All in with the Poker Hands dataset

Brett Tully, btully@bellarmine.edu
Jack Huff, jhuff@bellarmine.edu

ABSTRACT
The poker hands data set explores machine learning in a unique way. By having a largely disproportionate distribution of classifying results, many of the machine learning models were not fit for this dataset. Using the Decision Tree Classifier and the Random Forrest Classifier models with varying parameters, we were able to produce results with accuracy scores between the range of 50% and 63%. The dataset came with both a training dataset and a testing dataset, and these datasets were used to train, test, and validate the models.

I. INTRODUCTION

The poker hands dataset contains both a testing set with 1,000,000 rows and a training set with 25,010 rows. Each row contains 10 numerical values and 1 ordinal value. Each row represents a poker hand consisting of 5 cards randomly drawn from a standard deck of 52 cards. Every card is represented by 2 columns: one denoting the rank of the card (Ace, 2, 3... Jack, Queen, King) and the other denoting the suit of the card (Hearts, Spades, Diamonds, Clubs). The order of the cards (columns 1-10) is important in determining a poker hand (column 11), which is the classification target.

II. BACKGROUND

While poker is a game of skill, one's success is largely subject to luck. Knowing this, researchers decided to see how many times each specific poker hand occurred when sample poker hands were drawn 1,000,000 times. Furthermore, data scientist wanted to see if this set of data would be appropriate for machine learning. Since poker hands are based off the suit, rank, and order of the 5 cards drawn, the authors thought that these factors would work well in trying to predict the type of poker hand. The 9 different types of poker hands are as follows:

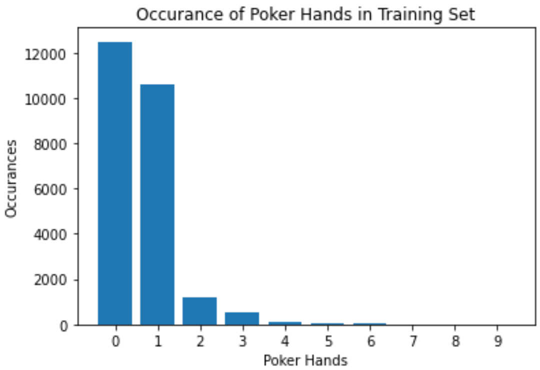
0: Nothing in hand; not a recognized poker hand
1: One pair; one pair of equal ranks within five cards
2: Two pairs; two pairs of equal ranks within five cards
3: Three of a kind; three equal ranks within five cards
4: Straight; five cards, sequentially ranked with no gaps
5: Flush; five cards with the same suit
6: Full house; pair + different rank three of a kind
7: Four of a kind; four equal ranks within five cards
8: Straight flush; straight + flush
9: Royal flush; {Ace, King, Queen, Jack, Ten} + flush

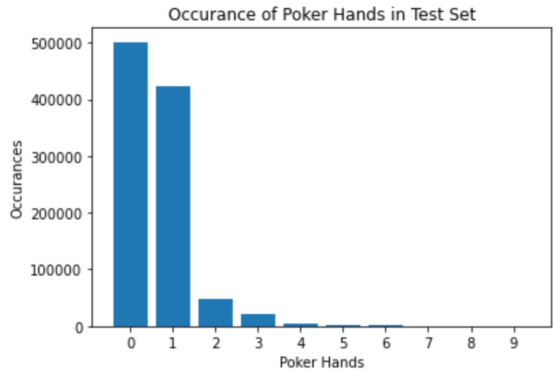
III. EXPLORATORY ANALYSIS

The Poker Hands data set contains both a testing set containing 1,000,000 samples with 11 columns and a training set with 25,010 samples and 11 columns. Both data sets have the same variable names and data types.

Table 1: Data Types

Variable Name	*Data Type*
V1: Suit of Card	Int-64, numerical
V2: Rank of Card	Int-64, numerical
V3: Poker Hand	Int-64, ordinal





A bar chart displaying the instances of each poker hand in both the training dataset and the testing dataset shows that the two data sets have near identical distributions. Additionally, it displays that there is no linear relationship present.

IV. METHODS

A. Data Preparation

The poker hands dataset did not require much data preparation. There was no missing data, and no variables could be dropped as the first 10 represent the 5 cards in the poker hand, and the order, rank, and suit of the cards denote the poker hand.

