In [1]:

```
import os
import pandas as pd
import numpy as np
import fnmatch

root = "C:/Users/c21012241/Dropbox"

##### Change paths as required
## 12 features - Binary crystals
path1 = root + "/12 Features - Binary Crystals + No Brightness/\
C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 12U LR 10^-3 E 200 MB 1024 - 12 of 12"

## 12 features - Continuous crystals
path2 = root + "/12 Features - Continuous Crystals + No Bright/\
C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024 - 12U LR10^-3 E200 MB1024 - 12of12"

# For use later
Feature_Analysis_Path_12 = root +"/Cardiff/Dissertation/XAI&I - Dissertation/\
Continuous vs Binary Crystals/12 Features - No Brightness/"

#Feature_Analysis_Path_13 = root + "/Cardiff/Dissertation/XAI&I - Dissertation/\
#Continuous vs Binary Crystals/13 Features"

model_parameter_name = "C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024"

filePath = Feature_Analysis_Path_12 + model_parameter_name + "/"
#filePath = Feature_Analysis_Path_13 + model_parameter_name + "/"


# Create lists of file paths
hybrid_binary_crystals_path1 = sorted([os.path.join(path1, file) for file in os.listdir(path1)], key=os.path.getmtime)
hybrid_continuous_crystals_path2 = sorted([os.path.join(path2, file) for file in os.listdir(path2)], key=os.path.getmtime)

# Rock names and the three variations of image arrangements used during training
rockNamesTen = ["Granite", "Obsidian", "Pegmatite", "Pumice", "Gneiss", "Marble", "Slate", "Breccia", "Conglomerate", "Sandstone"]
Ar1_index = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Ar2_index = [1, 4, 7, 5, 8, 10, 2, 9, 11, 3, 6, 12]
Ar3_index = [1, 6, 9, 2, 7, 10, 3, 8, 11, 4, 5, 12]

#pd.set_option('display.float_format', '{:.10f}'.format)
pd.set_option('display.float_format', '{:.2%}'.format)

# List with 30 rock names i.e. each rock has 3 validation images
all_rock_names = []
for name in rockNamesTen:
    all_rock_names.extend([name] * 3)

# Dictionary to store DataFrames
bi_Val_Hybrids = {key: pd.DataFrame() for key in ['1_2_3', '4_5_6', '7_8_9', '10_11_12', '1_4_7', '5_8_10',
                                                  '2_9_11', '3_6_12', '1_6_9', '2_7_10', '3_8_11', '4_5_12']}
con_Val_Hybrids = bi_Val_Hybrids.copy()

## Two keywords due to spelling mistakes or change of name
#keyword_Hybrid_1 = "netHybrid - Confidence of Prediciton"
#keyword_Hybrid_2 = "netHybrid - Confidence of Prediciton"

keyword_Hybrid_1 = "netHybrid-ConfidencePred"
keyword_Hybrid_2 = "netHybrid-ConfidencePred"

# Keywords of the 12 validation rock orders used
keyword_01_02_03 = "Val_1 2 3"
keyword_04_05_06 = "Val_4 5 6"
keyword_07_08_09 = "Val_7 8 9"
keyword_10_11_12 = "Val_10 11 12"
keyword_01_04_07 = "Val_1 4 7"
keyword_05_08_10 = "Val_5 8 10"
keyword_02_09_11 = "Val_2 9 11"
keyword_03_06_12 = "Val_3 6 12"
keyword_01_06_09 = "Val_1 6 9"
keyword_02_07_10 = "Val_2 7 10"
keyword_03_08_11 = "Val_3 8 11"
keyword_04_05_12 = "Val_4 5 12"
```

In [2]:

```
# A function to read the data from the csv files
def read_DataFrames(file_list, val_Keywords, data_Type_Keyword, all_rock_names):
    dfs = {}
    for file in file_list:
        if any(keyword in file for keyword in val_Keywords) and data_Type_Keyword in file:
            for keyword in val_Keywords:
                if keyword in file:
                    df_name = f"{keyword.replace('*', '').replace('?', '')}"
                    df = pd.read_csv(file, header=None)
                    df.columns = list(all_rock_names)
                    dfs[df_name] = df
                    break  # Break the loop after the first match
    return dfs
```

In [3]:

```
# Define keywords for arrangement 1, 2 and 3
val_Keywords_Ar1 = ["Val_1 2 3", "Val_4 5 6", "Val_7 8 9", "Val_10 11 12"]

val_Keywords_Ar2 = ["Val_1 4 7", "Val_5 8 10", "Val_2 9 11", "Val_3 6 12"]

val_Keywords_Ar3 = ["Val_1 6 9", "Val_2 7 10", "Val_3 8 11", "Val_4 5 12"]

# Ar1
# Read and assign DataFrames for hybrid_binary_crystals_path1
Ar1_bi_Val_Hybrid = read_DataFrames(hybrid_binary_crystals_path1, val_Keywords_Ar1, keyword_Hybrid_1, all_rock_names)
# Read and assign DataFrames for hybrid_continuous_crystals_path2
Ar1_con_Val_Hybrid = read_DataFrames(hybrid_continuous_crystals_path2, val_Keywords_Ar1, keyword_Hybrid_2,all_rock_names)

#Ar2
