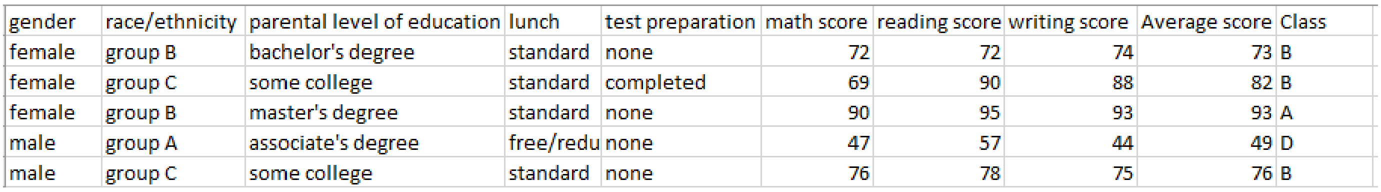
CA 270 - Classification Project – Adam Tegart – 19327493

My Idea: As of starting this project, my idea is to use a mixture of a Bayes classification approach and a Decision tree in an attempt to compare their accuracy against one another. The dataset which I found on Kaggle describes a set of students and contains attributes describing the ethnicity, socio-economic background and academic ethos. The set also includes several exam results for maths, reading and writing respectively, these will be averaged and put into a grade bracket **(A >= 85, A > B >= 70, B > C >= 55,C > D >= 40, Fail < 4**0). I will then try and use the classification algorithms mentioned to predict the grade bracket of a student given only the other attributes.

My Dataset: The dataset which I got from Kaggle is available here. It contains information on various students and attributes that I believe may help to classify them into a grade bracket. Below is a snippet of the first 5 tuples of my dataset and all attributes associated with them.



gender	race/ethnicity	parental level of education	lunch	test preparation	math score	reading score	writing score	Average score	Class
female	group B	bachelor's degree	standard	none	72	72	74	73	B
female	group C	some college	standard	completed	69	90	88	82	B
female	group B	master's degree	standard	none	90	95	93	93	A
male	group A	associate's degree	free/redu	none	47	57	44	49	D
male	group C	some college	standard	none	76	78	75	76	B

The gender of student and their ethnicity I don't believe will be too useful, but only analysis will tell. The other attributes seem promising however, a student who has parents with a high level of education will have been brought up being told of the value in education. The next attribute refers to a student's socio-economic status, disadvantaged students may not have the same opportunities as other students which could affect their grades. Lastly, and possibly most important, the students level of preparation for the exam.

A column which gets the average of a student's grade and a column containing the corresponding grade has been added to each tuple. This will make it easy to form a classification algorithm from as I have the data set out the way I need with the class of each tuple already classified.

The Plan: The plan for this project is to do majority of the heavy lifting through both Python and R. Python is a language I am familiar with and it means I can understand fully what is happening at each stage as I am writing the algorithm myself. I have a limited amount of experience with R, but I do know how useful it can be when calculating different attributes the data may have. This will hopefully allow me to find out some useful and interesting information regarding my data.

The analysis of my data will consist mainly of graphs and the insights that they gave me, along with any other general insights I get from combing the data.

The explanation of my algorithm will be separated into two parts, one for the Bayes algorithm and then one for the Decision Tree. It will consist of an explanation of the algorithm and how I used it.

The results will be displayed as an overall accuracy of each algorithm, and an explanation as to why that is the case. I will also look at any errors that arise from my algorithms and talk about how in future I can avoid these errors.

Overall, I feel I will be pleased with the results and (hopefully) working algorithm.

Analysis of the Dataset

Over the following three pages I will look at different aspects of the various attributes in the dataset and with the use of graphs and calculations I will show the trends within these attributes. As the main attributes I shall be classifying tuples with are categorical, all of my graphs will be bar-charts.

Gender: There was a split of about 48/52 between males and females respectively. The bar-chart of this data can be seen in Fig.1.1 below. Along the x-axis there are bins for each possible grade received and on the y-axis there are no. of students. Looking at the graph it can be seen that more females than males received an A, even taking the weighting into account from the additional 40 students (520 female - 480 male), females still had more A's with 15% getting an A compared to 9% of males getting an A.

