Comparing ML Algorithms

2024062806, Hajin Ju

Abstract

This study compares the performance of various machine learning algorithms on the Digit dataset using sklearn library. Experimental results demonstrate that non-linear models, specifically SVM (with rbf) and the Voting Classifier, achieved superior accuracy (over 98%), significantly outperforming linear baselines and Gaussian Naive Bayes. We conclude that model capacity regarding non-linearity is critical for high-dimensional image classification and highlights a distinct trade-off between accuracy and computational efficiency as a key criterion for model selection.

1. Problem Setting

1. Problem Definition

The primary objective of the assignment is to compare the performance of various machine learning algorithms using sklearn library. The problem is defined as a multi-class-classification.

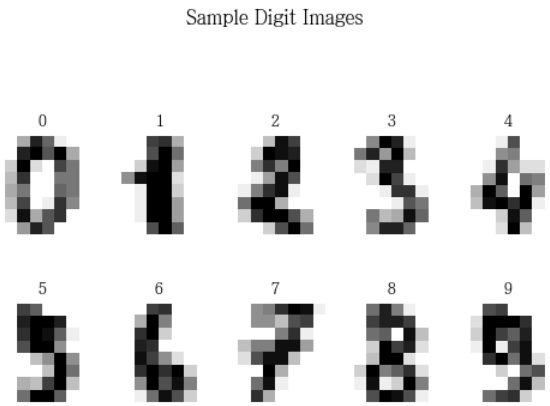
In the problem, I want to find the $f^*: X \rightarrow Y$ (X is a set of all possible data, Y is a set of all class) that generalizes well to unseen data.

2. Dataset Analysis

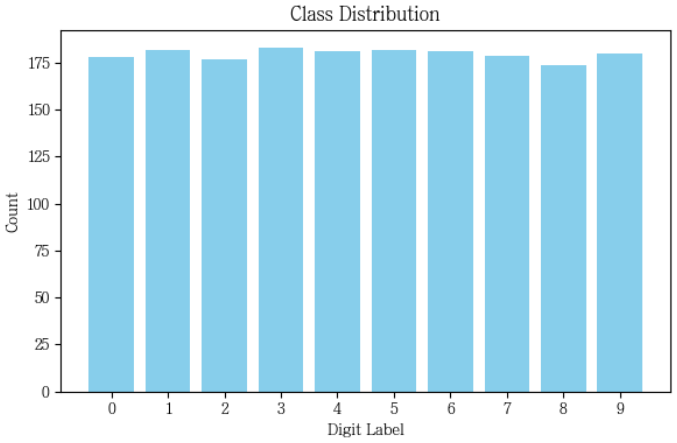
Prior to conducting training model, this section aims to identify the characteristics, features and properties of the dataset.

① Basic information of dataset

- Total number of samples: $N = 1797$
- Feature count: $d = 64$
- Class count: $k = 10$, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- Feature Range: $[0.0, 16.0]$ (actually integer type but represented as float)



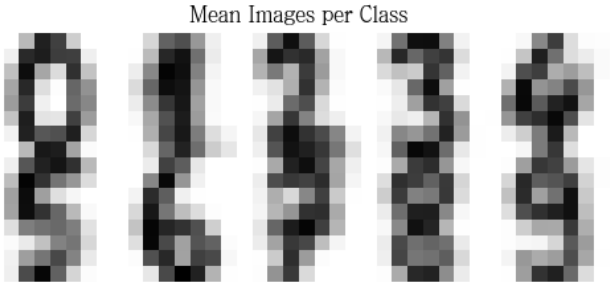
② Class distribution



The right figure illustrates the distribution of samples across the 10 distinct classes (digits 0–9). As depicted in the bar chart, the dataset exhibits a highly uniform distribution, with each class containing approximately 180 samples.

③ Mean Images of each class

To understand structural characteristics of the dataset, we computed and visualized the mean image (prototype) for each class.



