Anticipation of Winning Probability in Poker Using Data Mining

Shiben Sheth[1], Gaurav Ambekar[2], Abhilasha Sable[3], Tushar Chikane[4], Kranti Ghag[5]

[1,2,3,4] B.E Student, SAKEC, Chembur, Department of Information Technology, Mumbai University, India

[5] Assistant Professor, Information Technology, Shah and Anchor College of Engineering, Mumbai, India

Email Id: shiben20@gmail.com; abhilasha.sbl@gmail.com; ambekargaurav@gmail.com; tushar.c988@gmail.com; krantiis@rediffmail.com

Abstract— *Poker is one of the world's most popular and widely played card games. In Poker, there is a fixed set of winning conditions and the player with the highest winning condition wins the game. The main part of the game is to bet strategically and in a calculated manner so that there is less chance of risk and the opponents are not able to guess the cards in the hand. To help players understand when and how to bet smartly, this application will be developed. This system provides knowledge to the users about their probability of winning based on the cards available to them. The system which has been developed is lightweight and easy-to-use so that all types of players can use it. The aim of this system is to help gamblers bet better thereby increasing their winnings, addiction to Poker gambling and also generate greater revenue collections for gaming consortiums. There are numerous Poker tournaments held all over the world for which players travel long distances for a chance to win big pay-outs and also the fan-following of the game is crazy. The most important point of this paper is to show how we have used data mining and statistical probabilities to formulate an algorithm which gives out correct predictions of the winning hand. We formally define the system and outline the challenges that arose while developing technology to support it. We hope that this paper will encourage more research by the gaming consortiums and the gambling community in this exciting area of winning by probability calculations and card counting.*

Keywords— *Data mining, Machine learning, Poker, Winning Probability, Naïve Bayes Algorithm.*

INTRODUCTION

Data mining is the computational process of discovering patterns in large data sets can also be defined as the extraction of hidden predictive information from large databases. The overall goal of the data mining process is to extract information from a data set and transform it into an easily understandable structure for further use by various skilled users which involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, online updating, but also visualization. Data mining is a powerful technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions without large dependence on older methods like focus groups.

Poker is a game that caught the interest of the AI research community in the last decade. This game presents a radically different challenge to other games like chess where both players are always aware of the full state of the game. This means that it is possible somehow to understand the opponent's strategy by observing the movement of the game pieces. On the contrary, Poker game state is hidden: each player can only see his cards or the community cards. It is only at the end of each game that opponents may show their cards, thus being much more difficult to understand how the opponent plays.

Poker is also a stochastic game, i.e., it admits the element of chance since the player cards are randomly dealt. The following are the most important properties of poker:-

1) Imperfect information - This property creates a necessity for using and coping with deception and ensures a theoretical advantage of using randomized mixed strategies.

2) Non-deterministic dynamics - This means that the cards we get are stochastic.

3) Partial observable - Players can't always know the opponent's hole cards, even when a game is over.

4) Multi-players - There are at least two players.

There are 6 popular types of Poker that are played world-wide [1]:-

Omaha:-Omaha is a type of Hold 'Em that can be played by 2-10 players at a time. Players must make their best 5-card hands from two of their hole cards and three of the common.

7-Card Stud:-In 7-Card Stud, each player is dealt 7 cards, three down and four up. Players must make best possible 5-card hand from their 7.

5-Card Draw:-Each player is dealt 5 cards, but on the initial go around, the player may choose to trade in up to 3 of them.

High / Low Chicago:-This stud game can be played for the highest hand or the lowest. In High Chicago, the player with the highest spade face-down wins half the pot. In Low Chicago, the player with the lowest spade face-down wins half the pot. This game can be added to, and played simultaneously with, many other poker variations.

Follow the Queen:-This is a 7-card stud poker game in which the wild card is designated to be the next exposed card after a queen is flipped. If no queens are flipped, there are no wild cards that hand.

