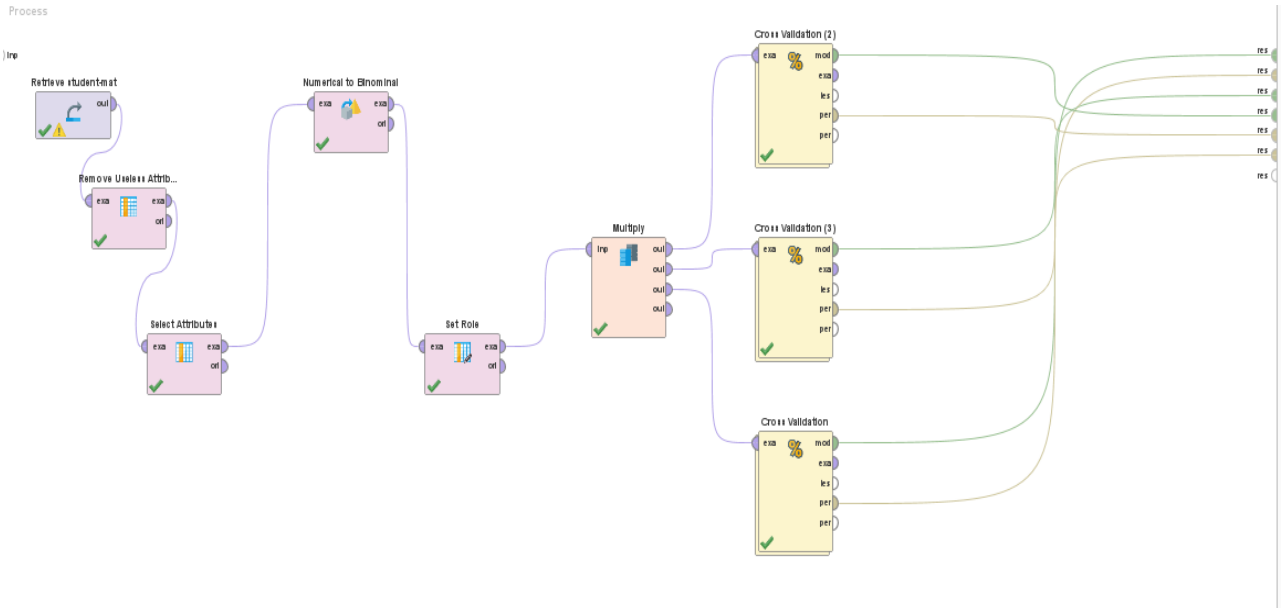
31 G1 - first period grade (numeric: from 0 to 20)

31 G2 - second period grade (numeric: from 0 to 20)

32 G3 - final grade (numeric: from 0 to 20, output target)

Preprocessing

First, I pulled the cvs data file I found from the internet into the design section using Retrieve. then, using the slit attributes operator, I deleted the rows I didn't need in my data so as not to process them, I want to find the student success from my data, that is, G1'yi numeric data binomial still translated the reason I did this G1'nin values as true and false I wanted to see. I then set the role as label because I had to show G1 as label, which determines student success using the set role operator. and by replicating the output from the set role operator by the multiply operator I have created 3 models that host cross validation operators connected. have made Simple Distribution and performance calculated by using my models contained in cross validation operators. The models I used were naive bayes, tree decision, and the adaboost algorithm. I also used performance classification to calculate performance in the continuation of these algorithms and I used the apply model operator to connect the performance and model to each other.



Classification

Decision Tree

Decision tree is a tree like collection of nodes intended to create a decision on values affiliation to a class or an estimate of a numerical target value. Each node represents a splitting rule for one specific Attribute. For classification this rule separates values belonging to different classes, for regression it separates them in order to reduce the error in an optimal way for the selected parameter *criterion*.

The building of new nodes is repeated until the stopping criteria are met. A prediction for the class label Attribute is determined depending on the majority of Examples which reached this leaf during generation, while an estimation for a numerical value is obtained by averaging the values in a leaf.

This Operator can process ExampleSets containing both nominal and numerical Attributes. The label Attribute must be nominal for classification and numerical for regression.

