Handwritten Digit Recognition using SVM and Random

Forest Classifier



Gayathree Basarahalli

EC Utbildning

Machine Learning

202503

# Abstract

The project focuses on classifying handwritten digits using the MNIST dataset. Support Vector Machine (SVM) and Random Forest models were trained with hyperparameter tuning. Both models were evaluated using validation and test data based on accuracy score and confusion matrix. The trained models were integrated into an interactive Streamlit app for real-time digit classification.

**Keywords:** MNIST dataset, Exploratory Data Analysis (EDA), Machine Learning (ML), Support Vector Machine (SVM), Random Forest model, hyperparameter tuning, Accuracy Score, Confusion Matrix, Streamlit.

Table of contents

[Abstract 2](#_Toc194702071)

[1 Introduction 4](#_Toc194702072)

[2 Theory 5](#_Toc194702073)

[2.1 Support Vector Machine (Non-linear classifier) 5](#_Toc194702074)

[2.2 Random Forest Model 6](#_Toc194702075)

[3 Method 7](#_Toc194702076)

[3.1 Import necessary Libraries: 7](#_Toc194702077)

[3.2 Data Preprocessing : 7](#_Toc194702078)

[3.3 Machine Learning Models 8](#_Toc194702079)

[3.3.1 SVM model 8](#_Toc194702080)

[3.3.2 Random Forest model 8](#_Toc194702081)

[3.4 Evaluation with Test data 9](#_Toc194702082)

[3.5 Construction of a Streamlit application 9](#_Toc194702083)

[4 Result and Discussion 10](#_Toc194702084)

[4.1 Accuracy Score 10](#_Toc194702085)

[4.2 Evaluation with test data 10](#_Toc194702086)

[4.2.1 Model accuracy : 10](#_Toc194702087)

[4.2.2 Confusion Matrix : 11](#_Toc194702088)

[5 Conclusions 12](#_Toc194702089)

[References 13](#_Toc194702090)

# Introduction

Machine learning is a process where data is used to help a computer learn without direct human interaction. It is considered a subset of artificial intelligence (AI). Machine learning uses algorithms to identify patterns within data, and these patterns are then used to create a data model that can make predictions. With more training data, the accuracy of machine learning models improves.

The current report considers the MNIST dataset (LeCun, 2024), which is a collection of handwritten digits with corresponding true values (e.g., labels), often used for practicing ML.  
By using ML with the MNIST dataset, which consists of 70,000 images of handwritten digits from 0 to 9, different machine learning models can be trained to classify these digits.  
The purpose of this report is to use different machine learning models to recognize handwritten digits.

In this project, two popular and powerful machine learning models were used: **Support Vector Machine (SVM)** and **Random Forest Classifier**. SVM is known for its effectiveness in high-dimensional spaces and its strong performance with well-separated classes, especially when using kernels like the radial basis function (RBF) for non-linear classification. Random Forest, on the other hand, is a robust ensemble model that reduces overfitting and performs well on a variety of classification problems. These models were selected for their proven reliability, interpretability, and ability to handle large datasets like MNIST efficiently.  
To achieve this purpose, the following research questions are relevant,

1. Develop models that achieve accuracy greater than 95%?
2. What is the impact of normalizing pixel values (scaling from 0 to 1) on the performance of the models?
3. Can handwritten digits be predicted accurately?
4. Is it good to consider whole data while working with models? How does the model behave with validation data and test data?
5. Construction of a Streamlit app allowing the user to write and identify digits?

# Theory

## Support Vector Machine (Non-linear classifier)

**SVM (Support Vector Machine)** is a supervised machine learning algorithm that can be used for classification or regression tasks. It works by finding a hyperplane that best separates the data into different classes. SVM is especially effective in high-dimensional spaces and is well-suited for complex, non-linear decision boundaries.

**Kernel:** When the data is not linearly separable, it is crucial to use a kernel, as it determines how the model handles and separates the data. Choosing the right kernel is important for achieving the best results, since different kernels can perform differently depending on the properties of the data.

A screenshot of a graph

AI-generated content may be incorrect.

Figure 1 SVM- classifier with an RBF kernel

## Random Forest Model

**Random Forest** is an ensemble learning algorithm based on decision trees, most often trained using “bagging” and sometimes “pasting.” It builds a collection of decision trees, each trained on a random subset of the data, and combines their predictions to improve accuracy. This method helps reduce overfitting compared to using a single decision tree, making it a powerful tool in machine learning.

