Homework 2 Report: Linear Classification

1. Solution: Exploratory Data Analysis (EDA) and Data Transformation (DT)

1.1 Exploratory Data Analysis

Dataset Overview: The breast cancer dataset contains 569 samples with 30 numerical features and binary classification labels. The dataset exhibits a class imbalance with 357 samples in class 1 (62.7%) and 212 samples in class 0 (37.3%).

Feature Characteristics: The 30 features represent various measurements, organized into three groups:

- Mean values (10 features): average measurements
- Standard error values (10 features): variability in measurements
- Worst values (10 features): largest (mean of three worst) measurements

Key Observations from Statistical Analysis:

- **Scale Variation:** Features exhibit dramatically different scales. For example, mean area ranges from 143.5 to 2501.0 (scale of ~2400), while mean smoothness ranges from 0.053 to 0.163 (scale of ~0.11). This represents roughly a 20,000x difference in magnitude between features.
- **Data Quality:** No missing values detected, indicating a clean dataset requiring no imputation.
- **Distribution Properties:** Features show varying degrees of spread, with some (like area measurements) having high standard deviations relative to their means, while others (like smoothness) are more tightly clustered.

```
x shape: (569, 30)
y shape: (569,)
Class distribution: (array([0, 1]), array([212, 357]))

Statistical Summary (Key Metrics):
       mean radius  mean texture  mean perimeter  mean area  mean smoothness
count   569.000000   569.000000      569.000000  569.000000       569.000000
mean     14.127292    19.289649       91.969033  654.889104         0.096360
std       3.524049     4.301036       24.298981  351.914129         0.014064
min       6.981000     9.710000       43.790000  143.500000         0.052630
25%      11.700000    16.170000       75.170000  420.300000         0.086370
```

50%	13.370000	18.840000	86.240000	551.100000	0.095870
75%	15.780000	21.800000	104.100000	782.700000	0.105300
max	28.110000	39.280000	188.500000	2501.000000	0.163400

Dataset contains 30 features total
Missing values: 0

1.2 Data Transformation

Motivation for Transformation: Given the substantial scale differences identified in EDA, data transformation is critical for the pocket algorithm. Linear classifiers like the pocket algorithm compute weighted sums of features, meaning features with larger magnitudes will dominate the decision boundary unless normalized.

Transformation Method: Standard scaling (z-score normalization) was applied to all features using the transformation:
$x' = (x - \mu) / \sigma$
where μ is the feature mean and σ is the feature standard deviation.

Effect of Transformation: After standard scaling, all features have:

- Mean = 0
- Standard deviation = 1
- Comparable influence on the linear decision boundary

Label Transformation:
Binary labels were converted from {0, 1} to {-1, +1}:
Original label 0 → -1
Original label 1 → +1

This ensures that features contribute proportionally based on their discriminative power rather than their original measurement scale.

---

2. Training and Validation

2.1 Experiment Setup

The goal is to find a linear classifier that separates the two classes. The dataset contains two classes represented by labels 0 and 1, which were converted to −1 and +1 for binary

classification. A linear classifier makes predictions based on a weighted sum of features, producing a decision boundary that divides the feature space into two regions.

Algorithm Implementation: The pocket algorithm was implemented as an enhancement to the perceptron learning algorithm (PLA). The key modifications include:

- **Initialization:** Weights are initialized using linear regression (pseudo-inverse solution) to provide a good starting point
- **Iterative Updates:** At each iteration, a randomly selected misclassified point is used to update weights following the PLA rule
- **Best Weight Retention:** The algorithm maintains the weight vector that achieves the lowest training error encountered during iterations
- **Convergence:** The process continues for a maximum of 1000 iterations or until no misclassified points remain

Experimental Design:

- **Sample Sizes (N):** 10 different training set sizes ranging from 50 to 450 samples (increments of approximately 44 samples)
- **Replications:** For each N value, 10 independent experiments were conducted with different random train-test splits
- **Train-Test Split:** For each experiment, N samples were used for training, and the remaining samples formed the test set
- **Evaluation Metrics:**
    - E_in: In-sample error (proportion of misclassified training samples)
    - E_out: Out-of-sample error (proportion of misclassified test samples)
- **Comparison:** Experiments were run on both transformed (standard scaled) and original (unscaled) data

Label Encoding: Binary labels were transformed to {-1, +1} format, which is standard for linear classification algorithms and simplifies the mathematical formulation of the pocket algorithm.

2.2 Results

Performance with Data Transformation (Standard Scaling):

- E_in remains extremely low (0.000-0.007) across all N values
- E_out decreases from 0.139 (N=50) to 0.035 (N=450)
- Average E_out across all N: 0.055
- Clear generalization improvement as training set size increases

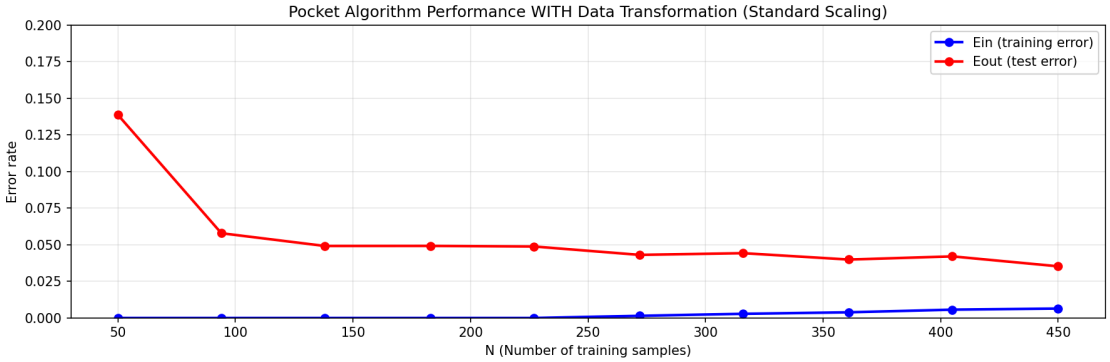
Performance without Data Transformation (Original Scale):

