COL761: Data Mining
Assignment 3

Team Members:
Supreeti Kumari - 2020CS10396
Eshan Jain - 2020CS50424
Goonjan Saha - 2020CS10494

Contributions:

All team members have contributed equally to this assignment.

Question 1:

Approach - We first pick up 1 million points uniformly from 0 to 1. Then we pick up 100 points from the dataset and update the dataset to remove those 100 points. Then we iterate over the query points and calculate the L1, L2 and L_inf distance. Finally we take the average over the 100 points.

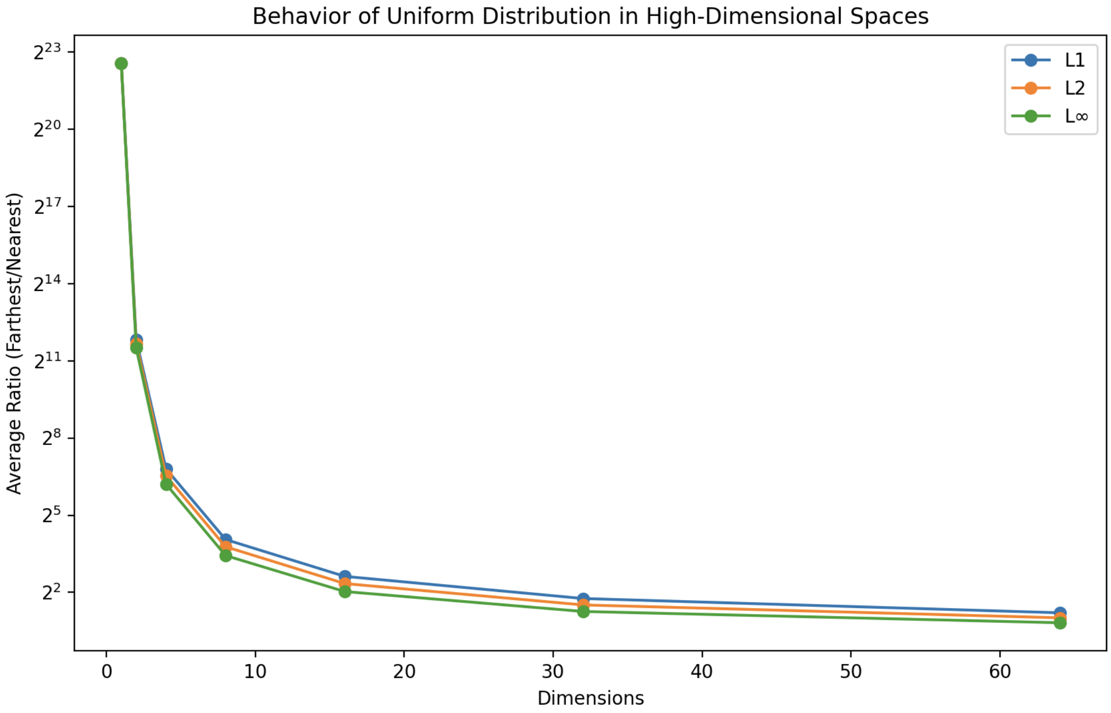
Conclusion-

1. As the dimension is increasing the ratio is decreasing - In high-dimensional spaces, a phenomenon known as the "curse of dimensionality" comes into play. As the number of dimensions increases, the volume of the space grows exponentially, leading to sparsity. In other words, points become more spread out in higher-dimensional spaces.

As the dimensionality increases, the distance between points tends to increase, and the notion of "closeness" becomes less meaningful. This makes it harder for points to be both close and far from a given query point simultaneously. Consequently, the ratio of the farthest and nearest distances tends to decrease as the dimensionality increases.

2. The ratio of L1 > L2 > Linf which is clearly evident through the plot for larger dimensions.
In high-dimensional spaces, the cumulative effect of differences along multiple dimensions influences the farthest distances.
The mitigating effect of the square root operation in L2 results in a smaller ratio compared to L1, and the focus on the maximum absolute difference along any dimension in L∞ tends to result in a decreasing ratio as the dimensionality increases.