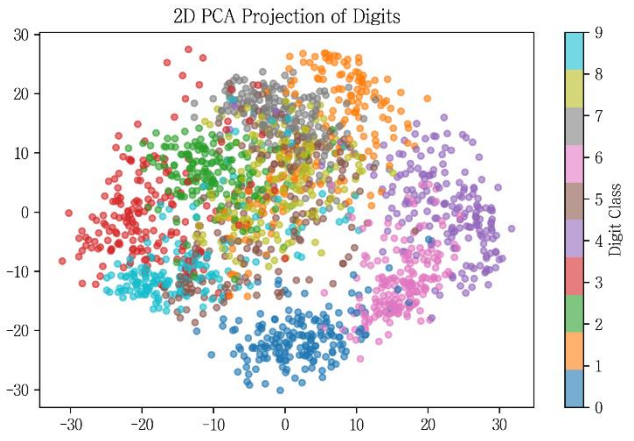
As observed in the visualization, high-intensity regions represent the core strokes that are consistent across samples while blurred regions indicate areas of high intra-class variance where handwriting styles differ significantly.

As the images show, images of class 0 and 6 are clear but mean images of class 8 and 7 show more diffuse patterns.

④ PCA projection

To obtain intuitive insights into the distribution of the 64-dims feature vectors, Principal Component Analysis (PCA) for dimensionality reduction.

PCA is a linear transform that projects high-dimensional data onto a lower-dimensional subspace defined by orthogonal axes (principal components) that maximize the variance of the data. In the analysis, the dataset is projected onto the first two principal components, allowing for the visualization of distribution of the data structure in a 2D plane.



The given plot illustrates the spatial distribution of the dataset. And this plot yields some observations:

Data are not randomly distributed but form distinct clusters. It means that the pixel intensities contain discriminative features.

However significant overlaps are also observed. These overlaps indicate that these classes share similarities, making them potential sources of misclassification. And this fact is implies that the dataset is not perfectly linearly separable in lower dimensions. Consequently, models capturing non-linear relationships are expected to reach high-performance than simple linear classifier models.

3. Dataset Preprocessing

To evaluate the performance of the models, the dataset should be split into training and test set with a ratio of '8:2' using `train_test_split` with a fixed random state 8 was used.

After splitting the dataset, Standardization was applied to the feature vectors to minimize the impact of varying feature scales. By using `StandardScalar` of sklearn, the data was transformed to have a mean of 0 and a standard devication of 1. Standard scalar transforms the dataset ($X \rightarrow Z$) by applying

$$Z = \frac{X - \mu}{s}$$

At this point, parameter μ and s must be derived from the training set. Using the entire set to compute these parameters would result in data leakage, which means the model indirectly leans the statistical distribution of the test set, violating the assumption that the test data.

This step is crucial for algorithms such as Support Vector Machines (SVM) and Logistic Regression, as it improves convergence speed and prevents features with larger magnitudes from dominating the objective function.

2. Models Explanation and Training Models

In this experiment we train:

- Logistic Regression
- SVM (Support Vector Machines)
- FDA (Fisher Discriminant Analysis)
- MLP (Multi-Layer Perceptron)
- Gaussian NB
- Decision Tree
- Random Forest
- Voting (Hard) with KNN, SVM, MLP

1. Model Explaination

① Logistic Regression

Logistic Regression is a linear model for classification. It models the probabilities describing the possible outcomes of a single trial using a logistic function.

It uses Softmax function to estimate the probability that an input vector x belongs to class k

$$P(y = k|x) = \text{softmax}(z)_k = \frac{e^{z_k}}{\sum_{j=1}^{k} z_j} \quad \text{where } z_k = w_k^T x + b_k$$

The model parameters are estimated by minimizing the Cross-Entropy Loss via optimization algorithms such as L-BFGS or SAGA.

In this experiment, default value, L-BFGS, is used.

