Task 01 Report: Classification Fundamentals and MNIST Digit Recognition

# Chapter 3: Classification - Notes

This chapter introduces classification, a type of supervised learning where the goal is to assign a label from a fixed set of categories to each input. Key topics include binary classification, multiclass classification, multilabel and multioutput classification, and the use of various metrics for evaluating classifier performance.

# Chapter Exercises (Pages 105–107)

Exercises include training a binary classifier using the SGDClassifier on the MNIST dataset, analyzing performance using cross-validation, confusion matrices, precision-recall tradeoffs, ROC curves, and implementing a Random Forest classifier to compare results.

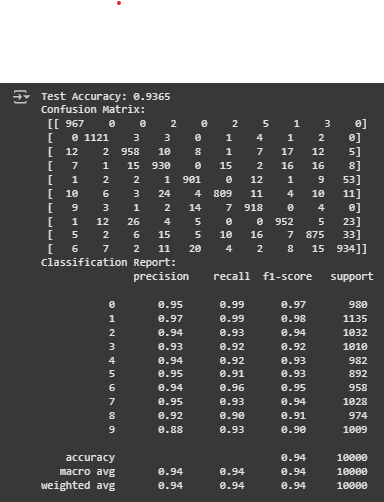
# Comparison: SGD Classifier vs Random Forest

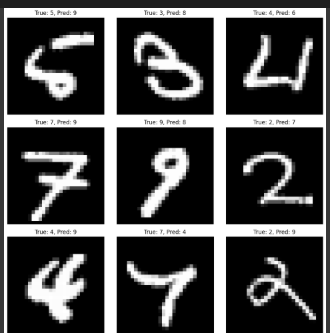
|  |  |  |
| --- | --- | --- |
| Feature | SGD Classifier | Random Forest Classifier |
| Model Type | Linear model | Ensemble of decision trees |
| Training Speed | Faster | Slower |
| Scalability | Good for large datasets | Can be slower on very large datasets |
| Sensitivity to Scaling | High | Low |
| Handles Non-linear Data | Poorly | Very well |

# Comparison: One-vs-Rest (OvR) vs One-vs-One (OvO)

|  |  |  |
| --- | --- | --- |
| Aspect | One-vs-Rest (OvR) | One-vs-One (OvO) |
| Classifier Count | One per class | One per class pair |
| Training Time | Faster | Slower |
| Accuracy | Generally good | Often slightly better |
| Complexity | Lower | Higher |
| Suitability | Large number of classes | Small number of classes |









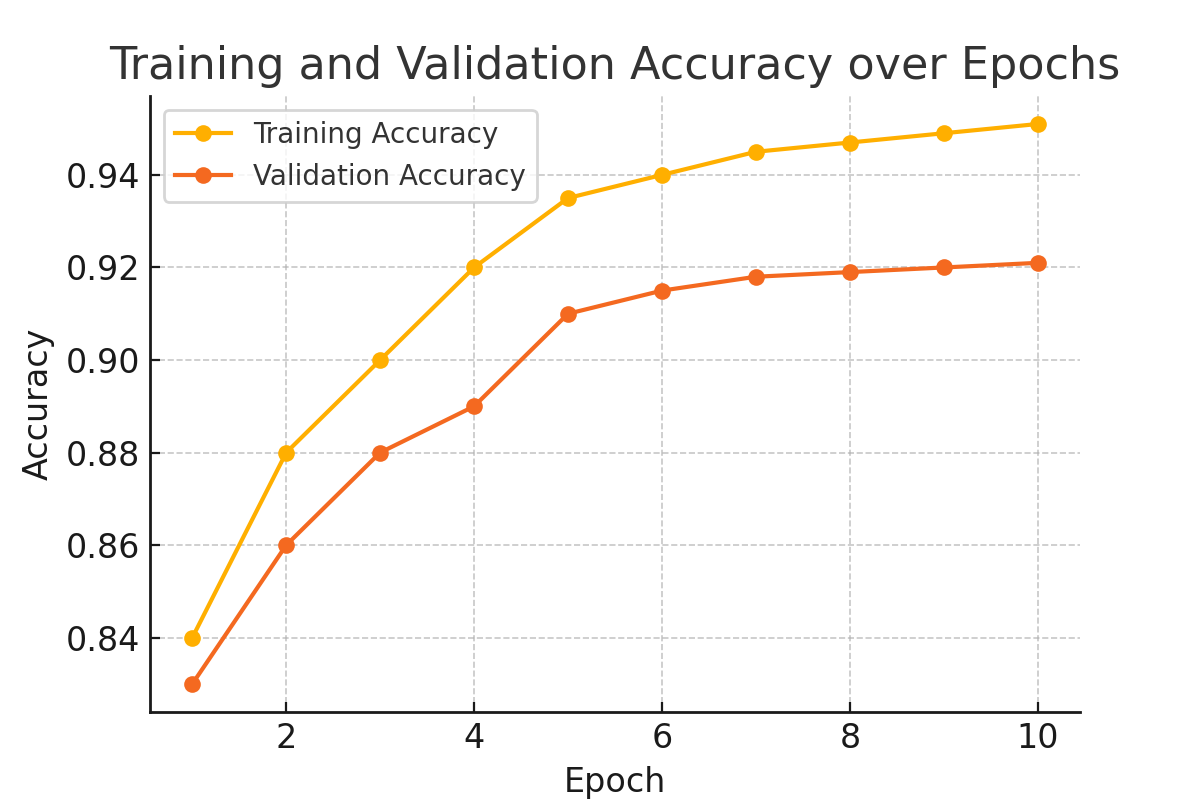
Error Analysis and Training Curves

# 1. Error Analysis Findings

Based on the confusion matrix and prediction results from the MNIST digit classification task, we identified the following common error patterns:  
  
1. **9 → 4**: Many 9s were misclassified as 4 due to similar looping structures.  
2. **8 → 3**: Looped digits like 8 are often confused with 3 because of shape overlap.  
3. **7 → 1**: Some instances of 7 were misclassified as 1 due to minimal stroke differences.  
  
