Outline

1. Introduction
2. Process and Goal
3. Data Description
4. Data Preprocessing
5. Data Analysis and visualization
6. PCA
7. Classification models

Introduction

Aim: To identify digits from tens and thousands of handwritten images.

Kaggle is one of the leading platform which has competitive competitions based on Machine Learning.

Problem: To compute the handwritten digits from a set of pixel image dataset.

Application: This project can be applied for Courier services, Banks and retail shops.

Process & Goal

Popular topic in the field of computer vision and deep learning; Correctly identify handwritten digit in dataset.

- Two challenges:
  1. Find ways to preprocess data so that they can be efficiently used in classification;
  2. Improve our models by tuning parameters in order to produce better performance;
- Goals:
  1. Build and find the best model: KNN, Decision Tree, Random Forest, Adaboost, Neural Networks, SVM;
  2. Explore the best preprocessing steps: PCA;
  3. Apply models and validation methods we learnt from this course: k-fold cross validation, evaluation matrix;

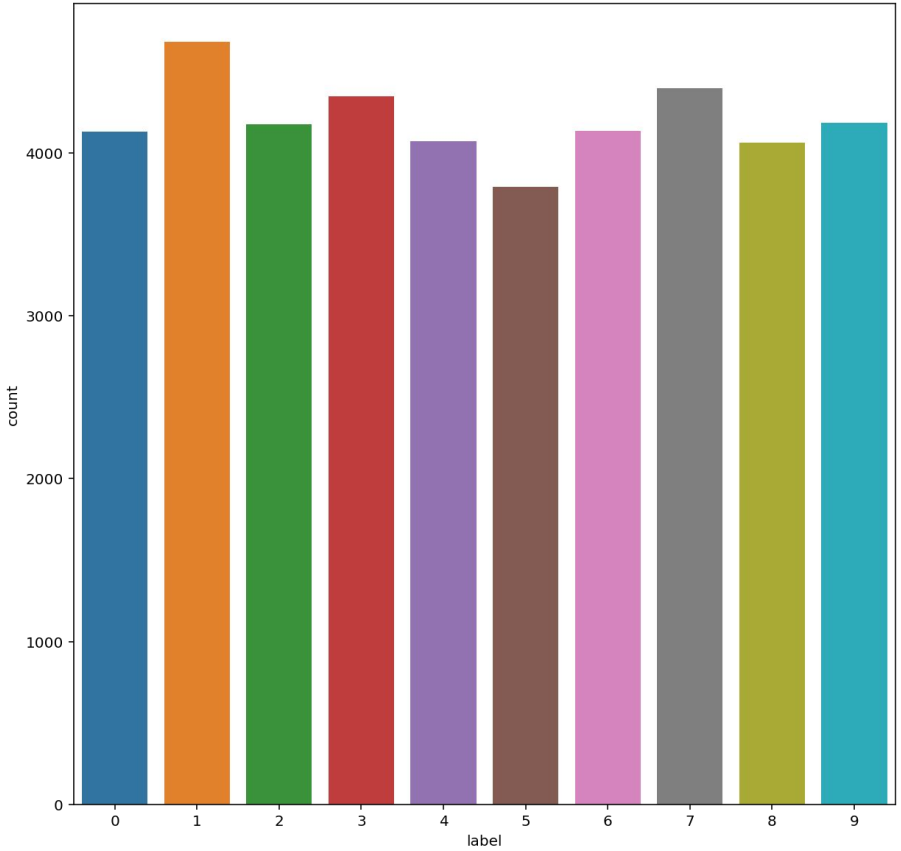
Data Description

- Data file:
  1. train - 785 columns/attributes with a label and remaining pixel values.
  2. test - 784 variable all containing pixel values.
- Contain gray scale images of handwritten digits from 0-9
- image pixel 28x28 = 784 pixels in total.
- pixel value(brightness) = 0-255 [higher value indicates more darkness]
- locate a pixel - i row and j column in 28x28 matrix (indexing by 0)
- example pixel31 is in 4th column 2nd row.