After generation, the decision tree model can be applied to new Examples using the Apply Model Operator. Each Example follows the branches of the tree in accordance to the splitting rule until a leaf is reached.

Naive Bayes

Naive Bayes is a high-bias, low-variance classifier, and it can build a good model even with a small data set. It is simple to use and computationally inexpensive. Typical use cases involve text categorization, including spam detection, sentiment analysis, and recommender systems.

The fundamental assumption of Naive Bayes is that, given the value of the label (the class), the value of any Attribute is independent of the value of any other Attribute. Strictly speaking, this assumption is rarely true (it's "naive"!), but experience shows that the Naive Bayes classifier often works well. The independence assumption vastly simplifies the calculations needed to build the Naive Bayes probability model.

To complete the probability model, it is necessary to make some assumption about the conditional probability distributions for the individual Attributes, given the class. This Operator uses Gaussian probability densities to model the Attribute data.

AdaBoost

The AdaBoost operator is a nested operator i.e. it has a subprocess. The subprocess must have a learner i.e. an operator that expects an ExampleSet and generates a model. This operator tries to build a better model using the learner provided in its subprocess. You need to have a basic understanding of subprocesses in order to apply this operator. Please study the documentation of the Subprocess operator for basic understanding of subprocesses.

AdaBoost, short for Adaptive Boosting, is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than most learning algorithms. The classifiers it uses can be weak (i.e., display a substantial error rate), but as long as their performance is not random (resulting in an error rate of 0.5 for binary classification), they will improve the final model.

AdaBoost generates and calls a new weak classifier in each of a series of rounds $t = 1,\ldots,T$. For each call, a distribution of weights $D(t)$ is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded correct classification.

Ensemble Theory Boosting is an ensemble method, therefore an overview of the Ensemble Theory has been discussed here. Ensemble methods use multiple models to obtain better predictive performance than could be obtained from any of the constituent models. In other words, an ensemble is a technique for combining many weak learners in an attempt to produce a strong learner. Evaluating the prediction of an ensemble typically requires more

computation than evaluating the prediction of a single model, so ensembles may be thought of as a way to compensate for poor learning algorithms by performing a lot of extra computation.

An ensemble is itself a supervised learning algorithm, because it can be trained and then used to make predictions. The trained ensemble, therefore, represents a single hypothesis. This hypothesis, however, is not necessarily contained within the hypothesis space of the models from which it is built. Thus, ensembles can be shown to have more flexibility in the functions they can represent. This flexibility can, in theory, enable them to over-fit the training data more than a single model would, but in practice, some ensemble techniques (especially bagging) tend to reduce problems related to over-fitting of the training data.

Empirically, ensembles tend to yield better results when there is a significant diversity among the models. Many ensemble methods, therefore, seek to promote diversity among the models they combine. Although perhaps non-intuitive, more random algorithms (like random decision trees) can be used to produce a stronger ensemble than very deliberate algorithms (like entropy-reducing decision trees). Using a variety of strong learning algorithms, however, has been shown to be more effective than using techniques that attempt to dumb-down the models in order to promote diversity.

Performance Metrics

MSE (average Square error)

Rmse (root mean square error)

Mae (mean absolute error)

MAPE (average absolute percentage error)

Smape (symmetric mean absolute percentage error)

MADP (average absolute deviation percentage)

MASE (mean absolute scaled error)

Model Evaluation: Quantifying The Quality Of Estimates

Classification Metrics

More than two multi-layers and multi-labels

Accuracy score

Cohen's kappa

Confusion matrix

Classification report

Hamming loss

Jaccard similarity coefficient score

Sensitivity, recall and F-Score

Hinge loss

Log loss

Matthews correlation coefficient

ROC curve (ROC)

