**Predicting Student Achievement using PySpark Machine Learning Models**

John Johnson, Katy Matulay, Justin Minnion

College of Computing & Informatics, Drexel University, Philadelphia, PA 19104, USA

**Abstract**—Student test scores and academic achievement can be influenced by many external factors. The impact they make can either be positive, negative, or neutral. The student performance dataset contains 33 columns, of which 30 are descriptive features, and 3 are performance measures representing exam grades on 3 exams. In our study, three machine learning algorithms were applied using PySpark in a Databricks environment. The models generated were used to predict the outcome of student achievement as a binary pass/fail grade on the final exam (G3). The classification models used were logistic regression (LR), random forest (RF), and naive bayes (NB). These models rely on supervised learning, so extensive work will be included for the validity of our data and to understand the target feature(s). Each model was tested with two sets of features and performance was evaluated by measuring accuracy, precision, recall, F1, and ROC curves. Overall the Logistic Regression Cross Validation Model 2 with weights (LR CV WT 2) performed the best with accuracy of 92% and AUC = 0.921. Random Forest Model 2 with weights (RF WT 2) performed second best with an accuracy of 90.7% and AUC = 0.901, but the LR CV Model 2 was close behind with an accuracy of 89.3% and AUC = 0.873. Model 2 features outperformed Model 1 regardless of the ML models used, thus highlighting that prior academic performance can predict future performance.

**Keywords**—Classification, Academia, PySpark. 

**1 INTRODUCTION**

Student performance is a topic which is frequently studied and discussed. Common performance metrics used to quantify student performance include grade point average (GPA), standardized test scores, or individual assignment/class grades. Much focus is given to readily apparent environmental factors which may influence a student’s performance, such as the quality of their schools or their teachers. There are numerous factors external to the classroom (e.g. education level of a student’s parents, internet access at home) as well as additional observations of a student’s behavior (e.g. weekly study time, number of school absences) which may aid or hinder a student’s performance. Herein we study these factors and observations and their impact on student performance.

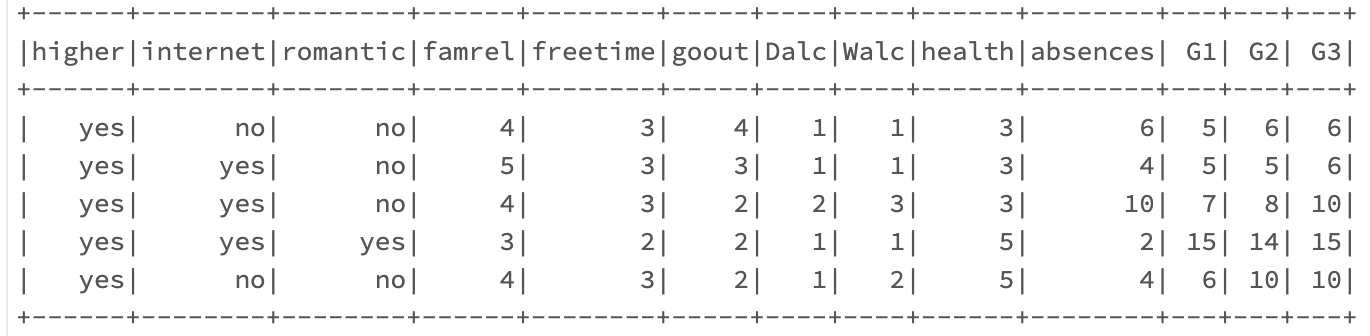
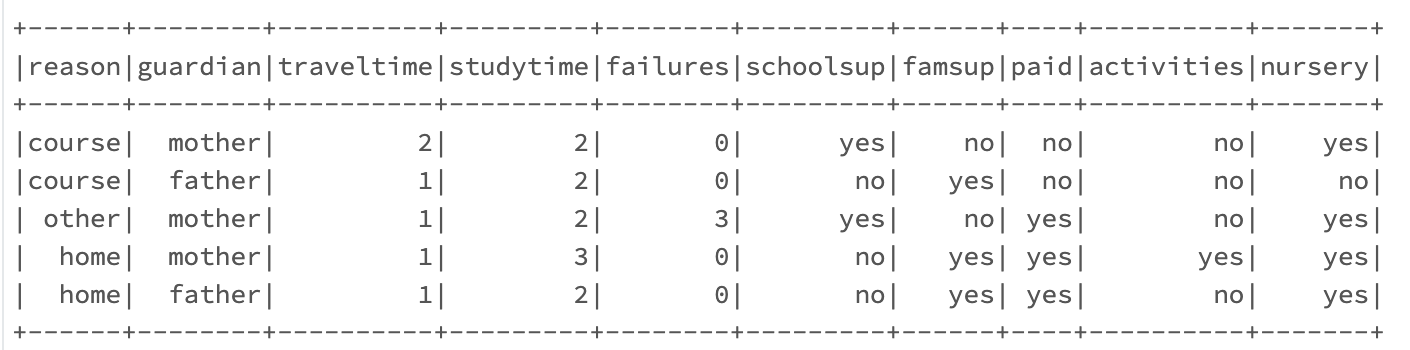
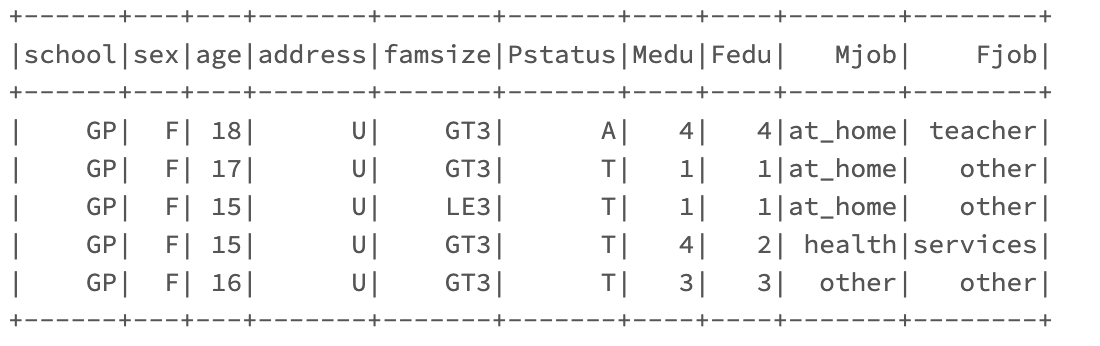
Countless studies have been conducted to identify factors which may predict improved student performance, but not all have a direct solution. Using machine learning (ML) algorithms within Spark and applying those algorithms with Spark’s Python API PySpark, our study seeks to take an empirical approach to understanding which factors have the most influence on student performance. Understanding the impact of these factors can lead teachers, professors, and academic advisors into directing students on how to improve their level of learning. By utilizing past students’ academic and demographic characteristics, we can attempt to predict how current students will perform and what factors they can change to help improve their academic success.

