D214 – Data Analytics Graduate Capstone

**Data Analytics Report**

Western Governors University

December 27, 2024

Kevin Rupe

Student ID: XXXXXXX

Email: krupe6@wgu.edu

Phone: (864) 704-2340

**RESEARCH QUESTION**

The research question posed on this project is whether certain variables related to an alarm on a customer network device have a high correlation to the alarm itself being cleared. Internet Service Providers (ISP) have an objective to deliver fast and reliable service to their customers. In our ever-evolving world of technology with the exponentially expanding Internet of Things (IoT), as a society in the United States we have more devices online than ever before. In today’s times, we all rely on our devices’ connectivity to the internet in our day-to-day lives.

Having quality, fast, and reliable internet service is no exception to business customers who depend and rely on them in order for their businesses to operate efficiently (Torkildson, n.d.). When businesses are offline, they often can’t function. This hampers their ability to run their business and ultimately, this costs them money. This can have a ripple effect on their customers as well.

For ISPs that are providing internet service to a business, it is imperative that when their customers are experiencing service affecting events that they be repaired and/or services are restored quickly, efficiently, and correctly. Getting their customers back online is the most important thing, and time is of the essence. Afterall, if the ISP struggles to provide excellent service to their customers, then in the end, their customers will find another ISP who can. Therefore, this has a negative impact on the ISPs revenue.

Businesses usually have multiple advanced network configurations deployed on their premises. Whenever one of these devices goes offline, an alarm case is created. By applying a Multiple Linear Regression model, we can gain insights into what factors might have a high correlation to the alarm clearing (Massaron, 2016). These insights would provide extremely useful intel to the ISP at helping them streamline processes, enhance their decision making, allocate resources, and even provide predictive insights. All of which would drive the company towards providing a better customer experience.

The null hypothesis of this research question is that there will not be any significant correlation in the related variables to the alarm being cleared. The alternative hypothesis is that there will be at least one variable that has significant correlation to the alarm clearing. If any variables are found to have significant correlation to the alarm clearing, then it would explain why the alarm cleared.

The alarm clearing means that the service affecting issue has been resolved. To understand on an analytical level why the alarm cleared would be very useful and valuable information to the ISP in determining how best to treat future service affecting issues, and possibly even how to proactively prevent them where they can. This ultimately would provide a better customer experience which would have a direct positive impact on ISP revenue.

**DATA COLLECTION**

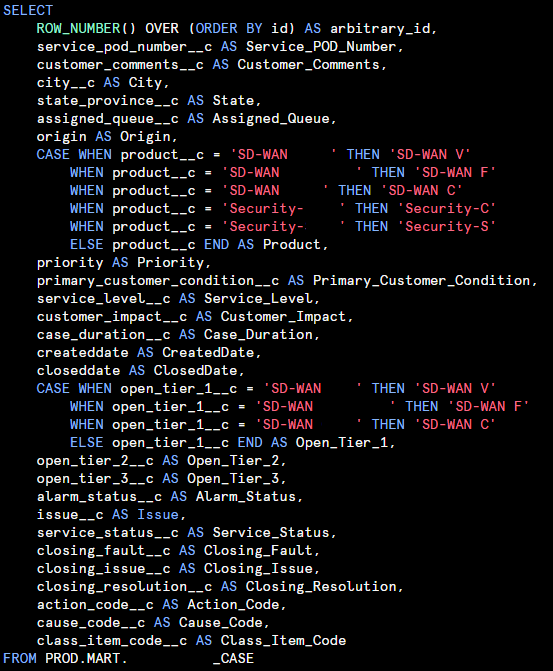
When an alarm occurs on a customer’s device an alarm case is generated. Each case has many variables that relate to a specific device that has an alarm event. Therefore, to determine if there are any factors that have a significant correlation to the alarm being cleared, we need to gather all of the variables on the alarm case. The dataset from the ISP was privately obtained, and contains 30 variables and 124,152 observations. The dataset has variables of Case\_Duration, CreatedDate, ClosedDate, City, and State. The predictor variables are broken down and explained below:

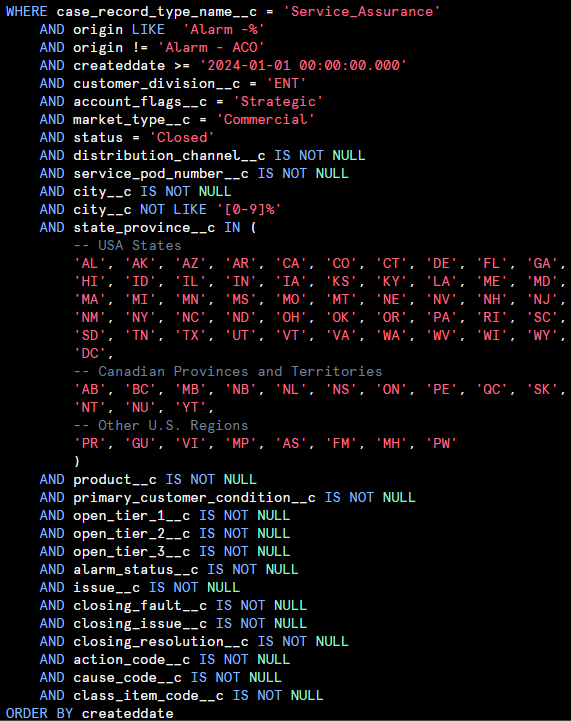
| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Arbitrary\_ID | Nominal Categorical | Arbitrarily created ID to mask any sensitive information |
| Service\_POD\_Number | Nominal Categorical | Unique group number indicating the service level team that will handle the case |
| Customer\_Comments | Discrete Quantitative | The number of times the customer interacted with the case |
| Assigned\_Queue | Nominal Categorical | The queue in which the case is assigned |
| Origin | Nominal Categorical | The originating system of the case |
| Product | Nominal Categorical | The product of the service affecting issue |
| Priority | Ordinal Categorical | The priority of the case |
| Primary\_Customer\_Condition | Nominal Categorical | The condition of the customer |
| Service\_Level | Ordinal Categorical | The service level of the customer |
| Customer\_Impact | Ordinal Categorical | The impact level of the service affecting issue |
| Open\_Tier\_1 | Nominal Categorical | First level tier related to the device and the alarm |
| Open\_Tier\_2 | Nominal Categorical | Second level tier related to the device and the alarm |
| Open\_Tier\_3 | Nominal Categorical | Third level tier related to the device and the alarm |
| Alarm\_Status | Nominal Categorical | Clear (indicates the alarm has been resolved) Active (indicates the issue is still ongoing) |
| Issue | Nominal Categorical | Service is Down (indicates the customer has no service) Service is Impaired (indicates the customer has partial service) |
| Service\_Status | Nominal Categorical | Out of Service (indicates that the customer is out of service) In Service (indicates that the customer is not out of service) |
| Closing\_Fault | Nominal Categorical | The fault of the alarm |
| Closing\_Issue | Nominal Categorical | The issue of the alarm |
| Closing\_Resolution | Nominal Categorical | The resolution of the alarm |
| Action\_Code | Nominal Categorical | The action of the alarm |
| Cause\_Code | Nominal Categorical | The cause of the alarm |
| Class\_Item\_Code | Nominal Categorical | The class item of the alarm |