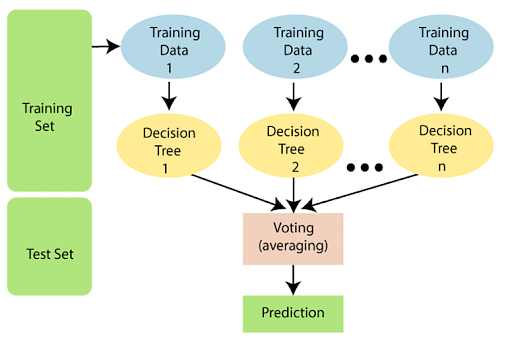


Figure 2 Random Forest Model

# Method

## Import necessary Libraries:

To work with machine learning model, we need to import all necessary libraries. Following are libraries we considered for MNIST data.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.datasets import fetch\_openml

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import GridSearchCV

from sklearn.decomposition import PCA

from sklearn.svm import SVC

from sklearn.model\_selection import StratifiedKFold

from sklearn.ensemble import RandomForestClassifier

import seaborn as sns

import joblib

from sklearn.ensemble import VotingClassifier

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

## Data Preprocessing :

The MNIST data was retrieved from Scikit-Learn's datasets and imported using the fetch\_openml function. The dataset consists of 70,000 handwritten digit images (0 to 9), with each image having a resolution of 28×28 pixels. The input data was stored in the variable x, while the target labels were stored in y. To gain a better understanding of the dataset, Exploratory Data Analysis (EDA) was performed, which included examining the data structure and visualizing the images.

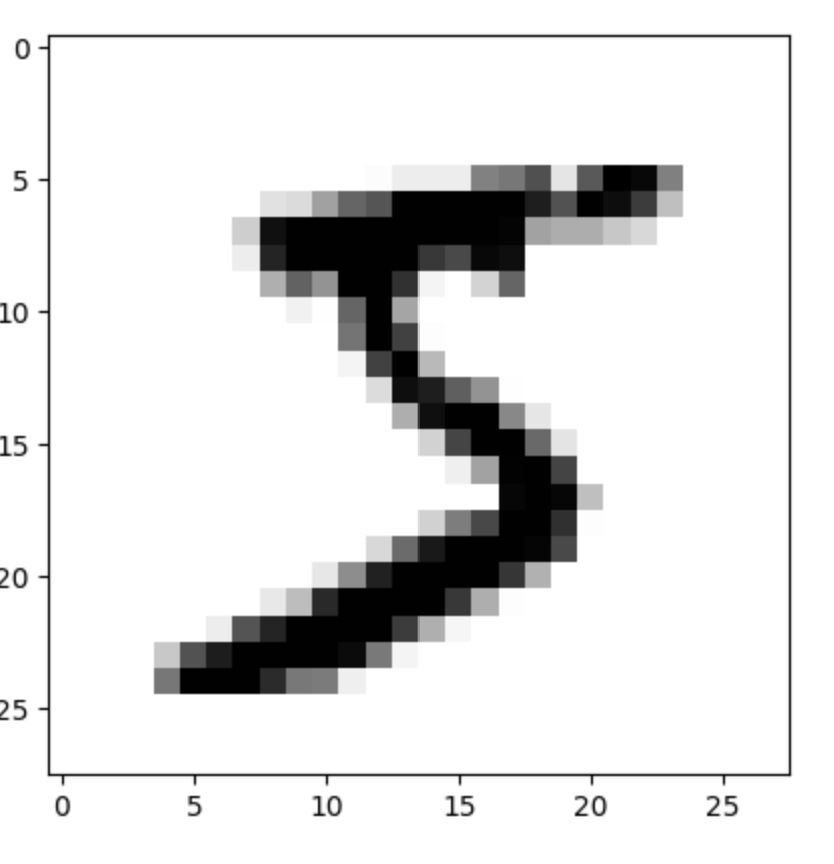


Figure 3 Example image of digit 5

x\_train = X[ :50000]

y\_train = y[ :50000]

x\_val = X[50000:60000]

y\_val = y[50000:60000]

x\_test = X[60000:70000]

y\_test = y[60000:70000]

The data was normalized so that all pixel values lie between 0 and 1

x\_train = x\_train / 255.0

x\_val = x\_val / 255.0

x\_test = x\_test / 255.0

## Machine Learning Models

### SVM model

In the model, we used the RBF kernel and Stratified K-Fold. We applied the Grid Search method to test different combinations of hyperparameters in order to identify the best ones for the model. Using the optimal parameters, the SVM model was trained on the training data. With the validation data, we achieved an accuracy score of 98.42% for the SVM model.