The environment used to conduct this study was enabled by the Databricks Community Edition, a free offering on the Databricks platform. The environment was a Jupyter-style interactive Python notebook with a cluster composed of a single worker node. The worker node utilized 15.3 GB of memory and two processor cores, and used the Databricks runtime version 10.4 LTS running Apache Spark 3.2.1 and Scala 2.12.

**2 DATASET**

The student performance dataset from UCI Machine Learning Repository [1] contains 395 rows and 33 columns which describe a variety of features related to students and their performance. The dataset was uploaded in 2014 and is derived from a research study published in 2008 [2] which surveyed students in two Portuguese secondary schools to determine why Portuguese students were failing at higher rates than European counterparts. Each row describes one student and various academic or demographic features, such as: past failures, family size, primary guardian, alcohol consumption, internet availability at home, and mother/father education level.

Within the 33 columns there exist a variety of variable types– from nominal to ordinal and discrete, and a majority of the data values are strings. The target variable is “G3”, which represents the student’s mathematics exam score within a range of (0, 20), with scores of 10 or above classified as a passing grade. Features G1 and G2 represent individual exam grades for prior terms. Additional details about the dataset can be found in *Appendix A* and *Appendix B*.

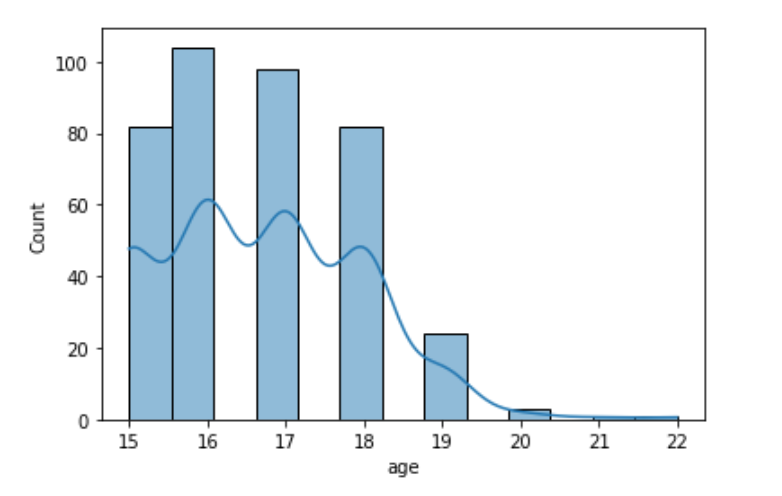


**Figure 1:** Dataset preview

**3 EXPLORATORY DATA ANALYSIS**

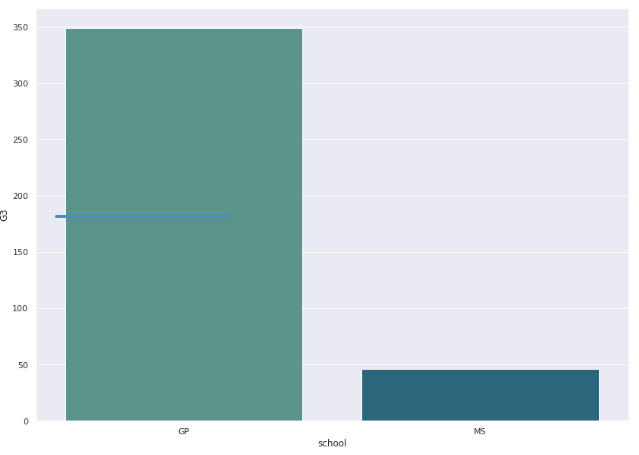
The dataset had no missing data as it was pre-processed by the original dataset authors prior to distribution, however there were imbalanced features and a column with significant outliers. With exploratory data analysis (EDA) we sought to understand the variety of the data points and data types in order to better pre-process the data for our models.

Using Python packages **matplotlib** and **Seaborn** on a **pandas** dataframe, it was possible to visualize the distribution of data in the “age” column. As expected, this dataset is mainly composed of secondary (high school) aged students, with 29/395 (7%) records representing students over the age of 18.

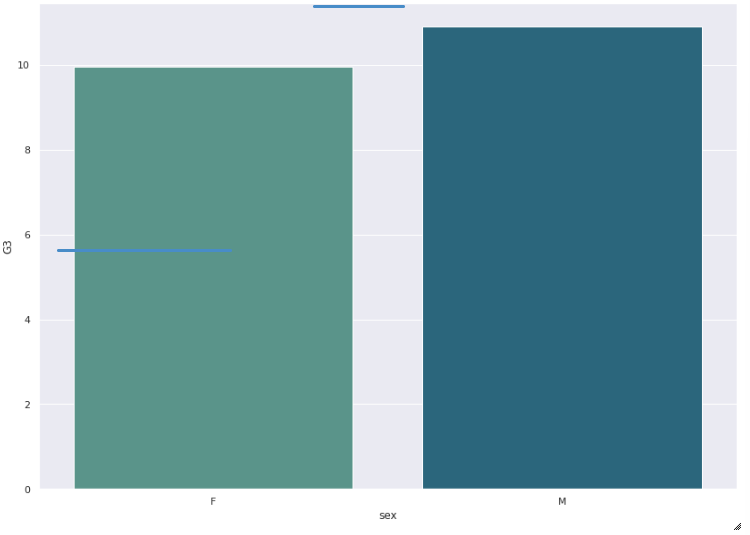


**Figure 2:** Age Distribution

Also with the imbalanced nature of the dataset extensive work was done looking into the different categorical values and determining if any biased features were found. The first pair of features examined in the study were “school” and “G3”. The results indicated that the academic performance between the two schools studied in the original dataset were very similar. The grades for the schools were similar, but this feature eventually ended up being removed due to the imbalanced number of students captured from each school. This imbalance is visualized in *Figure 3* below.