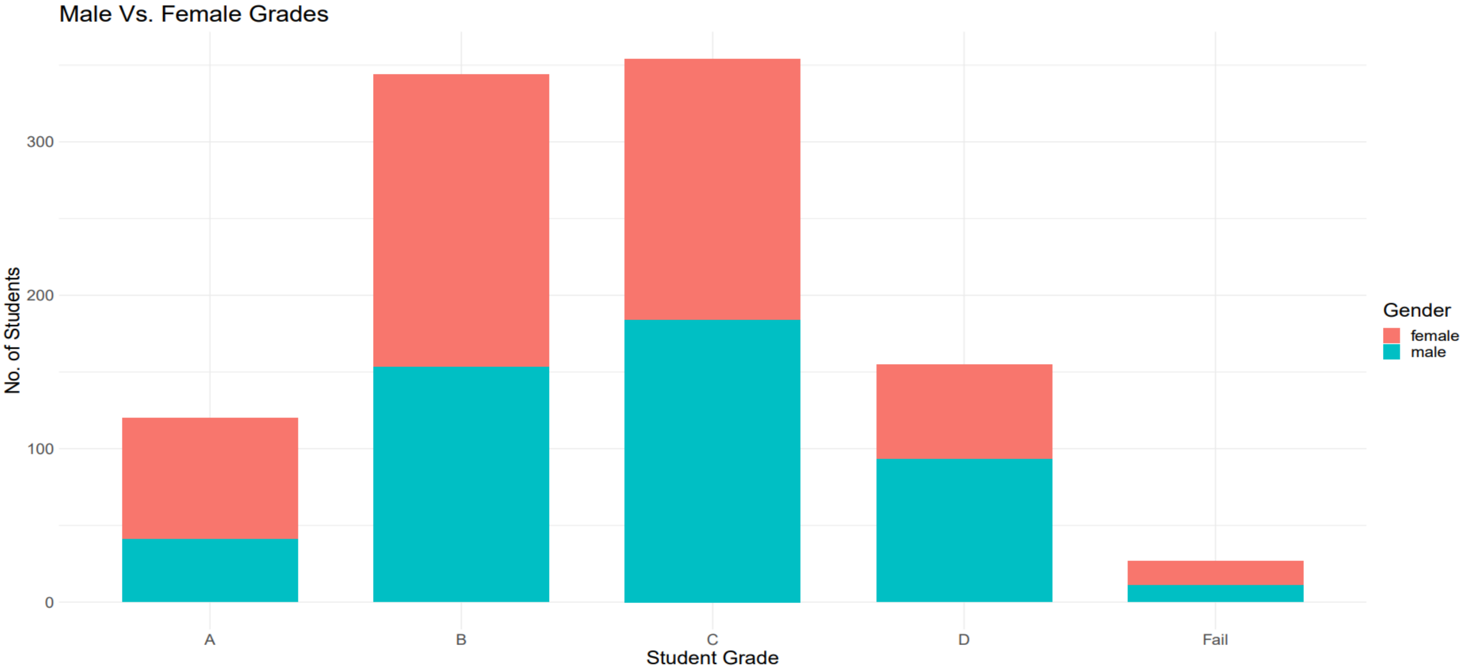


Fig 1.1. Bar-chart showing all grades and Male vs. Female split in each bin

It can also be seen from the data that there are more females failing than males. This to me shows that the male population is more consistent with regards to grades with 90% of the population within the B-D bracket. Females on the other hand seem to be more spread out with more high achievers (just over 50% got a B or higher) and more fails than the male population. Overall, it looks as though gender may be an attribute that could help classify my tuples.

Race Group: There are 5 different possibilities for this attributes, groups A through to E. This will allow for all groups to be considered equally. Firstly, it can be seen that Group E has a large majority of the A's with a small overall group size of 140, the 2nd smallest (Group A has 89). Within Group E, 22% of all students from that group got an A, which is double any other group. The majority of students within the group got high or at least fair grades with 1% failing and just over 10% getting less than a C. This may prove handy for classification as students from group E will be assumed to pass 99% of the time.

On the contrary to this, Group A which was the minority, had 70% of students getting less than a B. This was the lowest scoring group as other groups had 60% of the population or less scoring less than a B.

