MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent institution of MAHE, Manipal)

Project Report

On

MNIST Digit Dataset

Fundamentals of Machine Learning Lab

Subject Code: DSE 2242

Names	Registration No
Anirudh Swaminathan	220968280
Aditya Singh	220968396
Adwaiy Singh	220968424

Department of Data Science & Computer Applications,

Manipal Institute of Technology,

Manipal

JAN -MAY 2024

Table of Contents

ABSTRACT

In today's digitally driven world, machine learning has emerged as the center of attention in today's technological development, reshaping industries and driving innovation. In the midst of this tech boom, the challenge of accurately recognising human handwriting still remains. However, machine learning provides a promising solution to this issue.

By using different machine learning algorithms for digit classification, we aim to evaluate the performance of machine learning techniques such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gaussian Naive Bayes, XGBoost, Random Forest, and Voting Classifier to analyze which algorithm gives the best result.

KNN achieved an accuracy of 97.05% (for k = 3), XGBoost achieved the highest accuracy of 97.97% whereas Naive Bayes achieved the lowest accuracy of 55.44%.

The results indicate that ensemble-based algorithms such as XGBoost, Random Forest, and SVC excel in capturing the intricate features of handwritten digits, leading to superior classification performance compared to simpler algorithms like Naive Bayes.

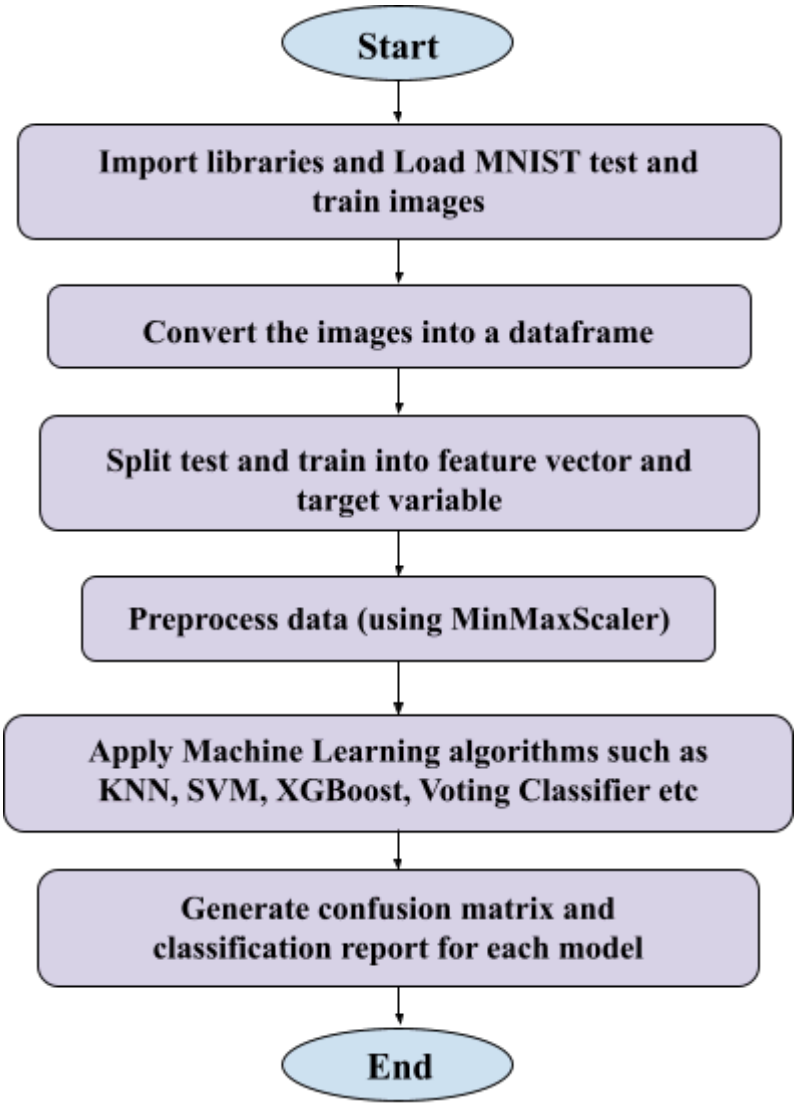
CHAPTER 1
INTRODUCTION

Machine learning encompasses various algorithms and techniques designed to enable systems to learn from data, recognise patterns, and make predictions or decisions with minimal human intervention. Unlike traditional software programs that rely on explicit instructions to perform tasks, machine learning models have the ability to autonomously improve their performance over time by analyzing vast amounts of data and refine their predictive capabilities with the arrival of new data.

