Students' Grades Machine Learning Predictions

Chris Wu

Introduction

The related graphs, code, and datasets can be found in the following repository: https://github.com/cheesuschris/320Homeworks/tree/main/Homework3. To check this homework out yourself, clone the repo and run the script.

Report summary

For the Fall 2025 class of CMSC320 (Intro to Data Science) at the University of Maryland - College Park, students' grades on their first midterm, as well as some survey answers about their academic and non-academic background, were collected in order to better understand what components factor into a student's successfulness when taking tests. To truly understand the relationship between student habits/qualities and their scores, various machine learning models, both classification and regression-based, were developed to try and connect the students' survey answers and their percentage scored on the first midterm. Within this report contains a study for how the Fall 2025 class of CMSC320's habits and background impacts their performance on midterms; an introductory exploratory data analysis/visualization/preprocessing, followed by classification models for student pass/fail rate, classification models for student letter grade, and regression models for student score percentages, and related graphs, confusion matrices , and explanations of the models' performances are offered within the related script.

Interesting findings

There were several interesting insights about student behavior affecting their midterm score that were revealed by some of the developed models. This included the lack of data supporting REALLY good accuracy or R^2 from the classification or regression algorithms. In addition, some of the models' performances themselves were studied and explained. For one, binary classification did better than letter grade classification, which did better than exact regression matching. Finally, the best possible study habits are: Sleep, Studying, Less Scrolling, Confidence in an A, and Doing All the Readings

Background

Explaining the dataset

The analysis and models were developed from the Fall Class of 2025's CMSC320 student study/daily habits survey & midterm scores data. The dataset includes:

- Student Response Timestamp
- Lecture Attendance Rate (Categorical)
- Prior Experience with ML/Data Science (Categorical)

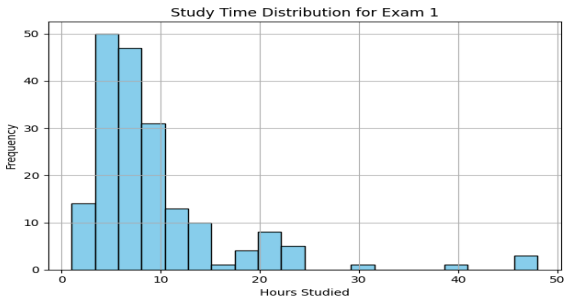
- Class Section Time (Categorical)
- Hours Studied for Midterm 1 (Float, Continuous)
- Standing by Year in School (Categorical)
- Student Readings Completion Rate (Categorical)
- Average Hours spent daily on Infinite Scroll Sites (Float, Continuous)
- Hours of Sleep received the Night before the Midterm (Float, Continuous)
- Finishing Midterm Early Status (Binary Categories)
- Expected Letter Grade on Midterm (Categorical)
- Whether they Faked the Survey Data (Binary Categories)
- Total Score Received & Max Available Points on the Midterm (Both Float, Continuous)

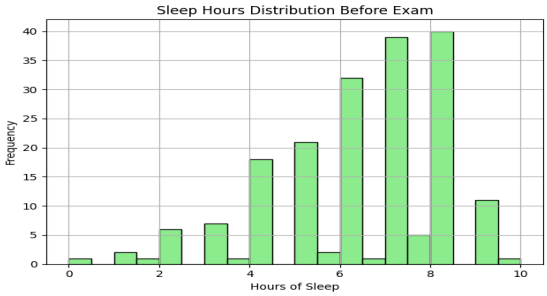
The Fall Class of 2025 had a total of 188 responses to the study/daily habits survey, along with 188 midterm scores.