Group C and D had very similar distributions and were what you would usually expect for a spread of test results. Group B performed a little worse than C and D and they had 4% of students failing but had a better spread than Group A.

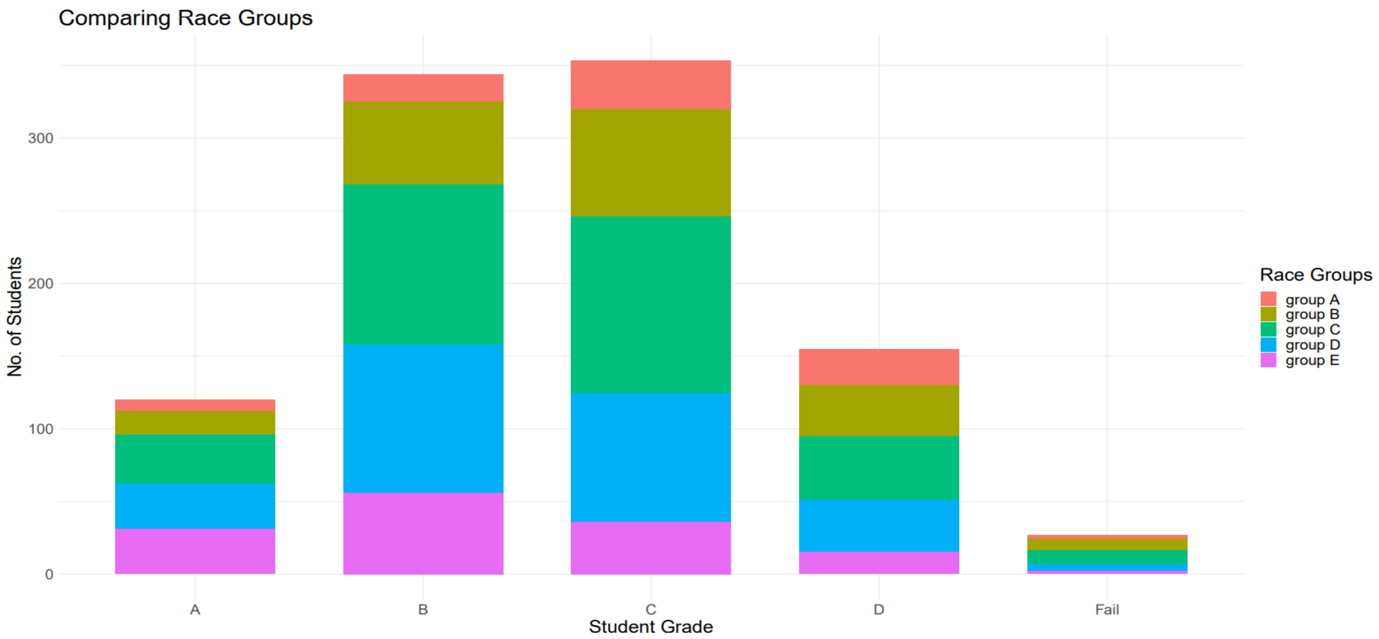


Fig 1.2. Bar-chart showing all grades and Race Group splits in each bin

Parental Education Level: I thought that this attribute would show some clear trends and I was not disappointed with my findings. Students whose parents had completed college had a much higher chance of getting an A or B. Students whose parents had a master degree had a 30% chance of getting an A and 40% of getting a B. In total 2 students failed who had parents finish college, so there was a 0.5% chance a student would fail if their parents fell into this category.

