## 2. User Social Networks

Fetch users may friend each other to share product recommendations, or compete for top spots on points leaderboards.

Graphs can be used to analyze these social networks.

For example, we can assess the relative influence of users, or predict the likelihood of user links or attributes. We can also detect communities.

The next data collection includes the nodes of all users who all have at least one friend in Wisconsin.

They each have one binary attribute indicating whether they, themselves, are from Wisconsin or not.

The edge data indicates which users are friended with whom.

Write a routine to ingest this data.

Select and train a model to describe how many natural communities of Fetch users exist in Wisconsin.

Your solution may include any or all of the following:

- Discuss different measures of betweenness that could apply to the problem

- Select one or more standard clustering algorithm(s) for the problem and explain your choice(s)

- Write your own clustering algorithm

- Write a training routine to arrive at your best model for this data
- Discuss how you tuned the model, especially highlighting different measures you may have tried
- Describe your clusters in terms of internal characteristics that unit them, or boundaries that separate them

Data(x=[19688, 2], edge_index=[2, 31198]]

Data:

What we have:

Data(x=[19688, 2], edge_index=[2, 31198]]

The data (x=[19688, 2], edge_index=[2, 31198]) indicates a graph structure where each row of the x array represents a user node, and the edge_index array represents the connections between users (i.e., friendships). Each column in the edge_indexarray represents an edge connecting two nodes (i.e., two users are friends).

I have a file x.csv. In this file I have two columns named 'index' and 'LABEL'.

x.csv file includes nodes of all users who all have at least one friend in Wisconsin.

Lets say 'index' refers to a unique identifier for each node, while 'LABEL' represent a category or class label assigned to each record.

Lets say 'Label' is a binary attribute indicating whether they, themselves, are from Wisconsin or not.
The values in the 'Label' column are either 0 or 1. 1 means the user is from Wisconsin and 0 means not from Wisconsin.

I have another file edge_index.csv. In this file I have two columns named 'index_x' and 'index_y'.

The edge data indicates which users are friended with whom.

Index_x and index_y are two users who are friends.

The Limitation of the given data:

To better understand and analyze the data, here are some additional information that could be helpful:

1. Column descriptions: We do not have descriptions for the 'index' and 'LABEL' columns in x.csv. These descriptions can provide important context for understanding the data.
2. Node attributes: A file that contains information about the nodes in the graph, such as user profiles or demographic information. These attributes can be used to improve the accuracy of the clustering results.

The edge_index.csv file contains information about connections between pairs of users, but without additional information, it is very difficult to determine the number of natural communities of users that exist.
Additional data that could be useful includes user profile information such as demographics, interests, or social network data, which could help identify clusters of users with similar characteristics.

3. Edge weights: The 'edge_index.csv' file may not contain information about the strength or weight of the connections between nodes. This information can help to identify more meaningful communities of users.
4. The limitation in data can have several impacts on clustering analysis. Limitations in the data, such as a small sample size or missing features, can affect the accuracy, generalization, and interpretability of the clustering results.

Additional data that could be useful includes user profile information, demographic information, or social network data, which could help identify clusters of users with similar characteristics.
Demographic data: This could be helpful to investigate the demographic characteristics of natural communities of Fetch users in Wisconsin. We may need additional demographic data, such as age, gender, occupation, or education level.
Social media activity data: This could be helpful to investigate the social media activity of Fetch users in Wisconsin. We may need additional data on their posting behavior, engagement with other users, or content preferences.

User profile data: This could be helpful to investigate the interests or preferences of Fetch users in Wisconsin. We may need additional data on their user profiles, such as their bio, interests, or topics of discussion.
Historical data: This could be helpful to investigate the evolution of natural communities of Fetch users in Wisconsin over time. We may need additional historical data on their network connections or community memberships at different time points.

