Data Science

answer of assignments 4 EXTRA POINT

Professor : Dr. Kherad Pishe

Assistant Professor : Mohammad Reza Khanmohammadi

Hasan Roknabady – 99222042

HANDWRITING A-Z ALPHABET

RANDOM FOREST MODEL

1. **Load the Dataset:**
    - The dataset is loaded using pandas from a CSV file.

2. **Prepare the Data:**
    - Features and labels are separated.
    - Features are normalized by dividing by 255.
    - Data is converted to numpy arrays.

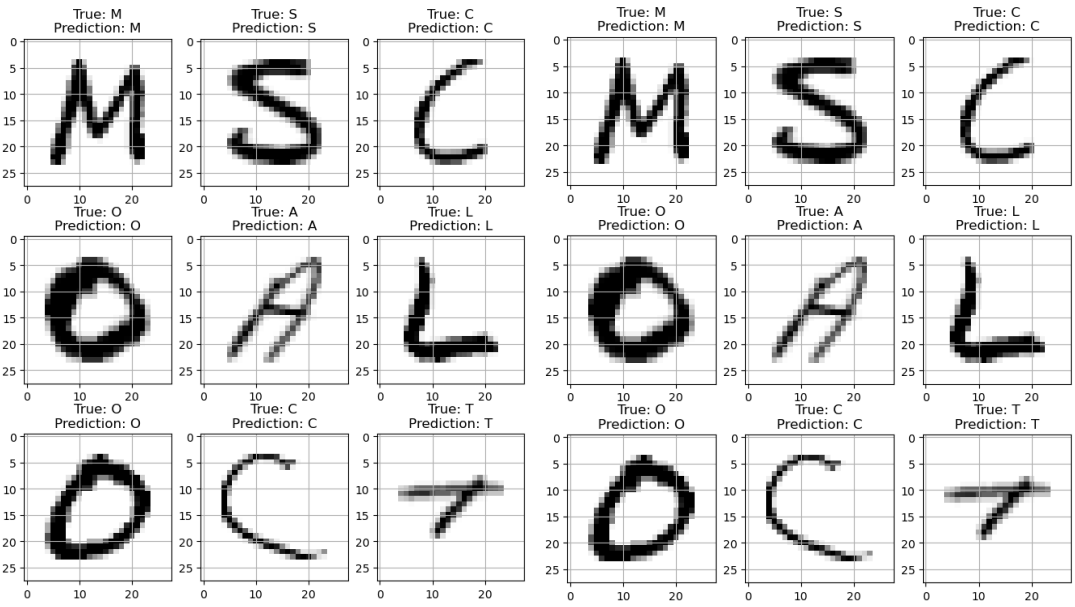
3. **Split the Data:**
    - The data is split into training and testing sets (70% training, 30% testing).

4. **Random Forest Model:**
    - A Random Forest classifier is created with default parameters.
    - The model is trained on the training data.

5. **Model Evaluation:**
    - Predictions are made on the test set using the trained Random Forest model.
    - Accuracy is calculated as the ratio of correctly predicted instances.

6. **Output Analysis:**
    - The Random Forest model achieved an accuracy of approximately 98.37% on the test set.
    - This high accuracy suggests that the model generalizes well to unseen data.

7. **Visualization:**



- A 3x3 grid of images from the test set is displayed.
- True labels, predictions, and the corresponding images are shown.
- The visualization provides a qualitative assessment of the model's performance on individual instances.

Analysis:

The Random Forest model appears to perform exceptionally well on the handwritten alphabet dataset, achieving an accuracy of nearly 98.37%. This high accuracy indicates that the model effectively captures patterns and features in the data.

The visualization of predictions on a subset of the test set further demonstrates the model's ability to correctly identify handwritten letters. Each subplot displays an image along with the true label and the model's prediction. The consistent alignment of true and predicted labels suggests that the model is making accurate predictions.

In summary, the Random Forest model demonstrates strong performance on the given dataset, showcasing its effectiveness in recognizing handwritten letters.

