# Data Science Capstone Projects - CSAFE Assessing and modeling quality of 3d topographic scans of fired bullets

**Project Sponsor: Heike Hofmann**

**Project Advisor:** Adisak Sukul, Associate Teaching Professor, Department of Computer Science

**Team Members:** Jacob Baalson, Yangfan Cai, Michael Egle, Kedell Guevara, Jacob Townsley

**Additional information**:

**Project description / goals:**

In CSAFE's process of scanning bullets for future analysis, there are cases where, for a variety of reasons, the scan turns out defective and can't be used. Scans can have holes, columns of missing data, chunks of missing data around their edges, and lighting deficiencies. In all of these cases, the scans must be sent back for rescans at CSAFE's lab so they can be improved. For a long time, Dr. Heike Hofmann, our project advisor, has manually sifted through completed scans and individually identified bad scans that require rescans.

Figure 1 - An example of a good bullet scan

The goal of our group for this project was to automate and standardize the process of identifying poor scans. Because of this, Dr. Hofmann's job of inspecting scans will be much quicker because she will have data that will suggest whether a scan is good or bad without looking at all the scans. She can simply send bad scans back for rescans and hold onto the good scans. In addition, the good and bad scans will be more standardized. Depending on when the inspector judges scans, the definitions of good and bad scans may fluctuate. With the help of this project, impartial data will be used to make decisions and these decisions will remain constant. With these new tools at Dr. Hofmann's disposal, CSAFE should be able to expedite the process of rescanning bullets and utilize more consistent bullet scans in the future.

There are a few tell-tale signs of scans that have problems. The first, and possibly most common, sign is called feathering. This occurs when there are thin, vertical holes in a scan. Another issue is called tank rash. Tank rash appears as a large, arc-shaped hole. Other problems include small holes distributed across the scan, called speckling, large and possibly streaky holes near the bottoms of scans, and other medium to large holes.

Figure 2 - Issues with bullet scans (L to R from top: lighting, staging, hole, tiny hole, speckles of missing, feathering, and rotation)

**Introduction**

The Center for Statistics and Applications in Forensic Evidence (CSAFE) is an organization that seeks to make developments in the field of forensics using data and statistical methods. One specific technique it uses is the digital scanning of bullets fired by firearms. The data collected from these scans can eventually be used to match bullets with the barrels they were fired from with a certain degree of confidence. This is possible because a gun barrel leaves distinct markings on a bullet fired from it, and matching occurs after analyzing similarities between barrels and bullet scans.

**Methodology**

The first step in implementing the algorithm to classify scans was feature, or variable, derivation. When a bullet is scanned, a digital representation of the bullet is created. This is saved as an x3p file. An x3p file contains information about a scan including a surface matrix that tracks the presence of values for each x- and y-coordinate, and metadata about the scan. The surface matrix is the most important piece of an x3p file for the purposes of our group's work.

The goal of feature derivation was to use the information stored in x3p files to identify what data values constitute good and bad scans. The variables that our team created and that were included in our final machine learning model were named assess_percentile_na_proportion, assess_middle_na_proportion, extract_na, assess_col_na, assess_rotation, assess_bottomempty, and lighting_protocal. For all of these functions except for lighting_protocal, larger values are typically common among bad scans. lighting_protocal is a categorical variable with two levels, 20% autolight and flood lighting.

assess_percentile_na_proportion is a function to quantify missing values in the middle portion of scans. It takes in parameters called chopoff, numlines, and percentile. Chopoff is a value between 0 and 0.5 that is the proportion of the scan that is disregarded on the left and right sides and is used to determine the boundaries that constitute the middle of a scan. Numlines is a positive integer that represents the number of rows sampled in a scan. This function samples rows instead of using all rows to increase its efficiency with little drawback to sampling. Percentile is a threshold for missing values in a row. This function operates by sampling rows in a scan and returning the proportion of rows that have missing values surpassing the threshold percentile.

assess_middle_na_proportion is similar to assess_percentile_na_proportion. It takes in a chopoff parameter and uses it to find a subset of columns. It then calculates the proportion of missing values within the middle region of a scan.

extract_na is a simple function that works over an entire scan with no chopoff. Its result is the proportion of an entire scan that is made up of missing values.

assess_col_na is a variable that takes in perc_of_col and threshold_prop as parameters. Perc_of_col is an integer between 0 and 100 that represents a threshold for missing values in a given column. Threshold_prop is a value between 0 and 1 that represents a threshold for the number of columns that exceeded the previous threshold. The result of this function is the quotient of the number of individual columns that exceeded the within-column threshold and threshold_prop.

assess_rotation makes calculations based on the sides of scans. It takes in a parameter called width. Width is a value between 0 and 0.5 that is the proportion of the scan on both the left and right sides that is inspected. The proportions of missing values in each section are calculated. The output of this function is the absolute value of the logarithmic transformation of the left side proportion divided by the right side proportion.

assess_bottomempty. This function takes n_cutoff in as a parameter. This value is between 0 and 1 and represents the proportion of rows, starting at the bottom of a scan, that are taken into consideration for calculations. The return value of this function is the proportion of missing values in this bottom section of the scan.

lighting_protocal logs the lighting setting for a particular scan.

There are other variables that were created over the course of the project that did not make the cut for the final model. This set of variables included multiple prototypes of the functions retained for the final machine learning model that have since been improved upon.

After the creation of these variables, we optimized their parameters to create the most separation in the individual variables. This was done by comparing the distributions of values for each of the continuous variables with numeric parameters for good and bad scans for each combination of parameter levels. For assess_percentile_na_proportion, the optimal values for chopoff, percentile, and numlines were 1/7, 0.8, and 200, respectively. The optimal chopoff for assess_middle_na_proportion was 0.125. For assess_col_na, the value for perc_col_na was 20 and for threshold_prop was 0.2. In assess_bottomempty, the optimal n_cutoff value was 0.2.

**Final Product (Dashboard, Online Tool, etc.)**

Our team used many resources to reach success in this project. We stored data sets in a CyBox folder to limit access and maintain confidentiality. We used a GitHub repository to house R code and documentation as well as to store the necessary components for an R package.

To make the findings of this project accessible and usable, our team created an R package called DS401. To run analysis on x3p files, it is also necessary to use the R packages named x3ptools and bulletxtrctr. Contained within this R package are many functions as well as some example bullet scans. The functions included were helper functions, prototype functions, and the final functions that contributed to the machine learning model. This package is free to install from the internet because R is open source. The plan is that this package will continue to be used and updated in the future by CSAFE and Dr. Hofmann to improve quality and usability.