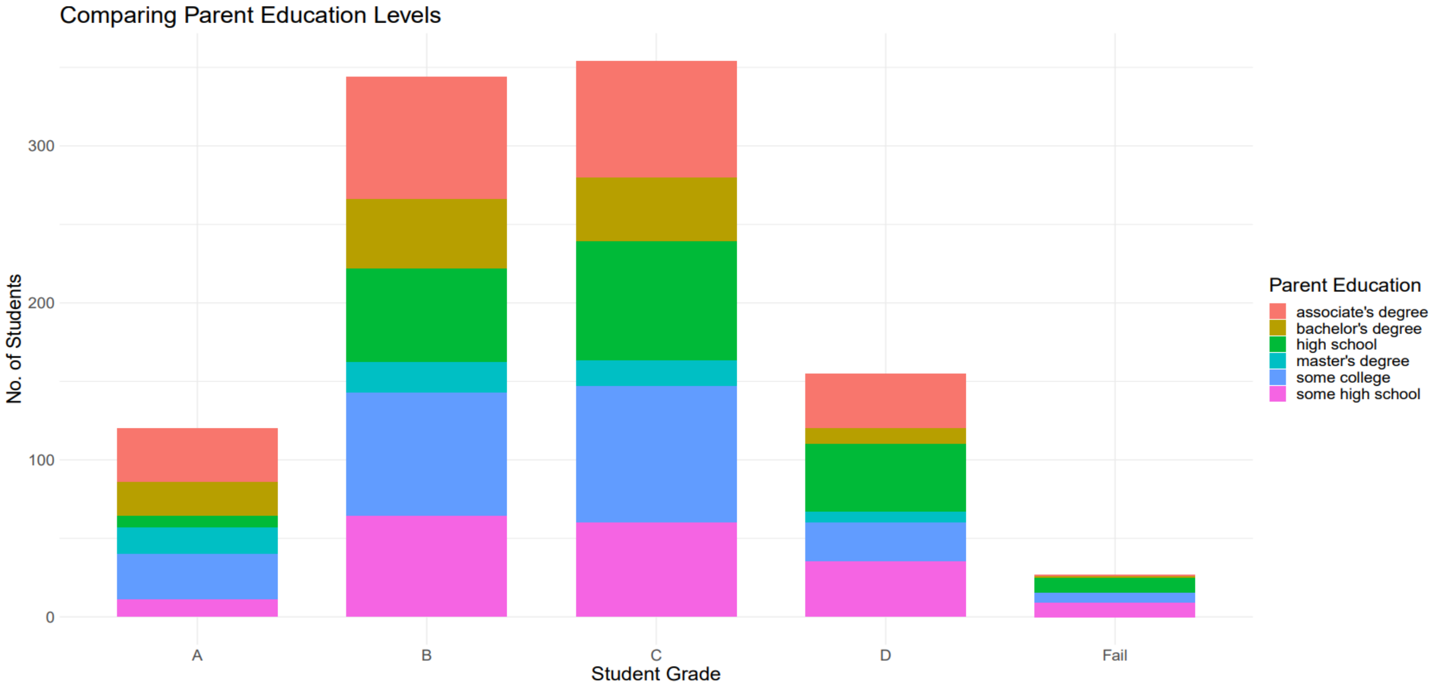


Fig 1.3. Bar-chart showing all grades and the split within each bin for parental education.

This was very clearly different to the students who had parents who dropped out of college, only completed high school, or didn't complete high school. The chance a student would fail went from 0% if a students parents had a masters, to 0.5 – 1% if they had another college degree, to 5% if their parents hadn't finished in college or only attended high school.

This shows to me that students who had high achieving parents from an academic standpoint were more likely to do well with regards to grades. Some students still got A's even if their parents didn't do great academically, and I believe this may be related to their parents pushing them to achieve more than they did.

Overall, I believe that the parents level of education will most definitely be vital to classifying a students grade without their actual result.

Student Lunch Package: I was unsure as to whether or not this attribute would be too useful with regards to classifying, but after analysis this attribute seems much more promising. This attribute refers to whether or not the student has paid for a standard lunch packet in school, or whether they have just received the one provided by the state free of charge. This means that it can represent the socio-economic status of a student and also whether the nutrition they get can affect their ability to retain the information.

The data shows that the students are split 64-36 with regards to standard lunch and free lunch respectively. Students who had the standard lunch had a 16% chance of getting an A vs. a 5% chance for students with the reduced/free lunch. Students with the reduced lunch were also 6 times more likely to fail. Only just over 10% of students with the standard lunch got less than a C, this was 30% for the case of the free/reduced lunch students.

This is most definitely as a result of the difference in socio-economic status, students from a higher class background have better opportunities such as tutors, better conditions and also less pressure to contribute financially to their family. It is for this reason I believe it will be a useful attribute.

