Classification of Poker Hands Using Ordinal Response Multicategory Logit Model

Sara Bey and Jayanth Sivakumar
{sbey1,jsivaku1}@binghamton.edu

April 30, 2019

Contents

List of Figures

List of Tables

1 Introduction

What are the odds of getting a Royal Flush in poker? What about a Straight? Can we predict our chances of winning a poker game given a certain hand? In order to investigate these ideas, we will use the Poker Hand Data Set provided by the UCI Machine Learning Repository to carry out our analysis. This data set contains examples of poker hands, consisting of 5 cards, drawn from a standard deck of 52 cards. Each card in the hand is described using it's suit and rank, which gives a total of 10 predictors for a given poker hand. The suit of each card is represented by *S1, ... S5*, and the rank is represented by *C1, ..., C5*. The suit is given by an ordinal value of $1-4$, which represent Hearts, Spades, Diamonds, and Clubs, respectively. The rank is given by a numerical value of $1-13$, representing Ace, $2, 3, ...$, Queen, and King, respectively. The last piece of information in the data set is called the *Class* of the poker hand. The *Class* is represented by an ordinal value between 0 and 9, where each *Class* represents a certain poker hand that a player can receive, as described below in Table 1. There are a total of $1,025,010$ total poker hands, $1,000,000$ in the test set and $25,010$ in the training set, provided in the Poker Hand Data set (Dua, Dheeru and Graff, Casey 2017).

Table 1: Corresponding Poker Hand for Each *Class*.

Class	Corresponding Poker Hand
9	Royal Flush
8	Straight Flush
7	Four of a Kind
6	Full House
5	Flush
4	Straight
3	Three of a Kind
2	Two Pairs
1	One Pair
0	Nothing in Hand

Provided below, in Figure 1, are some examples of each kind of poker hand that a player can have after being dealt 5 cards in the poker game.

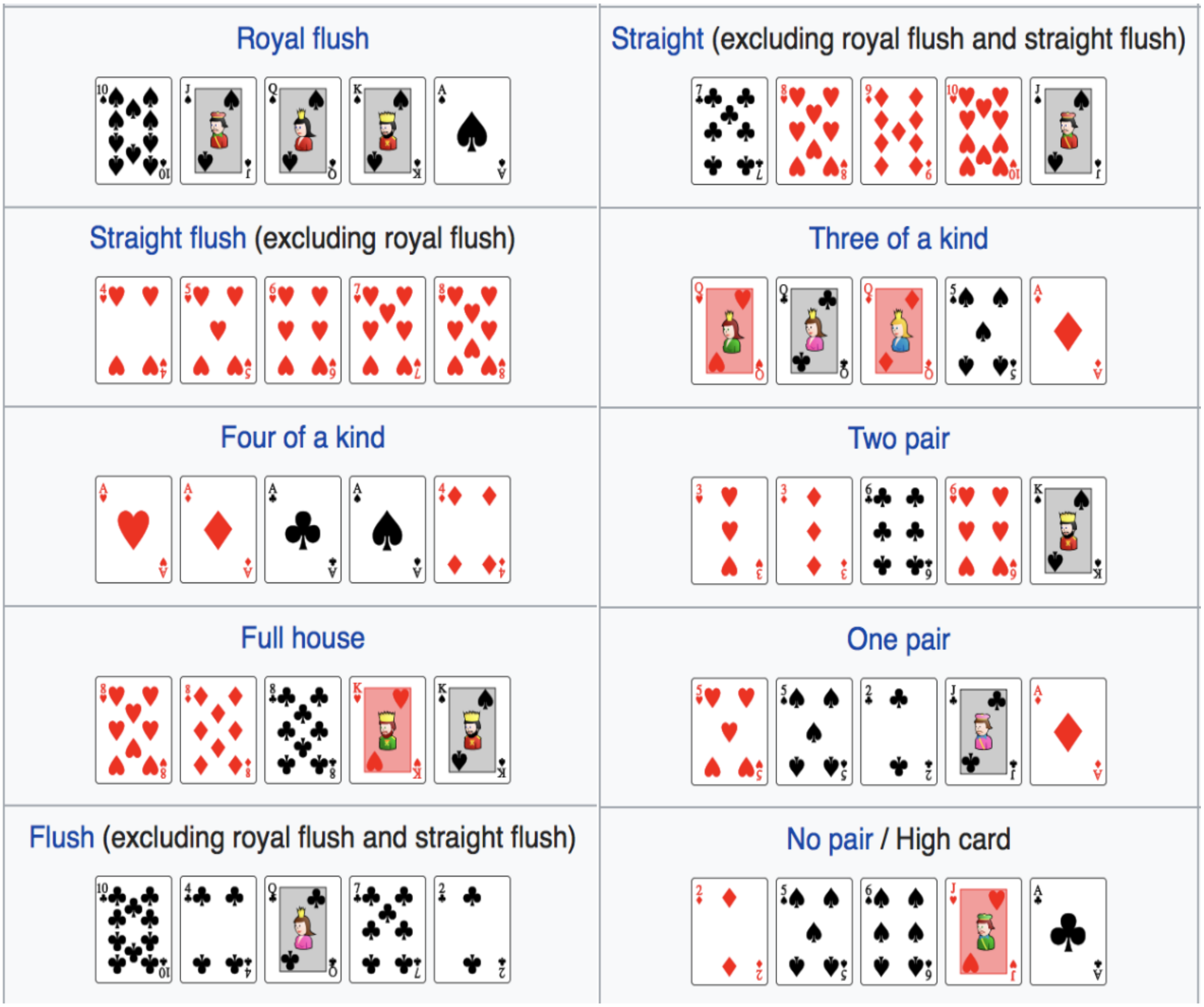

Figure 1: Examples of Each Poker Hand (Wikipedia 2019)

First, we must understand how the game of poker works. The data set considers a single player with a single poker hand. A poker hand is dealt 1 card at a time, without replacement. The player receives a total of 5 cards in each hand. Once all 5 cards are dealt, the player can have any one of the 10 classified hands described above in Table 1 and Figure 1. Given a standard deck of 52 cards, the number of ways that a player can obtain each hand, along with the probability and the odds of these outcomes, is already known (Wikipedia 2019, UHM Department of Mathematics 2005).