**Findings**

The purpose of our variables was to separate good quality from bad quality scans as much as possible. To evaluate their performance on this task, we decided to look at the median of good scans and bad scans for each of our variables. For assess_percentile_na_proportion, the median for good scans was 0.009169279 and for bad scans was 0.03787957. For assess_middle_na_proportion, the median for good and bad scans were 0.02382689 and 0.04615418, respectively. For extract_na, the median for good scans was 13.33729 and for bad scans was 15.86728. For assess_col_na, the median for good scans was 0.977541 and for bad scans was 1.120715. Finally, for assess_bottomempty, the median for good scans was 25.0769 and for bad scans was 15.86728. lighting_protocal is a categorical variable with two levels, 20% autolight and flood lighting.

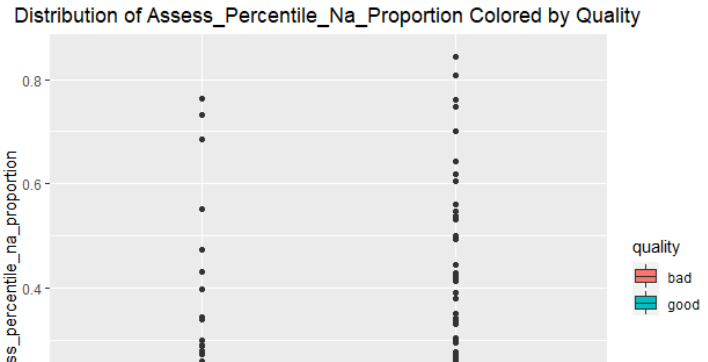

Figure 3 - Boxplots showing the distributions of assess_percentile_na_proportion scores for each quality

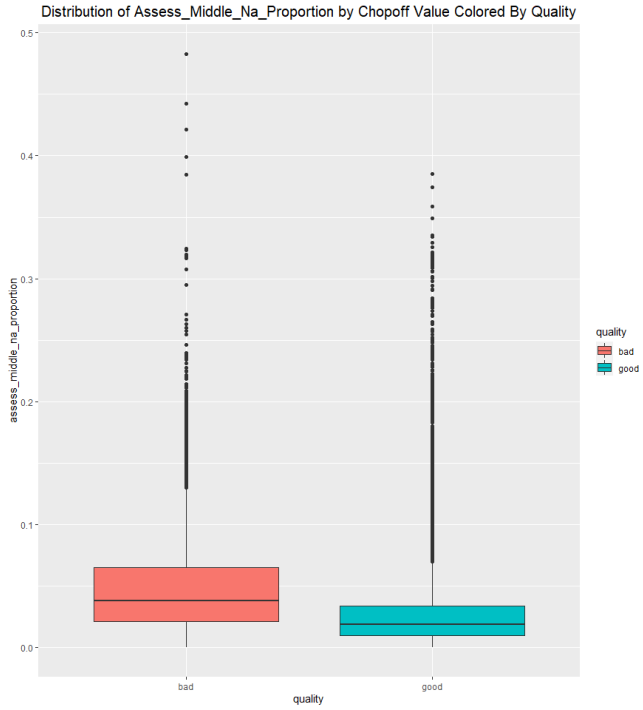


Figure 4 - Boxplots showing the distributions of assess_middle_na_proportion scores for each quality

Figure 5 - Boxplots showing the distributions of extract_na scores for each quality

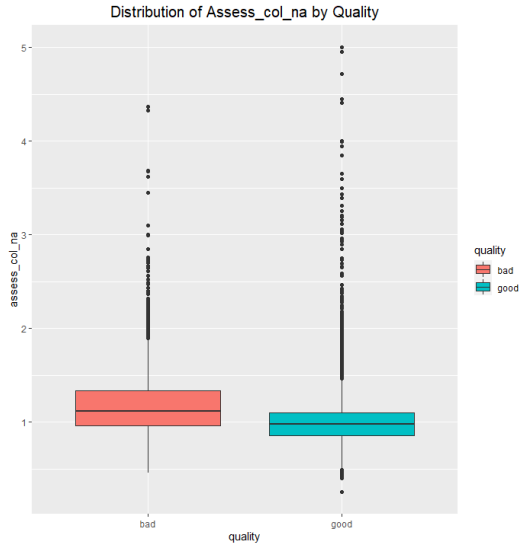


Figure 6 - Boxplots showing the distributions of assess_col_na scores for each quality

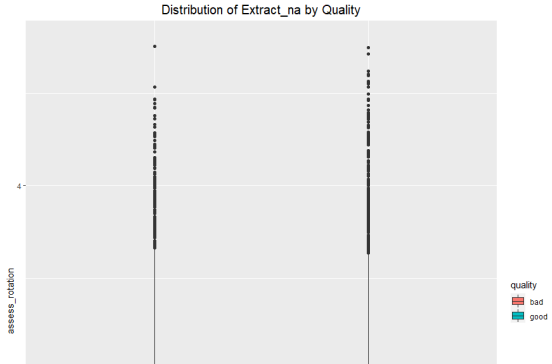

Figure 7 - Boxplots showing the distributions of assess_rotation scores for each quality

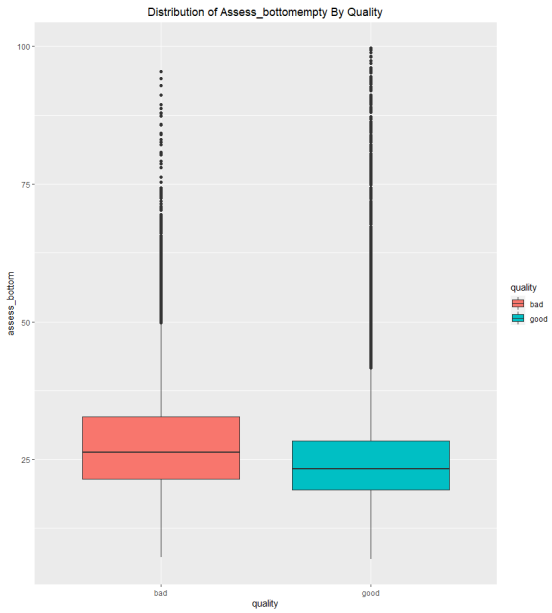


Figure 8 - Boxplots showing the distributions of assess_bottomempty scores for each quality

Because the goal of all variables used in this project was to individually classify scans, our group was aware of the possibility of multicollinearity between our features. There are high correlations between several pairs of variables, but since each variable is specialized to pick up on one or two specific scan problems, they are all necessary and serve purposes. This can be seen by examining outlier data points.

