**TEAM 3 FINAL PROJECT REPORT**

**POKER HAND PREDICTION**

I. Introduction:

1. Problem Description:

Poker, whether it be a five card draw, or Texas hold'em or any of the other poker games is a billion-dollar industry and is ever expanding. Some say it is a game of luck while other agree that it involves a lot of skill. We are trying to find the best possible hand when given a set of five cards dealt from a fair, shuffled deck.

2. Motivation:

The main motivation behind this project was the application of data mining algorithms along with the use of analytical thinking approach to solve a problem. In the process of the project completion, we would also get familiar to various processes involved in data mining such as data gathering, pre-processing, exploration and various types of algorithms such as classification, clustering, association rules.

Our motivation behind choosing this problem and this dataset was very straight forward. The data set was simple in structure (just numeric values), there were no complex fields, it had sufficient number of instances (overfitting or underfitting is avoided). This problem would also help us in achieving the project objective.

3. Description:

The report first starts with data exploration wherein we have described how we came to choose the dataset to work on. The Data Pre-processing; we have stated how we cleaned the dataset before applying the algorithm to make the model. Which algorithm gave the best result. The performance measure we used namely, the 10-fold cross validation. We have then gone on to write down our experiences using the Logic of Thinking Procedure. The advantages of using it and its benefits we bore. Finally, we concluded by comparing the models acquired by using the various models. Mentioning the patterns we discovered and which model works best and also the results we came upon.

II. Data Exploration:

The dataset used came from the University of California Irvine’s Machine Leaning repository. We have used for our dataset, twenty-five thousand instances. The dataset contains eleven attributes out of which one is the goal or the class value. After the feedback of the first checkpoint we were planning to add three more attributes out of which two were predictive and one was class value. The attributes to be added were Opponent Expression, opponent Decision and Player Decision. The dataset had no outlier. The only missing values in this dataset were the ones that the model would predict after applying the algorithm.

III. Methodology:

1] Data Pre-processing:

We had stated in the first checkpoint that our dataset needed no pre-processing. Upon looking at our dataset in a data mining environment like WEKA and studying the dataset, we observed that a few pre-processing steps were required.

Missing Data:

Field Names: The dataset was missing a header row, so all fields including the class field had no labels. The Field Names had to be manually entered. The field names entered were as follows Suite1, Rank1, Suite2, Rank2, Suite3, Rank3, Suite4, Rank4, Suite5, Rank5, Poker Hand.

Discretization:

Field Data Type change (Numeric to Nominal): Field – Poker Hand: By looking at the data types of the fields in the WEKA, we observed that all the fields had a numeric data type (even the class value field). This would have created a problem while building a model as the results would not represent accurate classification. Therefore, we used the NumericToNominal unsupervised attribute filter in WEKA and applied it to the Poker Hand field (class value).

2] Mining the Data:

Classification: J48: Correctly classified instances: 71.9% (Class 0: 11676, Class 1: 6168, Class 2: 116, Class 3: 13, Class 4: 12, Class 5: 2, Class 6: 0, Class 7: 0, Class 8: 1, Class 9: 1)

Parameters: Confidence level: 0.25 (default)

Minimum Leaf nodes: 2 (default)

Classification: JRip: Correctly classified instances: 52.3% (Class 0: 12440, Class 1: 522, Class 2: 28, Class 3: 67, Class 4: 0, Class 5: 23, Class 6: 0, Class 7: 2, Class 8: 0, Class 9: 0)

Parameters: folds: 3

usePruning: True

minimumInstanceWeight: 2.0

Classification: PART: Correctly classified instances: 73.6% (Class 0: 11340, Class 1: 6785, Class 2: 174, Class 3: 103, Class 4: 6, Class 5: 0, Class 6: 0, Class 7: 0, Class 8: 0, Class 9: 0)

Parameters: folds: 3

confidence level: 0.25

minimumInstances per rule: 2

Classification: Bagging: Correctly classified instances: 89.8% (Class 0: 12337, Class 1: 9496, Class 2: 384, Class 3: 215, Class 4: 14, Class 5: 8, Class 6: 7, Class 7: 2, Class 8: 1, Class 9: 2)

Parameters: batchSize: 1000

classifier: PART

numIterations: 10

Classification: AdaBoost: Correctly classified instances: 99.6% (Class 0: 12475, Class 1: 10532, Class 2: 1196, Class 3: 512, Class 4: 93, Class 5: 53, Class 6: 36, Class 7: 6, Class 8: 5, Class 9: 5)

Parameters: classifier: PART

numIteraions: 10

Performance measures:

Cross-validation 10-fold was the option applied for every algorithm while training. This performance measure was used as 10-fold validation because this value through experimentation has been known to give low bias and variance. This is one of the most useful and measures to deploy and helps in building a predictive model.