**Figure 3:** School Distribution

Another comparison that we conducted in our study was for feature “sex” against target “G3”. The findings of this comparison showed a slightly higher academic performance by males, but shouldn’t skew the model to favor them due to the roughly equivalent number of records falling under each value of “sex”.

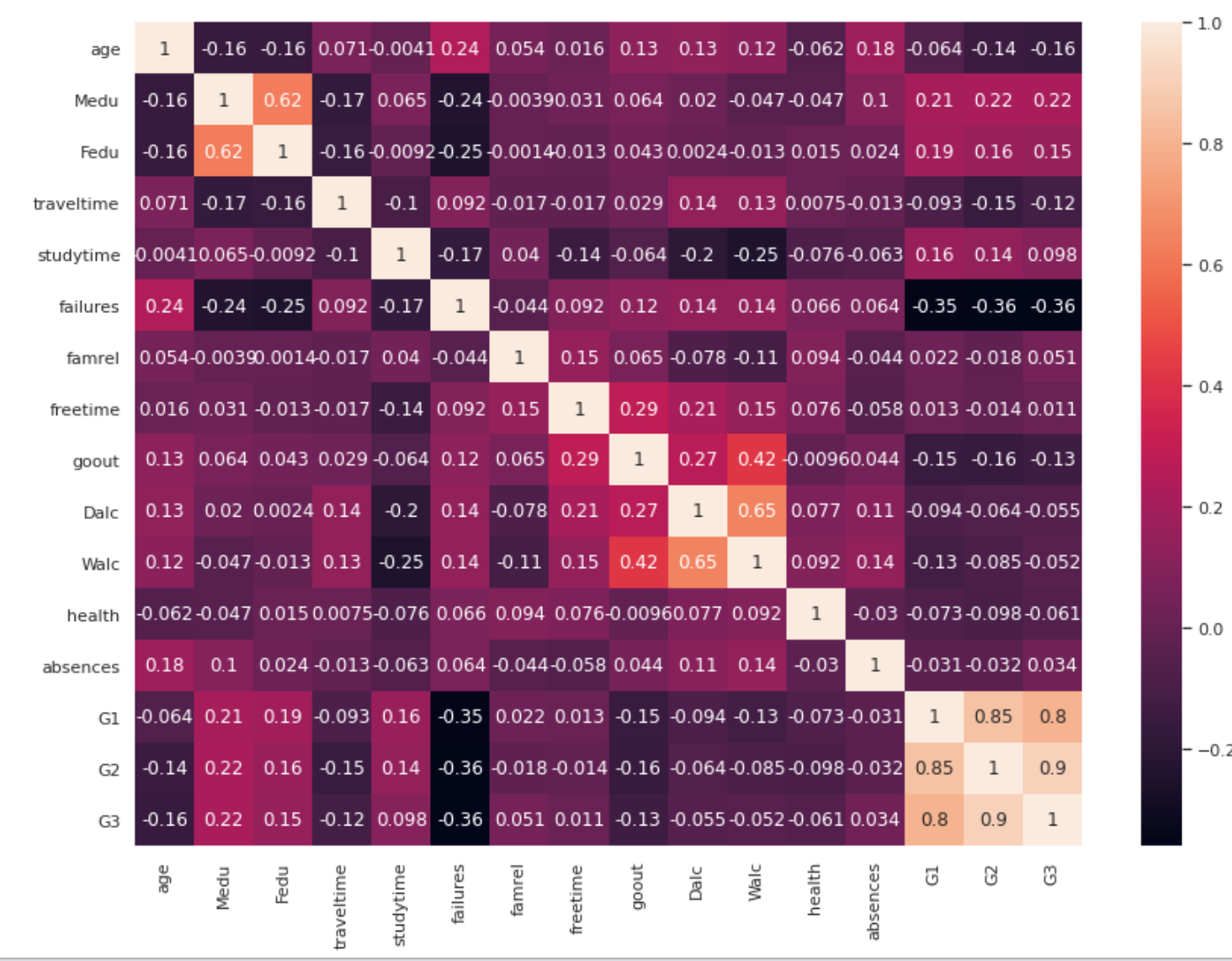


**Figure 4:** Sex/G3 Distribution

In an effort to identify the most relevant feature columns, a correlation heat map was generated along with raw correlation calculations. Feature pairs with the highest correlation in descending order were:

G2/G3, G1/G2, G1/G3, Weekend/Daily Alcohol Consumption (Walc/Dalc), Mothers/Fathers Education Level (Medu/Fedu)

As a result of the correlation analysis, two feature collections were created. “Model 1” included all columns except G1 & G2. “Model 2” removed 5 low-correlation columns but added G1 & G2.



**Figure 5:** Correlation heatmap

A box and whisker plot was used to determine the distribution of values and identify any columns with significant outliers (refer to *Figure C1* in Appendix C). Outliers were identified for the absences column, which would then need to be normalized during data-prep.

