Poker Hand Prediction and Winning Probability

CS419 Course Project

1st Rudraksh Kuchiya *Roll: 20D110021*
2nd Aarya Chaudhari *Roll: 20D110002*
3rd Muskan Bhutra *Roll: 200040085*
4th Raghav Rander *Roll: 200040113*
5rd Utkarsh Jindal *Roll: 200070086*

I. GitHub repository and notebook

- Link to the GitHub Repository: *Click Here*
- Link to the Google Colab Notebook: *Click Here*

II. Problem Description

For given 5 cards from a deck of 52 cards, we will predict the poker hand ranking (classification), also we'll calculate the winning probability of the player. The same problem can be solved deterministically but we will solve this using machine learning to save computing cost.

III. Introduction

To different people, the word "experiment" signifies different things. An experiment is an empirical process used by scientists (and preferably by diligent data scientists) to see if a result agrees or disagrees with a hypothesis. In a machine learning experiment, your hypothesis can be that one algorithm (for example, gradient boosted trees) is superior to others (such as random forests, SVM, linear models). You can collect data and assess results to accept or reject your hypothesis by performing an experiment and running many trials with different variable values. The scientific method is the name given to this process by scientists. Conducting and managing machine learning experiments is difficult regardless of whether you follow the scientific method or not.

Here's a partial list of what you would wish to keep track of:

- Hyperparameters, model designs, and training techniques are examples of parameters.
- Jobs like pre-processing, training, and post-processing use additional infrastructure resources including computation, networking, and storage.
- Training scripts, dependencies, datasets, checkpoints, and trained models are examples of artifacts.
- Metrics include training and evaluation accuracy, as well as loss.
- Weights, biases, gradients, losses, and optimizer state are all examples of debug data.
- Metadata includes the names of the experiment, trial, and job, as well as job parameters (CPU, GPU, and instance type) and artifact locations (e.g. S3 bucket)
- As a developer or data scientist, the last thing you want to do is waste time managing spreadsheets or databases to keep track of trials and their associated entities, as well as their interrelationships.

IV. Dataset

Dataset from UCI Machine Learning Repository has been used for training and testing the ML model. Link to the dataset: *Click here*

V. Technique

We will use neural networks for classifying poker hand ranking.

VI. What is poker ?

According to the rules of poker, participants form sets of five playing cards called hands. Each hand has a rank, which is compared to the ranks of the other hands in the showdown to determine who takes home the prize. The highest-ranking hands win in high games like Texas hold'em and seven-card stud. The lowest-ranking hands win in low games like razz. Both the highest-ranking and lowest-ranking hands win in high-low split games, albeit the high and low hands are ranked differently.

Each hand is assigned to a category based on the patterns created by its cards. A higher-ranking category hand always rates higher than a lower-ranking category hand. The rankings of a hand's cards are used to rank it within its category. From top to lowest, the following cards are ranked: A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3 and 2. Aces, on the other hand, have the lowest rank when used in ace-to-five low or ace-to-six low rules, or when used in a five-high straight or straight flush when using high rules. Because suits aren't ranked, hands that differ only in suit are of the same rank.

When playing a conventional 52-card deck, there are nine different types of hands, with the exception of ace-to-five low rules, which do not recognise straights, flushes, or straight flushes. When one or more wild cards are used, a new category called five of a kind is created.

The fewer hands a category contains, the higher its rank. There are $\frac{52!}{(52-2)!} = 311,875,200$ ways to deal five cards from the deck but only $\frac{52!}{(52-2)!5!} = 2,598,960$ distinct hands, because the order in which cards are dealt or arranged in a hand does not matter. Moreover, since hands differing only by suit are of equal rank, there are only 7,462 distinct hand ranks.

