## Please read through this doc as well as the comments in each of the notebooks linked.

## Overview:

Our problem is to find patterns among artworks and use them to generate new groupings for future collections. I foresee two approaches:

1. **Neural Network (supervised learning)**

This approach is heavily based on previous data and patterns of past exhibitions. The Neural Network answers the following question: given a network, which past exhibitions is it likely to belong to. For example, there can be art1, art2, and art3. Based on the data of the 3 art pieces, the neural network will produce their likelihood of being in each of the exhibitions. For simplicity, there are ex1 and ex2.

If the output from the neural network for the 3 art pieces by order is

[.1, .9] -- 90% of being in ex2 for art1

[.2, .8]

[.8, .2]

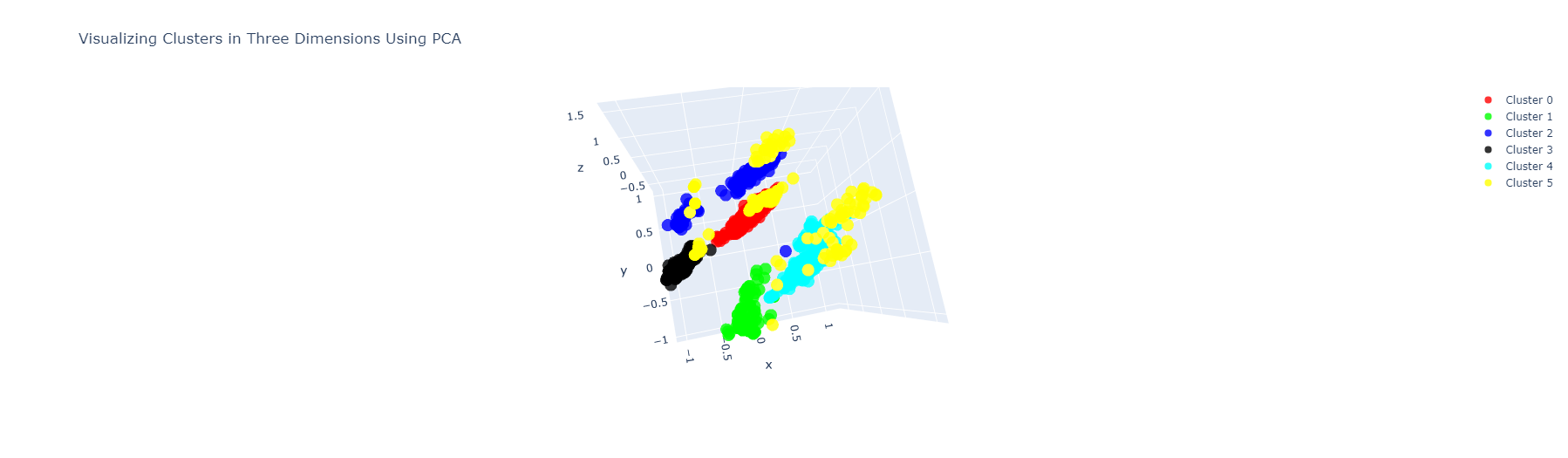
Then we can put art1 and art2 in a future grouping. Art3 would belong to a new approach. In a sense, we are measuring the distance between the output vectors of different artworks to evaluate their similarities.

**Critique:** We have a small amount of data for each exhibition, so there is likely not enough data for the network to learn and generalize from. A potential approach could be to use transfer learning, but we have to find a model that is also trained on similar data. I haven’t found any.

1. **K-means/K-proto (unsupervised learning)**

This approach immediately provides new groupings, and is quite straightforward to setup. The number of clusters k can also be found using the Silhouette metric, which describes how far each cluster is with each other. After finding each cluster, we can examine the statistics of each cluster to find the pattern within them. The challenge for this approach is that we have both numeric and categorical columns. The Euclidean distance is accurate for metric continuous variables (year), but not categorical data.

**Current Progress: 1)** Although a bit problematic, I found 6 clusters using K-means clustering. With visualization that clearly shows 6 clusters.



**2)** Lately I discovered that there is also the [K-prototype](https://pypi.org/project/kmodes/) for clustering data mixed with categorical and numerical data! I implemented a baseline version to generate clusters as well. The work of visualization and interpretation needs to be continued.

**Further Exploration: 1)** A more challenging approach is to use word2vec to embed categorical data into vectors, which will be inputted into the K-means model. However, whether this approach is more useful needs to be critiqued.

## **Conclusion**

**The K-prototype model best fits our data type, so it is the most promising candidate for further interpretation and analysis!**

**Some open challenges:**

1. **How should empty values be treated?**
2. **The “medium” column contains multiple values. How should they be treated?**
3. **How to choose the best k for k-prototype?**

**Bottleneck: The “Medium” column has multiple values inside. Each value has a stand alone meaning but combinations of values also make special meaning. ”included\_in\_exhibitions” also have multiple values if we use them as a feature. Currently, I omitted these two columns for simplicity. Also because of my limitations.**

Some approaches:

1. We can break these into multiple columns and perform one hot encoding, however if one value, say sculpture” has too low a frequency then the dummy column “sculpture” created for one-hot would become meaningless.
2. We can use word2vec to create vector embeddings of the values. This approach preserves semantics but requires more technicality. As word2vect has limitations on preserving semantics, whether the cluster center is a meaningful representative of the entire cluster is another question. I think this is the most promising yet challenging approach.

## Please read the comments in each of the following notebooks.

## Data & Code

[Hessel Dataset (preliminary](https://docs.google.com/spreadsheets/d/1dowyRB5cO0OHwF7bIFsQhPRZHsdDaLKeGBTwRP51VL0/edit#gid=716255966)): see the “Cleaned\_Data\_Schema” and “included\_in\_exhibitions” tabs. Note that this is not the final data, but the preliminary data that I use to run baseline ML models. Used by all notebooks except K-means.

[final data](https://docs.google.com/spreadsheets/d/1Fr1y3l9pDmSZ1cJWrzVMgKjWpgWy03yclnN7p3_4slw/edit?usp=sharing)

Notebooks:

* [Correlation Heat Map](https://colab.research.google.com/drive/1r0MnEhw8rx1wXAH7FXpMJWjJ6aLX6IjD?usp=sharing)
* Supervised Learning:
  + [Decision Tree](https://colab.research.google.com/drive/1YQyD_dJJO7Oeo-QLIgnQ3KXU1Ucwr7kH?usp=sharing)
  + [Neural Network](https://colab.research.google.com/drive/1ynb5GpgEwm0KIlzJKSPy7Y3by18XDOo3?usp=sharing)
* Unsupervised Learning:
  + [K-means](https://colab.research.google.com/drive/1ydwyqW7XdT_-YJcBuCLGhf_QWpvcB0Rr?usp=sharing) -- this one uses [final data](https://docs.google.com/spreadsheets/d/1Fr1y3l9pDmSZ1cJWrzVMgKjWpgWy03yclnN7p3_4slw/edit?usp=sharing)