Classification using Traditional Machine Learning Methods

Sawyer Bowerman

Prof. Tales Imbiriba

10/15/2024

1. Introduction

This project focuses on the different applications of commonly used machine learning techniques to classify data of varying complexities in dimension and size. Classification is highly important in the machine learning field, as there are very many problems that need solutions by means of image classification. Imagine a fully automated veterinary system for animals that requires proper medication doses to be provided to animals based on their size and species. A proper classification system would be able to determine the type of animal and provide a proper dosage to the pet, possibly even in a more accurate way than a human may. This is just one example, but the importance is apparent and everywhere in the field of computer vision.

We are expected to perform classification on a synthetic dataset that we create and plot those results. In the next problem, we are expected to do the same but with scikit-learn's digits dataset. I used a technique called stratified cross-validation for hyperparameter tuning for questions 2 and 3. Thanks, geeks4geeks. I still have a lot to learn about the topic but hopefully I'll be able to do more research at some point. For question 3, we do the same but with the CIFAR-10 dataset which should take a lot longer to process since it's a whole lot larger in size.

2. Implementation

2.1 Classifiers Used and Their Properties

To Preface:

A parametric method is an approach that uses a set of assumptions about the function in order to relate the input to the output. These methods have a fixed number of parameters, regardless of the size of the training set.

- **Nearest Neighbors: KNeighborsClassifier from sklearn**
  - Non-parametric method
  - Makes predictions based on the k closest training examples in the feature space
  - No assumptions about the underlying data distribution
  - Key hyperparameter: k (number of neighbors)

- **Linear Discriminant Analysis: LinearDiscriminantAnalysis from sklearn**
  - Parametric method
  - Assumes classes are normally distributed with equal covariance matrices
  - Finds a linear combination of features that best separates two or more classes
  - Can be used for dimensionality reduction

- **Quadratic Discriminant Analysis: QuadraticDiscriminantAnalysis from sklearn**
  - Parametric method
  - Similar to LDA but allows for different covariance matrices for each class
  - Can model non-linear decision boundaries
  - More flexible than LDA but requires more parameters to be estimated

- **Logistic Regression: LogisticRegression from sklearn**
  - Parametric method
  - Models the probability of an instance belonging to a particular class
  - Uses the logistic function to squash output to a range between 0 and 1
  - Can be extended to multi-class problems using techniques like one-vs-rest

```
classifiers = {
    'Nearest Neighbors': KNeighborsClassifier(),
    'Linear Discriminant Analysis': LinearDiscriminantAnalysis(),
    'Quadratic Discriminant Analysis': QuadraticDiscriminantAnalysis(),
    'Logistic Regression': LogisticRegression(random_state=42, max_iter=1000)
```

```
}
```

I organized the classifiers in a dictionary titled classifiers just to make it easier to manage. This holds true for all three problems.

2.2 Datasets

- Problem 1: Synthetic 2-class, 2-dimensional dataset

- Problem 2: Scikit-learn digits dataset

- Problem 3: CIFAR-10 dataset

3. Results

To Preface:

We can use sklearn's built in accuracy score and confusion matrix methods to help use determine the metrics and matrices.

```
from sklearn.metrics import accuracy_score, confusion_matrix
```