Even after such technological advancements, computers still struggle to correctly recognise handwritten numbers. This is due to the handwritten digits varying in style, size, slant and spacing which depends on the person. Furthermore, human handwriting often contains inconsistencies and ambiguities that are easily understood by humans but pose significant hurdles for computers.

To address this issue, we've opted to develop the best machine learning model which accurately classifies handwritten digits. For our project we are going to use one of the most popular datasets which is frequently employed in machine learning models, the MNIST dataset. MNIST, standing for Modified National Institute of Standards and Technology and is a widely-used repository of handwritten digits that serves as a benchmark for training and evaluating image processing systems.

CHAPTER 2
METHODOLOGY



- **Import libraries and Load MNIST -** Import the necessary Python libraries such as NumPy, Pandas, and Scikit-learn for data manipulation and machine learning algorithms. Load the MNIST dataset, containing handwritten digit images along with their corresponding labels, using tensorflow keras datasets.
- **Convert images to data frame -** Each image is flattened into a single row/record where each pixel becomes a feature. The resulting DataFrame consists of rows representing images and columns (785 to be exact) representing pixel values (from 0

to 255). This process allows for easy manipulation and analysis of image data using standard pandas functions..

- **Split test and train -** Divide the dataset into two separate subsets: the training set and the testing set. The training set is used to train the machine learning model, while the testing set is kept separate and used to evaluate the model's performance.
- **Preprocess data -** Scaling features to a specified range i.e. between 0 and 1. This transformation preserves the shape of the original distribution while ensuring that all features have the same scale.
- **Apply Machine Learning algorithms -** We use different machine learning algorithms or models because each algorithm has its own strengths, weaknesses, and assumptions about the data. By using multiple algorithms, we can identify which one best captures the underlying patterns in the dataset, leading to improved model performance and better generalization to unseen data.
- **Generate confusion matrix and classification report -** To gain insight into the performance of a classification model by summarizing the number of correct and incorrect predictions made for each class. It helps identify which classes are often confused with each other, providing information on the model's strengths and weaknesses.

CHAPTER 3
EXPERIMENTAL SETUP

Latest version of python software recommended (3.12.1) but the code would still work on older versions, it can be used on macOS and Linux but cannot be used on Windows version 7 or earlier (later versions are fine). Disk space should at least be 1Gb but 2-3 Gb is recommended. Processor should be an Intel Atom® processor or Intel® Core™ i5 processor. For a smooth code run RAM should be at least an 8Gb one.

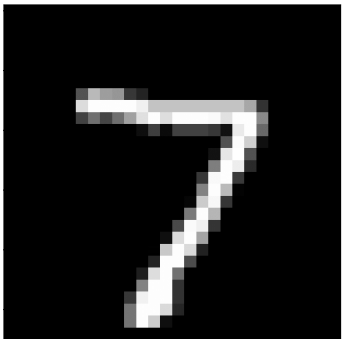
Also, a python simulator like Spyder, PyCharm, Visual Studio Code, Jupyter, etc is necessary (which supports .ipynb file type) to run our code. The directory should have the following python libraries installed -

- jupyter (version 1.0 or later) - provide interactive environment to run python code
- pandas (version 2.2 or later) - for data manipulation and analysis
- tensorflow keras (version 2.16 or later) - to load the MNIST dataset
- numpy (version 1.26 or later) - supports large multi dimensional arrays and matrices
- matplotlib (version 3.8 or later) - plotting library for python to create charts, graphs, etc
- seaborn (version 0.13 or later) - statical data visualization library, based on matplotlib
- sklearn (version 1.4 or later) - provides tools for data mining and data analysis

CHAPTER 4
DATASET

The MNIST dataset, formally known as the Modified National Institute of Standards and Technology database, is one of the most popular and well known dataset in the domain of machine learning and image processing. It was originally derived from a combination of the NIST Special Database 3, which consisted of digits written by employees of the United States Census Bureau, and Special Database 1, which comprised digits written by high school students.