Table 2: Probability and Odds of each Poker Hand

Class	Corresponding Poker Hand	Distinct Hands	Frequency	Probability	Odds
9	Royal Flush	1	4	0.000153908%	$649,737.8 : 1$
8	Straight Flush	9	36	0.00138517%	$72,192.3 : 1$
7	Four of a Kind	156	624	0.0240096%	$4,164.001 : 1$
6	Full House	156	$3,744$	0.144058%	$693.165 : 1$
5	Flush	$1,277$	$5,108$	0.19654%	$507.802 : 1$
4	Straight	10	$10,200$	0.392465%	$253.800 : 1$
3	Three of a Kind	858	$54,912$	2.11285%	$46.329 : 1$
2	Two Pairs	858	$123,552$	4.7539%	$20.035 : 1$
1	One Pair	$2,860$	$1,098,240$	42.2569%	$1.366 : 1$
0	Nothing in Hand	$1,277$	$1,302,540$	50.1177%	$0.995 : 1$
-	Total	$7,462$	$2,598,960$	100%	-

We can use the known values provided in Table 2 to compare against the probabilities and odds that we obtain from the test and training poker hands provided in the Poker Hand Data set.

2 Methodology

In order to analyze our data, we can use the *table()* function to get the distribution of *Class* within both the training and test sets. From this table, we can obtain the probability of each type of poker hand in both sets, along with the odds of each poker hand.

We can use the package *ggplot2* in order visualize the probabilities and odds for each *Class* in both the training and test set. We can also use this package to visualize a comparison between these obtained probabilities and odds against the known values presented above in Table 2.

We can fit the proportional odds model on the training set using the *vglm()* function from the "VGAM" package (Dang 2019, slides 35 - 36, 40 - 42). We will use this model since our data is an example of ordered multinomial data. Thus, we want to use an Ordinal Response Multicategory Logit Model to run our analysis (Dang 2019, slides 44 - 47). Using the results from this model, we can analyze the odds of getting each of the 10 different types of poker hands. We can also use the model and results to make predictions about what hand we will have, given the 5 cards we are dealt in the poker game. In order to make these predictions, we can use the *predictvglm()* function and our proposed model on the test set (Dang 2019, slides 44 - 48).

3 Results and Discussion

In order to carry out our testing, we first wanted to get an idea about how often each type of poker hand occurred in both the test set and the training set provided in the Poker Hand data set.

Table 3: Frequency and Probability of Getting Each Poker Hand in the Test Set

Class	Corresponding Poker Hand	Frequency	Probability
9	Royal Flush	3	0.0003%
8	Straight Flush	12	0.0012%
7	Four of a Kind	230	0.023%
6	Full House	1,424	0.1424%
5	Flush	1,996	0.1996%
4	Straight	3,885	0.3885%
3	Three of a Kind	21,121	2.1121%
2	Two Pairs	47,622	4.7622%
1	One Pair	422,498	42.2498%
0	Nothing in Hand	501,209	50.1209%
-	Total	1,000,000	100%

