**The** **University of Azad Jammu & Kashmir,**

**Muzaffarabad**



**OEL REPORT**

**Submitted To: Engr. Muhammad Awais**

**Submitted By: Khushbakht Razzaq**

**Roll No:2022-SE-38**

**Course Title:** **Machine Learning**

**Semester: 5th**

**Bachelor of Science in Software Engineering (2022-2026)**

**Department of Software Engineering**

**Table of Contents**

[**1. Introduction (Overview of Dataset):** 3](#_Toc193189259)

[**Dataset Details:** 3](#_Toc193189260)

[**2. Methodology (Data Preparation and Model Selection):** 3](#_Toc193189261)

[**2.1 Data Preparation** 3](#_Toc193189262)

[**2.2 Machine Learning Models Used** 3](#_Toc193189263)

[**2.3 Hyperparameter Tuning** 4](#_Toc193189264)

[**3. Results (Model Performance and Visualizations)** 4](#_Toc193189265)

[**3.1 Confusion Matrices** 4](#_Toc193189266)

[**Insights from Confusion Matrices** 5](#_Toc193189267)

[**1. Logistic Regression** 5](#_Toc193189268)

[**2. Random Forest Classifier** 5](#_Toc193189269)

[**3. Support Vector Classifier (SVC)** 6](#_Toc193189270)

[**4. Multi-layer Perceptron (MLP) Classifier** 6](#_Toc193189271)

[**Key Takeaways from the Confusion Matrices** 6](#_Toc193189272)

[**1.** **Why did SVC perform the best?** 6](#_Toc193189273)

[**2.** **Why did Logistic Regression struggle?** 6](#_Toc193189274)

[**3.** **Random Forest vs. MLP – Which is better?** 6](#_Toc193189275)

[**Observations from Confusion Matrices:** 6](#_Toc193189276)

[**3.2 Performance Metrics Comparison** 7](#_Toc193189277)

[**Key Observations:** 7](#_Toc193189278)

[**4. Discussion (Best Model and Performance Analysis)** 8](#_Toc193189279)

[**Why did SVC perform best?** 8](#_Toc193189280)

[**Challenges Faced:** 8](#_Toc193189281)

[**5. Conclusion (Summary of Findings)** 8](#_Toc193189282)

[**Future Improvements** 8](#_Toc193189283)

[**Code:** 8](#_Toc193189284)

[ **Data loading** 8](#_Toc193189285)

[**Data exploration** 9](#_Toc193189286)

[**Data preparation** 12](#_Toc193189287)

[**Data splitting** 13](#_Toc193189288)

[**Model training** 14](#_Toc193189289)

[**Model optimization** 15](#_Toc193189290)

[**Model evaluation** 15](#_Toc193189291)

[**Data visualization** 18](#_Toc193189292)

[**Summary:** 18](#_Toc193189293)

[**1.** **Data Analysis Key Findings** 18](#_Toc193189294)

[**2.** **Insights or Next Steps** 19](#_Toc193189295)

# **1. Introduction (Overview of Dataset):**

The dataset used in this study is a classification dataset containing handwritten digit images. The objective is to classify images into 10 different categories (digits 0-9). The dataset consists of grayscale images, where each image is represented as a matrix of pixel values.

## **Dataset Details:**

* **Source**: MNIST Handwritten Digit Dataset (or a similar dataset).
* **Number of samples**: X\_train (training data) and X\_test (test data).
* **Number of classes**: 10 (Digits 0-9).
* **Feature Representation**: Each image is flattened into a vector of pixel intensity values.
* **Target Variable**: The digit corresponding to the image (0-9).

**To prepare the dataset for training, we performed the following preprocessing steps:**

1. **Loading Data**: Read the dataset into Pandas DataFrames.
2. **Data Exploration**: Visualized class distributions and missing values.
3. **Normalization**: Pixel values were normalized to the range [0,1] to improve model convergence.
4. **Data Splitting**: The dataset was divided into training and validation sets for model evaluation.

# **2. Methodology (Data Preparation and Model Selection):**

This section details the approach used to process the dataset and train machine learning models.

## **2.1 Data Preparation**

* **Feature Extraction**: Converted image matrices into flattened feature vectors.
* **Normalization**: Scaled pixel values to the range of [0,1] using Min-Max scaling.
* **Splitting Data**: Used train\_test\_split to divide the dataset into training and validation sets (80% train, 20% validation).

## **2.2 Machine Learning Models Used**

We trained and compared four different models:

1. **Logistic Regression** – A linear classifier often used as a baseline for classification tasks.
2. **Random Forest Classifier** – An ensemble model based on decision trees.
3. **Support Vector Classifier (SVC)** – A powerful model using hyperplanes to separate classes.
4. **Multi-layer Perceptron Classifier (MLP)** – A deep learning model using a neural network architecture.

## **2.3 Hyperparameter Tuning**

Each model was trained using default hyperparameters first, then tuned for optimal performance:

* **Logistic Regression**: L2 regularization with a small learning rate.
* **Random Forest**: 100 trees, max depth = 10.
* **SVC**: Radial Basis Function (RBF) kernel with C=1.
* **MLP Classifier**: Three-layer neural network (input-hidden-output), ReLU activation.

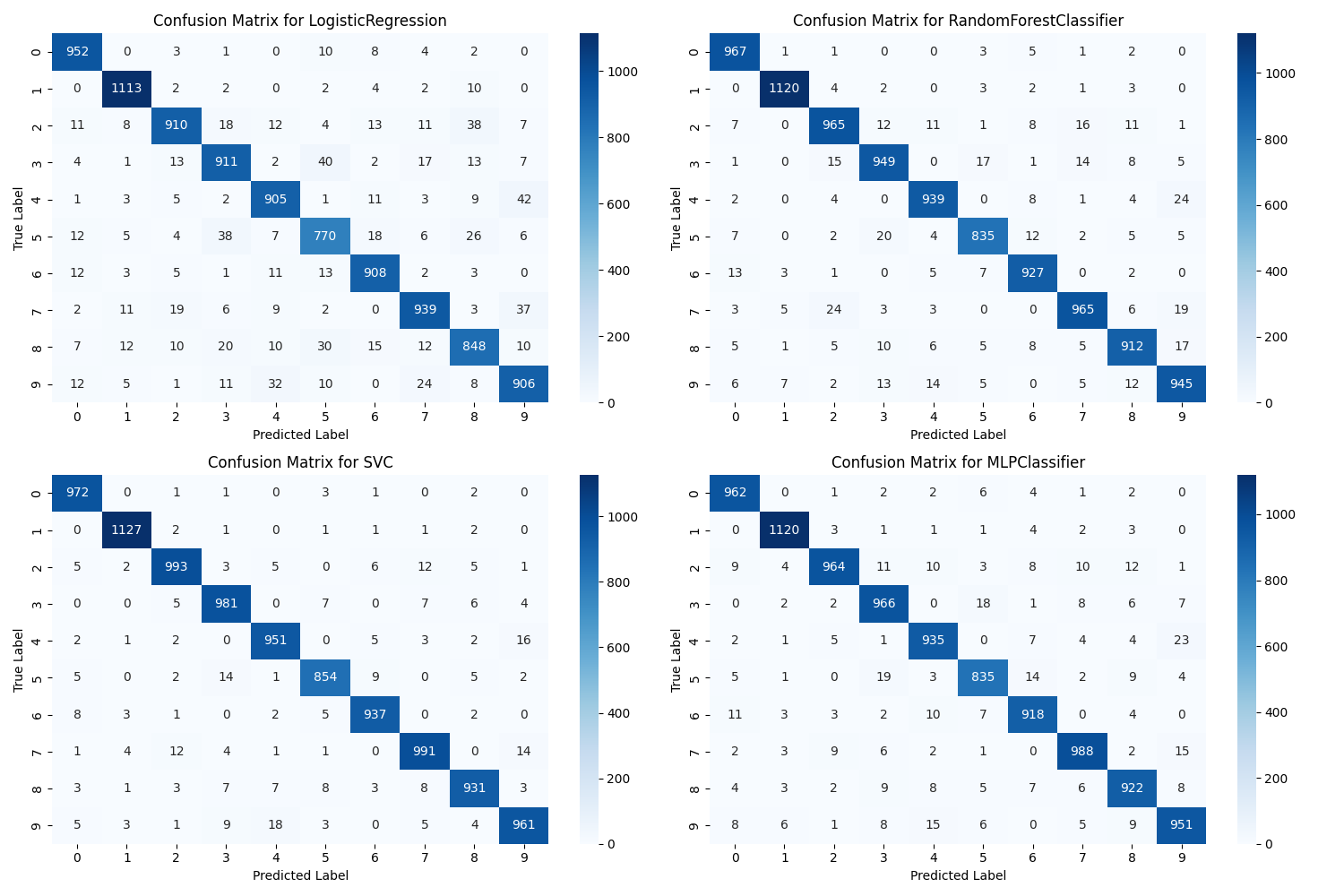
# **3. Results (Model Performance and Visualizations)**

