

output_Playing-Cards-Object-Detection-Dataset

October 26, 2025

0.1 Step1: Data Exploration

Objective: To explore and analyze the Playing Cards dataset in order to understand its structure and characteristics before the modeling phase.

Main Goals: - Perform an initial assessment of data quality and completeness. - Identify attribute types and their distributions. - Detect missing, inconsistent, or anomalous values. - Visualize data through appropriate plots to support interpretation.

Methods and Tools: - Statistical summaries using descriptive measures (mean, median, standard deviation, etc.). - Visual exploration with histograms, boxplots, and scatter plots. - Correlation analysis between features.

Outcome: - Clear overview of the dataset and its key properties. - Identification of relevant features and potential preprocessing needs. - Foundation for subsequent steps in the Intelligent System design process, following the KDD methodology (selection, preprocessing, modeling, and evaluation).

0.1.1 0) Setup

0.1.2 1) Images Dimensions and Aspect Ratios

We analyze the dimensions (width, height) of the images and compute their aspect ratios.

This helps us understand the variability in image sizes and shapes.

We then plot histograms of widths, heights, and aspect ratios.

<IPython.core.display.HTML object>

Processed 75,750 bounding boxes

=====

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

Class	Instances
3s	1588
3h	1580

8d	1562
4h	1557
7d	1550

<IPython.core.display.HTML object>

Class	Instances
9c	1398
5c	1384
9d	1383
3d	1366
Qc	1358

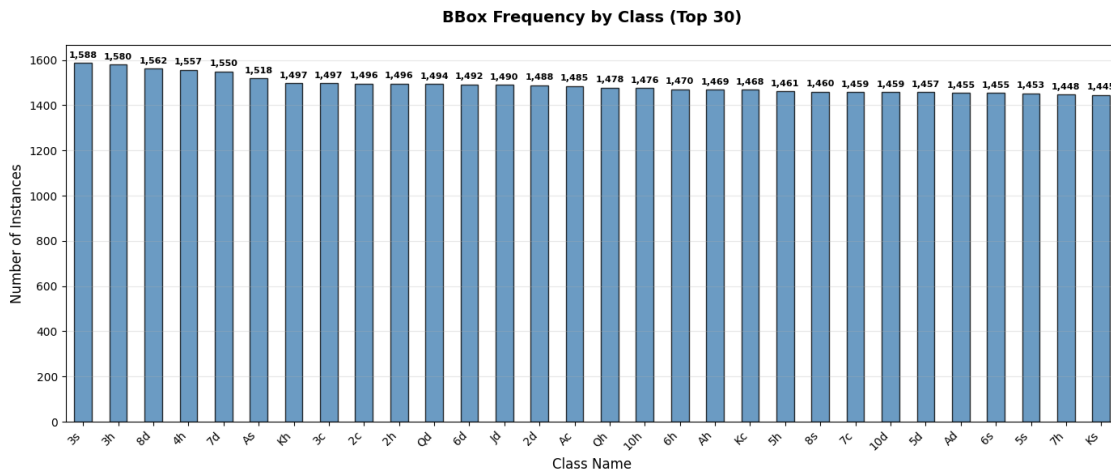
<IPython.core.display.HTML object>

Maximum instances per class:	1,588
Minimum instances per class:	1,358
Mean instances per class:	1,456.73
Median instances per class:	1,455.0
Total unique classes:	52

=====

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>



0.1.3 2) Classes Distribution Analysis

```
=====
GLOBAL CLASS STATISTICS
=====
```

Most frequent classes (Top 5):