One big advantage to using this dataset is the fact that I was able to pull the data directly from the source itself, which was located in a Snowflake database directly at the ISP. I used SQL to explore the alarm case table and all of its related column variables. Some of the data was missing, and since I had ample observations of data to use, I simply chose to remove any null values from the query before extracting the data into a dataset. I also chose to only find alarm cases that were closed in the calendar year 2024. Upon inspection, I noticed that several City and State names seemed to be bad data, so I decided to exclude any City name that did not begin with an alphabet letter, and I wrote an inclusive statement to only find States in the USA, Canada and US regions (e.g., Puerto Rico, Guam, Virgin Islands, etc.).

One disadvantage with this dataset is the fact that all the variables only point to what is known by the ISP. There are many factors that can cause an alarm that are not easily identifiable. For example, the internet is connected by all sorts of different ISPs, and the issue may be something related to another carrier’s network, and therefore might not be fully known. Other issues with alarms could be related to Utility Company power outages, or even natural disasters.

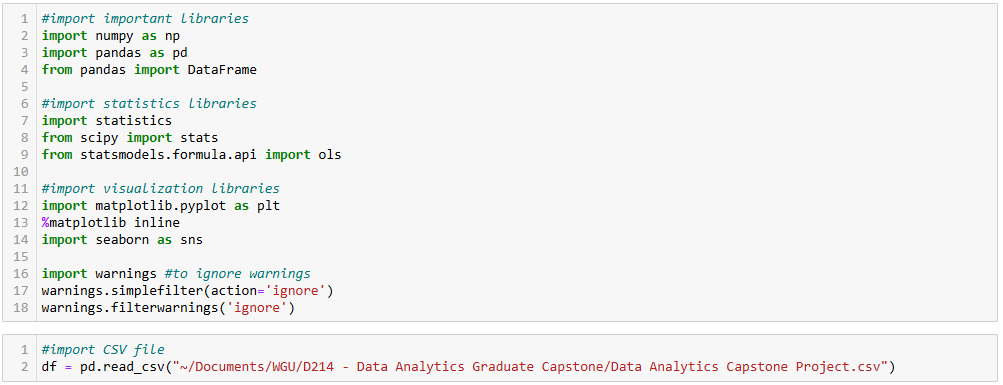
The SQL query used to gather the data is below:



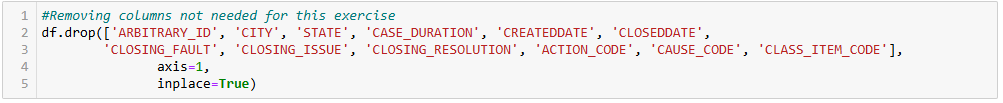


**DATA EXTRACTION AND PREPARATION**

Once the data has been gathered (above), the next step is to extract the data into a suitable format for reading into our Jupyter Notebook for analysis. The data is therefore extracted into a .csv file and loaded into Jupyter Notebook. Before we can begin, we must also import any relevant packages and libraries that will be necessary for this analysis.



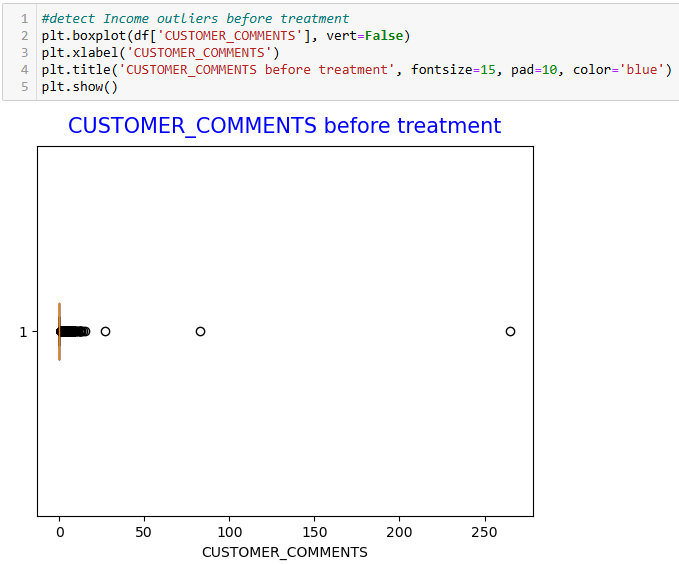
For this analysis, we will need to drop any columns that are not predictor variables. I also chose to drop each of the “Closing” variables, and each of the “Code” variables to reduce the dimensionality of the dataset. This will help with multicollinearity and will help prevent overfitting of the data (Geeks for Geeks, 2023).



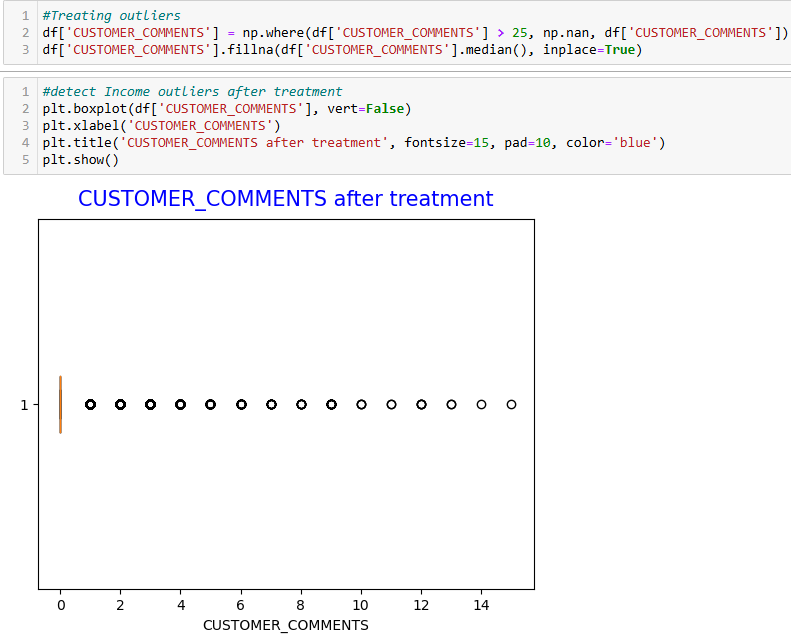
Next, we need to ensure that there are no null rows in this dataset. I did write the SQL query to reduce the possibility of null rows, but it’s still good to double-check anyway. The below output shows that we have no null data.