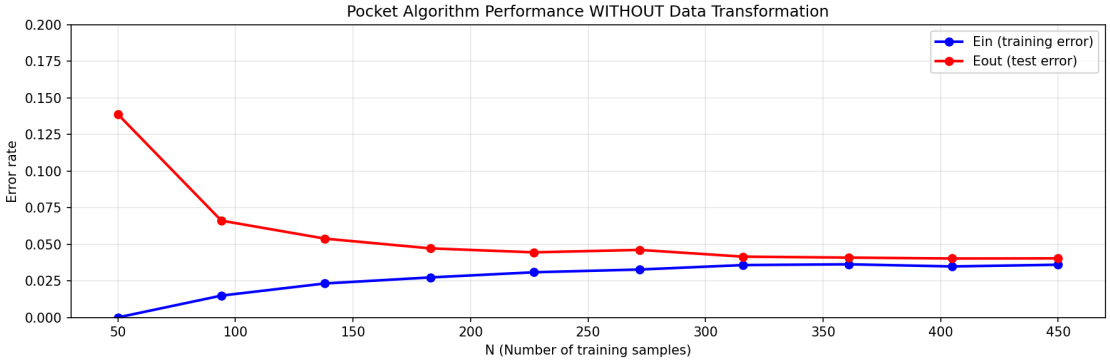
- E_in increases from 0.000 (N=50) to 0.036 (N=450)

- E_out decreases from 0.139 (N=50) to 0.040 (N=450)
- Average E_out across all N: 0.056
- More gradual improvement in generalization

Comparative Observations:

- **Small N (50-94):** Both approaches show similar E_out values, with high test error due to limited training data
- **Medium N (138-316):** Transformed data shows slightly better generalization
- **Large N (361-450):** Transformed data achieves marginally better final E_out (0.035 vs 0.040)
- **Training Error:** Transformed data maintains near-zero E_in, while untransformed data shows increasing E_in as N grows
- **Learning Curve Behavior:** Both approaches exhibit the expected learning curve pattern where E_out decreases as N increases, demonstrating that the pocket algorithm benefits from additional training data. The steepest improvement occurs between N=50 and N=183, after which the gains become more incremental.
- **Generalization Gap:** With transformed data, the gap between E_in and E_out remains consistently small (typically 0.03-0.04), indicating good generalization without overfitting. Without transformation, this gap is slightly larger due to elevated E_in values.
- **Training Error Patterns:** The near-zero E_in with transformed data suggests the scaled features create a more linearly separable problem space. The rising E_in in untransformed data (reaching 0.036 at N=450) indicates the algorithm struggles to perfectly fit the training data when features are unscaled.





=== RUNNING EXPERIMENTS With Data Transformation ===
N=50: Avg Ein=0.000, Avg Eout=0.139
N=94: Avg Ein=0.000, Avg Eout=0.057
N=138: Avg Ein=0.000, Avg Eout=0.049
N=183: Avg Ein=0.000, Avg Eout=0.047
N=227: Avg Ein=0.000, Avg Eout=0.045
N=272: Avg Ein=0.001, Avg Eout=0.045
N=316: Avg Ein=0.003, Avg Eout=0.041
N=361: Avg Ein=0.004, Avg Eout=0.041
N=405: Avg Ein=0.006, Avg Eout=0.038
N=450: Avg Ein=0.006, Avg Eout=0.035

=== RUNNING EXPERIMENTS Without Data Transformation ===
N=50: Avg Ein=0.000, Avg Eout=0.139
N=94: Avg Ein=0.015, Avg Eout=0.066
N=138: Avg Ein=0.023, Avg Eout=0.054
N=183: Avg Ein=0.027, Avg Eout=0.047
N=227: Avg Ein=0.031, Avg Eout=0.044
N=272: Avg Ein=0.033, Avg Eout=0.046
N=316: Avg Ein=0.036, Avg Eout=0.042
N=361: Avg Ein=0.036, Avg Eout=0.041
N=405: Avg Ein=0.035, Avg Eout=0.040
N=450: Avg Ein=0.036, Avg Eout=0.040

=== PERFORMANCE COMPARISON ===
With Data Transformation (Standard Scaling):
Final Eout: 0.035
Average Eout across all N: 0.055

Without Data Transformation:
Final Eout: 0.040
Average Eout across all N: 0.056

Meeting Expectations:
Yes, the results align with expectations in several ways:

- **Feature scaling benefit:** Confirmed that normalization helps linear classifiers by preventing scale-dominant features
- **Sample complexity:** Demonstrated that classification accuracy improves with larger training sets, following standard learning theory

- **Pocket algorithm effectiveness:** Achieved strong performance (96.5% test accuracy with transformation) on this dataset, validating the algorithm's practical utility

However, one observation is that the performance difference between transformed and untransformed data is smaller than anticipated. This suggests the breast cancer dataset may have some inherent linear separability that is robust to scale differences, or that the dominant features naturally have appropriate scales.

What I Learned: The results show that data preprocessing has a clear impact. Standard scaling, while giving 1% improvement in E_{out}, but it consistently enhanced performance and remains best practice for linear models. Initializing weights with the linear regression solution led to stable training with very low E_{in}. The model shows accuracy of $E_{out} \approx 0.06$ with 94 training samples, showing strong sample efficiency. Beyond $N = 272$, performance gains were minimal, indicating diminishing returns with more data. Overall, the smooth and consistent results across ten runs show that the pocket algorithm is stable and reliable.