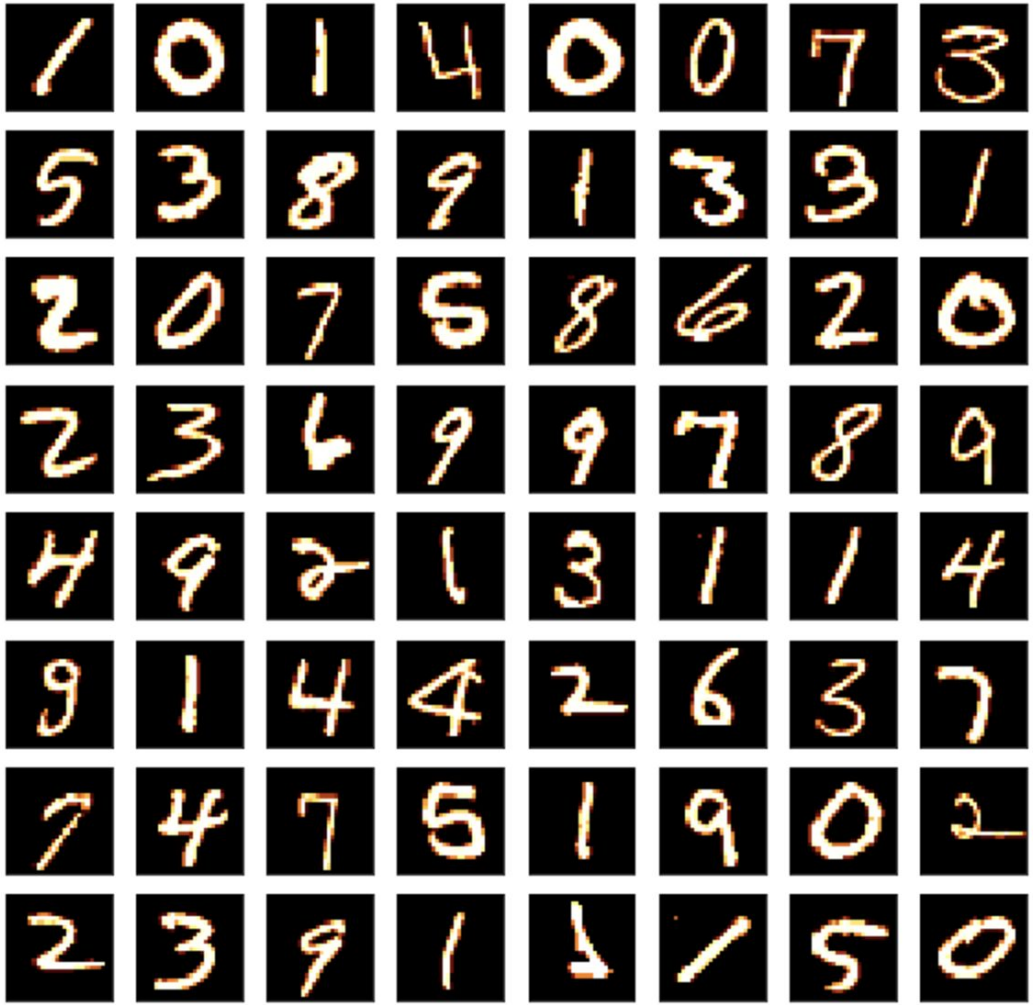
Data Pre-Processing

1. Important step in Data Analysis
2. Mostly used in Data Mining and Machine Learning Projects.
3. Data PreProcessing includes
   a. cleaning
   b. normalization
   c. transformation
   d. feature extraction and selection
4. Usually done on training data.
5. Data Split - 60:40 ie., train set has 42000 entries while test set has 28000.
6. There are no null values in the datasets.
7. Implementing normalization to reduce the illumination difference and also to eliminate redundancy.