② Support Vector Machine

SVM is a discriminative classifier that aims to find the optimal hyperplane which maximizes the margin between classes. The margin is defined as the distance between the decision boundary and the nearest data points called Support Vectors. By maximizing the margin, SVM can find robust generalization performance on further data.

Kernel Trick can be introduced to SVM to effectively handle non-linearly separatable data. It works by implicitly mapping the input vectors into a higher-dimensional feature space where a linear hyperplane can be

constructed to separate the classes.

In the sklearn, default kernel is rbf.

③ Fisher Discriminant Analysis

FDA is a supervised algorithm used for both dimensionality reduction and classification.

The core principle of FDA is to find a linear combination of features that maximizes the ratio of between-class scatter (S_B) to within-class scatter (S_W). That is FDA wants to find a vector w to maximize:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

It ensures that samples within the same class are tightly clustered, while centers of the different classes are pushed as far apart as possible. In the sklearn, FDA is implemented as `LinearDiscriminantAnalysis`.

④ MLP

The Multilayer Perceptron is a Neural Network consisting of an input layer, a hidden layer and an output layer. Unlike linear models, MLP can handle non-linear function by utilizing non-linear activation functions in its hiddens.

The network is trained via backpropagation, where the gradients of the loss function are progated backward to update the weights.

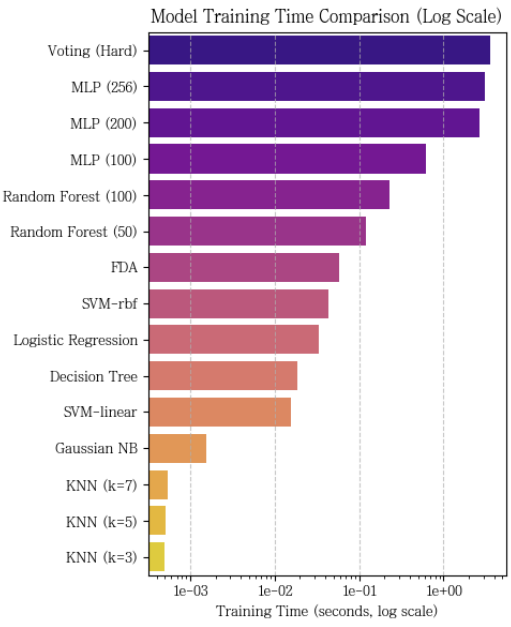
In this experiment, MLP uses ReLU in the hiddens, and Softmax in output.

⑤ Gaussian NB

Gaussian Naïve Bayes is a probabilistic classifier based on Bayes' Theorem with a strong independence assumption between features. It classifies samples by maximizing the posterior probability.

This algorithm assumes that the likelihood of the features (pixel intensities) follows a Gaussian.

$$P(x_i|y) \sim N(\mu_y, \sigma_y^2)$$



⑥ Decision Tree

Decision Tree is a non-parametric model that recursively splits data into branches based on feature to maximize information gain.

⑦ Random Forest

Random Forest is an ensemble learning that constructs multiple decision trees using bagging and random feature selection. And by aggregating the predictions of each tree via majority voting.

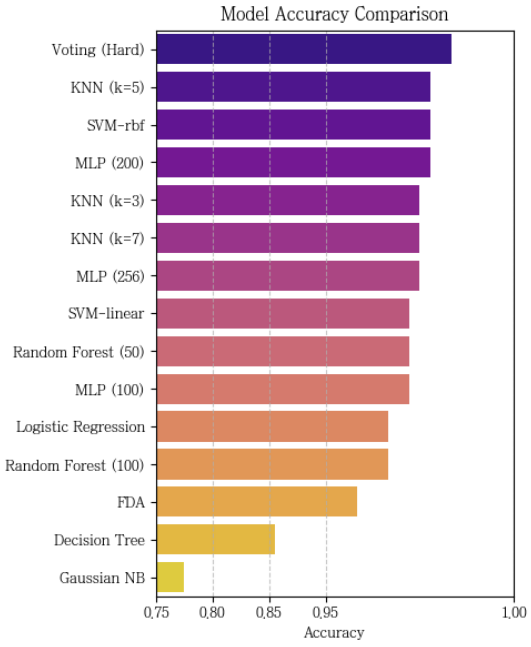
⑧ Voting (Hard)

Voting Classifier is an ensemble method that aggregates predictions from different classifier models. In this experiment, Hard Voting (Majority Voting) is employed and combines three models (KNN, SVM, MLP).