Test Preparation: This is an obvious one, but a students preparations for an exam will most certainly effect a students result. 36% felt they had completed their preparations while 64% said they had done no preparation. There was a clear difference between the 2, students who prepared were twice as likely to get an A and 4 times less likely to fail their exam with regards to the conditional probability of a student failing given they prepared. 90% of these students had a grade of a C or higher.

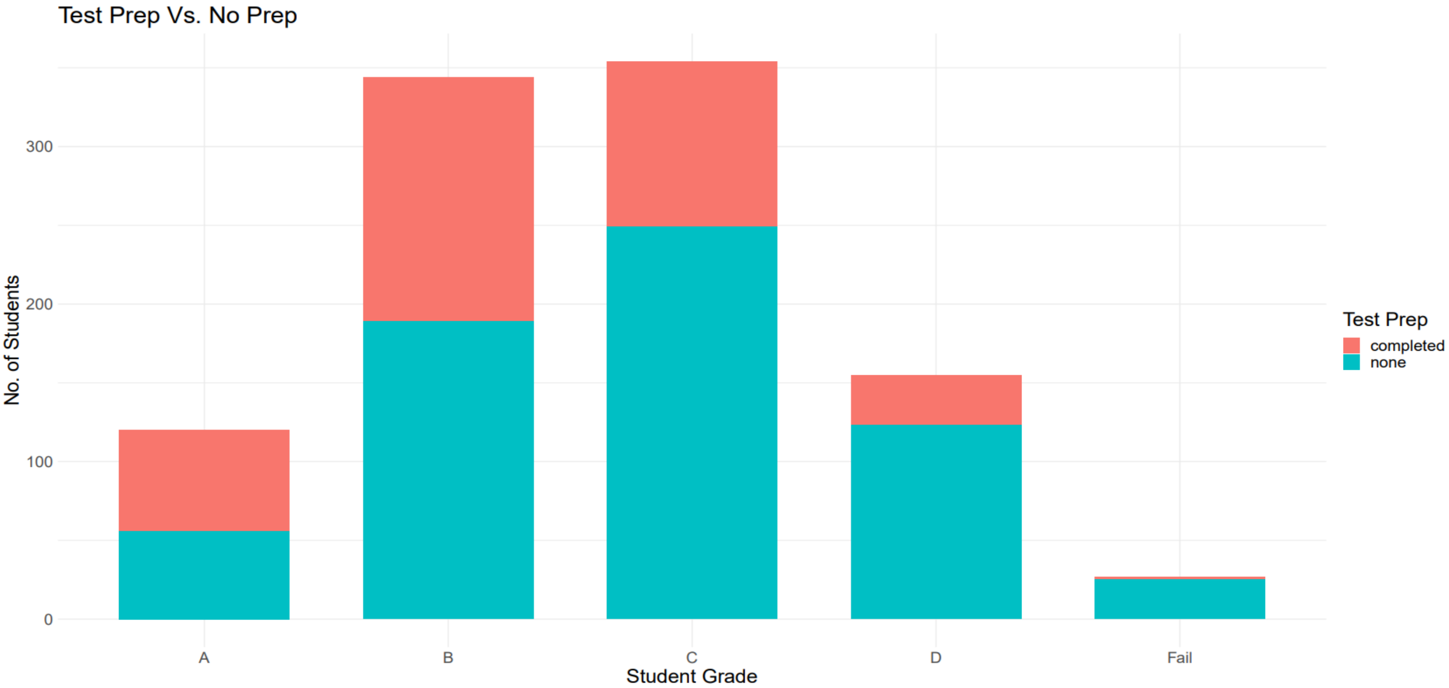


Fig 1.4. Bar-chart showing the grade bins and the split in each which corresponds to test prep completed and no prep done.

This is of course what I was expecting when I first skimmed the data, but it is nice to know that this will be useful in classifying unknown tuples.

Conclusion: The other attributes which are present in my dataset include 3 attributes for a score in reading, writing and maths respectively. To summarize these, maths was the test which students found the hardest and reading seemed to be the easiest. These however will not be of use to me as they were used to determine the class directly and will not be used to classify unknown tuples.