B. Experimental Design

Our dataset came pre-prepared with a testing set and a training set

Table X: Experiment Parameters

Experiment Number	Parameters
1	Testing dataset: 1,000,000 samples, Training dataset: 25,010 samples

C. Tools Used

In the poker hands dataset analysis, Python 3.8.13 was used in the Anaconda 2.03 environment for Apple Macintosh computer for all analysis. Python was selected because it has several libraries were utilized to aid the analysis. Libraries such as Pandas, Numpy, Matplotlib, Seaborn, and SKLearn were all used to allow machine learning models to be applied and manipulated.

V. RESULTS

A. Classification Measures

Model	Parameters	Results
DecisionTreeClassifier	*criterion='entropy'* *max_depth=11* *splitter = 'best'*	*Accuracy: 55%*
DecisionTreeClassifier	*criterion='gini'* *max_depth=11* *splitter = 'best'*	*Accuracy: 52%*
DecisionTreeClassifier	*criterion='entropy'* *max_depth=11* *min_samples_leaf=1*	*Accuracy: 55%*
RandomForrestClassifier	*criterion='entropy'*	*Accuracy: 61%*
RandomForrestClassifier	*n_estimators = 8* *,criterion = 'entropy* *,bootstrap=True*	*Accuracy: 62%*
RandomForrestClassifier	*n_estimators = 800*	*Accuracy:63%*

	criterion='gini' *bootstrap=True*	
RandomForrestClassifier	*n_estimators = 800* *criterion='gini* *bootstrap=True*	*Accuracy: 62%*

B. Discussion of Results

The Decision Tree Classifier with the parameters of criterion='entropy', max_depth=11 and min_samples_leaf=1 was the best model. While it did not have the highest accuracy score, it did correctly predict poker hands from all 10 classifications. The worst model was the Random Forrest Classifier with parameters n_estimators = 800, criterion='gini, and bootstrap=True. Although it had an accuracy of 62%, it only classified poker hands 0-5, thus not being a good model.

C. Comparison of Models

Decision Tree Classifier is the best model out of the two, because the Random Forest would not classify all 9 Poker hands. Rather, it would leave out on average, Poker Hands 6-9.

D. Problems Encountered

This dataset certainly is not the easiest to work with in terms of machine learning. Although we had no missing values or errors of that form, we had difficulty figuring out why we would not get decent results on any of the models, because you would think that predicting poker hands would be a straightforward problem.

E. Limitations of Implementation

From our research, it seemed as if the best way to get the most accurate predictions is through a neural network of some form, even though we were not able to use neural networks.

F. Improvements/Future Work

For the future, either it would be better to use a different data set, or to be able to use a form of a neural network. We researched the dataset some and found that almost every single person that has worked on this dataset has gotten very similar results without using neural networks but would get very good results when using them.

VI. CONCLUSION

Working with the Poker Hands dataset was a bittersweet experience. It was very nice to have gotten both a training set and a testing set, but we quickly realized this would be no walk in the park. Being limited to only certain ML models certainly hindered our possible results, but we did the best we could with what we had. Our dataset was split into Suits, Ranks, and Poker Hands; the order and type of suit and rank that each of the 5 cards had determines the poker hand. One slight change in either the Suit or Rank could drastically change the poker hand received, so it was interesting working with this type of data. A vast majority of this project was trial and error. The first models we attempted yielded poor results with Logistic Regression and Naives Bayes. We then decided to try out Decision Tree Classifiers and Random Forest Classifiers. At this point, it was all about adjusting different parameters for each model. Often, changing multiple things within the parameters would only yield us an increase in our results by 1%, which technically is still a success in some ways. Overall, this was quite an interesting dataset to work with although we were limited with our models and were unable to achieve 100% accuracy with any of our models. The best part about it, both of us did not have a single clue about anything Poker related prior to this project, but now we both feel confident enough to participate in the High Stakes Las Vegas tournaments.

REFERENCES

Stack Overflow

Division of Labor

Throughout this project, Jack mainly focused on the first half of the notebook, while Brett focused on the second half. Jack did more work exploring the data and getting it ready for machine learning models, but Brett did clean it up a bit and make it look more presentable. Brett created most models and Jack made some edits to the parameters to achieve better results. On the paper, both partners contributed equal amounts, and the same goes for the presentation.