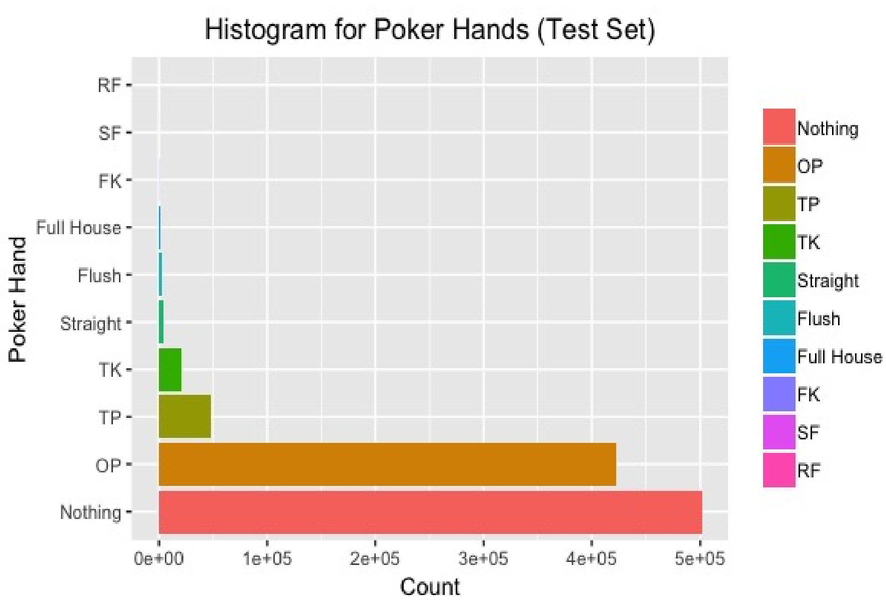

Figure 2: Histogram of *CLASS* for the Test Set

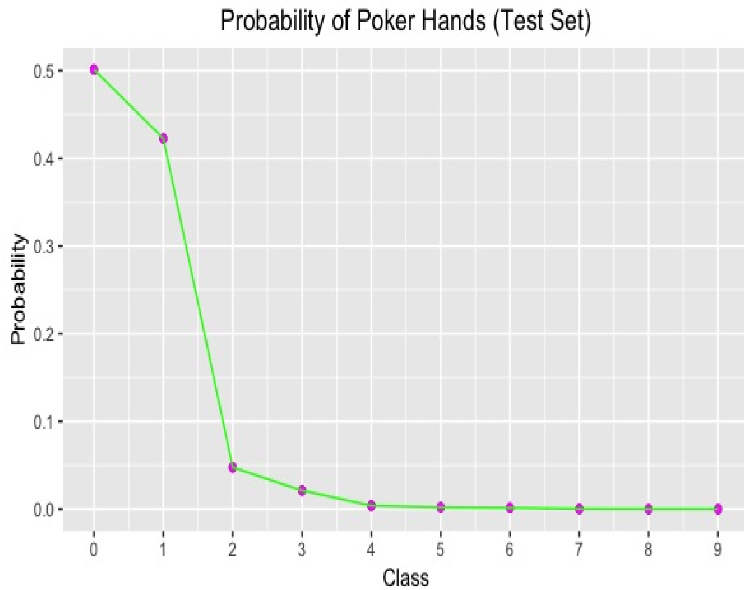

Figure 3: Probability of *CLASS* in the Test Set

In Table 3, we can see the corresponding frequency and probability of each poker hand appearing in the test set. The frequency is depicted in the form of a histogram in Figure 2 and the probability is depicted in Figure 3.

Table 4: Frequency and Probability of Getting Each Poker Hand in the Training Set

Class	Corresponding Poker Hand	Frequency	Probability
9	Royal Flush	5	0.019992%
8	Straight Flush	5	0.019992%
7	Four of a Kind	6	0.0239904%
6	Full House	36	0.1439424%
5	Flush	54	0.2159136%
4	Straight	93	0.3718513%
3	Three of a Kind	513	2.05118%
2	Two Pairs	1,206	4.822071%
1	One Pair	10,599	42.37905%
0	Nothing in Hand	12,493	49.95202%
-	Total	25,010	100%

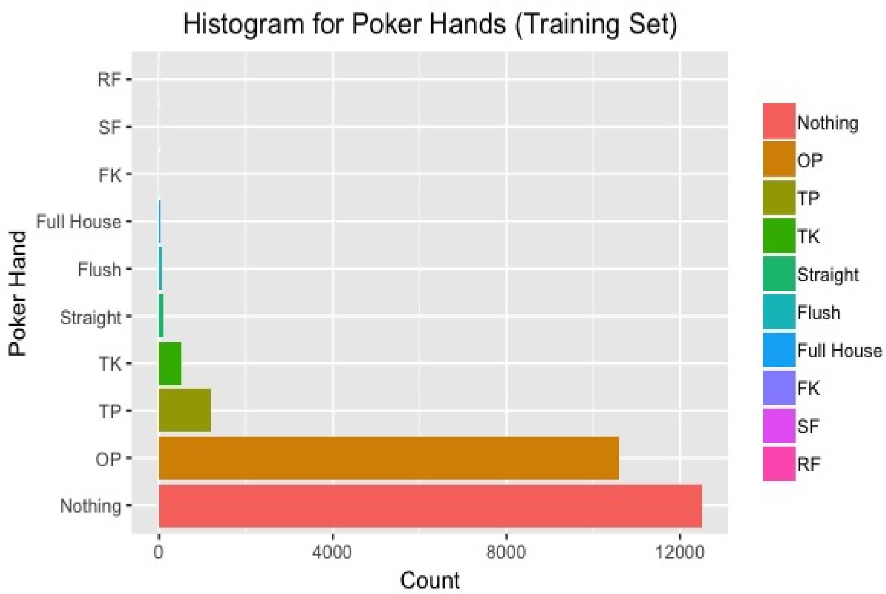

Figure 4: Histogram of *CLASS* for the Training Set

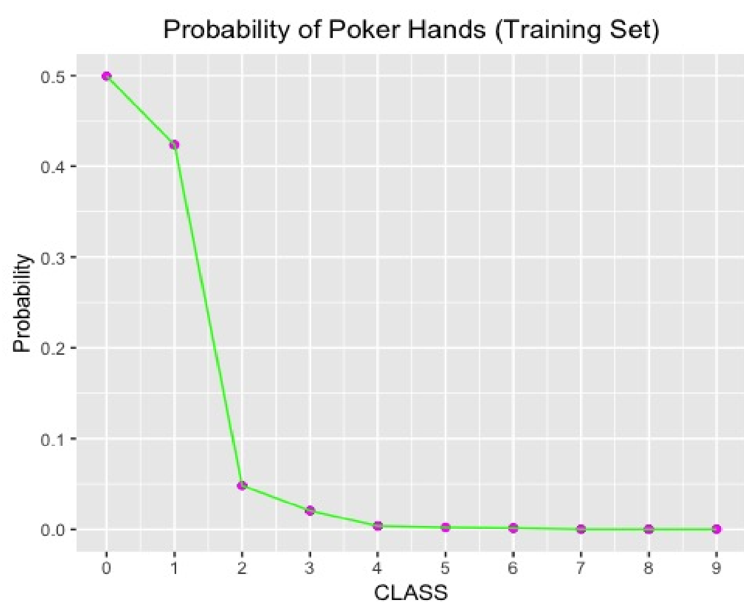

Figure 5: Probability of *CLASS* in the Training Set

In Table 4, we can see the corresponding frequency and probability of each poker hand appearing in the training set. The frequency is depicted in the form of a histogram in Figure 4 and the probability is depicted in Figure 5.