Texas Hold 'em:-Played in the World Series of Poker, Texas Hold 'Em is easily the most popular poker game. In Texas Hold 'Em, players are dealt two "pocket" or "hole cards" then wait for 5 community cards to be revealed. Betting takes place in four rounds: once after the hole cards are dealt, once after the first three community cards are revealed (referred to as "the flop"), once after the fourth community card is revealed ("the turn") and lastly after the fifth community card is flipped ("the river"). A showdown occurs after the river where the remaining players reveal their hole cards and the player with the best hand wins all the wagers in the pot. If two or more players have the same best hand then the pot is split amongst the winners. Players must make their best hands with any combination of 5 cards (their hole cards and the communal). We made use of this type of poker to predict the winning probability in the game.

LITERATURE SURVEY

Machine learning [2] investigates how computers can learn (or improve their performance) based on the data. A main research area is for computer programs to automatically learn to recognize complex patterns and make intelligent decisions based on data. Machine learning focuses on prediction, based on known properties learned from the training data. Data mining focuses on the discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.

Poker [3] is usually played with a standard deck of 52 cards. Each card is marked with one of 13 face values and one of 4 suits. In a common version of poker, a player receives a hand of five cards. Hands that match certain combinations, or patterns, have specific names like "FULL HOUSE" or "ROYAL FLUSH". When it comes to obtaining hand history data, poker sites can be grouped into three main classes:

- Those that record hand histories in a way that can be incorporated into a database without requiring additional software.
- Those that do not automatically record observed hand history data, but for which this data may be obtained using a separate software program.
- Those for which observed hand histories cannot at this point in time be obtained (though software do permit this, may well in time be developed).

BetOnline[4] is the most popular of all real-money poker sites those are available to U.S. players.

Play covers all 50 states in the U.S. PokerTrackerSoftware [5] LLC is the name of a poker tool software company that produces the popular PokerTracker line of poker tracking and analysis software. PokerTracker's software imports and parses the hand histories that poker sites create during online play and stores the resulting statistics/information about historical play into a local database library for self-analysis, and for in-game opponent analysis using a real-time Head-up display. The software allows the user to monitor each poker session's profit or loss, hands played, time played, and table style. It calculates and graphs statistics such as hands per hour, winnings per hand, wins per hour, cumulative profit and loss, and individual game profit and loss across multiple currencies.

A poker sites calculator is an application that lets you run any scenario that you see at a poker table. Once you say what cards you have, and what cards other players have, the poker calculator will go to work and, in a matter of seconds, tell you what your odds of winning are. There are no guarantees but, in the long run, using the kind of statistical information you get from a poker odds calculator can give you a real edge over players that don't realize what they're missing out on. PokerListings.com's Odds Calculator[6] is the fastest, most accurate and easy-to-use poker odds calculator on the Web! Know exactly what your chances of winning are at any point in a hand and make your decisions easier.

Table 1

Probability statistics [7]

Poker Hand	# of hands	Probability	# of
Royal Flush	4	0.00000154	480
Straight Flush	36	0.00001385	432
Four of a kind	624	0.0002401	748
Full house	3744	0.00144058	449
Flush	5108	0.0019654	612
Straight	10200	0.00392464	122
Three of a kind	54912	0.02112845	658
Two pairs	123552	0.04753902	148
One pair	1098240	0.42256903	131
Nothing	1302540	0.50117739	156
Total	2598960	1.0	311

The number of combinations represents the number of instances in the entire domain.

PROPOSED WORKING SYSTEM

The analysis and prediction of the best possible winning hand combination depending on the cards the user has in his hands has been calculated by the system. The user can enter 2-5 cards and the best possible winning hand will be displayed. The winning hand displayed will be according to the winning hand ranking combinations. They are shown below:

Figure 1

We have taken the dataset (training and test data) and used it to predict the class (i.e. the winning hand rank) in which they fall under. The final model of our project was decided after the pre-processing done before experimentation.