5. Difficulty in finding the desired number of clusters: It is also possible that the data might not be well-separated, making it difficult to find the desired number of clusters.
6. Small dataset: Having very few samples can make it difficult to identify meaningful clusters.
7. Additional data about the network and its properties can be useful to determine which measure of betweenness is most appropriate for the problem at hand.

## Select and train a model to describe how many natural communities of Fetch users exist in Wisconsin.

Selection of the model for predicting the number of natural communities of users in Wisconsin:

There are several machine learning models that can be used for predicting the number of natural communities of users in Wisconsin. The choice of the model depends on various factors such as the size of the dataset, the complexity of the problem, the nature of the features, and the desired performance metric.

Some alternative models to Random Forest Regression that could be suited for this problem are:

1. Gradient Boosting Regression: It is a tree-based ensemble model like Random Forest, but it builds trees in a sequential manner, where each subsequent tree tries to correct the errors of the previous tree. It can handle complex non-linear relationships between the features and the target variable.
2. Support Vector Regression: It is a kernel-based model that tries to find a hyperplane that maximizes the margin between the predicted values and the actual values. It can handle high-dimensional data and non-linear relationships between the features and the target variable.
3. Neural Networks: It is a class of models that can learn complex non-linear relationships between the features and the target variable. They can be used for both regression and classification problems. However, they require more data and computational resources compared to other models.

In general, Random Forest Regression is a good starting point for this problem. If you find that the performance of the model is not satisfactory, we can try some of the alternative models mentioned above and see if they perform better.

The performance of the machine learning model can be measured using various metrics such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared (R2) score. The choice of the metric depends on the nature of the problem and the desired evaluation criteria. In the case of predicting the number of natural communities of users in Wisconsin, you could use MSE or RMSE as a performance metric since they penalize large errors more than small errors.

Random Forest Regression is a good starting point for this problem for several reasons:

1. It can handle both linear and non-linear relationships between the features and the target variable.
2. It can handle high-dimensional data with many features.
3. It is less prone to overfitting than some other machine learning models since it builds multiple decision trees and averages their predictions.
4. It can handle both continuous and categorical data.
5. It can provide insights into feature importance, which can help in identifying the natural communities of Fetch users in Wisconsin.

However, it is important to note that the performance of the model depends on the quality of the data, the choice of hyperparameters, and the feature engineering techniques used. It is always a good practice to try multiple models and evaluate their performance using cross-validation to ensure that the selected model is the best fit for the problem.

Here are the steps you can follow to train a Random Forest Regression model:

1. Merge the two data files: You will need to merge the x.csv and edge_index.csv files based on the 'index' column. This will give you a dataset that includes information about which users are friends with whom, as well as whether they are from Wisconsin or not.
2. Feature engineering: You can create additional features from the data that may help improve the accuracy of the model. For example, you can calculate the degree of each node, which is the number of friends a user has.
3. Split the data: Split the data into training and testing sets. Use a stratified sampling technique to ensure that the proportion of Wisconsin users is the same in both the training and testing sets.
4. Train the model: Use Random Forest Regression to train a model on the training data. You can use techniques such as cross-validation to find the best hyperparameters for your model.
5. Evaluate the model: Use the trained model to make predictions on the testing data. Calculate the mean squared error (MSE) or another appropriate metric to evaluate the performance of the model.
6. Interpret the model: Once you have trained the model, you can interpret the results to gain insights into the natural communities of Fetch users in Wisconsin.

```
Feature Importances: Unnamed: 0_x    0.0
index          0.0
Unnamed: 0_y    0.0
degree         0.0
dtype: float64
```

The results indicate that the four features listed have no importance in predicting the target variable. This means that these features are not useful in determining the natural communities of Fetch users in Wisconsin.

To gain insights into the natural communities of Fetch users in Wisconsin, you need to look at other features that have higher importances. Feature importance is a measure of how much a feature contributes to the prediction of the target variable. You can analyze the feature importances of the model to identify the features that are most important in predicting the natural communities of Fetch users in Wisconsin.