Combined Findings:

1] AdaBoosting was the most accurate in predicting correct poker hand.

2] JRip was the least accurate of them all.

3] The class value 6 (Full house) was the most difficult to predict by all models.

4] The class value 0 (High card) was the easiest to predict.

Also, after observing the results, the dataset fails to simulate real-world scenario. The dataset contains way more class 0 and class 1 values than in reality. This makes the dataset skewed in that direction. This could be resolved by taking a larger training set containing real world values (or closest we can get to the real-world scenario).

IV. Logic of the Problem:

During the first checkpoint, having selected the project we could only make assumptions of what processes would be required to predict the hands. It was rudimentary at best. The inference made was that a large dataset was needed to create the model. After the feedback of the second checkpoint the purpose of the problem was updated by adding about two attributes i.e. facial expressions and opponent’s decision. The point of view also changed quite a lot. This changed the implications and assumptions vastly. In the logic of problem, we updated the problem statement by adding about the predictive decision which will be made by the machine. The Data Pre-processing checkpoint was an eye opener as we learnt by adding new attributes, we were changing the ground rules of the dataset. This would have made it unfeasible. From the feedback in the first checkpoint which was, our problem was strictly statistics based(probability), we understood that although the feedback was apt, the possible values were too large, and it would still classify as a data mining problem.

The advantage that this tool garners is that it breaks the task at hand into pieces. This gives a level inspection seldom possible. To analyse a problem at hand in this manner is advantageous. No, we did not find any cons with this procedure as the steps such as implications, point of view, assumptions helped us understand the task at hand quite well. The procedure itself is meticulous and quite helpful.

No, we do not feel there is a need to change the procedure by changing the steps. We felt that the steps were appropriate and concise. These steps helped us in understanding the problem in a truly analytical way by breaking it down. The fact that this logic can also be updated as the project progresses is a bonus because it happens quite often during a project that you begin with certain purpose and objectives and towards the end of the project, the goals, results as well as requirements tend to change.

V. Conclusions:

After comparing the five models, we have concluded that the ADABoost algorithm is better for our problem with an accuracy of 99.6% followed by Bagging, PART, J48 and last JRip. We also found that the class 6 (Full House) was the most incorrectly predicted class by all models. Although we were unable to find the reason why. The class 0 (High card) was most correctly predicted by all the algorithms. The dataset does not depict real world scenario.

Future work: By adding more instances, a real world scenario can be simulated and the dataset would be more complete than it now. These model can further be used on a similar dataset with a few added attributes, to simulate a game of poker between the player and computer.

Appendix:

Model outputs:

1] ADABoost:

User supplied test set

Relation: poker\_training-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: unknown (yet). Reading incrementally

Attributes: 11

=== Summary ===

Correctly Classified Instances 24913 99.6122 %

Incorrectly Classified Instances 97 0.3878 %

Kappa statistic 0.9932

Mean absolute error 0.0072

Root mean squared error 0.034

Total Number of Instances 25010

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.999 0.005 0.995 0.999 0.997 0.993 1.000 1.000 0

0.994 0.002 0.997 0.994 0.995 0.992 1.000 1.000 1

0.992 0.000 1.000 0.992 0.996 0.996 1.000 1.000 2

0.998 0.000 1.000 0.998 0.999 0.999 1.000 1.000 3

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 4

0.981 0.000 1.000 0.981 0.991 0.991 1.000 1.000 5

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 6

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 7

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 8

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 9

Weighted Avg. 0.996 0.004 0.996 0.996 0.996 0.993 1.000 1.000

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

12475 18 0 0 0 0 0 0 0 0 | a = 0

67 10532 0 0 0 0 0 0 0 0 | b = 1

0 10 1196 0 0 0 0 0 0 0 | c = 2

0 1 0 512 0 0 0 0 0 0 | d = 3

0 0 0 0 93 0 0 0 0 0 | e = 4

1 0 0 0 0 53 0 0 0 0 | f = 5

0 0 0 0 0 0 36 0 0 0 | g = 6

0 0 0 0 0 0 0 6 0 0 | h = 7

0 0 0 0 0 0 0 0 5 0 | i = 8

0 0 0 0 0 0 0 0 0 5 | j = 9

2] Bagging:

User supplied test set

Relation: poker\_training-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: unknown (yet). Reading incrementally

Attributes: 11

=== Summary ===

Correctly Classified Instances 22466 89.8281 %

Incorrectly Classified Instances 2544 10.1719 %

Kappa statistic 0.8133

Mean absolute error 0.0663

Root mean squared error 0.1528

Total Number of Instances 25010

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.988 0.106 0.903 0.988 0.943 0.886 0.992 0.991 0

0.896 0.084 0.887 0.896 0.891 0.811 0.972 0.956 1

0.318 0.000 0.982 0.318 0.481 0.549 0.993 0.924 2

0.419 0.000 0.973 0.419 0.586 0.634 0.999 0.966 3

0.151 0.000 1.000 0.151 0.262 0.387 0.999 0.937 4

0.148 0.000 1.000 0.148 0.258 0.385 0.999 0.906 5

0.194 0.000 1.000 0.194 0.326 0.441 1.000 0.997 6

0.333 0.000 1.000 0.333 0.500 0.577 1.000 0.892 7

0.200 0.000 1.000 0.200 0.333 0.447 1.000 0.798 8

0.400 0.000 1.000 0.400 0.571 0.632 1.000 0.967 9

Weighted Avg. 0.898 0.088 0.902 0.898 0.887 0.829 0.983 0.972

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

12337 156 0 0 0 0 0 0 0 0 | a = 0

1098 9496 5 0 0 0 0 0 0 0 | b = 1

98 724 384 0 0 0 0 0 0 0 | c = 2

26 271 1 215 0 0 0 0 0 0 | d = 3

54 25 0 0 14 0 0 0 0 0 | e = 4

38 8 0 0 0 8 0 0 0 0 | f = 5

1 25 1 2 0 0 7 0 0 0 | g = 6

0 0 0 4 0 0 0 2 0 0 | h = 7

4 0 0 0 0 0 0 0 1 0 | i = 8

3 0 0 0 0 0 0 0 0 2 | j = 9

3] PART

User supplied test set

Relation: poker\_training-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: unknown (yet). Reading incrementally

Attributes: 11

=== Summary ===

Correctly Classified Instances 18408 73.6026 %

Incorrectly Classified Instances 6602 26.3974 %

Kappa statistic 0.5074

Mean absolute error 0.0737

Root mean squared error 0.192

Total Number of Instances 25010

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.908 0.328 0.734 0.908 0.812 0.597 0.867 0.842 0

0.640 0.164 0.741 0.640 0.687 0.488 0.815 0.783 1

0.144 0.003 0.680 0.144 0.238 0.300 0.927 0.404 2

0.201 0.002 0.705 0.201 0.313 0.370 0.955 0.403 3

0.065 0.000 0.500 0.065 0.114 0.179 0.990 0.238 4

0.000 0.000 ? 0.000 ? ? 0.986 0.115 5

0.000 0.000 ? 0.000 ? ? 0.992 0.144 6

0.000 0.000 ? 0.000 ? ? 0.999 0.197 7

0.000 0.000 ? 0.000 ? ? 0.999 0.083 8

0.000 0.000 ? 0.000 ? ? 0.999 0.082 9

Weighted Avg. 0.736 0.234 ? 0.736 ? ? 0.850 0.782

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

11340 1139 11 3 0 0 0 0 0 0 | a = 0

3742 6785 47 21 4 0 0 0 0 0 | b = 1

162 860 174 10 0 0 0 0 0 0 | c = 2

97 298 15 103 0 0 0 0 0 0 | d = 3

53 22 6 6 6 0 0 0 0 0 | e = 4

39 14 0 1 0 0 0 0 0 0 | f = 5

6 27 2 1 0 0 0 0 0 0 | g = 6

0 5 0 1 0 0 0 0 0 0 | h = 7

3 1 0 0 1 0 0 0 0 0 | i = 8

2 1 1 0 1 0 0 0 0 0 | j = 9

4] J48:

User supplied test set

Relation: poker\_training-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: unknown (yet). Reading incrementally