3.1 Problem 1: Synthetic Dataset

- Accuracy metrics for each classifier

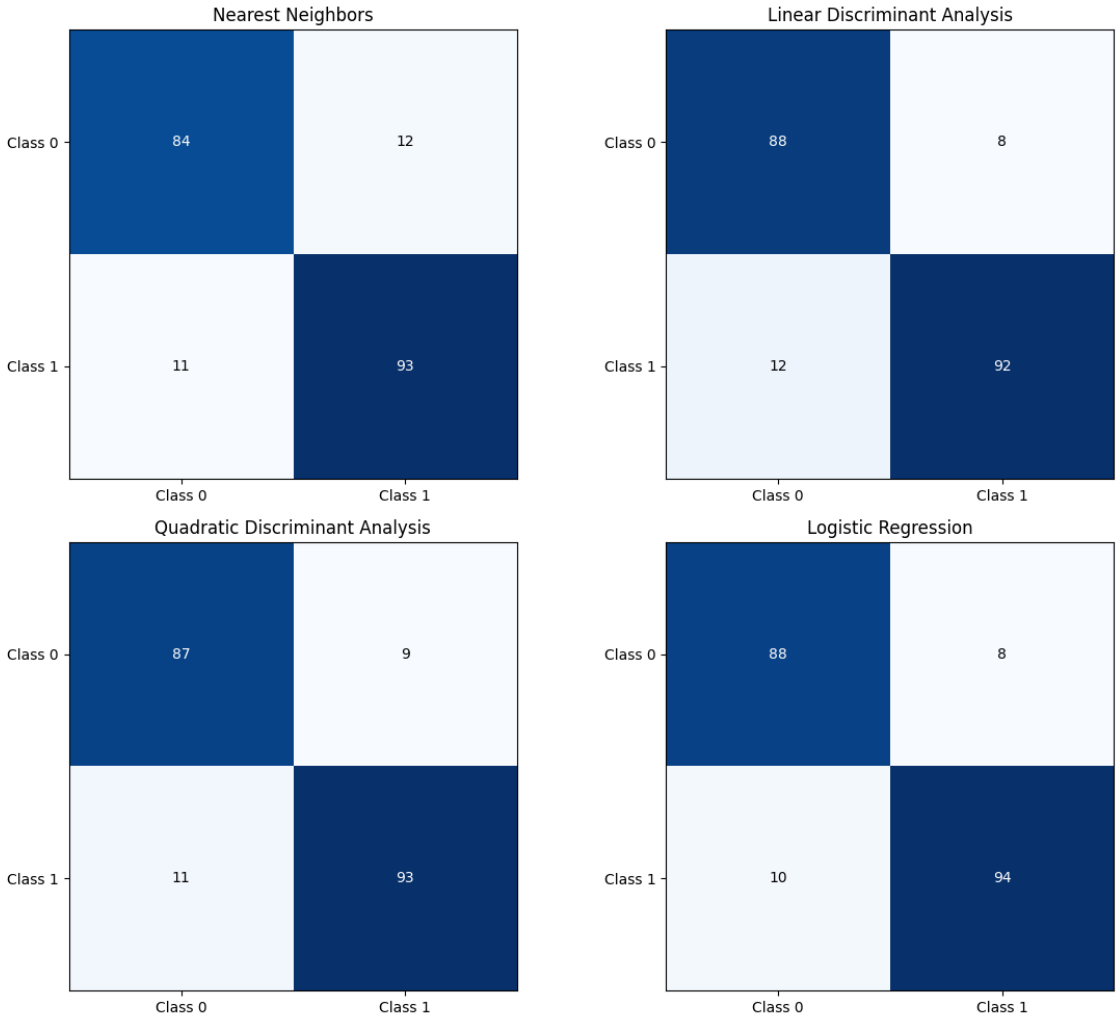
Nearest Neighbors: Accuracy: 0.9200

Linear Discriminant Analysis: Accuracy: 0.9150

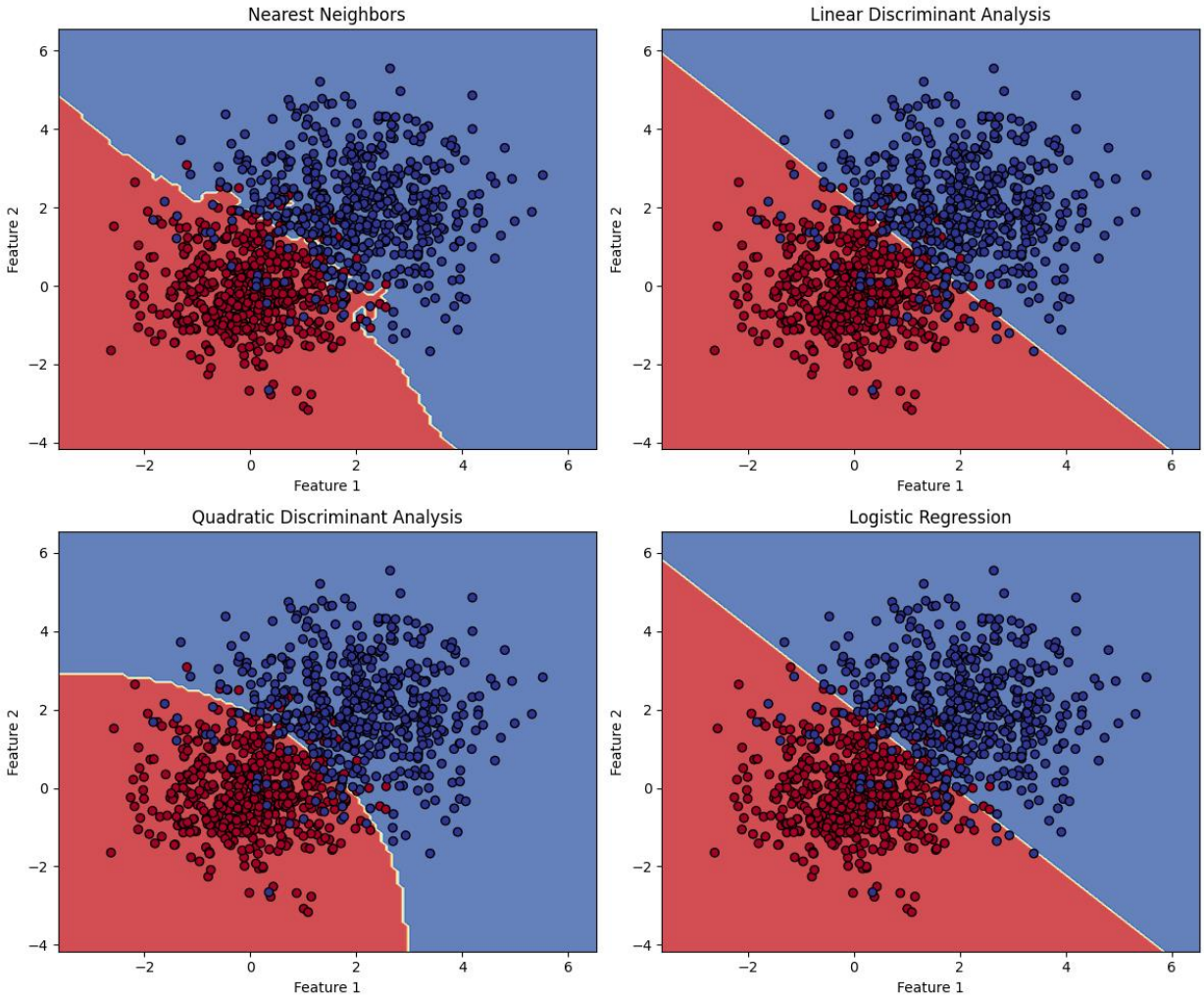
Quadratic Discriminant Analysis: Accuracy: 0.9050

Logistic Regression: Accuracy: 0.9250

- Confusion matrices



- Decision boundary plots



- Brief discussion of results

In this pass of problem 1, all classifiers perform well, with linear methods having a slight edge. This suggests that when creating the synthetic data, the classes were generated with a predominantly linear separation, but with some overlap to make the classification task non-trivial.

This was expected though, as I created a primarily linearly separable dataset in my code.

Here is my function to generate the dataset.

```
def generate_dataset(n_samples=1000):
    n_samples_per_class = n_samples // 2

    # Class 0
    mean1 = [0, 0]
    cov1 = [[1, 0], [0, 1]]
    x0 = np.random.multivariate_normal(mean1, cov1, n_samples_per_class)
```

```
    # Class 1
    mean2 = [2, 2]
    cov2 = [[1.5, 0], [0, 1.5]]
    x1 = np.random.multivariate_normal(mean2, cov2, n_samples_per_class)

    X = np.vstack((x0, x1))
    y = np.hstack((np.zeros(n_samples_per_class), np.ones(n_samples_per_class)))

    return X, y
```

As the mean from Class 0 to Class 1 are diagonally separated, this will lead to linear methods performing better. However, because the covariance of Class 1 is slightly larger than Class 0, there will be some overlap, the problem is non-trivial.