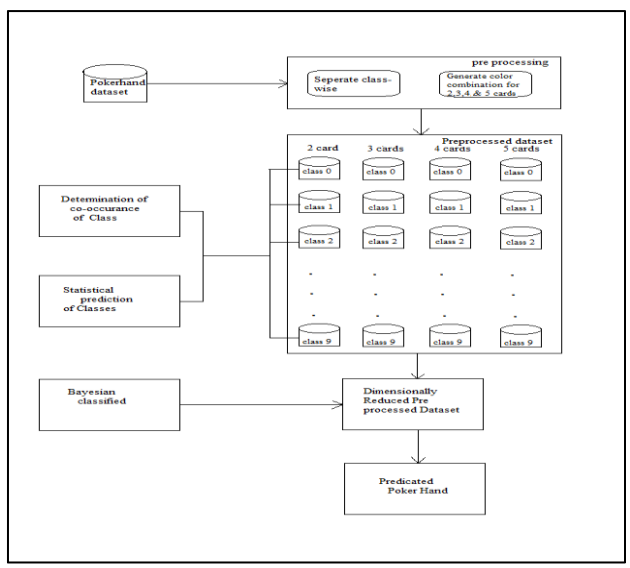

Figure 2

The **poker hand dataset** is obtained from machine learning site. This dataset is pre-processed before applying any suitable algorithm of classification. The **pre-processing block** involves: Adjusting Dataset and manipulating data as their priority in a deck of cards. Class-wise separation of data and forming different card combinations from the given set of card attributes. Class-wise separation of data means segregating out instances of each class value. Different card combinations dataset of 2 cards, 3 cards and 4 cards are prepared. The next step involves **determination of co-occurrence of classes**. This can be elaborated as: for a given set of card combination, find out the different classes which occur simultaneously. This is done for every card combination. The co-occurrence of each class is shown in the matrices for 2, 3, 4 cards combinations. **Statistical prediction of classes:** Hardcode conditions are implemented. This step helps to select classes from all the classes. These are conditions developed from the reasons and logic we understand while playing the game. **Bayesian classifier** along with statistical probabilities is used. The pre-processed dataset along with the modified Bayesian classifier predicts the class for a given card combination.

EXPERIMENTS PERFORMED

The first experiment was combining the attributes. In the dataset we have combined the first two numbers to show the suit number and card number. The number at the end of each line depicts the class value given to a set of i.e. the ranking of cards according to winning hand order.

1	10	1	11	1	12	1	13	1	1	9
110		111		112		113		101		9
10 of spades		Jack of spades		Queen of spades		King of spades		Ace of spades		Royal Flush Score 9

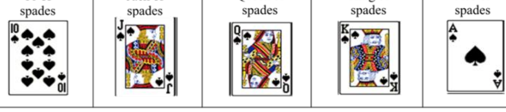

Figure 3

*Suit *100 + Card value= Card (Attribute)*

In the second experiment, interchanging of the suit and card attributes was done.

In this step, we have interchanged the positions of the suit number and card number. The card number is written first and then the suit number is written.

1	10	1	11	1	12	1	13	1	1	9
101		111		121		131		11		9
10 of spades		Jack of spades		Queen of spades		King of spades		Ace of spades		Royal Flush Score 9

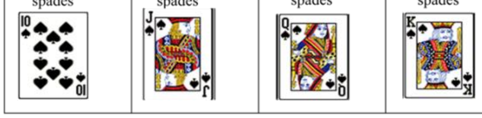

Figure 4

*Card value*100 + Suit = Card (Attribute)*

For the third experiment, altering priority level for suits was performed.

1	10	1	11	1	12	1	13	1	1	9
104		114		124		134		144		9
10 of spades		Jack of spades		Queen of spades		King of spades		Ace of spades		Royal Flush Score 9

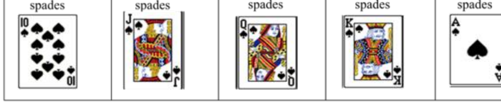

Figure 5

i. Spade : 1 → 4
   Hearts: 2 → 3
   Diamonds : 3 → 2
   Clubs : 4 → 1
ii. Ace : changed from 1 to 14 being the highest order

Step 4: Class wise division of cards (0-9)
Combinations: Dataset is divided into following combinations of 2-Cards, 3-Cards and 4-Cards. It is done to formulate various strategies that are applicable to specific conditions. The combinations are such that they can be operated on easily due to size reduction of the entire dataset.

In the 4th experiment, we formulated algorithms for each class on basis of judgemental analysis. The analysis includes basic knowledge of the game for predicting the class of your cards. It helped to generate hardcode selection of classes from the set of the 10. This indirectly helped to increase the probability. For 10 classes probability of one class is 0.1 by reducing class probability increases to 0.25(if down to 4 classes).

