

Report for project 3

Yihong Zhou (yzhou8)

Mengwen Li (mli2)

We have 7 features to extract from a raw board state.

1. Which player has a piece at the bottom left corner of the board
2. Which player has more pieces in the center rows
3. Which player has more continuous pieces in vertical lines
4. Which player has more continuous pieces in horizontal lines
5. Which player has more continuous pieces in vertical lines and horizontal lines
6. Which player can connect more live-2 pieces vertically
7. Which player can connect more live-2 pieces horizontally

Feature 1. Which player has a piece at the bottom left corner of the board :

Description:

The bottom left corner of the board is checked and its value is returned.

Design Strategy:

This feature is in “PieceLeftCorner” class, it gets the element at location (1, a) of the board and returns the value.

Why do you think it would work:

This is the provided feature, it does not relate too strongly to the outcome of the game.

In Decision tree:

```
=== Run information ===
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.Remove-R3-6
Instances: 1000
Attributes: 2
    winner
    PieceLeftCorner
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1
Size of the tree :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      737          73.7 %
Incorrectly Classified Instances    263          26.3 %
Kappa statistic                     0
Mean absolute error                 0.3877
Root mean squared error             0.4403
Relative absolute error             99.9355 %
Root relative squared error         99.9999 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===
               TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
               ----  -
               1      1      0.737      1      0.849      0.495      1
               0      0      0      0      0      0.495      2
Weighted Avg.   0.737   0.737   0.543   0.737   0.625   0.495

=== Confusion Matrix ===
      a  b  <-- classified as
737  0 1  a = 1
263  0 1  b = 2
```

```
=== Classifier model (full training set) ===
```

```
J48 unpruned tree
```

```
-----
```

```
: 1 (1000.0/263.0)
```

```
Number of Leaves :    1
```

```
Size of the tree :    1
```

```
Time taken to build model: 0 seconds
```

```
=== Stratified cross-validation ===
```

```
=== Summary ===
```

Correctly Classified Instances	737	73.7	%
Incorrectly Classified Instances	263	26.3	%
Kappa statistic	0		
Mean absolute error	0.3877		
Root mean squared error	0.4403		
Relative absolute error	99.9358	%	
Root relative squared error	99.9999	%	
Total Number of Instances	1000		

The result of decision tree base on only this attribute is not poor, but not very good as well. This model is not relevant for predicting final winner. Although the accuracy is 73.7% and error rate is 26.3%, we can see from confusion matrix, all opponents are classified to player1.

In Neural networks training:

```
=== Run information ===
```

```
Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R3-8
Instances: 1000
Attributes: 2
           winner
           PieceLeftCorner
Test mode:10-fold cross-validation
```

```
=== Classifier model (full training set) ===
```

```
Sigmoid Node 0
  Inputs  Weights
  Threshold  0.6635314419874241
  Node 2  1.5713723620845133
  Node 3  0.8574606386082793
Sigmoid Node 1
  Inputs  Weights
  Threshold  -0.6635314419874241
  Node 2  -1.5713723620845137
  Node 3  -0.8574606386082797
Sigmoid Node 2
  Inputs  Weights
  Threshold  -1.5444078662650198
  Attrib PieceLeftCorner=0  0.05226553179022001
  Attrib PieceLeftCorner=1  -1.5128252095145283
  Attrib PieceLeftCorner=2  2.957126233000844
Sigmoid Node 3
  Inputs  Weights
  Threshold  -2.37459856124584
  Attrib PieceLeftCorner=0  2.824223998890615
  Attrib PieceLeftCorner=1  -0.780562772101071
  Attrib PieceLeftCorner=2  0.33109879224394123
Class 1
  Input
  Node 0
Class 2
  Input
  Node 1
```

```
Time taken to build model: 0.25 seconds
```

```

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      737          73.7 %
Incorrectly Classified Instances    263          26.3 %
Kappa statistic                     0
Mean absolute error                 0.3652
Root mean squared error            0.4303
Relative absolute error            94.1512 %
Root relative squared error        97.7272 %
Total Number of Instances         1000

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1         1       0.737      1      0.849    0.633    1
      0         0         0         0         0     0.633    2
Weighted Avg.   0.737   0.737    0.843   0.737   0.625    0.633

=== Confusion Matrix ===
  a  b  <-- classified as
737  0  |  a = 1
263  0  |  b = 2

```

As the same before, all opponents are classified to player1.

Feature 2. Which player has more pieces in the center column :

Description:

This feature checks which player has more pieces at center region. The definition for center region is all rows in col 3 to col 5.

Design Strategy:

The implementation for this feature is in “CenterControl” class. All rows from col 3 to col 5 is checked and the number of player 1’s piece and player 2’s piece is counted. The returned value of this function is the difference between the number of player 1’s pieces and that of player 2’s pieces.

Why do you think it would work:

Typically, if a player has more pieces in the center region, it will have more spaces to make moves and make connections to the borders. As a result, the player has more chance to win the game.

In Decision tree:

```

=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.Remove-R2,4-6
Instances: 1000
Attributes: 2
          winner
          CenterControl
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----

CenterControl = -3: 1 (9.0/4.0)
CenterControl = -2: 2 (106.0/44.0)
CenterControl = -1: 1 (236.0/103.0)
CenterControl = 0: 1 (293.0/63.0)
CenterControl = 1: 1 (253.0/24.0)
CenterControl = 2: 1 (88.0/6.0)
CenterControl = 3: 1 (13.0/1.0)
CenterControl = 4: 1 (2.0)

Number of Leaves :    8
Size of the tree :    9

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      744           74.4 %
Incorrectly Classified Instances    256           25.6 %
Kappa statistic                    0.1586
Mean absolute error                 0.3452
Root mean squared error             0.4188
Relative absolute error             88.9836 %
Root relative squared error         95.1351 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.946    0.821    0.763    0.946    0.845    0.686    1
          0.179    0.054    0.54    0.179    0.269    0.686    2
Weighted Avg.   0.744    0.62    0.705    0.744    0.693    0.686

=== Confusion Matrix ===

```

```

=== Confusion Matrix ===

```

```

      a   b   <-- classified as
697  40 |   a = 1
216  47 |   b = 2

```

```

=== Classifier model (full training set) ===

```

```

J48 pruned tree

```

```

-----

CenterControl = -3: 1 (9.0/4.0)
CenterControl = -2: 2 (106.0/44.0)
CenterControl = -1: 1 (236.0/103.0)
CenterControl = 0: 1 (293.0/63.0)
CenterControl = 1: 1 (253.0/24.0)
CenterControl = 2: 1 (88.0/6.0)
CenterControl = 3: 1 (13.0/1.0)
CenterControl = 4: 1 (2.0)

```

```

Number of Leaves :    8

```

```

Size of the tree :    9

```

```

Time taken to build model: 0 seconds

```

We can see this model has relevant to predicate the winner. It has the highest accuracy 74.4% among other single attribute models. It classifies some player1s to player2 classes, which is totally different from the result of Feature1. But it also classifies some player1s to player2 classes and player2 data to player1 classes. So the accuracy is enough. This model also make a lot of mistake when player2 is just 1 more than player1 in central area. Nearly 50 % classifications are wrong there.

