# Evaluation Report

## Explanation of Implementation:

I implemented various machine learning models to classify and predict the outcomes of tic-tac-toe games based on given datasets. The models used include:

* Support Vector Machine (SVM) Classification
  + Hyperparameters: kernel=’rbf’, gamma=’scale’, probability=True, decision\_function\_shape=’ovr’
* K-Nearest Neighbors (KNN) Classification and Regression
  + Number of classification neighbors: 1
  + Number of regression neighbors: 9
* Multi-Layer Perceptron (MLP) Classification and Regression
  + Hyperparameters: solver=’adam’, alpha=1e-6, max\_iter=300, hidden\_layer\_sizes=(256,256,128,), activation=’relu’, early\_stopping=True, validation\_fraction=0.1
* Linear Regression (using normal equations)

Each model was trained on three datasets: tictac\_single.txt, tictac\_final.txt, and tictac\_multi.txt. The models were then evaluated based on their accuracy and, where applicable, confusion matrices were generated.

## Evaluation Results:

The accuracy results for each model on the respective datasets are as follows:

| **Model** | **Dataset** | **Accuracy (%)** | **R^2 Score (if provided)** |
| --- | --- | --- | --- |
| SVM Classification | tictac\_single.txt | 81.59 | - |
| SVM Classification | tictac\_final.txt | 97.57 | - |
| SVM Classification | tictac\_multi.txt | 92.14 | - |
| KNN Classifier | tictac\_single.txt | 75.94 | - |
| KNN Classifier | tictac\_final.txt | 99.65 | - |
| KNN Classifier | tictac\_multi.txt | 71.31 | - |
| KNN Regressor | tictac\_single.txt | 69.08 | - |
| KNN Regressor | tictac\_final.txt | 79.53 | - |
| KNN Regressor | tictac\_multi.txt | 77.88 | - |
| MLP Classifier | tictac\_single.txt | 85.71 | - |
| MLP Classifier | tictac\_final.txt | 97.92 | - |
| MLP Classifier | tictac\_multi.txt | 74.26 | - |
| MLP Regressor | tictac\_single.txt | 75.76 | 0.7576 |
| MLP Regressor | tictac\_final.txt | 92.60 | 0.9260 |
| MLP Regressor | tictac\_multi.txt | 78.50 | 0.7850 |
| Linear Regression | tictac\_single.txt | 13.63 | - |
| Linear Regression | tictac\_final.txt | 64.93 | - |
| Linear Regression | tictac\_multi.txt | 78.31 | - |

## Overall Performance:

* Overall, the highest accuracy was K-NN classifier on the final dataset, as the accuracy was 99.65%. However, the SVM classification seems to be the most reliable model, but depending on the specific requirements and constraints, other models, like MLP Classifier, may be considered.
* KNN Classifier performs exceptionally well on the tictac\_final.txt dataset but has a significant drop in accuracy on the tictac\_multi.txt dataset.
* MLP Classifier and MLP Regressor both show good performance across datasets, with the Classifier variant slightly outperforming the Regressor.
* Linear Regression has the lowest accuracy on the tictac\_single.txt dataset, suggesting it might not be the best model for this particular dataset.

## Dataset Specific Observation:

* tictac\_single.txt: This dataset seems to be challenging for most models, with Linear Regression struggling the most. SVM Classification and MLP Classifier are the top performers here.
* tictac\_final.txt: Almost all models achieve high accuracy on this dataset, with KNN Classifier reaching an impressive 99.65%.
* tictac\_multi.txt: SVM Classification remains consistent, while KNN Classifier sees a significant drop in performance.

## Instructions on How to Run the Code:

Navigate to the directory containing the evaluation.py script.

Run the script using the command: /usr/bin/python3 /path/to/evaluation.py

The script will evaluate each model on the three datasets and print the results.

## Bugs and Difficulties:

* This assignment was difficult because there were no specific instructions.
* The linear regression model showed a significantly lower accuracy on the tictac\_single.txt dataset compared to the other models. This suggests that linear regression might not be the best fit for this particular dataset.
* The tictac\_single.txt and tictac\_final.txt datasets have similar structures, but the models performed differently on them. This could be due to the distribution of positive and negative outcomes or specific game states in each file.

## Explain why certain models scale better to larger dataset than others:

When a substantial fraction of ground truth values in a dataset are corrupted by random noise, different machine learning models react in varied ways. For Support Vector Machines (SVM), the introduction of noise can shift the optimal hyperplane, potentially leading to misclassifications, especially if the noise causes overlap between classes in the feature space. Multi-Layer Perceptrons (MLP) or neural networks might overfit to the noisy data, especially if they have a large number of parameters. They might learn the noise as if it's a genuine pattern, which can degrade performance on clean test data. K-Nearest Neighbors (KNN), being a non-parametric method, is directly influenced by the data it's trained on. If the training data has noise, the decision boundaries can become erratic, leading to inconsistent classifications. KNN's reliance on local data points means that noisy data can heavily influence predictions, especially if the number of neighbors (k) is small. In essence, while SVM tries to find a hyperplane that best separates the classes and might struggle with noisy overlap, MLP risks learning the noise, and KNN's predictions can be directly swayed by noisy neighbors.