The class with the highest probability is the predicted by our algorithm. The Accuracy of Bayesian Algorithm was increased as we used only the selected datasets.

As all combinations were not present in the datasets that we had made, correct accuracy could not be obtained. Therefore, we combined training and test dataset to form a much larger dataset with possibly all entries which was proved upon experimenting. After joining, the final dataset was again split into previous formats (2, 3, 4 or 5 card combinations each containing 10 datasets based on class values).

RESULTS AND DISCUSSIONS

In experiment 1, we first transformed the attributes of dataset according to the decision model requirement. The suit and card no. attributes are combined into a single attribute. Conversion of numeric data set to nominal data set. The conversion was done to make the dataset workable in Weka. The values were adjusted as per needed. The data set was also disintegrated into 4 parts they are:

i. Column of two cards and score

ii. Column of three cards and score

iii. Column of four cards and score

iv. Column of five cards and score

*Suit *100 + Card value= Card (Attribute)*

These datasets were obtained by hardcoding in java to separate the card combinations. In the dataset we have combined the first two numbers to show the suit number and card number. The number at the end of each line depicts the class value given to a set of i.e. the ranking of cards according to winning hand order. After the combinations were done, they were then input to Weka and various decision trees were applied to it. Decision stump, FT tree, M5P and J48 were applied to it. This transformation was done to make the computations that were to be performed easier. The combination done is for the dataset to be readable and understandable to the compiler.

The biggest upside to combining was that the number of attributes was reduced to 6 from 11. This helped the compiler to read less number of attributes without the meaning of the attributes being changed. But, they did not avail the necessary output. The rules obtained were insufficient as not all the values were classified correctly. Besides the accuracy and support was very less for the result to be used for prediction in any manner. The accuracy of the generated decision tree should be high (>55%).Decision trees generated did not satisfy the solution requirements. Decision tree had to be modified as per requirements or the dataset had to be further transformed. Furthermore, the dataset had to be manipulated and changed into a form which could yield better accuracy at least more than 60%. Therefore, it can be easily stated that this experiment was unsuccessful in obtaining the desired accuracy and so a new approach for transformation had to be generated and adopted.

Table 2: Accuracies of different classification algorithms

Algorithm	**Accuracy**
J48	49.95%
MP5	48.34%
REP	49.92%
Random forest	54.18%
Random Tree	50.25%
BF Tree	No answer
Decision stump	49.91%
LAD	49.90%
Naïve Bayesian	56.68%

In experiments two and three, we then further transformed the dataset further into 10 more datasets. This was done by dividing the previously obtained class combinations based on their class attribute value. Due to the separation of card combinations into class combinations, the datasets became smaller thereby reducing the time needed for computations in

Weka. These datasets were obtained by hardcoding in java to separate the classes. Again these datasets were input to Weka and the accuracy obtained was acceptable as it was greater than 60%.Now, we formulated algorithms for each class and hardcoded them in Java. The class datasets of all card combinations were loaded and experimented to view if the predictions were correct or not.

On experimenting, we found that not all predictions made were correct. After examining the code no error was found in it. Further after searching in the dataset for the card combination it was found that quite a lot of possible combinations were not in the dataset. This led us to find that not all possible combinations were present in the datasets. Due to unavailability of all possible combinations in the dataset, no accurate accuracy could be obtained making the dataset redundant. Therefore, some changes had to be made to the dataset so that all combinations were present (i.e. most likely the missing combinations had to be added manually making it very tedious, lengthy but also impractical in the long run due to the fact that each combination had to be searched before being added so as to reduce redundancy) or that the code had to be manipulated to only accept the combinations that were in the dataset. This would make the dataset biased as correct accuracy could not be obtained as predictions would be made on only those combinations that were in the dataset. Hence, some change had to be made and it was only logical to improve the dataset. But the hurdle was to make changes to the dataset so as to include all combinations without manually inserting the combinations.

