**Poker Comb without ML Model Accuracy-1.00**

**Created by Kaushik Kar**

**Employment id- 2216027**

1. **Importing Libraries:**

First, importing the important external Python packages using the pip package manager.

1. swifter is a Python package that provides a faster alternative to the apply method in Pandas. It works by parallelizing the apply method, allowing it to process data faster using multiple CPU cores. The apply method in Pandas is used to apply a function to each row or column of a DataFrame. However, it can be slow when working with large datasets because it is processed sequentially on a single CPU core. swifter addresses this issue by using the Dask library to distribute the apply method across multiple CPU cores.
2. NumPy is used for mathematical operations like addition, subtraction, multiplication, division, etc. on arrays and matrices.
3. Pandas provides data structures for efficiently storing and manipulating large datasets, and tools for reading and writing data to and from various file formats, including CSV, Excel, and SQL databases
4. Seaborn is a data visualization library based on Matplotlib which is a plotting library used for creating static, interactive, and animated visualizations in Python.
5. **Upload and Display the data:**

With the help of pandas library we are able to upload the data and with the .head() method we are able to display the data.

1. **EDA:**

Exploratory data analysis has several things such as .info(), .shape. .describe(), isnull(), unique(), Value\_counts()

* .info() is a method in pandas that can be used to display information about a DataFrame, such as the number of rows and columns, data types, and memory usage.
* .shape is an attribute in pandas that can be used to get the number of rows and columns in a DataFrame. It returns a tuple that contains the number of rows followed by the number of columns, and can be used to quickly check the size and dimensions of a DataFrame.
* The .describe() method provides summary statistics for a DataFrame, such as the count, mean, standard deviation, minimum, and maximum values for each numeric column.
* isnull() is the process by which we can check whether there is any null value present in the data or not i.e. isnull() is part of the data cleaning stage. Here in the data, we found that there is no missing values.
* .nunique() is a method used to calculate the number of unique values in each column of a DataFrame data. It returns a pandas Series where the index is the column name and the values are the number of unique values in that column.
* .value\_counts() is a method used to count the number of occurrences of each unique value in a DataFrame data. It returns a pandas Series where the index is the unique value and the values are the counts of that value in the DataFrame.

Through Visualization we have shown the distribution of the columns from the train\_data.

Now we are trying to make functions of each possibilities of pair. The functions check whether a given set of cards and suits constitutes a certain type of hand (e.g. pair, two pair, three of a kind, straight, flush, full house, etc.).

Each function takes a row of data containing five cards and suits, and uses list comprehensions, loops, and conditional statements to evaluate whether the given hand matches the criteria for the specific type of hand.

The functions can be used as part of a larger program or system that deals with poker hands, such as a poker game or simulator.

This appears to be a code block that creates a new column called "combinations" in a copy of the "train\_data" dataframe. The "swifter" library is used to speed up the application of the "poker\_combinations" function to each row of the "compare\_combinations" dataframe.

The "poker\_combinations" function is likely another function that evaluates poker hands, and it is applied to each row of the dataframe to determine the type of hand that the five cards in that row represent. The resulting hand type is then stored in the "combinations" column for that row.

This process essentially generates a new dataset where each row corresponds to a poker hand represented by five cards, and the "combinations" column contains the type of hand represented by those cards. This new dataset could be used to train a machine learning model to predict the hand type based on the five cards given as input.

Now we are using the loop. This code appears to be a loop that iterates through the integers 0 to 9 and prints information about the distribution of poker hands in the "compare\_combinations" dataset.

For each iteration of the loop, the code prints the current class being evaluated (0 to 9), the true number of samples in the dataset that correspond to that class (determined by counting the number of rows where the "hand" column equals the current class), and the number of samples that the "poker\_combinations" function has predicted to be in that class (determined by counting the number of rows where the "combinations" column equals the current class).

Now uploading the submissonfile and try to predict the test data. This code appears to be performing a prediction on a test dataset in chunks, as the dataset is too large to process at once.

The variable n\_start is initialized to 0, n\_features to 25000, and n\_iterations to the number of chunks needed to cover the entire test dataset with size n\_features.

The loop iterates over n\_iterations and selects a chunk of n\_features rows from the test\_data dataframe using the indices from n\_start to n\_features.

Then, the poker\_combinations function is applied to this chunk using swifter.apply method, which is a faster alternative to pandas.apply. The resulting array of predictions for this chunk is then appended to the result array using numpy.append.

Finally, n\_start and n\_features are updated to select the next chunk of rows.