We proceed by fitting the proportional odds model using all 10 variables, the suit and rank for each of the 5 cards, as predictors for the response *Class* in the training set.

```
> m1 <- vglm(CLASS~S1 + C1 + S2 + C2 + S3 + C3 + S4 + C4 + S5 + C5, family = propodds(reverse = FALSE), data =
↪  poker_hands_train)
>
> summary(m1)

Call:
vglm(formula = CLASS ~ S1 + C1 + S2 + C2 + S3 + C3 + S4 + C4 +
    S5 + C5, family = propodds(reverse = FALSE), data = poker_hands_train)

Pearson residuals:
                        Min        1Q    Median       3Q     Max
logitlink(P[Y<=1])   -1.166 -1.093632 -0.032608 0.979835  1.0448
logitlink(P[Y<=2])   -4.507  0.169423  0.172929 0.368261  0.3842
logitlink(P[Y<=3])   -6.980  0.090532  0.093170 0.111491  1.2234
logitlink(P[Y<=4]) -15.998  0.044937  0.046308 0.047936  1.3636
logitlink(P[Y<=5]) -20.182  0.030109  0.030622 0.031187  5.4339
logitlink(P[Y<=6]) -25.550  0.020807  0.021112 0.021457  7.0133
logitlink(P[Y<=7]) -60.494  0.011602  0.011770 0.011955  5.3419
logitlink(P[Y<=8]) -62.751  0.008860  0.008987 0.009127 24.4133
logitlink(P[Y<=9]) -68.270  0.006842  0.006939 0.007042 24.3813

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept):1 -0.0495686  0.0836927  -0.592   0.5537
(Intercept):2  2.4412451  0.0860272  28.378   <2e-16 ***
(Intercept):3  3.4831885  0.0909992  38.277   <2e-16 ***
(Intercept):4  4.7788260  0.1090881  43.807   <2e-16 ***
(Intercept):5  5.4124210  0.1277055  42.382   <2e-16 ***
(Intercept):6  6.1267866  0.1615713  37.920   <2e-16 ***
(Intercept):7  7.3068854  0.2633894  27.742   <2e-16 ***
(Intercept):8  7.7771275  0.3269150  23.789   <2e-16 ***
(Intercept):9  8.4704740  0.4548318  18.623   <2e-16 ***
S1            -0.0070404  0.0109871  -0.641   0.5217
C1            -0.0017249  0.0032703  -0.527   0.5979
S2             0.0053476  0.0109355   0.489   0.6248
C2             0.0036423  0.0032552   1.119   0.2632
S3             0.0008664  0.0109254   0.079   0.9368
C3             0.0045837  0.0032758   1.399   0.1617
S4            -0.0051708  0.0109883  -0.471   0.6379
C4            -0.0049055  0.0032720  -1.499   0.1338
S5             0.0198628  0.0109653   1.811   0.0701 .
C5             0.0002178  0.0032766   0.066   0.9470
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Number of linear predictors:  9

Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2]), logitlink(P[Y<=3]), logitlink(P[Y<=4]),
logitlink(P[Y<=5]), logitlink(P[Y<=6]), logitlink(P[Y<=7]), logitlink(P[Y<=8]), logitlink(P[Y<=9])

Residual deviance: 49277.64 on 225071 degrees of freedom
```

```
Log-likelihood: -24638.82 on 225071 degrees of freedom

Number of Fisher scoring iterations: 3

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):4', '(Intercept):5', '(Intercept):6', '(Intercept):7', '(Intercept):8', '(Intercept):9'


Exponentiated coefficients:
       S1        C1        S2        C2        S3        C3        S4        C4        S5        C5
0.9929843 0.9982766 1.0053619 1.0036489 1.0008668 1.0045942 0.9948426 0.9951065 1.0200614 1.0002179
>
> AICvlm(m1)
[1] 49315.64
```

The summary of the model with all variables as predictors indicates that some of these variables are not significant, as they have a p-value higher than 0.05. However, we will not proceed with any variable selection methods. We can not reduce the number of predictors, as the resulting *Class* for a poker hand is directly related to both the rank and suit of each of the 5 cards in hand. Therefore, we can use the full model when investigating and making predictions on the test set.

We can compare the obtained probabilities for both the training and the test set to the known values provided in Table 2.

Table 5: Comparison of Probability for Test and Training Sets against Known Values

Class	Poker Hand	Known Probability	Prob. for Test Set	Prob. for Training Set
9	Royal Flush	0.000153908%	0.0003%	0.019992%
8	Straight Flush	0.00138517%	0.0012%	0.019992%
7	Four of a Kind	0.0240096%	0.023%	0.0239904%
6	Full House	0.144058%	0.1424%	0.1439424%
5	Flush	0.19654%	0.1996%	0.2159136%
4	Straight	0.392465%	0.3885%	0.3718513%
3	Three of a Kind	2.11285%	2.1121%	2.05118%
2	Two Pairs	4.7539%	4.7622%	4.822071%
1	One Pair	42.2569%	42.2498%	42.37905%
0	Nothing in Hand	50.1177%	50.1209%	49.95202%
-	Total	100%	100%	100%

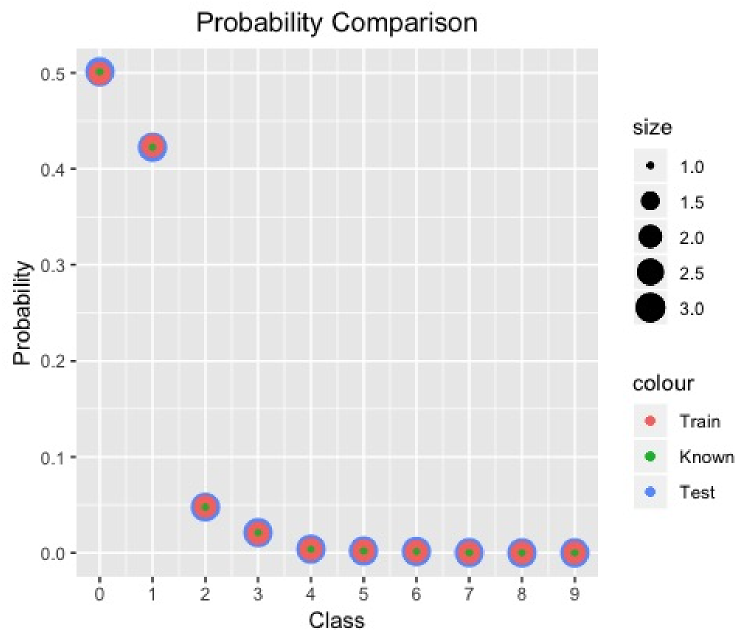

Figure 6: Probability Comparisons for Test and Training Set to Known Values

As can be seen above in both Table 5 and Figure 6, the probabilities for both the test and training set are close to the known probabilities. All of the values are within 0.17% of the known value, with the highest discrepancy being between the training set and the known values for $Class = 9$.