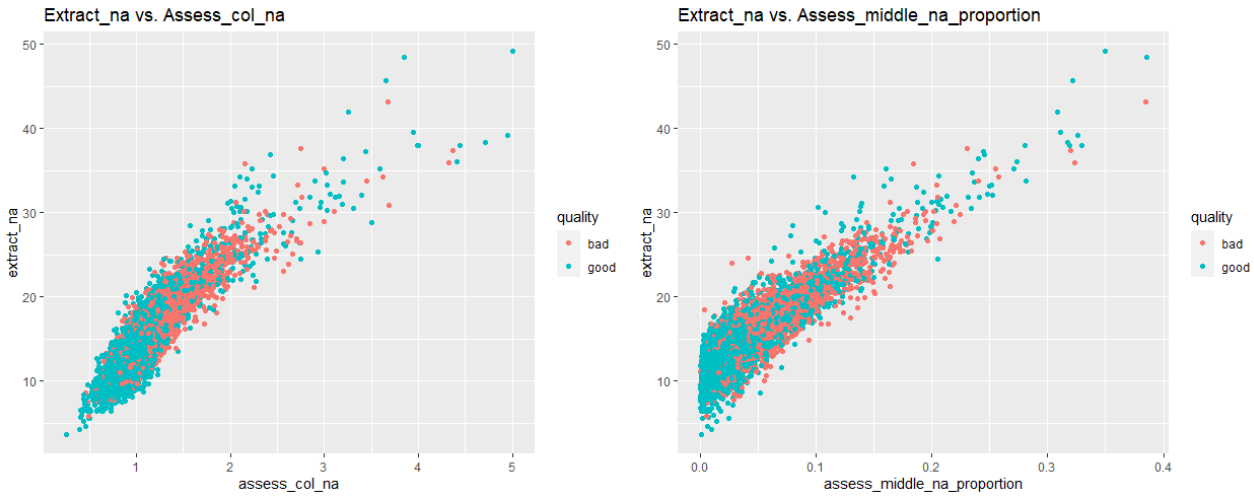


Figure 9 - Plots comparing values for extract_na against assess_col_na and extract_na against assess_middle_na_proportion

These scatter plots show the importance of different, but similar, variables. For example in both plots along the line y = 30, it can be seen that there are different trends in classification depending on the value of assess_col_na or assess_middle_na_proportion. This shows that extract_na is not a definite classifier and multiple specialized variables are necessary to make accurate predictions.

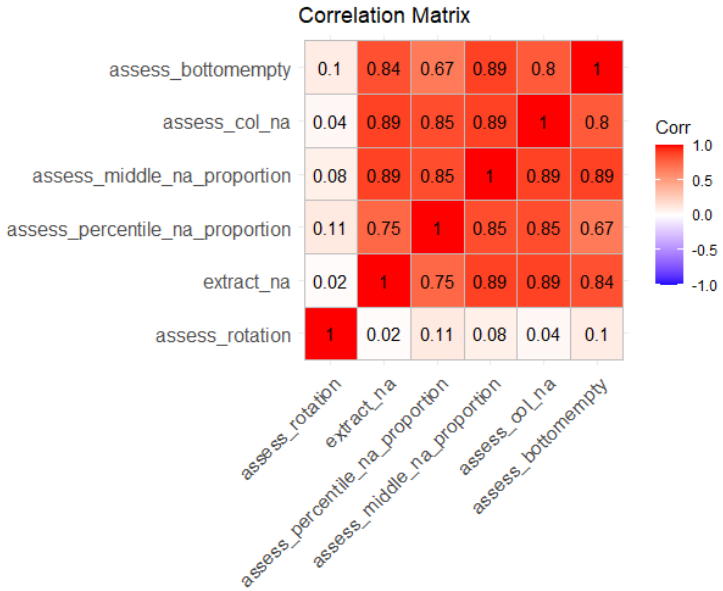


Figure 10 - A correlation matrix for the continuous variables included in the random forest models

To conduct our scan classification, our team implemented a random forest model. A random forest is a type of machine learning model designed to create many different decision

trees based on training data and classifies observations based on these trees. We chose this model because our model's purpose is classification, and random forests work very well with classification problems with a large number of observations.

The variables included in the random forest model all serve a purpose, and the model would be less effective in the absence of any one of them. assess_percentile_na_proportion and assess_middle_na_proportion both target the middle region of a scan. assess_col_na exclusively focuses on the columns of a scan. assess_bottomempty looks at the bottom of a bullet scan. These are just a few examples of how each function is specialized and were decided to be included in our final machine learning model. All of these variables, except for lighting_protocal, were developed over the course of the semester and have improved when compared to their previous iteration(s). lighting_protocal stayed consistent as it was a simple function to define categories.

To create the random forest, another function called predict_quality was employed. This multifaceted function takes in a list of x3p objects and a corresponding vector of names. This function creates a data set of each scan with their values for each function, their random forest score for quality, and the quality_type if a scan is determined to be bad. The quality_type is the predicted reason for why a scan is bad.

The primary goal of this project, as stated previously, was the creation of a random forest model for scan classification. We trained this random forest with a train-test split of 80-20, meaning 80% of the data was used for model training, and the model's performance was evaluated on the remaining 20% of the data. The number of trees used was 500, and the number of variables to choose from at every split was the square root of 5.

The focus of the model was minimizing the misclassification rate of actual bad scans because an error of a scan not being classified as bad is worse than an error of an actual good scan being classified as bad. For this reason we chose a cutoff value in random forest predicted probability of at least 0.57 to classify a scan as good quality, which was the optimal value to reduce actual bad scan misclassification rate while keeping the overall misclassification error rate at a low value. We found that this setup gave us an overall misclassification rate of 0.2047 and an actual bad scan misclassification rate of 0.2584493. Along with this random forest, there was a variable importance plot. This plot showed that order of the most influential variables for this random forest and, assess_percentile_na_proportion was the most important.

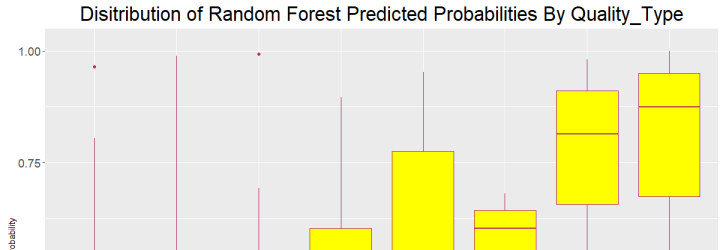

Figure 11 - Distributions of random forest scores by quality_type

A second random forest was created for the purposes of classifying the reason that a bad scan is considered bad. This was not created for our final product, but was a prototype of a system to identify which problem a scan suffers from. This random forest sought to predict quality_type based on the same variables as the previous model except for assess_middle_na_proportion. The possible classes the result could fall in were good, staging, rotation, feathering, tiny hole, hole, lighting, or speckles of missing. Similarly to the quality random forest, the most important variable for making predictions was assess_percentile_na_proportion. This model did not do as good a job of classification when compared to the quality random forest. Its misclassification rate was 0.625. However, this makes sense because we did not take time to optimize this model, and its purpose is to be used as a suggestion for the rescanner. The quality random forest is much more important to the purposes of this project than the quality_type model.

