

1. When you prepare the data for training the models, did you discover any attribute to remove or any new attribute to add? If you did, discuss the choices.
1. When preparing the data, I discovered I needed to drop the name attribute of the initial dataset because the name of one player should not be relevant to the success of another player. Additionally, the different algorithms expect numeric data, and this attribute did not meet that criterion. I was getting various runtime errors when it was still being included, which were resolved once it was dropped.
2. Normalizing (a.k.a., scaling) features is desirable for distance-based models, e.g., k-nearest neighbors. Did you try feature normalization for some of the models? If so, talk about if any improvement.
2. I tried with and without scaling for each of the four classification models assigned in this project, K-Nearest Neighbors classifier, Random Forests Classifier, Logistic Regression Classifier, and Artificial Neural Networks (Multi-Layer Perceptron Classifier). On the final test sets, scaling led to a worse result on K-NN Classifier 0.7935 (Without) vs 0.7833 (With). On the other hand, it made no difference in the result for Random Forest 0.7955 (Without) vs 0.7955 (With), while providing superior results for Logistic Regression 0.7845 (Without) vs 0.7978 (With) and Multi-layer Perceptron Classifier 0.7887 (Without) vs 0.8075 (With).
3. Regularization is a common practice to battle overfitting. How is varying the penalty parameter in logistic regression affect the performance  $F_1$  score on testing? (The logistic regression penalty parameter may be 'none', 'l1', 'l2' or 'elasticnet'.)
3. Interestingly, exploring different penalty params for logistic regression led to worse results during the 10-Fold Cross Validation Average F1 Scores, but better results during the final test set. For LR without scaling, changing the Penalty Parameter used caused C found during Grid Search to change from 1 to 10 for L1, Elastic net, and None, while C remained at 0.1 for LR with scaling with the same penalty params. The F1 score reaches the range  $0.800 < \text{score} < 0.801$  for LR with scaling with the 'elastic net' and the 'none' penalty parameters but does worse on cross

validation, possibly implying that they do not generalize as well as the best found hyperparameters during Grid Search of LR.

4. These models have hyperparameters. When training, experiment using GridSearch to select hyperparameters for your models. What are the best hyperparameters among those you tried?

4. I ran Grid Search for each of the different models with and without scaling to try to find good hyperparameters. I limited the number of hyperparameters for which optimal values were searched, and the number of potential values for each to save run time given Grid Search grows exponentially to the number of hyperparameters and their possible values. The hyperparameters expected to generalize the best-found during Grid Search for each are as follows:

- KNN Without Scaling:
  - Best k: 29
- KNN with Scaling:
  - Best k: 25
- RF Classifier without scaling:
  - Best # of Trees: 500
  - Best Max Depth Per Tree: 5
  - Best Max Number of Features Per Tree Training Subset: Square Root of Total Number of Features in Training Set
- RF Classifier:
  - Best # of Trees: 500
  - Best Max Depth Per Tree: 5
  - Best Max Number of Features Per Tree Training Subset: Square Root
- LR Classifier Without Scaling:
  - Best C (Regularization Strength): 1
  - Best Penalty Parameter (Regularization Type): L2 - Add squared values of the coefficients to the loss function.
- LR Classifier With Scaling:
  - Best C: 0.1
  - Best Penalty Parameter: L1 – Add absolute values of the coefficients to the loss function.
- MLP Classifier Without Scaling:
  - Best Learning Rate: 0.1
  - Best Alpha (Regularization Strength): 0.0001
  - Best Hidden Layer Sizes (# Num of Hidden Layers and Their Sizes): (100) – 1 Hidden Layer with 100 Neurons

- Best Solver Algorithm Used: Adam
- Best Activation Function: Relu
- MLP Classifier With Scaling:
  - Best Learning Rate: 0.01
  - Best Alpha (Regularization Strength): 0.1
  - Best Hidden Layer Sizes (# Num of Hidden Layers and Their Sizes): (100) – 1 Hidden Layer with 100 Neurons
  - Best Solver Algorithm Used: Adam
  - Best Activation Function: Logistic

## 5. Which model you experimented with gives the best $F_1$ score on testing?

5. Out of all the models with the best hyperparameters found during Grid Search, MLP with scaling gives the highest F1 score on the test set, implying that it will generalize the best to new data overall. Hence, I would recommend MLP Classifier with Scaling using the best found hyperparameters listed above for the optimal result.