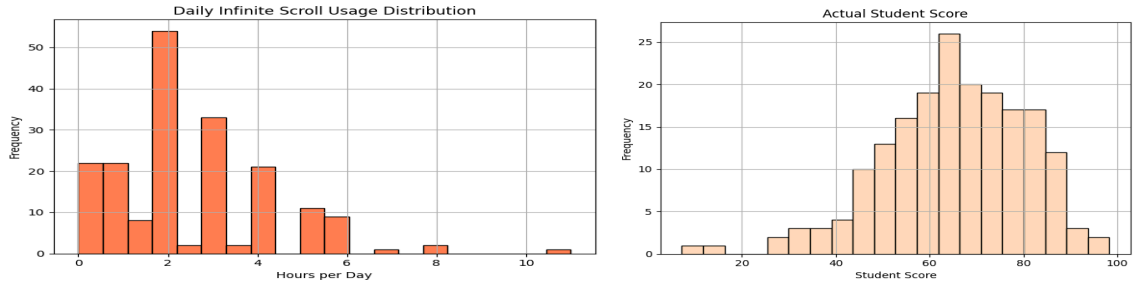
Data Exploration, Preprocessing, & Feature Engineering

Upon processing the dataset, there were some irrelevant features that I dropped, some features in which I replaced their string values with integers for better decision-making during classification, one feature that I created a missing answer column for, and one feature whose N/A answers were valuable – I had to perform imputations on this column. More specifically, I dropped the 'Unnamed: 0,' and 'Timestamp' columns, which I decided were not relevant for predicting student performance in the midterm. I then replaced the yes/nos from the 'Did you leave the exam early?' and 'I wanted the extra credit but just put down random responses (you'll still get the extra credit if you say yes)' columns with binary 1.0s and 0.0s, in order to help my models make decisions (numbers over strings preferred). Interestingly, the 'I've had prior machine learning / data science experience' column had 75 NaN responses, while the 'Which section are you in?' column had 37 NaN responses. After looking through the raw dataset, I found that students LEGITIMATELY answered "None" on the survey for the previous experience question (this was an actual option), which got misinterpreted as NaN in the dataframe, thus leading me to impute the N/As for this column back to the "None" string. Values were ACTUALLY missing for the section question, though, so I had to create an additional 'is_missing_section' column (1s and 0s) specifically to handle this.

To better visualize and understand the data, I also created graphs for the continuous features, as well as the continuous version of the Y variable (student score percentages). There was a strong right skew for hours studied, a moderately strong left skew for hours slept before the exam, a moderate right skew for daily hours spent on sites with infinite scroll, and a pretty uniform distribution of student scores with a mode/center ~63% (there is also a slight left skew, and no truncation of values above 100% since nobody even scored 100%).

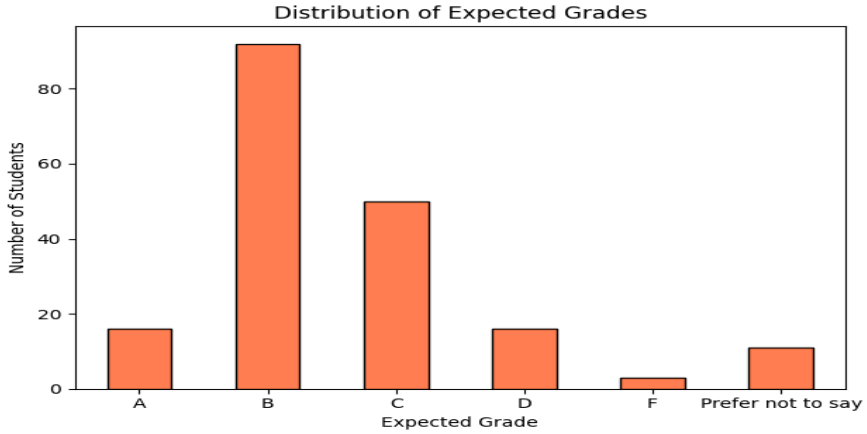






Histograms for the continuous features of the dataset, as well as the continuous version of the output variable (student scores as a percentage)

I was also interested in visualizing the categorical variable of what letter grade people THINK they got – most people thought they got a B.



Histogram for the CATEGORICAL feature of students' expected grades

Additional Notes:

- As for the rest of the categorical variables, I used Pandas get_dummies() feature to one-hot encode them as usable features.
- I removed the Total Score and Max Points features from the X variable, and used them to calculate the respective Y outcome variables.
- I also considered creating a feature correlation heatmap for this problem, but decided against it because most responses were categorical (not good for heatmaps), and also I could tell feature relevance/collinearity anyways by looking at the column title.