Attributes: 11

=== Summary ===

Correctly Classified Instances 17989 71.9272 %

Incorrectly Classified Instances 7021 28.0728 %

Kappa statistic 0.4686

Mean absolute error 0.0755

Root mean squared error 0.1943

Total Number of Instances 25010

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.935 0.402 0.699 0.935 0.800 0.566 0.856 0.840 0

0.582 0.136 0.759 0.582 0.659 0.470 0.814 0.784 1

0.096 0.001 0.829 0.096 0.172 0.273 0.906 0.359 2

0.025 0.000 1.000 0.025 0.049 0.158 0.931 0.206 3

0.129 0.000 0.750 0.129 0.220 0.310 0.989 0.344 4

0.037 0.000 1.000 0.037 0.071 0.192 0.988 0.191 5

0.000 0.000 ? 0.000 ? ? 0.983 0.081 6

0.000 0.000 ? 0.000 ? ? 0.993 0.052 7

0.200 0.000 1.000 0.200 0.333 0.447 0.997 0.320 8

0.200 0.000 1.000 0.200 0.333 0.447 1.000 0.508 9

Weighted Avg. 0.719 0.259 ? 0.719 ? ? 0.843 0.775

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

11676 816 0 0 1 0 0 0 0 0 | a = 0

4414 6168 16 0 1 0 0 0 0 0 | b = 1

363 727 116 0 0 0 0 0 0 0 | c = 2

136 356 8 13 0 0 0 0 0 0 | d = 3

62 19 0 0 12 0 0 0 0 0 | e = 4

47 5 0 0 0 2 0 0 0 0 | f = 5

5 31 0 0 0 0 0 0 0 0 | g = 6

0 6 0 0 0 0 0 0 0 0 | h = 7

3 0 0 0 1 0 0 0 1 0 | i = 8

1 2 0 0 1 0 0 0 0 1 | j = 9

5] JRip:

User supplied test set

Relation: poker\_training-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: unknown (yet). Reading incrementally

Attributes: 11

=== Summary ===

Correctly Classified Instances 13082 52.3071 %

Incorrectly Classified Instances 11928 47.6929 %

Kappa statistic 0.0561

Mean absolute error 0.1105

Root mean squared error 0.235

Total Number of Instances 25010

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.996 0.938 0.515 0.996 0.678 0.162 0.529 0.515 0

0.049 0.012 0.757 0.049 0.092 0.113 0.523 0.446 1

0.023 0.001 0.651 0.023 0.045 0.117 0.533 0.076 2

0.131 0.000 0.957 0.131 0.230 0.350 0.599 0.170 3

0.000 0.000 ? 0.000 ? ? 0.517 0.004 4

0.426 0.000 0.885 0.426 0.575 0.613 0.722 0.387 5

0.000 0.000 ? 0.000 ? ? 0.570 0.017 6

0.333 0.000 0.667 0.333 0.444 0.471 0.758 0.268 7

0.000 0.000 ? 0.000 ? ? 0.613 0.009 8

0.000 0.000 ? 0.000 ? ? 0.709 0.036 9

Weighted Avg. 0.523 0.473 ? 0.523 ? ? 0.529 0.454

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

12440 53 0 0 0 0 0 0 0 0 | a = 0

10065 522 12 0 0 0 0 0 0 0 | b = 1

1103 75 28 0 0 0 0 0 0 0 | c = 2

405 37 3 67 0 0 0 1 0 0 | d = 3

93 0 0 0 0 0 0 0 0 0 | e = 4

31 0 0 0 0 23 0 0 0 0 | f = 5

31 3 0 2 0 0 0 0 0 0 | g = 6

3 0 0 1 0 0 0 2 0 0 | h = 7

4 0 0 0 0 1 0 0 0 0 | i = 8

3 0 0 0 0 2 0 0 0 0 | j = 9

References:

1] “Elements and standards learning tool” [online] Available at: <http://www.criticalthinking.org/ctmodel/logic-model.php> [Accessed last on 20 April 2019]

2] “Weka 3 - Data Mining with Open Source Machine Learning Software in Java” [online] Available at: <https://www.cs.waikato.ac.nz/ml/weka/documentation.html> [Accessed on 03 March 2019, 10 March 2019]

3] “Tableau Help | Tableau Software” [online] Available at: <https://www.tableau.com/support/help> [Accessed on 14 April 2019, 16 April 2019]