Part 1. Data Visualization

Do a quick exploratory data analysis to get a feel for the distributions of the data.

Justification

I started off this prompt by doing some basic data analysis, so that I could find any biases of the data. This would allow me to choose between Scikit-Learn's many models, based on how strong the correlation between data points would be. To visualize the data, I downloaded the data set as a .csv file and started working on a python program – you can find the results and code as PDFs under the "Raw Data Visualization and Code" folder.

I chose to use the python language because of how well it manipulates data objects and it's ease of use. Python not only has access to efficient libraries useful for ML and data manipulation (like scikit-learn), but the syntax is also very trivial. I used modules like PdfPages and Pandas to create graphs relatively easily and store them as pdfs.

Process:

1. I first plotted all the different data variables (year, major, university, time, order) in the entries so that I could find the significant groups. This allowed me to find which groups were the largest, as it's best to use a larger sample size when trying to find correlations between a variable and the order. This also allowed me to look at demographics, which helps find any business use cases with the data.

   This data can be found in **Variables_To_Count.pdf**

2. Taking the larger groups for each data point, I plotted out the correlation of that specific group to orders. For example, Chemistry majors to their order, IUPUI students to their orders, etc. I only took the larger groups so that I would be able to find statistically significant outcomes, which part (1) helped me narrow down.

   This data can be found in **Orders_By_Year.pdf**, **Orders_By_Major.pdf**, **Orders_By_University.pdf** and **Orders_By_Time.pdf**.

Outcomes - *Report any visualizations and findings used and suggest any other impactful business use cases for that data.*

I found that there were statistically significant correlations between all four other variables and the order type. For example, looking at the data from majors, the most ordered menu item changed significantly between majors. Mathematics majors had a clear preference for the "*Ultimate Grilled Cheese Sandwich (with bacon and tomato)"*, with the number of orders for this item almost doubling the next most popular order. Likewise, other majors preferred the "*Indiana Pork Chili", "Sugar Cream Pie",* etc, which showed a **clear cause-and-effect** between the major and their order.

The same was true for time, with the most likely order shifting from the "*Breaded Pork Tenderloin Sandwich*" to the "*Indiana Corn on the Cob (brushed with garlic butter)"* to other menu items throughout the day. Again, there was a **clear correlation** between time and the order. This was also true for universities, and for the year that the students were in.

All of this data can be accessed in the RawData Visualization and Code folder. This data can not only be used for this ML project to try and predict what students will order, but overall to suit the demographics. I think that the **time** and **university** variables have the most potential in business use cases. This is because these variables are easiest to distinguish – you can't cater to specific majors or years in a cafeteria, but you can cater to specific universities at specific times.

Business Use Case: A food catering company could use this data and adapt to the preferences at specific universities and specific times – i.e. predict how much to produce at certain times, right before those menu items are most popular. This would allow you to minimize food waste and have fresher food when most students want it.

For example, serving more of the "*breaded pork tenderloin sandwich"* in the morning (9-10) would be most popular, followed by the "*corn on the cob"* in the afternoon (11-12), the hoosier pulled pork sandwich (13-14) and so on.

Part 2. Implications of Data Collection, Storage and Data Biases

---

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

There are a few implications of data collection, storage and biases that we should consider.

Ethical:

- Is predicting orders based on personal information ethical?
  => Do models perpetuate biases? For example, racial stereotypes?
  => How many variables is too much to collect? Which variables are too far?
  => Should we even collect this data in the first place?

- Is there a security risk or risk of abuse with the personal information of students?

- Do customers appreciate being predicted?

  => Does the model negatively affect their experiences?

Business Outcomes:

- What is the expected ROI on a model, based on the increased costs of data storage/collection, maintenance and creating the model itself?

- What are potential risks here from bugs? E.g. wildly incorrect predictions.

- Are there data biases that stop profits? For example, do certain items run out, leading to them not being seen as popular?

Technical Implications:

- Is a solution flexible? Does adding menu items mess with the data?
- Can models be maintained properly?
- Is security feasible?
- Is a solution scalable? Is the model efficient for large scale implementation?

Part 3. Model and Design

Build a model to predict a customer's order from their available information.

Justification

I chose the KNN (k-nearest neighbors) model to create predictions for this prompt. The reason I chose this was because it can be used as a classification algorithm. As we've shown in part 1, the four variables of year, major, university and time have a strong effect on the order. This means that we can make groupings, or classifications, and similar points should be found close to one another. Hopefully, this means that people with a certain major, university and time will be very close in the KNN model and an accurate prediction can be made based off others.

The KNN model is also historically used for simple recommendation systems like this one (according to IBM). This means that there is precedent for usage in cases like these. Furthermore, the data set for this model shouldn't be too large, so the usual drawback of KNN becoming inefficient due to its lazy-learning model (doing the computations when predictions are being made) should be mitigated.

I chose to split up the data points, since reusing them in training and testing would skew the accuracy upwards. We instead want to simulate predicting entirely new customers. I gave the majority (75%) to training the model and the rest to testing it. I chose this percentage because, when incrementing values of the size of the data set, I found 0.25 (25% in the testing set) to return the highest accuracy with the specific random state.

Process / Explanation of Code

My python code is somewhat simplistic. It does the following steps:

1. Load in the dataset in a .csv format and encodes the variables using LabelEncoder.
2. Splits the dataset into two – 75% to train, 25% to test.
3. Runs the scikit-learn KNN model on the dataset.
4. After training, it tests itself on the remaining 25% of the model to calculate metrics. The most significant metric is the accuracy that it predicts the rest of the data with.

The model code can be found inside the **ML Model** folder. In the end, my model predicted orders with a 62.64% accuracy.

Part 4. Evaluation

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?

Implementing a machine model like this to a point where it is viable for business would take a lot more work, since there are a lot more considerations that could upset a machine model trained on this unspecific data. My current model can only predict what a student will order around 63% of the time. It also cannot account for other events that will change what students order. For example, what if there's a holiday, and most students are on vacation? I'd say that a lot more work is required to have a prediction accuracy of 90%+, which may not even be profitable.

There are a few limitations with the current implementation, which I would need to make considerations for. The minimum I would do before implementing a solution like this is:

- **Increase the accuracy of the model**. This can be done by getting a larger data set or taking into account more variables.
- **Find the market demand.** Is there room for expansion, i.e. other campuses or narrowing down the menu item surplus? How much does predicting the students' orders really contribute to the business model? A properly done model will potentially cost thousands – do the profits outweigh this?
- **Increase robustness.** How much will events like Christmas and Thanksgiving mess up the model – are the potential losses and problems from incorrect guesses because of events worth it? Do new menu items mess up the model?

Even after all this, I have a few ethical considerations.

- If we take more variables, at what point does it become unethical? When does knowing too much about the customer become a problem?
- Do users appreciate being predicted like this?

All in all, a solution is potentially worth it, but there are many considerations to take into account – both in terms of business and ethics.