Once you identify the important features, you can use them to cluster the users into natural communities based on their similarity in those features. You can also use other techniques such as PCA or t-SNE to visualize the clusters and explore their characteristics. This can give you insights into the preferences, behavior, and demographics of the users in different natural communities.

Please refer to RandomForestRegression.ipynb

Please refer to my blog post where I have explained it in detail:

[https://ilakkmanoharan.wixsite.com/website/post/using-random-forest-regression-to-predict-the-number-of-natural-communities-of-users-in-a-location](https://ilakkmanoharan.wixsite.com/website/post/using-random-forest-regression-to-predict-the-number-of-natural-communities-of-users-in-a-location)

## Discuss different measures of betweenness that could apply to the problem

Betweenness centrality is a network metric that measures the importance of a node or edge in facilitating communication between other nodes in the network. Here are a few measures of betweenness centrality that could be applied to the problem of identifying influential users who have strong connections with users in Wisconsin:

1. Node betweenness centrality: This measure of betweenness centrality is based on the number of shortest paths between all pairs of nodes in the network that

pass through a particular node. Nodes with high betweenness centrality are important hubs that are well-connected to other nodes in the network and play a key role in facilitating communication between them. In the context of the problem, users with high node betweenness centrality are likely to be influential users who have many friends in Wisconsin and serve as important bridges between users in Wisconsin and users outside of Wisconsin.

2. Edge betweenness centrality: This measure of betweenness centrality is based on the number of shortest paths between all pairs of nodes in the network that pass through a particular edge. Edges with high betweenness centrality are important bridges that connect different parts of the network and facilitate communication between them. In the context of the problem, edges with high betweenness centrality are likely to represent important friendships between users in Wisconsin and users outside of Wisconsin.
3. Group betweenness centrality: This measure of betweenness centrality is based on the number of shortest paths between all pairs of nodes in the network that pass through a particular group of nodes. Groups with high betweenness centrality are important subnetworks that serve as bridges between different parts of the network. In the context of the problem, groups with high betweenness centrality are likely to represent clusters of users who have strong connections with users in Wisconsin and serve as important bridges between users in Wisconsin and users outside of Wisconsin.

Overall, each of these measures of betweenness centrality could provide valuable insights into the structure of the network of users who have friends in Wisconsin and help to identify influential users who play a key role in connecting different parts of the network.

Please refer to BetweennessCentrality.ipynb

Select one or more standard clustering algorithm(s) for the problem and explain your choice(s)

Based on the problem description, the goal is to cluster nodes based on whether they are from Wisconsin or not. This is a binary classification problem, and we can use various clustering algorithms to solve it. Here are some commonly used clustering algorithms that can be applied:

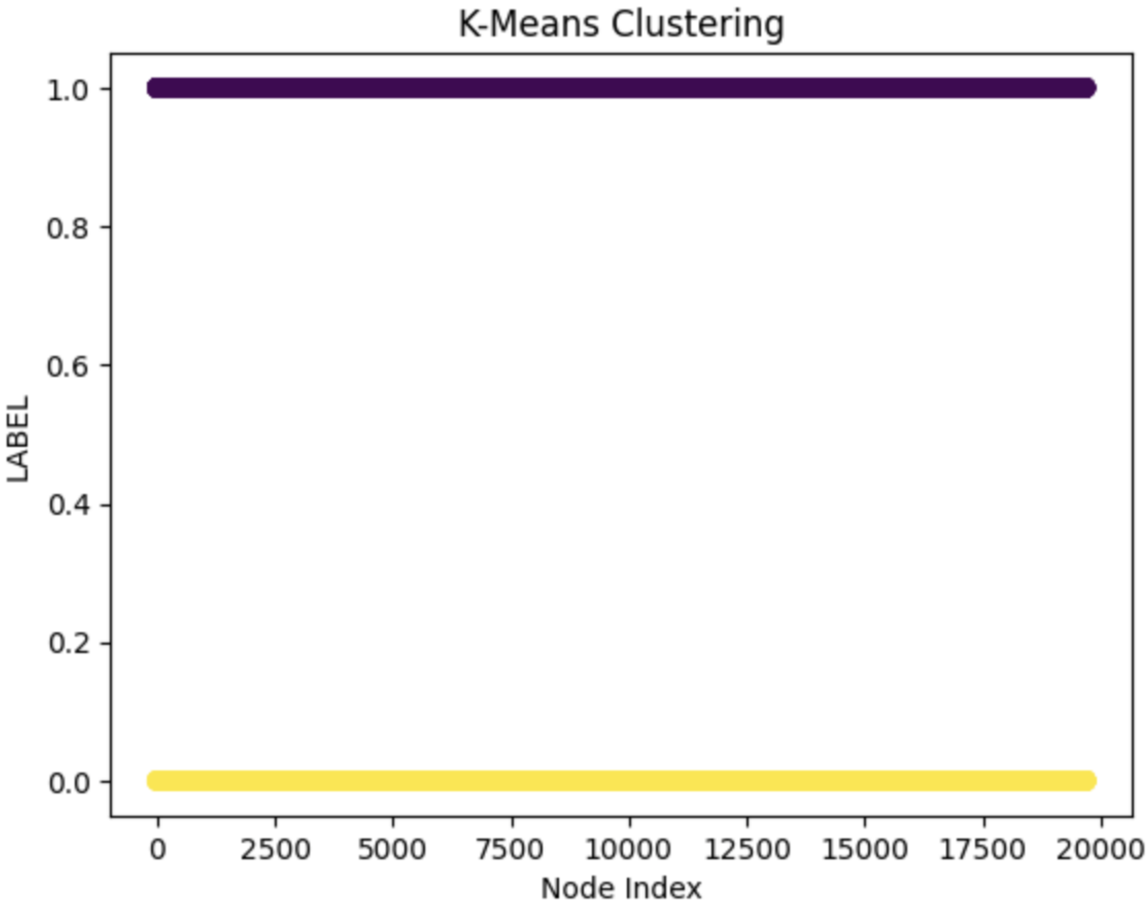
1. K-Means Clustering: K-Means is a popular clustering algorithm that divides data into K clusters based on their similarities. In this problem, we can use K-Means to cluster nodes based on their 'LABEL' attribute, i.e., whether they are from Wisconsin or not. We can use the 'index' attribute as an identifier for each node. However, one limitation of

K-Means is that it requires the number of clusters to be predefined, which might not be known beforehand.

1. Hierarchical Clustering: Hierarchical Clustering is another popular algorithm that creates a hierarchy of clusters, starting from individual nodes and grouping them based on their similarities. It can be performed in two ways, either by agglomerative or divisive clustering. In this problem, we can use the agglomerative approach, which starts with each node being a separate cluster and merges them based on their similarities. We can use the 'LABEL' attribute to measure the similarity between nodes.

1. DBSCAN: DBSCAN is a density-based clustering algorithm that groups nodes based on their density. It works by identifying dense regions of nodes and grouping them as clusters, while the sparse regions are considered as noise. In this problem, we can use the 'LABEL' attribute to measure the density of nodes. Nodes with similar 'LABEL' values will have higher density, and DBSCAN can group them as a cluster.