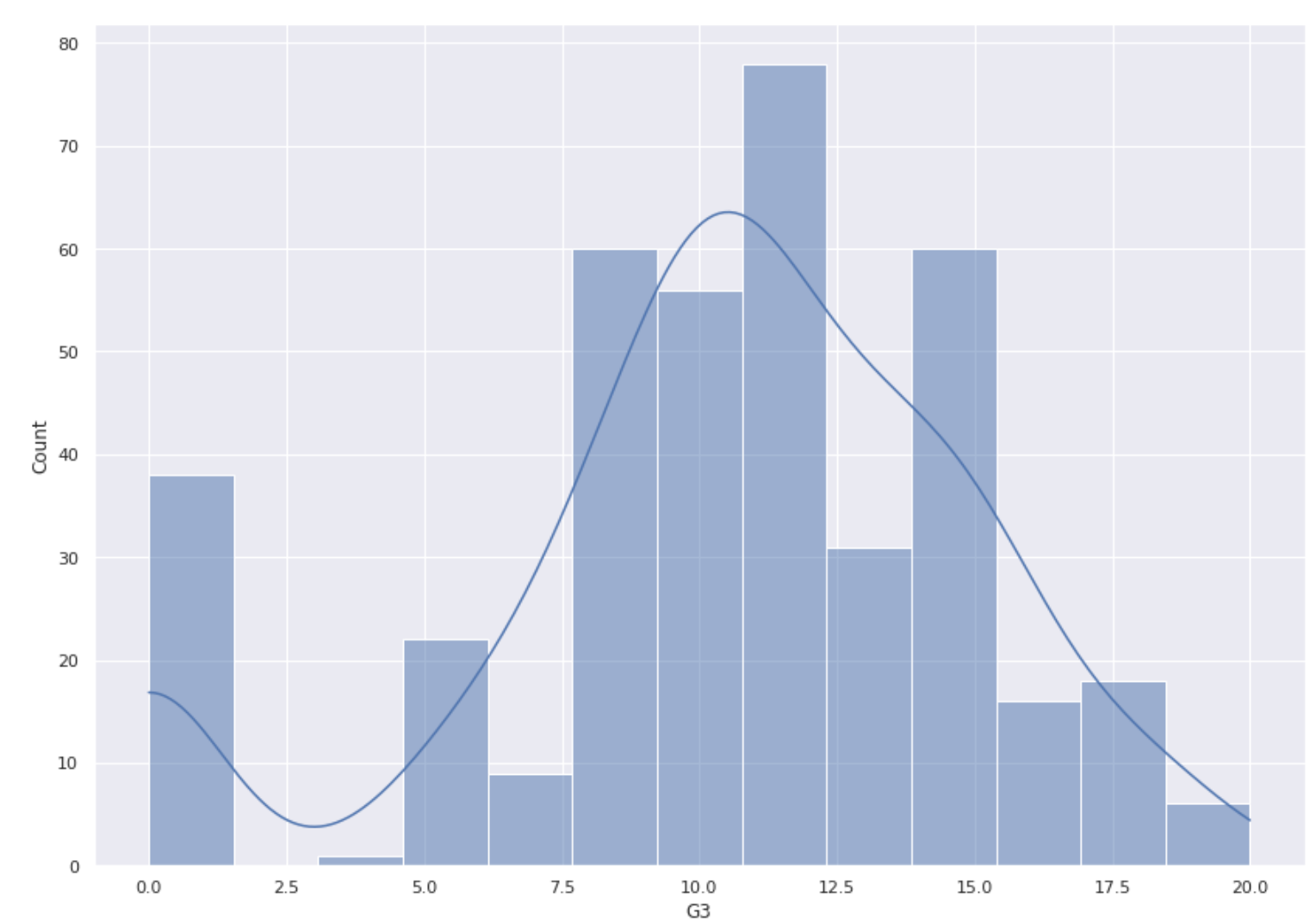
**4 METHODOLOGY**

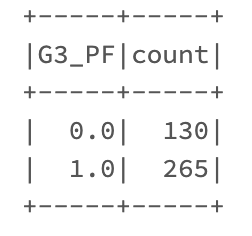
**4.1 Data Preprocessing**

Because of the variety of columns and the decision to use a binary classification machine learning model, several pre-processing steps were necessary.

*4.1.1 - Data Prep*

In order to utilize a binary classification machine learning model, the target variable “G3” was converted to a binary variable to represent if the student passed (1) or failed (0) the exam. A user-defined function was applied to convert the integer value based on scores 10 and above being considered a passing score [2],[3]. Intuitively, more students fell into the *pass (1)* than *fail (0)* category.

**Figure 6:** Distribution of Grade (G3) before binary encoding to Pass/Fail



**Figure 7:** Distribution of Grade (G3) after binary encoding to Pass/Fail using mapping:   
[0,10) → Fail (0) and [10,20] → Pass (1)

*4.1.2 - Normalization*

The feature “absences” was the only discrete ordinal variable with significant outliers and was normalized using a UDF. Normalization didn’t significantly change the outcome of any models, and the resulting data type wasn’t compatible with Naive-Bayes. Because of this, the normalized “absences” feature was discarded and original “absences” feature data retained.

*4.1.3 - Weights*

The imbalanced outcome of student performance was an issue that required extra attention. To combat this issue of bias, cross-validation was used for all LR models. This helped resample different portions of the data so the models steered clear of overfitting.

Another method of balancing student performance in the dataset was including a weighted column in the parameters of all models [5]. This allowed the models to treat each algorithm with equal classes.

*4.1.4 - String Indexer*

Columns with string values were identified and passed to the **StringIndexe**r using a pipeline. Some columns had nominal binary string values (“Yes”, “No”), while others such as “reason” had more than two nominal values, thus requiring one-hot encoding.

*4.1.5 - One-hot Encoding*

Because not all columns required both **StringIndexe**r and **OneHotEncoder**, the encoding was split into two pipelines and executed separately using variables to hold each column list. Dummy variables were created from the string-indexed variables through one-hot encoding. Refer to *Figure C2* in *Appendix C* for code used to execute one-hot encoding.

*4.1.6 - Vector Assembler*

Finally, the feature columns were identified and assembled using the **VectorAssembler**. In total, 30 features were used for the “Model 1” feature collection, and 27 features for the “Model 2” feature collection.

*4.1.7 - Data Splitting*

The featurized dataset with columns “label” and “features” was split at an 80/20 ratio into train and test data frames using the **randomSplit** method and a seed for reproducibility.

**4.2 Machine Learning Models**

The two feature collections were curated for each Logistic Regression, Random Forest, and Naive Bayes models, referred to respectively as “Model 1” and “Model 2”.

**“Model 1” feature collection:**

* All feature columns except for G1 & G2
* G3 was converted to G3\_PF – binary Pass/Fail (1/0)

**“Model 2” feature collection:**

* Added indexed cols G1 & G2 as pass/fail binary features
* Removed feature columns that were less correlated or redundant per the research study (school, dalc, walc, mjob, fjob) [2],[3]
* G3\_PF retained as target variable

**Weighted “Model 2” feature collection:**

* Retained all features of “Model 2” with the addition of the weight column to balance the dataset

The training dataset was first trained on three base binary classification models: Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB). Logistic Regression was expanded to include a cross validation (LR CV) model. Weights (WT) were also introduced to all three models to balance the dataset.