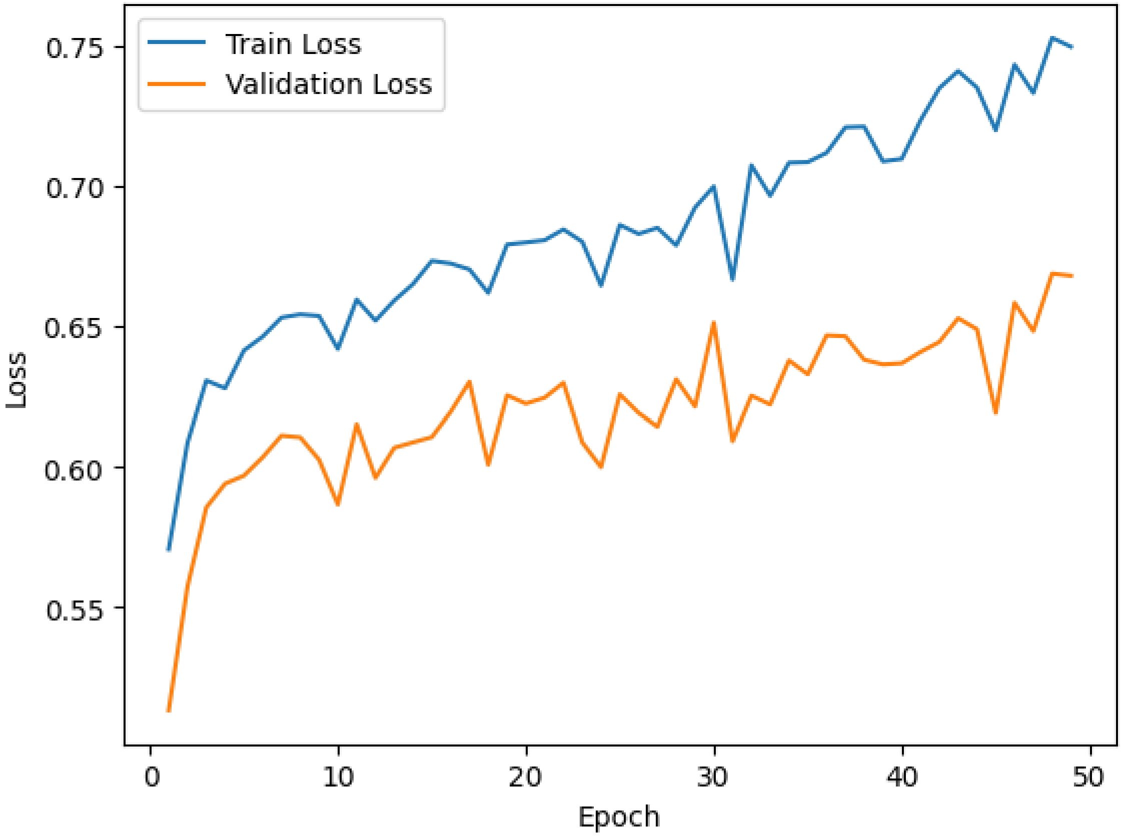
Plot - (Note that we have rounded the final answer as given on piazza. If we round the dataset points, the graph remains similar but the 2^23 value decreases to 2^16)


Behavior of Uniform Distribution in High-Dimensional Spaces
Average Ratio (Farthest/Nearest)
2^{23}
2^{20}
2^{17}
2^{14}
2^{11}
2^{8}
2^{5}
2^{2}
0
10
20
30
40
50
60
Dimensions
L1
L2
L∞

Question 2.

Classification:

Training and Validation Learning Curves -



Firstly we observe that the validation loss is less than the train loss and therefore the model has good generalisabilty. We also faced the problem of vanishing gradients, so we tried various solutions like **gradient clipping** and our final model incorporates them.

Baseline Models:

Random Classifier:

Accuracy: 0.4772
Precision: 0.3020
Recall: 0.4841
F1: 0.3720

ROC-AUC: 0.4790

The Random Classifier serves as a naive baseline where predictions are made randomly without considering any features or structure of the graphs. It assigns labels purely by chance. The performance metrics are close to what would be expected from random chance, indicating that the dataset might be somewhat balanced between the two classes.

Logistic Regression:

Accuracy: 0.7183
Precision: 0.5893
Recall: 0.3929
F1: 0.4714
ROC-AUC: 0.6321

Logistic Regression is a linear model that works by predicting the probability that a given input belongs to a particular class. Its performance is significantly better than random chance. However, its linear nature might limit its ability to capture complex relationships between graph features, leading to relatively moderate performance compared to more sophisticated models like Graph Neural Networks (GNNs).

Explanation:
Complexity of Features: Graphs frequently have relational and structural features that are intricate and may not be linearly separable. For graph data, it's possible that the linear relationship between features and the target that logistic regression assumes isn't true.