Table 3: Card combinations

2 Cards	3 Cards	4 Cards
1-2, 1-3, 1-4, 1-5 2-3, 2-4, 2-5 3-4,3-5 4-5	1-2-3, 1-2-4, 1-2-5, 1-3-4, 1-3-5, 1-4-5 2-3-4, 2-3-5, 2-4-5 3-4-5	1-2-3-4 1-2-3-5 1-2-4-5 1-3-4-5 2-3-4-5

The cards combinations are such that each card is combined with 1, 2 or 3 so that all cards are combined in one or another combination.

Separating classes -Dataset is broken down into 10 distinct dataset for classes 0-9. The task is performed for all the 3 types of card combinations i.e. 2, 3 and 4 cards.

In experiment 4:

The selected classes were given as input to Bayesian Classifier Algorithm. The Algorithm gave probabilities of each class. Bayesian Algorithm was modified by equation 1.

$$P(CC,C1) = P(CC1/TC1)*P(C1) \quad ....eq(1)$$

Bayesian Formula to find probability

1. CC= no. of times the input has occurred in class 1 / total instances in class 1
2. TC1= total no. of class 1 instances in type of card combination(e.g. 4 cards)
3. P(C1) : count of class 1 instances in original dataset/ total instances

These probabilities were sorted in descending order and the class with the best probability is the predicted class.

Further training and test datasets are compared for the instances present. Instances absent in the training set are added from the test set.

Instances except for class 0 are added.

As the test data and training data were combined, we obtained a much larger dataset containing all possible card combinations. With this dataset, we used Naïve Bayesian classifier and got accuracy of 92% instead of the previous 56.68%.

This was also due to the fact that accuracy was calculated in parts and not as a whole. This increased the accuracy to acceptable levels with any hint of a bias. The accuracy calculated had to be done separately because the datasets were already separated and that the calculation would be easier and faster due to size reduction. We formulated the logic to get all possible classes and then found the probability among them. Conditions for finding the requirements of each winning hand for 2, 3 or 4 card combinations. In any card condition, there is always a possibility of class zero occurring.

Table 4: 2 cards

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
C0	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
C1	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
C2	✓	✓	-	✓	✓	✓	✓	✓	✓	✓
C3	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
C4	✓	✓	✓	✓	-	✓	✓	✓	✓	✓
C5	✓	✓	✓	✓	✓	-	✓	✓	✓	✓
C6	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
C7	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
C8	✓	✓	✓	✓	✓	✓	✓	✓	-	✗
C9	✓	✓	✓	✓	✓	✓	✓	✓	✗	-

Class 1 (one pair) = 100 % possibility
Class 2 (two pair) = 100 % possibility
Class 3 (3 of a kind) = 100 % possibility
Class 4 (straight) = difference between card numbers should be less than or equal to 4
Class 5 (flush) = both cards have to be from the same suit
Class 6 (full house) = 100 % possibility
Class 7 (four of a kind) = 100 % possibility
Class 8 (straight flush) = cards should be less than Ace having the same suit. Also the difference between the 2 cards is less than or equal to having card value 4.
Class 9 (royal flush) = cards should be from the same suit with card values greater than or equal to 10.

This table show the possibility of occurrence of each class with respect to each other class for 2 card combinations.

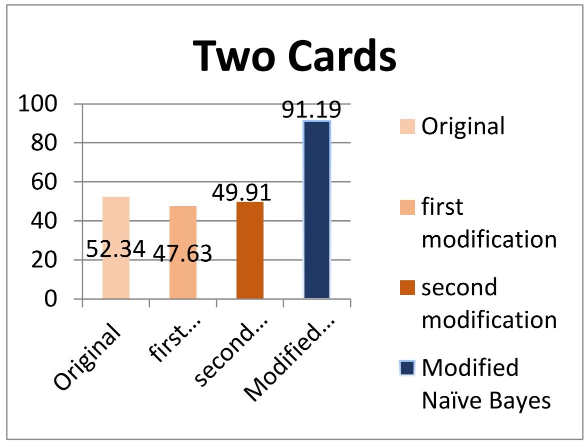

Figure 6

Accuracy by Naive Bayes: 52.34%

Accuracy by Modified Naïve Bayes: 91.19%

This proved that the pre-processing performed and the modified algorithm used was suitable for achieving the needed accuracy level for 2 card combination.