3.2 Problem 2: Digits Dataset

- Accuracy metrics for each classifier

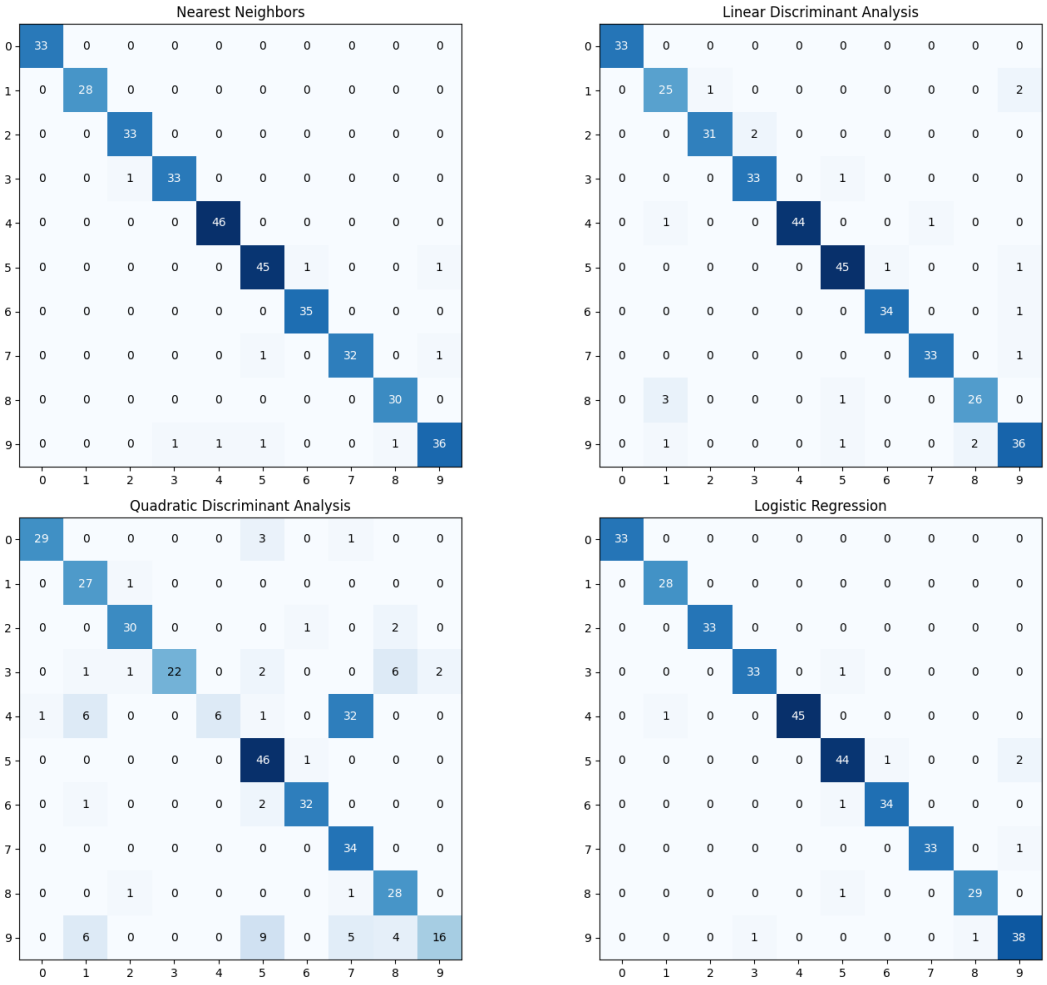
Nearest Neighbors: Accuracy: 0.9750

Linear Discriminant Analysis: Accuracy: 0.9444

Quadratic Discriminant Analysis: Accuracy: 0.7500

Logistic Regression: Accuracy: 0.9722

- Confusion matrices



- Brief discussion of results

Nearest Neighbors performed well, likely due to its ability to capture local patterns in the digit images. Linear Discriminant Analysis (LDA) showed good performance, suggesting that the digits are somewhat linearly separable in the feature space. Quadratic Discriminant Analysis (QDA) performed similarly to LDA, indicating that the added complexity of different covariance matrices for each class didn't provide significant benefits. Logistic Regression also performed well, which is expected for a relatively simple dataset like handwritten digits.

3.3 Problem 3: CIFAR-10 Dataset

- Accuracy metrics for each classifier

Nearest Neighbors: Accuracy: 0.3397

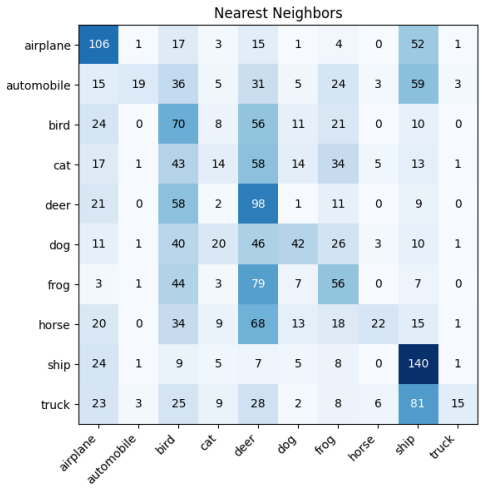
Linear Discriminant Analysis: Accuracy: 0.3712