Data Analysis and Visualization



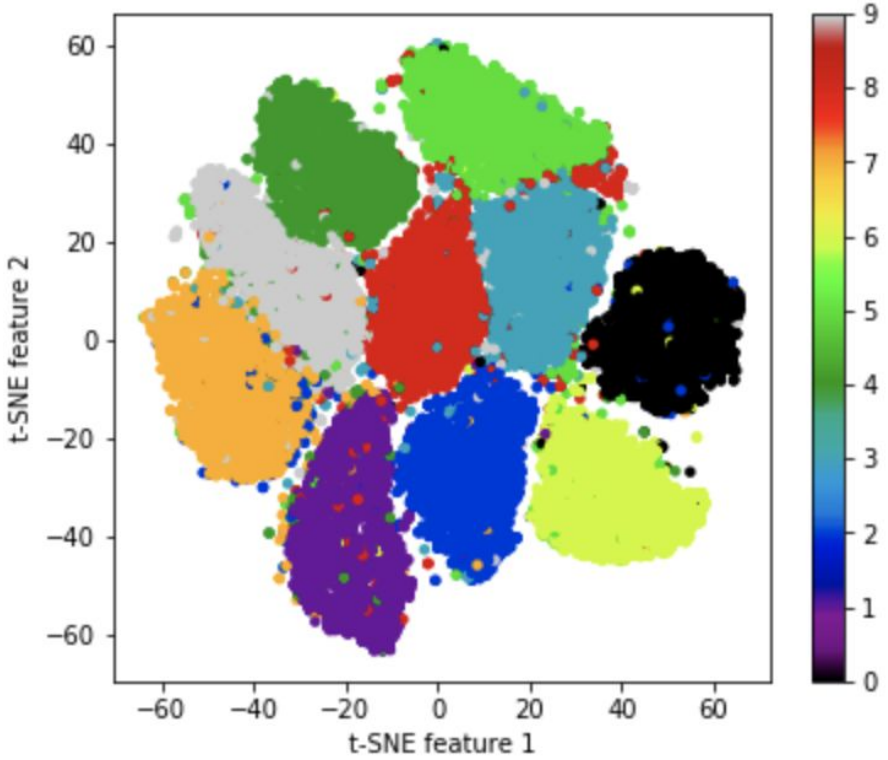
- The histogram illustrates the count of digits in the training data for each number.
- Check unequal sample size among the digits.



t-SNE

Each observation is colored by its class (0-9)

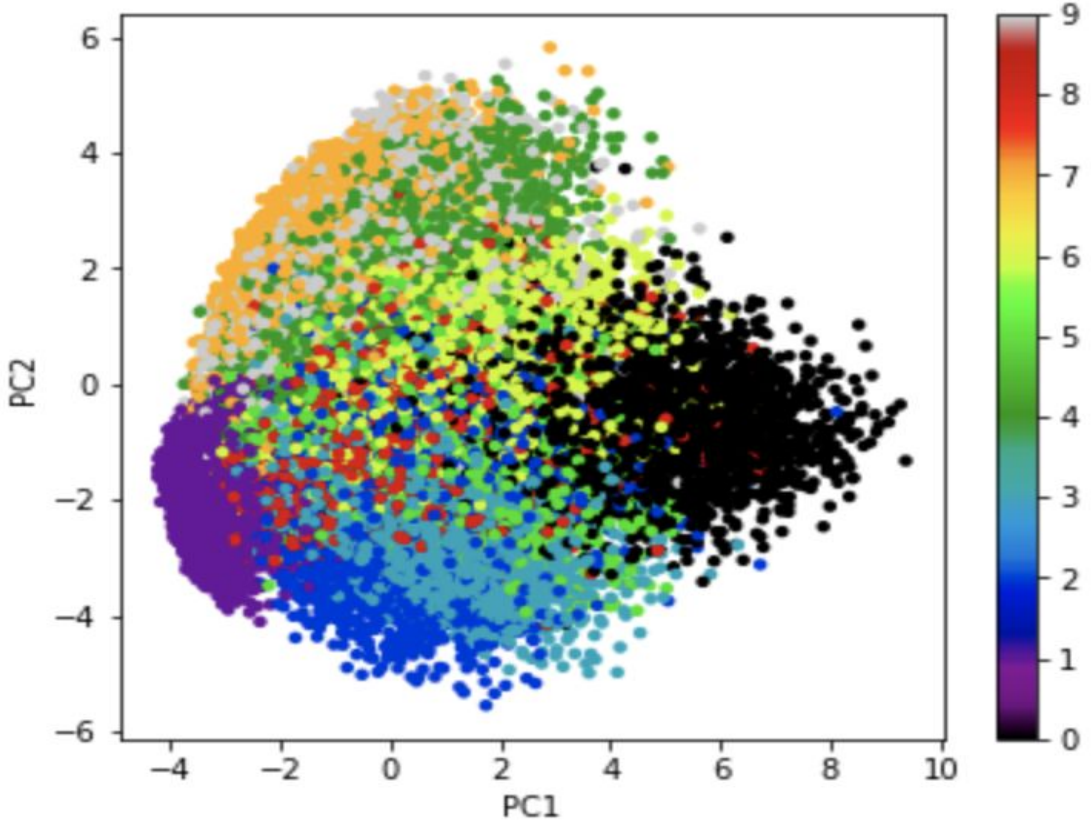
All classes are clearly separated using 2 derived features of t-SNE



