Cmput 466 Project Report

Titus Li

Introduction:

- Understanding the factors that affect students' performance in school is crucial for educators, parents, and policymakers. Identifying these factors allows interventions and support to enhance students' learning experiences. The research can hugely benefit areas like targeted policy development, personalized educational strategies, and resource allocation. Thus, I aim to find out the most important factors about a student regarding their performance.

Data Used:

- The data used in this project is one of the public data sets from the UCI Machine Learning Repository, it approached 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.
- The data set is given in two separate sets, one of them contains the data from students who enrolled in math class, and the other contains data from students who enrolled in Portuguese class. There are 395 students in the math set, 649 students in the Portuguese set, and there are 382 students that belong to both datasets. For this project, we are only going to use the 382 students who appeared in both sets with similar attributes.

- There are 33 attributes in both data sets, and after combining and filtering out the data needed, 36 attributes are used in real practice.
    1. school - student's school (binary)
    2. sex - student's sex (binary)
    3. age - student's age (numeric: from 15 to 22)
    4. address - student's home address type (binary: U/R)
    5. famsize - family size (binary: >=3)
    6. Pstatus - parent's cohabitation status (binary:T/A)
    7. Medu - mother's education (numerical)
    8. Fedu - father's education (numerical)
    9. Mjob - mother's job (categorical)
    10. Fjob - father's job (categorical)
    11. reason - reason to choose this school (categorical)
    12. guardian - student's guardian (categorical)
    13. traveltime - home to school travel time (numerical)
    14. studytime - weekly study time (numerical)
    15. failures - number of past class failures (numerical)
    16. schoolsup - extra educational support (binary)

17. famsup - family educational support (binary)
18. paid - extra paid classes within the course subject (binary)
19. activities - extra-curricular activities (binary)
20. nursery - attended nursery school (binary)
21. higher - wants to take higher education (binary)
22. internet - Internet access at home (binary)
23. romantic - with a romantic relationship (binary)
24. famrel - quality of family relationships (numerical)
25. freetime - free time after school (numerical)
26. goout - going out with friends numerical)
27. Dalc - workday alcohol consumption numerical)
28. Walc - weekend alcohol consumption (numerical)
29. health - current health status (numerical)
30. absences - number of school absences (numerical)
31. MG1 - first-period math grade (numerical)
32. MG2 - second-period math grade (numerical)
33. MG3 - final math grade (numerical)
34. PG1 - first-period Portuguese grade (numerical)
35. PG2 - second-period Portuguese grade (numerical)
36. PG3 - final Portuguese grade (numerical)

- Due to efficiency and usability, all binary and categorical attribute is handled with one hot method and mapped out in mapping.txt

Approaches and Baselines:

- Just like I mentioned above, datasets do contain categorical and binary data, thus in the project I used pandas (a third-party library) to read and handle the CSV data, after using categorical_encoder (a third-party library relies on sklearn) to perform one hot method, it expands my data features into 56 features.

- The project runs the data through 5 different algorithms: Decision tree, Random Forest, Naive Bayes, Linear regression and weighted Quadratic Discriminant Analysis. Each algorithm will run 100 times to tune its parameters while selecting its best accuracy. The data will be split into a 75:20 ratio as training data and testing data during the data organization stage. Training data will further split into a 9:1 ratio as training data and validation data during the train-validate-test phase of the program. All algorithms used are from sklearn library, except for linear regression, which is a direct modification of the code in Coding Assignment 1. In this case, the linear regression used min-batch gradient descent for better accuracy.

Evaluation Metric:

- The goal of the project is to find algorithms with a relatively short run time while getting relatively higher accuracy to the output. Thus, multiple runs of the program are conducted to make sure the best algorithm with a relatively stable outcome which means the outcome of this project will not be an approximation.
- Top 10 feature of each algorithm will be displayed and conducted on frequency analysis, if the important feature is less than 10, it will use how many it has.

Result:

The result of each algorithm is recorded and further coded into a detailed report in Report.txt, the following table is a direct translation of that result.

	1	2	3	4	5	6	7	8	9	10
Decision Tree	sex_2	school_2	sex_1	address_1	school_1	age				
Random Frost	goout	absences	Fedu	age	failures	Medu	higher_1	school_1	health	higher_2
Naive Bayes	nursery_1	nursery_2	famsup_1	famsup_2	Mjob_4	freetime	traveltime	romantic	studytime	higher_2
Linear Regression	Medu	Fedu	age	goout	absences	studytime	failures	higher_1	address_1	romantic
Weighted Quadratic Discriminant Analysis	activities_1	activities_2	Fjob_2	famsup_2	famsup_2	Mjob_1	romantic	romantic_2	reason_3	sex_1

These are the top frequency features:

10	9	8	7	6	5	4	3	2	1
age	failures	romantic	sex_1	address_1	school_1	goout	absences	Fedu	Medu

These are the algorithms used with their accuracy and run time:

Algorithm	Run time (Seconds)	Accuracy (%)
Decision Tree	0.38099	86.20689
Random Frost	7.00497	82.75862
Naive Bayes	0.48579	72.41379
Linear Regression	0.07260	73.95833
Weighted Quadratic Discriminant Analysis	0.43226	63.54166

As you can see, the above tables are similar in their accuracy but very different in their importance of features. After some testing and reruns, I found that the total run time for 5 algorithms to finish running is around 7.5 to 9 seconds for 328 sets of data, where random frost takes the majority of the time and linear regression takes the least. In term of accuracy Decision tree and random frost is most likely to land in the 80% - 90% range, Naive Bayes and linear regression tends to land in the 70% - 80% range, then the weighted quadratic discriminant analysis tends to be the lowers with less than 65% of accuracy, sometimes it can be lower than 50%.

While the significance of various features may appear somewhat convoluted, it is noteworthy that over 15 features emerged consistently in two or more algorithmic results. This suggests that there are numerous attributes to consider when evaluating a student's performance in a class. Among these, commonly recognized factors such as age and the number of previous classes failed are crucial considerations. Students often exhibit improved academic performance when experiencing stress related to university, yet a history of repeated failures may still impede success. Nevertheless, it is imperative to also account for factors such as parental education levels, as parental perspectives on education can significantly influence a student. Additionally, factors like the student's relationship status may introduce distractions from academic pursuits, and aspects of their home environment or the motivations behind their attendance at school should not be overlooked.

Database:

Cortez,Paulo. (2014). Student Performance. UCI Machine Learning Repository. https://doi.org/10.24432/C5TG7T.

Third-Party Libraries:
Pandas: https://pypi.org/project/pandas/
Sklearn: https://scikit-learn.org/stable/
Encoder: https://pypi.org/project/category-encoders/
Numpy: https://numpy.org/

Reference website:
https://www.datacamp.com/tutorial/decision-tree-classification-python
https://www.datacamp.com/tutorial/random-forests-classifier-python
https://www.datacamp.com/tutorial/naive-bayes-scikit-learn
https://www.statology.org/quadratic-discriminant-analysis-in-python/
https://www.freecodecamp.org/news/sort-dictionary-by-value-in-python/
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html