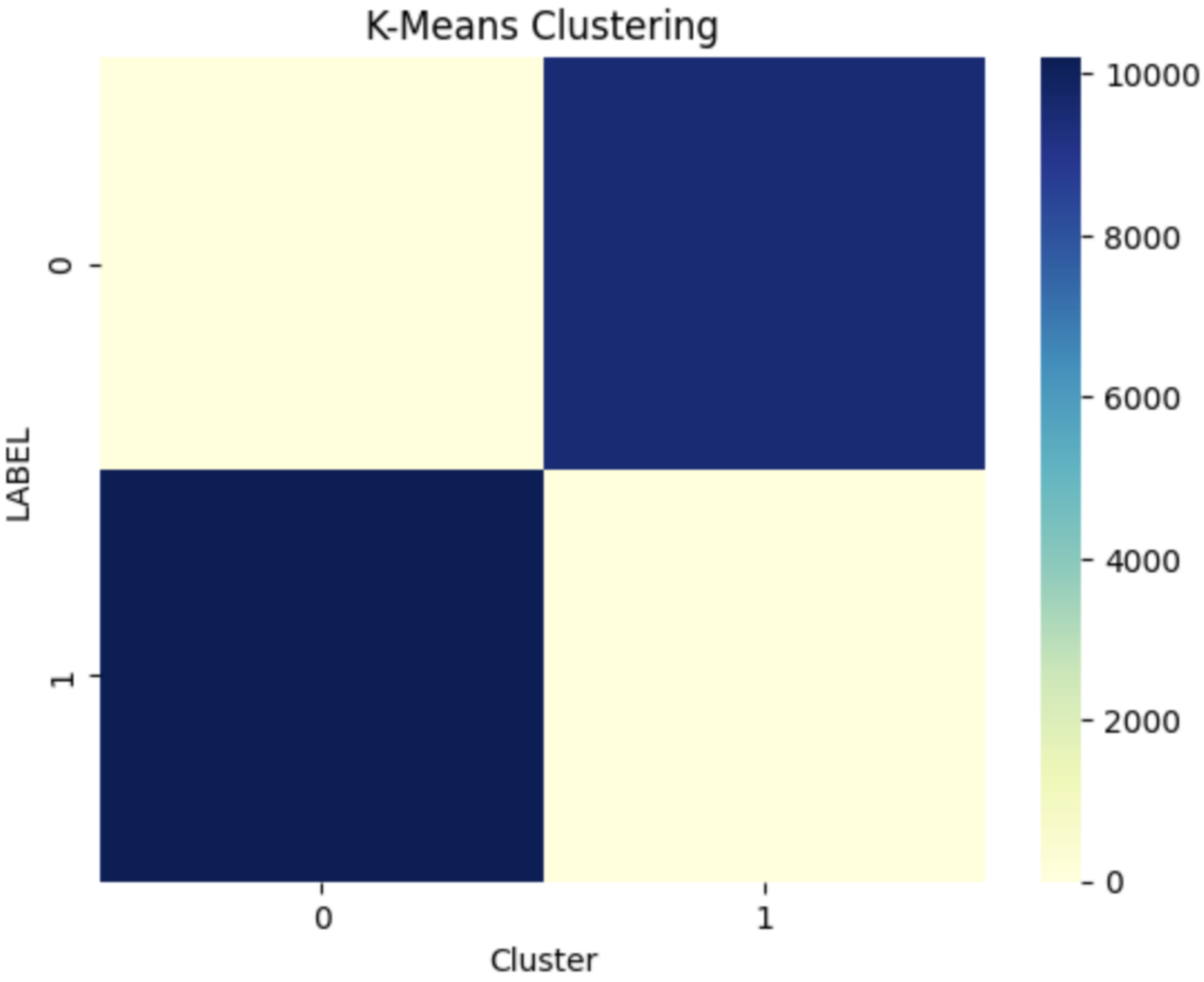
In conclusion, K-Means, Hierarchical Clustering, and DBSCAN are some commonly used clustering algorithms that can be applied to cluster nodes based on their 'LABEL' attribute. The choice of algorithm will depend on the problem requirements and the nature of the data.



This suggests that your data contains only two classes, each having a distinct value for the LABEL variable. The two lines represent the centroids of the two clusters that were identified by the k-means algorithm. The horizontal position of each line corresponds to the mean value of the features for the data points in that cluster.