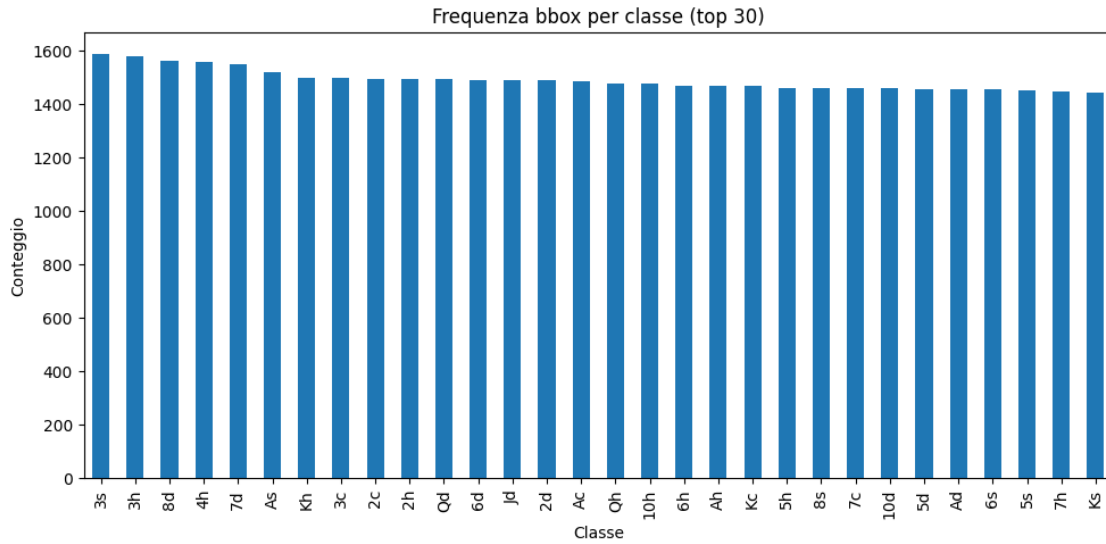
Class	Instances
3s	1588
3h	1580
8d	1562
4h	1557
7d	1550

Least frequent classes (Bottom 5):

Class	Instances
9c	1398
5c	1384
9d	1383
3d	1366
Qc	1358

```
CLASS DISTRIBUTION STATISTICS
-----
```

```
Max instances per class: 1588
Min instances per class: 1358
Mean instances per class: 1456.73
Total unique classes: 52
=====
```



0.1.4 7) Bounding Box Analysis

We check for anomalies in the labels such as: - bounding boxes with normalized coordinates outside [0,1] - bounding boxes with zero area (width or height equal to zero) - images without any bounding boxes

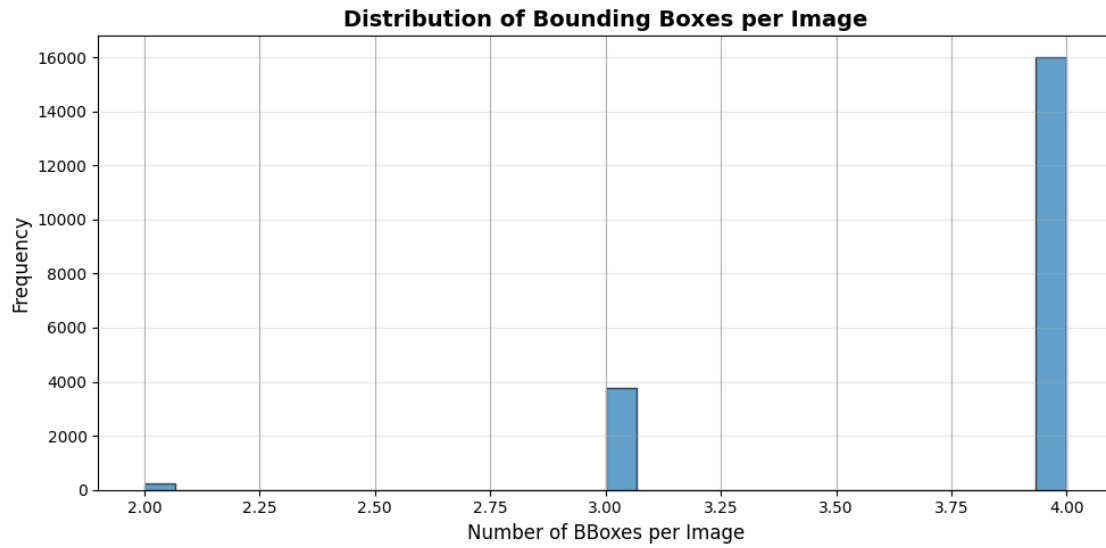
<IPython.core.display.HTML object>

Total bounding boxes:	75,750
BBoxes with coordinates outside [0,1]:	0
BBoxes with near-zero area:	0
BBoxes with very large dimensions (w>0.9 or h>0.9):	0
Images without any bbox:	0.00% (0 / 20,000)