Table 5: 3 cards

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
C0	-	✓	✓	✓	✓	✓	✗	✗	✓	✓
C1	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
C2	✓	✓	-	✓	✓	✓	✓	✓	✓	✓
C3	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
C4	✓	✓	✓	✓	-	✓	✗	✗	✓	✓
C5	✓	✓	✗	✗	✗	-	✗	✗	✓	✓
C6	✗	✓	✓	✓	✗	✗	-	✓	✗	✗
C7	✗	✓	✓	✓	✗	✗	✓	-	✗	✗
C8	✓	✓	✓	✓	✓	✓	✗	✗	-	✓
C9	✓	✓	✓	✓	✓	✓	✗	✗	✓	-

Class 1 (one pair) = 100 % possibility
Class 2 (two pair) = 100 % possibility
Class 3 (3 of a kind) = 100 % possibility

Class 4 (straight) = difference between card values should be less than or equal to 4

Class 5 (flush) = cards have to be from the same suit

Class 6 (full house) = 2 out of 3 cards have to become a pair (i.e. one pair)

Class 7 (four of a kind) = 2 out of 3 cards have to become a pair (i.e. one pair)

Class 8 (straight flush) = cards should be less than Ace having the same suit. Also the difference between the 2 cards is less than or equal to having card value 4.

Class 9 (royal flush) = cards should be from the same suit with card values greater than or equal to 10.

This table show the possibility of occurrence of each class with respect to each other class for 3 card combinations.

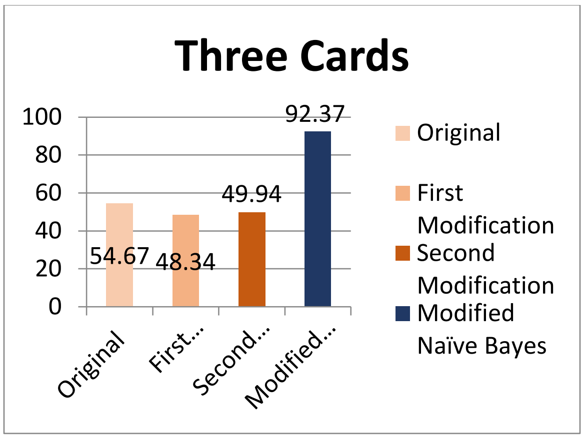

Figure 7

Accuracy by Naive Bayes: 54.67%

Accuracy by Modified Naïve Bayes: 92.37%

This proved that the pre-processing performed and the modified algorithm used was suitable for achieving the needed accuracy level for 3 card combination.

TABLE 6: 4 cards

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
C0	-	✓	✗	✗	✓	✓	✗	✗	✓	✓
C1	✓	-	✓	✓	✓	✓	✗	✗	✓	✓
C2	✗	✓	-	✓	✗	✗	✓	✗	✗	✗
C3	✗	✓	✓	-	✗	✗	✓	✓	✗	✗
C4	✓	✓	✗	✗	-	✗	✗	✗	✗	✗
C5	✓	✓	✗	✗	✗	-	✗	✗	✓	✓
C6	✗	✗	✓	✓	✗	✗	-	✓	✗	✗
C7	✗	✗	✗	✓	✗	✗	✓	-	✗	✗
C8	✓	✓	✗	✗	✗	✓	✗	✗	-	✗
C9	✓	✓	✗	✗	✗	✓	✗	✗	✗	-

Class 1 (one pair) = 100 % possibility

Class 2 (two pair) = 2 out of 4 cards have to be of the same kind (i.e. at least one-pair should be present)

Class 3 (3 of a kind) = 2 out of 4 cards have to be of the same kind (i.e. at least one-pair should be present)

Class 4 (straight) = difference between card values should be less than or equal to 4

Class 5 (flush) = cards have to be from the same suit

Class 6 (full house) = 2 out of 3 cards have to become a pair (i.e. one pair) or 3 out of 4 cards have to be 3 of a kind

Class 7 (four of a kind) = 3 out of 4 cards have to be 3 of a kind

Class 8 (straight flush) = cards should be less than Ace having the same suit. Also the difference between the 2 cards is less than or equal to having card value 4.

Class 9 (royal flush) = cards should be from the same suit with card values greater than or equal to 10.