# Read and assign DataFrames for hybrid_binary_crystals_path1
Ar2_bi_Val_Hybrid = read_DataFrames(hybrid_binary_crystals_path1, val_Keywords_Ar2, keyword_Hybrid_1, all_rock_names)
# Read and assign DataFrames for hybrid_continuous_crystals_path2
Ar2_con_Val_Hybrid = read_DataFrames(hybrid_continuous_crystals_path2, val_Keywords_Ar2, keyword_Hybrid_2, all_rock_names)

#Ar3
# Read and assign DataFrames for hybrid_binary_crystals_path1
Ar3_bi_Val_Hybrid = read_DataFrames(hybrid_binary_crystals_path1, val_Keywords_Ar3, keyword_Hybrid_1, all_rock_names)
# Read and assign DataFrames for hybrid_continuous_crystals_path2
Ar3_con_Val_Hybrid = read_DataFrames(hybrid_continuous_crystals_path2, val_Keywords_Ar3, keyword_Hybrid_2, all_rock_names)
```

In [4]:

```
# Function to name the rocks in each set of three validation images
def arrangeValidationByRockName(dict_of_dfs):
    rock_orders = [
        ("Granite", 0, 3),
        ("Obsidian", 3, 6),
        ("Pegmatite", 6, 9),
        ("Pumice", 9, 12),
        ("Gneiss", 12, 15),
        ("Marble", 15, 18),
        ("Slate", 18, 21),
        ("Breccia", 21, 24),
        ("Conglomerate", 24, 27),
        ("Sandstone", 27, None)
    ]

    categorized_dfs = {rock_name: pd.DataFrame() for rock_name, _, _ in rock_orders}

    for df_name, df in dict_of_dfs.items():
        for rock_name, start, end in rock_orders:
            if end is None:
                categorized_dfs[rock_name] = pd.concat([categorized_dfs[rock_name], df.iloc[:, start:]], axis=1)
            else:
                categorized_dfs[rock_name] = pd.concat([categorized_dfs[rock_name], df.iloc[:, start:end]], axis=1)

    return categorized_dfs.values()
```

In [5]:

```
# Arrange by rock name (Ar1 already is?) and turn into a list
Ar1_bi_Val_Sorted_Name = list(arrangeValidationByRockName(Ar1_bi_Val_Hybrid))
Ar1_con_Val_Sorted_Name = list(arrangeValidationByRockName(Ar1_con_Val_Hybrid))
Ar2_bi_Val_Sorted_Name = list(arrangeValidationByRockName(Ar2_bi_Val_Hybrid))
Ar2_con_Val_Sorted_Name = list(arrangeValidationByRockName(Ar2_con_Val_Hybrid))
Ar3_bi_Val_Sorted_Name = list(arrangeValidationByRockName(Ar3_bi_Val_Hybrid))
Ar3_con_Val_Sorted_Name = list(arrangeValidationByRockName(Ar3_con_Val_Hybrid))

# Extracted DataFrames for each rock type and arrangement
Granite_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[0]
Obsidian_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[1]
Pegmatite_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[2]
Pumice_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[3]
Gneiss_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[4]
Marble_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[5]
Slate_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[6]
Breccia_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[7]
Conglomerate_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[8]
Sandstone_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[9]

Granite_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[0]
Obsidian_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[1]
Pegmatite_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[2]
Pumice_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[3]
Gneiss_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[4]
Marble_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[5]
Slate_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[6]
Breccia_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[7]
Conglomerate_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[8]
Sandstone_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[9]

Granite_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[0]
Obsidian_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[1]
Pegmatite_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[2]
Pumice_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[3]
Gneiss_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[4]
Marble_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[5]
Slate_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[6]
Breccia_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[7]
Conglomerate_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[8]
Sandstone_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[9]

Granite_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[0]