The data set also underwent a principal component analysis (PCA). The results of this PCA showed that about 89% of the variance in the data could be represented by the first two principal components. The first principal component showed that there were strong, positive correlations between all of the aforementioned variables in the quality random forest model except for assess_rotation. It shows that most of these variables contribute primarily to the first principal component in similar manners. assess_rotation contributed highly to the second principal component.

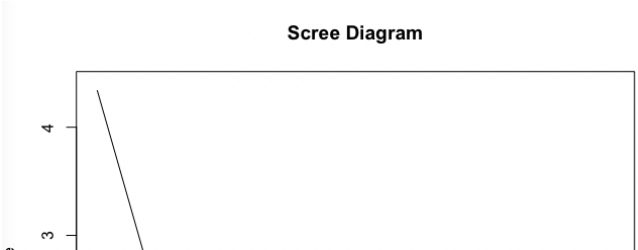

Figure 12 - A scree diagram showing the variances of each principal component

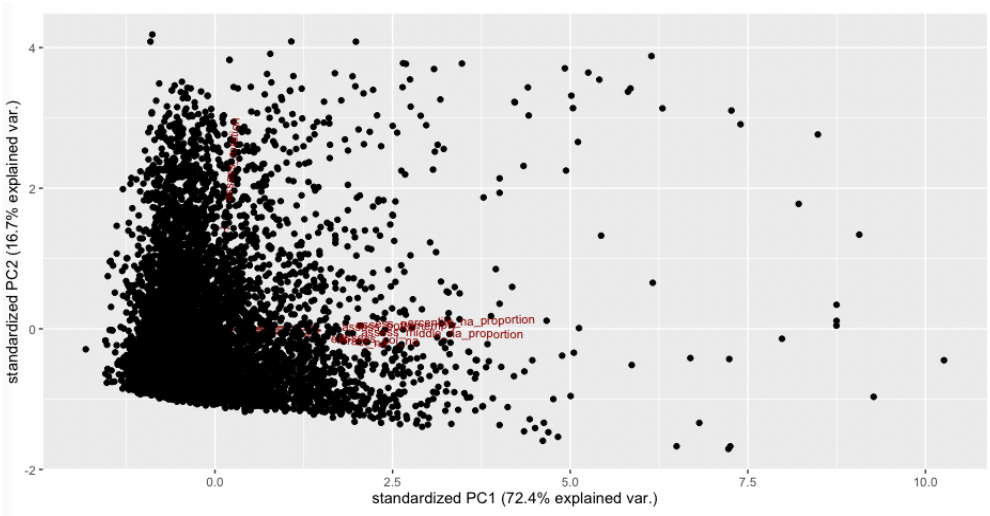


Figure 13 - The contributions of each variable in the random forest to the first two principal components

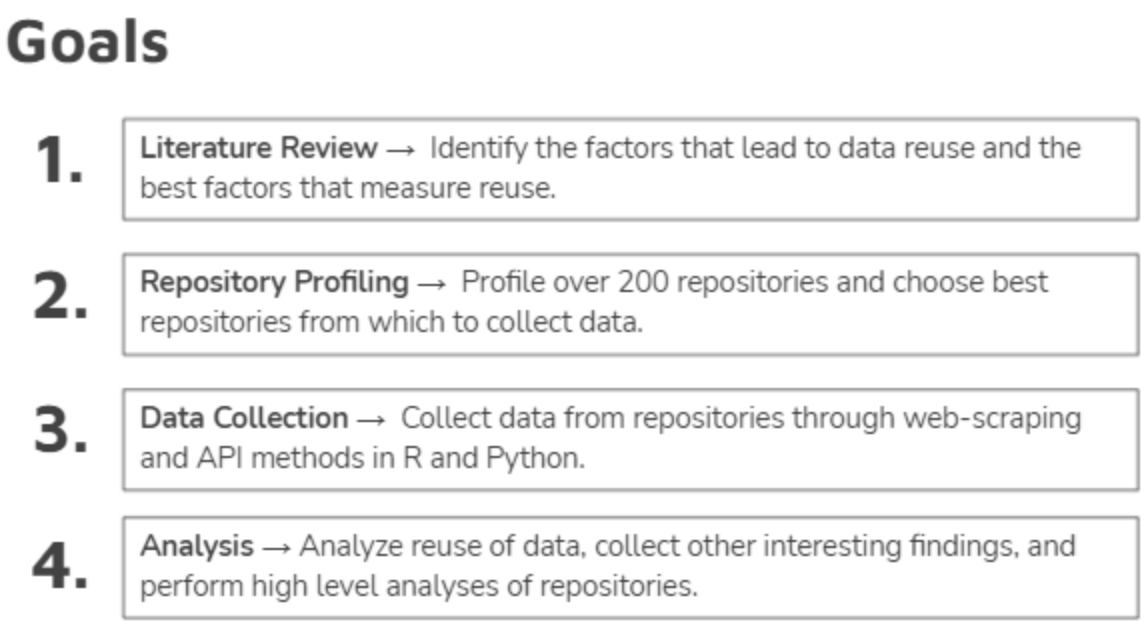


Figure 1: Specific goals for our project (by Aditi Mahabal, who is on the University of Virginia DSPG team)

**What Data is Being Used for this Project, and Why is it Relevant in Scientific Research:**

This project specifically dealt with publicly accessible research data, which is data that is accessible to the public mainly through repositories. Publicly accessible research data is one of the best tools available for helping scientific researchers propel their research process, because publicly accessible research data allows scientific researchers to easily be transparent about where they get their information from to support their research. Our client for this project was the National Science Foundation, and our mission was to convey insight to them as to what factors lead to higher data reusability for publicly accessible research data.

**Methodology / Our Approach:**

This is how our process will look like for this project:

1. Discover, inventory, profile, and document a set over 200 repositories and what information they provide about the reuse of a data source. Develop a data source with information and associated metadata such as number of citations, size of repository, number of views, number of downloads, and reusability Scores) .
2. Evaluate the reuse information derived from repositories, including what that information signals about reuse on a shared data source and the quality of the repository information on reuse.
3. Critique the usefulness of the various reuse metrics, recommend what metrics are useful and for what purpose, and suggest metrics that should be developed if appropriate.

4. Using the Python working environment (BeautifulSoup, Selenium, and ReGex package) for HTML Scraping, and Python graphical plots and Google Data Studio for data visualizations, then applying statistical analysis to those graphical plots and data visualizations.