Each classification model was tested using the two feature collections, with the feature collection noted in the name of the model (see *Figure 8*). The models were then scored for accuracy, precision, recall, and F1.

None of the Classification models using Model 1 features performed as well as those using Model 2 features. Both LR Model 2 and RF Model 2 performed well overall, proving that a student’s prior test performance strongly indicates future performance, as discussed by Cortez & Silva [2]. However, the LR CV Model 2 slightly outperformed the RF Model 2 when measured by AUC (0.873 vs 0.863).

AUC (area under the ROC curve) values were obtained from the **BinaryClassificationMetrics** PySpark ML library. Overall, the LR CV WT 2 performed the best with accuracy of 92% and AUC = 0.921 Model performance is summarized in *Figure 8* and *Figure 9*.

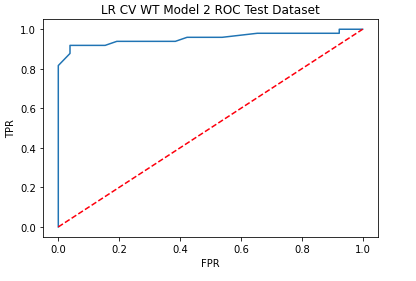
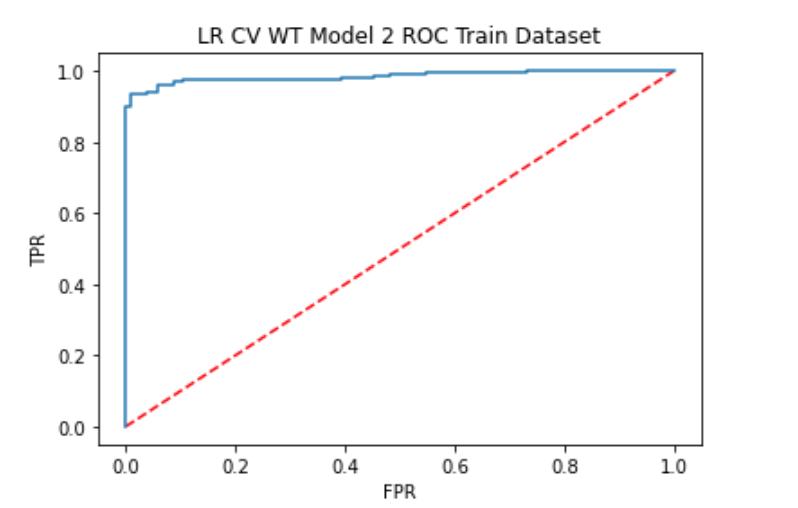
| ***ML Model and***  ***Feature Collection*** | *Accuracy %* | *Precision*  *%* | *Recall*  *%* | *F1*  *%* |
| --- | --- | --- | --- | --- |
| **LR CV Model 1** | 69.3 | 70.3 | 91.8 | 79.7 |
| **LR CV Model 2** | 89.3 | 90.2 | 93.9 | 92.0 |
| **LR CV WT Model 2** | 92.0 | 95.7 | 91.8 | 93.8 |
| **LR Model 2** | 73.3 | 77.4 | 83.7 | 80.4 |
| **LR WT Model 2** | 73.3 | 78.4 | 81.6 | 80.0 |
| **RF Model 1** | 69.3 | 69.7 | 93.9 | 80.0 |
| **RF Model 2** | 88.0 | 90.0 | 91.4 | 90.9 |
| **RF WT Model 2** | 90.7 | 93.8 | 91.8 | 92.8 |
| **NB Model 1** | 68.0 | 72.7 | 81.6 | 76.9 |
| **NB Model 2** | 84.0 | 83.6 | 93.9 | 88.5 |
| **NB Wt Model 2** | 88.0 | 88.5 | 93.9 | 91.1 |

**Figure 8*:*** Binary Classification Model Performance

| **ML Model and**  **Feature Collection** | **AUC** |
| --- | --- |
| LR CV 1 | 0.594 |
| LR CV 2 | 0.873 |
| LR CV WT 2 | 0.921 |
| LR 2 | 0.688 |
| LR WT 2 | 0.700 |
| RF 1 | 0.585 |
| RF 2 | 0.863 |
| RF WT 2 | 0.901 |
| NB 1 | 0.620 |
| NB 2 | 0.796 |
| NB WT 2 | 0.854 |

**Figure 9:**Area Under Curve (AUC) Performance

The values for LR Model 2 true positive rate (TPR) and false positive rate (FPR) were then obtained from the **model.summary** method for the train dataset, but were not as simple to generate for the train dataset as a new class of **BinaryClassificationMetrics** was necessary to implement in Python [4]. The results for both were then graphed using **matplotlib**.



**Figure 10**: ROC Curve for   
LR CV WT Model 2 Train and Test Datasets

**5 RESULTS AND DISCUSSION**

Using a Databricks Community Edition cluster, the LR CV Models required ~15 minutes to train, likely due to the cross validation. This was much slower compared to the RF and NB models without cross validation, which required less than 5 seconds to run. The additional time and resources needed to implement the LR cross validation only provided an average increase of 1.3% across all performance measurements when compared to the RF Model 2. The addition of weights to both LR CV and RF Model 2 improved performance across the board.

Because both the “Model 1” and “Model 2” feature collections contain over 20 columns, knowing which features are the most informative will assist in helping students increase their academic performance or identify early interventions that could prevent poor performance. Examining the outcomes of the models, a coefficient matrix was made to capture the models’ as a linear equation. The equation below represents the best model.

The results of these models can all be classified as effectively predicting the outcome of academic performance. While accuracy was one of the most important performance metrics of interest, the F1 score sums up the predictive performance by combining precision and recall. These metrics represent the overall quality of the models through correctly labeling true positives [6]. With the lowest model accuracy at 69.3% and the best model at 92%, all models would be labeled as good or excellent . Testing was also done to see which models represented an accepted harmonic mean (F1 Score). After measuring the outcomes, every single model ended up having an acceptable harmonic mean, which is set to be 0.5 or higher.

**6 CONCLUSION AND FUTURE WORKS**

In the education domain, a vast amount of data on student performance has been collected, however it requires both the implementation of ML models to predict performance and dissection of features to identify the most relevant attributes to focus on. Future expansion of this project would ideally explore feature dissection in order to isolate and identify the most critical features.