Looking at the dataset earlier, it seemed like the CUSTOMER\_COMMENTS variable was going to have outliers, so to check for this I will visualize this using a boxplot chart from Matplotlib’s pyplot function.

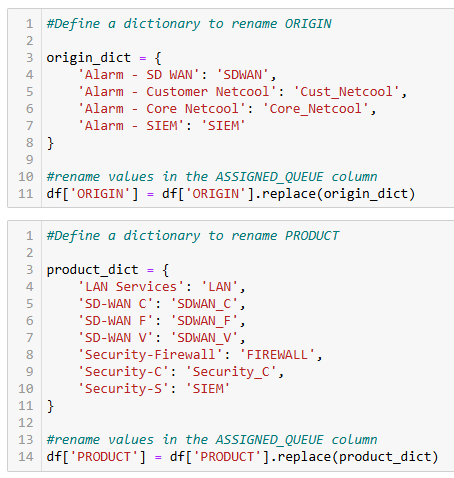
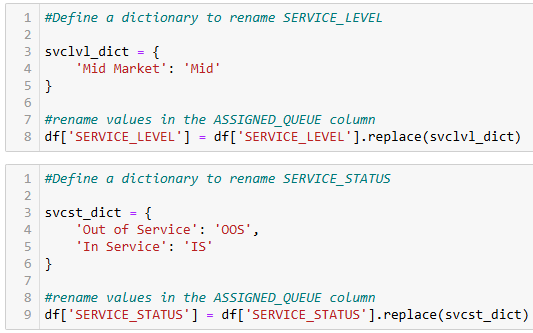


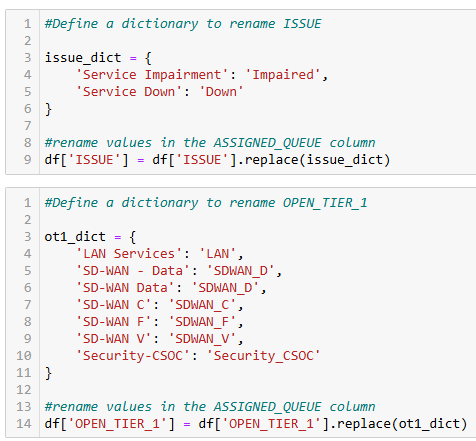
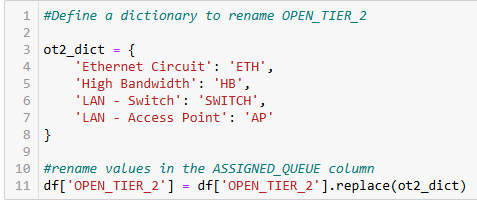
Looking at this boxplot graph, I would interpret this as having 3 outliers. Therefore, I will remove these 3. The remainder of the values look bunched very closely to the upper whisker, so even though they are outside the whisker I am going to leave them.



MLR requires all of the data to be numeric, therefore, I am going to need to convert most of these variables using One-hot encoding, Ordinal encoding, or use dummies (Pandas, 2024). Upon inspection, these variables will be Origin, Product, Priority, Primary\_Customer\_Condition, Service\_Level, Customer\_Impact, Open\_Tier\_1, Open\_Tier\_2, Open\_Tier\_3, Issue, Service\_Status, Assigned\_Queue, and Alarm\_Status.

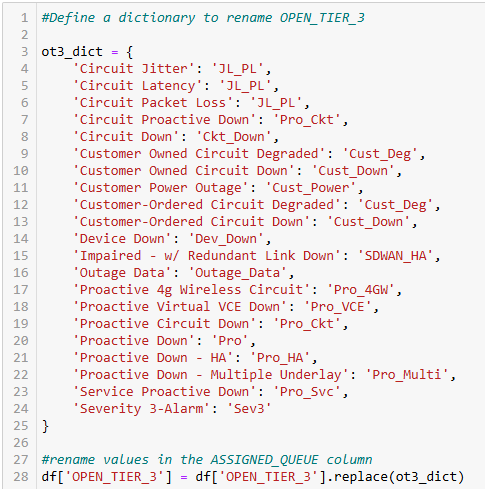
Statsmodels package in Python will help determine how closely the independent variables correlate to the dependent (Seabold, 2010). The slight drawback to the OLS function is that it requires variables to have no spaces, therefore, before converting these categorical variables into numerical ones, I will need to rename some of the variables to remove any unnecessary spaces in the names. The advantage of OLS far outweighs this slight disadvantage. In a very simple few lines of Python code, OLS will display statistical data that will explain a great deal of data including R-squared, Adjusted R-squared, F-statistic, AIC, BIC, p-values, and many other relevant values.

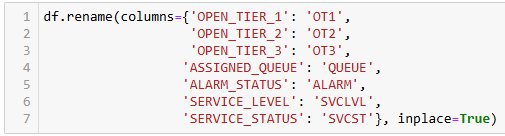
 

Certain variables, such as Assigned\_Queue and Open\_Tier\_3, have many unique options. This will certainly add dimensionality and the possibility of multicollinearity. Therefore, I chose to combine certain factors. I have spent time exploring this data, and my role at my employer gives me a closeup view of this data on a daily basis. I feel my experience affords me the knowledge of performing this task, but I would not do this on just any dataset.

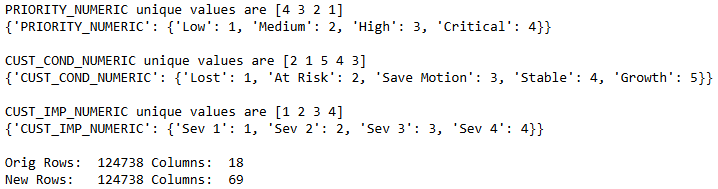




To simplify the names of these columns more succinctly, which will make them better for the next step of re-expression, I am choosing to shorten the names of several of these categories.