The collection of 70,000 grayscale images of handwritten numbers comprises digits from 0 to 9. Each image within the MNIST dataset is normalized to fit into a consistent 28x28 pixel bounding box and is also grayscaled. This normalization process ensures uniformity across all samples, thereby eliminating potential biases or inconsistencies that might arise from variations in handwriting styles or image quality.



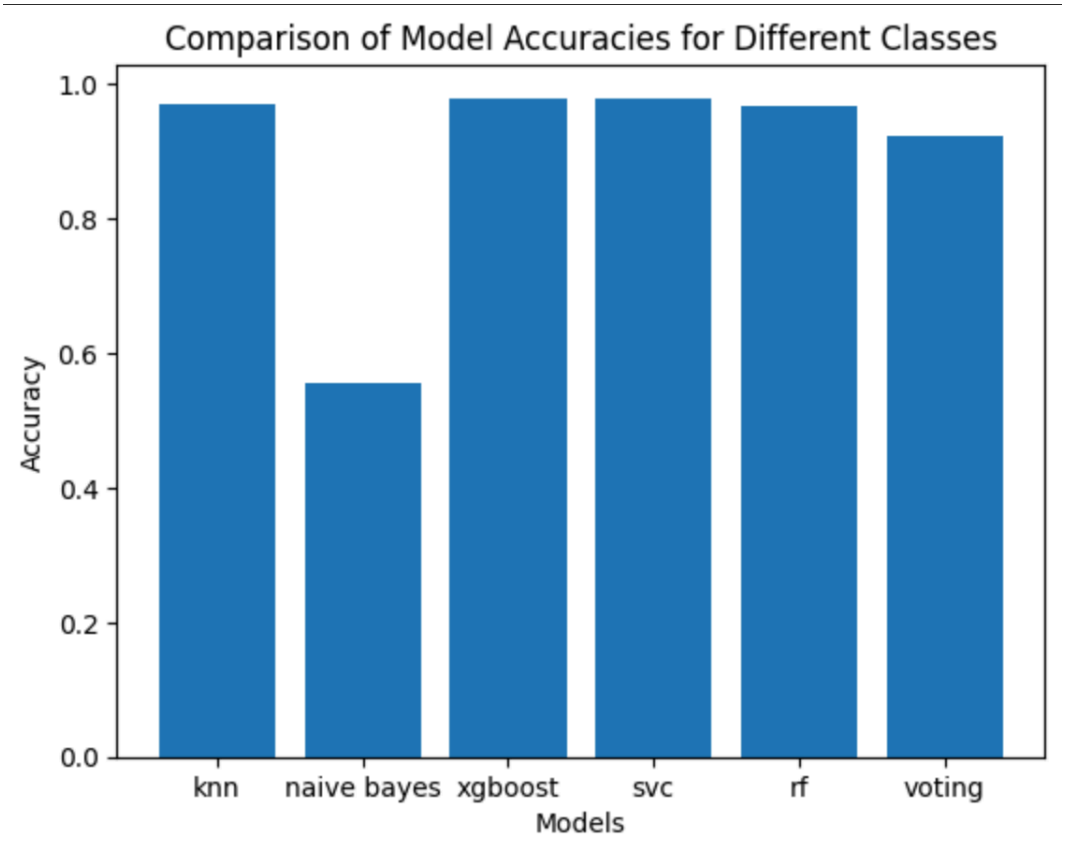
CHAPTER 5
RESULT AND DISCUSSION

The results obtained showcases the effectiveness of various algorithms in digit recognition tasks. Among the algorithms used, k-nearest neighbors (KNN) demonstrated consistent accuracy across different values of 'k', with an average accuracy of approximately 96%. KNN achieved high precision, recall, and F1-scores across all digit classes, indicating its ability to correctly classify digits with minimal errors.