This particular dataset was feature heavy with 30 descriptive features and three additional grade or performance metrics. The decision to test two different feature collections (“Model 1” vs. “Model 2”) came from two papers that examined the statistical correlation and relevance of the features [3] [4]. Knowing that past performance (G1 & G2) was highly correlated to future performance (G3), removing that from the first dataset aimed to identify if other external features were strong enough alone to predict performance. Unfortunately, none of the models using the “Model 1” feature collection achieved higher than 0.6 AUC or 60% accuracy.. Thus, “Model 2” features clearly dominate across all ML model implementations.

Though the LR CV WT Model 2 outperformed the RF WT Model 2, it did take much longer to run (15 min vs. ~5 seconds) and required more computing resources. Running the notebook on a dataproc cluster with 24 cores and 60 GB ram did significantly improve run times from 15 min to 3-4 minutes for the model. For future expansion topics it would be worth using dataproc or upgrading the Databricks cluster.

**REFERENCES**

[1]  “Student Performance Data Set.” *UCI Machine Learning Repository*, 27 November 2014, <https://archive.ics.uci.edu/ml/datasets/Student+Performance>. Accessed 17 August 2022.

[2]  P. Cortez and A. Silva. “Using Data Mining to Predict Secondary School Student Performance.” *In A. Brito and J. Teixeira Eds., Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008)* pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7 <http://www3.dsi.uminho.pt/pcortez/student.pdf>

[3] Molin, Camilla. A Statistical Analysis of the Performance in Mathematics of Secondary Students in Portugal. 2020, <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-418811>. Accessed 2 August 2022.

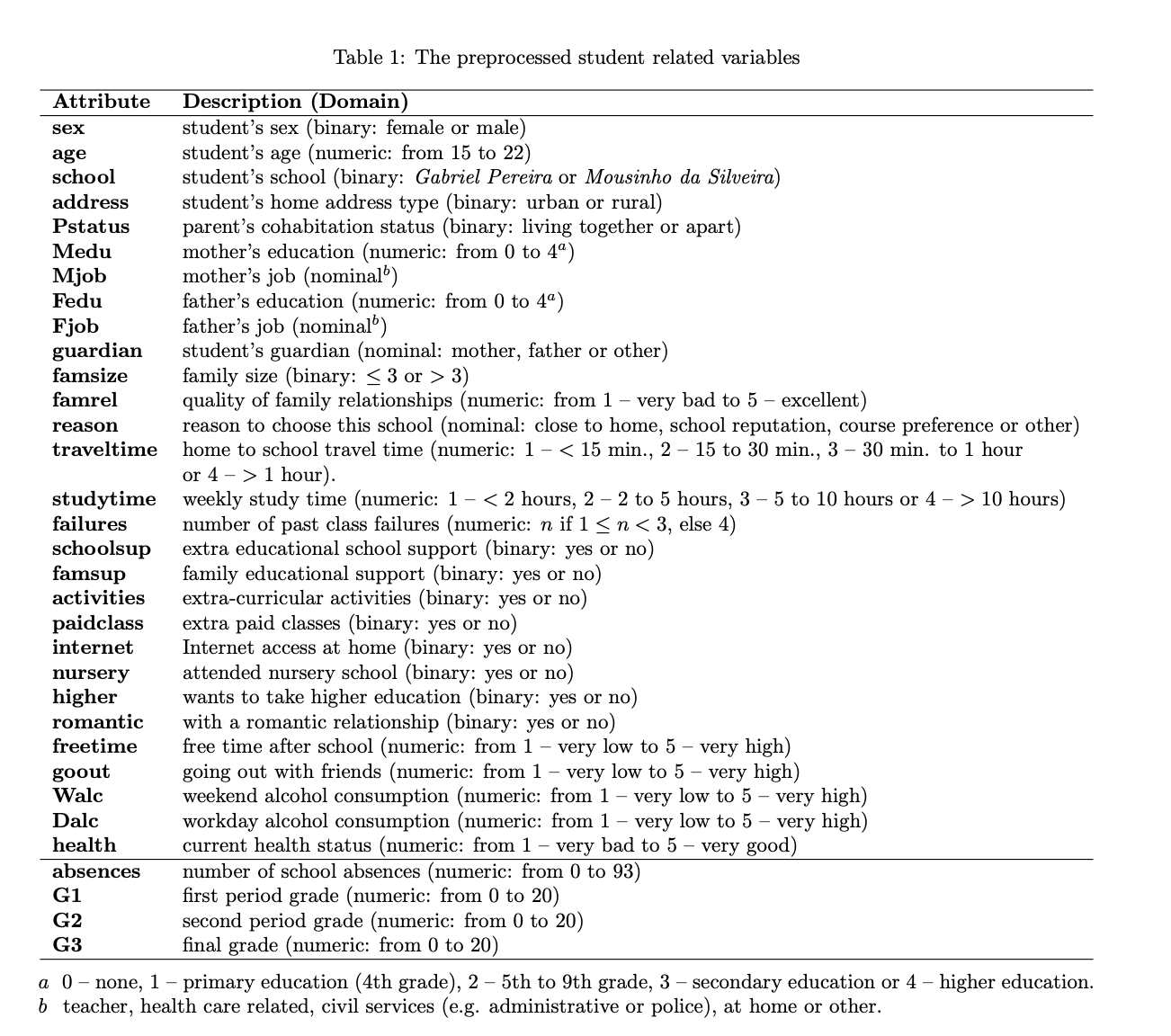
[4] “Pyspark extract ROC curve – iTecNote.” *iTecNote*, <https://itecnote.com/tecnote/pyspark-extract-roc-curve/>. Accessed 22 August 2022.

[5] “Introduction to Spark 3.0 - Part 4 : Handling Class Imbalance Using Weights.” *Madhukar's Blog*, 2 April 2020, <http://blog.madhukaraphatak.com/spark-3-introduction-part-4/>. Accessed 23 August 2022.

[6] Koehrsen, Will, and Sadrach Pierre. “When Accuracy Isn't Enough, Use Precision and Recall to Evaluate Your Classification Model.” *Built In*, 3 August 2021, <https://builtin.com/data-science/precision-and-recall>. Accessed 23 August 2022.