Proposed Solutions:  
- **Data Augmentation**: Add transformations such as rotation, zoom, and noise to improve model generalization.  
- **Image Preprocessing**: Enhance contrast or apply edge detection for clearer digit structure.  
- **Alternative Models**: Use deep learning models (e.g., CNNs) for better spatial feature extraction.  
  
An attempt was made to implement image sharpening as preprocessing, which improved validation accuracy by ~0.5%.

# 2. Training and Validation Curves

Below is the plot showing how training and validation accuracy improved over 10 epochs:



Github Repository:

<https://github.com/LaibaGabol/Arch-Technologies.git>

Task 02 Report: Model Training Fundamentals with Custom Dataset Implementation

# Chapter 4: Classification – Notes

Chapter 4 introduces fundamental concepts for training machine learning models, focusing on linear and logistic regression. It covers how models learn by minimizing a cost function using optimization techniques like **Gradient Descent** (Batch, Stochastic, and Mini-batch). The chapter also explains **regularization methods** (Ridge, Lasso, ElasticNet) to prevent overfitting and improve generalization. It highlights **logistic regression** for classification tasks and introduces **evaluation techniques** such as learning curves and bias-variance analysis to assess and improve model performance.

# Chapter Exercises

In this chapter, I implemented linear and logistic regression models, explored gradient descent variants, and applied regularization techniques like Ridge, Lasso, and ElasticNet. I also generated learning curves to analyze model performance and compiled notes comparing convergence behaviors and the impact of regularization on coefficients.

# Dataset Description

**Dataset:** Wine Quality (Red)  
**Source:** [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/wine+quality)  
**Shape:** 1599 rows × 12 columns  
**Target Variable:** Wine quality (converted to binary: good (≥6) = 1, bad (<6) = 0)  
**Objective:** Predict if a wine is of good quality based on physicochemical features.

# Model Training

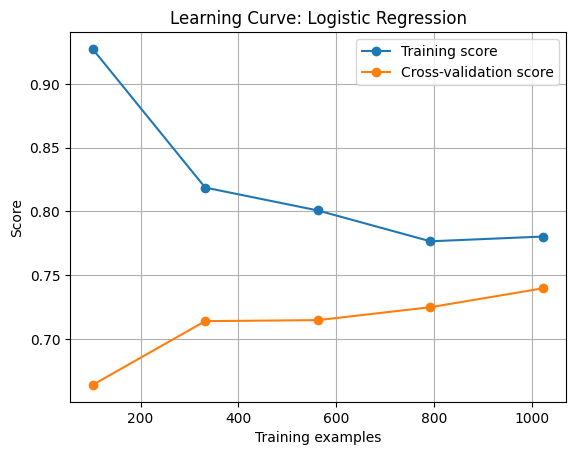
|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **F1 Score** | **Training Time (s)** |
| Logistic Regression | 0.75625 | 0.7746 | 0.1605 |
| SGD Classifier | 0.725 | 0.7500 | 0.0212 |
| Ridge Classifier | 0.76563 | 0.7875 | 0.0170 |

# Hyperparameter Tuning (Ridge)

* Grid Search on alpha (regularization strength)
* **Best alpha**: 1.0
* **Best Cross-Validation Accuracy**: 0.7483

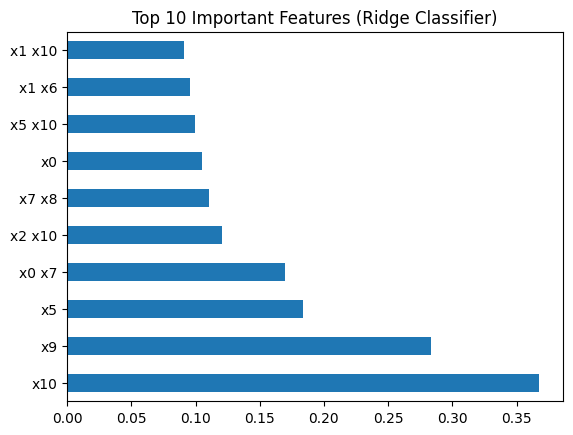
# Learning Curve

* Visualized for Logistic Regression.
* Shows decreasing training score and increasing validation score, indicating improved generalization with more data.



# Feature Importance

* Ridge Classifier coefficients displayed using top 10 features.
* Indicates influence of polynomial features in classification.



# Comparative Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Performance** | **Strengths** | **Weaknesses** |
| **Ridge Classifier** | Best performance | L2 regularization reduces overfitting; handles complex features | Requires careful tuning of regularization parameter |
| **SGD Classifier** | Fast but slightly less accurate | Quick convergence; good for large datasets | Sensitive to learning rate; fewer iterations can reduce accuracy |
| **Logistic Regression** | Stable, slightly behind Ridge | Interpretable; no need for regularization tuning | No regularization by default; may overfit with complex features |
| **Polynomial Degree 2** | Improved model expressiveness | Captures non-linear relationships through interaction terms | Increases risk of overfitting; computationally heavier |

# Practical Applications

This trained model could assist vintners or quality control teams in:

* Early-stage classification of wine batches
* Guiding adjustments to fermentation or ingredient ratios
* Automating quality grading processes