PCA

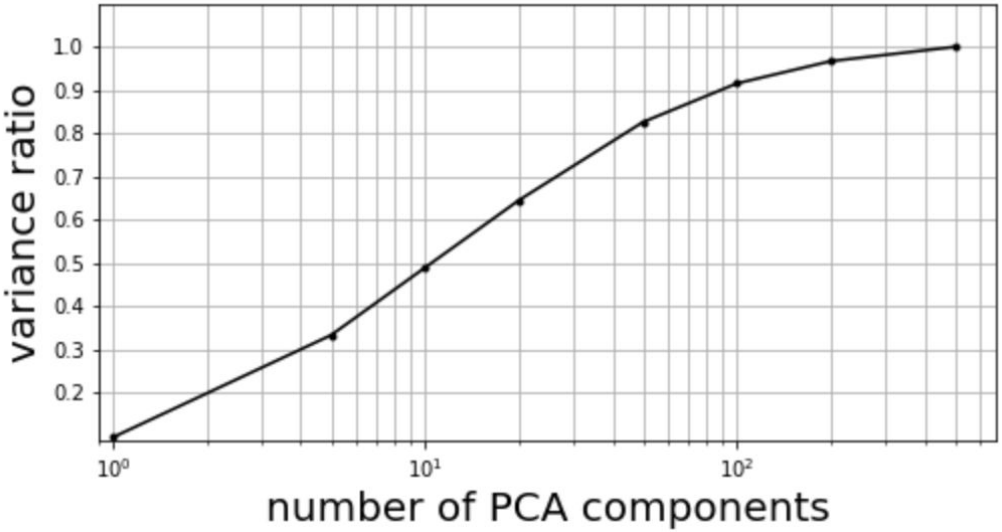
PCA separates the feature space into visible clusters for 2 components. In this situation, classes are not well-separated.

Next, let's look at what happens if we increase the number of components in PCA



PCA

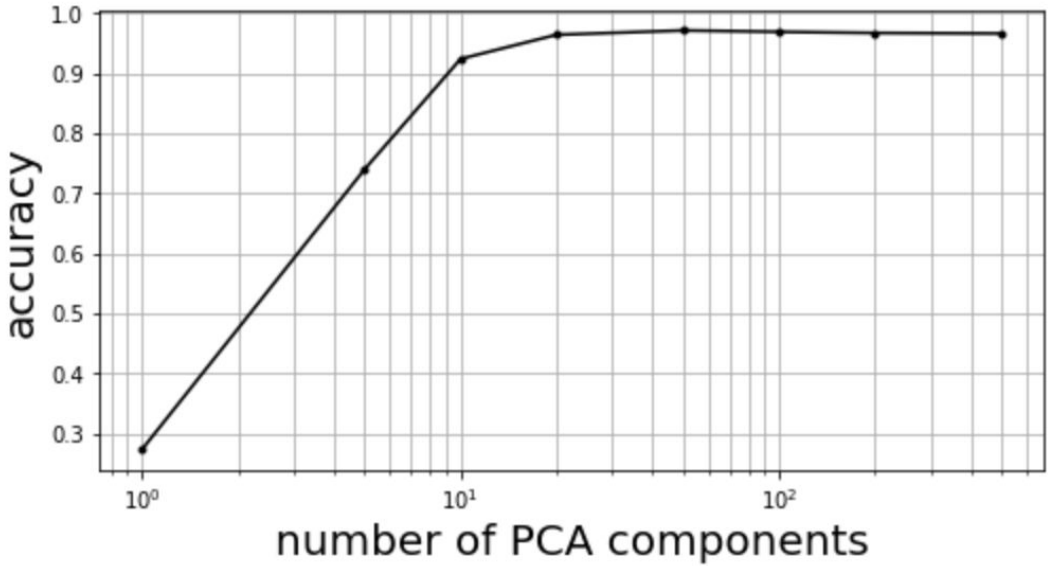
We would like to know how many components are needed to capture most of the variance in the data. We will use the pca.explained_variance_ratio function.



We see that 100 components are needed to capture 90% of the variance in the data. It's a big amount of components. We'll not do that much.

