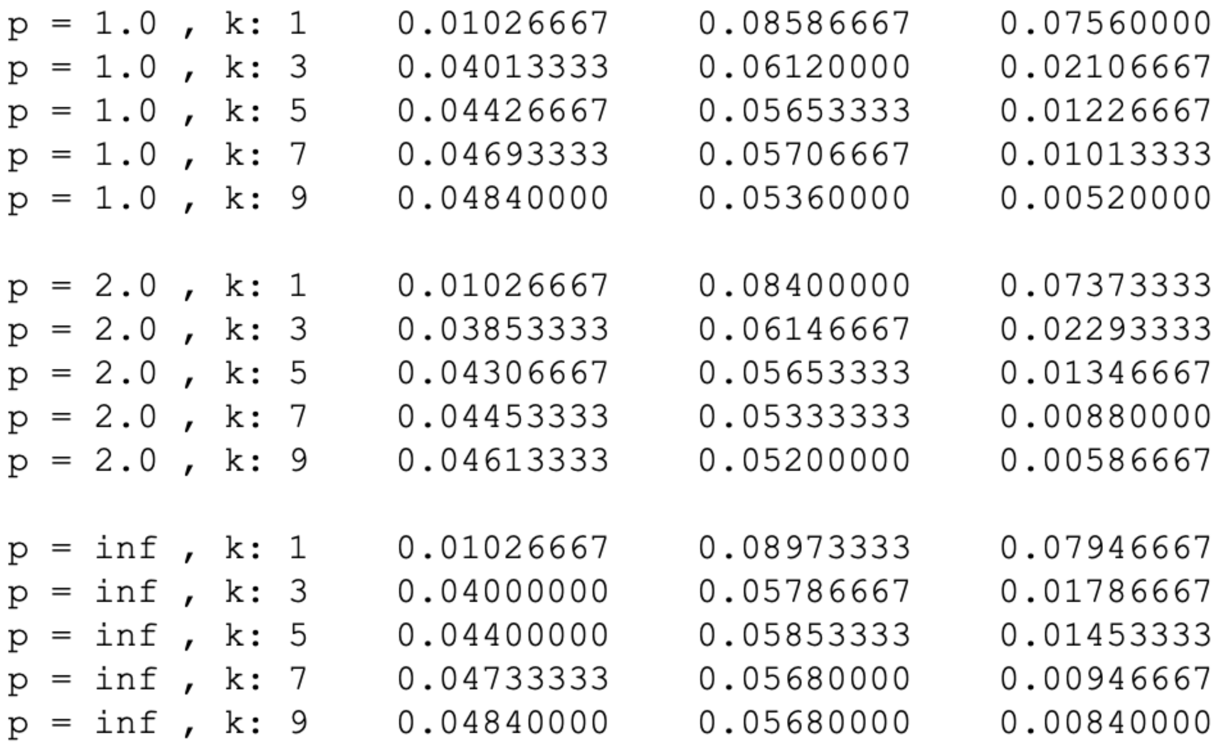
Submitter: Tomer Gozlan ID 314770058
Github : Github Link

Ariel University
Machine Learning
Homework 3

Part (A) - Analysis of Overfitting and the Best Parameters from the 100 Iterations Results



```
p = 1.0 , k: 1    0.01026667    0.08586667    0.07560000
p = 1.0 , k: 3    0.04013333    0.06120000    0.02106667
p = 1.0 , k: 5    0.04426667    0.05653333    0.01226667
p = 1.0 , k: 7    0.04693333    0.05706667    0.01013333
p = 1.0 , k: 9    0.04840000    0.05360000    0.00520000

p = 2.0 , k: 1    0.01026667    0.08400000    0.07373333
p = 2.0 , k: 3    0.03853333    0.06146667    0.02293333
p = 2.0 , k: 5    0.04306667    0.05653333    0.01346667
p = 2.0 , k: 7    0.04453333    0.05333333    0.00880000
p = 2.0 , k: 9    0.04613333    0.05200000    0.00586667

p = inf , k: 1    0.01026667    0.08973333    0.07946667
p = inf , k: 3    0.04000000    0.05786667    0.01786667
p = inf , k: 5    0.04400000    0.05853333    0.01453333
p = inf , k: 7    0.04733333    0.05680000    0.00946667
p = inf , k: 9    0.04840000    0.05680000    0.00840000
```

Based on the provided results, we can determine whether overfitting occurs by analyzing the training error, test error, and error difference for different values of k and p.