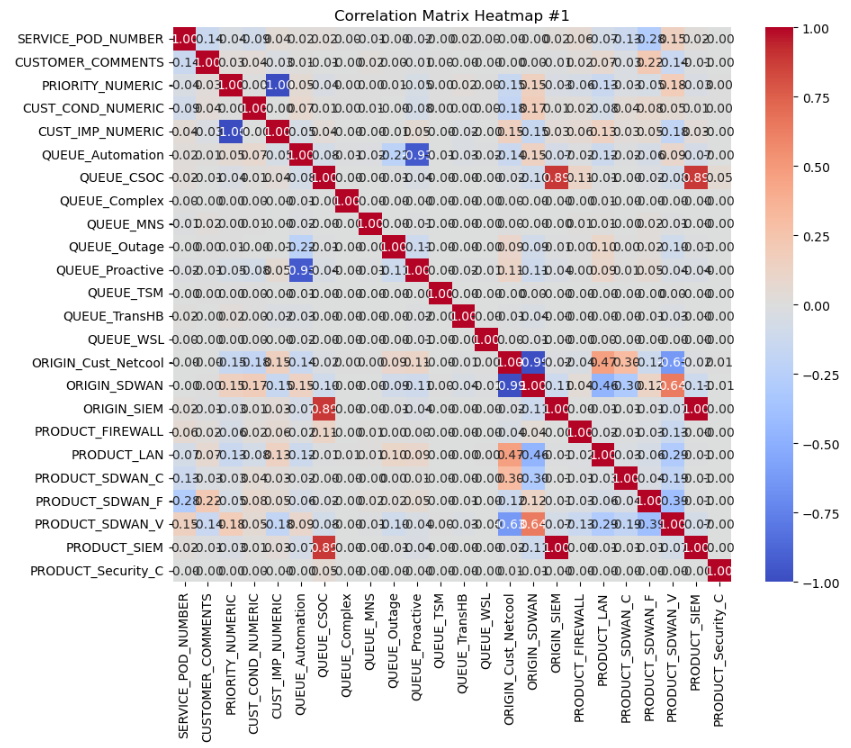


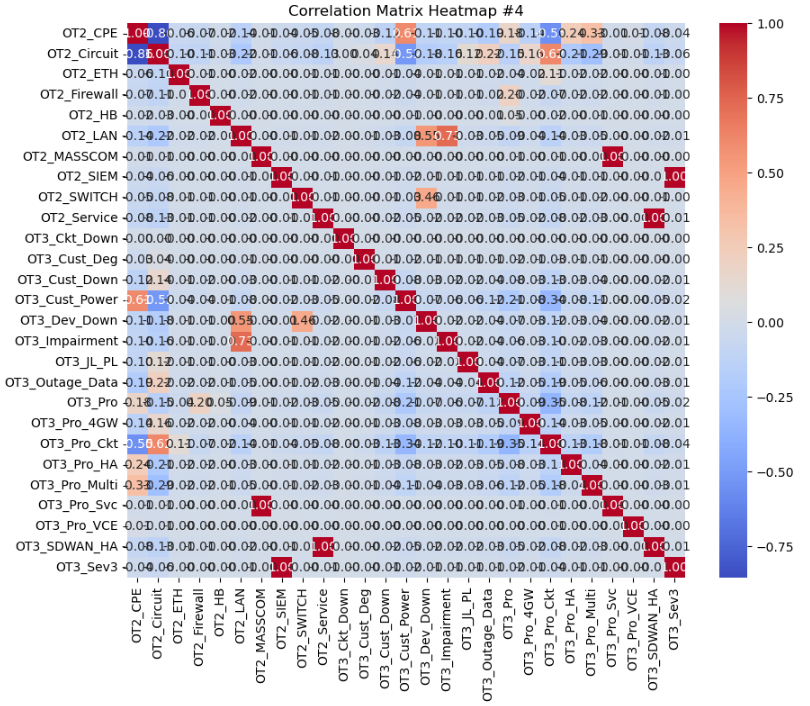
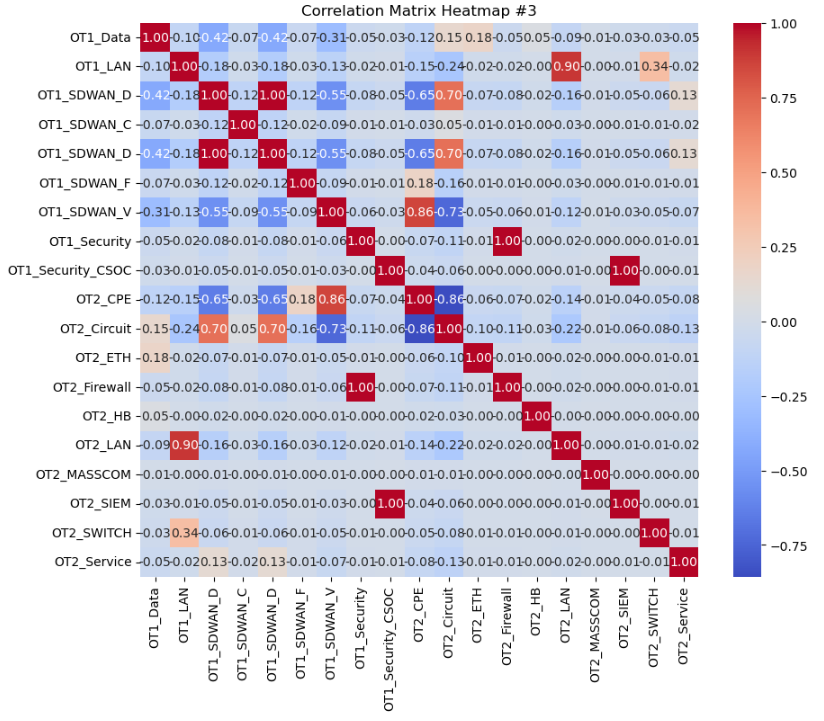
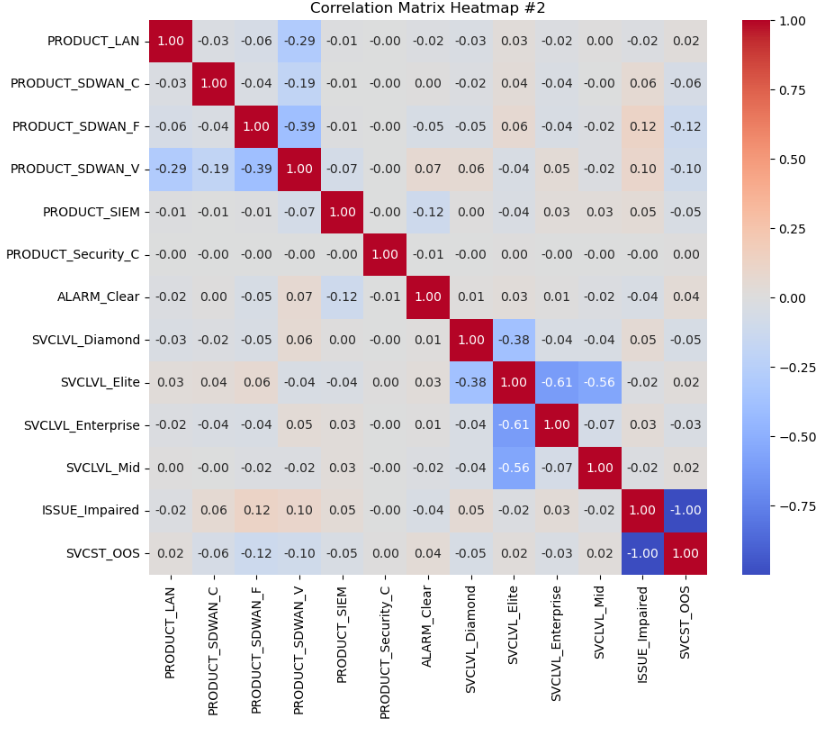
Ordinal Encoding is used for Priority, Primary\_Customer\_Condition, and Customer\_Impact. The remainder variables I will use dummy variables using a prefix list. Given all my efforts thus far in reducing dimensionality, this process still expands our dataset from 18 original columns to 69!