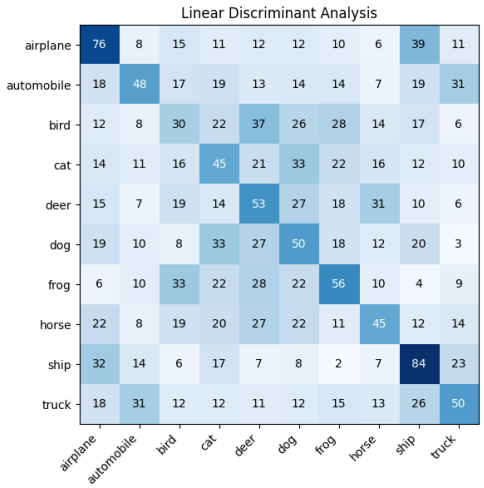
Quadratic Discriminant Analysis: Accuracy: 0.3622

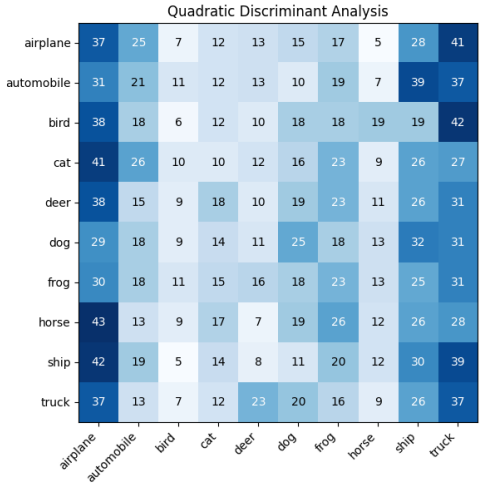
Logistic Regression: Accuracy: 0.3674

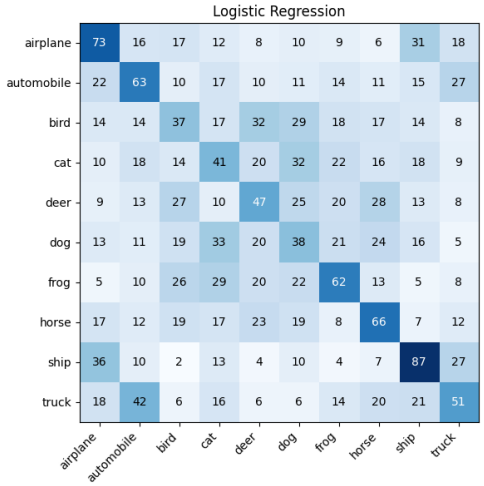
- Confusion matrices

I first ran a test with only 10000 samples because I wanted to make sure that my code was working, so here's that first.

