Nathan Lo

Initial Exploratory Data Analysis Report

Some initial observations of the show that mostly year 2 and year 3 utilized the app, while barely any year 1 and 4 students used it. Most of the orders came from the same subset of universities (Butler, Indiana State, Ball State, and IUPU as the most common) while the most frequent students studied some degree related to the STEM field (the top 5 were chemistry, biology, astronomy, physics, and mathematics). Additionally, the most frequent order time was between 10:00am and 2:00pm. In regard to the orders themselves, all the products were ordered at fairly the same rate.

After conducting a comprehensive analysis of the dataset, it appears that the data lacks any evident patterns or structure. This conclusion is substantiated by observations from many aspects of the analysis.

One of the first signs of an absence of patterns in the data is the inability to identify any substantial correlations. Heatmaps and pair plots indicated that there is no correlation amongst the variables on an individual level (1 independent variable and 1 dependent variable). This suggests that the data may be noisy, or that the relationships are very complex. However, the latter has evidence against that idea. While trying simple regression models, I varied the number of inputs to predict orders; across all my tests, none of the models had high R^2 values.

Additionally, I was unable to determine any notable clusters in the dataset. This analysis was performed through looking at several different K-Means clusters. There was not a strong (or flat) elbow curve across the groups. Rather, there were only modest reductions in Within-Cluster Sum of Squares values (WCSS), indicating that there were not clusters in the data.

I also attempted to reduce the complexity of the dataset through a simple Principle Component Analysis. Looking at the explained variance ratios at different numbers of components/dimensions, the ratios were low, indicating that dimensionality reduction would not be beneficial for the data.

I also developed some simple regression models with varying inputs to predict the output, "order." After splitting the data into training and testing, the model did not perform well across all the tests, with R^2 values very low.

This lack in discernible patters in the data can have several causes: noisy data, small sample size, and complicated interactions to name a few. I believe it may be one, or a combination, of these particular factors. Roughly 5000 data points may not be enough information to reveal complex patterns, and so some of the noise in the data can make it especially difficult to find a pattern in the small sample data. Thus, the business may need to wait and collect more data to further analyze the data.

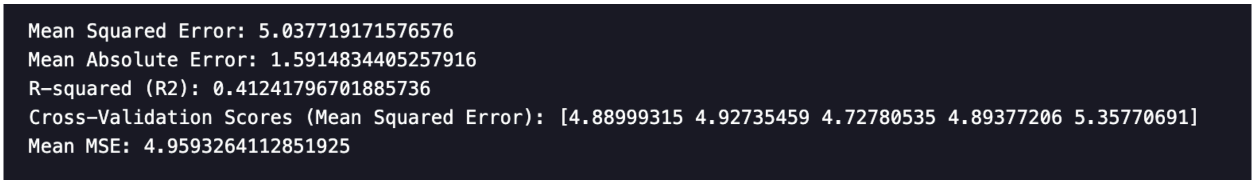
Predictive Model Outline

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

1. Outline your process for model selection, training and testing. Including data preparation.
2. Design a function that prepares your data by loading the provided dataset and processes it into an appropriate machine readable format if necessary.
3. Design a function to train your model and pickle it.
4. Train and test your model. Submit any training, testing and model selection visuals or metrics.
5. Upload your work to GitHub and link the repository, make sure it is public

The goal of my model was to use a model that can potentially find a pattern in a dataset that seemingly lacks correlation. Considering that my simple linear regression model was ineffective during my exploratory analysis phase, I wanted to utilize a different model. I ultimately used a random forest regressor to potentially discover non-linear patterns in the data.

I first manually converted the data to be all numerical. Afterward, I split the data and used a standard scaler on it. I designed the model so that the input variables are "University", "Major," "Year," and "Time," while the predicted variable is "Order." I then used sklearn functions to split the data into a training set and test set. I trained a random forest regressor model on the training portion of the input and output data and tested it on the input test data. I used mean squared error, mean absolute error, and R^2 scoring to calculate the performance.



```
Mean Squared Error: 5.037719171576576
Mean Absolute Error: 1.5914834405257916
R-squared (R2): 0.41241796701885736
Cross-Validation Scores (Mean Squared Error): [4.88999315 4.92735459 4.72780535 4.89377206 5.35770691]
Mean MSE: 4.9593264112851925
```

The regressor performed decently well, as the mean squared error (MSE) was about 5.04, the mean absolute error (MAE) was about 1.59, and the R^2 value was 0.41. The moderately low values of MSE and MAE are still indicative that the model did not guess correctly often. Considering that the categorical values of each order was converted to respective numbers, the MAE being greater than 1 is indicative of the model guessing the wrong food often. In addition, the low R^2 suggests that the model could not explain much of the variance of the data.

Final Conclusion

To determine the feasibility of continuing work on this model, I would consider three main ideas: quantity and quality of data, cost-benefit analysis, and time sensitivity. Considering that the initial data analysis revealed an absence of discernible patterns, it appears that the data is not currently suitable for predictive modeling. Thus, I would suggest expanding the number of measured variables and waiting for additional data to come in. With more variables and more information, predictive patterns will emerge more readily. I would also consider the cost-benefit of investing time in an analysis like this. Presuming that future data will have more apparent patterns, I believe it is worth spending resources to develop a strong predictive model. However, I would consider using stronger models, such as XGBoost, so that the model can build a more confident prediction. Lastly, I believe time sensitivity is important to consider. If a model needs to come out in the new future, I do not think it is worth investing in at all. However, with a few more months of quality data collection, a strong model could be built relatively fast.