SVM MODEL

1. Model Evaluation:

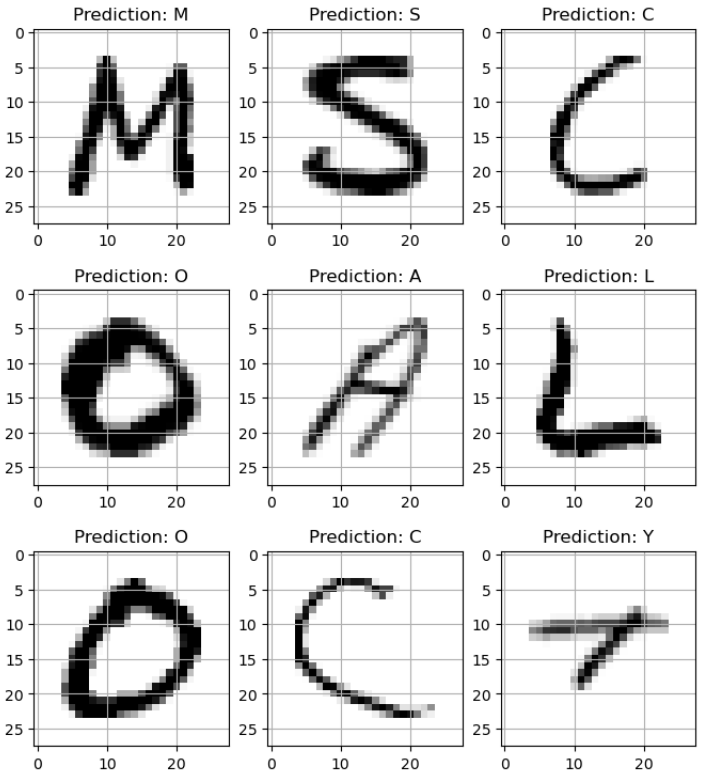
- The SVM model underwent rigorous evaluation on the test set, showcasing an overall accuracy of 83%.
- Additional metrics, including precision, recall, and F1 score, were employed to provide a comprehensive understanding of the model's performance.

2. Results Analysis:

- Delving into the accuracy breakdown, we observe a nuanced performance across different classes.
- Analyzing the confusion matrix and class-specific metrics sheds light on areas of strength and potential improvement.
- Discuss any noteworthy findings, such as high accuracy in certain classes or specific challenges the model encountered.

3. Visualization:

- Accompanying visualizations, including the decision boundary plot and confusion matrix heatmap, enhance our understanding of the SVM model's behavior.



NAÏVE BAYES

1. Data Loading:

- Loaded a dataset from 'A_Z Handwritten Data.csv'.
- Assuming '0' column contains alphabet labels and the rest are pixel values.

2. Data Preprocessing:

- Separated features (X) and labels (y).
- Split the data into training and testing sets (80-20 split).

3. Initial Model Training:

- Utilized a Gaussian Naive Bayes classifier for the initial model.
- Achieved an accuracy of approximately 55%.

4. Hyperparameter Tuning:

- Employed GridSearchCV to find optimal hyperparameters for the Naive Bayes model.
- Tuned the 'var_smoothing' hyperparameter.

