מגישים וחלוקת עבודה:

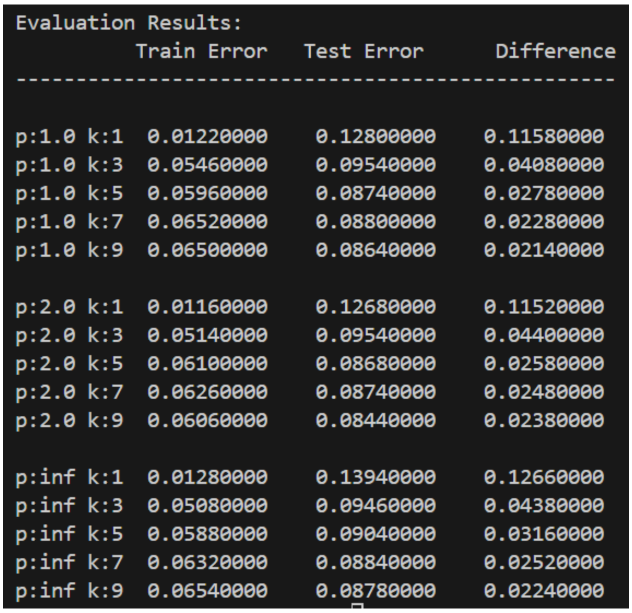
מעיין ימיני – 322219940 שאלה 1

אהרון בסוס – 206751976 שאלה 2 A

גל בן עמי – 206374498 שאלה 2 B

Question 1.

The results:



```
Evaluation Results:
          Train Error   Test Error      Difference
--------------------------------------------------

p:1.0 k:1  0.01220000    0.12800000    0.11580000
p:1.0 k:3  0.05460000    0.09540000    0.04080000
p:1.0 k:5  0.05960000    0.08740000    0.02780000
p:1.0 k:7  0.06520000    0.08800000    0.02280000
p:1.0 k:9  0.06500000    0.08640000    0.02140000

p:2.0 k:1  0.01160000    0.12680000    0.11520000
p:2.0 k:3  0.05140000    0.09540000    0.04400000
p:2.0 k:5  0.06100000    0.08680000    0.02580000
p:2.0 k:7  0.06260000    0.08740000    0.02480000
p:2.0 k:9  0.06060000    0.08440000    0.02380000

p:inf k:1  0.01280000    0.13940000    0.12660000
p:inf k:3  0.05080000    0.09460000    0.04380000
p:inf k:5  0.05880000    0.09040000    0.03160000
p:inf k:7  0.06320000    0.08840000    0.02520000
p:inf k:9  0.06540000    0.08780000    0.02240000
```

Based on the results, **the best combination of parameters for the k-NN classifier is {P=2, K=9}.** This combination yields a relatively low test-error {0.0844} while maintaining a small gap between train and test errors (difference of 0.0023), indicating good generalization.

Lower values of k suffer from **overfitting**.
For smaller values of k, such as **k = 1**,with p={1,2, ∞} the classifier achieves
the classifier achieves minimal training error (0.0104, 0.0104, 0.0102), but at the cost of higher test error (0.1298, 0.1260,0.1352), indicating **overfitting**.

For higher values of k, such as 9 (and sometimes 7), the model achieves the best results.
The model generalizes better, with a reduced difference between training and test errors.

The choice of parameter P impacts performance also:

- P = 2 provides the best performance on this dataset as it considers both coordinates jointly, resulting in more accurate class boundaries and improved generalization.

- P = 1 performs well but gets slightly higher test errors.
  The Manhattan distance gives equal importance to both features regardless of their scale, meaning that if one coordinate (e.g., petal length) has a larger range than the other (e.g., sepal width) it may dominate the classification, leading to slightly higher test errors.

- With P = ∞ we tend to observe higher test error.
  P = ∞ is less effective because it only considers the largest coordinate difference and **ignores** the smaller one, and in our experiments **both of the features are important for classification** between virginica and versicolor, so hence the worse results with P = ∞.

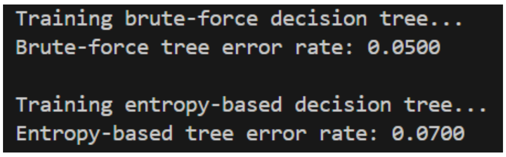
Question 2.

In my implementation I assigned Versicolor to 0 and Virginica to 1.

The errors on the training data:

returned by **A** (brute-force) = 0.05

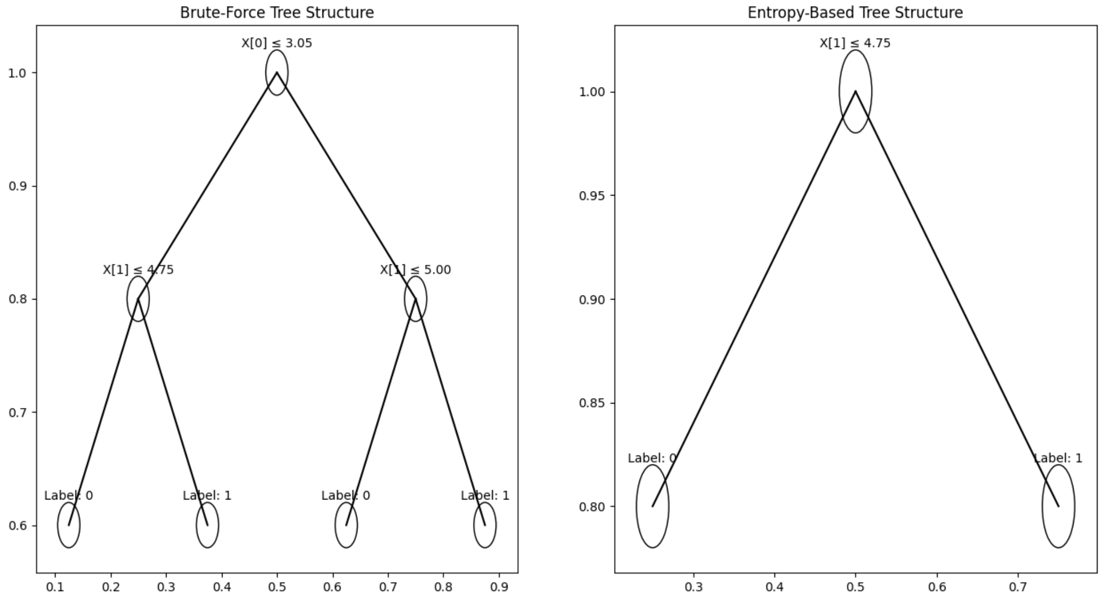
returned by **B** (Binary entropy) = 0.07



```
Training brute-force decision tree...
Brute-force tree error rate: 0.0500

Training entropy-based decision tree...
Entropy-based tree error rate: 0.0700
```

The Trees:



Brute-Force Tree Structure

Entropy-Based Tree Structure

The brute-force approach evaluates all possible splits up to the maximum depth and selects the tree that gives the lowest classification error on the training data, this approach may lead to overfitting due to the exhaustive search for optimal splits, which can fit noise in the training data.

The Binary entropy tree makes splits based on entropy reduction and stops when further splits don't provide significant entropy reduction resulting in a simpler tree that is **preferable** for better generalization, as discussed in class.