Project B: Workflows

INFO-633 Information Visualization
Professor Chaomei Chen

Group 3: Aviv Farag, Anthony Rogers, and Kelly Wetherbee

Oct 28, 2022

Abstract – In this study workflows were designed in Orange in order to utilize various clustering algorithms on two different datasets, and to visualize the results in order to assess their quality. The aim is to create a workflow that can fit a variety of datasets with the least amount of changes.

Keywords – Workflows, Orange (software), Clustering, t-SNE, MDS.

Introduction

Orange is a software that enables users to create workflows using a variety of widgets, each of which has an input and output. This concept allows the user to create the visualization without coding. In this study, we explore the Orange software by creating workflows that implement clustering of objects from two different datasets, MNIST handwritten digits and the Movies dataset.

Data Exhibition

The MNIST dataset contains 10,000 rows and 785 columns. The first column is the target which is a number between 0-9 (inclusive). Other columns are numbers between 0 and 255 (inclusive) representing shades of gray (1-254), black (0), and white (255) as can be seen in the figure below:

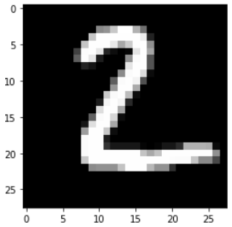


Figure 1 An instance in the MNIST dataset

The image above was created in Python with the goal of visualizing and understanding the features in the MNIST dataset. There are 784 pixels, which means each image is 28x28. After resizing the row into a 2-D array, the image was created and reveals the digit.

The second dataset is the Movies csv file. It contains features like the Movie's title (name), rating, year, budget, length, votes, r1-r10, mpaa class, and boolean features: Action, Animation, Comedy, Drama, Documentary, Romance and Short. In our study, the workflow ignores r1-r10 features, since we would like to avoid making assumptions, and a description of their meaning is missing. The number of the movie is also ignored because it simply represents an identification number, and thus doesn't contain any significant information.

Tools

Two tools were utilized in this project: Orange and Python. The portion of Python is minor since we used it to visualize and plot a few MNIST instances in order to understand the significance of the features, and also to investigate an issue we encountered in Orange. All other work, and the majority of this project was implemented in Orange.