In Neural networks training:

```
=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation:      out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2,4-8
Instances:      1000
Attributes:      2
                  winner
                  CenterControl
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs  Weights
  Threshold  0.003857019125954002
  Node 2  0.7318599579518353
  Node 3  1.0061689848808526
  Node 4  0.7032452649277364
  Node 5  0.6691757385286715
  Node 6  1.5166639538030988
Sigmoid Node 1
  Inputs  Weights
  Threshold -0.0038189557849780276
  Node 2  -0.7239790373531337
  Node 3  -1.0062297888604244
  Node 4  -0.7270895336415132
  Node 5  -0.6542814952908834
  Node 6  -1.5166803278591086
Sigmoid Node 2
  Inputs  Weights
  Threshold -0.5634182257692961
  Attrib CenterControl=-3  1.116447691067436
  Attrib CenterControl=-2  -1.583382381759434
  Attrib CenterControl=-1  1.033738611762838
  Attrib CenterControl=0  -0.7844950547181951
  Attrib CenterControl=1  0.10416439700369531
  Attrib CenterControl=2  0.9275680296410296
  Attrib CenterControl=3  0.8219538760613874
  Attrib CenterControl=4  1.7123346047235721
Sigmoid Node 3
  Inputs  Weights
  Threshold -0.16400630945981035
  Attrib CenterControl=-3  -0.532055619628793
  Attrib CenterControl=-2  -2.8220388448676226
  Attrib CenterControl=-1  -0.6618563693410883
  Attrib CenterControl=0  -1.9696027848238719
  Attrib CenterControl=1  2.133183251186786
  Attrib CenterControl=2  1.6907752412006
  Attrib CenterControl=3  1.3657335125596285
  Attrib CenterControl=4  1.4930715355687394
Sigmoid Node 4
  Inputs  Weights
  Threshold -0.5307283600270424
  Attrib CenterControl=-3  0.9717947004292701
  Attrib CenterControl=-2  -1.708995390476727
  Attrib CenterControl=-1  0.991403986337009
  Attrib CenterControl=0  -0.9309496919992465
  Attrib CenterControl=1  0.07024161567688159
  Attrib CenterControl=2  1.0175715087680344
  Attrib CenterControl=3  0.9401540054172903
  Attrib CenterControl=4  1.904278886863951
Sigmoid Node 5
  Inputs  Weights
  Threshold -0.5418938292821709
  Attrib CenterControl=-3  0.42142393566582176
  Attrib CenterControl=-2  -1.7974573970696357
  Attrib CenterControl=-1  1.1868365840780937
  Attrib CenterControl=0  -0.8816380490039071
  Attrib CenterControl=1  0.36146891053191
  Attrib CenterControl=2  1.2903305618627074
  Attrib CenterControl=3  1.111035092157095
  Attrib CenterControl=4  1.8244553836093444
Sigmoid Node 6
  Inputs  Weights
  Threshold 0.10545148701580646
  Attrib CenterControl=-3  -2.0278046582686473
  Attrib CenterControl=-2  -4.0825975414634295
  Attrib CenterControl=-1  -1.8434148368609786
  Attrib CenterControl=0  1.5351336472932948
  Attrib CenterControl=1  1.261262344701379
  Attrib CenterControl=2  1.7212008559046825
  Attrib CenterControl=3  1.393920848665247
  Attrib CenterControl=4  1.1959208908494947
Class 1
  Input
  Node 0
Class 2
  Input
  Node 1
```

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      740          74   %
Incorrectly Classified Instances    260          26   %
Kappa statistic                     0.1908
Mean absolute error                 0.3245
Root mean squared error             0.4085
Relative absolute error             83.6645 %
Root relative squared error         92.7809 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.919    0.76    0.772    0.919    0.839      0.733    1
      0.24    0.081    0.512    0.24    0.326      0.733    2
Weighted Avg.   0.74    0.582    0.704    0.74    0.704      0.733

=== Confusion Matrix ===

  a  b  <-- classified as
677 60 |  a = 1
200 63 |  b = 2

```

This model is relevant good at predict winner. The accuracy of correctness is higher than other attributes.

Feature 3. Which player has more continuous pieces (live 2-piece and live 3-piece) in vertical lines:

Description:

This feature checks which player has more 2-pieces and 3-pieces that can be connected to live 3-pieces and 4-pieces on the next move vertically.

Design Strategy:

The code used to implement this feature is in “VerticalTwo” class and “verticalN” function in “Helper” class. The function checks how many spots player can go to form a live (n + 1)-piece vertically. It will return the difference between the spot number for player 1 and player 2. For example, if the following board is given, the count for the first player is 1 since player 1 can put piece on spot (2, 1) to form a live 3-piece and the count for the second player is 2 since player 2 can put piece on spot (3, 2) and (2, 6) to form two live 3-pieces.

```

1
*****
0000000
0000000
0000000
0020000
0120002
0110012
*****
fpCnt is 1 spCnt is 2

```

In Decision tree:

```
=== Run information ===
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.Remove-R2-3,5-6
Instances: 1000
Attributes: 2
          winner
          Vertical connected two + three
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1
Size of the tree :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      737           73.7 %
Incorrectly Classified Instances    263           26.3 %
Kappa statistic                     0
Mean absolute error                 0.3877
Root mean squared error            0.4403
Relative absolute error            99.9358 %
Root relative squared error        99.9999 %
Total Number of Instances         1000

=== Detailed Accuracy By Class ===
          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1         1         0.737       1       0.849     0.495    1
          0         0         0         0       0         0.495    2
Weighted Avg.   0.737   0.737   0.543   0.737   0.625     0.495

=== Confusion Matrix ===
      a  b  <-- classified as
      737  0 |  a = 1
      263  0 |  b = 2
```

J48 unpruned tree

```
-----

Vertical connected two + three = -2: 2 (1.0)
Vertical connected two + three = -1: 1 (204.0/91.0)
Vertical connected two + three = 0: 1 (615.0/146.0)
Vertical connected two + three = 1: 1 (179.0/25.0)
Vertical connected two + three = 2: 1 (1.0)

Number of Leaves :    5

Size of the tree :    6
```

This model should work, but the result is not good as we thought. The guess is because the data set doesn't have much continuous 2 pieces and 3 pieces in vertical lines. From the unpruned tree, we can tell that when the return value of the feature is 0, the classification result is not very good, this is because both player has a chance to win when they all don't have vertically continuous 2 pieces and 3 pieces. When the return value is -1, most of the winning player is 1, this is probably because player 1 blocks the live 3-piece in the following moves. When the return value is -2, we can tell that player 2 wins and when the return value is 2, we can tell that player 1 wins. The classification doesn't have enough evidence for the model to classify based on this attribute.

In Neural networks training:

```
=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2-3,5-8
Instances: 1000
Attributes: 2
          winner
          Vertical connected two + three
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs  Weights
  Threshold 0.27424125738765404
  Node 2 0.25791394570787574
  Node 3 1.2263056624382043
  Node 4 1.7616548464589223
Sigmoid Node 1
  Inputs  Weights
  Threshold -0.2744290892458886
  Node 2 -0.25311885266596
  Node 3 -1.2261881692555312
  Node 4 -1.7628126647390976
Sigmoid Node 2
  Inputs  Weights
  Threshold -0.9684141673897032
  Attrib Vertical connected two + three=-2 -0.826050229059913
  Attrib Vertical connected two + three=-1 0.3559373850005666
  Attrib Vertical connected two + three=0 -0.10520784375160745
  Attrib Vertical connected two + three=1 1.4210053589354112
  Attrib Vertical connected two + three=2 2.0455007198665
Sigmoid Node 3
  Inputs  Weights
  Threshold -0.6655278847523084
  Attrib Vertical connected two + three=-2 -1.8404134117987743
  Attrib Vertical connected two + three=-1 0.22990261740224585
  Attrib Vertical connected two + three=0 2.754300428405979
  Attrib Vertical connected two + three=1 -1.4215840697024158
  Attrib Vertical connected two + three=2 2.029789460883198
Sigmoid Node 4
  Inputs  Weights
  Threshold -0.6465347046399341
  Attrib Vertical connected two + three=-2 -1.4172663561428362
  Attrib Vertical connected two + three=-1 -0.7243577141063515
  Attrib Vertical connected two + three=0 -0.9901880001847763
  Attrib Vertical connected two + three=1 3.1299799306816665
  Attrib Vertical connected two + three=2 2.0356732018276715
Class 1
  Input
  Node 0

Class 2
  Input
  Node 1

Time taken to build model: 0.32 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      725      72.5 %
Incorrectly Classified Instances    275      27.5 %
Kappa statistic                    0.0012
Mean absolute error                 0.3644
Root mean squared error             0.4311
Relative absolute error             93.9371 %
Root relative squared error         97.9237 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.974    0.973    0.737    0.974    0.839    0.627    1
      0.027    0.026    0.269    0.027    0.048    0.627    2
Weighted Avg.   0.725    0.724    0.614    0.725    0.631    0.627

=== Confusion Matrix ===

  a  b  <-- classified as
718 19 | a = 1
256  7 | b = 2
```

This attribut is not as strong as Feature2. But it is still relavant to the prediction result. The weights of positive results and the weights of negtive results is not fully distinguished, especially the weight of 1 and 0.

Feature 4. Which player has more continuous pieces in horizontal lines:

Description:

This feature checks which player has more 2-pieces and 3-pieces that can be connected to continuous live 3-pieces and 4-pieces on the next move horizontally.

Design Strategy:

The code used to implement this feature is in “HorizontalTwo” class and “horizontalN” function in “Helper” class. The function checks how many spots player can go to form a continuous live $(n + 1)$ -piece horizontally. It will return the difference between the spot number for player 1 and player 2. For example, if the following board is given, the count for the first player is 0 since no live 3-piece or 4-piece can be formed on the next move. The count for the second player is 1 since player 2 can put piece on spot (0, 3) to form a continuous live 3-pieces.

