Report for Part I

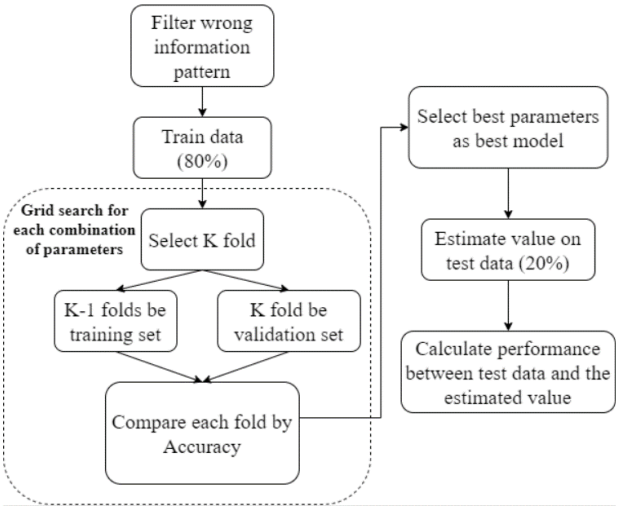
Team: Hao-Yu Liao, Shuaizhou Hu, Xinyao Zhang

Implementation the programs

In order to perform the programs, we take the following steps:

1. Complete the codes for each classifier and regressor.
2. Change the parameters of each program to find the best-fit classifier and regressor.
3. Design the human gameplay program.
4. Perform the human gameplay program and evaluate the accuracy.

The experimental design is presented as follows:



We divide the dataset to train data (80% all dataset) and test data (20% all dataset). The train data will be used for cross-validation with 10 folds for each model. The test data will not be used in the process of cross-validation. The way to find the best parameters is by the grid search method. After the process of cross-validation, the best model with the best hyperparameters will be tested to check the model performance.

Evaluation results

1. The accuracy for the classifiers

Model Type	Training_Accuracy	Test_Accuracy
Classify_LinearSVM_multilabel	0	0
Classify_RBFSVM_multilabel	1	0.85
Classify_KNN_multilabel	1	0.86
Classify_multiperceptron_multilabel	1	0.88
Classify_LinearSVM_singlelabel	0.37	0.35
Classify_RBFSVM_singlelabel	1	0.94
Classify_KNN_singlelabel	1	0.92
Classify_multiperceptron_singlelabel	1	0.95
GA_Classify_singlelabel	1	0.95

The training accuracy is from cross-validation, and the test accuracy is calculated by the remaining 20% of the dataset. The results show that the best performance of classifiers is the multilayer perceptron model using a single-label dataset. The classification model trained by the single label has better performance than the model trained by the multi-label. Besides, the linear SVM cannot be trained well even though we try to use different parameters. We think the kernel with linear function is not suitable because it seems that the dataset is non-linear. However, if we change the kernel function with RBF, the SVM can perform better.

2. The accuracy for the regressors

Model Type	Training_Accuracy	Test_Accuracy
Regressor_KNN_multilabel	1	0.84
Regressor_LinearReg_multilabel	0.04	0.04
Regressor_multiperceptron_multilabel	0.99	0.87

The best performance of regressors is also the multilayer perceptron model using a multilabel dataset. The parameter of linear regression is calculated by the normal equation. The linear regression model has the same condition as linear SVM.

3. Investigate the accuracy of the classifiers under different situations.

Model Type	Test_Accuracy (full dataset)	Test_Accuracy (1/10 dataset)	Test_Accuracy (add noise)
Classify_LinearSVM_multilabel	0	0	0.01
Classify_RBFSVM_multilabel	0.85	0.29	0.01
Classify_KNN_multilabel	0.86	0.3	0.01
Classify_multiperceptron_multilabel	0.88	0.45	0.01
Classify_LinearSVM_singlelabel	0.35	0.38	0.79
Classify_RBFSVM_singlelabel	0.94	0.6	0.81
Classify_KNN_singlelabel	0.92	0.56	0.81
Classify_multiperceptron_singlelabel	0.95	0.69	0.81

After we reduce data or add noise to the data, the performance of each model is reduced. The model's performance trained by multi-label is decreased seriously, but the model's performance trained by single label still has the performance. It seems that the models rained by multi-label datasets are more sensitive than the models trained by single-label datasets. In addition, We think the multilayer perceptron model can better scale to larger datasets, because the model can include more parameters, and changing parameters helps to fit larger datasets.

4. The performance of the human gameplay program

We input each program model with best-fit parameters. For classification models, we cannot beat the computer, and sometimes we will lose the game if we are not careful. Overall, classification models can perform better than regression models. Because the output dataset is binary, the problem is not suitable for regression models. The regression models output numeric numbers instead of 0 or1. In addition, for regression models, we can win the computer very easily because of poor trained performance. In conclusion, the computer has a better performance when we played with it based on classification models.

Instruction on running the programs

Three programs, including 'Classifiers.py', 'Regressors.py' and 'PlayTicTacToe.py', can be run directly. The 'ReadMeFirst.txt' file explains contents of each file. Another one is GA_ClfMLP_singleLabel. We implement genetic algorithm to find best hyperparameters for multilayer perceptron. The programs can print the accuracy and confusion matrix.

Discussion

1. We think that the final board dataset can be used for reinforcement learning. In this way, the computer ought to take action in an environment to make optimal moves.
2. According to the results shown several training accuracies equals 1, we think this is overfitting. We had tried different parameters by the grid search method, but the models still have slight overfitting.
3. We also do additional research on the kernel function for SVM. After comparing RBFSVM to LinearSVM models, we think the RBFSVM can generate more accurate results. Because the dataset is nonlinear, the RBF kernel can perform better than linear kernel.
4. We also explore the genetic algorithms to find optimal multilayer perceptron parameters. The following steps of genetic algorithms are the 'Initial Population > Evaluation > Selection > Crossover > Mutation > repeat until converge'.