(25%) Given the data set, do a quick exploratory data analysis to get a feel for the distributions and biases of the data. Report any visualizations and findings used and suggest any other impactful business use cases for that data.

There are 16 Majors reported, 4 Years, 10 Universities, and 10 Times. Of course, there do exist more than 16 college majors and 10 universities, so this data can only provide information on students at those 10 universities and of those 16 majors.

There are disproportionately few students of Year 1 and Year 4, so the model will likely be less accurate for these students. Also, some Universities only had a few hundred students in the data set, while others had over 1000. The same was true for Majors. This could lead to discrepancies between the model's accuracy rates for different categories.

(30%) Consider implications of data collection, storage, and data biases you would consider relevant here considering Data Ethics, Business Outcomes, and Technical Implications

Discuss Ethical implications of these factors

Though things like year and major are not sensitive information by any means, there are plenty of people who don't want AI to "know" anything about them whatsoever – thus, we'd need some way for customers to opt out, even if they are eligible to participate. The General Data Protection Regulation (GDPR) outlines 7 key principles for data ethics, and it would be important to look over them thoroughly before implementing a program like this. I could see a program like this eventually selling the data it collected to other companies; similarly, FoodX might be tempted to use it as a way to engineer menu items that are more profitable to a larger number of students. Both of these are violations of the GDPR's 'purpose' principle, and would be considered unethical data usage.

This data set is minimal – it collects only the factors relevant to the program, and doesn't include names or similarly identifying information, meeting the 'minimisation' principle. It is also anonymous, which is good for privacy but less good if users want their data to be deleted, as is their right according to GDPR. However, this could probably be solved by associating each datum with a user ID that FoodX cannot index with.

Discuss Business outcome implications of these factors

Obviously, the better the model, the fewer 10% discounts we must give to customers. That means we'd need to get the model's accuracy rate far higher than my 55%-60% to make money – likely anything below 90%-95% accuracy would lose FoodX quite a lot of money, and it would be hard to get scores like that without a data set much larger and more diverse than the 5000 samples provided. I'll discuss this issue below as well.

Not everyone is a big fan of AI these days – people are scared of losing their jobs, and implementing an AI solution to explicitly replace a job that was once given to your employees might drive them away, even if it's a job they don't particularly like. This could cause problems with respect to employee retention and hiring.

Discuss Technical implications of these factors

We'd need to determine the exact method of data collection, as discussed above – will we have employees continue guessing for the pre-AI program and use the data students provide them to compare against, or simply ask users to input the data for this purpose (the more ethical option)? Also, our data already shows huge discrepancies in terms of the number of samples we have for each feature value, which will lead to bad predictions – there are a disproportionately large number of Indiana State University students, for example, and disproportionately few Evansville students, so our AI will likely be relatively good at predicting the orders of ISU students and relatively bad at predicting the orders of Evansville students. However, the number of students at each of these universities does vary widely, so providing the AI with relatively few examples of, say, Northwestern students, will tell it that more likely than not, they are dealing with a student from a different university – we'd just need to make sure all those numbers are proportional, perhaps with things like SKLearn's scalar functions.

(35%) Build a model to predict a customer's order from their available information. You will be graded largely on your intent and process when designing the model, performance is secondary. It is strongly suggested that you use SKLearn for this model as to not take too much time. You may use any kind implementation you would like though, but it must be pickelable and have a ".predict()" method similar to SKLearn

I chose to use an MLPClassifier, mostly because we've been studying MLPs in one of my courses and I thought it would be good practice to write one. However, I realized once I finished writing the MLPClassifier version that there exists a CategoricalNB classifier, which might have been optimal since I wouldn't have had to preprocess the data with LabelEncoder. Alas – we live and learn.

Preprocessing was a bit of a challenge, since the features in this data set are nominal (except for Time, which I just didn't use LabelEncoder on), and the MLPClassifier can only take numeric values. SKLearn provides a few functions that help us process categorical data numerically: OneHotEncoder, OrdinalEncoder, and LabelEncoder. I used LabelEncoder since the features we were given weren't ordinal or scalable, but rather discrete labels with no inherent relation to each other. I guess Year is an exception to this, but le.fit_transform() would still turn those values into int64s, and the model doesn't need to associate meaning with those values as long as they remain consistent with the arbitrary values provided by LabelEncoder, so I didn't feel the need to complicate the code with another encoder. When I printed the values of Year after running LabelEncoder, I saw that they were indexed with Year 2 ⇒ index 1, but again, the model doesn't care about that as long as those values still exhibit the pattern it's looking for.

I used an 80/20 split for training and testing data – that is, I trained the model on 80% of the data (4000 data units), and tested it with the last 20% (1000 data units). I know it would've given me better results if I'd used cross-validation, but I am unfortunately not knowledgeable enough about how to use SKLearn to implement that yet – or at least, it would've taken me longer than a weekend to figure out how to do it. If I had been, though, I would have first split the data into 4 groups, by the feature with the fewest extant values (Year, 4 possibilities). Then, I would've split the dataset again by the feature with the most values (Major, 16 possibilities).

(10%) Given the work required to bring a solution like this to maturity and its performance, what considerations would you make to determine if this is a suitable course of action?

First, we'd need to consider whether or not the increased business we get from this solution will pay for the labor cost. Programmers are expensive, and creating a model that would perform well enough to not lose the company money in the long run would take a lot of data – likely more than the 5000 unbalanced samples we have. Moreover, that data collection process would require either the continued use of our dissatisfied analog guessers, which could result in employee loss, or the use of a young model trained only on that data set, which – though it might be amazing at predicting the orders of Year 2 or 3 students from Indiana State University – would likely struggle to classify students in other categories, and lose us even more money.

It's also important to remember that this program is designed exclusively for college students. This means that, no matter how good our AI model gets, the new customers it will pull can only come from that relatively small population – and (at least in my experience), college students are largely susceptible to cheap and lazy advertising anyway. If this solution were to be implemented and remain specifically targeted towards college students, I don't see it earning FoodX much more money than another, cheaper advertising campaign – an ice cream truck-esque tune, for example, would probably be similarly effective.

However, business is all about risk and reward. If the model gets good enough, FoodX can ride the AI hype wave – with the general population's growing unease about AI, it could be a fun challenge for customers to "beat the system" by ordering something unexpected for their demographic. That could give us bad data though, so it might be best to save this marketing campaign for when the model has already eaten however much data we're planning to feed it.