**Standard Definitions of Metrics Scraped From Each Repository:**

The standard metrics we have encountered that helped us analyze ways data is highly reusable are views, data downloads, citations, and reusability scores. Views can be defined as the number of times a given dataset (from a repository) was viewed by other researchers. Data downloads are referred to as the number of times a given dataset was downloaded for use. Citations are represented as the number of times a given dataset was cited as evidence by other researchers. Finally, reusability scores are indications of the extent towards which a given dataset is considered highly reusable.

**How do these Standard Metrics Help Us Better Understand the Process of Making Data Reusable:**

We have observed that these standard metrics demonstrate how the process of data reusability can best be represented as a cycle. In order for data to be reused, the researcher must initially search for relevant data and later "view" it. Then, if the researcher is interested in exploring the data further, they must "download" it. Finally, if the downloaded data was used to support the researcher's various claims and statements, the researcher must "cite" the data source to ensure transparency and credibility. It's important to note that this is just a simple explanation of what the process for making data reusable looks like. However, there are various repositories that have other metadata, not just these standard metrics. Below, you'll find all of the repositories that were scraped by the Iowa State University team (if you are interested in knowing the repositories that the University of Virginia team scraped click here).

**Results from each repository:**

1. **MorphoBank** (by Tiancheng Zhou)

MorphoBank is a web application for collaborative evolutionary research, homology of phenotypes over the web, and it's a database of peer-reviewed morphological matrices. This database is

supported by the National Science Foundation (NSF), the American Museum of Natural History, and Phoneni Bioinformatics.

There are 932 publicly accessible projects as of July 29, 2021, in MorphoBank. Publicly available projects contain 142,902 images and 594 matrices. MorphoBank also has an additional 1,488 projects that are in progress. These include an additional 183,240 photos and 1,273 matrices. These will become available as scientists complete their research and release these data. Each project has details regarding number of views and number of media downloads, which are the associated pictures.

There's also details regarding the number of matrix downloads, such that a matrix is a table of taxons that shows if each taxon has uncensored and censored cells, as well as some other biological features. Hence, the matrix shows the differences between each taxon in the project. After extracting these metrics, I created a correlation table and a plot, and I found no strong correlation between any metrics.

The one that has the highest correlation is between project views and matrix downloads, which is about 0.34, so it might tell us that people usually download the matrix when they open and view a project to reuse the data. Still, we are more confident to say that the producers of the dataset might benefit from knowing that number of media and matrix downloads as a measure of impact on how reusable the data is demonstrates how people usually reuse the data based on looking at the status of the metrics that are provided by the repository.

| | Project Views | Media Downloads | Matrix Downloads |
|---|---|---|---|
| Project Views | 1.00 | 0.07 | 0.34 |
| Media Downloads | 0.07 | 1.00 | -0.02 |
| Matrix Downloads | 0.34 | -0.02 | 1.00 |

Figure 2: Correlation Table

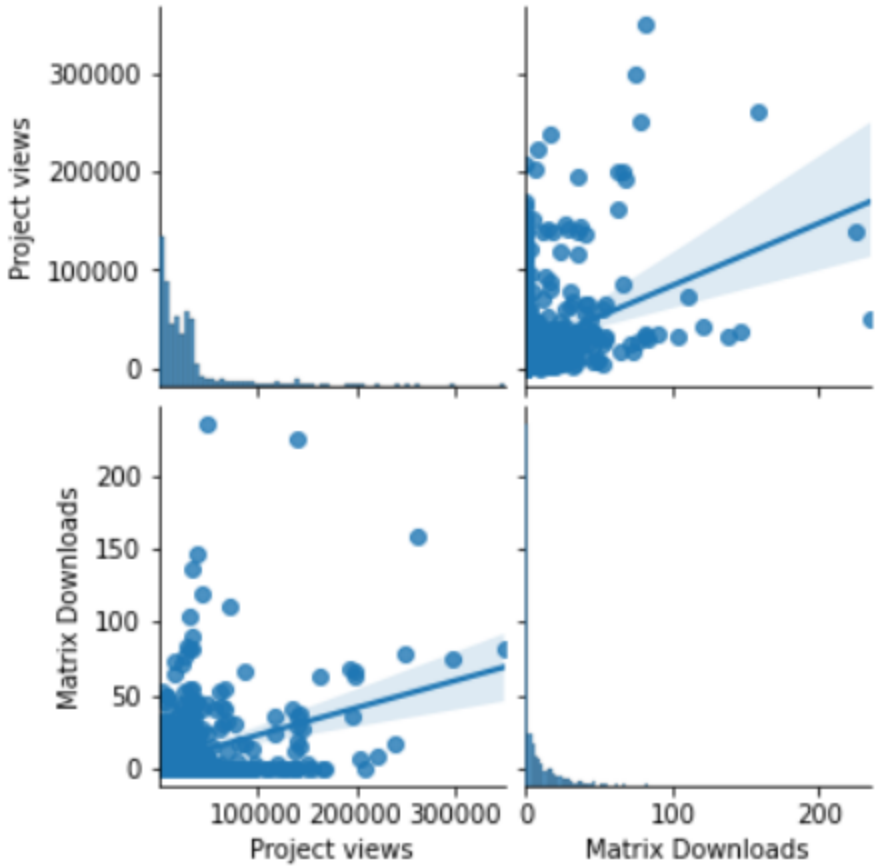


Figure 3 : Correlation plot between project views and matrix downloads

2. **Global Biodiversity Information Facility** (by Tiancheng Zhou)

Global Biodiversity Information Facility is an international network and data infrastructure funded by the world's governments and aimed at providing anyone, anywhere, open access to data about all types of life on Earth. This repository contains 61,292 public accessible datasets and are used by 6,048 peer-reviewed papers. I have extracted matrices from about 3,300 datasets; we have the number of occurrences, which are the events in the world that got recorded; the number of downloads and number of citations from each dataset. After I scraped the matrices, I created a correlation table and plot again. This time, the correlation table and plot show a strong correlation between the number of downloads and the number of citations, which is about 0.82. Hence, we can be pretty confident to say that there is a decent relationship between downloads and citations, which is also an indicator of impact for the producer's data source, showing how people manipulate the datasets to reuse them. And since this repository has a large number of datasets, which are also publicly accessible and reusable, citations and downloads would reasonably be the factors that have some impact on making the datasets reusable if we can investigate further. And lastly, since we have a very low correlation between occurrences and other metrics, we don't see how this metric has any impact on data reusability.

| | Number of Occurrences | Number of Downloads | Number of Citations |
|---|---|---|---|
| Number of Occurrences | 1.00 | 0.11 | 0.06 |
| Number of Downloads | 0.11 | 1.00 | 0.82 |
| Number of Citations | 0.06 | 0.82 | 1.00 |