Naive Bayes, achieved the lowest accuracy of 55.44% and showed significant difference in precision, recall, and F1-scores across different digit classes, suggesting difficulties in capturing the patterns in the dataset.

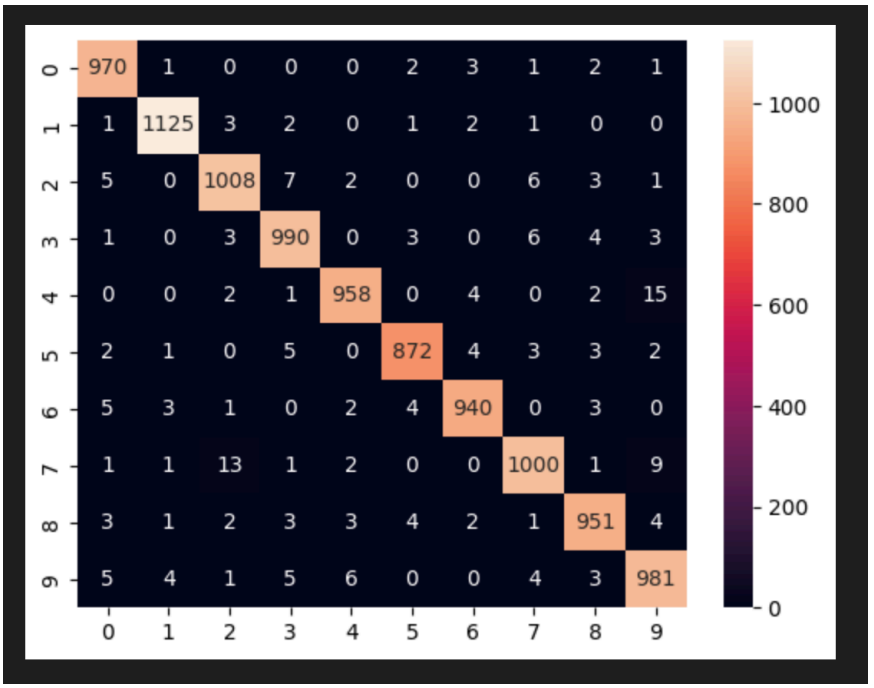
On the other hand, XGBoost, Random Forest, and Support Vector Classifier (SVC) achieved high accuracies, ranging from 97% to 98%, with consistent precision, recall, and F1-scores across all digit classes.

Voting Classifier also achieved a high accuracy of 92% for the combined models of KNN and GaussianNB.



XGBOOST(Best Model):
Classification Report and Confusion Matrix

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.99	0.99	0.99	1135
2	0.98	0.98	0.98	1032
3	0.98	0.98	0.98	1010
4	0.98	0.98	0.98	982
5	0.98	0.98	0.98	892
6	0.98	0.98	0.98	958
7	0.98	0.97	0.98	1028
8	0.98	0.98	0.98	974
9	0.97	0.97	0.97	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000



CHAPTER 6
CONCLUSION

It is clear that algorithm selection plays a crucial role in determining the performance and accuracy of models in digit recognition tasks.

The results illustrate that while simpler algorithms like k-nearest neighbors (KNN) can achieve high accuracies but more sophisticated approaches such as XGBoost, Random Forest, and Support Vector Classifier (SVC) consistently outperform them, reaching accuracies which exceed 97%. These ensemble-based algorithms excel in capturing the features of the images, and perform the classification task better.

Conversely, Naive Bayes struggled to effectively capture the underlying patterns in the dataset, resulting in significantly lower accuracy and reduction in precision, recall, and F1-scores across different digit classes.

Therefore, the choice of algorithm should be made carefully, considering the complexity of the data and the desired level of accuracy, with ensemble-based algorithms emerging as preferable choices for tasks like digit recognition where intricate patterns are involved.