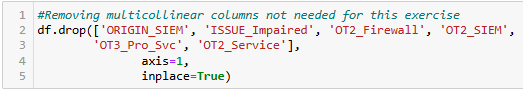
The next thing to do in our data preparation is to check for multicollinearity. Before I began this analysis, I expected some multicollinearity, but nothing prepared me for just how many variables I would find. This phase of the process was very tedious and repetitive which given time and resources, typically would be viewed as a disadvantage. However, nonetheless was it also very important to ensure our MLR model was appropriate and not overfit.

To first check for multicollinearity, I decided to view a correlation matrix using Seaborn’s heatmap function (Waskom, 2021). There were simply too many rows to be able to visualize it, so I had to split up the data as meaningfully as I could 4 different times. Each chart provided certain variables that were an exact 1-to-1 match for multicollinearity.



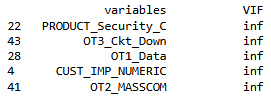


I then removed all of the values in the heatmap that showed 1.0. This shows that there is a perfect multicollinearity in these variables, so therefore, they are not needed.

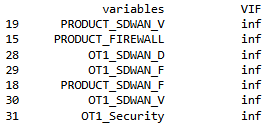


Furthering on in my search for multicollinearity, I next decided to use Statsmodel’s variance\_inflaction\_factor (VIF) function. Any values over 10 will need to be removed, however, I also need to only remove a few variables at a time as removing them may lessen the multicollinearity in the remaining variables. Running this VIF function the first few times showed “inf” output which means that the variable’s multicollinearity is *infinitely large*. Thus, these “inf” variables will need to be removed first.

On the first run, these variables were removed:

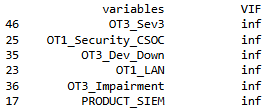


On the second run, these next variables were removed:

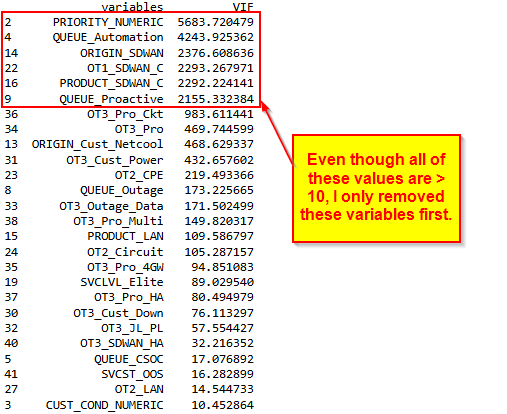


The limitation (or disadvantage) to using this VIF function is that when using too many variables, often it will only find the first few “inf” values. It’s imperative to iteratively check for new variables, if any are found, remove them, and then check again. Repeat this process until there is no longer any multicollinearity remaining in the dataset.

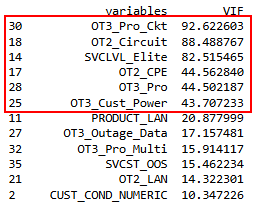
On the third run, these next variables were removed:



On the fourth run, I finally had no more “inf” values, but I now had plenty of values larger than 10. Rather than removing all values larger than 10 instantly, I chose to remove them in smaller sets, and re-checking each time for multicollinearity.



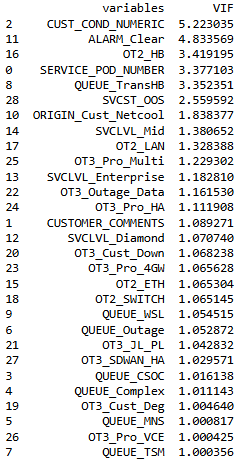
On the fifth run, I again chose to only remove the largest values, not necessarily all values larger than 10. For example, before the fourth run, OT3\_Pro\_Ckt had a VIF of 983, but afterwards the VIF dropped to just 92. This VIF is now less than 1/10th what it was before. This is why it’s important to slowly and methodically remove multicollinear variables, choosing the largest ones first.



On the sixth and final run, I only had 2 remaining variables larger than 10. I chose to just remove PRODUCT\_LAN and leave OT2\_LAN in the dataset.

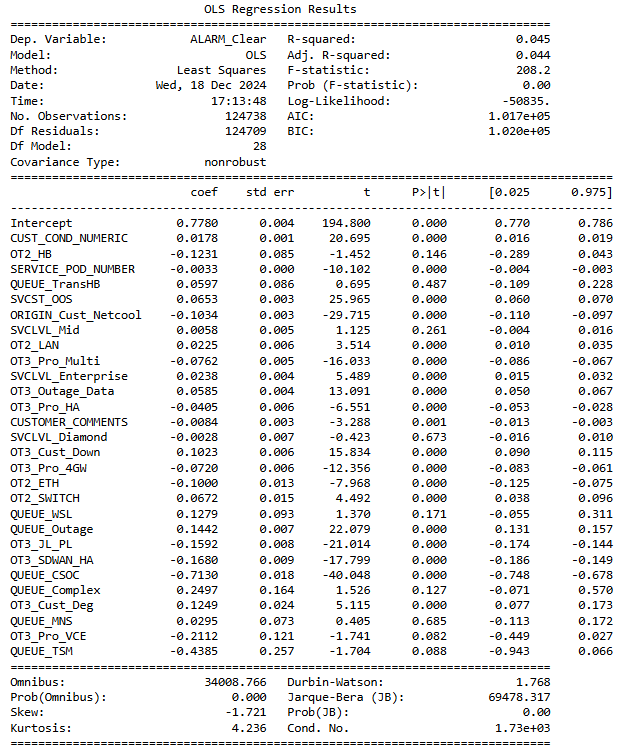


Checking one last time for multicollinearity I now have 29 remaining non-multicollinear variables out of the original 69.

