Feature Selection for Student Performance

Filippo Caliò Dario Del Gaizo Victor Lopo
calio | dariodg | lopo @kth.se

October 28, 2022

1 Problem Description

Education is a really important matter when it comes to students' future. There are several aspects that may influence their performance more than others, identifying them is crucial to understand the context they come from and help them during their education path. Both what happens in and outside the classroom has an influence on students' lives. The atmosphere at home, study habits, parents' salaries, class attendance or even extracurricular activities are some of the data that can give us clues about students' final grades. If teachers know which features are the most relevant and influential in students' grades, they can either enhance them or pay more attention to those students whose circumstances are more unfavourable.

2 Tools

The tools, frameworks and APIs that have been used during the project are:

- Databricks
- Apache Spark API
- PySpark: interface for Apache Spark in Python
- Spark.ml library
  - Feature selection
  - Cross Validation
  - Model for training
- Spark SQL for data visualization

The use of Databricks Notebooks does not require the creation of a SparkSession and a SparkContext because it is done automatically.

3 Data

In this project two Kaggle datasets [1] [2] are used and merged coherently, obtaining at the end over 1000 tuples and 33 features. They are related to students' personal life, past failures, mid-term grades, the target final grade and and these values were acquired from a survey of secondary school math students.

4 Methodology and Algorithms

4.1 Data Preprocessing

We had to determine whether the two datasets contained null data and included the same data before uniformizing it. We merged them together after observing that they lacked any common tuples, and we then examined the type of each property. We created four kinds of grade: very_bad, bad, good, and very_good for the last column, "G3", which is the output target (numeric: from 0 to 20) and contains the final year grade. Following that, we plotted the column's distribution [Figure 1] to determine whether or not its values correspond to a Gaussian Distribution.

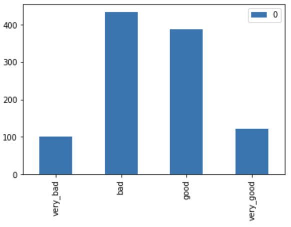

Figure 1: G3 labels distribution

The class StringIndexer [3], which converts strings from columns of labels to columns of label indices, was used to change the type of the columns that contained strings as well.

4.2 Correlation Matrix

We plotted the Correlation Matrix [Figure 2] to determine the correlation between all the variables, and we noticed that Grade 1 and Grade 2, which are the grades from the course's prior test, have the highest correlation.

[1] https://www.kaggle.com/datasets/larsen0966/student-performance-data-set

[2] https://www.kaggle.com/datasets/devansodariya/student-performance-data

[3] https://spark.apache.org/docs/2.3.0/api/java/index.html?org/apache/spark/ml/feature/StringIndexer.html

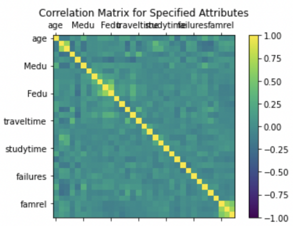

Figure 2: Correlation Matrix

4.3 Train-Test Split

We divided the entire dataset into a training set and a test set, with the split percentage set at 85%. Using the training dataset, we trained the feature selection, and the test set served as the basis for our prediction.

4.4 Feature Selection

Our aim was to find the best features, among the 33 attributes of the dataset, and to do we used the pyspark class `UnivariateFeatureSelector` [4], that performs feature selector based on univariate statistical tests against the labels, setting the `featureType` categorical and `labelType` as well. We kept in a dictionary the greatest feature combinations with a range of 5 to 10 features.

4.5 Cross Validation

We performed model selection using K-fold cross validation by dividing the dataset into a number of non-overlapping randomly partitioned folds that were utilized as independent training and test datasets, using the pyspark class CrossValidator[5].

5 Results

To determine which model (optimal number of features) is the best, we calculated the accuracies of the models with various features and plotted them onto a graph [Figure 3].

We have observed that the model with six features `['failures', 'Grade1_Idx','Grade2_Idx', 'higher_Idx', 'Medu', 'Mjob_Idx']`, which are, respectively, the number of prior class failures, the first and second period grades, the motivation to pursue higher education, the mother's education, and the mother employment, performs better (accuracy = 0.86 on the training set).

[4] https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.UnivariateFeatureSelector.html

[5] https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.tuning.CrossValidator.html?highlight=crossvalidator

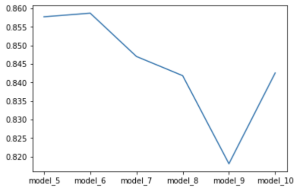

Figure 3: Model accuracies

Following that, we ran our 6-features model over the test set and got an accuracy score of 0.84, which is significantly higher than the accuracy score of 0.74 obtained by the 32-features model over the same test set.

The accuracy of the PCA analysis we attempted to do in the end was lower than that of the earlier models (accuracy = 0.6), and we tried to figure out why by examining the variance explained by each component.