1) Explain the importance of Precision with regards to the model you have just created using malheur. Provide the formula/equation and also why the precision calculation is important to evaluating the model's performance.

Precision reflects how well individual clusters agree with malware classes (Rieck et al., 2011). In regards to my model, I was able to achieve scores of: 61.1%, 62.1%, and 100% for the training, testing, and sample data sets respectfully. The formula, shown below, calculates the percentage of correct predictions divided by the sum of correct and incorrect predictions in regard to accurate clusters (Brownlee, 2020). The lower this percentage, the higher the recall percentage. This shows me how correct the clusters within each data set were, and how many extra clusters there may have been for the training and testing sets.

$$P = \frac{1}{n} \sum_{c \in C} \#_c$$

2) Explain the importance of Recall with regards to the model you have just created using malheur. Provide the formula/equation and also why the Recall calculation is important to evaluating the model's performance.

Recall measures the extent classes are scattered across clusters (Rieck et al., 2011). In regards to my model, I was able to achieve scores of: 81.1%, 83.3%, and 60% for the training, testing, and sample data sets respectfully. This formula, shown below, calculates the percentage of correct predictions divided by the sum of correct and incorrect predictions in regards to scattered classes (Brownlee, 2020). The lower this percentage, the higher the precision percentage. This shows me that the training and testing sets have a lot of extra classes scattered across the model that could've been added to an existing cluster.

$$R = \frac{1}{n} \sum_{y \in Y} \#_y$$

3) Explain the importance of f-score with regards to the model you have just created using malheur. Provide the formula/equation and also why the f-score calculation is important to evaluating the model's performance.

F-score is an aggregated performance score that combines both precision and recall percentages (Rieck et al., 2011). In regards to my model, I was able to achieve scores of: 69.7%, 71.2%, and 75% for the training, testing, and sample data sets respectfully. This gives me an overall score of how well the model was able to discover classes within a dataset. The formula, shown below, multiplies the product of precision and recall by two, and divides this result by the sum of precision and recall, known as a harmonic mean (Brownlee, 2020). This shows me the overall performance of my configured model, and how the percentages of precision and recall can really impact its effectiveness.

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

4) A key aspect to malheur and performing unsupervised machine learning is the use of 'prototypes'. Please explain in detail what 'prototypes' are with regards to malheur?

Malheur identifies prototypes that are representative for the full data set. Prototypes are used as a quick overview of recorded behavior to guide inspection (Rieck, 2012). This data is also what makes up cluster groups that are used to measure different types of data throughout a generated report.

5) Why are prototypes important to performing clustering and classification in malheur? Explain how they are related to describing the clustering group.

Malheur identifies groups of reports containing similar, or unknown, behavior groups; these clusters are made up of prototypes that are extractracted from the dataset. Clustering allows for discovering classes of malware and provides the backbone for specific detection and defense strategies (Rieck, 2012). Classification enables identifying unknown variants of malware that are used to filter behavior prior to inspection (Rieck, 2012).

6) Please identify and explain the importance of the key mathematical distance calculation Malheur uses to calculate distances between samples? In particular, explain how this calculation is used in respect to q-grams.

Malheur primarily uses the key mathematical distance calculation to evaluate the performance of the proposed clustering using prototypes, and potentially identify areas to improve. We obtain different prototype sets by varying the distance parameter. Using this, we evaluate different settings for embedding reports where we choose length of q of instruction q-grams from the set {1, 2, 3, 4} (Rieck et al., 2011).

7) Why is it important to have API call names (our features) in the order calls were actually made in the malware execution? Also, what happens if the order is lost?

Displaying the API call names in order is important because aggregated textual reports limit machine learning methods, since the actual sequences of observed behavioral patterns are not directly accessible. If the order is lost, this could cause the complexity of textual representations to increase the size of reports, which badly impacts run-time of analysis (Rieck et al., 2011).

8) Explain the difference between intra-cluster cohesion and inter-cluster separation. What is the commonly used measure for intra-cluster cohesion?