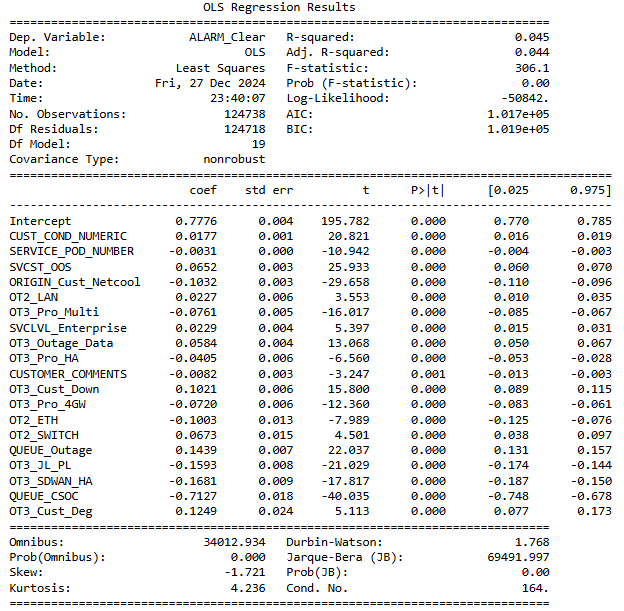


**ANALYSIS**

Now that the data is prepared, it is time to run Statsmodels’ MLR OLS function. Using ALARM\_Clear as the dependent variable, I add all 29 independent variables to the function.



I can see that there are several p-values larger than 0.05 which shows that they are not statistically significant and could hinder our model. Therefore, I will remove them from the function one by one, but will not necessarily remove them from the dataset. I will start with the largest values first until all p-values are less than or equal to 0.05.



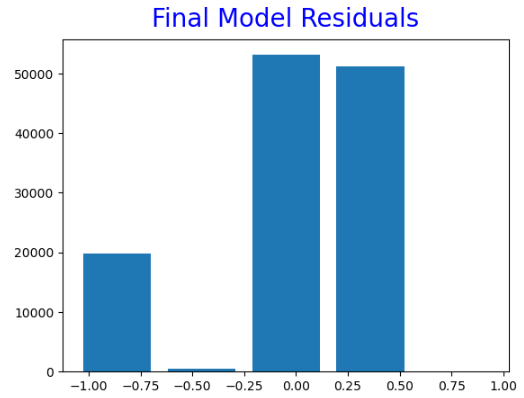
This is the final model that will be used to summarize and interpret the data. All insignificant p-values have been removed which lowered the condition number from 1,730 to 164. All numerical issues have been removed from the model.

**DATA SUMMARY AND IMPLICATIONS**

A relatively low R-squared value shows that just 4.5% of the variability in ALARM\_Clear is explained in the model. The same came be said for the low 4.4% Adjusted R-Squared value. This suggests plainly that the independent variables do not account for much of the variability in ALARM\_Clear. In other words, you would not be wise to attempt to predict if the alarm were to clear based on the independent variables in this model.

Several notable variables have a strong impact on the dependent variable. QUEUE\_CSOC has the strongest negative effect which indicates that increases in QUEUE\_CSOC drastically decreases ALARM\_Clear. Another queue, QUEUE\_Outage has the opposite effect on ALARM\_Clear, though not nearly as strong with a coefficient of 0.1439. This queue has the highest positive coefficient.

Both the Omnibus and Jarque-Bera tests have significantly large values which indicate that there is a non-normality in the residuals. Viewing these residuals is evident that there is a negative left skew, which is also shown in the -1.721 Skew value. This is a limitation in the dataset.



Given a high F-statistic of 306.1, and a significantly low p-value for this F-statistic of 0.00, we are assured that this model bears an overall significance with respect to the independent variables explaining the variance in the dependent variable. Therefore, we must reject the null hypothesis and rather accept the alternative hypothesis that there is significant correlation between at least one of the independent variables and the alarm being cleared.

The variables that are significantly positively correlated are Primary Customer Condition, Service Status, Service Level of Enterprise, Assigned Queue of Outage, Open Tier 2 codes of LAN and Switch, and Open Tier 3 codes of Outage Data, Customer Down, and Customer Degraded.

The variables that are significantly negatively correlated are Service POD Number, Origin of Customer Netcool, Customer Comments, Assigned Queue of CSOC, Open Tier 2 code of Ethernet, and Open Tier 3 codes of Proactive Multiple Underlay, Proactive HA, Proactive 4G Wireless, Jitter, Latency, Packet Loss, and SD-WAN HA.

The recommended course of action is for the ISP to simplify the level of detail the alarm cases are given. This dataset shows that too much information is not a “good thing.” Given the complexity of the data, it makes it extremely difficult to analyze properly without first removing multicollinear variables as well as statistically insignificant ones. Teams within the ISP are working hard to determine how to cut down on the amount of time an alarm stays active, and how to proactively clear alarms. Using this dataset will make this task impossible given that this model is only able to predict 4.5% of the variability in the alarm clearing. Predictive Analytics can greatly help an ISP become more proactive in its approach to network optimization than strictly being reactionary (Shillingsburg, 2024). Steps should be taken to reduce the dimensionality of this dataset which would aid the analysts who are looking to predict certain behaviors from the data.

The first direction of approach on a future study would be to remove the Open Tier Codes and replace them with the previously removed “Closing” codes to evaluate if this model is able to better predict the alarm clearing. For this ISP, the “open” codes are determined when the case is opened, and the “closing” codes are assigned once the case is closed. Since more is known about the issue at the end of the case rather than the beginning, this could improve the model’s explanatory power.

Another direction to take would be to change the dependent variable to SERVICE\_STATUS and see how well the independent variables explain the variance there instead. The ALARM\_STATUS variable is updated during the life of the case once the alarm clears, but the SERVICE\_STATUS is a historical object only. Therefore, we could learn which variables are statistically significant to SERVICE\_STATUS which would be useful information for the ISP that might explain more about why some products, queues, origins, etc., tend to correlate to “Out of Service” events.

**SOURCES FOR THIRD-PARTY CODE**

Massaron, L. & Boschetti, A. (2016). Regression Analysis with Python: Learn the Art of Regression Analysis with Python. Packt Publishing.

Pandas (2024, September 20). Pandas documentation. Retrieved December 10, 2024, from <https://pandas.pydata.org/docs/index.html>.

Seabold, S. & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Scientific Conference. Retrieved December 10, 2024, from <https://www.statsmodels.org/stable/index.html>.

Waskom, M. L. (2021). Seaborn: statistical data visualization. Retrieved December 10, 2024, from <https://seaborn.pydata.org/index.html>.

**SOURCES**

Geeks for Geeks (May 06, 2023). Introduction to Dimensionality Reduction. Retrieved December 27, 2024, from <https://www.geeksforgeeks.org/dimensionality-reduction/>.

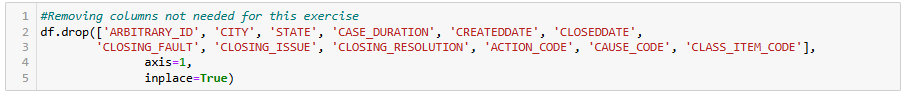
Shillingsburg, S. (2024, October 30). Predictive Analytics for Network Optimization. Retrieved December 19, 2024, from <https://sonar.software/blog/predictive-analytics-for-network-optimization>.