MNIST - Digits dataset

Workflow

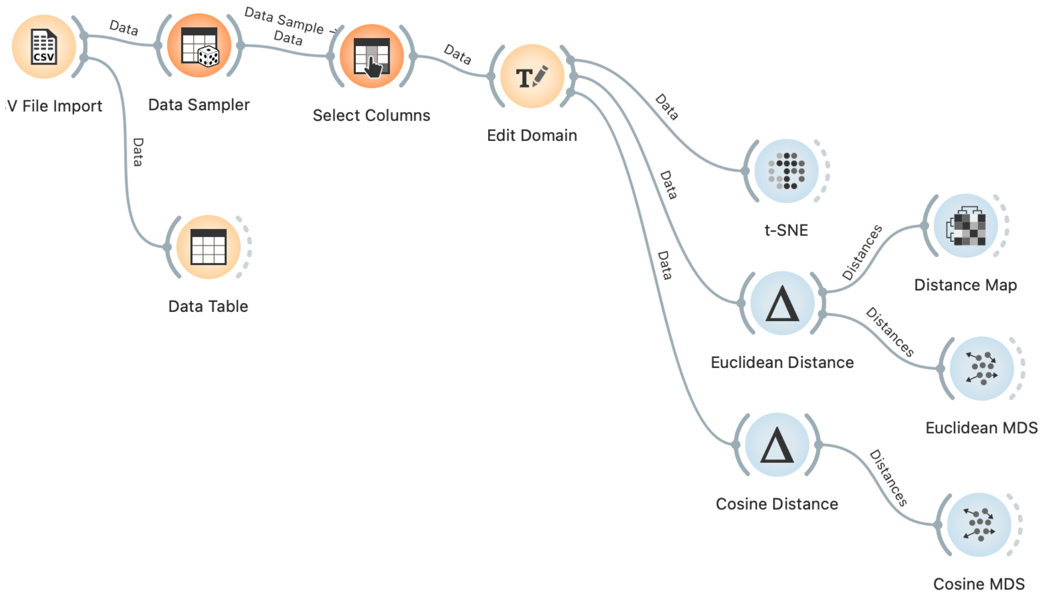


Figure 2 Proposed workflow for MNIST dataset

Figure 2 shows the proposed workflow starting with importing a csv file, sampling 3,000 instances from the data, and selecting the first column as the target. There are 2 clustering methods that were tested in this workflow: t-SNE, and MDS. The latter was implemented using both Euclidean and Cosine distance in order to explore the differences between them.

While importing the csv file, an error was encountered in some columns. Their assumed type (Categorical) didn't match their actual type (Numerical), and caused a warning sign to appear in the Cosine Distance widgets, meaning that the categorical features were ignored in the calculation, thus introducing an error in the computation done by MDS widget.

Fig. 3 Cosine Distance Warning

To further investigate this issue, Python was utilized to check the categorical columns, and we found that there was only one unique value (0) over all rows in the sampled dataset. Same results were achieved while investigating other samples, and other columns, and therefore we understand that categorical type is assumed for all columns who have only one unique value. The solution we came up with is to manually assign Numerical type to all columns that represent pixels (1-784), thus they will not be ignored throughout the workflow.

Quality

First clustering method that was utilized is t-SNE, and following hyperparameter tuning the results are shown below:

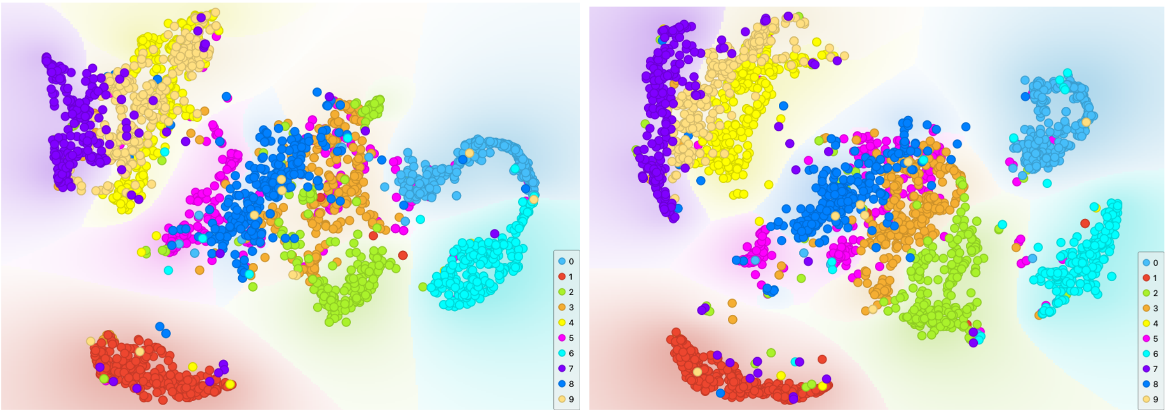


Figure 4 t-SNE clustering. Left image - PCA = 10, right image PCA = 50, in both images perplexity = 50.

The images show ten clusters (0-9), each of which has a different color, and the regions of each cluster are also painted in the same color. Those concepts follow the Law of Proximity, the Law of Similarity (same color), and the Law of Common Region [1]. In accordance with the 7 tasks [2], at first glance (overview), the images look similar, but a zoom-in reveals several differences.

First of all, the clusters representing 0, and 6 are connected at the left image, while they are completely separated on the right image, thus classifying is better on the latter. The same could be said regarding clusters representing the digits 4, and 3 (near the purple cluster). Additionally, the cluster representing the digit 2 is also better classified in the right image. However, there is more misclassification occurring in clusters 8, and 5 in the right image. Overall, the quality of t-SNE is good since there is a defined separation between most clusters.