```
0
*****
0000000
0000000
0000100
0020200
0010100
1020200
*****
p1Cnt is 0 p2Cnt is 1
```

In Decision tree:

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.Remove-R2-4,6
Instances: 1000
Attributes: 2
    Winner
    Horizontal connected two + three
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1
Size of the tree :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      737           73.7 %
Incorrectly Classified Instances    263           26.3 %
Kappa statistic                     0
Mean absolute error                 0.3877
Root mean squared error            0.4403
Relative absolute error             99.9358 %
Root relative squared error         99.9999 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1      1      0.737      1      0.849      0.495      1
      0      0      0      0      0      0.495      2
Weighted Avg.   0.737   0.737   0.543   0.737   0.625   0.495

=== Confusion Matrix ===

  a  b  <-- classified as
737  0  |  a = 1
263  0  |  b = 2
```

J48 unpruned tree

: 1 (1000.0/263.0)

Number of Leaves : 1

Size of the tree : 1

This model should work, but the result is not good as we thought. The guess is because the data set doesn't have much continuous 2 pieces and 3 pieces in horizontal lines. The classification doesn't have enough evidence for the model to classify based on this attribute.

In Neural networks training:

```
=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2-4,6-8
Instances: 1000
Attributes: 2
           winner
           Horizontal connected two + three
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs  Weights
  Threshold 1.0056672656224572
  Node 2 1.165502340236737
  Node 3 1.1094224054087747
  Node 4 1.129285695236212
Sigmoid Node 1
  Inputs  Weights
  Threshold -1.0056726240976321
  Node 2 -1.1655004813332812
  Node 3 -1.101904394314706
  Node 4 -1.1367920642308948
Sigmoid Node 2
  Inputs  Weights
  Threshold -0.743876175746937
  Attrib Horizontal connected two + three=-2 -1.7029915866585033
  Attrib Horizontal connected two + three=-1 -1.597442213361852
  Attrib Horizontal connected two + three=0 0.0810581662698627
  Attrib Horizontal connected two + three=1 3.097999315447559
  Attrib Horizontal connected two + three=2 2.3389477343652727
Sigmoid Node 3
  Inputs  Weights
  Threshold -1.4607992557561873
  Attrib Horizontal connected two + three=-2 -0.5205030252945073
  Attrib Horizontal connected two + three=-1 -0.4406572182526489
  Attrib Horizontal connected two + three=0 1.2292459745120916
  Attrib Horizontal connected two + three=1 0.5771663652202085
  Attrib Horizontal connected two + three=2 3.292557042016619
Sigmoid Node 4
  Inputs  Weights
  Threshold -1.3923181298501184
  Attrib Horizontal connected two + three=-2 -0.49382800130089427
  Attrib Horizontal connected two + three=-1 -0.41349351497123854
  Attrib Horizontal connected two + three=0 1.248864605622817
  Attrib Horizontal connected two + three=1 0.5916436118895472
  Attrib Horizontal connected two + three=2 3.3380046364655627
Class 1
  Input
  Node 0
```

```

Class 2
Input
Node 1

Time taken to build model: 0.33 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      737          73.7 %
Incorrectly Classified Instances    263          26.3 %
Kappa statistic                     0
Mean absolute error                 0.3723
Root mean squared error             0.4332
Relative absolute error             95.9835 %
Root relative squared error         98.3849 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1      1      0.737      1      0.849      0.597      1
      0      0      0      0      0      0.597      2
Weighted Avg.  0.737  0.737  0.543  0.737  0.625  0.597

=== Confusion Matrix ===
      a  b  <-- classified as
737  0  |  a = 1
263  0  |  b = 2

```

This model should work, but the result is not good as we thought. The guess is because the data set doesn't have much continuous 2 pieces and 3 pieces in horizontal lines. The classification doesn't have enough evidence for the model to train so its classification is not good enough.

Feature 5. Which player has more continuous pieces in vertical lines and horizontal lines

Description:

This feature checks which player has more 2-pieces and 3-pieces that can be connected to continuous live 3-pieces and 4-pieces on the next move both horizontally and vertically.

Design Strategy:

The code used to implement this feature is in "Combine" class, "horizontalN" and "verticalN" functions in "Helper" class. This feature is the combination of feature 3 and feature 4. It will return the difference between the spot number for player 1 and player 2. For example, if the following board is given, the count for the first player is 3 since it can put piece on spot (0, 0), (0, 3) and (2, 1) to form continuous live 3-pieces. The count for the second player is 2 since it can put pieces on spot (3, 2) and spot (2, 6) to form continuous live 3-pieces.

```

1
*****
0000000
0000000
0000000
0020000
0120002
0110012
*****

```

In Decision tree:

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.Remove-R2-5
Instances: 1000
Attributes: 2
          winner
          Horizontal + Vertical
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1

Size of the tree :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      737           73.7 %
Incorrectly Classified Instances    263           26.3 %
Kappa statistic                     0
Mean absolute error                 0.3877
Root mean squared error            0.4403
Relative absolute error            99.9358 %
Root relative squared error        99.9999 %
Total Number of Instances         1000

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1         1      0.737       1      0.849    0.495    1
          0         0         0         0         0    0.495    2
Weighted Avg.   0.737   0.737    0.543   0.737    0.625    0.495

=== Confusion Matrix ===

  a  b  <-- classified as
737  0  |  a = 1
263  0  |  b = 2
```

J48 unpruned tree

```
-----

Horizontal + Vertical = -3: 2 (6.0/1.0)
Horizontal + Vertical = -2: 2 (50.0/22.0)
Horizontal + Vertical = -1: 1 (241.0/88.0)
Horizontal + Vertical = 0: 1 (397.0/113.0)
Horizontal + Vertical = 1: 1 (236.0/28.0)
Horizontal + Vertical = 2: 1 (66.0/1.0)
Horizontal + Vertical = 3: 1 (4.0)
```

Number of Leaves : 7

Size of the tree : 8

This attribute combines previous two attributes to one, so the result should be the same as previous vertical and horizontal attributes. This model should work, but the result is not good as we thought. The guess is because the data set doesn't have continuous 3 pieces in vertical lines and in horizontal lines. And when classify the result of -1 and 0, it is hard for model to distinguish who is the winner. The classification doesn't have enough evidence for the model to classify based on this attribute.

In Neural networks training:

```
=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2-5,7-8
Instances: 1000
Attributes: 2
    winner
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs  Weights
  Threshold -0.2701009398422478
  Node 2 2.529652994134559
  Node 3 0.48004010156566457
  Node 4 1.4239411479883903
  Node 5 1.7387942106496872

Sigmoid Node 1
  Inputs  Weights
  Threshold 0.27039140186907734
  Node 2 -2.5297828024479956
  Node 3 -0.4815509154617798
  Node 4 -1.4242466214780827
  Node 5 -1.7305692612781562

Sigmoid Node 2
  Inputs  Weights
  Threshold -0.25867441094520716
  Attrib Horizontal + Vertical=-3 -2.56456610344218
  Attrib Horizontal + Vertical=-2 -0.7705335103273209
  Attrib Horizontal + Vertical=-1 -2.549239459536954
  Attrib Horizontal + Vertical=0 -0.9734647203314529
  Attrib Horizontal + Vertical=1 3.5647744296357002
  Attrib Horizontal + Vertical=2 2.4842725111940025
  Attrib Horizontal + Vertical=3 2.0979033650647287

Sigmoid Node 3
  Inputs  Weights
  Threshold -0.3683845266995889
  Attrib Horizontal + Vertical=-3 -2.141622670747173
  Attrib Horizontal + Vertical=-2 0.5054808033025991
  Attrib Horizontal + Vertical=-1 -0.9520690304335752
  Attrib Horizontal + Vertical=0 0.16459515741407665
  Attrib Horizontal + Vertical=1 0.14981301428421093
  Attrib Horizontal + Vertical=2 2.277781252735999
  Attrib Horizontal + Vertical=3 1.6429432407788476

Sigmoid Node 4
  Inputs  Weights
  Threshold -0.20021180522230458
  Attrib Horizontal + Vertical=-3 -2.372257235776849
  Attrib Horizontal + Vertical=-2 -2.1171356799279755
  Attrib Horizontal + Vertical=-1 1.8222305785378956
  Attrib Horizontal + Vertical=0 2.9837585386063705
  Attrib Horizontal + Vertical=1 -2.4729076208010077
  Attrib Horizontal + Vertical=2 1.7930580670908047
  Attrib Horizontal + Vertical=3 1.3037566953242965

Sigmoid Node 5
  Inputs  Weights
  Threshold -0.7155724464366479
  Attrib Horizontal + Vertical=-3 -1.167322253669606
  Attrib Horizontal + Vertical=-2 -0.2891881483525491
  Attrib Horizontal + Vertical=-1 -1.1924674380160103
  Attrib Horizontal + Vertical=0 0.509970124416502
  Attrib Horizontal + Vertical=1 -0.2479292021146156
  Attrib Horizontal + Vertical=2 3.2596722844126154
  Attrib Horizontal + Vertical=3 2.791820566488929