We can proceed by calculating the odds of getting a poker hand of each *Class* in both the test and the training set.

Table 6: Comparison of Odds for Test and Training Sets against Known Values

Class	Poker Hand	Known Odds	Odds for Test Set	Odds for Training Set
9	Royal Flush	$649,737.8 : 1$	$333,332.3 : 1$	$5001 : 1$
8	Straight Flush	$72,192.3 : 1$	$83,323.3 : 1$	$5001 : 1$
7	Four of a Kind	$4,164.001 : 1$	$4,346.826 : 1$	$4,167.333 : 1$
6	Full House	$693.165 : 1$	$701.247 : 1$	$693.722 : 1$
5	Flush	$507.802 : 1 : 1$	$500.002 : 1$	$462.148 : 1$
4	Straight	$253.800 : 1$	$256.400 : 1$	$267.925 : 1$
3	Three of a Kind	$46.329 : 1$	$46.346 : 1$	$47.752 : 1$
2	Two Pairs	$20.035 : 1$	$19.999 : 1$	$19.738 : 1$
1	One Pair	$1.366 : 1$	$1.367 : 1$	$1.360 : 1$
0	Nothing in Hand	$0.995 : 1$	$0.995 : 1$	$1.002 : 1$

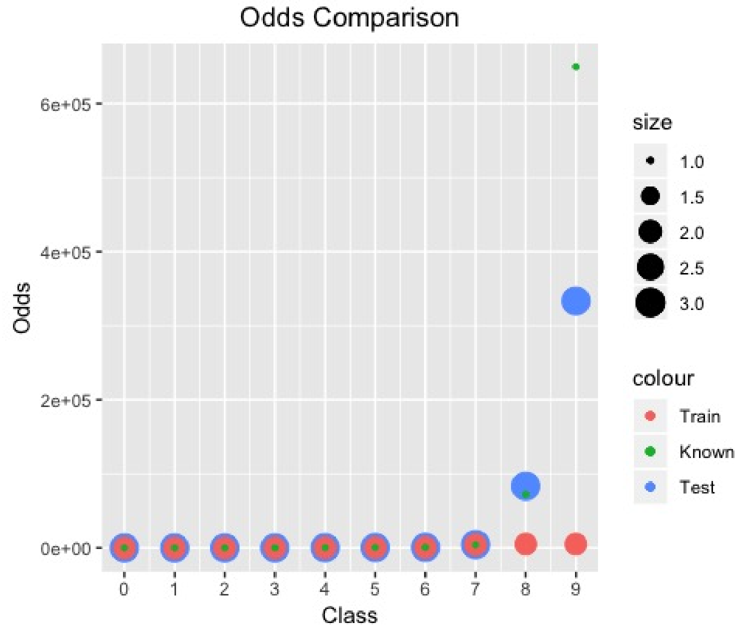

Figure 7: Odds Comparisons for Test and Training Set to Known Values

As can be seen above in both Table 6 and Figure 7, the odds for both the test and training set are close to the known odds for *Class* values from 0 to 7. For *Class* = 8, the odds for the test set are higher than the known value, while the odds for the training set are lower. The odds for the training set and test set are both lower than the known value for the *Class* of 9. These differences are due to the idea that even seeing a Royal Flush in one more instance out of the test and training sets can lead to a change in the odds by a large factor.

We can proceed by making some predictions about the chances of getting a certain hand given our first 2 cards in poker. For instance, if we are dealt the 4 of hearts and the 4 of diamonds, we can guarantee that we will not get a poker hand corresponding to having Nothing in Hand ($Class = 0$), a Straight ($Class = 4$), a Flush($Class = 5$), a Straight Flush ($Class = 8$), or a Royal Flush ($Class = 9$). What more can we say if the third card dealt to us is the 4 of spades? Now, we can not have a poker hand with just One Pair ($Class = 1$). Furthermore, what if the fourth card dealt to us is the 4 of clubs? Now, we can not have a poker hand with just Three of a Kind ($Class = 3$). We also can not get Two pairs ($Class = 2$) or a Full House ($Class = 6$).The only hand that is possible, given that our first 4 cards are the 4 of diamonds, hearts, spades and clubs, is Four of a Kind ($Class = 7$), regardless of what the fifth and final card turns out to be.

However, when trying to run predictions using our model, we do not obtain these results. Suppose we try to predict the outcome given that we get the 4 of hearts as our first card ($S1 = 1, C1 = 4$), the 4 of diamonds as our second card ($S2 = 3, C2 = 4$), the 4 of spades as our third card ($S3 = 2, C3 = 4$), the 4 of clubs as our fourth card ($S4 = 4, C1 = 4$) and some arbitrary card, say the 3 of hearts, as our fifth card ($S5 = 1, C5 = 3$). Running a prediction using the full model with all 10 predictors gives us the following probabilities for each *Class*.

