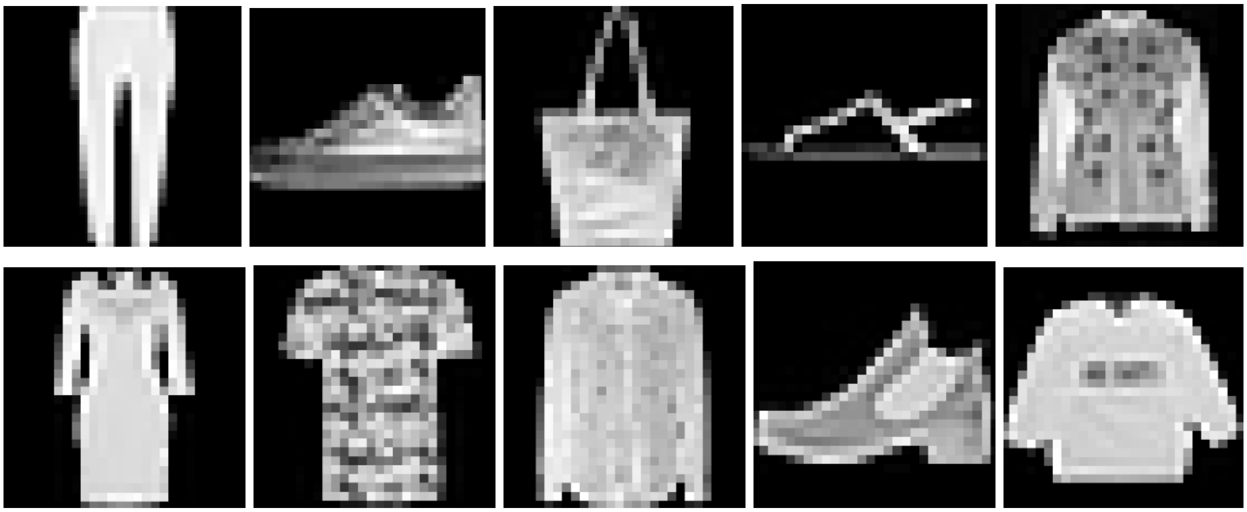
Case Study 1
Fashion MNIST Classification

Joshua Richardson

Introduction:
This lab uses a dataset of 28x28 black and white images of clothes. Each type of clothing is put into a class (0-9). The goal is to train a machine learning algorithm to learn to recognize which class a given article of clothing belongs to. Part 1 of this lab uses binary classification, where only class 0 and 1 are used. Part 2 uses the entire dataset.

Analyzing the dataset:
Each type of clothing from the data can be seen here.



Data Preprocessing:
To preprocess the data, several steps are taken. Firstly, the outline of each image is removed, shrinking the size from (28,28) to (26,26). After that, the data is scaled using the standard scaler.

For binary classification, all images of clothes that are not class 0 or 1 are dropped from the training and test data sets. Four different algorithms are used, and five performance evaluation metrics are used.

Binary Classification:

Logistic Regression:
Hyperparameter set 1:
Solver: lbfgs
Max iterations: 10,000

Average score: 98.4%

Evaluation	Score (%)
Accuracy Score	98.4
Precision Score	98.0
Recall Score	98.8
F1 Score	98.4
ROC-AUC Score	98.4

Hyperparameter set 2:
Solver: lbfgs
Max iterations: 100,000

Average Score: 98.4%
Increasing the max iterations did not yield any different results.

Evaluation	Score (%)
Accuracy Score	98.4
Precision Score	98.0
Recall Score	98.8
F1 Score	98.4
ROC-AUC Score	98.4

Hyperparameter set 3:
Solver: sag
Max iterations: 10,000

Average Score: 98.4%
Despite the solver taking 8 minutes to finish, the results were still the same.

Evaluation	Score (%)
Accuracy Score	98.4
Precision Score	98.0
Recall Score	98.8
F1 Score	98.4
ROC-AUC Score	98.4

Evaluation	Score (%)
Accuracy Score	98.3

Hyperparameter set 4:
Solver: liblinear
Max iterations: 10,000

Average score: 98.3%

Precision Score	97.7
Recall Score	98.9
F1 Score	98.3
ROC-AUC Score	98.3

Hyperparameter set 5:
Solver: newton-cholesky
Max iterations: 10,000

Average Score: 97.8%

Evaluation	Score (%)
Accuracy Score	97.9
Precision Score	97.1
Recall Score	98.6
F1 Score	97.8
ROC-AUC Score	97.8

The scores remained similar through all the sets. Sag was an outlier in terms of time, as it took 8 minutes to complete, while all the others took under a minute.