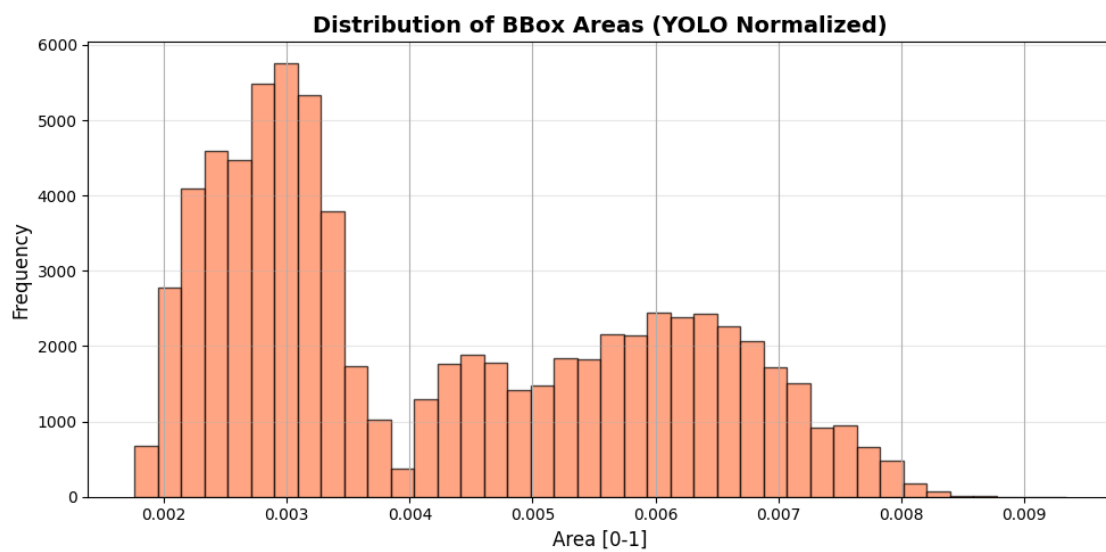
=====

<IPython.core.display.HTML object>

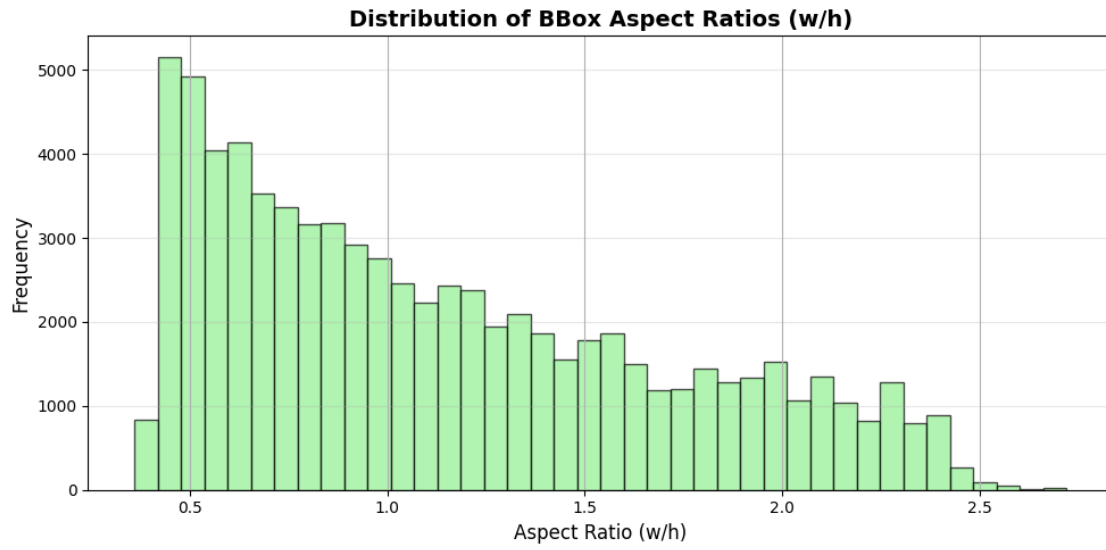
<IPython.core.display.HTML object>



<IPython.core.display.HTML object>

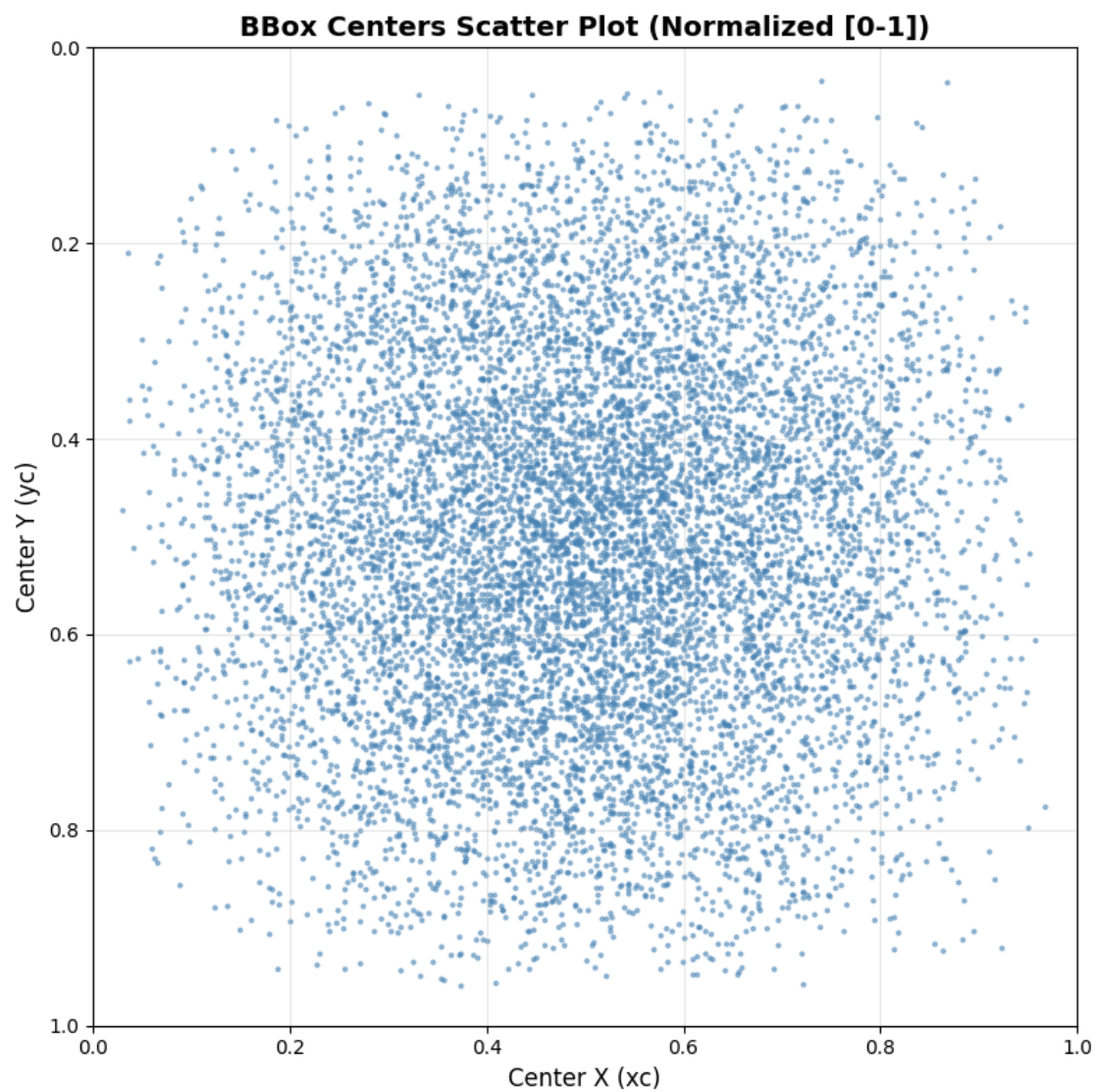


<IPython.core.display.HTML object>

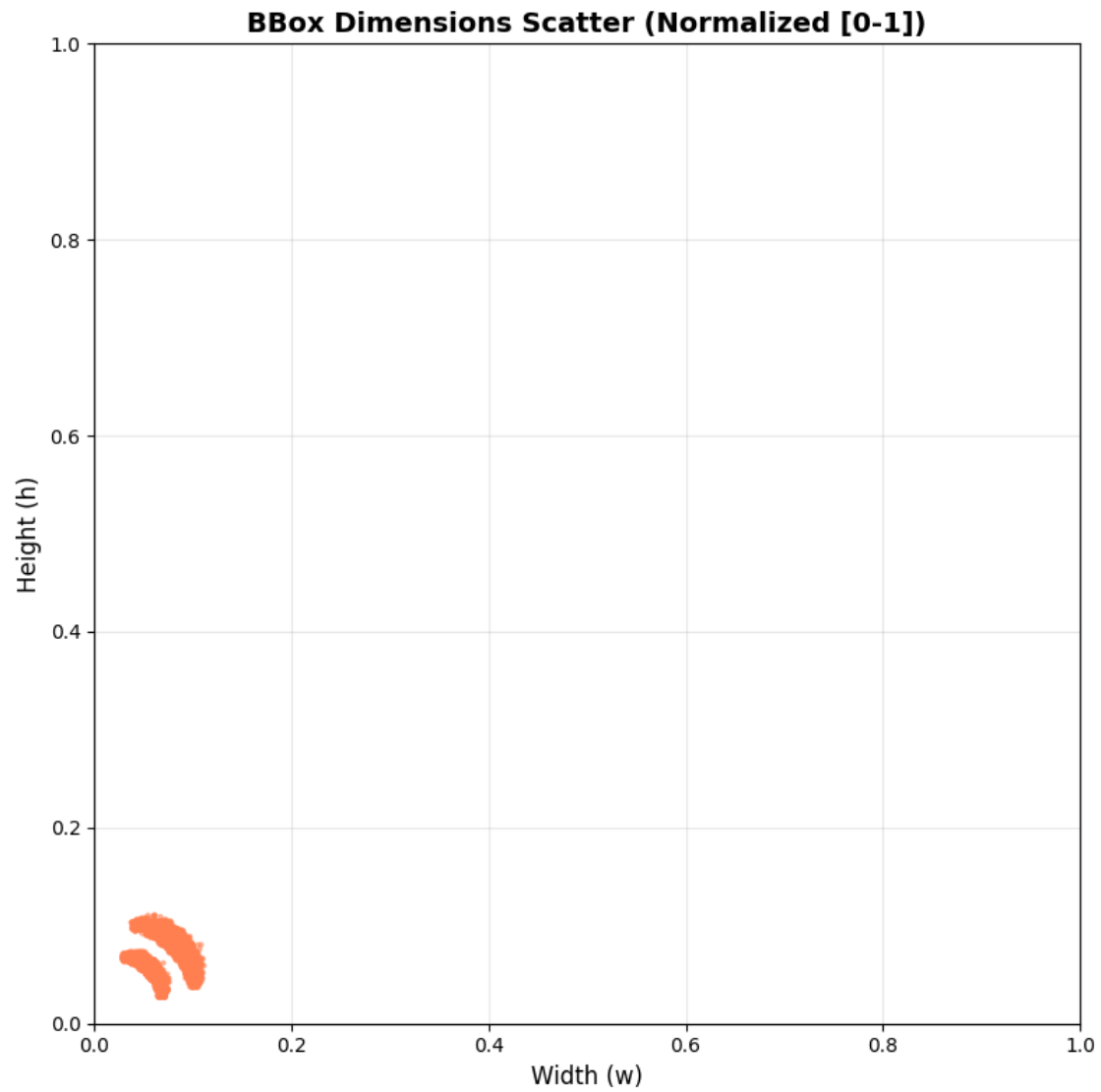