This table show the possibility of occurrence of each class with respect to each other class for any and all 4 card combinations.

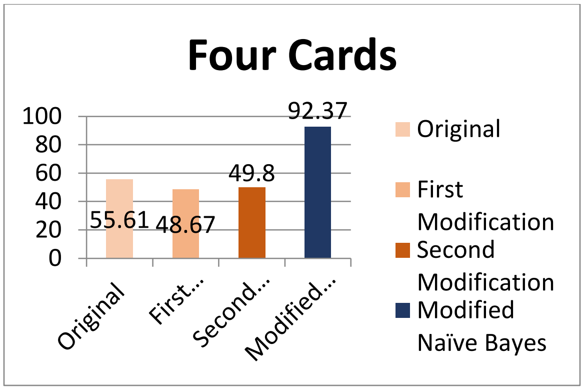

Figure 8

Accuracy by Naive Bayes: 55.61%

Accuracy by Modified Naïve Bayes: 92.37%

This proved that the pre-processing performed and the modified algorithm used was suitable for achieving the needed accuracy level for 4 card combination.

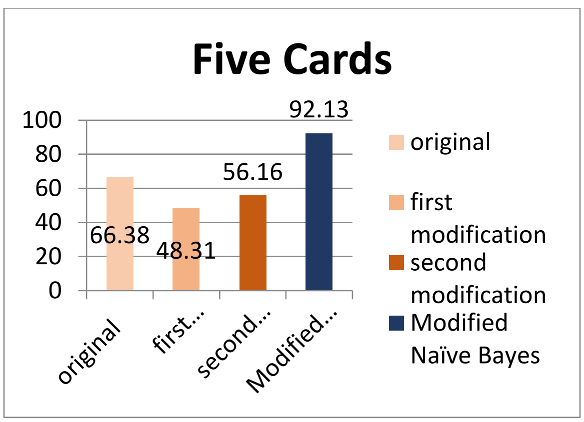

Figure 9

Accuracy by Naive Bayes: 66.38%

Accuracy by Modified Naïve Bayes: 92.13%

This proved that all the pre-processing done and the algorithm used was suitable for achieving the needed accuracy level after the final modifications.

Table 7: Dataset accuracy comparison

Data set	Five cards
Original	66.38%
First experimental modification	48.31%
Second experimental modification	56.16%
Final experimental modification	92.13%

CONCLUSION

Thus the importance of data mining techniques for predicting the winning hand possibility in poker has been clearly outlined in this paper. We have achieved an accuracy of 92.13% by calculating the probability of the cards the user has in his hands.

This paper depicts a clear view of the accuracy that we have achieved versus the accuracy that has been achieved by only using a statistical approach by directly using the dataset. It also examines the comparison of different transformations needed to achieve optimal accuracy. Also that the importance of transformation of the datasets necessary to achieve the highest accuracy using both the probabilistic statistical formulae and the cross-referencing of datasets to it for obtaining correct winning hand predictions has been effectively stated and reasoned.

We hope that this paper will gain momentum amongst poker companies as well as data mining research enthusiasts to study the necessity of using statistics for probability related data mining for various games.

REFERENCES

[1]. TeachingPoker.com (2012).*Types of Poker*. Available: http://teachingpoker.com/Types.html

[2] Han, Kamber, “*Data Mining concepts and techniques*”, Morgan Kauffman 2 edition. Amsterdam, NED: Illinious University, March 2006.

[3] Suquamish Clearwater Casino Resort(2014).*6 POPULAR TYPES OF POKER*[Blog]. Available:http://www.clearwatercasino.com/6-popular-types-of-poker/

[4] BetOnline.(2013). *Online Promotions*[Advertisement]. Available: http://www.betonline.ag/poker

[5] StevenM.(2010, Sept. 24). *Poker Tracker* [Blog]. Available: http://www.pokertracker.com/

[6] Cards Chat A worldwide poker community (2004). *Easy-to-use Poker Odds Calculator*[Blog]. Available: http://www.cardschat.com/poker-odds-calculator.php

[7] Robert Cattral. Carleton University(2002).*Poker hand Data Set* [Data]. Available: http://archive.ics.uci.edu/ml/machine-learning-databases/poker/poker-hand.names