**AI and Machine Learning**

Dani Tuck, Delanee Morgan, Kayson Hertzler Carson Hurd

**Project Overview**: In this project, we were presented with a database of a manufacturing company and were asked to use machine learning and learn how to use the database.

**Problem Statement**: The problem that we were asked to solve by the company was to predict when a package would be received by the customers.

**Data**: The database was provided by KOCH and accessed through the Snowflake program. In the database there are 1.1 million rows of data that include different vendors, dates, and product descriptions.

First, we noticed that the data needed to be cleaned. Many of the rows contained NULL values and had to be removed. [see NULL.txt]

To further clean the data, we noticed that all of the date columns had dates stored as integers using the format YYYYMMDD (I.e. 20210807), which can't be used to make calculations. We changed the dates by converting them into CHAR and then used the TO\_DATE function to convert it into a date datatype. [see DateConvert.txt]

Once we did this, we were able to start using the data in calculations. [see Aggregate.txt]

You can see in the Aggregate.txt file that the secret to our prediction calculations was based on subtracting the FIRST\_GR\_POSTING\_DATE from the DELIVERY\_DATE. This gets us the difference between the expected delivery date and the actual delivery date.

**Methods**: To be able to create the visualations and a UI for predictions regarding the arrival date, we used Python to do the heavy lifting. We first initialized a Snowpark session to be able to access the edited dataset within Snowflake. Then we converted the data from snowpark into a DataFrame that is able to be used by methods from the Pandas module. We used the groupby and median methods provided by Pandas to be able to narrow down the median difference between expected and actual delivery dates based on vendor.

We then used the Matplotlib module methods to plot out and display a graph of the Median differences by vendor.

For the UI we used the tkinter module to create a simple UI where you can enter in a Vendor ID and the estimated delivery provided by the Vendor. We then would use the calculate\_arrival method to filter out the vendor ID we are looking for and do that calculation to determine the data based arrival date that we estimate. Since we are relying on user input we have it set up to catch exceptions if the data is not entered in a way we need.

Using the data that we found we created a pop window within python that the company can use to enter a vendor ID and the expected date of delivery given by the vendor and compare that to the Median of delivery dates that the vendor has had in the past and output an expected delivery date based on that information.

**Results**: Using the median the company can see all vendors and their delivery dates and using that can makes decisions if they want to keep using that vendor based on time that they normally deliver their products.

**Visualizations**: Using Python we then created a graph to show the medians for each vendor. Each number on the graph is the median number of days that each vendor took to deliver the product. If the number is negative that means the vendor is usually early in comparison to its predicted delivery date and if the number is positive then the vendor had a tendency to deliver its products after its expected delivery date.

**Code**: We used the code in snowflakeCode.py to generate the graph for visualization along with creating a UI that allows the user to input in the vendor ID and the date that the vendor estimated the package would arrive and then give a more accurate estimate based on the data provided. The connection.json file allows for access into the Snowflake database.

The code was run using the snowpark-ml-hol terminal enviroment provided by Anaconda Navigator

<https://github.com/Tuck-Danielle/Querious_George/>

**Challenges**: One of the challenges that we faced was using our time wisely. While working on the QuickStart we took too much time trying to get everyone to do the QuickStart rather then one person doing it on one computer and we all learn together. Another challenge that we had was that we did not use one anothers strengths in the beginning. Had we used each others strengths sooner we could have completed multiple parts of the project more efficiently. The final problem that we ran into was recording information. We were not very good at writing down our steps in the early parts of our process so we had to do it later which makes it harder.

**Future Work**: One thing that would enhance the project would be a clarification in the naming of the columns. There was a lot of confusion as to which columns meant what and took some digging to find which ones we needed to use in order to predict the date of delivery. We would recommend that the company finds a more standardized method of collecting data from the vendors. Within the data there are a lot of null values and a lot of formatting issues. With these new corrections the company could by more easily evaluate the data

**Conclusion**: We found that between all of the vendors that the company purchaes from there are more vendors that have a tendency to deliver early however of the vendors that deliver late, they tend to deliver well past their expected delivery date. The company would need to decide if they want to continue working with those vendors or have a conversation with those vendors to find the reason that those vendors tend to deliver late. One solution that we would recommend is that the company provides incentives to the vendors if they deliver early or on time and they lose those incentives if they deliver late. We also recommend the company provide compensation to the customers who received their packages late. With these changes with believe that the company could not only be able to promise delivery dates as a whole but the customer would be more satisfied because based on data they know when their package will arrive so that it won't be late.