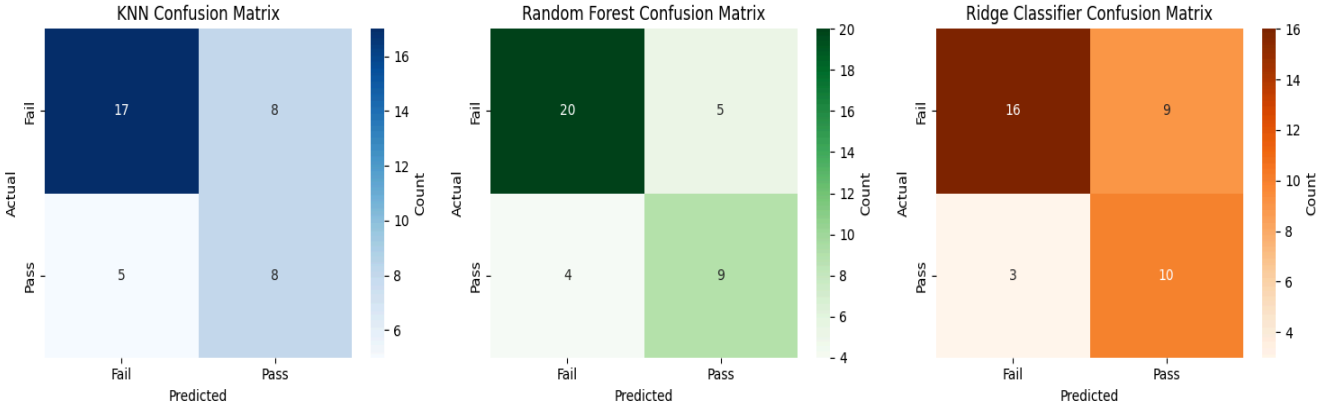
Findings

Training/Splitting for all three parts remained test_size=0.2 for the split and random_state=42 for reproducibility.

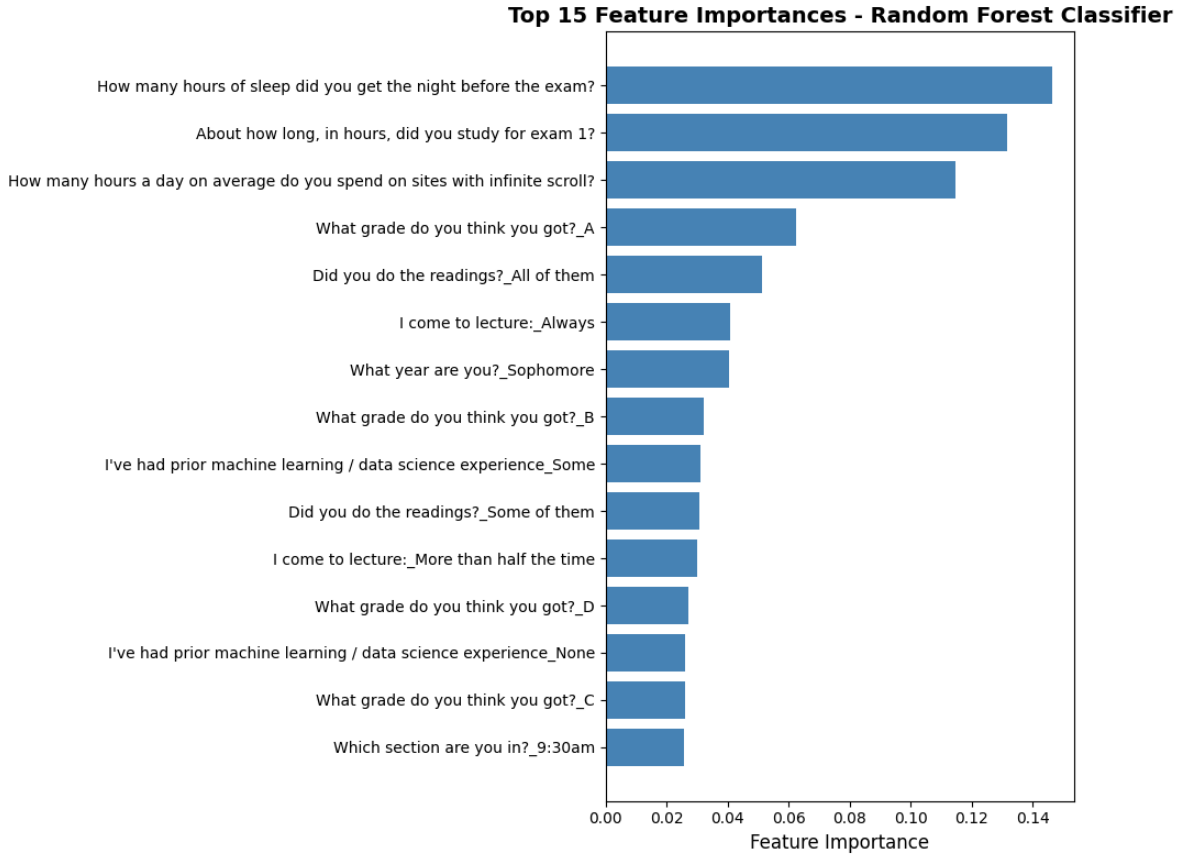
Part 1: I binned each grade into passing (A, B, C), and failing (D, F) as the Y or output variable. Specifically, it was y = ((df['Total Score'] / df['Max Points'])>= 0.7).astype(int). Then, I trained three classification algorithms – K-Nearest Neighbors, RandomForestClassifier, and RidgeClassifier to try and classify students into passing/failing based on their responses to the