Zero one loss

Brier score loss

Multi-label sorting metrics

Coverage error

Average accuracy in tag ranking

Ranking loss

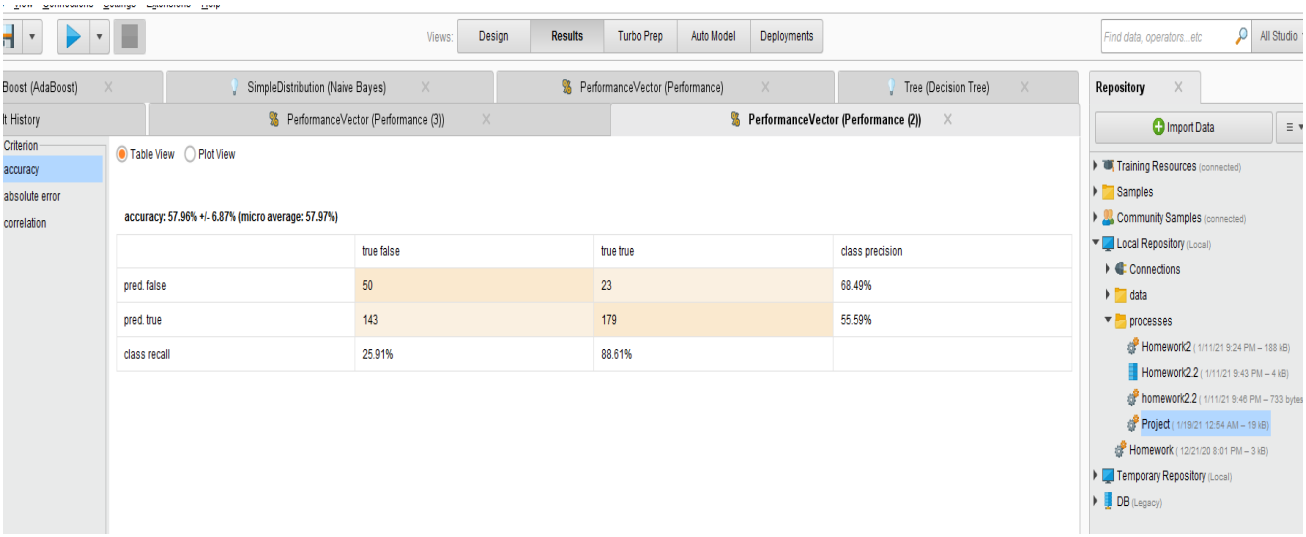
Regression metrics

Variance score described

Average absolute error

Mean checkered error

- **Parameter tuning / Comparison**

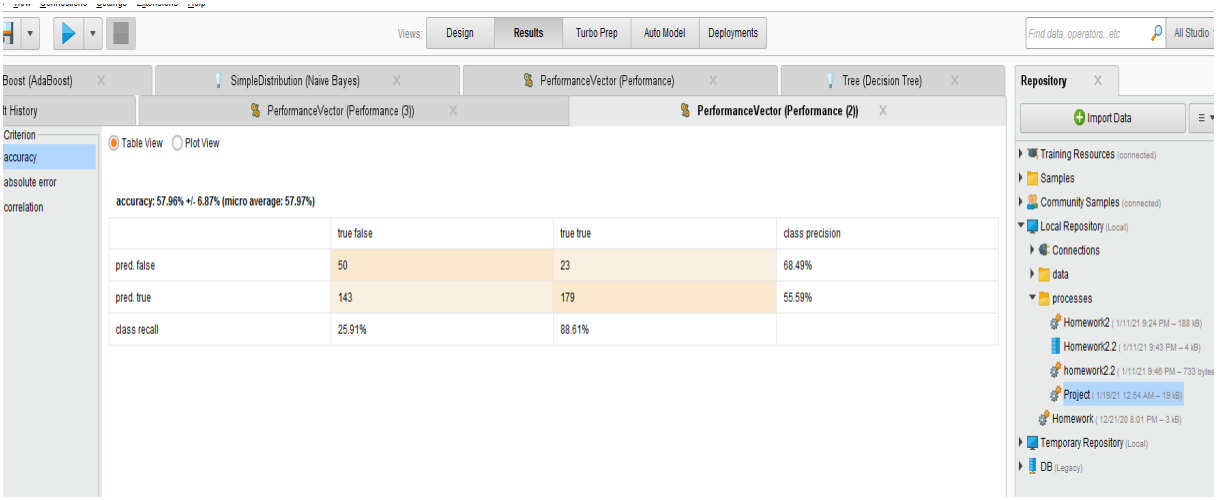


Adaboost, the most successful of the models. I found a higher value as yield than others