Overall, I believe I will have a lot to work with using the 5 attributes analysed above and I have a good feeling about the accuracy of a classification algorithm based on these attributes.

My Algorithms of Choice

Bayes Classifier: I think the use of a Bayes classification algorithm will be both easy to implement and will also be effective with the dataset that I am using.

A Bayes Classifier algorithm makes use of conditional probability to estimate which class is most likely based off the conditional probabilities computed from the training set. An example of this in my case may be, which class does a student belong to if they are male, in Group A, parents dropped out of high school, they have a free lunch and they didn't prepare for the test. For each class a probability is computed to show how likely it is this student belongs to that class. We then simply pick the most likely.

So lets say the student has a 0.1 probability of getting an A as a male, that is multiplied by 0.05 for getting an A in group A and so on. This is of course repeated for each class and the most probable outcome is chosen. This will be my approach to a Bayes Classification algorithm.

I will use python to compute the probabilities and output the class the algorithm believes the student to be in beside the class which they are actually a part of. This will allow me to quickly compute the accuracy.

Decision Tree: I will compare the accuracy of my Bayes Classifier algorithm against a top-down induction decision tree algorithm. I hope that this will prove insightful and hopefully show which would be the better choice for a classification algorithm for this particular dataset.

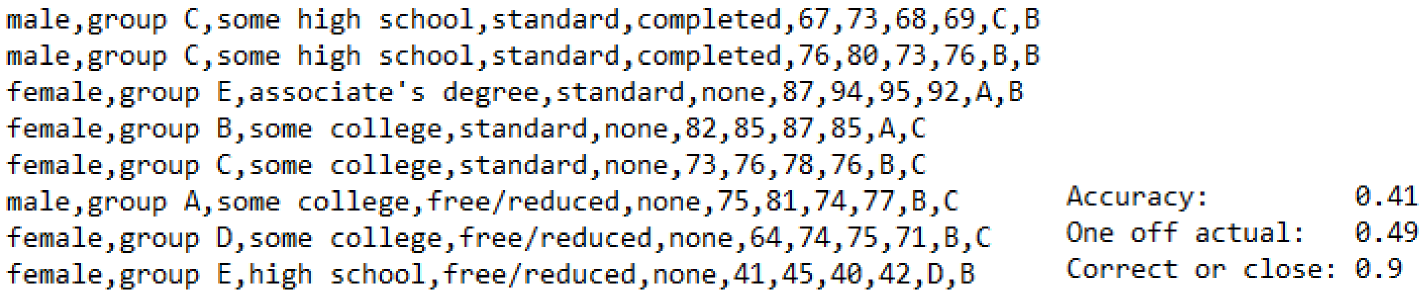
A decision tree algorithm works by finding a splitting criterion on which to partition the tuples in the dataset. The aim is to make a tree using a test set which becomes more pure as the partitions go on(Pure means that all the tuples in a partition belong to the same class). In this way, all tuples residing at a leaf node are given a label and in this way, classified.

I will be making use of some code that I got online via Google Developers. I decided to use this as I feel it will be the best use of my time for this project. The program works by making binary splits on the splitting criterion that will result in both the biggest information gain and lowest Gini index(impurity). This then becomes the node and a split occurs to the rows of tuples, recursion is used to work down through all the branches. I made some changes to accommodate classifying tuples how I'd like and computing the accuracy.

The questions to split on are True and False ones relating to the attributes. An example would be, "Is the student from race group A". In this way, the tree may become overfitted, but I will address this later.

The Results of My Algorithms

Bayes Classifier: My Bayes Classifier Algorithm was easy to implement and record the results of. After reading in the training set of 800 tuples it classified 200 tuples with an accuracy of 41%. A snippet of the results obtained can be seen below in Fig 2.1. A tuple is shown containing all of its original fields, at the end there has been a new grade appended. This is the predicted grade. To the right is a small summary of the algorithms aggregated results across all the tuples.



```
male,group C,some high school,standard,completed,67,73,68,69,C,B
male,group C,some high school,standard,completed,76,80,73,76,B,B
female,group E,associate's degree,standard,none,87,94,95,92,A,B
female,group B,some college,standard,none,82,85,87,85,A,C
female,group C,some college,standard,none,73,76,78,76,B,C
male,group A,some college,free/reduced,none,75,81,74,77,B,C       Accuracy:          0.41
female,group D,some college,free/reduced,none,64,74,75,71,B,C     One off actual:    0.49
female,group E,high school,free/reduced,none,41,45,40,42,D,B      Correct or close:  0.9
```

Fig 2.1. Snippet of Bayes Algorithm Results.