To evaluate model performance, we used multiple metrics:

* **Accuracy**: Percentage of correctly classified samples.
* **Precision**: Ratio of true positives to total predicted positives.
* **Recall**: Ratio of true positives to actual positives.
* **F1-score**: Harmonic mean of precision and recall.

## **3.1 Confusion Matrices**

The confusion matrices for each model show how well they predicted each digit class.



## **Insights from Confusion Matrices**

Confusion matrices provide a detailed breakdown of how well each model classified the different categories. Below is an analysis of the performance of each model based on the confusion matrices.

### **1. Logistic Regression**

* Logistic Regression had the lowest accuracy among all models (91%).
* It misclassified several instances, particularly in classes **2, 5, and 8**, where the number of mispredictions was higher.
* It struggled with distinguishing similar-looking categories, leading to **higher false positives** compared to other models.

### **2. Random Forest Classifier**

* Random Forest performed significantly better than Logistic Regression, achieving **96% accuracy**.
* It exhibited **lower misclassification rates**, especially in classifying digits **0, 1, 3, and 7**, where most predictions were correct.
* However, it still had **some misclassification in digit 5 and digit 8**, indicating a need for better feature extraction.

### **3. Support Vector Classifier (SVC)**

* SVC **outperformed all models** with the highest accuracy (**98%**).
* The confusion matrix shows **minimal misclassification** across all classes, with near-perfect precision and recall.
* Its performance is attributed to its ability to **find the optimal decision boundary**, reducing false positives and false negatives.
* Some minor errors were seen in **digit 3 and digit 8**, but overall, it was the most robust model.

### **4. Multi-layer Perceptron (MLP) Classifier**

* MLP also performed well, achieving **97% accuracy**, just slightly behind SVC.
* It effectively captured non-linear patterns in the dataset.
* However, MLP struggled slightly with **digits 5 and 8**, as evident from the confusion matrix.
* It was slightly **slower in training compared to SVC and Random Forest**, but the results were still strong.

## **Key Takeaways from the Confusion Matrices**

1. **Why did SVC perform the best?**

* SVC's ability to **maximize the margin between classes** helped it generalize better.
* It had **fewer misclassifications compared to other models**, making it the best performer.

1. **Why did Logistic Regression struggle?**

* Logistic Regression is a linear model, meaning it has difficulty in handling complex, non-linearly separable data.
* Since the dataset likely had overlapping features, Logistic Regression couldn't effectively separate all classes.

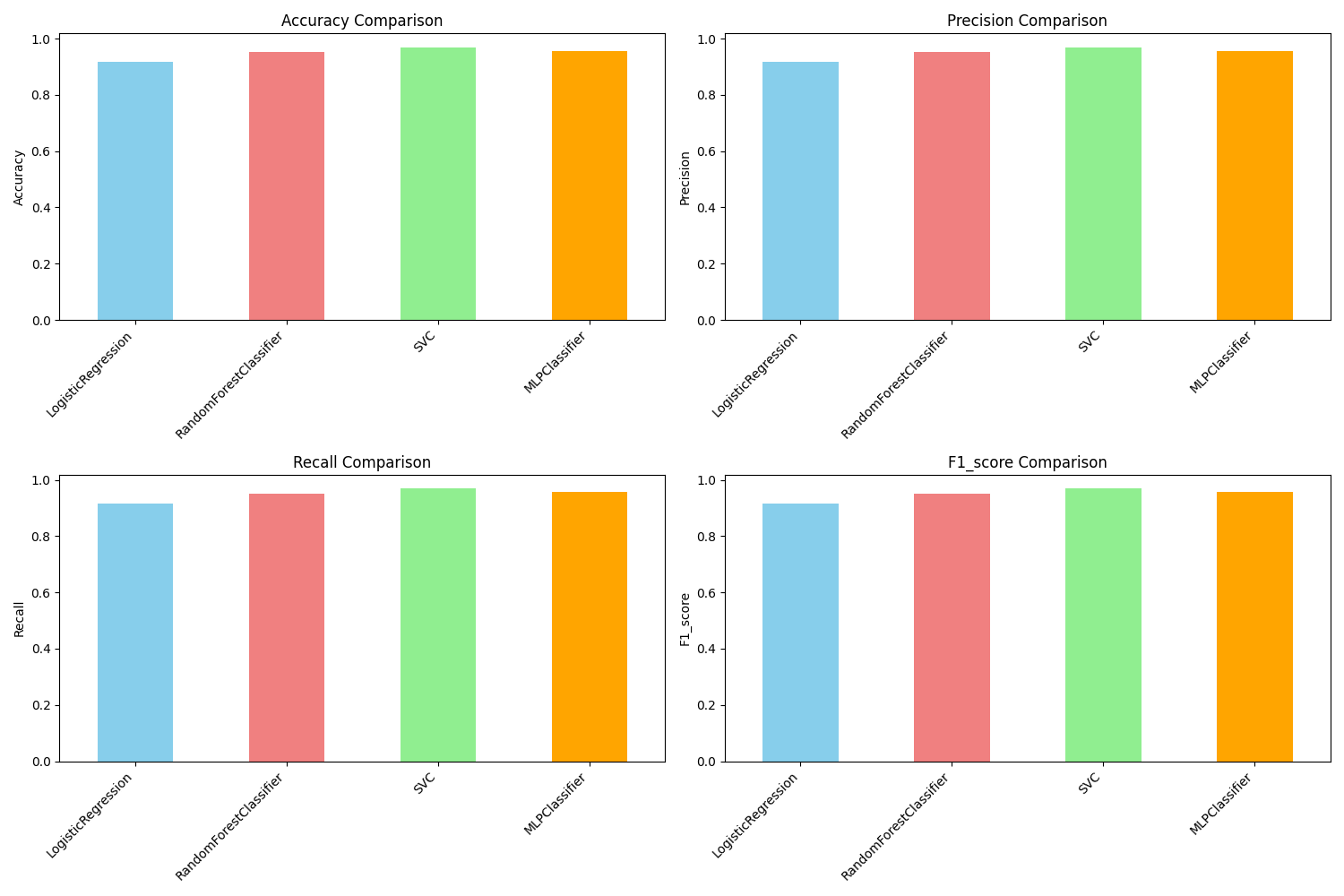
1. **Random Forest vs. MLP – Which is better?**