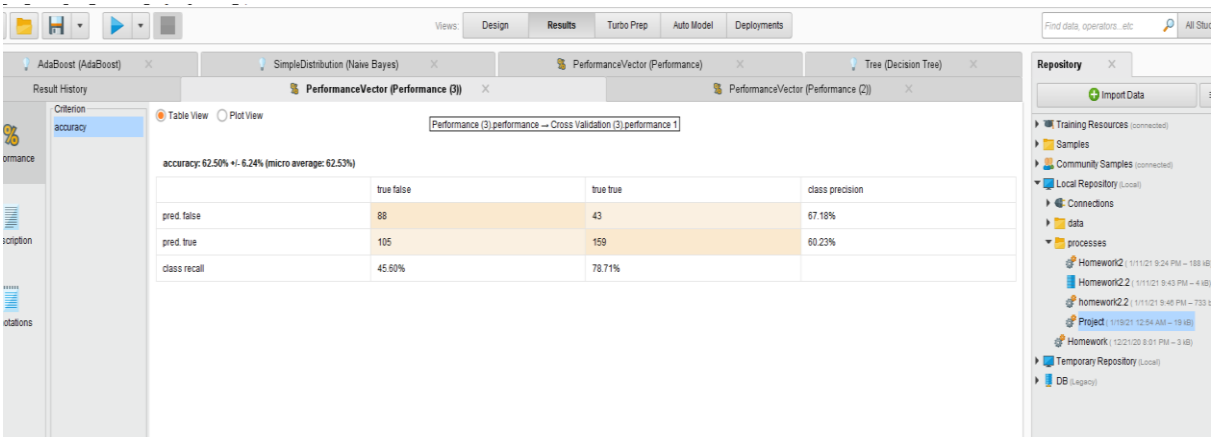
Feature Selection and Comparison

The remove useless attributes operator is an operator that deletes unnecessary and unused groups in the data.I used this operator in the second step of my design, but it didn't make any changes.

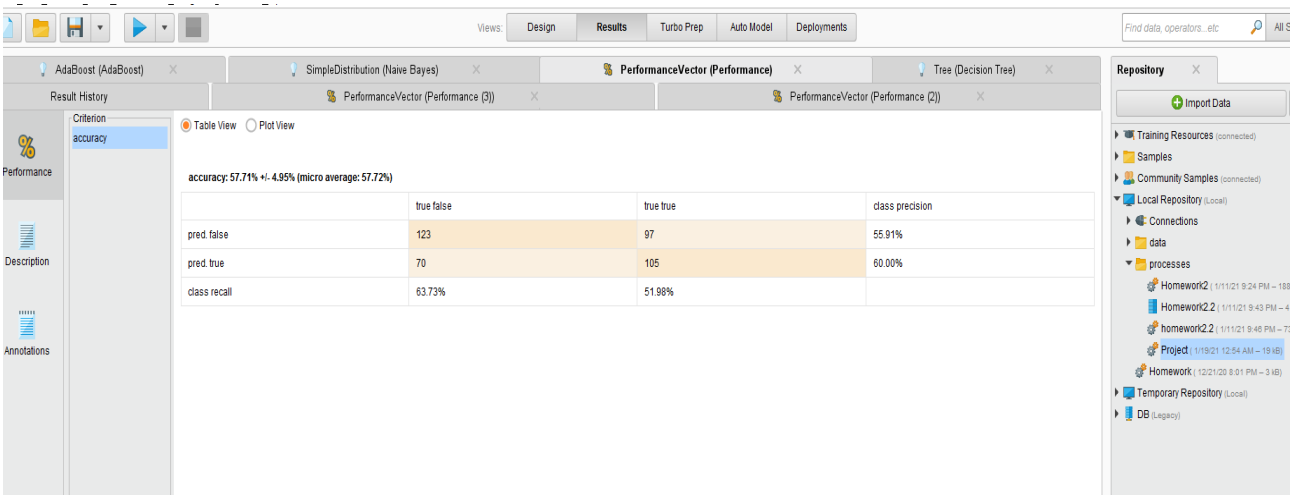
Adaboost



Naive Bayes



Decision Tree



Oğuz Kaan Satan
1621221026

18.01.2021