Figure 4: Correlation Table

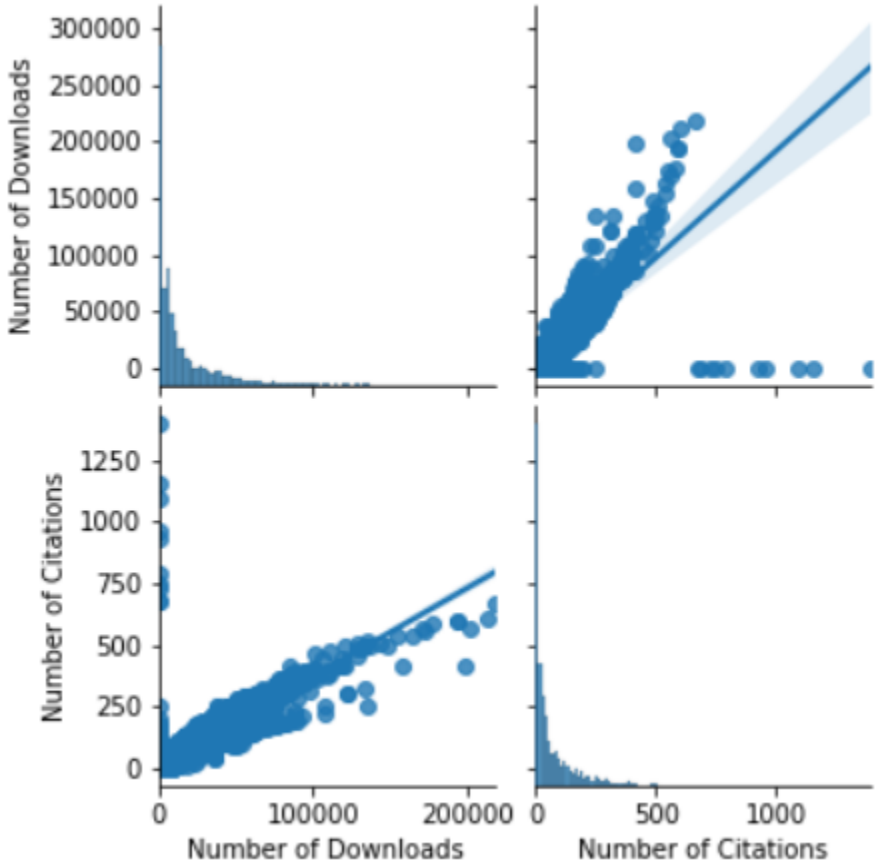


Figure 5 : Correlation plot between number of downloads and citations

3. [Astrophysics Data-System](#) (by Saul Varshavsky)

The Astrophysics Data-system Repository (ADSR) has a total of 639 datasets that were each individually scraped. Specifically, I scraped the number of citations, reads, downloads, and references from every dataset. The citations represent how many times a given dataset was cited as evidence by a different researcher, reads represent how many different times a given dataset was read (explored), downloads represent the number of times that a given dataset was downloaded for use, and references represent how many citations were referred to when collecting data for a given dataset. As I scraped the ASDR, I have encountered two major successes. The first success was being able to consistently scrape the same types of metrics from each dataset, which were the number of citations, reads, downloads, and references in this case. The second success was being able to discover a strong correlation between the number of times a given dataset was read (explored) vs. the number of times that a given dataset was downloaded. We can infer that this may be the case, because we can also see that reads and citations have a stronger correlation than reads and downloads. Reads and citations have a stronger correlation, because it appears that the more a dataset is cited by other researchers, the more that given dataset must have been read (explored). Before a dataset can be downloaded, it must be read, which supports our inference that citations and reads are more strongly correlated, since they are more directly influenced as opposed to citations and downloads. Additionally, since a dataset must be read before being downloaded, this demonstrates how reads and downloads can also have a direct influence on one another, which may explain why reads and downloads have the strongest correlation with one another compared to other variables. Refer to the visuals below for reference:

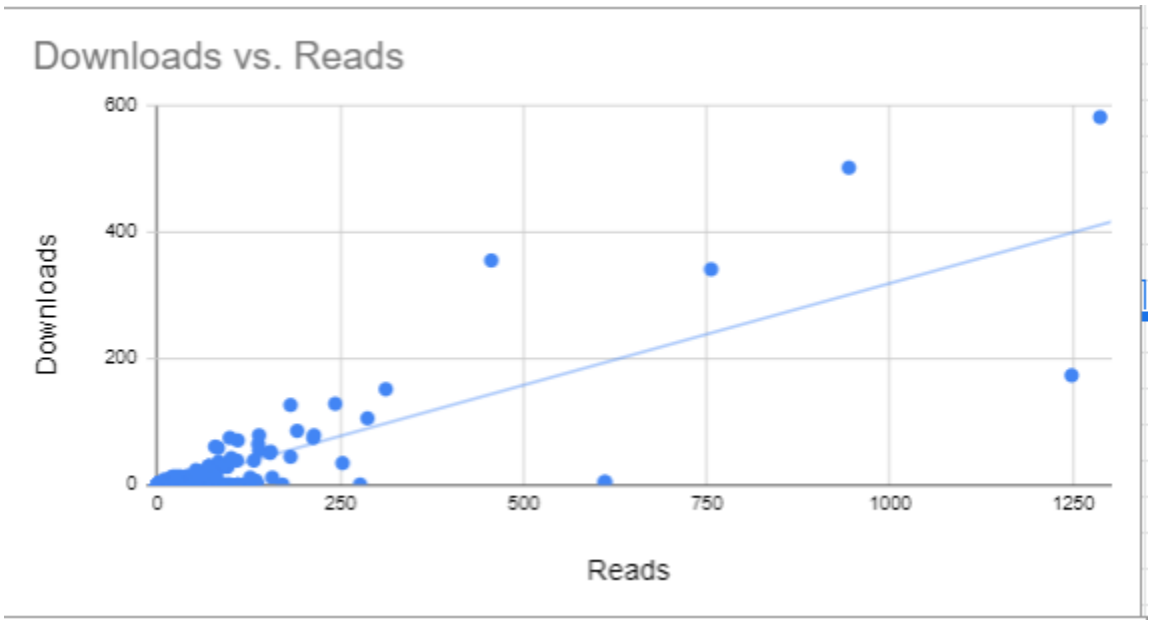


Figure 6: Linear regression scatter plot on downloads vs. reads

| | Citations | Reads | Downloads | References |
|---|---|---|---|---|
| **Citations** | 1 | 0.48 | 0.23 | 0.16 |
| **Reads** | 0.48 | 1 | 0.82 | 0.46 |
| **Downloads** | 0.23 | 0.82 | 1 | 0.55 |
| **References** | 0.16 | 0.46 | 0.55 | 1 |

Figure 7: Correlation matrix showing the correlations between citations, reads, downloads, and references

4. **NeuroMorpho** (by Sonyta Ung)

NeuroMorpho is a centrally curated inventory of digitally reconstructed neurons. It contains contributions from hundreds of laboratories worldwide and is continuously updated as new morphological reconstructions are collected, published, and shared. The goal of NeuroMorpho is to provide free access to all available neuronal reconstruction data in the neuroscience community. Throughout the research, the repository has some interesting features such as this repository allows users easy access and there are clear instructions API, repository's structure and detail, up-to-date information, as well as convenience when it comes to contributing data. The NeuroMorpho.org repository has about 5,000 publication uploads and 377,000 neurons. There are 1,681 data availability which has 176,847 neurons.