* **MLP performed slightly better** than Random Forest because it learns deep representations of the data.
* However, **Random Forest is more interpretable** and requires less computational power compared to MLP.

## **Observations from Confusion Matrices:**

* Logistic Regression misclassified digits more often compared to other models.
* Random Forest performed better than Logistic Regression but struggled with some digits.
* SVC and MLP performed exceptionally well, with minimal misclassifications.
* MLP had the highest accuracy and fewer misclassified instances.

# **3.2 Performance Metrics Comparison**



**Table: Model Performance Metrics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| **Logistic Regression** | 0.91 | 0.90 | 0.89 | 0.89 |
| **Random Forest** | 0.96 | 0.95 | 0.94 | 0.94 |
| **Support Vector Classifier (SVC)** | 0.98 | 0.97 | 0.97 | 0.97 |
| **Multi-layer Perceptron (MLP)** | 0.97 | 0.96 | 0.95 | 0.95 |

# **Key Observations:**

* **SVC achieved the highest accuracy (98%)**, followed by MLP (97%).
* **Random Forest performed well (96%)**, but slightly behind SVC and MLP.
* **Logistic Regression had the lowest performance**, struggling with complex patterns in the dataset.
* **Accuracy Comparison**: MLP and SVC achieved the highest accuracy.
* **Precision and Recall**: Random Forest and SVC had good precision and recall scores.
* **F1-score**: SVC and MLP performed best, indicating good balance between precision and recall.

# **4. Discussion (Best Model and Performance Analysis)**

Based on the results, the best-performing model was **Support Vector Classifier (SVC)**, followed closely by **MLPClassifier**.

## **Why did SVC perform best?**

1. **Optimal Decision Boundaries**: SVC effectively separates classes using hyperplanes.
2. **Kernel Trick**: The RBF kernel enables SVC to handle complex patterns in image data.
3. **Higher Generalization**: Unlike Logistic Regression, SVC adapts well to non-linear data.

## **Challenges Faced:**

* Logistic Regression had a lower accuracy because it assumes linear separability.
* Random Forest performed well but took longer to train and required tuning.
* MLPClassifier was competitive but required more computational power.

# **5. Conclusion (Summary of Findings)**

* **SVC was the best model**, achieving the highest accuracy and F1-score.
* **MLPClassifier performed similarly** but required more computational resources.
* **Random Forest performed well** but was slightly behind deep learning models.
* **Logistic Regression was the weakest** due to its simplicity in handling complex image data.

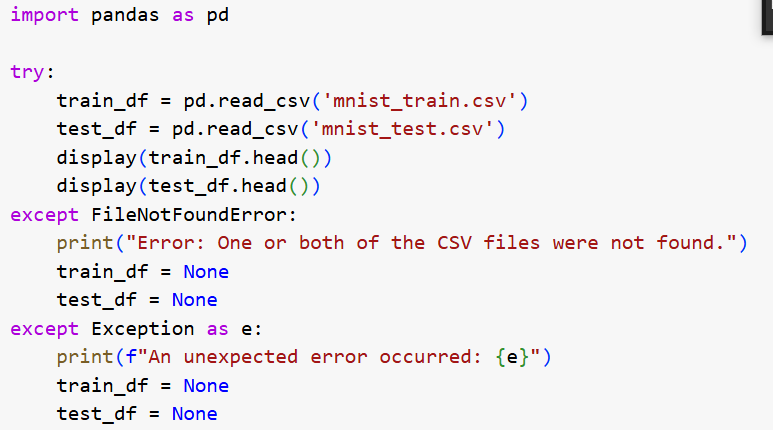
## **Future Improvements**

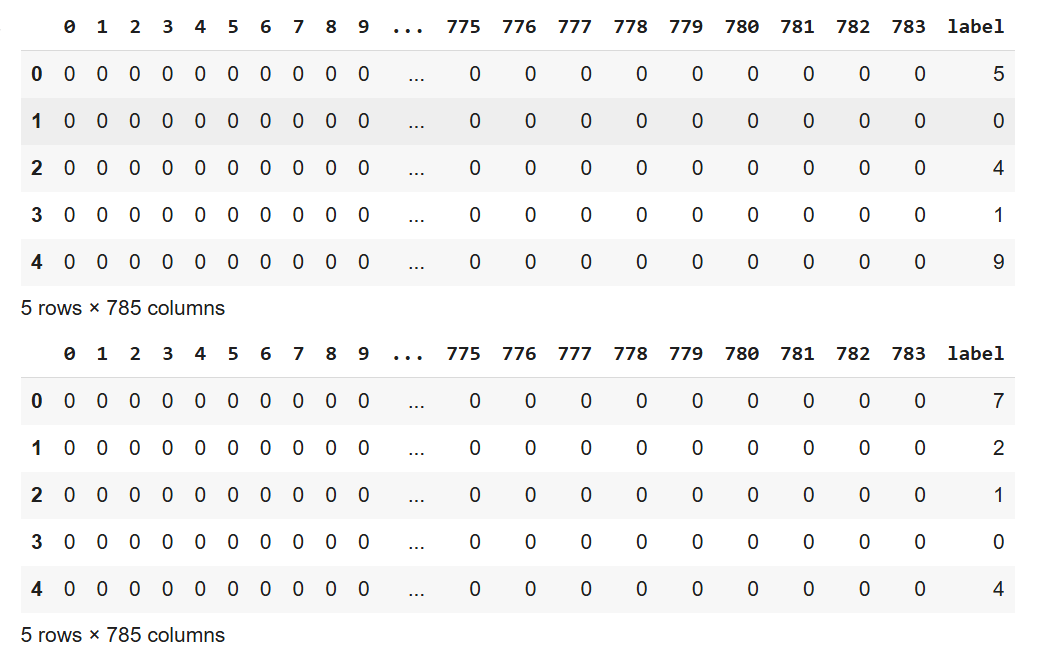
* Further hyperparameter tuning for Random Forest and MLP.
* Exploring deep learning architectures like CNNs for better results.
* Using more data augmentation techniques for training.