Next, we will train a kNN classifier on the PCA output and figure out the best component number for PCA.

PCA + KNN



The accuracy seems to saturate at 0.95 for almost 50 PCA components. In fact, the accuracy even seems to drop for much larger numbers, even though a larger number of PCA components captures more of the variance in the data.

Thus, we choose 50 as the number of components.

Classification Models

Classification models used for the project:

1. KNN
2. SVM
3. Decision Tree
4. Adaboost
5. Random Forest
6. Neural Networks(MLP Classifier)

Classification Models

1. KNN or lazy learning:
    - Classifies based on majority votes from its neighbors.
2. SVM
    - Supervised learning model with associated algorithm that analyzes data for classification analysis.
3. Decision Tree
    - Decision support tool
    - Uses tree like model of decisions and its consequences

Classification Models

1. Random Forest
   - Ensemble learning method for classification
   - Operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes
2. Adaboost/ Adaptive Boost
   - Machine learning meta-algorithm
   - Can be used in conjunction with many other types of learning algorithms to improve performance
3. Neural Network
   - Computes system similar to biological neural network(human brain)
   - Output from one neuron is the input for the other

Classification Evaluation

Method: 5-fold cross validation

- KNN: K = [1, 5, 10, 15]
- Decision Tree: max_depth = [10, 20, 40, 60, 80, 100]
- Random Forest: n_estimator = [10, 50, 100, 150, 200]
- AdaBoost: RandomForestClassifier
- Neural Network
- SVM: C = [0.1, 1, 5, 10] ; gamma = [0.1, 1, 5]

	KNN (K = 5)	Decision Tree (max_depth = 20)	Random Forest (n_estimators = 200)	AdaBoost	Neural Network	SVM (C = 5, gamma = 0.1)
Accuracy	0.9719	0.8346	0.9502	0.9603	0.938	0.9792

Classification Evaluation

Precision	0	1	2	3	4	5	6	7	8	9
SVM (C = 5, gamma = 0.1)	0.9900892	0.99154691	0.94910714	0.97742239	0.98923679	0.97863248	0.98849473	0.98384201	0.96439169	0.97997998
KNN (K = 5)	0.97949219	0.97844113	0.97309833	0.97089202	0.98120673	0.96875	0.96732026	0.97165633	0.98024948	0.94736842
AdaBoost	0.97828233	0.98810535	0.94977169	0.94304388	0.96771037	0.94532072	0.97245964	0.96855346	0.94501018	0.93939394

Recall	0	1	2	3	4	5	6	7	8	9
SVM (C = 5, gamma = 0.1)	0.98423645	0.98571429	0.98700093	0.97102804	0.97775629	0.98494624	0.98754789	0.97077059	0.9798995	0.96358268
KNN (K = 5)	0.98817734	0.99159664	0.97400186	0.96635514	0.95938104	0.96666667	0.99233716	0.97165633	0.94773869	0.95669291
AdaBoost	0.97635468	0.97731092	0.96564531	0.94392523	0.95647969	0.96666667	0.98084291	0.95482728	0.93266332	0.94586614

Kaggle Performance

We found that the best model is SVM (C = 5, gamma = 0.1) by comparing all the accuracy, precision and recall values.

	Model_names	Kaggle Performance
0	KNN	0.97285
1	Decision_Tree	0.82985
2	Neural_Network	0.93228
3	Random_Forest	0.94814
4	AdaBoost	0.96242
5	SVM	0.97942

- The table illustrates the classification model performance in Kaggle
- KNN, SVM and Neural Network have performance greater than 90%
- Comparatively SVM and KNN are performing better than other models.

Literature Review

Anca Ignat and Bogdan Aciobanitei:

- New feature extraction method: rotating and edge filtering with sobel;
- Rotated input images and computed new grid for pixels, then used vertical, horizontal, and diagonal edge detection filters to get dimensional feature vector.

Mahmoud M. Abu Ghosh and Ashraf Y. Maghari:

- Comparison: CNN, DBN, DNN.
- Accuracy/performance: DNN and CNN > DBN, DNN is more time efficient.

Naigong Yu, Panna Jiao, Yuling Zheng:

- Proposed a combination of LeNet5 and SVM algorithms, using SVM to predict the labels of last two layers of LeNet5.
- Error rate: 0.85% > SVM(1.4%) and CNN (0.95%)