Class 1
  Input
  Node 0
Class 2
  Input
  Node 1

Time taken to build model: 0.43 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      740          74   %
Incorrectly Classified Instances    260          26   %
Kappa statistic                    0.0825
Mean absolute error                 0.3536
Root mean squared error             0.4252
Relative absolute error             91.1595 %
Root relative squared error         96.5694 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
              0.973    0.913    0.749    0.973    0.847    0.662    1
              0.087    0.027    0.535    0.087    0.15    0.662    2
Weighted Avg.   0.74    0.68    0.693    0.74    0.663    0.662

=== Confusion Matrix ===

  a  b  <-- classified as
717 20 |  a = 1
240 23 |  b = 2
```

This attribute is reliable when to train. The weights of two extrem is clear and it is easy to separated than other attributes. It also has a very high accuracy.

Figure 6. Which player can connect more live 2-pieces vertically

Description:

This feature checks which player has more 1-piece that can be connected to a live 2-piece on the next move vertically. Live 2-piece means that the 2-piece can be further connected to 3-piece if not blocked by the opponent.

Design Strategy:

The code used to implement this feature is in “VerticalOne” class and “verticalN” function in “Helper” class. The function checks the top piece of each column and returns the difference between the number of top pieces of player 1 and player 2. If the top piece row is less than 2, this piece is ignored since it won’t create a live 2-piece after a move.

For example, the board shown below will produce 3 for the first player and 2 for the second player. The return result will be 1 accordingly.

```
68
*****
0000000
0000000
1000000
1000000
2000000
2210210
*****
fpCnt is 3 spCnt is 2
```

The board shown below will produce 2 for the first player since one of the first player’s piece won’t produce a live 2-piece on the next move.

```
75
*****
0000000
0001000
0002000
0002000
0001000
0102021
*****
fpCnt is 2 spCnt is 1
```


In Decision tree:

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2-6,8
Instances: 1000
Attributes: 2
          winner
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1

Size of the tree :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances  737          73.7 %
Incorrectly Classified Instances  263          26.3 %
Kappa statistic                0
Mean absolute error            0.3877
Root mean squared error        0.4403
Relative absolute error        99.9358 %
Root relative squared error     99.9999 %
Total Number of Instances      1000

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1         1         0.737      1         0.849      0.495      1
      0         0         0         0         0         0.495      2
Weighted Avg.   0.737   0.737   0.543   0.737   0.625      0.495

=== Confusion Matrix ===

  a  b  <-- classified as
737  0  1  a = 1
263  0  1  b = 2
```

```
J48 unpruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1

Size of the tree :    1
```

This model should be relevant to predict winner, but the result is not as good as we thought. The classification doesn't have enough evidence for the model to classify based on this attribute. Probably is because of in data set, there is very few potential move at the top to make a connect-2 piece.

In Neural networks training:

```
=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2-6,8
Instances: 1000
Attributes: 2
          winner
          Vertical connected one
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs  Weights
  Threshold  0.7437960562852284
  Node 2  0.590237897763195
  Node 3  0.6091999721473552
  Node 4  0.5996084760644853
  Node 5  0.6291645560099843
  Node 6  0.6069259543173887

Sigmoid Node 1
  Inputs  Weights
  Threshold  -0.7438197592908369
  Node 2  -0.5796060510339132
  Node 3  -0.619306927496906
  Node 4  -0.6195897550644702
  Node 5  -0.5977839573777135
  Node 6  -0.619431819025034

Sigmoid Node 2
  Inputs  Weights
  Threshold  -0.5170464965280037
  Attrib Vertical connected one=-4  -0.45624756392587074
  Attrib Vertical connected one=-3  -0.9622626683947932
  Attrib Vertical connected one=-2  -1.7964142082781411
  Attrib Vertical connected one=-1  1.1194852857230315
  Attrib Vertical connected one=0  1.12689754488156
  Attrib Vertical connected one=1  1.2520106276497238
  Attrib Vertical connected one=2  1.15080527504668198
  Attrib Vertical connected one=3  1.635825105632028

Sigmoid Node 3
  Inputs  Weights
  Threshold  -0.553064091690111
  Attrib Vertical connected one=-4  -0.5038667319988696
  Attrib Vertical connected one=-3  -1.0046466794198607
  Attrib Vertical connected one=-2  -1.8486473600005928
  Attrib Vertical connected one=-1  1.0551675074054463
  Attrib Vertical connected one=0  1.1069564851261078
  Attrib Vertical connected one=1  1.1352349919569227
  Attrib Vertical connected one=2  1.0925263416072135
  Attrib Vertical connected one=3  1.9998320604855024

Sigmoid Node 4
  Inputs  Weights
  Threshold  -0.5073586269966417
  Attrib Vertical connected one=-4  -0.4801732702086646
  Attrib Vertical connected one=-3  -0.9742141861653673
  Attrib Vertical connected one=-2  -1.8124460707576464
  Attrib Vertical connected one=-1  1.0838671475921975
  Attrib Vertical connected one=0  1.1276489856828475
  Attrib Vertical connected one=1  1.1705360728941978
  Attrib Vertical connected one=2  1.1148997752599288
  Attrib Vertical connected one=3  1.8851627685365264

Sigmoid Node 5
  Inputs  Weights
  Threshold  -0.4775515435443456
  Attrib Vertical connected one=-4  -0.47669099099372064
  Attrib Vertical connected one=-3  -0.9859994318368062
  Attrib Vertical connected one=-2  -1.8221387429501397
  Attrib Vertical connected one=-1  1.0677188252207503
  Attrib Vertical connected one=0  1.118381107894933
  Attrib Vertical connected one=1  1.1488559783790153
  Attrib Vertical connected one=2  1.1034446192387606
  Attrib Vertical connected one=3  1.976829942451827

Sigmoid Node 6
  Inputs  Weights
  Threshold  -0.5428771543885735
  Attrib Vertical connected one=-4  -0.49434729535599803
  Attrib Vertical connected one=-3  -0.991516630262034
  Attrib Vertical connected one=-2  -1.8319350299157362
  Attrib Vertical connected one=-1  1.0574646173047164
  Attrib Vertical connected one=0  1.1078942078551048
  Attrib Vertical connected one=1  1.1390027022803237
  Attrib Vertical connected one=2  1.0933089067855193
  Attrib Vertical connected one=3  1.9637219205552927

Class 1
  Input
  Node 0
Class 2
  Input
  Node 1

Time taken to build model: 0.55 seconds
```

```

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      737          73.7 %
Incorrectly Classified Instances    263          26.3 %
Kappa statistic                     0
Mean absolute error                 0.3782
Root mean squared error            0.4398
Relative absolute error             97.5063 %
Root relative squared error        99.9019 %
Total Number of Instances         1000

=== Detailed Accuracy By Class ===
               TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
               -----  -----  -
               1         1         0.737      1         0.849      0.537      1
               0         0         0         0         0         0.537      2
Weighted Avg.   0.737    0.737    0.543    0.737    0.625    0.537

=== Confusion Matrix ===
   a  b  <-- classified as
737  0  |  a = 1
263  0  |  b = 2

```

The classification doesn't have enough evidence for the model to classify based on this attribute. Probably is because of in data set, there is very few potential move at the top to make a connect-2 piece.

Feature 7. Which player can connect more live 2-pieces horizontally

Description:

This feature checks which player has more 1-piece that can be connected to a live 2-piece on the next move horizontally.

Design Strategy:

The code used to implement this feature is in "HorizontalOne" class and "horizontalN" function in "Helper" class. The function checks how many spots player can go to form a live (n + 1)-piece horizontally, if 1 is passed in as a parameter, it will check how many spots player can go so that a live 2-piece will be formed after the move. It will return the difference between the spot number for player 1 and player 2.