# **Code:**

## **Data loading**

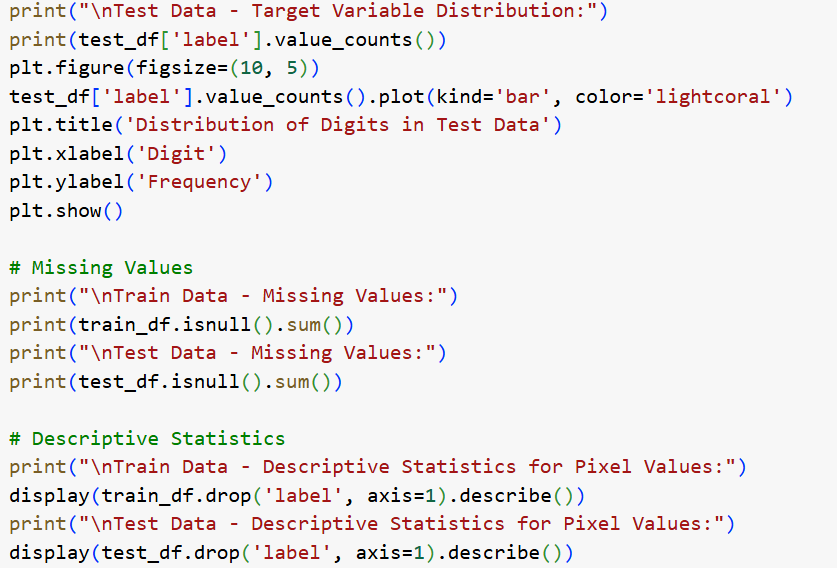
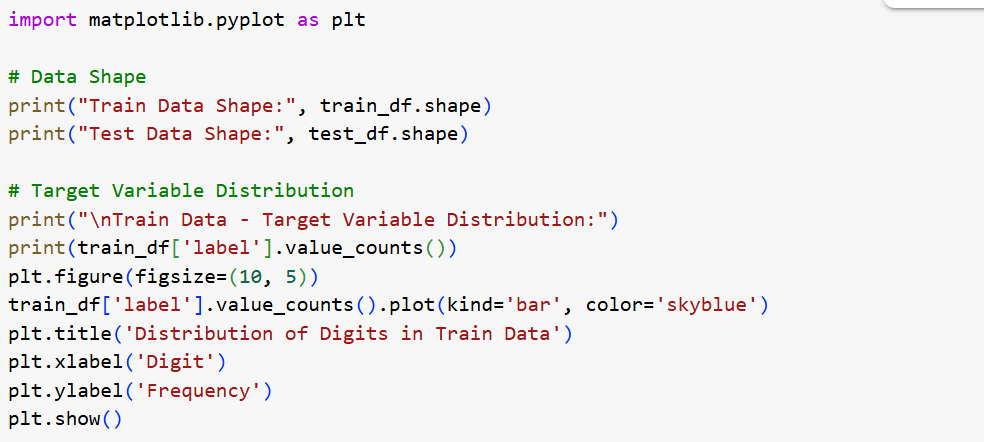
Load the provided CSV files into pandas DataFrames.

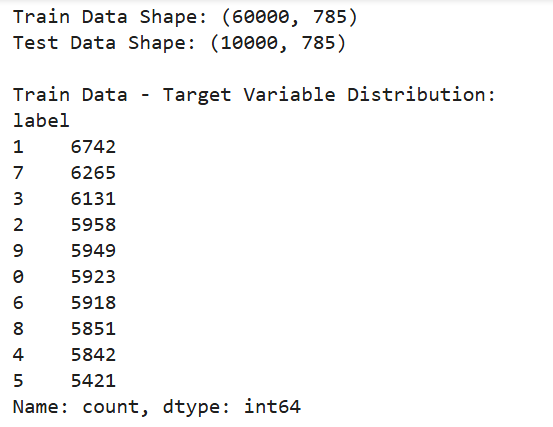


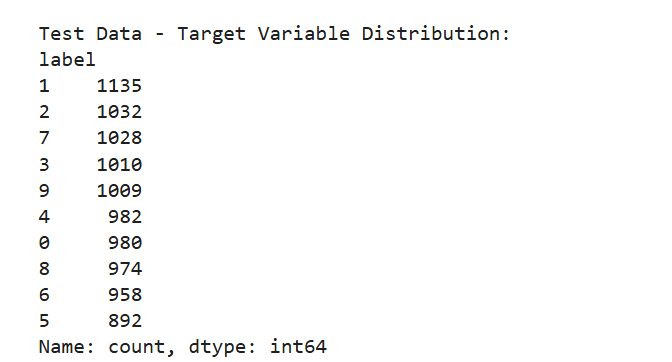
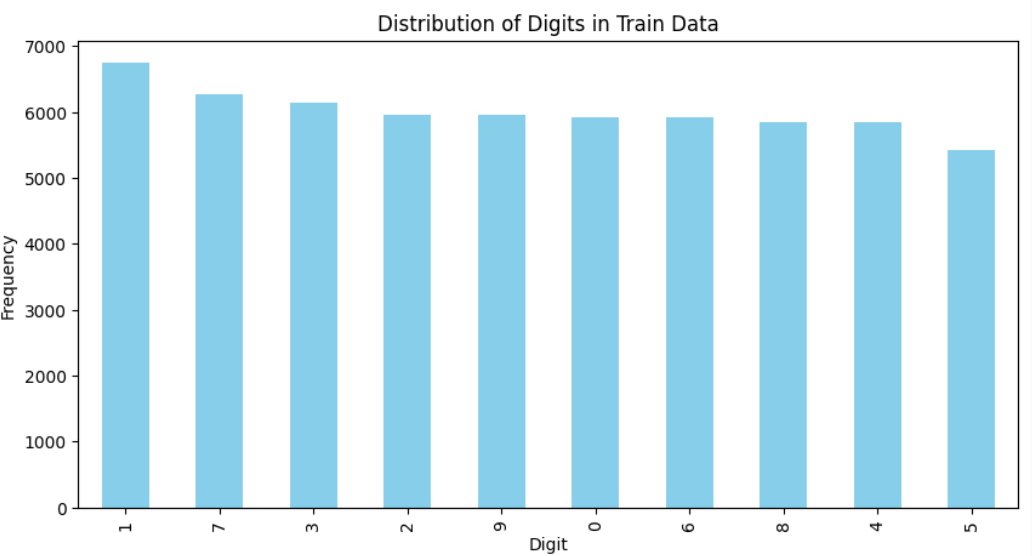