Which parameters of k,p are the best and Why is this?

The optimal parameters for the k-NN model are **k=9,p=2**, in **Euclidean distance**, as this configuration yields the lowest test error while maintaining a minimal train-test error gap, indicating strong generalization. The choice of k=9 ensures that the model benefits from a sufficient number of neighbors, reducing variance while avoiding overfitting, which is evident at lower k values, particularly k=1. When k is too small, the model is highly sensitive to noise in the training data, leading to memorization and poor generalization, as reflected in the large discrepancy between training and test errors. Conversely, excessively large k values could lead to underfitting, as the model becomes too generalized and fails to capture essential patterns. The use of p=2 in (Euclidean distance) is also justified as it consistently produces lower test errors compared to Manhattan distance and Chebyshev distance. Euclidean distance provides a natural measure of similarity that performs well in high-dimensional spaces, making it a robust choice for k-NN classification. Given that the test error for k=9,p=2 is among the lowest and the error difference between training and testing is minimal, this parameter selection optimally balances bias and variance, ensuring reliable and well-generalized predictions.

How do you interpret the results and is there overfitting?

The results indicate that the k-NN model's performance varies significantly depending on the choice of k and p. The observed trend shows that for lower values of k, particularly k=1, the training error is very low, but the test error is considerably higher, leading to a large error gap. This suggests that the model is overfitting, as it memorizes the training data but struggles to generalize to unseen test samples. As k increases, the test error gradually decreases, and the gap between training and test errors reduces, which is a strong indication that the model is generalizing better. The best generalization occurs at k=9,p=2, where the test error is minimized and the difference between training and test errors is the smallest, indicating minimal overfitting. Additionally, the results show that Euclidean distance outperforms Manhattan and Chebyshev, as it consistently yields lower test errors across different kk values. This suggests that Euclidean distance provides a more natural way to measure similarity in this dataset.

Part (B) - Analysis of Decision Tree Models: Brute-Force vs. Entropy-Based Splitting:

The visualizations of the Brute-Force Decision Tree and the Entropy-Based Decision Tree reveal fundamental differences in their structure and decision-making processes. Both models were trained with a maximum depth of three levels to classify the dataset efficiently, and their classification errors were **8.00%** for the Brute-Force model and **6.67%** for the Entropy-based model.

Brute-Force Decision Tree

- **a.** The tree exhibits a more symmetric and deeper structure, indicating that Gini-based splitting may have favored slightly more complex branches.
- **b.** The decision nodes demonstrate a greater reliance on feature X[1] (at multiple depths), suggesting that this feature played a crucial role in splitting the dataset.
- **c.** Leaf nodes appear at deeper levels, meaning some branches required more splits to reach pure classifications.
- **d.** The increased branching complexity might contribute to a slightly higher misclassification error compared to the entropy-based model.

Entropy-Based Decision Tree

- **e.** The entropy-based model follows a more compact and optimal structure, suggesting that entropy effectively minimized uncertainty at each split.
- **f.** The first-level split at X[1] ≤ 1.90 immediately distinguishes an entire class, leading to a simpler structure.
- **g.** This model achieved fewer splits overall, meaning that it reached a decision with fewer steps, potentially leading to improved generalization.
- **h.** The tree's leaf nodes appear at shallower levels, meaning the classification decisions required fewer comparisons, contributing to a lower error rate (6.67%).

Performance and Interpretability

- The entropy-based decision tree achieved a lower misclassification rate, suggesting that minimizing information entropy led to better decision boundaries.
- The Brute-Force Gini approach, while still effective, resulted in deeper trees and more splits, which could indicate suboptimal feature selection.
- The entropy-based model’s structure suggests that it generalizes better to unseen data, as it reduces unnecessary complexity and prioritizes more informative splits.

Conclusion

From both an academic and practical perspective, the Entropy-Based Decision Tree emerges as the superior approach in this classification scenario due to its lower error rate, fewer decision nodes, and more efficient structure. While the Brute-Force Gini approach remains a valid method, its slightly higher misclassification rate and increased tree complexity suggest that it may not always identify the most optimal feature splits. Therefore, for datasets where minimizing classification errors and optimizing decision boundaries are critical, the entropy-based approach is the preferred choice. Future work could include feature importance analysis, hyperparameter tuning, and cross-validation to further refine these decision tree models and confirm their robustness across different datasets.