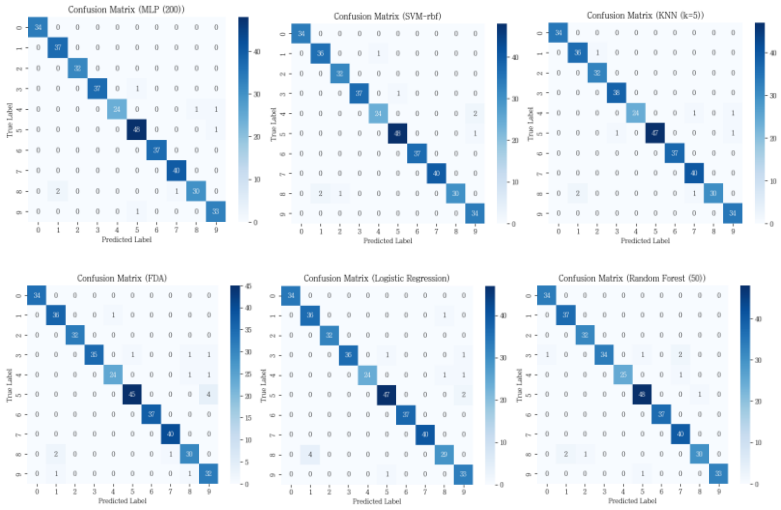
2. Training

Since all the above algorithms are all in sklearn, they can be easily learned through `model.fit`. The learning time was also recorded, but KNN recorded very little learning time while MLP showed a very large learning time.

3. Results



Confusion Matrix of Some Model



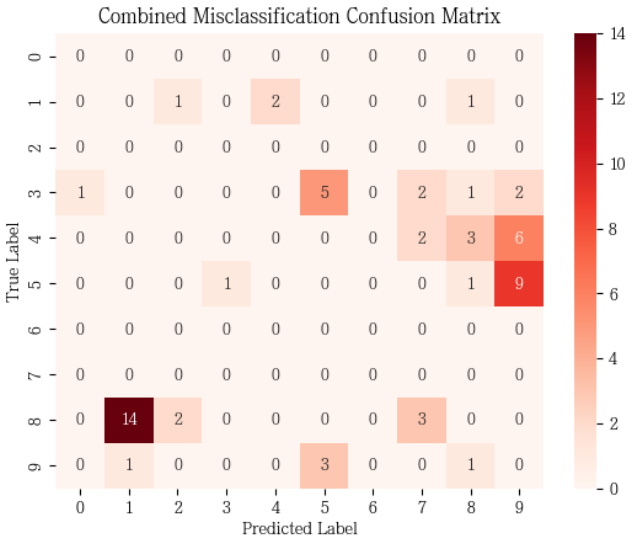
4. Discussion & Conclusion

Nonlinear models likes KNN, SVM-rbf achieve high accuracy. KNN is the most unusual of all; it shows lowest training time. However, there is a fatal trap in this data. There is a problem that the dataset is very small. The problem of KNN is that it can be more inefficient if there is enough data.

Hard voting clasifier performs the best as well as three good models combined, but it can be the worst efficient.

Gaussian NB performs the worst performance. It shows that the data is complex.

On the right is the Combined Misclassification Matrix for the above six models. As the right shows, the model tends to confuse 9 and 5 or 4, and 1 and 8.



Therefore, MLP is more inefficient than I thought, and in fact, in this case, I think KNN is efficient and SVM is more efficient if we have to deal with a larger amount of data. Other algorithms seem to be ineffective for this problem.