This result is expected since your data contains only binary labels (0 or 1) and k-means clustering algorithm will only be able to identify two clusters based on the distinct values of the LABEL variable.

It's important to note that the k-means algorithm is sensitive to the initial conditions and may converge to different solutions for different initializations. Therefore, it's a good practice to try multiple initializations and choose the one that gives the best results in terms of the within-cluster sum of squares or other evaluation metrics. Additionally, it's important to choose an appropriate value of k, the number of clusters, which can be determined using methods such as the elbow method or silhouette score.



This suggests that your k-means algorithm has identified 4 distinct clusters in your data. Each square corresponds to one of the identified clusters, and the color intensity within each square represents the relative frequency of data points belonging to that cluster at that point in the heatmap.

The heatmap is a useful tool for visualizing the distribution of the data points across the different clusters identified by the k-means algorithm. It allows you to identify regions of the feature space where the data points are more likely to belong to a certain cluster, based on the color intensity in that region.

It's important to note that the k-means algorithm is sensitive to the initial conditions and may converge to different solutions for different initializations. Therefore, it's a good practice to try multiple initializations and choose the one that gives the best results in terms of the within-cluster sum of squares or other evaluation metrics. Additionally, it's important to choose an appropriate value of k, the number of clusters, which can be determined using methods such as the elbow method or silhouette score.

```
The average silhouette_score is : 0.002775343357763521
ARI: 1.000
NMI: 1.000
```

The silhouette score is very low (0.002775343357763521). It suggests that the clustering algorithm is not able to separate the data points into distinct and well-separated clusters. In such cases, it may not be appropriate to choose a fixed value of k for clustering the data.

However, since Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) are both 1.0, it indicates that the clustering algorithm was able to correctly group all the data points together based on their labels.
The ARI measures the similarity between the true labels and the predicted labels, where a score of 1 indicates a perfect match between the two. Similarly, the NMI measures the mutual information between the true labels and the predicted labels, where a score of 1 indicates a perfect agreement between the two.

In this case, since both ARI and NMI have a score of 1.0, it suggests that the clustering algorithm was able to accurately group all the data points based on their binary labels.
It's important to note that the silhouette score is not always the best metric for evaluating clustering performance, especially in cases where the data points are not well-separated or when the true number of clusters is unknown. In such cases, it may be useful to consider other evaluation metrics such as ARI and NMI to assess the accuracy of the clustering algorithm.