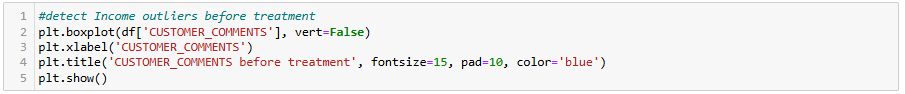
Torkildson, A. (2017-2024). Why The Internet is so Important for Modern Business. Retrieved December 19, 2024, from <https://www.thinkers360.com/tl/blog/members/why-the-internet-is-so-important-for-modern-business>.

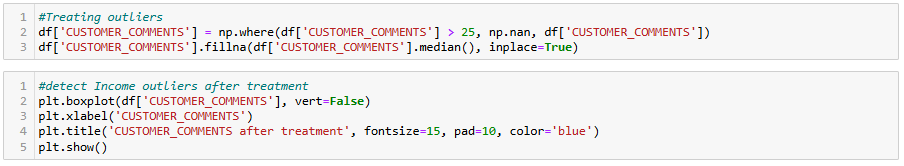
**APPENDIX**

**The Python code in Jupyter Notebook**

****

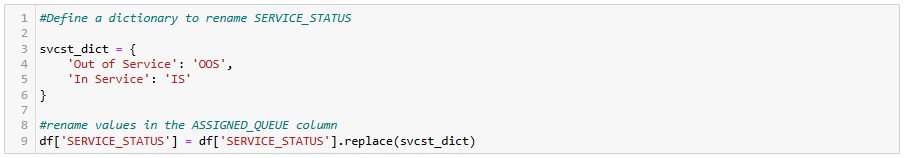
****

****

****

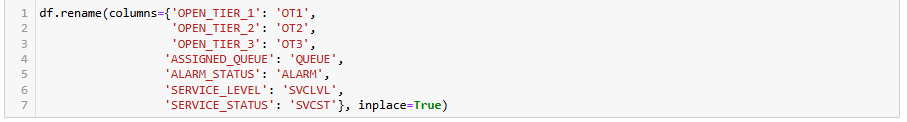
****

****

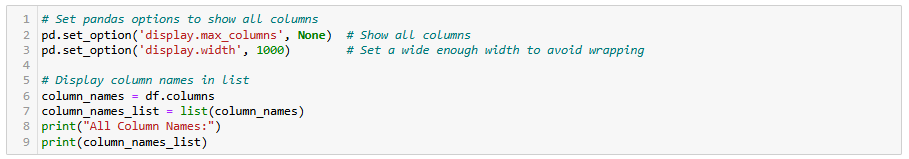
****

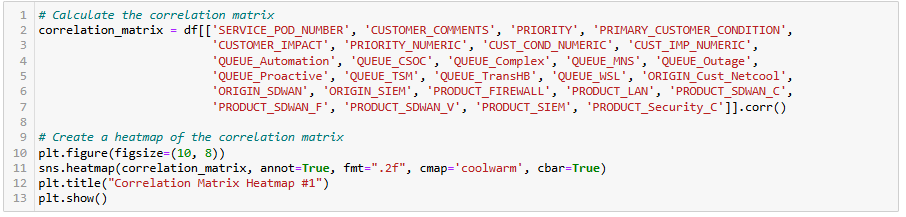
****

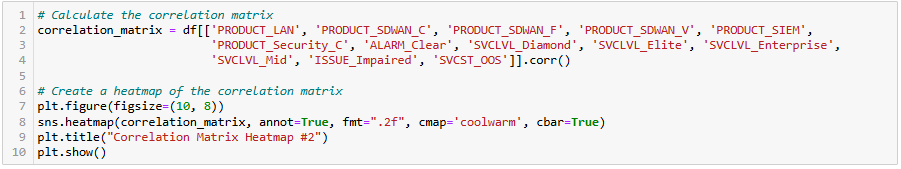