Table 7: Predicted Probability for Example Poker Hand

Class	Poker Hand	Probability
9	Royal Flush	0.0206%
8	Straight Flush	0.0206%
7	Four of a Kind	0.02472%
6	Full House	0.14831%
5	Flush	0.22244%
4	Straight	0.38301%
3	Three of a Kind	2.11115%
2	Two Pairs	4.9533%
1	One Pair	42.93125%
0	Nothing in Hand	49.18462%

As we can see in Table 7, the model gives a probability for each *Class* of poker hands. Meanwhile, as we discussed in the analysis above, the only possible outcome with this example hand is to get Four of a Kind, corresponding to a *Class* of 7. Therefore, we should see a value for the probability of all other poker hands of 0% and a value of 100% for the *Class* of 7, corresponding to having Four of a Kind in hand.

This problem arises with making predictions for other hands of poker in the test set as well. By using the *predict()* function and the proportional odds model, with the *Polr()* function from the *MASS* package, we obtain a distribution for the classification of hands in the test set summarized below in Table 8.

Table 8: Expected and Predicted Values for the Frequency of Each Poker Hand in the Test Set

Class	Corresponding Poker Hand	Expected Frequency	Predicted Frequency
9	Royal Flush	3	0
8	Straight Flush	12	0
7	Four of a Kind	230	0
6	Full House	1,424	0
5	Flush	1,996	0
4	Straight	3,885	0
3	Three of a Kind	21,121	0
2	Two Pairs	47,622	0
1	One Pair	422,498	44,847
0	Nothing in Hand	501,209	955,153
-	Total	1,000,000	1,000,000

Table 9: Confusion Matrix for Prediction on the Test Set

Predicted *Class*	Nothing in Hand	One Pair	Two Pairs	Three of a Kind	Straight	Flush	Full House	Four of a Kind	Straight Flush	Royal Flush
Nothing in Hand	478,607	403,558	45,555	20,229	3,702	1,907	1,361	219	12	3
One Pair	22,602	18,940	2,067	892	183	89	63	11	0	0
Two Pairs	0	0	0	0	0	0	0	0	0	0
Three of a Kind	0	0	0	0	0	0	0	0	0	0
Straight	0	0	0	0	0	0	0	0	0	0
Flush	0	0	0	0	0	0	0	0	0	0
Full House	0	0	0	0	0	0	0	0	0	0
Four of a Kind	0	0	0	0	0	0	0	0	0	0
Straight Flush	0	0	0	0	0	0	0	0	0	0
Royal Flush	0	0	0	0	0	0	0	0	0	0

We can also see in Table 8 that our predicted frequencies do not match the expected value provided in the Poker Hand Test Set. The obtained predictions are not accurate. When trying to make predictions, the poker hands were allocated to a value for *Class* of 0 or 1, even if the 5 cards were known to give a different poker hand that did not correspond to having Nothing in Hand or One Pair.

4 Limitations and Challenges

The training data set of 25,000 instances was largely biased towards poker hands corresponding to having Nothing in Hand ($Class = 0$) and One Pair ($Class = 1$), having more than 20,000 instances of those two alone, as compared to only 5 being Royal Flush (*Class* $= 9$). With the overall probability of predicting One Pair or Nothing being around 92.33%, any model like the proportional odds model or multinomial model gives predictions that allocate the poker hands to these two *Classes*. The test dataset is fairly large and the number of observations for *Classes* other than having Nothing in Hand or One Pair are very small in comparison. Inculcating Support Vector Machine or Neural Networks with many hidden layers is one potential solution to this problem. One of the main things to note is that the model should have the ability to differentiate between Three of a Kind and Full House, as well as One Pair and Full House (Bhat and Selvam 2016). To overcome some of these limitations and successfully classify the poker hands, we can add additional observations for *Classes* from 2 to 9 to reduce or remove this bias and increase the size of the training data set (Bhat and Selvam 2016).

5 Conclusion

Based off of the analysis above, we are able to calculate the probability and odds of having one of the 10 *Classes* of poker hands described in Table 1 and Figure 1 for both the training and test sets provided in the Poker Hand Data set (Dua, Dheeru and Graff, Casey 2017). These computed probabilities and odds are very similar to the known values, provided in Table 2, for a standard deck of 52 cards. Since the training and test set are ordered multinomial data, we fit a proportional odds model using the *vglm* function. The model used for both sets contained all 10 predictors, the suit and rank corresponding to each of the 5 cards in a particular poker hand.

This model did not prove to be accurate in making predictions for the poker hands in the test set. When trying to make predictions, the poker hands were allocated to a value for *Class* of 0 or 1, even if the 5 cards were known to give a different poker hand that did not correspond to having Nothing in Hand or One Pair. Since we had a disproportional data set, with a large proportion being in *Class* $= 0$, we tried to remove these cases from the data set. This led to the predictions allocating the poker hands to a value for *Class* of 1 or 2, corresponding to having One Pair or Two Pairs, even if the hand was known to be of a different *Class*. These limitations and issues in using our model did not lead to reliable or accurate predictions for the poker hands in the test set. Future work includes finding a package or function in R that is able to manage the size of the test set and take into account the disproportionate data in the test set. With such a function in R, the goal would be to find a way to make more reliable predictions and accurate classifications of the poker hands.

6 Bibliography

1. Bhat, Gautam and Selvam, Kaviarsan 2016, *NN-based Poker Hand Classification and Game Playing*, Boston University, https://pdfs.semanticscholar.org/905a/0f27e520b3f2627863f9fa94a87fda7060cd.pdf, Accessed on 30, Apr 2019.

2. Dang, S 2019, *Multinomial Regression*, lecture notes, Regression II MA532, Binghamton University, delivered 10, Apr 2019.

3. Dua, Dheeru and Graff, Casey 2017, *UCI Machine Learning Repository, Poker Hand Data Set*, University of California, Irvine, School of Information and Computer Sciences, https://archive.ics.uci.edu/ml/datasets/Poker+Hand, Accessed on 28, Apr 2019.

4. UHM Department of Mathematics 2005, *5-CARD POKER HANDS*, http://www.math.hawaii.edu/~ramsey/Probability/PokerHands.html, Accessed on 28, Apr 2019.