It can be seen in the summary of the above fig that on tuples it didn’t get correct, it was only one grade off about 5 out of 6 times(As it is incorrect 1 – 0.41 = 0.59. So given that it classed it incorrectly, what is the probability of it being one off. 0.5 / 0.59 = roughly 5/6). So it was nice that the label I was classifying on was ordinal categorical data and had a natural ordering.

My personal opinion on the results was of course that I was a little surprised that the accuracy was very low. I had expected the algorithm to work well, and it did, but there was another issue. As can be seen in the first 2 tuples in Fig2.1. there is a slight issue. Both tuples contain the exact same attributes as each other, but they don’t have the same label. This is a big issue for any classification algorithm, not just a Bayes algorithm. The algorithm correctly guesses one of the 2 tuples, and the other is only 1% off a B! The issue when working with datasets like this is that attributes don’t always define an object, especially when the object being described is a human. This is the reason why I decided that having a percentage to show both when it is correct and when it is only one grade off would show that the algorithm was at least close. It’s not as though it just picked a random label to give to a tuple. Theoretically, picking randomly for every tuple would give an accuracy of 0.2 for any given tuple(5 labels and only one is the correct one, 1/5 = 0.2). I believe an accuracy of 0.41 is a big improvement and given that there is no way of getting an algorithm to have a 90-100% accuracy due to tuples with the same attributes having different classes, I don’t think the algorithm was half bad.

Another issue with the algorithm was that due to the fact an A grade was rare, the algorithm almost never predicted an A. Even when I introduced some slight weighting to put emphasis on attributes like Parental education and Race group, the results were unchanged. The following tuple was present in my data

female, group E, master's degree, standard, completed, 88, 99, 95, 94, A, B

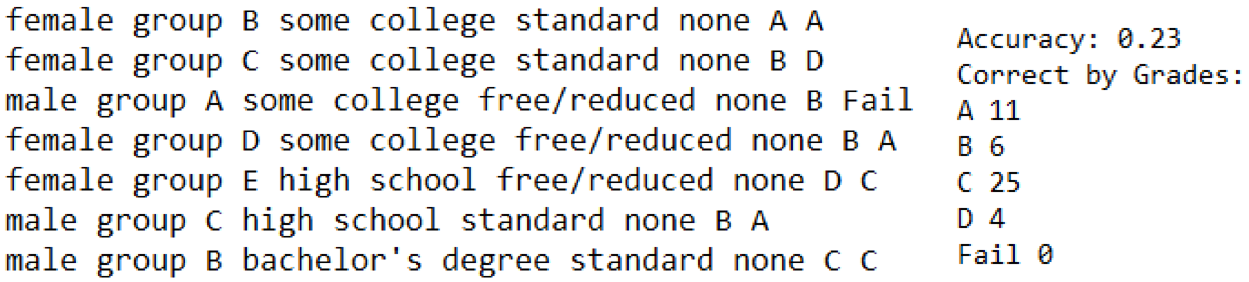
This tuple was by far the one which should have been, in the eyes of the algorithm, predicted an A grade. Race group E had a minority of the students, but all of these students performed well academically. Likewise, students who had parents with a Master’s usually performed well. The algorithm however still predicted that this student got a B.

One thought I have on this is that I could specify to the algorithm that there must be a certain percentage of students who for example have parents with a Master’s that get an A. This would of course use more memory as every probability for a students in this group to get an A must be recorded. But it would solve the issue of no A’s being received. So let’s say there are 60 students in this group and typically a student with a parent who has a Master’s gets an A with probability 0.1, then the 6 most promising students will receive an A.

Overall, I am happy with the outcome from my Bayes classification algorithm.

Decision Tree: The algorithm for the decision tree that I used came from here. I made some changes to fit my dataset, but the algorithm operates the same. I decided not to write it from scratch as I only needed this algorithm as a comparison to my other one.