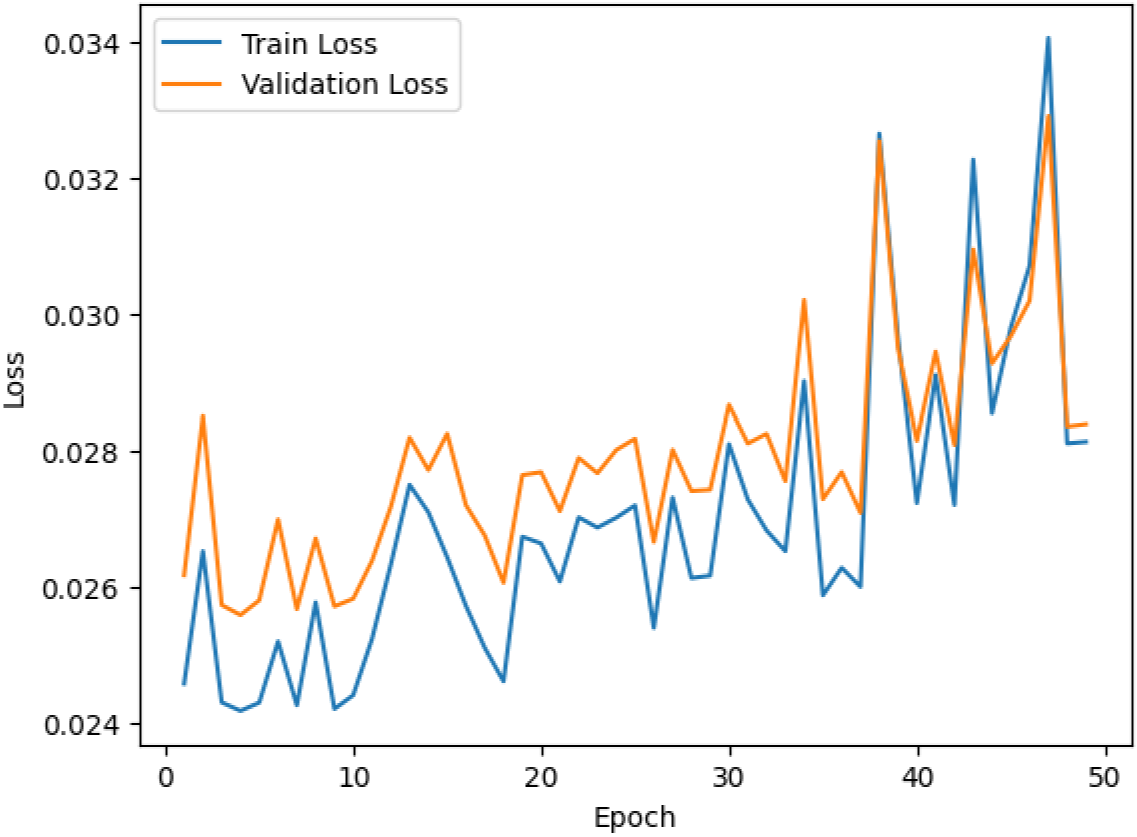
Graph Structure: By combining data from nearby nodes and capturing both local and global graph properties, GNNs are able to efficiently capture the graph structure. Because graph structure is not taken into explicit consideration in logistic regression, important information may be overlooked.

Non-Linearity: Unlike logistic regression, which is essentially linear, GNNs can model non-linear relationships within the graph data because they can carry out multiple iterations of message passing and aggregation.

Expressiveness: GNNs are able to capture more complex patterns found in the data because they are more expressive models and can learn complex representations of nodes and graphs.

Regression:

Training and Validation Learning Curves -



Firstly we observe that the validation loss is higher than the training loss and therefore the model is overfitted.

Baseline Models:

Random Classifier:

Mean Squared Error: 4.7079
R^2 Score: -2.1764

The Random Classifier for regression task likely generates predictions randomly, without considering any features or graph structures. The high MSE and negative R^2 score indicate that its predictions are significantly deviating from the actual values. The negative R^2 score means that this model performs worse than a model that always predicts the mean of the target values. It essentially indicates that this model is not learning anything from the data and performs poorly.

Logistic Regression:

Mean Squared Error: 1.3280
R^2 Score: 0.1040

Similar to the logistic regression used in classification tasks, here, logistic regression is applied for regression purposes. However, logistic regression is inherently designed for binary classification and may not be suitable for regression tasks. Although the MSE is lower than the random classifier, the R^2 score indicates that the model's performance is only marginally better than predicting the mean of the target values. This suggests that logistic regression is insufficient in capturing the complexity of the data for regression purposes.

Explanation:

Model Suitability: Not meant for regression tasks, the Random Classifier produces random predictions. Its incapacity to derive any significant insights from the data is evident in its subpar performance metrics.

Logistic Regression in Regression Task: Because logistic regression predicts probabilities for classification by nature, it may not be the best model to use for regression tasks, even though it works well for binary classification.

Complex Relationships: The intricate relationships present in graph-structured data are not adequately represented by either model. Regression tasks may involve complex relationships between nodes and edges in the graph properties, which these baseline models are not able to learn well.

Graph Structure: The purpose of Graph Neural Networks (GNNs) is to comprehend and make use of the underlying graph data structure. They are more effective at capturing the hierarchical relationships and interdependencies found in graphs, which are essential for forecasting graph properties.