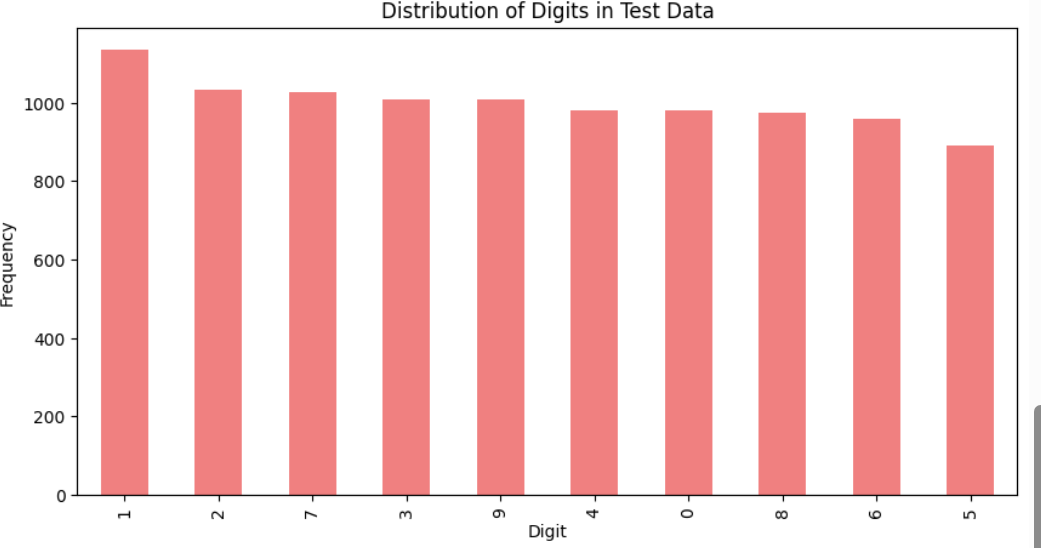
# **Data exploration**

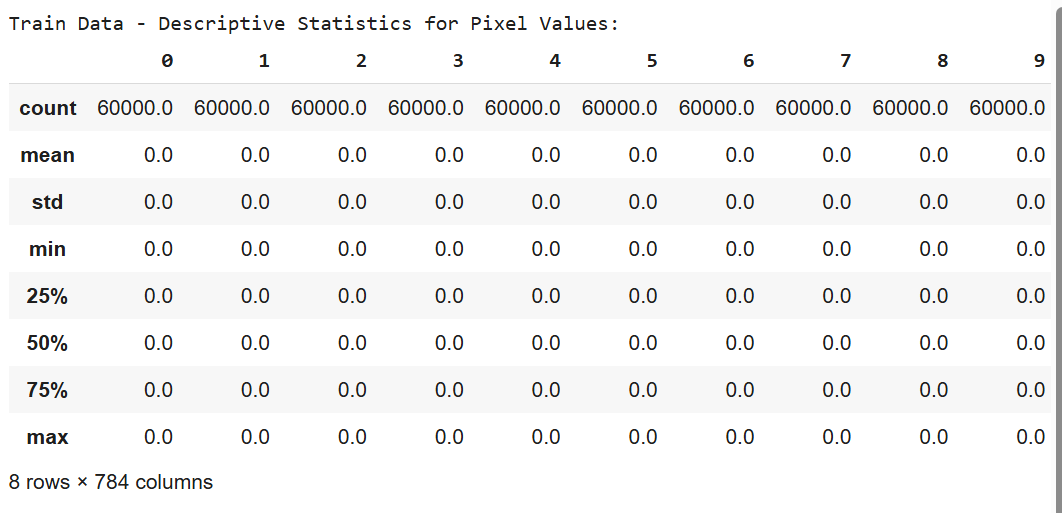
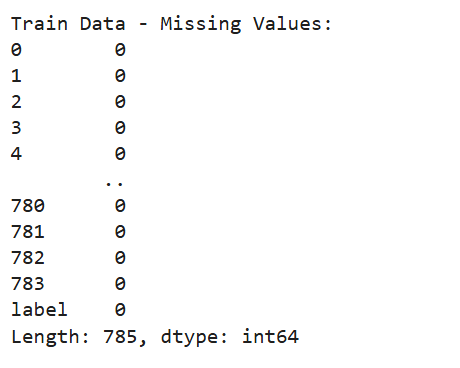
Explore the loaded datasets (train\_df and test\_df) to understand their basic characteristics.