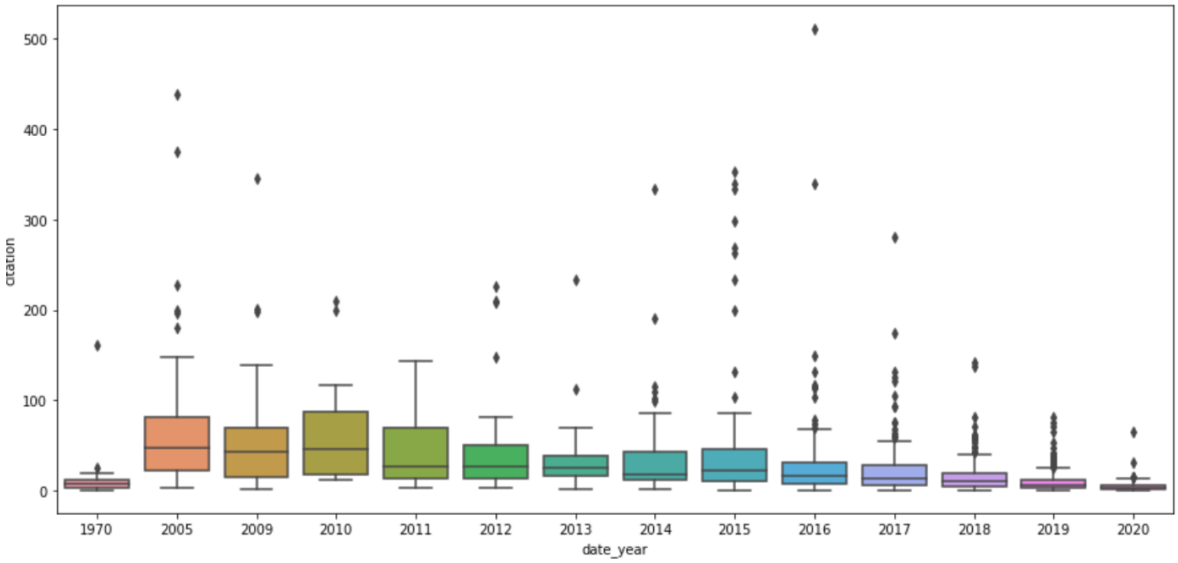


Figure 8: Whisker plot on the number of citations by year

This plot shows variaty of citations of each publication by year. Most of the publications have been cited more than one up to hundreds times. There are some outliers that make the median of these citations have variation.

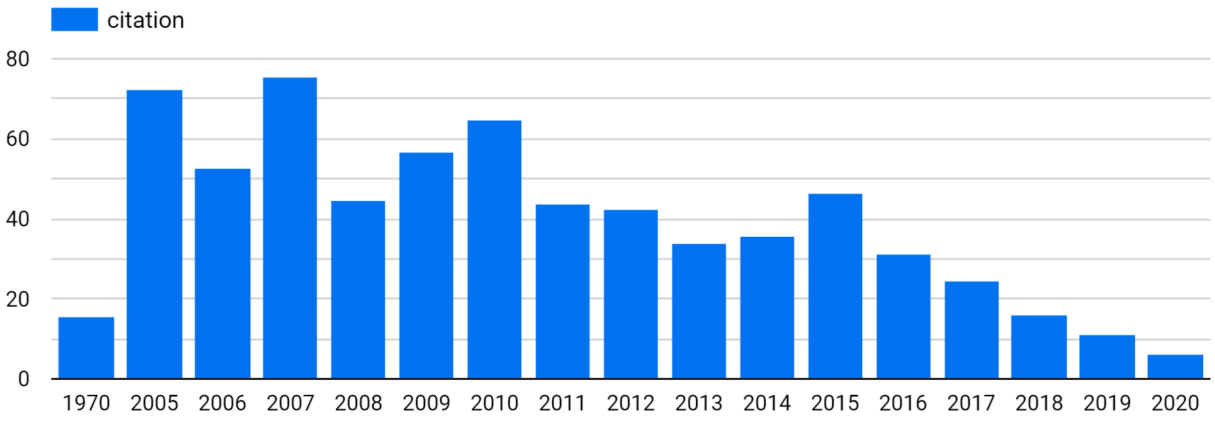


Figure 9: Bar plot on average of citations by year

The formula to compute the average of citations is the total number of citations divided by total number of publications by year. If we look closer, the average number of citations from 2005 to 2020, shows that the total number of publications makes the average citation of publications that are published for a longer period of time have higher average citation than recent published years. This brings up an important question: is the number of citations correlated to the number of publications?

5. **Kaggle** (by Jack Studier)

Kaggle is an online community of data scientists and practitioners of machine learning. Kaggle allows users to publish and explore data sets and their reusability metrics, such as size, view count, vote count, update date, download count, and their proprietary usability score. I explored the datasets by collecting 6,900 records through the API and using web scraping tools to retrieve metadata.

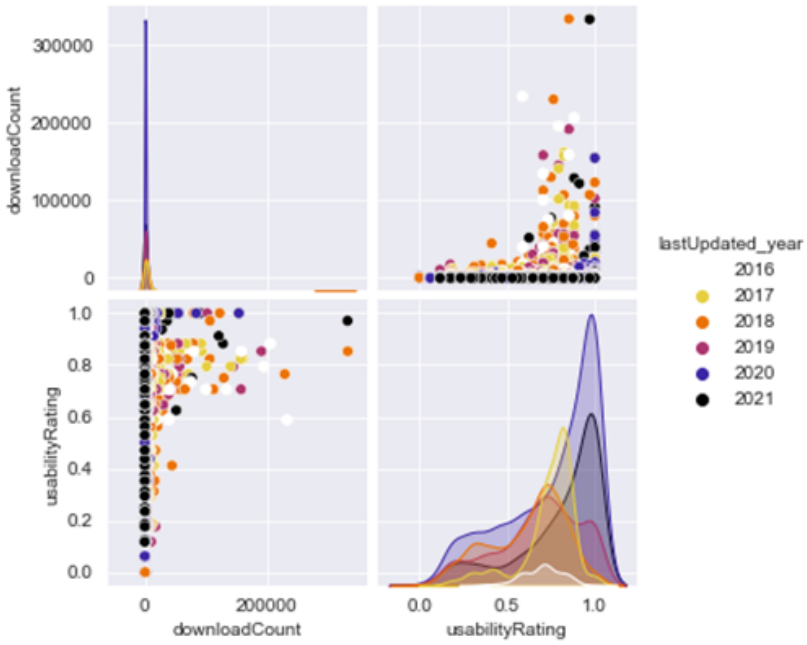


Figure 10: Graphs of download count and usability rating

Through our research, we had some interesting findings. Many of these revolved around Kaggle's usability rating. According to Kaggle's website, "It's a single number we calculate for each dataset that rates how easy-to-use a dataset is based on a number of factors, including level of documentation, availability of related public content like kernels as references, file types and coverage of key metadata" (Goldbloom and Hamner, 2010). This metric gave us some interesting findings, as seen in graphs like the figure above. There is a strong correlation between the usability score and the download count in a given dataset, which could indicate that users seek out data with a higher usability rating.