5. Wikipedia 2019, *Poker probability*, https://en.wikipedia.org/wiki/Poker_probability, Accessed on 28, Apr 2019.

7 Appendix

1. #Loading Data and Libraries
```
poker_hands_test <- read.csv(url("https://archive.ics.uci.edu/ml/machine-lea⌋
↪  rning-databases/poker/poker-hand-testing.data"),header=FALSE)
poker_hands_train <- read.csv(url("https://archive.ics.uci.edu/ml/machine-le⌋
↪  arning-databases/poker/poker-hand-training-true.data"),header=FALSE)
colnames(poker_hands_test) <-
↪  c("S1","C1","S2","C2","S3","C3","S4","C4","S5","C5","CLASS")
colnames(poker_hands_train) <-
↪  c("S1","C1","S2","C2","S3","C3","S4","C4","S5","C5","CLASS")
library(ggplot2)
library(VGAM)
library("MASS")
library(dplyr)
```

2. #Table of count for each class in test set
```
table(poker_hands_test$CLASS)
```

3. #Histogram for Test Set
```
poker_hands_test$CLASS = factor(poker_hands_test$CLASS,levels =
↪  0:9,labels=c("Nothing","OP","TP","TK","Straight","Flush","Full
↪  House","FK","SF","RF"))
ggplot(data = poker_hands_test, aes(poker_hands_test$CLASS), fill =
↪  factor(poker_hands_test$CLASS)) + geom_bar(aes(fill =
↪  factor(poker_hands_test$CLASS))) + xlab("Poker Hand") + ylab("Count") +
↪  ggtitle("Histogram for Poker Hands (Test Set)") + theme(legend.title =
↪  element_blank(), plot.title = element_text(hjust = 0.5)) +
↪  scale_x_discrete(labels =
↪  c("Nothing","OP","TP","TK","Straight","Flush","Full
↪  House","FK","SF","RF")) + coord_flip()
```

4. #Graph for Probability in Test Set
```
l9 = length(which(poker_hands_test$CLASS == 9))
l9
l8 = length(which(poker_hands_test$CLASS == 8))
l8
l7 = length(which(poker_hands_test$CLASS == 7))
```

```
l7
l6 = length(which(poker_hands_test$CLASS == 6))
l6
l5 = length(which(poker_hands_test$CLASS == 5))
l5
l4 = length(which(poker_hands_test$CLASS == 4))
l4
l3 = length(which(poker_hands_test$CLASS == 3))
l3
l2 = length(which(poker_hands_test$CLASS == 2))
l2
l1 = length(which(poker_hands_test$CLASS == 1))
l1
l0 = length(which(poker_hands_test$CLASS == 0))
l0
s = sum(l0+l1+l2+l3+l4+l5+l6+l7+l8+l9)
pl9 = l9/s
pl9
pl8 = l8/s
pl8
pl7 = l7/s
pl7
pl6 = l6/s
pl6
pl5 = l5/s
pl5
pl4 = l4/s
pl4
pl3 = l3/s
pl3
pl2 = l2/s
pl2
pl1 = l1/s
pl1
pl0 = l0/s
pl0
```

```
p = sum(pl9+pl8+pl7+pl6+pl5+pl4+pl3+pl2+pl1+pl0)
p
x = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
y = c(pl0, pl1, pl2, pl3, pl4, pl5, pl6, pl7, pl8, pl9)
qplot(x, y, xlab = "Class", ylab = "Probability") + ggtitle("Probability of
↪  Poker Hands (Test Set)") + theme(plot.title = element_text(hjust = 0.5))
↪  + geom_point(colour = I("magenta")) + geom_line(colour = I("green")) +
↪  scale_x_continuous(breaks = seq(0, 9, by = 1))
```

5.
```
#Table of count for each class in training set
table(poker_hands_train$CLASS)
```

6.
```
#Histogram for Training Set
poker_hands_train$CLASS = factor(poker_hands_train$CLASS,levels =
↪  0:9,labels=c("Nothing","OP","TP","TK","Straight","Flush","Full
↪  House","FK","SF","RF"))
ggplot(data = poker_hands_train, aes(poker_hands_train$CLASS), fill =
↪  factor(poker_hands_train$CLASS)) + geom_bar(aes(fill =
↪  factor(poker_hands_train$CLASS))) + xlab("Poker Hand") + ylab("Count") +
↪  ggtitle("Histogram for Poker Hands (Training Set)") + theme(legend.title
↪  = element_blank(), plot.title = element_text(hjust = 0.5)) +
↪  scale_x_discrete(labels =
↪  c("Nothing","OP","TP","TK","Straight","Flush","Full
↪  House","FK","SF","RF")) + coord_flip()
```

7.
```
#Graph for Probability in Training Set
ll9 = length(which(poker_hands_train$CLASS == 9))
ll9
ll8 = length(which(poker_hands_train$CLASS == 8))
ll8
ll7 = length(which(poker_hands_train$CLASS == 7))
ll7
ll6 = length(which(poker_hands_train$CLASS == 6))
ll6
ll5 = length(which(poker_hands_train$CLASS == 5))
ll5
ll4 = length(which(poker_hands_train$CLASS == 4))
```

```
ll4
ll3 = length(which(poker_hands_train$CLASS == 3))
ll3
ll2 = length(which(poker_hands_train$CLASS == 2))
ll2
ll1 = length(which(poker_hands_train$CLASS == 1))
ll1
ll0 = length(which(poker_hands_train$CLASS == 0))
ll0
ss = sum(ll0+ll1+ll2+ll3+ll4+ll5+ll6+ll7+ll8+ll9)
pll9 = ll9/ss
pll9
pll8 = ll8/ss
pll8
pll7 = ll7/ss
pll7
pll6 = ll6/ss
pll6
pll5 = ll5/ss
pll5
pll4 = ll4/ss
pll4
pll3 = ll3/ss
pll3
pll2 = ll2/ss
pll2
pll1 = ll1/ss
pll1
pll0 = ll0/ss
pll0
pp = sum(pll9+pll8+pll7+pll6+pll5+pll4+pll3+pll2+pll1+pll0)
pp
y2 = c(pll0, pll1, pll2, pll3, pll4, pll5, pll6, pll7, pll8, pll9)
```

```
qplot(x, y2, xlab = "CLASS", ylab = "Probability") + ggtitle("Probability of
↪  Poker Hands (Training Set)") + theme(plot.title = element_text(hjust =
↪  0.5)) + geom_point(colour = I("magenta")) + geom_line(colour =
↪  I("green")) + scale_x_continuous(breaks = seq(0, 9, by = 1))
```