The algorithm itself had an accuracy of 23% when working on the 200 unseen tuples. I implemented a way of taking the most probable prediction as the algorithm outputted a set of the labels and a count for each. A snippet of my decision tree results is shown below in Fig 2.2.

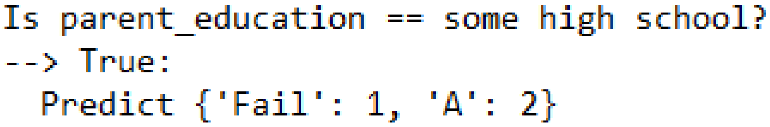


```
female group B some college standard none A A
female group C some college standard none B D
male group A some college free/reduced none B Fail
female group D some college free/reduced none B A
female group E high school free/reduced none D C
male group C high school standard none B A
male group B bachelor's degree standard none C C
```

```
Accuracy: 0.23
Correct by Grades:
A 11
B 6
C 25
D 4
Fail 0
```

Fig 2.2. Decision Tree results and summary

It can be seen from the results that this algorithm has the capability of predicting an A grade for a student, which the Bayes algorithm struggles with. It also seems that the algorithm cannot correctly predict when someone fails. Looking at the list of predictions it frequently predicted that someone who got a B or C would fail, so although it can predict an A, it doesn't have the same sense of reliability that its prediction is at least moderately close. Upon looking at the set of predictions at certain leaf nodes I was quite surprised. A snipped of one such set of predictions at a leaf node can be seen below in Fig 2.3.



```
Is parent_education == some high school?
--> True:
  Predict {'Fail': 1, 'A': 2}
```

Fig 2.3. Certain prediction for a path in my decision tree.

As can be seen, this set cannot be partitioned anymore and is left as it is. My changes mean that it will classify any tuple that falls in there are an A, but there was a fail in the training set that managed to make its way in there. This would be more acceptable if it was a B or C instead of a fail, but there is a big difference between an A and a Fail.

I believe that this algorithm shows just how good the Bayes algorithm performed on this dataset as it is only half as accurate and is only marginally better then just randomly assigning a label to any tuple. This is again as a result of the certain element of randomness among the tuples. 2 tuples may have the exact same attributes, but they have a different grade as people can have both bad and good days when it comes to taking exams.

The only improvement I can think of would be to change the splitting from a binary split to a splitting method that would partition the set into several sets, one for each value of the attribute. I think this would be beneficial as the tree would be a lot easier to visualize compared to the current one. This would make it much easier for me to walk through the tree and understand every path a tuple could take down the tree.

Conclusion: Although my attributes to classify on seemed promising, I have learned something very important. People can be described by their attributes, but you can't predict their performance with any sort of accuracy however as a persons capability is not captured well on paper. In future I will always be expecting a lower than normal accuracy when working with data relating to people and their performances.

Looking at both algorithms I can now say that although both were not very accurate, the Bayes algorithm was a lot more accurate than the decision tree. My understanding as to why this is the case is that the decision tree was affected more so by these tuples that are identical in all attributes but the label. This lead the tree to have several wrong predictions, this along with the fact that tuples had some degree of randomness in how they were labelled led to a low accuracy.

The Bayes algorithm on the other hand didn’t work off any one path to a prediction. The fact that it looked at the probability of getting a certain grade for any attribute value meant that it is less likely to be influenced by a handful of random tuples. So for this reason I believe that a Bayes algorithm is more robust to this certain decision tree algorithm.

As a result of this project I now understand that sometimes just because you know the attributes it doesn’t mean you’ll be able to predict a label for certain tuples. Especially when 2 identical tuples have 2 different labels! In future I will always make sure to check that identical tuples are labelled the same, otherwise I will know not to expect a perfect accuracy.

To round up my conclusion, I was thinking this whole exercise of classifying this particular dataset was very similar to the recent news about issues with predicted grades and the algorithm that computed them. Having finished this project now, I know that grades are very difficult to predict accurately. Just because on paper 2 students are the same, that doesn’t mean they have the same capabilities. People are complex, while computers are simple. I don’t think a persons future can be decided by any one algorithm, especially going by the ones similar to the one I’ve been working with!