****

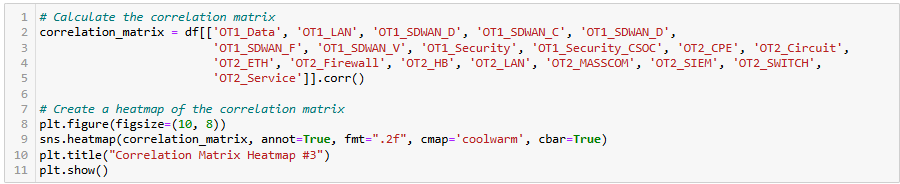
****

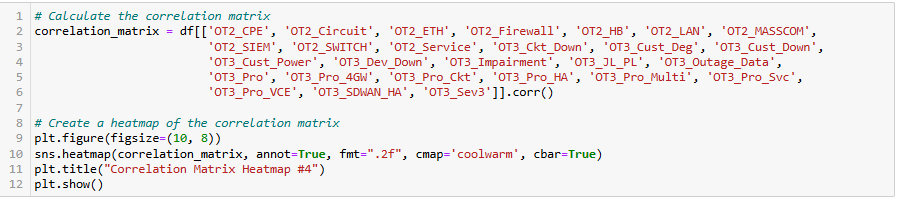
****

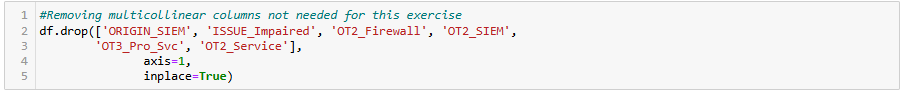
****

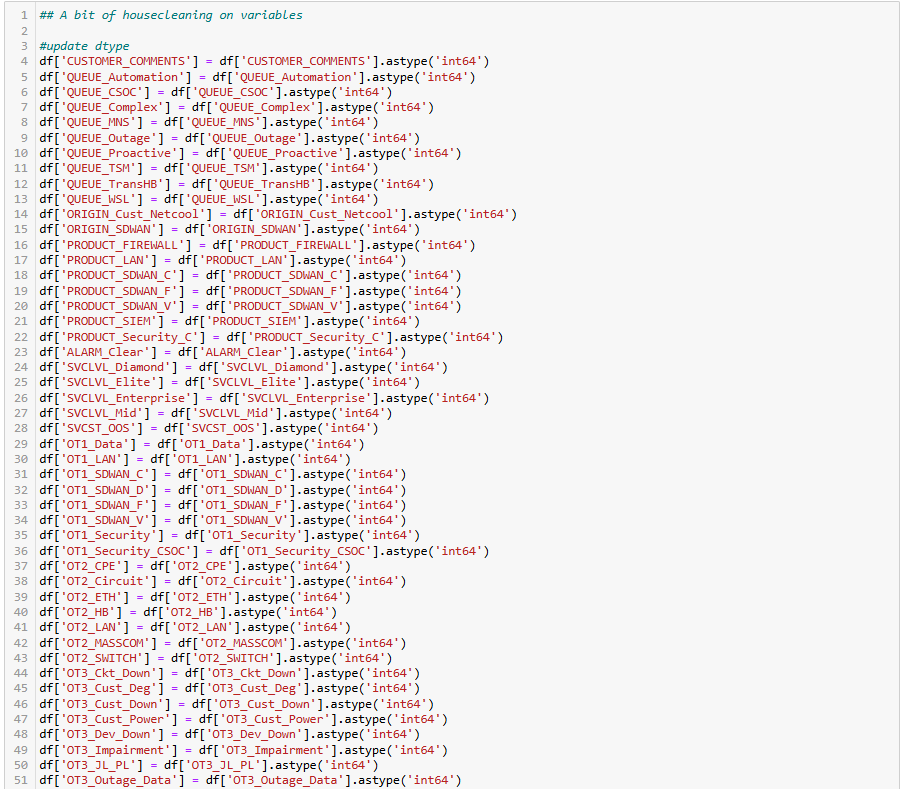
****

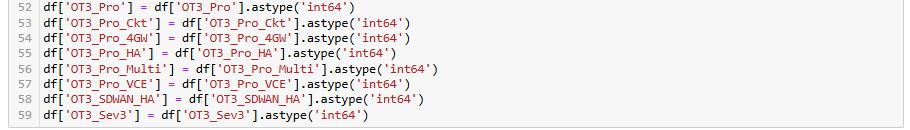
****

****

****

****

****

****

****

****

****

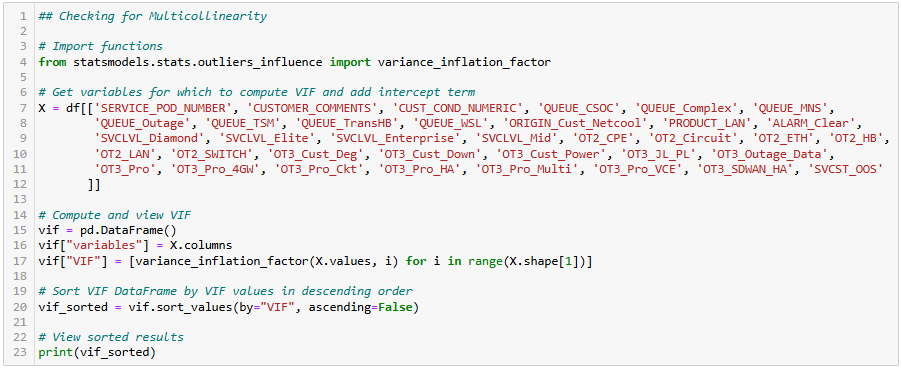
****

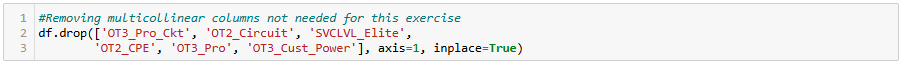
****

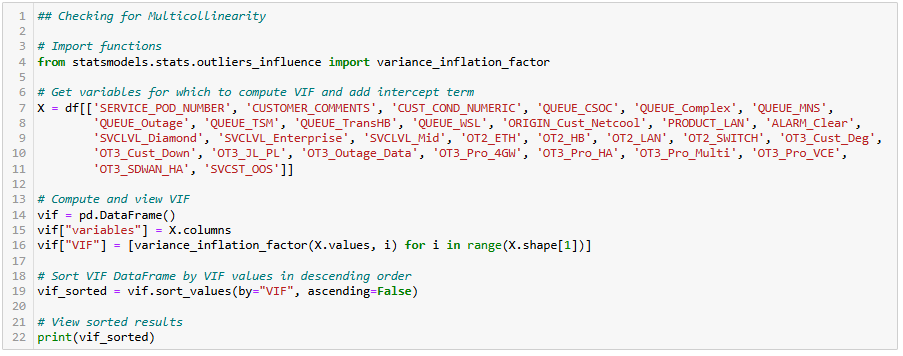
****

****

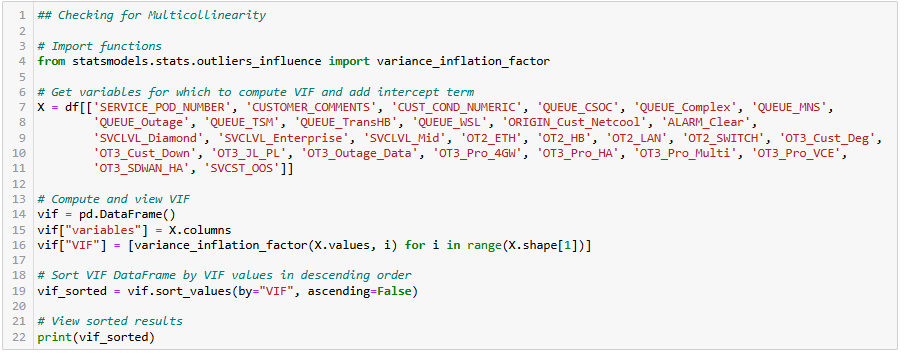
****

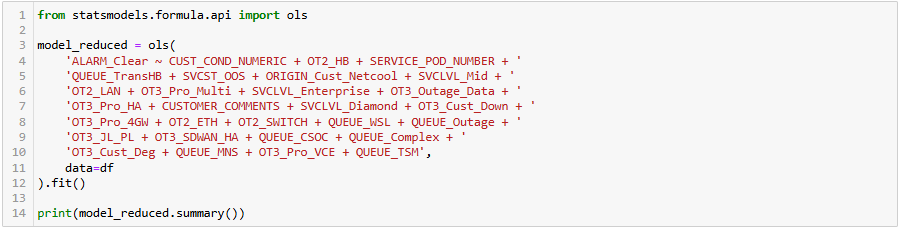
****

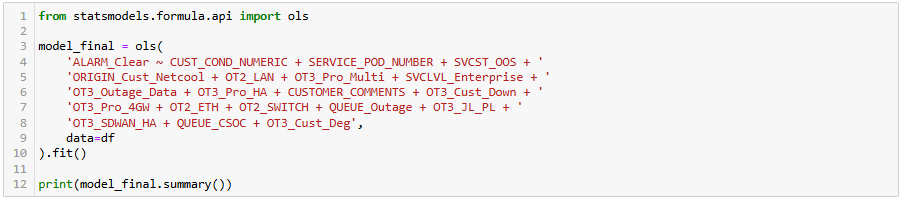
****

****

****

****

****

****

****