<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

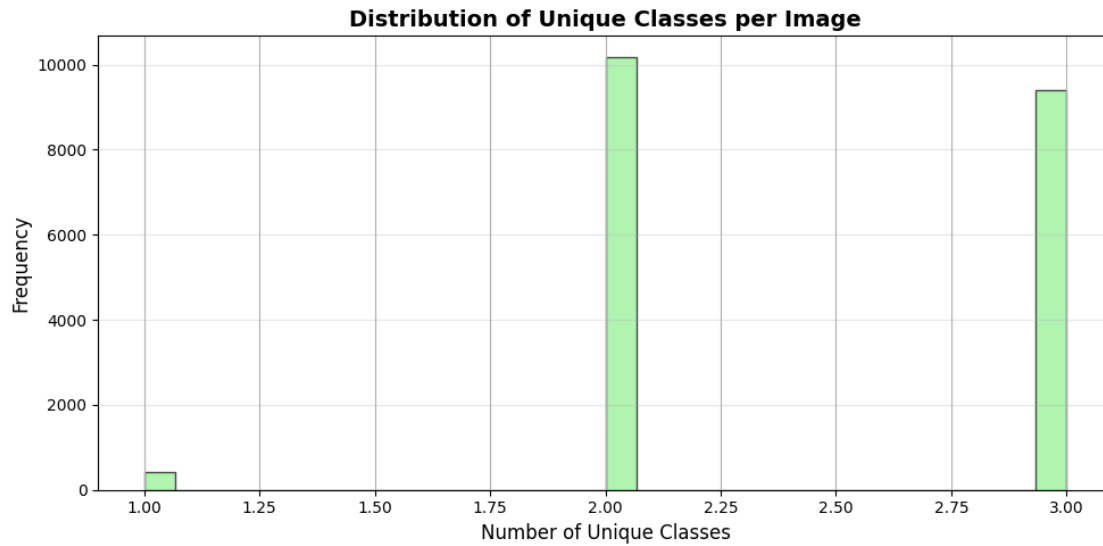


<IPython.core.display.HTML object>



<IPython.core.display.HTML object>

<IPython.core.display.HTML object>



0.1.5 3) Data Visualization

We visualize a few sample images from each split with their corresponding bounding boxes drawn on them.

This helps us qualitatively assess the quality of the annotations and get a better understanding of the dataset content.

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

Co-occurrence matrix computed: 52x52 classes

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

Class 1	Class 2	Co-occurrences
10d	6d	47
5d	Qh	45
4s	6d	44
2d	8s	44
Ac	Js	44
7c	8s	43
6d	Kh	43
6d	Kd	43
5s	Ah	43
5d	Ah	43
2h	4d	43

3s	8d	42
6c	Jd	42
2c	Qh	42
3h	Ac	42

<IPython.core.display.HTML object>

Class 1	Class 2	Co-occurrences
3d	Jh	17
3d	7d	17
3d	5s	17
10c	5d	16
7d	Qc	15
4s	Qc	15
Kd	Qh	14
6s	Qc	14
7h	9s	13
4d	4s	11