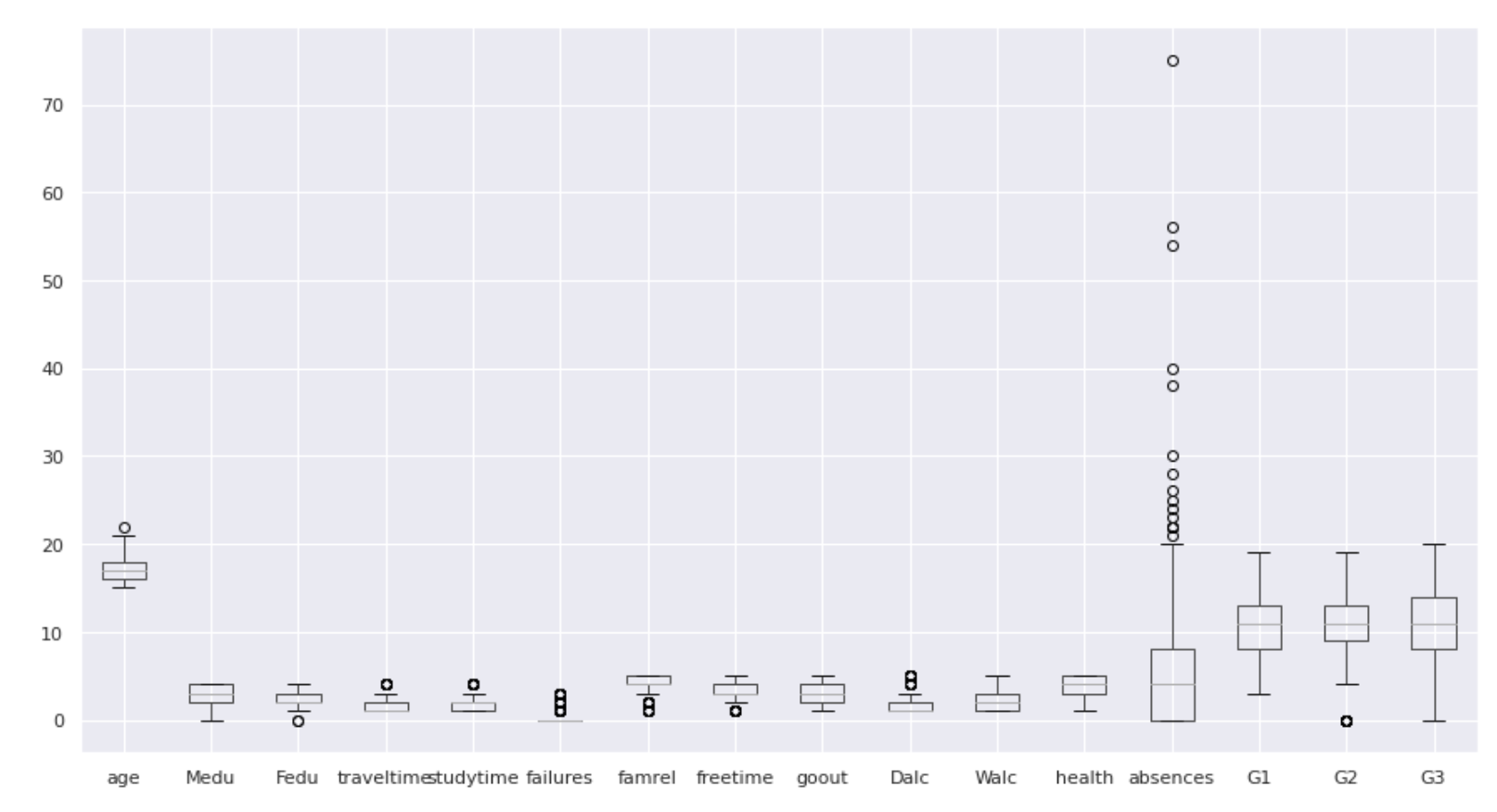
**APPENDICES**

**Appendix A:** Additional tables and figures

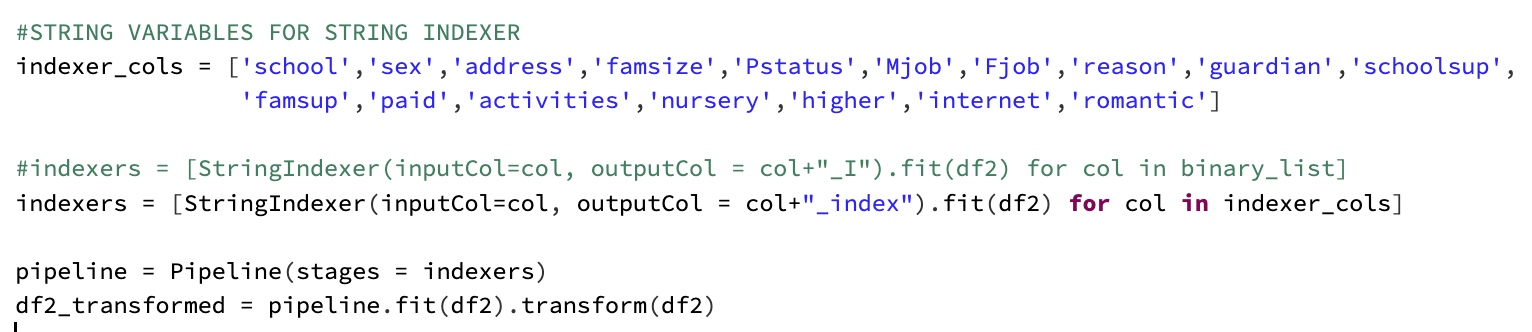


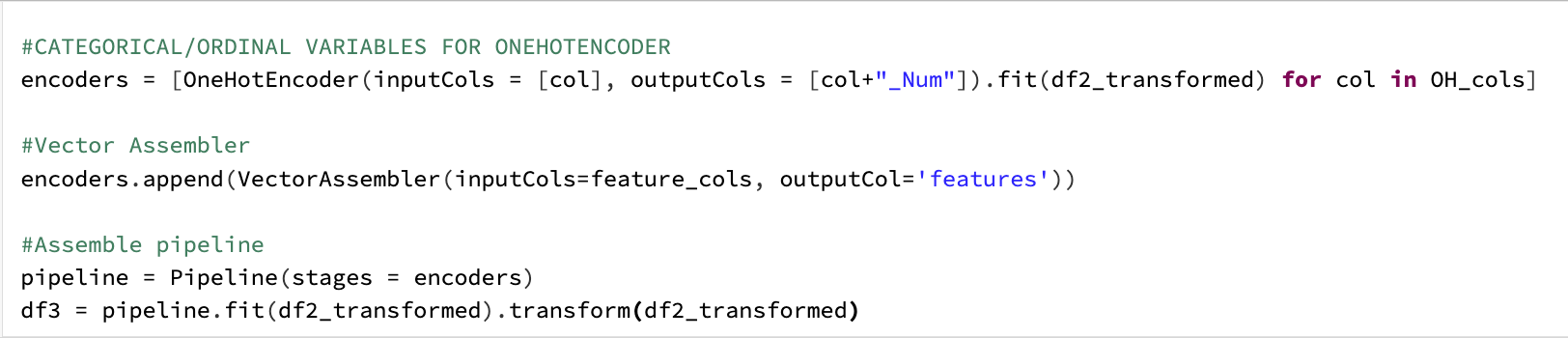
**Figure A1:** Student dataset columns, as described in P.Cortez and A. Silva [2]

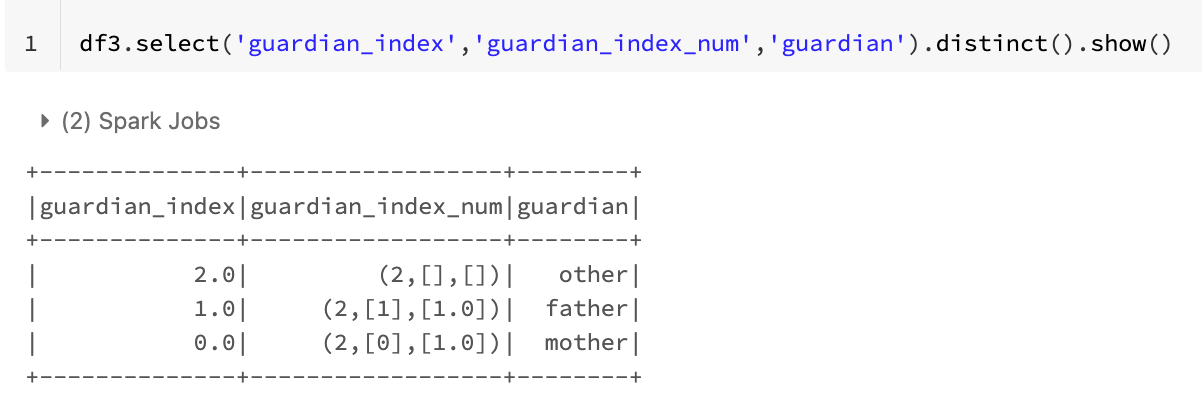
| **Column** | **Data Type** | **Variable Type** | **Description** | **Data Values** |
| --- | --- | --- | --- | --- |
| school | String | Nominal, binary | Name of school | GP- Gabriel Pereira, MS- Mousinho da Silveira |
| sex | String | Nominal, binary | Sex (Gender) | F- Female, M- Male |
| age | Integer | Discrete | Student' Age | 15-22 years |
| address | String | Nominal, binary | Students home address | U- Urban, R- Rural |
| famsize | String | Nominal, binary | Family size | LE3 - <= 3 people, GT3 - > 3 people |
| Pstatus | String | Nominal, binary | Parents cohabitative status | T- living together, A- living apart |
| Medu | String | Ordinal | Mothers education level | 0-none, 1-primary education (4th grade), 2- 5th-9th grade, 3-secondary education, 4-higher education |
| Fedu | String | Ordinal | Fathers education level | 0-none, 1-primary education (4th grade), 2- 5th-9th grade, 3-secondary education, 4-higher education |
| Mjob | String | Nominal | Mothers job | 'teacher', 'health care related', 'civil services', 'at home', 'other' |
| Fjob | String | Nominal | Fathers job | 'teacher', 'health care related', 'civil services', 'at home', 'other' |
| reason | String | Nominal | Reason to choose the school | close to home', 'school reputation', 'course preference', 'other' |
| guardian | String | Nominal | Students Guardian | mother, father, other |
| traveltime | Integer | Ordinal | Home to school travel time | 1- <15 min, 2- 15-30 min, 3- 30min-1hr, 4- >1hr |