### Random Forest model

In this model, we used Random Forest Classifier and trained the data. With the validation data, we achieved an accuracy score of 97.34% for the Random Forest model.

## Evaluation with Test data

To evaluate the models, **Accuracy Score** and **Confusion Matrix** were used for both the **SVM** and **Random Forest** models. The Accuracy Score provides a clear understanding of each model's performance on new, unseen data. The Confusion Matrix offers a more detailed view by showing where and how often the models made incorrect predictions. These evaluation metrics helped assess and compare the effectiveness of both models in recognizing handwritten digits.

## Construction of a Streamlit application

Streamlit application was built with the following list of user allowed functionalities:

The Streamlit application was built with the following set of user-allowed functionalities:

* Select the model using radio button.
* Draw the digit on an interactive canvas.
* **Button** to trigger the digit prediction.
* **Preprocessing** of the drawn image to match the model input format.
* **Display** of the processed image for user confirmation.
* **Predicting** the correct digit using the selected machine learning model.

The code for the Streamlit application was developed in PyCharm. The preprocessing steps used the same code as in previous models. The functionalities were created by referring to both ChatGPT and the official Streamlit documentation (Sam Altman, 2022; Streamlit documentation, 2024)

# Result and Discussion

## Accuracy Score

Table 1: *We evaluated the results from two models using the MNIST dataset. The accuracy score for the two models with* ***validation data****:*

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Hyper parameters |
| SVM | 98.42 | Kernal : rbf, C = 10, Cross validation = Stratified Cross validation (n\_splits =5), gamma = scale |
| Random Forest | 97.34 | n\_jobs = -1 ,random\_state =42 |

## Evaluation with test data

### Model accuracy :

After the model is saved using Joblib, we evaluate the model using test data, which is unseen by the model during training. This allows us to evaluate the model's performance and understand how well it generalizes to new, unseen data.

*Table 2: We evaluated the results from two models using the MNIST dataset. The accuracy score for the two models with* ***test data****:*

|  |  |
| --- | --- |
| Model | Accuracy |
| SVM | 98.36 |
| Random Forest | 96.80 |

### Confusion Matrix :

A screenshot of a graph

AI-generated content may be incorrect.

*Figure 4* ***Confusion Matrix for SVM****: In the image, we can see the actual and predicted values, as well as how many times the model made incorrect predictions for each digit.*

*A graph of a graph with numbers

AI-generated content may be incorrect.*

*Figure 5* ***Confusion Matrix for Random Forest Classifier****: In the image, we can see the actual and predicted values, as well as how many times the model made incorrect predictions for each digit.*

# Conclusions

The following conclusions regarding set out objectives, tasks and research questions can be drawn:

* Both the models SVM and Random Forest Classifier achieved an accuracy of 98.42 and 97.34 respectively, which is greater than 95%.
* Normalizing pixel values by scaling them to lie between 0 and 1, had a positive impact on model performance.
* Handwritten digits can indeed be predicted accurately with machine learning models. By doing proper preprocessing steps we can achieve appropriate results.
* Using the whole dataset can be useful for better predictions. When most of the data is used for training the model, it performs well on the validation and test datasets, demonstrating its ability to generalize effectively to unseen data and underscoring the importance of proper data splitting.
* A **Streamlit app** was successfully deployed, enabling users to draw handwritten digits and receive predictions in real-time. The model predicts the digits correctly as reflected in the confusion matrix.

# References

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, & TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Prgomet, A. (2023, June 3). Maskininlärning [Video]. YouTube. <https://www.youtube.com/watch?v=zORv5vxrwok&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45>

GeeksforGeeks. (2025). A comprehensive guide to ensemble learning. GeeksforGeeks. <https://www.geeksforgeeks.org/a-comprehensive-guide-to-ensemble-learning/?ref=header_outind>

Sam Altman. (2022). Hämtat från <https://chat.openai.com/>

Streamlit documentation. (2024). Hämtat från https://docs.streamlit.io/