Rows: 5
Columns: 1 named attack_type

Code for validating by checking for missing values, separating column name, and creating a YAML file

In [1]:
```
pip install pyyaml
```

```
Requirement already satisfied: pyyaml in /Users/parwindersingh/anaconda3/lib/python3.10/site-packages (6.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import yaml
import pandas as pd
from pydantic import BaseModel, ValidationError
```

In [3]:
```
# Created a dataframe for CSV file
data = pd.read_csv('ibeta_info.csv')
```

In [4]:
```
data.head()
```

Out[4]:

	attack_type;description
0	mask;printed portraits of people with cut-out ...
1	mask3d;portraits consisting of several connect...
2	monitor;demonstrations of a person's photo on ...
3	outline;printed outlines of people's photos
4	outline3d;printed portraits of people attached...

In [5]:
```
# Checked column names and type
print(data.columns)
```

```
Index(['attack_type;description'], dtype='object')
```

In [7]:
```
# Checked for missing values in each column
missing_values = data.isnull().sum()
print(missing_values)
```

```
attack_type;description    0
dtype: int64
```

In [10]:
```
# Splited the column names using the semicolon ';'
split_column_names = data.columns.str.split(';')
```

In [12]:
```
# Got the individual column names as a list
individual_column_names = [column[0] for column in split_column_names]
```

In [13]:
```
print(individual_column_names)
```

```
['attack_type']
```

In [15]:
```
# Generated a dictionary containing column names and their corresponding validation results
validation_results = {column: missing_values[column] for column in data.columns}
```

In [16]:
```
# Converted the dictionary to a YAML string
yaml_string = yaml.dump(validation_results)
```

In [17]:
```
# Wrote the YAML string to a YAML file
with open("validation_results.yaml", "w") as f:
    f.write(yaml_string)
```