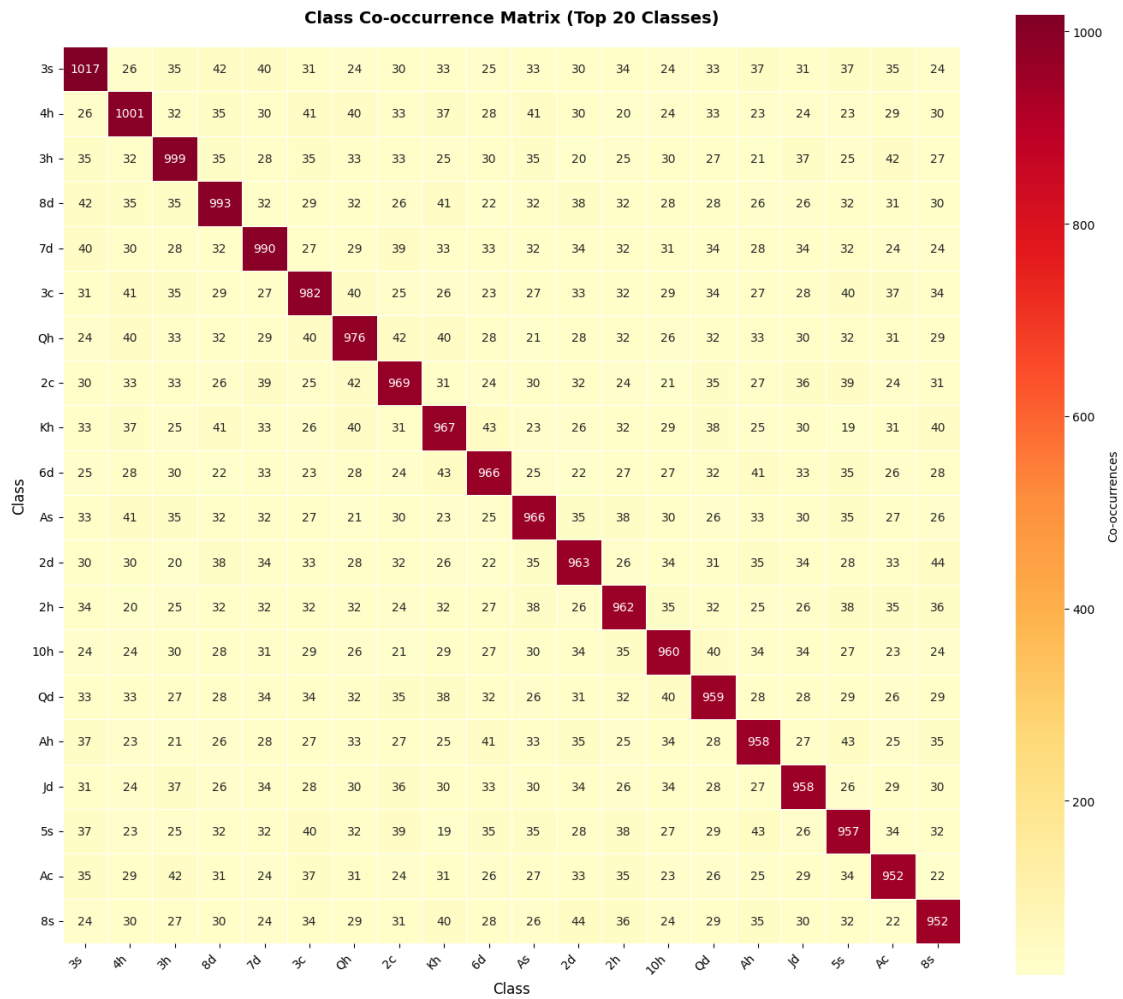
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

Given Class	Then Class	P(Then Given) %
10d	6d	5.01
6d	10d	4.87
5d	Qh	4.74
4s	6d	4.73
4d	2h	4.71
Js	Ac	4.70
Kd	6d	4.67
Ac	Js	4.62
8s	2d	4.62
Qh	5d	4.61
7c	8s	4.59
2d	8s	4.57
6d	4s	4.55
6c	Jd	4.54
Jh	9h	4.54

<IPython.core.display.HTML object>



<IPython.core.display.HTML object>