For example, the board shown below will return 2 for the first player and 2 for the second player. For the first player, the next move can be (2, 2) or (0, 3) to make a live 2-piece, and for the second player, the next move can be (0, 4) or (0, 6) to make a live 2-piece. As a result, the return value in this case will be 0.

```

6
*****
0000000
0000000
0200000
0100000
0210010
0210020
*****
p1Cnt is 2 p2Cnt is 2

```

In Decision tree:

```
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.Remove-R2-6-weka.filters.unsupervised.attribute.Remove-R2-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances: 1000
Attributes: 2
          winner
          Horizontal connected one
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves :    1
Size of the tree :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      737          73.7 %
Incorrectly Classified Instances    263          26.3 %
Kappa statistic                     0
Mean absolute error                 0.3577
Root mean squared error            0.4403
Relative absolute error             99.9358 %
Root relative squared error        99.9999 %
Total Number of Instances         1000

=== Detailed Accuracy By Class ===
          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1      1      0.737      1      0.849      0.495      1
          0      0      0      0      0      0.495      2
Weighted Avg.   0.737   0.737   0.543   0.737   0.625   0.495

=== Confusion Matrix ===
  a  b  <-- classified as
 737  0 | a = 1
 263  0 | b = 2
```

J48 unpruned tree

```
Horizontal connected one = -5: 2 (1.0)
Horizontal connected one = -4: 1 (22.0/11.0)
Horizontal connected one = -3: 1 (57.0/23.0)
Horizontal connected one = -2: 1 (123.0/44.0)
Horizontal connected one = -1: 1 (150.0/48.0)
Horizontal connected one = 0: 1 (211.0/46.0)
Horizontal connected one = 1: 1 (197.0/52.0)
Horizontal connected one = 2: 1 (134.0/27.0)
Horizontal connected one = 3: 1 (77.0/9.0)
Horizontal connected one = 4: 1 (27.0/1.0)
Horizontal connected one = 5: 2 (1.0)
```

Number of Leaves : 11

Size of the tree : 12

This model should be relevant to predict winner. The classification doesn't have enough evidence for the model to classify based on this attribute. From the unpruned tree, we can tell that when the return value is negative, player 1 wins most of the time. This is not exactly the same as what we expected. This is probably because since a live 2-piece is still far from winning the game, the return value is not that convincing.

In Neural networks training:

```
=== Run information ===

Scheme:weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R2-7
Instances: 1000
Attributes: 2
    winner
    Horizontal connected one
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs  Weights
  Threshold  0.3344362840098688
  Node 2  1.0808573981714444
  Node 3  0.5213330549560964
  Node 4  0.6264875824316608
  Node 5  0.561294722479021
  Node 6  0.6304593500503614
  Node 7  1.0785924691360926

Sigmoid Node 1
  Inputs  Weights
  Threshold  -0.33499582438142794
  Node 2  -1.0806780764495263
  Node 3  -0.5348592076293456
  Node 4  -0.5785882571579297
  Node 5  -0.5355065423143848
  Node 6  -0.6926044155592375
  Node 7  -1.0780166627178787

Sigmoid Node 2
  Inputs  Weights
  Threshold  -0.37117567438263044
  Attrib Horizontal connected one=-5  -1.0282257218661512
  Attrib Horizontal connected one=-4  -1.480206955908929
  Attrib Horizontal connected one=-3  0.012990934577153515
  Attrib Horizontal connected one=-2  0.4438694657763302
  Attrib Horizontal connected one=-1  0.5487637916150652
  Attrib Horizontal connected one=0  0.0014174728844187482
  Attrib Horizontal connected one=1  -1.1701841574037484
  Attrib Horizontal connected one=2  0.29048104524343454
  Attrib Horizontal connected one=3  4.085528717919282
  Attrib Horizontal connected one=4  2.6895255757231404
  Attrib Horizontal connected one=5  -1.2040390423155816

Sigmoid Node 3
  Inputs  Weights
  Threshold  -0.49111322397526186
  Attrib Horizontal connected one=-5  -0.6798401147169928
  Attrib Horizontal connected one=-4  -1.161530014965882
  Attrib Horizontal connected one=-3  0.602509055230196
  Attrib Horizontal connected one=-2  1.2579652374125083
  Attrib Horizontal connected one=-1  1.4872700234822058
  Attrib Horizontal connected one=0  0.8297108899462544
  Attrib Horizontal connected one=1  -0.7215941991292619
  Attrib Horizontal connected one=2  0.4616715191195211
  Attrib Horizontal connected one=3  0.6279641062651455
  Attrib Horizontal connected one=4  2.2481489290865615
  Attrib Horizontal connected one=5  -0.8817264324089629

Sigmoid Node 4
  Inputs  Weights
  Threshold  -0.42924043359104813
  Attrib Horizontal connected one=-5  -0.5358720187649102
  Attrib Horizontal connected one=-4  -1.0113357003963004
  Attrib Horizontal connected one=-3  0.5005281700284624
  Attrib Horizontal connected one=-2  1.07566449530539804
  Attrib Horizontal connected one=-1  1.281268498472106
  Attrib Horizontal connected one=0  0.6330679862266372
  Attrib Horizontal connected one=1  -0.656662239477362
  Attrib Horizontal connected one=2  0.6314090026614618
  Attrib Horizontal connected one=3  -0.15261693530675574
  Attrib Horizontal connected one=4  2.831992483400831
  Attrib Horizontal connected one=5  -0.6658823688235773

Sigmoid Node 5
  Inputs  Weights
  Threshold  -0.42471237001815115
  Attrib Horizontal connected one=-5  -0.641936022709378
  Attrib Horizontal connected one=-4  -1.123877019478238
  Attrib Horizontal connected one=-3  0.6861247587719307
  Attrib Horizontal connected one=-2  1.1985938753422924
  Attrib Horizontal connected one=-1  1.428371785845723
  Attrib Horizontal connected one=0  0.7690316047727109
  Attrib Horizontal connected one=1  -0.7256618782307018
  Attrib Horizontal connected one=2  0.4666706123493697
  Attrib Horizontal connected one=3  0.3450732943494167
  Attrib Horizontal connected one=4  2.5145830472661266
  Attrib Horizontal connected one=5  -0.8318716989292307
```

```

Sigmoid Node 6
Inputs  Weights
Threshold -0.4668573228953942
Attrib Horizontal connected one=-5 -0.5668663445234626
Attrib Horizontal connected one=-4 -1.0490948434028518
Attrib Horizontal connected one=-3 0.36894232120996545
Attrib Horizontal connected one=-2 1.0651999583310103
Attrib Horizontal connected one=-1 1.2798262649502512
Attrib Horizontal connected one=0 0.6490877559280003
Attrib Horizontal connected one=1 -0.6634763831803683
Attrib Horizontal connected one=2 0.6661310658108203
Attrib Horizontal connected one=3 -0.09760495297405718
Attrib Horizontal connected one=4 3.0339639190449548
Attrib Horizontal connected one=5 -0.688140604948332

Sigmoid Node 7
Inputs  Weights
Threshold -0.01937826840724208
Attrib Horizontal connected one=-5 -2.4879654960172313
Attrib Horizontal connected one=-4 -2.3157127506944497
Attrib Horizontal connected one=-3 -1.2836361025805623
Attrib Horizontal connected one=-2 -0.9951389160019519
Attrib Horizontal connected one=-1 -0.8471062704089858
Attrib Horizontal connected one=0 1.88898535459388
Attrib Horizontal connected one=1 1.5184921841824572
Attrib Horizontal connected one=2 3.3388401286422758
Attrib Horizontal connected one=3 2.623515526842251
Attrib Horizontal connected one=4 1.6955242769173837
Attrib Horizontal connected one=5 -2.617744339363336

Class 1
Input
Node 0
Class 2
Input
Node 1

Time taken to build model: 0.75 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      737          73.7 %
Incorrectly Classified Instances    263          26.3 %
Kappa statistic                     0
Mean absolute error                 0.3673
Root mean squared error             0.435
Relative absolute error             94.6797 %
Root relative squared error         98.8087 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1         1      0.737      1      0.849      0.603    1
      0         0         0         0         0         0.603    2
Weighted Avg.  0.737  0.737  0.543  0.737  0.625  0.603

=== Confusion Matrix ===
      a  b  <-- classified as
      737  0 |  a = 1
      263  0 |  b = 2

```

The classification doesn't have enough evidence for the model to classify based on this attribute. The weights are not very clear on the two exeterms. And many of the weight is between -1 to 1 and close to 0. This would make the model hard to distinguish the result.

About Cross Validation

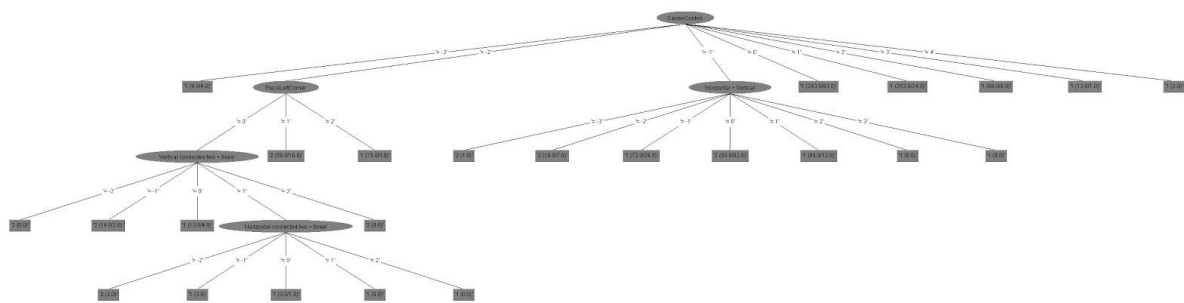
The k fold cross validation is partitioning data set to k groups, using one of k groups in turns for validation and the rest for training. It can prevent problems like overfitting. Using cross validation can maximize the training data we can use comparing to just separate the data set into training set and validation set.