Obsidian_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[1]
Pegmatite_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[2]
Pumice_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[3]
Gneiss_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[4]
Marble_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[5]
Slate_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[6]
Breccia_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[7]
Conglomerate_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[8]
Sandstone_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[9]

Granite_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[0]
Obsidian_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[1]
Pegmatite_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[2]
Pumice_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[3]
Gneiss_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[4]
Marble_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[5]
Slate_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[6]
Breccia_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[7]
Conglomerate_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[8]
Sandstone_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[9]

Granite_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[0]
Obsidian_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[1]
Pegmatite_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[2]
Pumice_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[3]
Gneiss_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[4]
Marble_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[5]
Slate_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[6]
Breccia_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[7]
Conglomerate_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[8]
Sandstone_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[9]
```

In [6]:

```
# Function to rearrange data based on new index variables for arrangement 2 & 3
def setSortIndex(rockName, index):
    rockName = rockName.set_axis([index], axis = 1)
    rockName = rockName.sort_index(axis = 1)
    return rockName
```

In [7]:

```
# Lets use the function then :)

sorted_1to12_Granite_Ar1_bi_Val_Sorted_Name = setSortIndex(Granite_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Obsidian_Ar1_bi_Val_Sorted_Name = setSortIndex(Obsidian_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pegmatite_Ar1_bi_Val_Sorted_Name = setSortIndex(Pegmatite_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pumice_Ar1_bi_Val_Sorted_Name = setSortIndex(Pumice_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Gneiss_Ar1_bi_Val_Sorted_Name = setSortIndex(Gneiss_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Marble_Ar1_bi_Val_Sorted_Name = setSortIndex(Marble_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Slate_Ar1_bi_Val_Sorted_Name = setSortIndex(Slate_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Breccia_Ar1_bi_Val_Sorted_Name = setSortIndex(Breccia_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Conglomerate_Ar1_bi_Val_Sorted_Name = setSortIndex(Conglomerate_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Sandstone_Ar1_bi_Val_Sorted_Name = setSortIndex(Sandstone_Ar1_bi_Val_Sorted_Name, Ar1_index)

sorted_1to12_Granite_Ar1_con_Val_Sorted_Name = setSortIndex(Granite_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Obsidian_Ar1_con_Val_Sorted_Name = setSortIndex(Obsidian_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pegmatite_Ar1_con_Val_Sorted_Name = setSortIndex(Pegmatite_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pumice_Ar1_con_Val_Sorted_Name = setSortIndex(Pumice_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Gneiss_Ar1_con_Val_Sorted_Name = setSortIndex(Gneiss_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Marble_Ar1_con_Val_Sorted_Name = setSortIndex(Marble_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Slate_Ar1_con_Val_Sorted_Name = setSortIndex(Slate_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Breccia_Ar1_con_Val_Sorted_Name = setSortIndex(Breccia_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Conglomerate_Ar1_con_Val_Sorted_Name = setSortIndex(Conglomerate_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Sandstone_Ar1_con_Val_Sorted_Name = setSortIndex(Sandstone_Ar1_con_Val_Sorted_Name, Ar1_index)