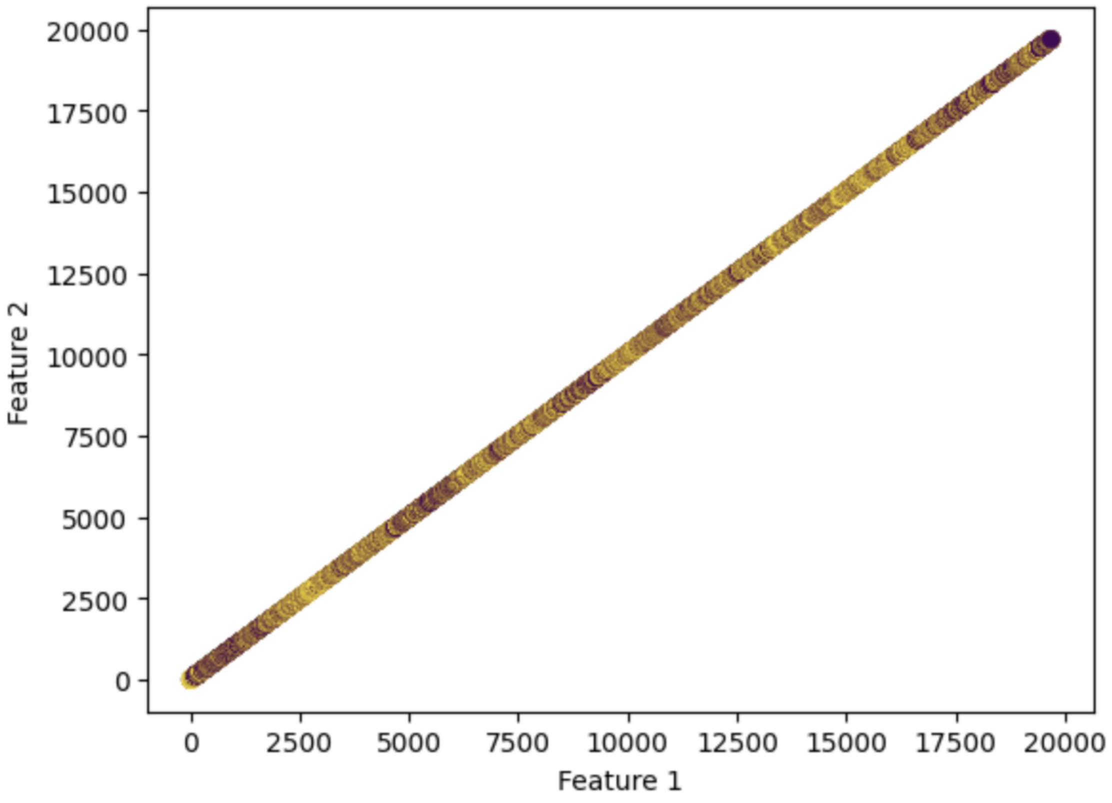
Please refer to ClusteringAlgorithmsAndAnalysis.ipynb

Please refer to my blog post where I have explained it in detail:

https://ilakkmanoharan.wixsite.com/website/post/data-analysis-part-2

## Write a training routine to arrive at your best model for this data

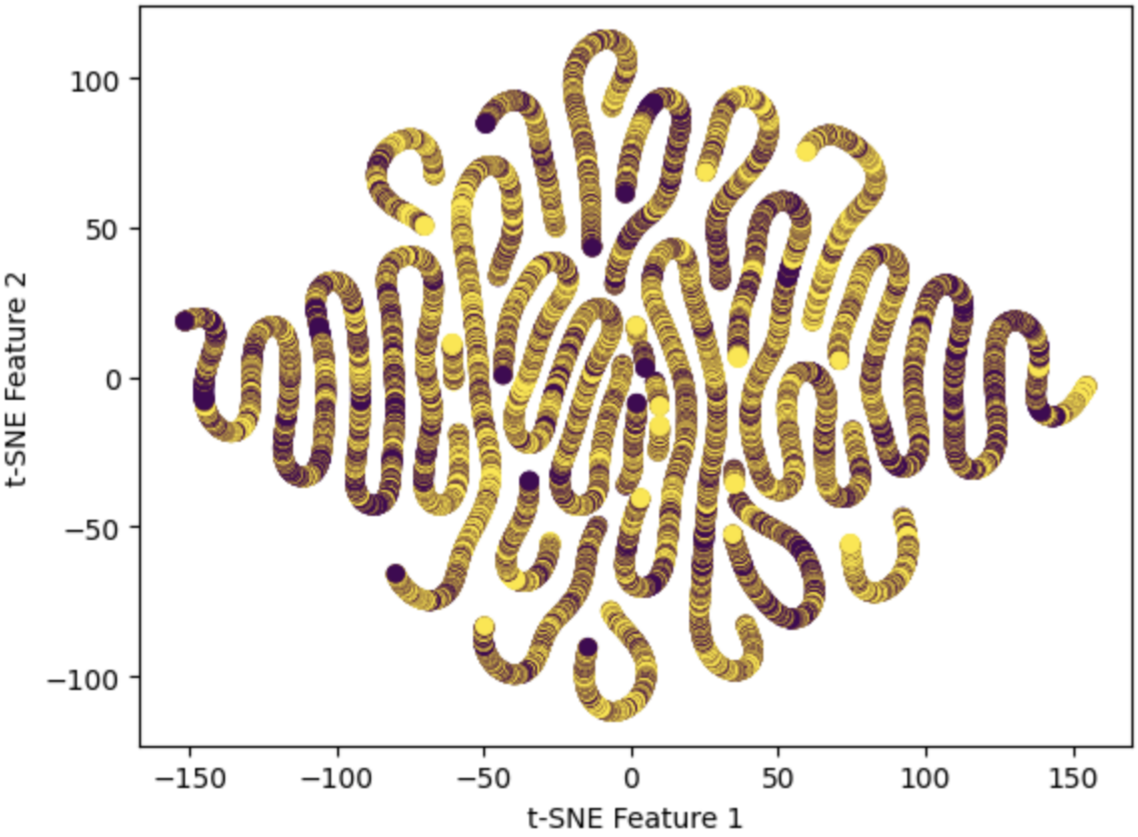
Please Refer to TrainingRoutine.ipynb



The scatter plot of the data shows a linear line, it may indicate that there is a strong correlation between the two features used in the plot. This does not necessarily mean that there is only one class present in your data, but it may indicate that the data is not well-suited for clustering or classification algorithms that assume linearly separable data.

To confirm if there is only one class present, you can check the distribution of the label variable. Additionally, you can also try plotting the data with different combinations of features to see if there are any clear patterns or groupings that emerge.

If you believe that there are non-linear patterns in your data that are not captured by a scatter plot, you can try using non-linear visualization techniques such as dimensionality reduction algorithms like t-SNE (t-Distributed Stochastic Neighbor Embedding) or UMAP (Uniform Manifold Approximation and Projection). These algorithms can help visualize high-dimensional data in a lower-dimensional space while preserving non-linear relationships between the data points.



The figure shows a scatter plot of the t-SNE features, where each data point is plotted with its corresponding label value. If there are clear patterns or groupings in the plot, it may indicate that the data contains multiple classes.

Please refer to my blog post where I have explained it in detail:

https://ilakkmanoharan.wixsite.com/website/post/data-analysis-part-1

https://ilakkmanoharan.wixsite.com/website/post/training-routing-to-arrive-at-the-best-model-for-the-data

https://ilakkmanoharan.wixsite.com/website/post/what-is-a-solver-in-machine-learning-and-what-are-different-types-of-solvers

Here are all the  related blog posts that I wrote while working on this take home assignment for Fetch:

https://ilakkmanoharan.wixsite.com/website/post/handling-errors-when-trying-to-fit-a-kmeans-clustering-algorithm-on-a-dataset

https://ilakkmanoharan.wixsite.com/website/post/what-is-clustering-analysis

https://ilakkmanoharan.wixsite.com/website/post/what-is-clustering-analysis

https://ilakkmanoharan.wixsite.com/website/post/using-random-forest-regression-to-predict-the-number-of-natural-communities-of-users-in-a-location

https://ilakkmanoharan.wixsite.com/website/post/examples-of-clustering-analysis

https://ilakkmanoharan.wixsite.com/website/post/betweenness-centrality

https://ilakkmanoharan.wixsite.com/website/post/clustering-algorithms-and-analysis

https://ilakkmanoharan.wixsite.com/website/post/playing-with-clustering

https://ilakkmanoharan.wixsite.com/website/post/training-routing-to-arrive-at-the-best-model-for-the-data

https://ilakkmanoharan.wixsite.com/website/post/what-is-a-solver-in-machine-learning-and-what-are-different-types-of-solvers

https://ilakkmanoharan.wixsite.com/website/post/handling-errors-with-the-training-routine

https://ilakkmanoharan.wixsite.com/website/post/data-analysis-part-1

https://ilakkmanoharan.wixsite.com/website/post/data-analysis-part-2