We use 10 fold cross validation.

It is a right number for training and validation. Each time 90% of full data is used for training. 90% is not too far from full 100%, which means that cross validation provides a fair estimation of test performance. Having 5 folds only trained 80% of data, which can be shown

to have great effect on the robustness of the produced model. Having more than 10 folds, the calculation is computationally more demanding and there is an escalating problem with small datasets.

To summary up, there are multiple features should be relevant to the winner result. So we combine all the features to make a decision tree to look at the impact of each feature.



=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: 1000

Attributes: 8

winner
PieceLeftCorner
CenterControl
Vertical connected two + three
Horizontal connected two + three
Horizontal + Vertical
Vertical connected one
Horizontal connected one

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

CenterControl = -3: 1 (9.0/4.0)

CenterControl = -2

| PieceLeftCorner = 0

| | Vertical connected two + three = -2: 2 (0.0)

| | Vertical connected two + three = -1: 2 (12.0/2.0)

| | Vertical connected two + three = 0: 1 (14.0/6.0)

| | Vertical connected two + three = 1

| | | Horizontal connected two + three = -2: 2 (2.0)

| | | Horizontal connected two + three = -1: 1 (3.0)

| | | Horizontal connected two + three = 0: 1 (4.0/1.0)

| | | Horizontal connected two + three = 1: 1 (0.0)

| | | Horizontal connected two + three = 2: 1 (0.0)

| | Vertical connected two + three = 2: 2 (0.0)

| PieceLeftCorner = 1: 2 (56.0/16.0)

| PieceLeftCorner = 2: 1 (15.0/3.0)

CenterControl = -1

| Vertical connected one = -4: 1 (0.0)

| Vertical connected one = -3: 2 (8.0/1.0)

| Vertical connected one = -2: 2 (36.0/14.0)

| Vertical connected one = -1

| | Vertical connected two + three = -2: 1 (0.0)

| | Vertical connected two + three = -1: 1 (3.0/1.0)

| | Vertical connected two + three = 0: 2 (29.0/13.0)

| | Vertical connected two + three = 1: 1 (13.0/3.0)

| | Vertical connected two + three = 2: 1 (0.0)

| Vertical connected one = 0: 1 (85.0/33.0)

| Vertical connected one = 1: 1 (40.0/13.0)

| Vertical connected one = 2: 1 (20.0/8.0)

| Vertical connected one = 3: 1 (2.0)

CenterControl = 0: 1 (293.0/63.0)

```

CenterControl = 1: 1 (253.0/24.0)
CenterControl = 2: 1 (88.0/6.0)
CenterControl = 3: 1 (13.0/1.0)
CenterControl = 4: 1 (2.0)

Number of Leaves :      29

Size of the tree :      35

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      761           76.1 %
Incorrectly Classified Instances    239           23.9 %
Kappa statistic                    0.3101
Mean absolute error                 0.3206
Root mean squared error            0.4169
Relative absolute error             82.6459 %
Root relative squared error        94.6942 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.897    0.62    0.802    0.897    0.847    0.709    1
          0.38    0.103    0.568    0.38    0.456    0.709    2
Weighted Avg.   0.761    0.484    0.741    0.761    0.744    0.709

=== Confusion Matrix ===

  a  b  <-- classified as
661 76 |  a = 1
163 100 | b = 2

```

This result improves a lot. The general result of single attribute is 73.7%. And combining them make a better result.

node 1

Time taken to build model: 7.15 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	727	72.7	%
Incorrectly Classified Instances	273	27.3	%
Kappa statistic	0.279		
Mean absolute error	0.2794		
Root mean squared error	0.4849		
Relative absolute error	72.0289	%	
Root relative squared error	110.1323	%	
Total Number of Instances	1000		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.828	0.555	0.807	0.828	0.817	0.74	1
	0.445	0.172	0.48	0.445	0.462	0.74	2
Weighted Avg.	0.727	0.454	0.721	0.727	0.724	0.74	

=== Confusion Matrix ===

```
a  b  <-- classified as
610 127 |  a = 1
146 117 |  b = 2
```

But when using neural network training, the combination result accuracy decreases to 72.7%.

Interestingly, we found out feature 6 (Which player can connect more live-2 pieces vertically) would do harm to entire prediction.

```

=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R7
Instance: 1000
Attributes: 7
winner
PieceLeftCorner
CenterControl
Vertical connected two + three
Horizontal connected two + three
Horizontal + Vertical
Horizontal connected one
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----

CenterControl = -3: 1 (9.0/4.0)
CenterControl = -2
| PieceLeftCorner = 0
| | Vertical connected two + three = -2: 2 (0.0)
| | Vertical connected two + three = -1: 2 (12.0/2.0)
| | Vertical connected two + three = 0: 1 (14.0/6.0)
| | Vertical connected two + three = 1
| | | Horizontal connected two + three = -2: 2 (2.0)
| | | Horizontal connected two + three = -1: 1 (3.0)
| | | Horizontal connected two + three = 0: 1 (4.0/1.0)
| | | Horizontal connected two + three = 1: 1 (0.0)
| | | Horizontal connected two + three = 2: 1 (0.0)
| | Vertical connected two + three = 2: 2 (0.0)
| PieceLeftCorner = 1: 2 (56.0/16.0)
| PieceLeftCorner = 2: 1 (15.0/3.0)
CenterControl = -1
| Horizontal + Vertical = -3: 2 (1.0)
| Horizontal + Vertical = -2: 2 (18.0/7.0)
| Horizontal + Vertical = -1
| | Vertical connected two + three = -2: 1 (0.0)
| | Vertical connected two + three = -1: 2 (24.0/10.0)
| | Vertical connected two + three = 0: 1 (44.0/15.0)
| | Vertical connected two + three = 1: 1 (6.0/2.0)
| | Vertical connected two + three = 2: 1 (0.0)
| Horizontal + Vertical = 0
| | PieceLeftCorner = 0: 1 (37.0/16.0)
| | PieceLeftCorner = 1: 2 (31.0/9.0)
| | PieceLeftCorner = 2: 1 (20.0/9.0)
| Horizontal + Vertical = 1: 1 (45.0/13.0)
| Horizontal + Vertical = 2: 1 (10.0)
| Horizontal + Vertical = 3: 1 (0.0)

CenterControl = 0: 1 (293.0/63.0)
CenterControl = 1: 1 (253.0/24.0)
CenterControl = 2: 1 (88.0/6.0)
CenterControl = 3: 1 (13.0/1.0)
CenterControl = 4: 1 (2.0)

Number of Leaves : 30

Size of the tree : 37

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 778 77.8 %
Incorrectly Classified Instances 222 22.2 %
Kappa statistic 0.3421
Mean absolute error 0.3161
Root mean squared error 0.4118
Relative absolute error 81.4964 %
Root relative squared error 93.5384 %
Total Number of Instances 1000

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.921 0.624 0.805 0.921 0.859 0.714 1
0.376 0.079 0.631 0.376 0.471 0.714 2
Weighted Avg. 0.778 0.48 0.759 0.778 0.757 0.714

=== Confusion Matrix ===

a b <-- classified as
679 58 | a = 1
164 99 | b = 2

```

The accuracy without feature 6 actually improved. From 76.1% to 77.8% when making decision trees. So the feature is relevant to the prediction, but not predict as good as other features working on this data set.

But when train neural networks, feature 6 do not do harm to the prediction. The guess is, when all features are combined together, the weight of feature 6 decreases during training to decrease its impact on the prediction.

The following table shows the single-feature accuracy and the accuracy after combine models drop one feature in decision tree:

	single accuracy	Accuracy drop when dropping the feature	single accuracy rank	drop accuracy rank
feature 1	73.70%	2.30%	2	2
feature 2	74.70%	3.50%	1	1
feature 3	73.70%	0.80%	2	3
feature 4	73.70%	0.00%	2	5
feature 5	73.70%	0.30%	2	4
feature 6	73.70%	-1.70%	2	7
feature 7	73.70%	-0.30%	2	6

From this table, we can see the rank of each feature. When the model is built by single attribute, the rank is not obvious. But when the model is build by dropping one of feaures, the ranks is obvious. Some of features do harms to the prediction and some greatly improve the accuracy of predition.

The best single feature predictor is feaure2 (Which player has more pieces in the center rows). If total combination of features kick out feaure2, the accuracy of prediction would has the biggest drop from 76.1% to 73.3%. And also, when do the single attribute performance, the model with this feature has highest accuracy.

The best multi feature result is combing all the feaures without feaure6 (Which player can connect more live-2 pieces vertically) with highest score of 77.8%.