| studytime | Integer | Ordinal | Weekly Study Time | 1- <2 hrs, 2- 2-5 hrs, 3- 5-10 hrs, 4- >10 hrs |
| failures | Integer | Ordinal | Number of past failures | 0, 1, 2, 3 |
| schoolsup | String | Nominal, binary | School provides extra educational support | Yes, No |
| famsup | String | Nominal, binary | Family education support | Yes, No |
| paid | String | Nominal, binary | Extra paid classes within course | Yes, No |
| activities | String | Nominal, binary | Extracurricular activities | Yes, No |
| nursery | String | Nominal, binary | Attended nursery school | Yes, No |
| higher | String | Nominal, binary | Wants to take higher education | Yes, No |
| internet | String | Nominal, binary | Internet access at home | Yes, No |
| romantic | String | Nominal, binary | Has a romantic relationship | Yes, No |
| famrel | Integer | Ordinal | Quality of family relationships | 1- very bad to 5- excellent |
| freetime | Integer | Ordinal | Free time after school | 1- very low to 5-very high |
| goout | Integer | Ordinal | Going out with friends | 1- very low to 5-very high |
| Dalc | Integer | Ordinal | Daily alcohol consumption, weekday | 1- very low to 5-very high |
| Walc | Integer | Ordinal | Weekend alcohol consumption | 1- very low to 5-very high |
| health | Integer | Ordinal | Current health status | 1- very bad to 5-very good |
| absences | Integer | Discrete | Number of school absences | 0 to 93 |
| G1 | Integer | Discrete | First period grade | 0 to 20, >= 10 is passing |
| G2 | Integer | Discrete | Second period grade | 0 to 20, >= 10 is passing |
| G3 | Integer | Discrete | Third period grade | 0 to 20, >= 10 is passing |



**Figure C1:** Box and Whisker Plot







**Figure C2:** One-hot encoding implementation in Python