survey. The results are as follows (precision, recall, f1, and other metrics can be found in the source code):

- KNN Classifier Accuracy (n_neighbors=18): 0.658 | CV Mean Accuracy (5-fold): 0.670
- Random Forest Classifier Accuracy (n_estimators=20): 0.763 | CV Mean Accuracy (5-fold): 0.632
- Ridge Classifier Accuracy (alpha=1.0): 0.684 | CV Mean Accuracy (5-fold): 0.650
- The confusion matrices for the respective models are as follows:

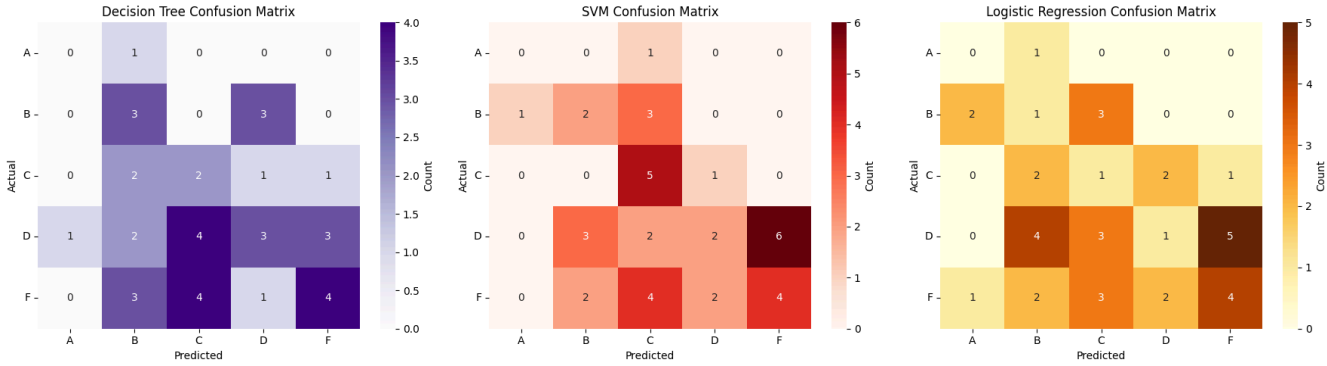


- The feature importances for SPECIFICALLY RANDOM FOREST CLASSIFIER are shown:



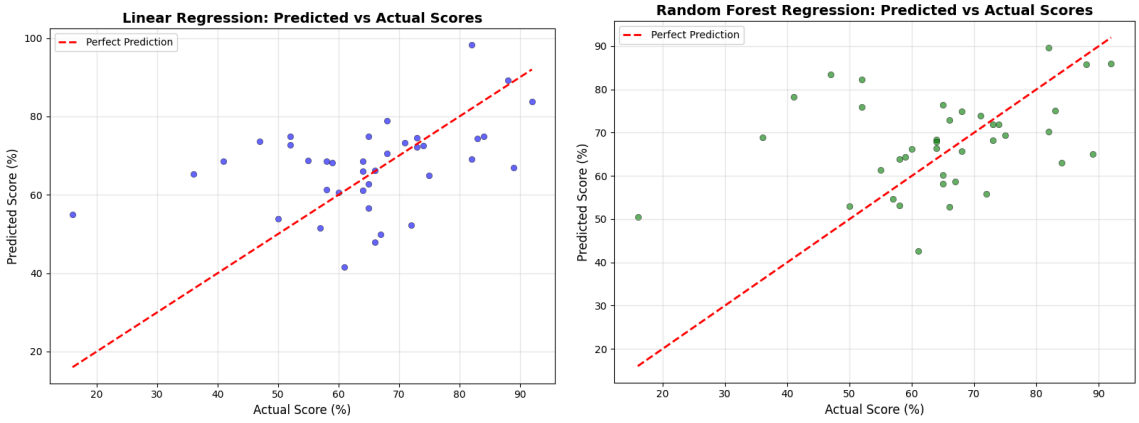
Part 2: I binned each grade into letter grades (A, B, C, D, F) as the Y or output variable. Specifically, it was y = ((df['Total Score'] / df['Max Points']).apply(lambda x: "A" if x >= 0.90 else "B" if x >= 0.80 else "C" if x >= 0.70 else "D" if x >= 0.60 else "F")). Then, I trained three classification algorithms – DecisionTreeClassifier, SupportVectorMachine, and LogisticRegression (still a classification algorithm EVEN THOUGH it says regression) to try and classify students into the letter grade they earned based on their responses to the survey. The results are as follows (precision, recall, f1, and other metrics can be found in the source code):

- Decision Tree Classifier Accuracy (criterion="entropy"): 0.316. CV Mean Accuracy (3-fold): 0.350
- Support Vector Machine Classifier Accuracy (kernel="linear"): 0.342. CV Mean Accuracy (3-fold): 0.399
- Logistic Regression Classifier Accuracy (max_iter=100, class_weight="balanced"): 0.184. CV Mean Accuracy (3-fold): 0.383
- The confusion matrices for the respective models are as follows:



Part 3: Without any binning, I trained the LinearRegression and RandomForestRegression models to try and predict the student's actual continuous grades (as a percentage). Specifically, the output was y = ((df['Total Score'] / df['Max Points']) * 100).round(0). The R^2 is shown here:

- Linear Regression R^2: 0.0399. CV Mean R^2 (5-fold): 0.1648
- Random Forest Regression R^2: -0.0765. CV Mean R^2 (5-fold): 0.1536
- The scatter plots for the predicted vs actual scores (test set) are as follows:



Conclusions

There are a few main points that can be made from this paper:

- In general, across all the models used in this problem, there was simply not enough data to get REALLY good results, whether it be accuracy for classification or R^2 for regression. Regression especially suffered from this issue, and was severely underfitted.
- Binary Classification achieved higher scores than Letter Grade Classification, which achieved higher scores than Exact Regression Matching. This is to be expected, as being more precise with predictions is more hard for a machine to get right than simple binary classification into pass/fail categories.
- According to the feature importances of Random Forest Classification (among the best accuracy that was produced in this paper), the top 5 best possible habits for getting a good grade on a midterm are (in order): Getting Good Sleep Before the Exam, Studying Hours, Avg Daily Hours on Sites w/ Infinite Scroll, Expected Grade of an A, and Doing All of the Readings
    - The other 4 studying habits are to be expected and can be easily and trivially explained to directly impact student grades, however there is also the factor of CONFIDENCE IN DOING WELL of a student that impacts their grade as well (which is more of an effect than a cause of the good grade, but still).