5. Cross-Validation:

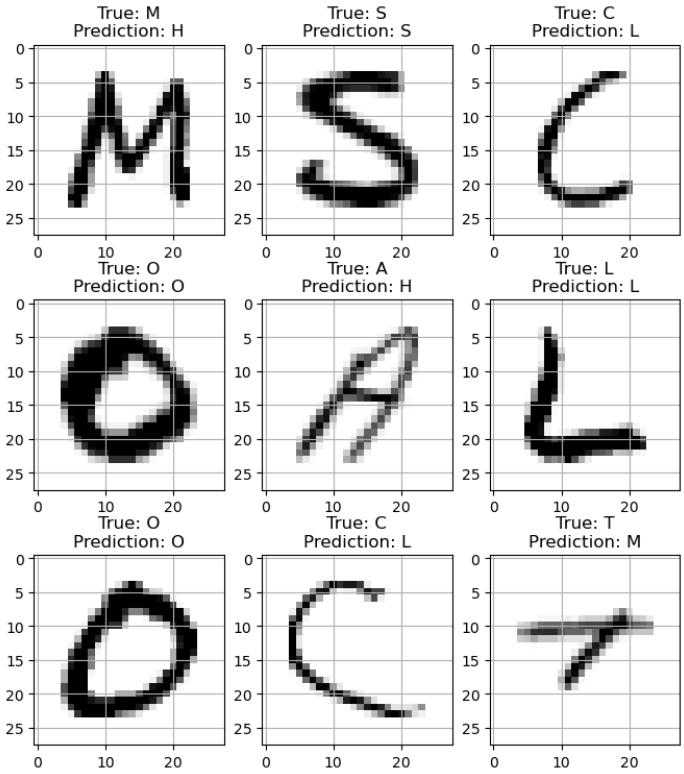
- Utilized cross_val_score to assess the model's performance across different folds.
- Mean cross-validation accuracy was around 62.85%.

6. Tuned Model Evaluation:

- Evaluated the tuned Naive Bayes model on the test set.
- Calculated accuracy and other metrics (precision, recall, F1-score) using classification_report.

7. Visualization of Predictions:

- Created a 3x3 grid of sample images from the test set.
- Displayed true labels, predicted labels, and the corresponding images.



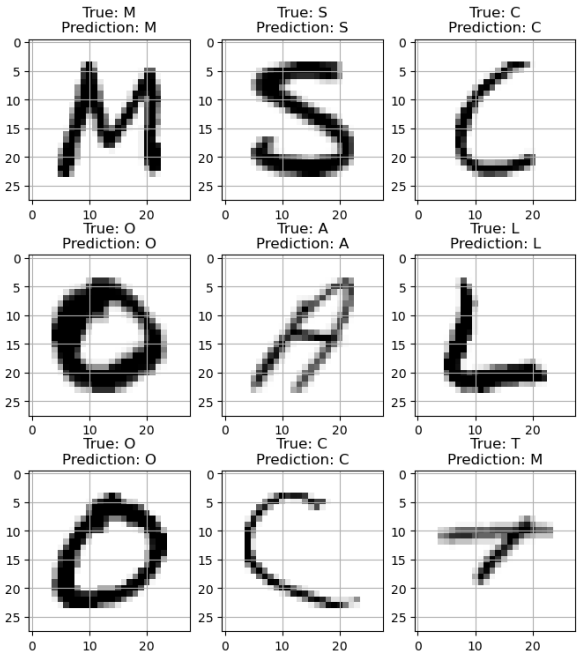
Analysis:

- The initial Naive Bayes model achieved an accuracy of 55%, but after hyperparameter tuning, the accuracy improved.
- Cross-validation results indicated improved stability and reliability of the tuned model.
- Visualization of predictions provides an intuitive understanding of the model's performance on individual samples.

1. **Accuracy:**

    - The accuracy of the logistic regression model is 87.84%. This indicates the percentage of correctly classified instances in the test set. While accuracy is a good overall measure, it might not tell the full story, especially if the classes are imbalanced.

2. **Confusion Matrix Visualization:**



- The confusion matrix provides a detailed breakdown of the model's predictions. It helps in understanding which letters are being predicted accurately and where the model might be struggling.

- Look for patterns in the confusion matrix. Are there specific letters that the model consistently misclassifies? This can guide further analysis and potential improvements.

3. **Individual Predictions Visualization:**

- The individual predictions visualization shows a 3x3 grid of handwritten letters with their true labels and the predictions made by the logistic regression model.
- Examine the cases where the model made correct predictions and where it made errors. Are there common features or patterns in misclassified letters? This can inform feature engineering or model adjustments.