8. #vglm for training set

```
m1 <- vglm(CLASS~S1 + C1 + S2 + C2 + S3 + C3 + S4 + C4 + S5 + C5, family =
↪  propodds(reverse = FALSE), data = poker_hands_train)
summary(m1)
AICvlm(m1)
```

9. #graph for probability comparison

```
ggplot() + ggtitle("Probability Comparison") + theme(plot.title =
↪  element_text(hjust = 0.5)) + geom_point(aes(x, y, color="red", size =
↪  3)) + geom_point(aes(x, y2, color="blue", size = 2)) + geom_point(aes(x,
↪  y3, color = "green", size = 1)) + scale_x_continuous(breaks = seq(0, 9,
↪  by = 1)) + scale_colour_discrete(breaks=c("blue", "green",
↪  "red"),labels=c("Train", "Known", "Test")) + xlab("Class") +
↪  ylab("Probability")
```

10. #odds for test set

```
(1-y)/y
```

11. #odds for training set

```
(1-y2)/y2
```

12. #odds for known values

```
y3 = c(0.501177, 0.422569, 0.047539, 0.0211285, 0.00392465, 0.0019654,
↪  0.00144058, 0.000240096, 0.0000138517, 0.00000153908)
(1-y3)/y3
```

13. #graph for odds comparison

```
o1 = (1-y)/y
o1
o2 = (1-y2)/y2
o2
o3 = (1-y3)/y3
```

```
o3
ggplot() + ggtitle("Odds Comparison") + theme(plot.title =
↪  element_text(hjust = 0.5)) + geom_point(aes(x, o1, color="red", size =
↪  3)) + geom_point(aes(x, o2, color="blue", size = 2)) + geom_point(aes(x,
↪  o3, color = "green", size = 1)) + scale_x_continuous(breaks = seq(0, 9,
↪  by = 1)) + scale_colour_discrete(breaks=c("blue", "green",
↪  "red"),labels=c("Train", "Known", "Test")) + xlab("Class") + ylab("Odds")
```

14.
```
#prediction for example poker hand
p = predictvglm(m1,data.frame(S1 = 1, C1 = 4, S2 = 3, C2 = 4, S3 = 2, C3 =
↪  4, S4 = 4, C4 = 4, S5 = 1, C5 = 3))
p
exp(p)/(1+exp(p))
```

15.
```
#frquency table for predictions on test set
train = poker_hands_train
test = poker_hands_test
test$S1 = factor(test$S1,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
test$C1 = factor(test$C1,levels = 1:13,labels=1:13,ordered = TRUE)
test$S2 = factor(test$S2,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
test$C2 = factor(test$C2,levels = 1:13,labels=1:13,ordered = TRUE)
test$S3 = factor(test$S3,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
test$C3 = factor(test$C3,levels = 1:13,labels=1:13,ordered = TRUE)
test$S4 = factor(test$S4,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
test$C4 = factor(test$C4,levels = 1:13,labels=1:13,ordered = TRUE)
test$S5 = factor(test$S5,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
test$C5 = factor(test$C5,levels = 1:13,labels=1:13,ordered = TRUE)
test$CLASS = factor(test$CLASS,levels =
↪  0:9,labels=c("Nothing","OP","TP","TK","Straight","Flush","Full
↪  House","FK","SF","RF"),ordered = TRUE)
test_X = test[,-11]
test_Y = test[,11]
```

```
train$S1 = factor(train$S1,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
train$C1 = factor(train$C1,levels = 1:13,labels=1:13,ordered = TRUE)
train$S2 = factor(train$S2,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
train$C2 = factor(train$C2,levels = 1:13,labels=1:13,ordered = TRUE)
train$S3 = factor(train$S3,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
train$C3 = factor(train$C3,levels = 1:13,labels=1:13,ordered = TRUE)
train$S4 = factor(train$S4,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
train$C4 = factor(train$C4,levels = 1:13,labels=1:13,ordered = TRUE)
train$S5 = factor(train$S5,levels =
↪  c(1,2,3,4),labels=c("H","S","D","C"),ordered = TRUE)
train$C5 = factor(train$C5,levels = 1:13,labels=1:13,ordered = TRUE)
train$CLASS = factor(train$CLASS,levels =
↪  0:9,labels=c("Nothing","OP","TP","TK","Straight","Flush","Full
↪  House","FK","SF","RF"),ordered = TRUE)
options(contrasts=c("contr.treatment", "contr.poly"))
polrMod <- polr(CLASS ~S1+C1+S2+C2+S3+C3+S4+C4+S5+C5, data=train)
predictedClass <- predict(polrMod, test_X,type ="class")  # predict the
↪  classes directly
head(predictedClass)
freq_table <- data.frame(table(predictedClass))
colnames(freq_table) <- c("Hands","Count")
freq_table <- freq_table[order(freq_table$Count),]
freq_table
```

16.
```
#Confusion Matrix for testset and Predicted set
table(predictedClass,test_Y)
```