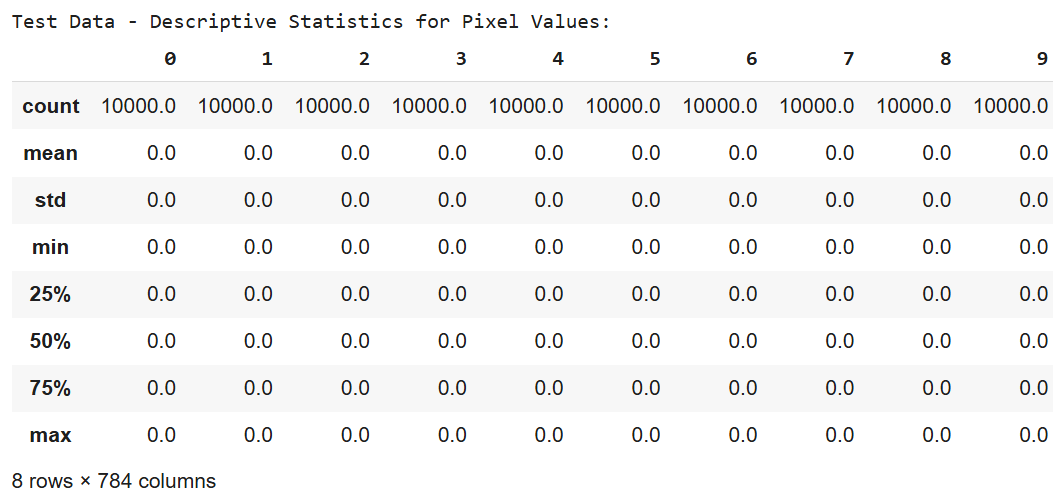






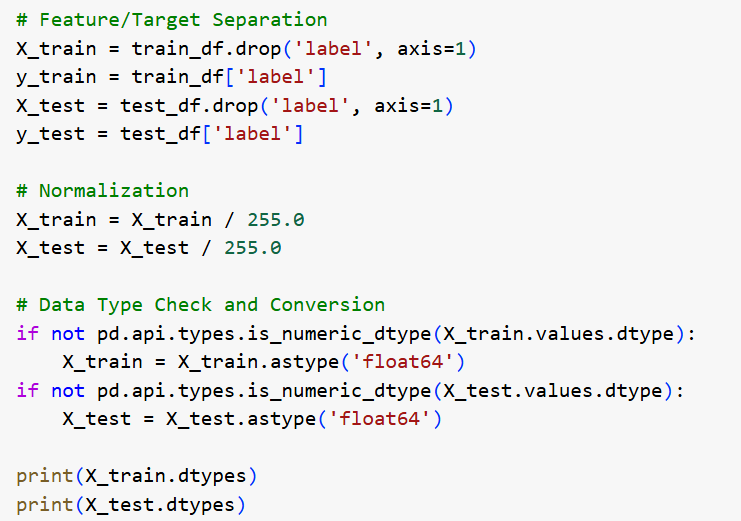


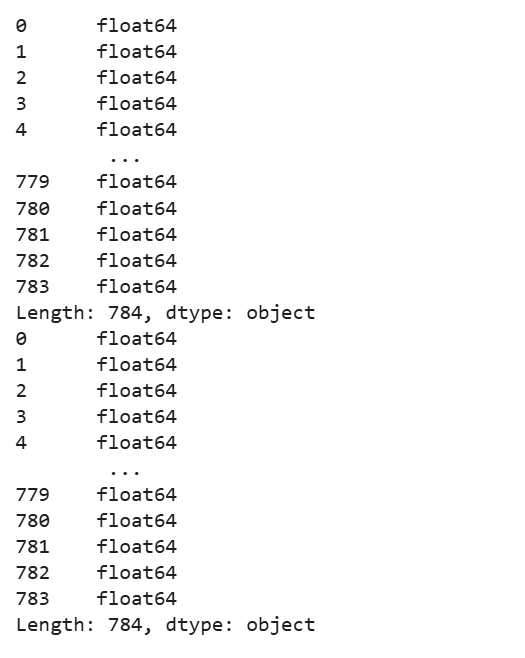




# **Data preparation**

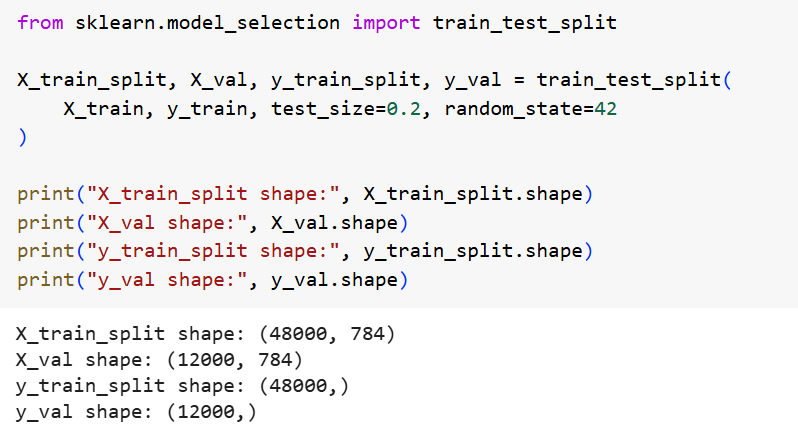
Prepare the data for model training by separating features and target variables, and normalizing pixel values.





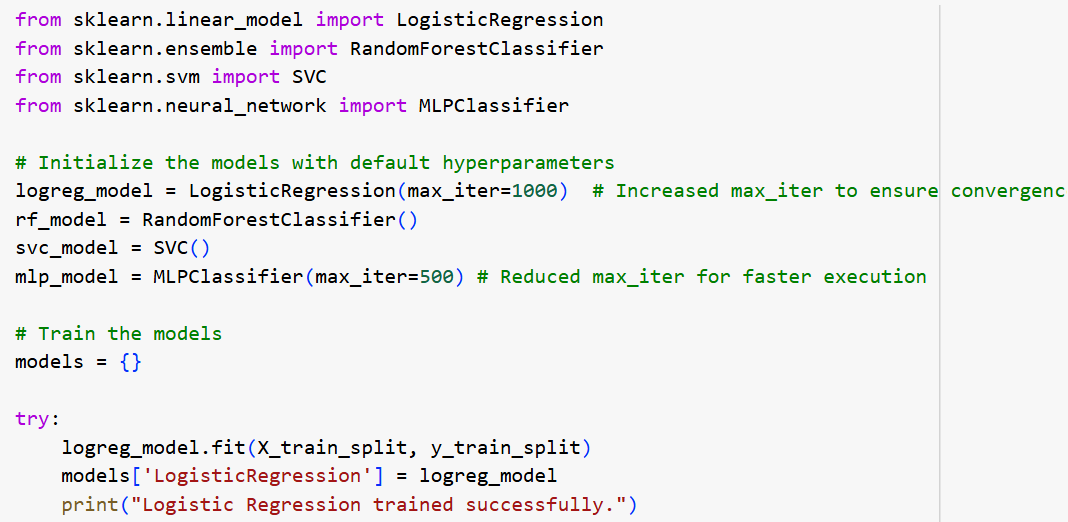
# **Data splitting**

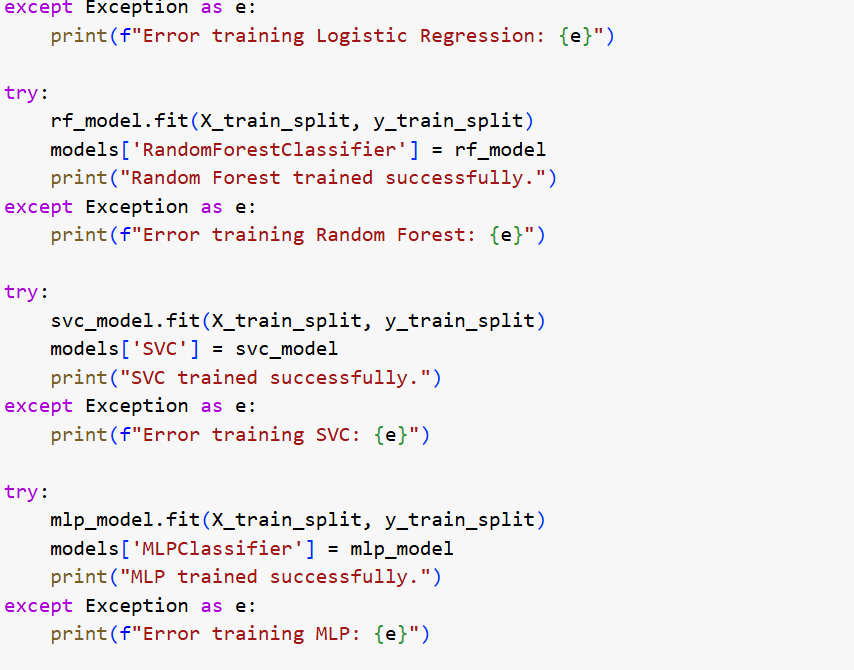
Split the training data (X\_train, y\_train) into training and validation sets.

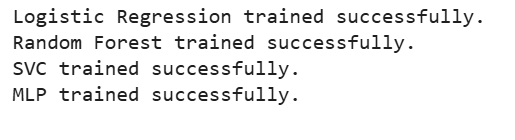


# **Model training**

Import necessary libraries and train the specified classification models using the provided training data.