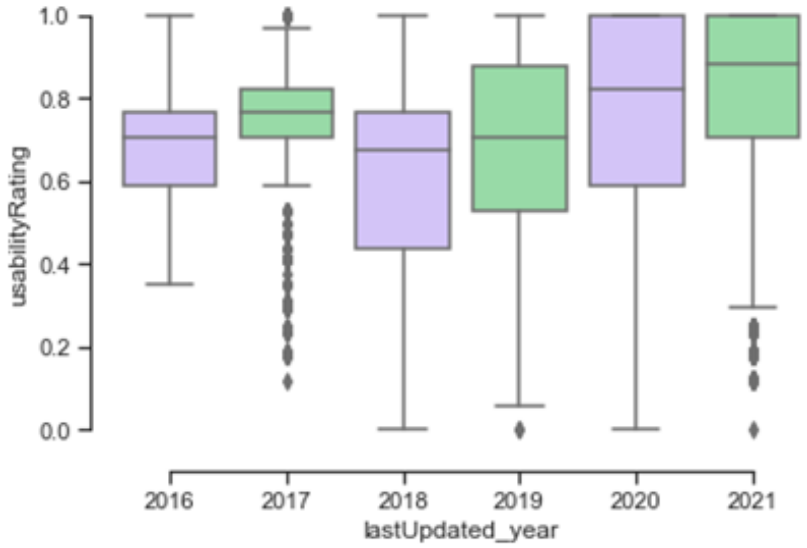


Figure 11: Usability Rating and Year of Last Update

We also found that the median usability rating of a dataset has increased overtime for the last four years. The Usability Rating was introduced in May of 2019. The datasets created prior have been retrofitted with the Usability Rating calculation. Since then, the median usability rating of datasets has gradually climbed. This could indicate that the existence of such a metric has encouraged data producers to reach a higher Usability Rating by completing the metadata requirements and propel a culture of reusability on Kaggle.

**Observations:**

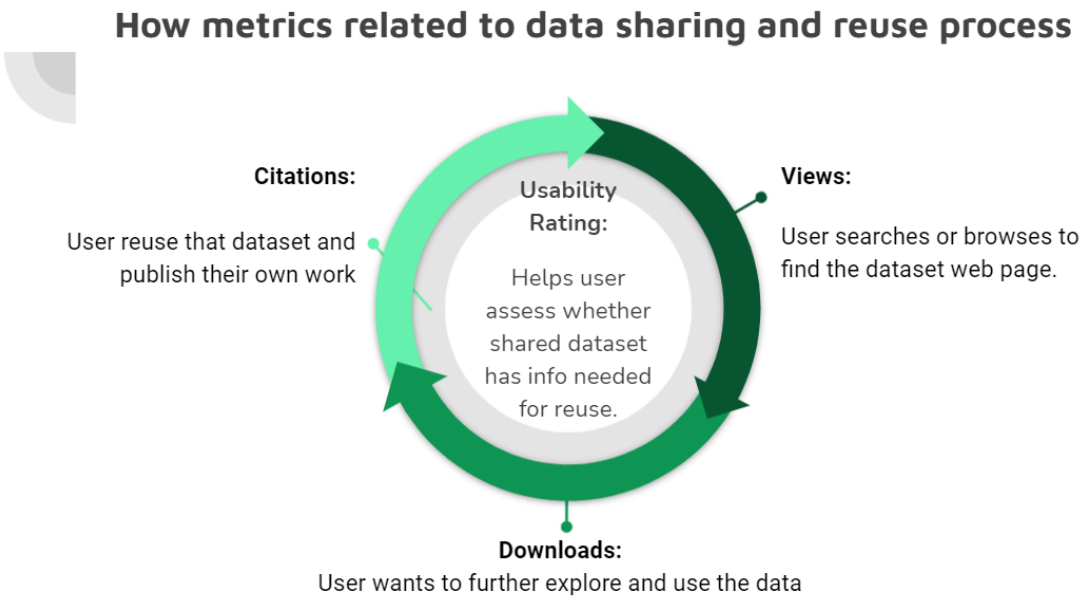


Figure 12: Data sharing, reuse process, and metrics relationship

**Conclusion and Future work:**

For this project, we have worked on obtaining metadata from several repositories, such that all of the metadata have the metrics that can impact the reusability of datasets for each repository. First, we tried using the method of API request, but we later realized that HTML scraping would be a better option for obtaining the metadata. After scraping each repository, we created numbers of plots for analysis, such as correlation plots, whisker plots, and bar plots. Based on these data visualizations, we have made observations about how each metric affects data reusability. In the future, the plan would be to collect more samples from data repositories that include different metrics we have not yet analyzed. This would allow us to further observe which factors make data highly reusable. Here are also important questions to consider when taking this project further:

How are data reusability scores correlated with data downloads, citations, and views?
How do citations and publications correlate?
How do more citations lead to more data sharing?

**Links to Our Work:**

- Data studio: Here you can find additional data visualizations related to our work
- Presentation: Here you can find the presentation that we have collaborated with the University of Virginia DSPG team and presented to the National Science Foundation, as well as the DSPG 2021 symposium
- UVA website
- Github link

**Citations:**


- Astrophysics Data-System Repository. Quick search from: https://ui.adsabs.harvard.edu/search/q=citation&sort=date%20desc%2C%20bibcode%20desc&p_=0
- Global Biodiversity Information Facility [GBIF]. Free and open access to biodiversity data from https://www.gbif.org/
- Kaggle. Dataset from: https://www.kaggle.com/datasets
- MophoBank.Org. Published project from: https://morphobank.org/index.php/Projects/Index/list_sort/author/list_sort_direction/ASC
- NeuroMorpho.Org. Availability Status of Literature Coverage from: http://neuromorpho.org/LS_queryStatus.jsp?status=Available&page=0