Intra-cluster cohesion measures how near the data points in a cluster are to the cluster centroid, while inter-cluster separation measures different cluster centroids that should be far away from one another (Ullman et al., 2014). To measure intra-cluster cohesion, usually the sum of squared error method is used. This is the sum of differences between the predicted value and the mean of the dependent variable (Pathak, 2020).

9) Provide a detailed summary explaining how you have achieved your proposed values for malheur's parameters in task 1 (i.e. strategy, rationale, steps), and how those values relate to your results (i.e. precision, recall, f-scores). No credit will be given for this question in case you choose not to complete Phase 3 Goal 1.

ngram_len: This parameter analyzes the length of event sequences to be mapped to the vector space in n-grams. The default value of this was set to 2, but in my configuration file I updated this value to 1 instead. This value is only set to 1 if the events in the reports are not in sequential order. Even though the API calls are encoded in sequential order, I was able to get a better performance because of the value I set in the **vect_embed** parameter.

vect_embed: This parameter controls how the features are embedded in the vector space. While the default value for this parameter is **bin**, which associates each dimension with a binary value, I changed the value to **cnt** instead. This parameter value associates each dimension with a count value for occurances, which causes the API call list to appear non-sequentially. Because of this, I was able to cut the distance for the sample datasets under 1.0.

max_dist (prototypes): This parameter is responsible for the maximum distance prototypes can be away from each other. I chose a distance of 0.10 because this value had a big impact on the testing f-score. The higher the distance allowed between prototypes, the higher the recall percentage, thus improving my f-score on the testing dataset. This coupled with the cluster's **min_dist** parameter allowed me to get the goal coverage state.

max_num: This parameter defines the maximum number of prototypes allowed in an analysis. While this helps run time, this parameter provided no value in improving the testing f-score, and caused some samples to measure over a 1.0 distance because of the allowable prototype limit. This is why I kept the value for this parameter at 0.

link_mode: This parameter specifies the clustering mode used during analysis. Other than the default **complete** clustering setting, there are also **average** and **single** modes as well. Since this had no positive or negative impact on the testing/classify results, I kept this at the default value.

min_dist: This parameter sets the minimum distance between clusters that are made during analysis. I was able to get the best results in regards to the testing f-score and sample data distances when setting this parameter to 1.0. Any value higher would result in higher sample distances, and lower values made the testing f-score worse.

reject_num: This parameter sets a limit to the minimum number of members allowed in a cluster. Much like the **max_num** parameter, limiting the amount of clusters allowed greatly impacted the testing f-score and sample data distances. Because of this negative impact, I kept this value at 0.

shared_ngrams: This parameter allows the report to extract a shared amount of members from clusters in n-grams. Unlike the **reject_num** and **max_num** parameters, this had no impact on the testing f-score and sample data distance results. This could be because the values I tested (1.0, 5.0, and 10.0) did not meet the ratio criteria needed to merge any members of clusters made within the report. The higher the value set, the more memory that was used during report generating, which caused a longer wait period. Since this parameter didn't help or hurt the results, I kept it at 0.0 as well.

References:

1. Brownlee, J. (2020, August 1). How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. Machine Learning Mastery. https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/
2. Pathak, R. (2021, December 15). ANOVA for Regression - Towards Data Science. Medium. https://towardsdatascience.com/anova-for-regression-fdb49cf5d684
3. Rieck, K., Trinius, P., Willems, C., Holz, T. (2011). Automatic Analysis of Malware Behavior using Machine Learning. Journal of Computer Security, IOS Press. http://www.mlsec.org/malheur/docs/malheur-jcs.pdf
4. Rieck, K. (2012). Malheur - Automatic Analysis of Malware Behavior. Machine Learning for Computer Security. http://www.mlsec.org/malheur/manual.html
5. Ullman, S., Poggio,T., Harari, D., Zysman, D., Seibert, D. (2014). Unsupervised Learning Clustering. Center for Brains, Minds & Machines. http://www.mit.edu/~9.54/fall14/slides/Class13.pdf