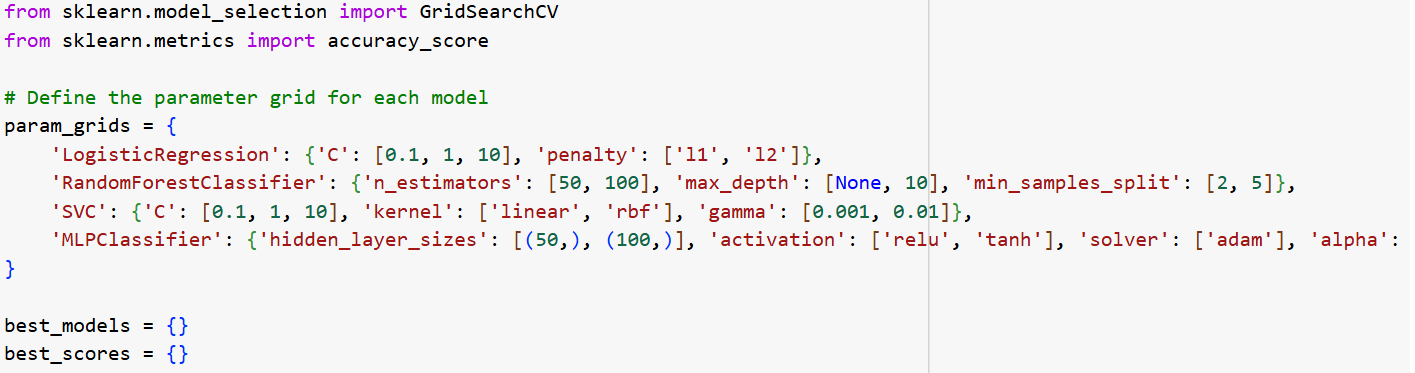


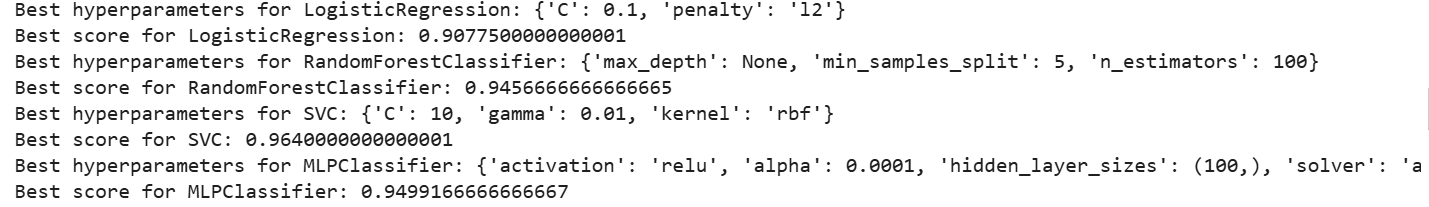
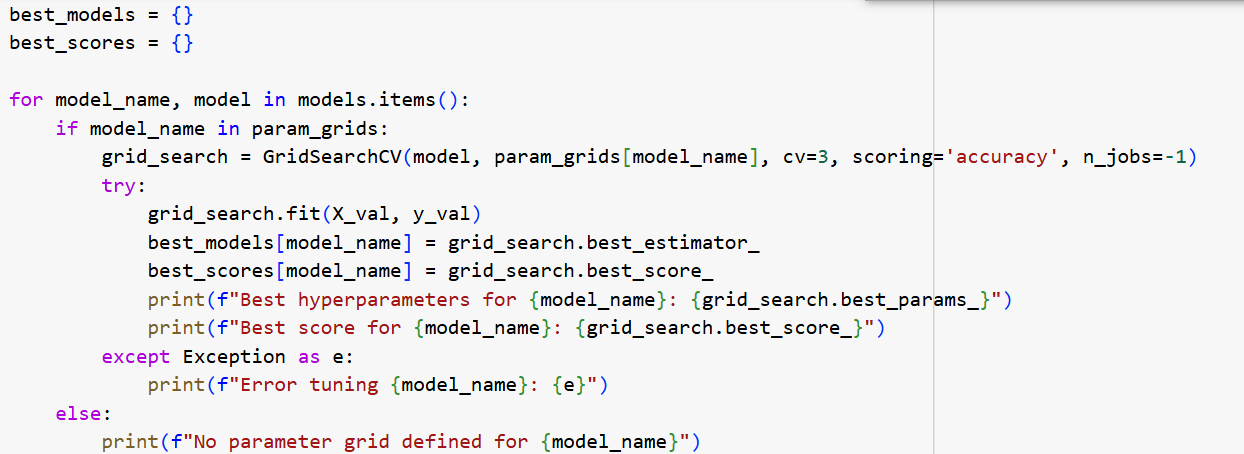




# **Model optimization**

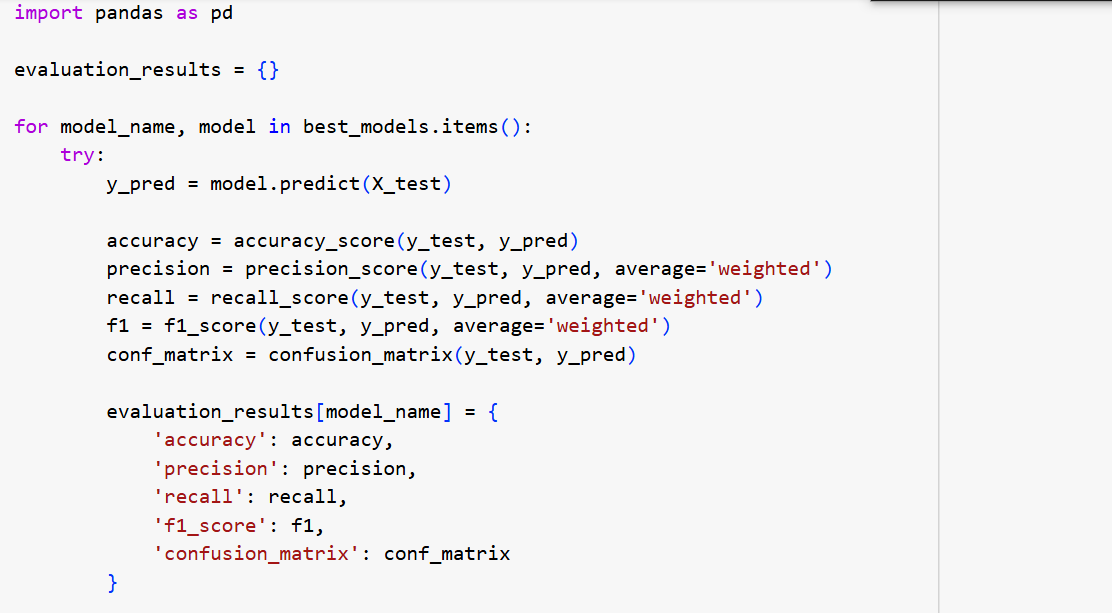
Optimize the hyperparameters of the trained classification models using the validation set.