Second method is MDS. For that method the "Distance" widget was used twice, once for Euclidean and another for Cosine. The results are shown below:

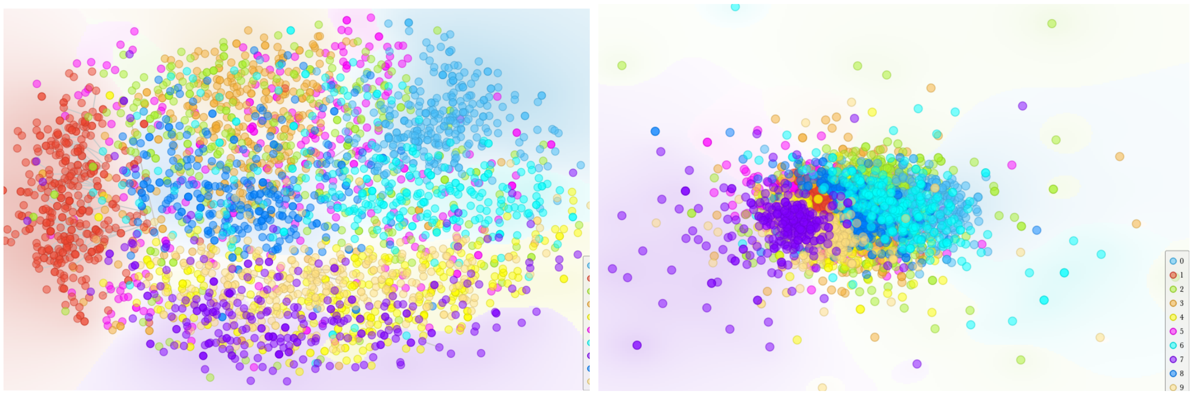


Figure 5 Cosine Distance MDS (left), and Euclidean Distance MDS (right).

MDS results are the worst achieved in this workflow. The left image shows MDS following Cosine distances and the right image shows MDS following Euclidean distances. The Cosine MDS is better than Euclidean because there is a separation between the red, purple, light blue and the other clusters, whereas in the Euclidean it seems that the clusters are all stacked in the same region. Moreover, the colors of the regions are more apparent in the Cosine MDS which also implies that the quality of the clustering is better than the Euclidean, where there is only one color (purple) that is greatly marked.

However, the distance between each element in a specific cluster is greater in the left image, thus reducing its overall quality. To put it simply, the closer the instances within a cluster, the greater its quality.