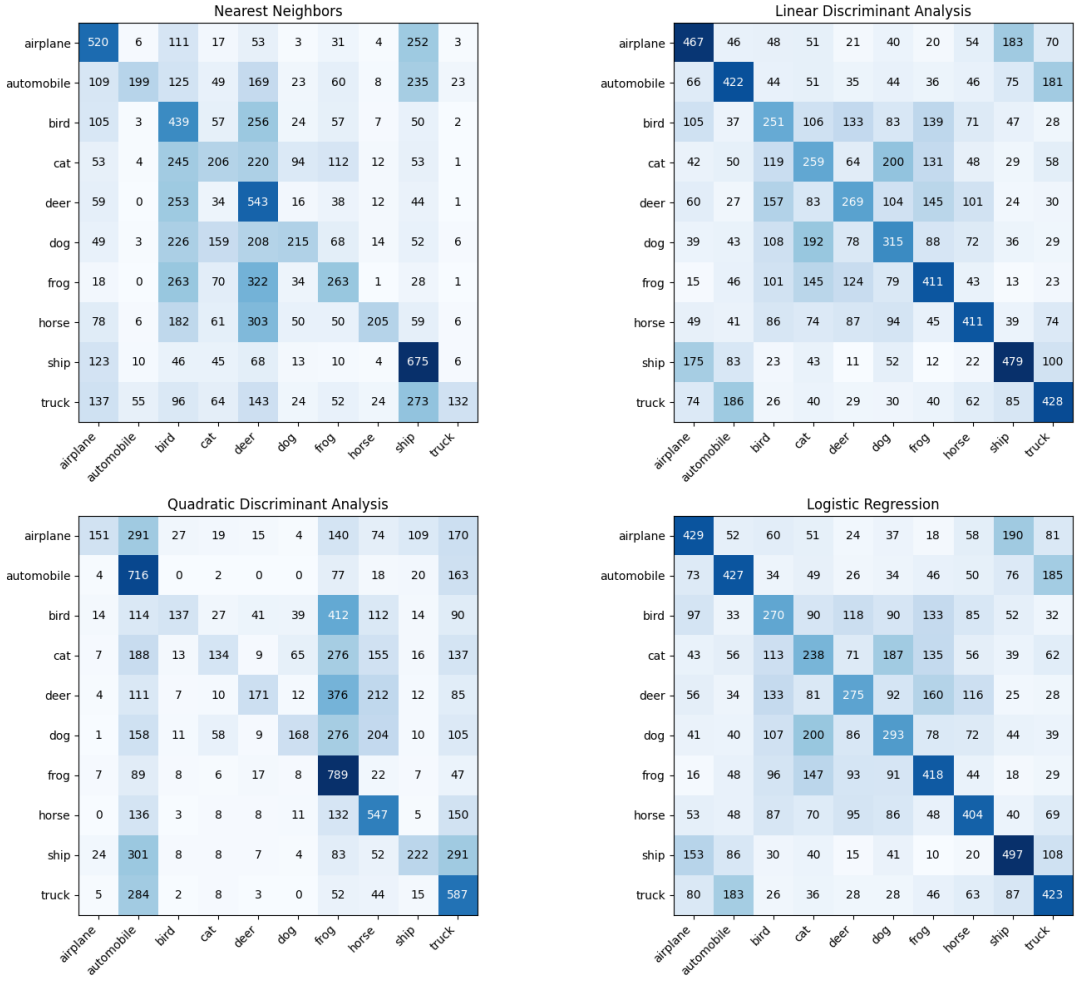








And here is the calculation on the full CIFAR-10 dataset in which the accuracy score corresponds to.



- Brief discussion of results

I was initially a bit concerned about the accuracy of my methods utilizing the CIFAR-10 dataset but once I realized that with 60000 samples, a low accuracy is to be expected, it wasn't too bad. Luckily I noticed this before I redid anything! Overall performance was lower compared to the digits dataset, which is expected given the Nearest Neighbors likely struggled with the high dimensionality of the data. LDA and QDA may have had difficulty capturing the complex relationships in natural images. Logistic Regression might have performed poorly due to the non-linear nature of the image classification task.

4. Discussion

4.1 Nearest Neighbors

- Advantages: Non-parametric, can capture complex decision boundaries, works well for smaller datasets.
- Limitations: Sensitive to the curse of dimensionality, computationally expensive for large datasets, requires careful tuning of k.

4.2 Linear Discriminant Analysis

- Advantages: Works well for linearly separable classes, performs dimensionality reduction, computationally efficient.
- Limitations: Assumes normal distribution with equal covariance matrices, may struggle with complex, non-linear relationships.

4.3 Quadratic Discriminant Analysis

- Advantages: Can capture more complex decision boundaries than LDA, allows for different covariance matrices per class.
- Limitations: Requires more data to estimate parameters accurately, may overfit on smaller datasets

4.4 Logistic Regression

- Advantages: Works well for linearly separable data, provides probabilistic outputs, less prone to overfitting compared to QDA.
- Limitations: May struggle with highly non-linear relationships, assumes linear decision boundaries.

5. Conclusion

Performance varied significantly across datasets, with all classifiers performing best on the synthetic data and worst on CIFAR-10. This highlights the increasing difficulty of classification tasks from simple data to complex image datasets. Nearest Neighbors performed consistently well across all datasets, showcasing its versatility. However, its performance might degrade with very high-dimensional data like CIFAR-10 due to the increased dimensions. LDA and QDA performed similarly in many cases. They worked well on the simpler datasets but also struggled with the complexity of CIFAR-10. Logistic Regression showed good performance on simpler datasets but likely struggled with the non-linear nature of complex image classification in CIFAR-10.

I would like to think that the choice of classifier depends heavily on the nature of the dataset. For simpler, more linearly separable data, methods like LDA or Logistic Regression may be sufficient. Maybe for more complicated data, neural networks would be necessary, which I hope is the case because I've been wanting to learn about them for years.

I got some help and information on how to plot this data from online, as I was a bit confused about how to get it all working correctly! All good now though.