VII. Head ranking categories

VIII. Poker Hands Probability

There are 2,598,960 distinct 5-card hands that can be made/dealt in poker. Let's take a closer look at how these combinations break down into the various ranks of hands that one can attain in this poker probability chart:

Poker Hand	Frequency (Distinct Hands)	Probability (%)	Odds (~)
Royal Flush	4	0.000154%	1 in 649,740
Straight Flush	36	0.00139%	1 in 72,193
4-of-a-Kind	624	0.0240%	1 in 4,165
Full House	3,744	0.1441%	1 in 694
Flush	5,108	0.1965%	1 in 509
Straight	10,200	0.3925%	1 in 255
3-of-a-Kind	54,912	2.1128%	1 in 47
Two Pair	123,552	4.7539%	1 in 21
One Pair	1,098,240	42.2569%	1 in 2.4
High Card	1,302, 540	50.1177%	1 in 2

IX. Additional experiments

The Chances of Getting Specific Hole Cards in Poker Multiply 52 cards in a regular deck by 51 (for the second card) and divide by 2 (AcK deal is the same as KsAc deal, so only the order in which the cards are dealt is different.). At Texas Hold'em, you can get a combination of two cards of 1,326 different types.

If you take a hand like AK, you can multiply the four Aces by four Kings to get a total of 16 different AK combinations, both suited and unsuited. Because there are four suits in poker, four of these AK combinations will be suited, leaving twelve unsuited unpaired hand combinations. (Any non-paired poker hand has the same statistics.) There will be six different permutations of each value for pocket pairs.

With the above information, we can perform some rudimentary algebra to predict the likelihood of receiving specific hole card pairings. For example, if we divide 6 combinations by 1,326 total combinations, we can calculate that we'll get this hand about once every 221 hands on average. Because there are only four combinations of each holding (rather than six as with pocket pairs), you'll get suited non-paired hands (of certain values) less frequently than pocket pairs for every given suited hand. As a result, you'll be dealt a hand like AKs every 1 in 332 hands on average, which means you'll be dealt AAs more often than AKs. To calculate your chances of getting dealt hand X or better, simply add the number of permutations for each holding together.

For example, if we wanted to know the odds of being dealt QQ+ and/or AK, we can see that there are 18 pocket pair combos and 16 AK combos, totaling 34. When you divide 1,326 overall combos by 34 of these precise combos, you'll get AK or QQ+ once every 39 hands. Here's a chart that summarizes the above-mentioned hole card information, as well as various odds for being dealt specific combinations of holdings:

Hand	Total Hand Combinations	Probability (%)	Odds
Any 2 Cards	1326	100%	1 in 1
AK (any specific hand)	16	c	1 in 82.8
AKs (any specific suited hand)	4	0.3%	1 in 331.5
AKo (any specific off-suit hand)	12	0.9%	1 in 110.5
AA (any pocket pair)	6	0.5%	1 in 221
KK+	12	0.9%	1 in 110.5
QQ+	18	1.4%	1 in 73.7
JJ+	24	1.8%	1 in 55.3
TT+	30	2.3%	1 in 44.2
QQ+, AK	34	2.5%	1 in 39
JJ+, AK	40	3.0%	1 in 33.2
TT+, AK	46	3.4%	1 in 28.8
Any Unpaired, Suited Cards	312	23.5%	1 in 4.3
Any Unpaired, Unsuited Cards	936	70.6%	1 in 1.4
Any Pocket Pair	78	5.8%	1 in 17
Any Suited	52	3.9%	1 in 25.5

X. STRUCTURE OF REPOSITORY

Required libraries:

- keras.models
- keras.layers
- numpy
- mathplotlib
- math

This repo contains following files:

- poker-hand-testing.data
- poker-hand-training-true.data
- poker-hand-testing.csv
- poker-hand-training-true.csv
- generate_csv.py
- poker_prediction.ipynb

The poker-hand-testing.csv and poker-hand-training-true.csv files are generated from poker-hand-testing.data and poker-hand-training-true.csv files respectively.

The generate_csv.py file is used to convert .data files to .csv files. It can be used as

python generate_csv.py [filename]

For example,

python generate_csv.py poker-hand-testing.data

This will return a poker-hand-testing.csv file which is used in our model.

The poker_prediction.ipynb file contains all the codes for reading, analysing and predicting data. It also displays results of accuracy and precision of the model.

Note: *For further details go through Readme of the repository*

XI. WORK DISTRIBUTION

- *Rudraksh Kuchiya (20D110021):* Formulating the model, writing the python notebook, fitting the model and generating results; compiling the github repo and readme file, compiling report on Latex.

- *Aarya Chaudhari (20D110002):* Writing the report and presentation
- *Muskan Bhutra (200040085):* Writing the report and presentation
- *Raghav Rander (200040113):* Writing the report and presentation
- *Utkarsh Jindal (200070086):* Writing the report and presentation