Movies

Workflow

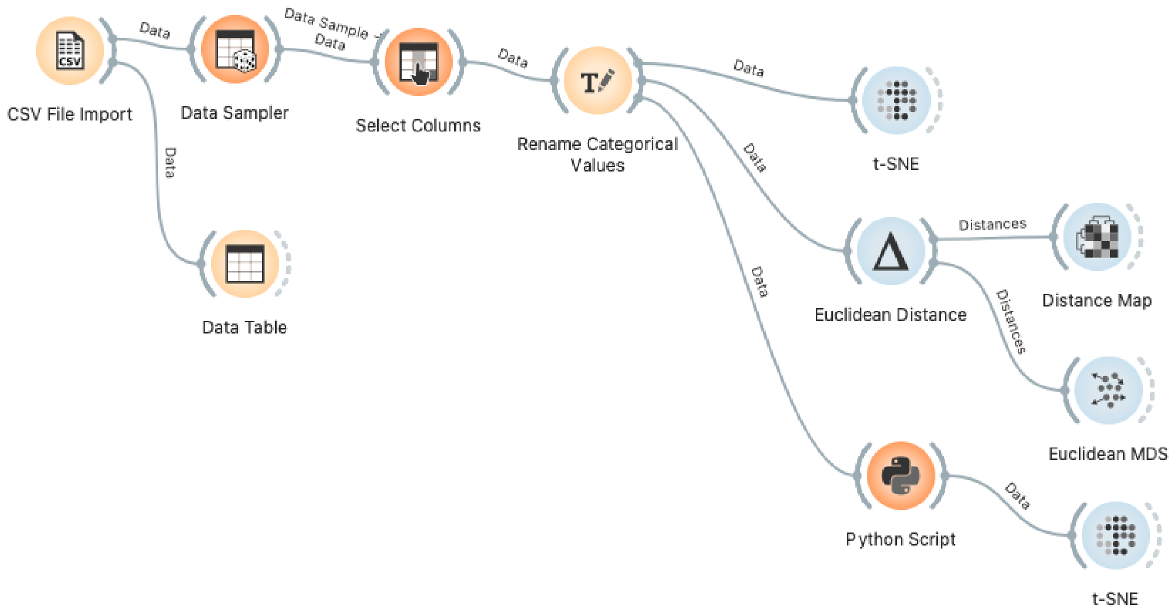


Figure 6 Proposed workflow for Movies dataset

The Movies workflow is similar to the MNIST workflow. There are only three major differences between the workflows: renaming categorical variables, removing Cosine MDS, and adding a Python Script in order to visualize movies that do not fit any category (have 0 for all genres).

Renaming categorical variables was necessary for the legend in the final visualizations. The categorical features (Short, Comedy, Action, etc.) are binary features that have 1, indicating True, and 0, indicating False. For each feature, the values were mapped as strings (0 - Other, 1 - feature's name).

Removing the Cosine Distance widget, as well as the following MDS, was necessary because Cosine distance widgets ignore categorical features, and therefore introduce computational errors that should be avoided.

Last modification is the Python Script that was added in order to add a feature to mark all instances that do not fit any of the following categories:

- Short
- Action
- Romance
- Animation
- Drama
- Comedy
- Documentary

Quality

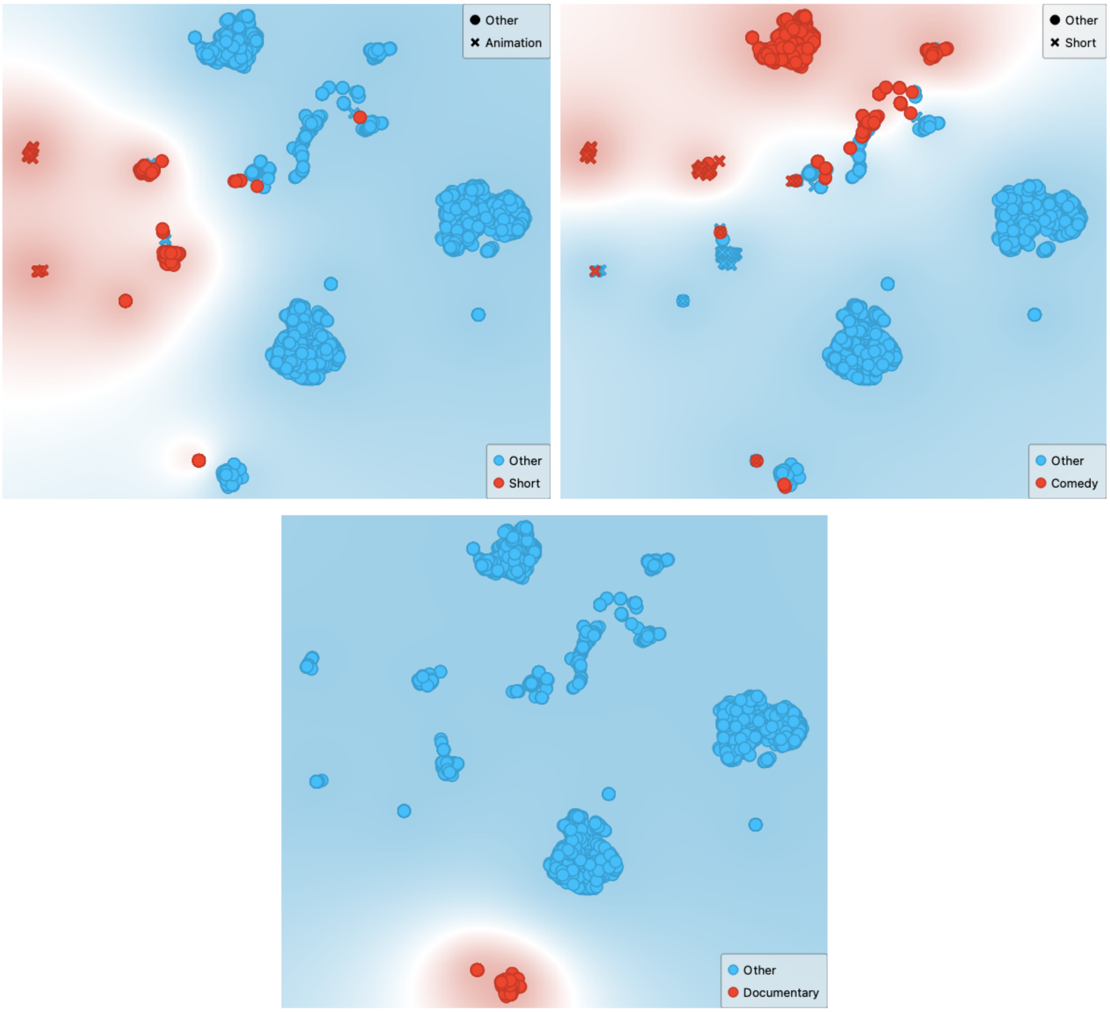


Figure 7 t-SNE clustering.
Top Left - Animation (Shape) and Short (Color) features
Top Right - Short (Shape) and Comedy (Color)
Bottom - Documentary (Color)

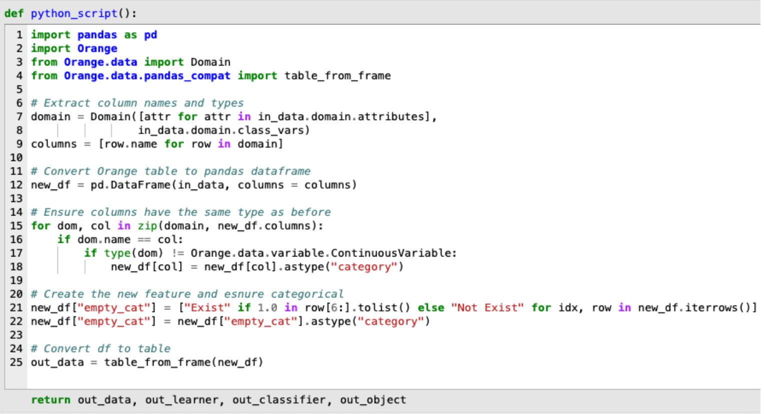
The top left image shows a clear separation between short (red) and not short movies (blue). There are several instances that were missed such as one movie appearing red in a blue region and vice versa. Additionally, animation movies are presented as ‘x’ and most instances are inside the “short” region and are marked as short. Once again, there were a few instances that were misclassified, but overall the separation is clear.

The top right image presents the Comedy cluster in red, and short movies marked as ‘x’. Here the region separation is also clear, but there are more instances in the border, thus the quality is worse. This is also true when looking at short movies, some of them are Comedy and some aren’t, thus there is no correlation between these two features.

Moreover, in both images on the top, one can see that inside the red zone there are sub-groups. This is the result of mapping high dimensional data in a 2D image. In other words, there are other features that were taken into account during computation, features that sub-groups share in common. Our visualizations in Orange are limited to Color, Size and Shape, thus it is necessary to create several visualizations in order to properly visualize all clusters and sub-clusters.