The following table shows the single-feature accuracy and the accuracy after combine models drop one feature in neural network traning:

	single accuracy	Accuracy drop when dropping the feature	single accuracy order	drop accuracy order
feature 1	73.70%	0.30%	2	3
feature 2	74.00%	2.5%	1	1
feature 3	72.50%	0.00%	6	4
feature 4	73.70%	-0.5%	2	6
feature 5	73.70%	-0.8%	7	7
feature 6	73.70%	0.00%	2	4
feature 7	73.70%	0.9%	2	2

In the neural network training, feature ranks are different. The ranks are similar between two rank methods. It is not distinct in single accuracy but very distinct after the drop from the combination. The best and the worst are always the same in two methods. The best single feature is still feature 2 whereas the worst feature is feature 5 now.

The best multi feature model will be the model that includes all the features except feature 4 and feature 5 as when they are added to the model, the accuracy of the model will decrease.

We also tried to use “ChiSquaredAttributeEval” to evaluate our attributes based on the “winner” and the result is shown as the following:

```
=== Run information ===

Evaluator:      weka.attributeSelection.ChiSquaredAttributeEval
Search:weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:      out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:     1000
Attributes:    8
               winner
               PieceLeftCorner
               CenterControl
               Vertical connected two + three
               Horizontal connected two + three
               Horizontal + Vertical
               Vertical connected one
               Horizontal connected one
Evaluation mode:evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
  Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 winner):
  Chi-squared Ranking Filter

Ranked attributes:
155.471   3 CenterControl
 94.471   6 Horizontal + Vertical
 55.396   2 PieceLeftCorner
 54.563   4 Vertical connected two + three
 46.407   8 Horizontal connected one
 40.786   5 Horizontal connected two + three
 32.408   7 Vertical connected one

Selected attributes: 3,6,2,4,8,5,7 : 7
```

From the figure above, we can tell that “CenterControl” has the highest rank, then, the “Horizontal + Vertical”. The least ranked attribute is “Vertical connected one”. The ranking result corresponds closely to the rank we get from analyzing the decision tree and neural network.

Extra credit:

1. Since you have all these continuous features (as opposed to discrete features), you might want to play with regression trees.

We tried the “weka.classifiers.trees.m5.M5P -R” on our data set. The tree we get was shown as following:

```
=== Run information ===

Scheme:weka.classifiers.trees.M5P -R -M 4.0
Relation:      out-weka.filters.unsupervised.attribute.Remove-R1-42
Instances:     1000
Attributes:    8
               winner
               PieceLeftCorner
               CenterControl
               Vertical connected two + three
               Horizontal connected two + three
               Horizontal + Vertical
               Vertical connected one
               Horizontal connected one
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

M5 pruned regression tree:
(using smoothed linear models)

CenterControl <= -0.5 :
|   Vertical connected two + three <= -0.5 : LM1 (85/110.048%)
|   Vertical connected two + three > -0.5 :
|   |   Vertical connected one <= -0.5 : LM2 (105/111.694%)
|   |   Vertical connected one > -0.5 : LM3 (161/107.239%)
CenterControl > -0.5 :
|   Horizontal + Vertical <= 0.5 :
|   |   Vertical connected one <= -0.5 : LM4 (137/106.676%)
|   |   Vertical connected one > -0.5 :
|   |   |   Vertical connected two + three <= -0.5 : LM5 (77/105.206%)
|   |   |   Vertical connected two + three > -0.5 : LM6 (199/59.964%)
|   Horizontal + Vertical > 0.5 : LM7 (236/45.754%)
```

```

LM num: 1
winner =
    + 1.5883

LM num: 2
winner =
    + 1.554

LM num: 3
winner =
    + 1.3477

LM num: 4
winner =
    + 1.3091

LM num: 5
winner =
    + 1.2748

LM num: 6
winner =
    + 1.0924

LM num: 7
winner =
    + 1.0533

Number of Rules : 7

Time taken to build model: 0.04 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient          0.4103
Mean absolute error             0.3228
Root mean squared error         0.4015
Relative absolute error         83.1445 %
Root relative squared error     91.0586 %
Total Number of Instances      1000

```

From the result tree, we can tell that the “PieceLeftCorner” feature is pruned out, which means this feature is not relevant to the outcome of the game. The other features are used to classify the data set.

We tried the “weka.classifier.trees.REPTree” on our data set also.

The result was shown as the following:

```
=== Run information ===
```

```
Scheme:weka.classifiers.trees.REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1
Relation:      out-weka.filters.unsupervised.attribute.Remove-R1-42
Instances:     1000
Attributes:    8
               winner
               PieceLeftCorner
               CenterControl
               Vertical connected two + three
               Horizontal connected two + three
               Horizontal + Vertical
               Vertical connected one
               Horizontal connected one
Test mode:10-fold cross-validation
```

```
=== Classifier model (full training set) ===
```


REPTree

=====

```
CenterControl < -0.5
|   Horizontal connected two + three < 0.5
|   |   Vertical connected one < -0.5 : 1.63 (73/0.23) [39/0.24]
|   |   Vertical connected one >= -0.5
|   |   |   Vertical connected two + three < -0.5 : 1.66 (34/0.22) [16/0.24]
|   |   |   Vertical connected two + three >= -0.5
|   |   |   |   PieceLeftCorner < 0.5 : 1.25 (41/0.21) [15/0.14]
|   |   |   |   PieceLeftCorner >= 0.5
|   |   |   |   |   CenterControl < -1.5
|   |   |   |   |   |   PieceLeftCorner < 1.5 : 1.76 (14/0.17) [7/0.21]
|   |   |   |   |   |   PieceLeftCorner >= 1.5 : 1.13 (3/0.22) [5/0.11]
|   |   |   |   |   CenterControl >= -1.5
|   |   |   |   |   |   PieceLeftCorner < 1.5
|   |   |   |   |   |   |   Vertical connected one < 1.5 : 1.38 (21/0.24) [8/0.22]
|   |   |   |   |   |   |   Vertical connected one >= 1.5
|   |   |   |   |   |   |   |   Horizontal connected two + three < -1 : 2 (2/0) [0/0]
|   |   |   |   |   |   |   |   Horizontal connected two + three >= -1 : 1.5 (2/0.25) [4/0.25]
|   |   |   |   |   |   |   |   PieceLeftCorner >= 1.5 : 1.25 (8/0.19) [8/0.19]
|   |   Horizontal connected two + three >= 0.5
|   |   |   Horizontal + Vertical < 1.5
|   |   |   |   Horizontal connected one < -0.5 : 1.5 (8/0.25) [0/0]
|   |   |   |   Horizontal connected one >= -0.5
|   |   |   |   |   Horizontal connected one < 0.5 : 1 (8/0) [3/0]
|   |   |   |   |   Horizontal connected one >= 0.5 : 1.45 (11/0.25) [9/0.25]
|   |   |   Horizontal + Vertical >= 1.5 : 1.08 (8/0) [4/0.25]
CenterControl >= -0.5
|   Horizontal + Vertical < 0.5
|   |   Vertical connected two + three < -0.5
|   |   |   Horizontal connected one < -1.5 : 1.6 (13/0.21) [7/0.31]
|   |   |   Horizontal connected one >= -1.5
|   |   |   |   Vertical connected one < 1.5
|   |   |   |   |   Horizontal connected one < 3.5
|   |   |   |   |   |   Vertical connected one < -0.5
|   |   |   |   |   |   |   Vertical connected one < -1.5 : 1.25 (7/0.2) [5/0.17]
|   |   |   |   |   |   |   Vertical connected one >= -1.5 : 1.75 (8/0.19) [4/0.19]
|   |   |   |   |   |   |   Vertical connected one >= -0.5
|   |   |   |   |   |   |   |   CenterControl < 0.5 : 1.47 (12/0.25) [7/0.25]
|   |   |   |   |   |   |   |   CenterControl >= 0.5 : 1.13 (15/0.12) [9/0.1]
|   |   |   |   |   |   |   |   |   Horizontal connected one >= 3.5 : 1.2 (4/0) [1/1]
|   |   |   |   |   |   |   |   |   Vertical connected one >= 1.5 : 1.07 (10/0) [4/0.25]
|   |   |   |   |   |   |   |   |   Vertical connected two + three >= -0.5
|   |   |   |   |   |   |   |   |   |   Vertical connected one < -0.5 : 1.29 (80/0.2) [28/0.2]
|   |   |   |   |   |   |   |   |   |   Vertical connected one >= -0.5
|   |   |   |   |   |   |   |   |   |   |   CenterControl < 0.5 : 1.13 (70/0.13) [36/0.08]
|   |   |   |   |   |   |   |   |   |   |   CenterControl >= 0.5 : 1.01 (54/0.02) [39/0]
|   |   Horizontal + Vertical >= 0.5 : 1.04 (160/0.03) [76/0.06]
```

Size of the tree : 47

Time taken to build model: 0.02 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.4132
Mean absolute error	0.3012
Root mean squared error	0.405
Relative absolute error	77.5847 %
Root relative squared error	91.8604 %
Total Number of Instances	1000

2. Weka Exploration: Play around with options in the J48. You can get it to be less selective.

We tried to change the “confidenceFactor” value in “weka.classifiers.trees.J48” options.

