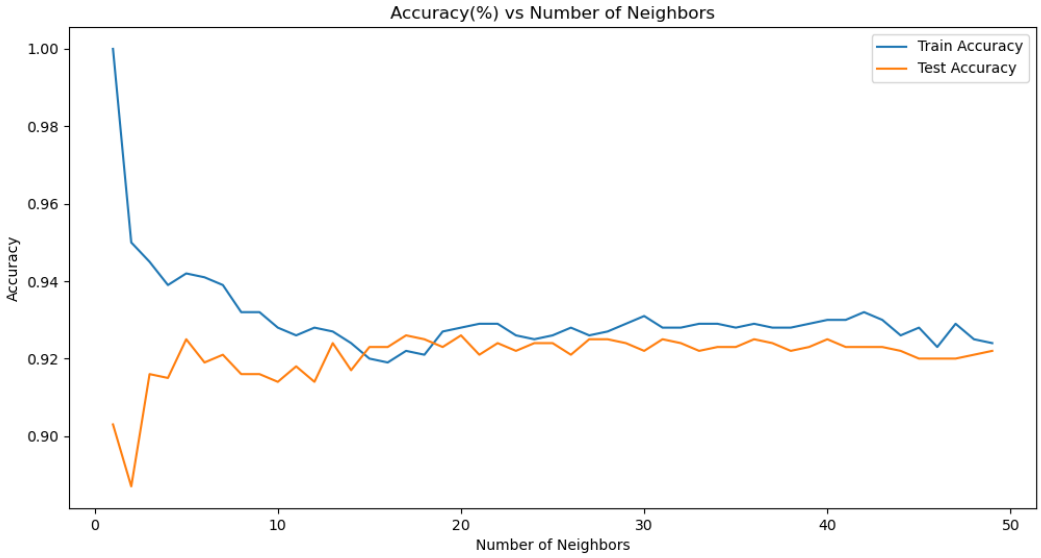
Jinghan Sun HW2

1d.



1e. I would choose k = 17 because it has the highest testing accuracy, indicating the model predicts accurately. The number of neighbors and the train accuracy is moderate to avoid overfitting or underfitting. The training accuracy and testing accuracy are relatively close, suggesting that the model has a good balance between bias and variance, which is typically the goal for a well-fit model.

1f. The prediction function involves calculating distance, sorting distance, selecting neighbors, and counting most common labels. For distance calculation, it calculates the distance to each training sample n across all features d so the time complexity is O(dn). Calculate the Euclidean distance is O(d) for each calculation since it involves subtracting and squaring each of the d features. For sorting distance, the time complexity is O (n log n), when sorting an array of each sample. For selecting k neighbors, the complexity is O(k). For find commonest label, The np.unique function with return counts is used to find the most common label among the k neighbors through voting for all the samples, making this step O(k). To combine, the overall complexity is O(nd+nlogn) after we only consider significant steps.

3e.

Preprocessing	KNN Accuracy	NB Accuracy
No Pre-processing	0.750767	0.746549
Standard_Scale	0.766488	0.738497
MinMax_Scale	0.751917	0.738497
add_irr_feature	0.749233	0.746933

3f. i. I would choose KNN classifier because it has higher accuracy among all preprocessing methods.

ii. For KNN classifier, after using standard scale and minmax scale methods the accuracy score became better compared to no preprocessing method, with the highest accuracy after standard scaling, indicating KNN benefits from all features share the same scale as it measures distance. Compared to standard scaling the minmax scale accuracy improves slightly, around 0.001 from 0.7508 to 0.7519.

For NB classifier, the accuracy drops from 0.7465 to 0.7385 for both standard scaling and minmax scaling. Pre-processing lowers the performance of the classifier in this case.

iii. The KNN classifier shows a slight decrease in accuracy when irrelevant features are added, dropping from 0.7508 to 0.7492. This indicates that while the performance of the KNN classifier is somewhat sensitive to noise, the impact is small though, suggesting a moderate level of robustness to irrelevant features. The Naive Bayes classifier shows a small increase in accuracy after adding the irrelevant feature, which suggests that the NB classifier may be more robust to noise.