The last image on the bottom shows Documentary movies. This cluster is split into sub-clusters, the main one is for movies that are documentaries, and the small one (on the left), which is red, is for both documentary (bottom image) and comedy (top right image).

By zooming in and further examining the clusters, our study reveals a unique cluster. In order to show it in t-SNE a Python Script was added to the workflow:



```
def python_script():
import pandas as pd
import Orange
from Orange.data import Domain
from Orange.data.pandas_compat import table_from_frame

# Extract column names and types
domain = Domain([attr for attr in in_data.domain.attributes],
                in_data.domain.class_vars)
columns = [row.name for row in domain]

# Convert Orange table to pandas dataframe
new_df = pd.DataFrame(in_data, columns = columns)

# Ensure columns have the same type as before
for dom, col in zip(domain, new_df.columns):
    if dom.name == col:
        if type(dom) != Orange.data.variable.ContinuousVariable:
            new_df[col] = new_df[col].astype("category")

# Create the new feature and esnure categorical
new_df["empty_cat"] = ["Exist" if 1.0 in row[6:].tolist() else "Not Exist" for idx, row in new_df.iterrows()]
new_df["empty_cat"] = new_df["empty_cat"].astype("category")

# Convert df to table
out_data = table_from_frame(new_df)

    return out_data, out_learner, out_classifier, out_object
```

Figure 8 - Python Script

The script simply creates a new feature that has the values “Exist” if one or more categories (short, documentary, comedy etc.) are 1, otherwise “Not Exist”. The results of the new t-SNE clustering appear in Fig. 9. Overall, the results of t-SNE clustering are great, and show natural grouping with clear separations between clusters. Additionally, in each cluster, the elements are close to one another, meaning they share features and are similar in nature. Since the data is high dimensional, there are many clusters to explore.

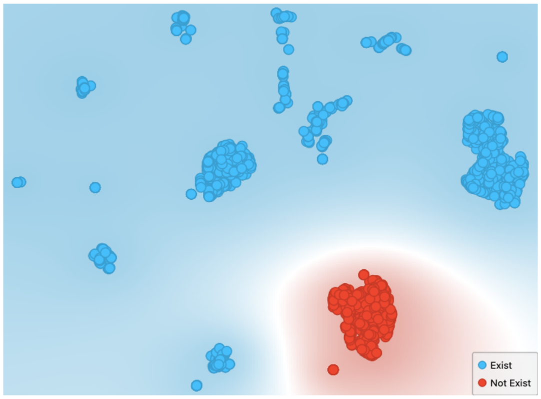


Figure 9 t-SNE clustering. Red cluster shows movies whose genre is unknown

MDS was also utilized, following Euclidean distance widget, and the results are worse than t-SNE:

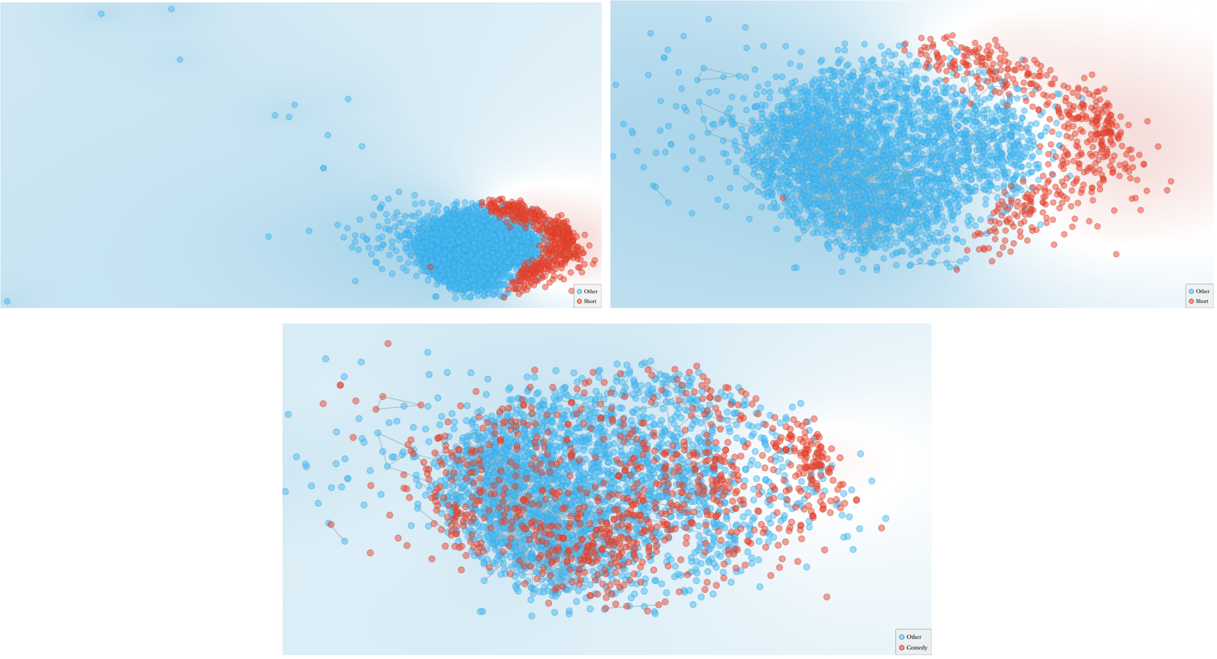


Figure 10 MDS clustering.
Top Left - Overview of Short movies.
Top Right - Zoom-in Short movies
Bottom - Zoom-in Comedy movies

MDS results show the clustering of Short movies (top images), and Comedy movies (bottom image). Comparing MDS and t-SNE results, the latter have better quality because of the clear separation between the clusters.

The top left image shows an overview of MDS clustering where red dots are Short movies. Then, a zoom-in was utilized on the top right image in order to get more details about that clustering. The border seems to have both blue and red dots, thus reducing the overall quality of such clustering. To further investigate, the bottom image was created showing Comedy movies which are spread all over the blue cluster, therefore, the t-SNE results are better.

Discussion

Orange software allows designers to create a workflow that contains different types of widgets from importing files, transforming the data, creating a model, and visualizing the data. One simply drags and drops the widgets, and connects them to create the desired workflow.

However, the platform is not perfect, and has some drawbacks that were encountered in our study. While using t-SNE clustering for movies dataset, a unique cluster was identified after the zoom-in task. It was marked as unique because the movies in this cluster do not fit any category (features). To visualize this cluster a Python Script was required to create a new feature. It could be useful to be able to manually add and change colors of the clusters in the image, or add annotations.

There was another issue with the type of some columns which affected the cosine distance widget. It was necessary to change column type from *Auto* to *Numerical* while reading the csv file in order to prevent computational errors in the workflow.

Conclusion

This project was an exploration in how to visualize two data sets in Orange software workflows. We produced various types of visualizations in order to explore how the same data set can be presented differently and which ways were most and least effective. We chose to use the MNIST and Movies datasets. One of our goals was to see what workflow we could create that would require the least amount of changes or modifications for the second dataset. The software was relatively easy to use but when we discovered an anomaly we chose to mediate the issue by running a Python script through the dataset inside Orange. Additionally, we were unable to change the colors in the visualizations and after recent learnings we understand these visualizations may not be as inclusive or accessible as possible.

In our study, MDS and t-SNE widgets were utilized for clustering of MNIST and Movies datasets. A comparison between those algorithms was executed, and t-SNE's results were better than MDS because the clusters could be identified more easily, and the elements within each cluster were close to one another.

References

1. Chen, C. (2022, September 27). Week 2C: Natural Groupings and the Simplicity of the Whole [PowerPoint Slides]. Philadelphia, PA; Drexel University.

2. Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, 336–343. https://doi.org/10.1109/vl.1996.545307.