When we decrease this value to 0.1, we got the tree looks like the following:

```
Scheme:weka.classifiers.trees.J48 -C 0.1 -M 2
Relation:   out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:  1000
Attributes: 8
            winner
            PieceLeftCorner
            CenterControl
            Vertical connected two + three
            Horizontal connected two + three
            Horizontal + Vertical
            Vertical connected one
            Horizontal connected one
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----
: 1 (1000.0/263.0)

Number of Leaves  :    1
Size of the tree  :    1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      728           72.8   %
Incorrectly Classified Instances    272           27.2   %
Kappa statistic                    0.0337
Mean absolute error                 0.3792
Root mean squared error             0.4405
Relative absolute error             97.742   %
Root relative squared error        100.0612   %
Total Number of Instances         1000

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.967    0.943    0.742    0.967    0.84       0.543    1
          0.057    0.033    0.385    0.057    0.099    0.543    2
Weighted Avg.   0.728    0.704    0.648    0.728    0.645    0.543

=== Confusion Matrix ===

  a  b  <-- classified as
713 24 |  a = 1
248 15 |  b = 2
```

From the figure, we can tell that the tree has only one leaf and the accuracy is relatively low.

When we increase this value to 0.3, we got the tree looks like the following:

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.3 -M 2
Relation:      out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:      1000
Attributes:      8
                winner
                PieceLeftCorner
                CenterControl
                Vertical connected two + three
                Horizontal connected two + three
                Horizontal + Vertical
                Vertical connected one
                Horizontal connected one
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
-----

CenterControl = -3: 1 (9.0/4.0)
CenterControl = -2
|   PieceLeftCorner = 0
|   |   Vertical connected two + three = -2: 2 (0.0)
|   |   |   Vertical connected two + three = -1: 2 (12.0/2.0)
|   |   |   |   Vertical connected two + three = 0: 1 (14.0/6.0)
|   |   |   |   |   Vertical connected two + three = 1
|   |   |   |   |   |   Horizontal connected two + three = -2: 2 (2.0)
|   |   |   |   |   |   |   Horizontal connected two + three = -1: 1 (3.0)
|   |   |   |   |   |   |   |   Horizontal connected two + three = 0: 1 (4.0/1.0)
|   |   |   |   |   |   |   |   |   Horizontal connected two + three = 1: 1 (0.0)
|   |   |   |   |   |   |   |   |   |   Horizontal connected two + three = 2: 1 (0.0)
|   |   |   |   |   |   |   |   |   |   |   Vertical connected two + three = 2: 2 (0.0)
|   |   |   |   |   |   |   |   |   |   |   |   PieceLeftCorner = 1: 2 (56.0/16.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   PieceLeftCorner = 2: 1 (15.0/3.0)
CenterControl = -1
|   Horizontal + Vertical = -3: 2 (1.0)
|   Horizontal + Vertical = -2: 2 (18.0/7.0)
|   Horizontal + Vertical = -1
|   |   Vertical connected two + three = -2: 1 (0.0)
|   |   |   Vertical connected two + three = -1: 2 (24.0/10.0)
|   |   |   |   Vertical connected two + three = 0: 1 (44.0/15.0)
|   |   |   |   |   Vertical connected two + three = 1: 1 (6.0/2.0)
|   |   |   |   |   |   Vertical connected two + three = 2: 1 (0.0)
|   |   Horizontal + Vertical = 0
|   |   |   Vertical connected one = -4: 2 (0.0)
|   |   |   |   Vertical connected one = -3: 2 (4.0)
|   |   |   |   |   Vertical connected one = -2: 2 (16.0/6.0)
|   |   |   |   |   |   Vertical connected one = -1
|   |   |   |   |   |   |   |   Vertical connected two + three = -2: 2 (0.0)
```

```

| | | Vertical connected two + three = -1: 2 (0.0)
| | | Vertical connected two + three = 0: 2 (14.0/4.0)
| | | Vertical connected two + three = 1: 1 (4.0/1.0)
| | | Vertical connected two + three = 2: 2 (0.0)
| | Vertical connected one = 0: 1 (30.0/15.0)
| | Vertical connected one = 1: 1 (12.0/3.0)
| | Vertical connected one = 2
| | | Horizontal connected one = -5: 1 (0.0)
| | | Horizontal connected one = -4: 1 (0.0)
| | | Horizontal connected one = -3: 2 (1.0)
| | | Horizontal connected one = -2: 1 (1.0)
| | | Horizontal connected one = -1: 2 (3.0/1.0)
| | | Horizontal connected one = 0: 1 (2.0)
| | | Horizontal connected one = 1: 2 (1.0)
| | | Horizontal connected one = 2: 1 (0.0)
| | | Horizontal connected one = 3: 1 (0.0)
| | | Horizontal connected one = 4: 1 (0.0)
| | | Horizontal connected one = 5: 1 (0.0)
| | Vertical connected one = 3: 2 (0.0)
| Horizontal + Vertical = 1: 1 (45.0/13.0)
| Horizontal + Vertical = 2: 1 (10.0)
| Horizontal + Vertical = 3: 1 (0.0)
CenterControl = 0: 1 (293.0/63.0)
CenterControl = 1: 1 (253.0/24.0)
CenterControl = 2: 1 (88.0/6.0)
CenterControl = 3: 1 (13.0/1.0)
CenterControl = 4: 1 (2.0)

Number of Leaves :      49

Size of the tree :      58

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances      759           75.9   %
Incorrectly Classified Instances    241           24.1   %
Kappa statistic                    0.3043
Mean absolute error                 0.3203
Root mean squared error             0.4178
Relative absolute error              82.5665 %
Root relative squared error         94.8895 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===

      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.896     0.624     0.801     0.896     0.846     0.709     1
      0.376     0.104     0.563     0.376     0.451     0.709     2
Weighted Avg.   0.759     0.487     0.738     0.759     0.742     0.709

=== Confusion Matrix ===

  a  b  <-- classified as
660 77 |  a = 1
164 99 |  b = 2

```

From the figure above, we can tell that the tree becomes much bigger, also, the accuracy of the classification result increases.

3. Compare your attempts to other ways of doing feature selection

We also tried some other methods of doing feature selection.

We tried the “FilteredAttributeEval” method and the result we got was shown as the following:

```
=== Run information ===

Evaluator:      weka.attributeSelection.FilteredAttributeEval -W "weka.attributeSelection.InfoGainAttributeEval" -F "weka.filters.supervised.instance.SpreadSubsample -M 0.0 -X 0.0 -S 1"
Search:weka.attributeSelection.Ranker -I -1.7976931348623157E308 -N -1
Relation:      out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:     1000
Attributes:    8
               winner
               PieceLeftCorner
               CenterControl
               Vertical connected two + three
               Horizontal connected two + three
               Horizontal + Vertical
               Vertical connected one
               Horizontal connected one
Evaluation mode:evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
  Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 winner):
Filtered Attribute Evaluator
Filter: weka.filters.supervised.instance.SpreadSubsample -M 0.0 -X 0.0 -S 1
Attribute evaluator: weka.attributeSelection.InfoGainAttributeEval

Filtered header:
@relation out-weka.filters.unsupervised.attribute.Remove-R1-42-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.supervised.instance.SpreadSubsample-M0.0-X0.0-S1

@attribute winner {1,2}
@attribute PieceLeftCorner {0,1,2}
@attribute CenterControl {-3,-2,-1,0,1,2,3,4}
@attribute 'Vertical connected two + three' {-2,-1,0,1,2}
@attribute 'Horizontal connected two + three' {-2,-1,0,1,2}
@attribute 'Horizontal + Vertical' {-3,-2,-1,0,1,2,3}
@attribute 'Vertical connected one' {-4,-3,-2,-1,0,1,2,3}
@attribute 'Horizontal connected one' {-5,-4,-3,-2,-1,0,1,2,3,4,5}

@data

Ranked attributes:
0.1141    3 CenterControl
0.0761    6 Horizontal + Vertical
0.0414    2 PieceLeftCorner
0.0383    4 Vertical connected two + three
0.0354    8 Horizontal connected one
0.0353    5 Horizontal connected two + three
0.0225    7 Vertical connected one

Selected attributes: 3,6,2,4,8,5,7 : 7
```

From the rank, we can tell that this is the same as the “ChiSquaredAttributeEval” and is almost the same as our prediction.