Since all scores are similar, and due to time constraints, only accuracy scores will be used going forward.

Nearest Neighbor:
Hyperparameter set 1:
N Neighbors: 2
Weights: uniform

Score: 99.15%

Hyperparameter set 2:
N Neighbors: 3
Weights: Uniform

Score: 99.2%

Hyperparameter set 3:
N Neighbors: 3
Weights: Distance

Score: 99.2%

Hyperparameter set 4:
N Neighbors: 3
Weights: Uniform
Algorithm: ball_tree

Score: 99.2%

Hyperparameter set 5:
N Neighbors: 3
Weights: Uniform
Algorithm: kd_tree

Score: 99.2%

The most important hyperparameter seems to be the number of neighbors used.

Support Vector Machines:
Hyperparameter set 1:
Kernel: rbf
C: 1.0

Score: 99.35%

Hyperparameter set 2:
Kernel: rbf
C: 2.0

Score: 99.55%

Hyperparameter set 3
Kernel: linear
C: 1.0

Score: 98.2%

Hyperparameter set 4:
Kernel: poly
C: 1.0

Score: 99.2%

Hyperparameter set 5:
Kernel: sigmoid
C: 1.0

Score: 69.45%

Sigmoid kernel hyperparameter severely underperformed compared to the rest of the sets. All of the other hyperparameter sets scored near 100% with little variance between them.

Decision Tree:
Hyperparameter set 1:
Criterion: gini
Splitter: best
Max_depth: inf

Score: 97.75%

Hyperparameter set 2:
Criterion: entropy
Splitter: best
Max_depth: inf

Score: 98.65%

Hyperparameter set 3:
Criterion: gini

Splitter: random
Max_depth: inf

Score: 98.55%

Hyperparameter set 4:
Criterion: gini
Splitter: best
Max_depth: 1

Score: 92.25%

Hyperparameter set 5:
Criterion: gini
Splitter: best
Max_depth: 2

Score: 96.7%

The maximum depth reach seemed to make the most impact, as was expected.

Overall, nearest neighbor had the highest single score, as well as the highest average score among all 5 hyperparameter sets.

Multi-Class Classification:

Logistic Regression:
Hyperparameter set 1:
Solver: lbfgs
Max iterations: 10,000

Score: 72.6%

Hyperparameter set 2:
Solver: newton-cholesky
Max iterations: 10,000

Score: 84.6%

Hyperparameter set 3:
Solver: newton-cg
Max iterations: 10,000
Multi_class: multinomial

Score: 73.26%

Nearest Neighbor:
Hyperparameter set 1:
N Neighbors: 2
Weights: uniform

Score: 31.17%

Hyperparameter set 2:
N Neighbors: 3
Weights: Uniform

Score: 31.79%

Hyperparameter set 3:
N Neighbors: 3
Weights: Distance

Score: 31.77%

Hyperparameter set 4:
N Neighbors: 3
Weights: Uniform
Algorithm: ball_tree

Score: 31.79%

Hyperparameter set 5:
N Neighbors: 3
Weights: Uniform
Algorithm: kd_tree

Score: 31.77%

Support Vector Machines:
Hyperparameter set 1:
Kernel: rbf
C: 1.0

Score: 10%

Hyperparameter set 2:
Kernel: rbf
C: 2.0

Score: 10%

Hyperparameter set 3
Kernel: linear
C: 1.0

Score: 71.9%

Hyperparameter set 4:
Kernel: poly
C: 1.0

Score: 65.17%

Hyperparameter set 5:
Kernel: sigmoid
C: 1.0

Score: 12.9%

Decision Tree:
Hyperparameter set 1:
Criterion: gini
Splitter: best
Max_depth: inf

Score:

Hyperparameter set 2:
Criterion: entropy
Splitter: best
Max_depth: inf

Score: 10.89%

Hyperparameter set 3:
Criterion: gini
Splitter: random
Max_depth: inf

Score: 11.39%

Hyperparameter set 4:
Criterion: gini
Splitter: best
Max_depth: 1

Score: 10%

Hyperparameter set 5:
Criterion: gini
Splitter: best
Max_depth: 2

Score: 10%

Conclusion:

The scores for multi-class classification were significantly lower than the binary classification. The lowest score for multi-class classification was 10%, and the lowest for binary classification was 69.45%. The highest from each were also quite different, Multi-class only scored a high of 84.6%. According to the data, the best algorithm for each type of classification is nearest neighbor for binary classification, and logistic regression for multi-class classification.