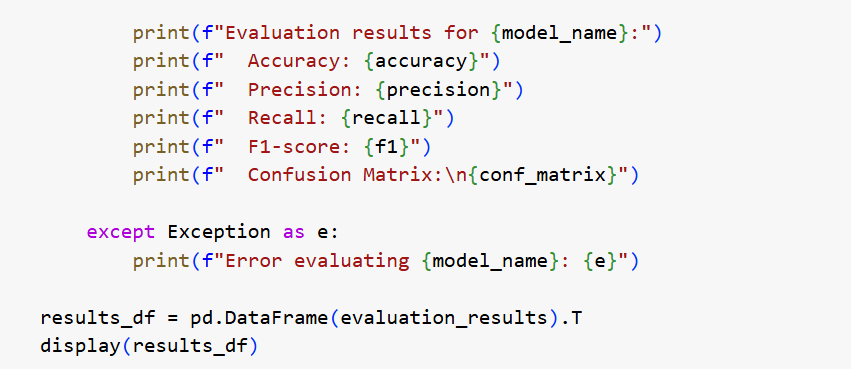


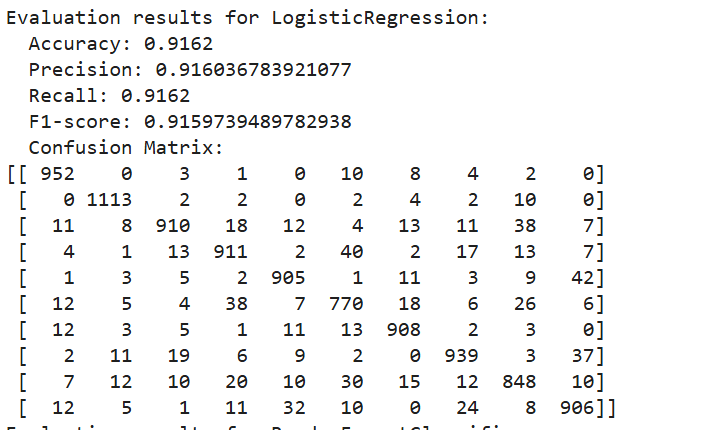


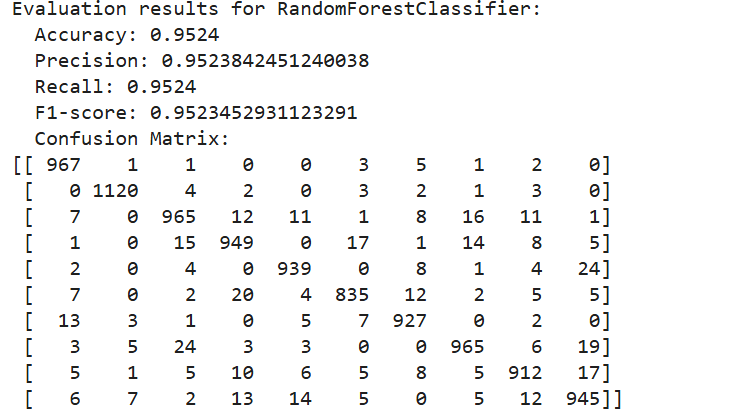
# **Model evaluation**

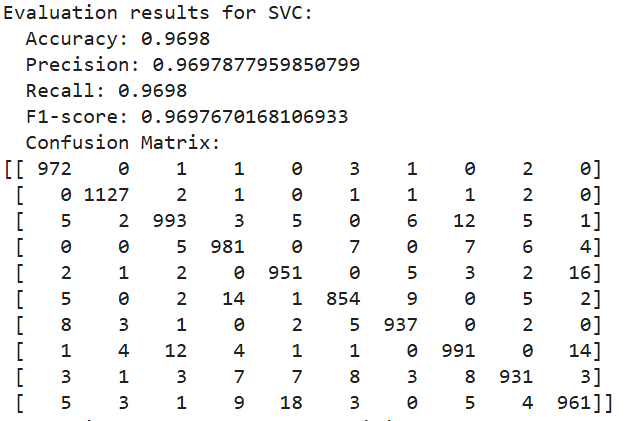
Evaluate the performance of the optimized models on the test set (X\_test, y\_test)

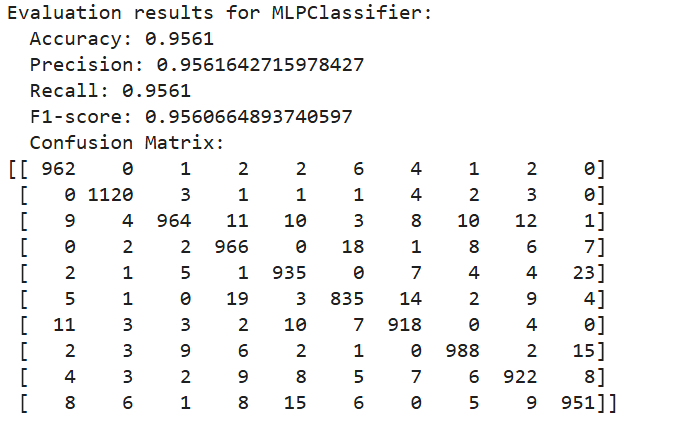


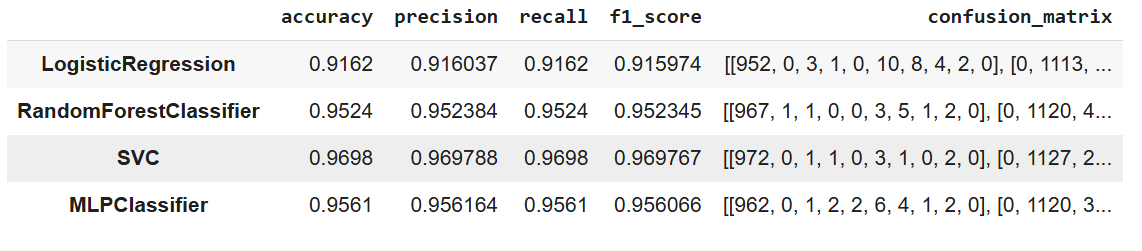








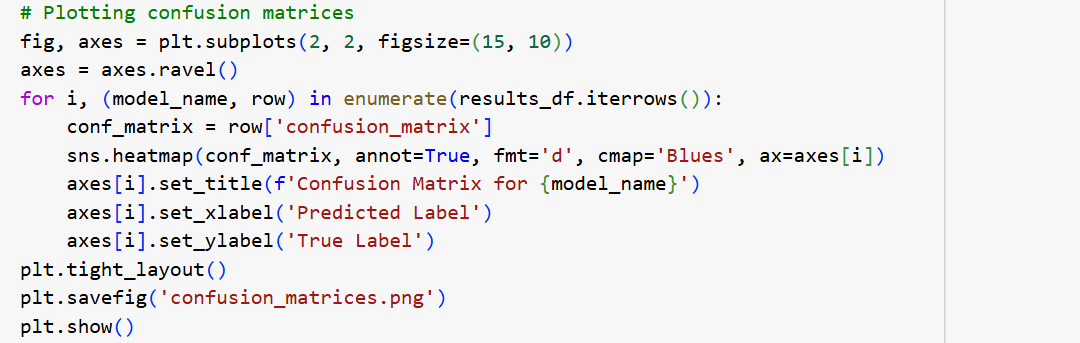




# **Data visualization**

Visualize the performance metrics and confusion matrices of the trained models using bar charts and heatmaps.





## **Summary:**

## **Data Analysis Key Findings**

* **Data characteristics:** The MNIST dataset consists of 60,000 training samples and 10,000 testing samples, each with 784 pixel features and a corresponding digit label (0-9). The distribution of digits is relatively balanced in both datasets. No missing values were found.
* **Model Performance (Test Set):**
  + **SVC:** Achieved the highest accuracy of 0.9698, along with highest precision, recall and F1-score.
  + **RandomForestClassifier:** Accuracy of 0.9524.
  + **MLPClassifier:** Accuracy of 0.9561.
  + **LogisticRegression:** Accuracy of 0.9162.
* **Hyperparameter Tuning Impact:** Hyperparameter tuning using GridSearchCV led to improved performance for all models, most notably the SVC. The best hyperparameters found for each model can be found in the detailed results.

## **Insights or Next Steps**

* **SVC as the top performer:** The SVC model consistently outperformed others, suggesting its suitability for image classification tasks with MNIST-like data.
* **Further exploration of MLP:** While MLP performed well, further investigation into its hyperparameter space or exploring deeper network architectures could potentially lead to higher accuracy and better understanding of the model.