sorted_1to12_Granite_Ar2_bi_Val_Sorted_Name = setSortIndex(Granite_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Obsidian_Ar2_bi_Val_Sorted_Name = setSortIndex(Obsidian_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pegmatite_Ar2_bi_Val_Sorted_Name = setSortIndex(Pegmatite_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pumice_Ar2_bi_Val_Sorted_Name = setSortIndex(Pumice_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Gneiss_Ar2_bi_Val_Sorted_Name = setSortIndex(Gneiss_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Marble_Ar2_bi_Val_Sorted_Name = setSortIndex(Marble_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Slate_Ar2_bi_Val_Sorted_Name = setSortIndex(Slate_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Breccia_Ar2_bi_Val_Sorted_Name = setSortIndex(Breccia_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Conglomerate_Ar2_bi_Val_Sorted_Name = setSortIndex(Conglomerate_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Sandstone_Ar2_bi_Val_Sorted_Name = setSortIndex(Sandstone_Ar2_bi_Val_Sorted_Name, Ar2_index)

sorted_1to12_Granite_Ar2_con_Val_Sorted_Name = setSortIndex(Granite_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Obsidian_Ar2_con_Val_Sorted_Name = setSortIndex(Obsidian_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pegmatite_Ar2_con_Val_Sorted_Name = setSortIndex(Pegmatite_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pumice_Ar2_con_Val_Sorted_Name = setSortIndex(Pumice_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Gneiss_Ar2_con_Val_Sorted_Name = setSortIndex(Gneiss_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Marble_Ar2_con_Val_Sorted_Name = setSortIndex(Marble_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Slate_Ar2_con_Val_Sorted_Name = setSortIndex(Slate_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Breccia_Ar2_con_Val_Sorted_Name = setSortIndex(Breccia_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Conglomerate_Ar2_con_Val_Sorted_Name = setSortIndex(Conglomerate_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Sandstone_Ar2_con_Val_Sorted_Name = setSortIndex(Sandstone_Ar2_con_Val_Sorted_Name, Ar2_index)

sorted_1to12_Granite_Ar3_bi_Val_Sorted_Name = setSortIndex(Granite_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Obsidian_Ar3_bi_Val_Sorted_Name = setSortIndex(Obsidian_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pegmatite_Ar3_bi_Val_Sorted_Name = setSortIndex(Pegmatite_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pumice_Ar3_bi_Val_Sorted_Name = setSortIndex(Pumice_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Gneiss_Ar3_bi_Val_Sorted_Name = setSortIndex(Gneiss_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Marble_Ar3_bi_Val_Sorted_Name = setSortIndex(Marble_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Slate_Ar3_bi_Val_Sorted_Name = setSortIndex(Slate_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Breccia_Ar3_bi_Val_Sorted_Name = setSortIndex(Breccia_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Conglomerate_Ar3_bi_Val_Sorted_Name = setSortIndex(Conglomerate_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Sandstone_Ar3_bi_Val_Sorted_Name = setSortIndex(Sandstone_Ar3_bi_Val_Sorted_Name, Ar3_index)

sorted_1to12_Granite_Ar3_con_Val_Sorted_Name = setSortIndex(Granite_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Obsidian_Ar3_con_Val_Sorted_Name = setSortIndex(Obsidian_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pegmatite_Ar3_con_Val_Sorted_Name = setSortIndex(Pegmatite_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pumice_Ar3_con_Val_Sorted_Name = setSortIndex(Pumice_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Gneiss_Ar3_con_Val_Sorted_Name = setSortIndex(Gneiss_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Marble_Ar3_con_Val_Sorted_Name = setSortIndex(Marble_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Slate_Ar3_con_Val_Sorted_Name = setSortIndex(Slate_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Breccia_Ar3_con_Val_Sorted_Name = setSortIndex(Breccia_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Conglomerate_Ar3_con_Val_Sorted_Name = setSortIndex(Conglomerate_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Sandstone_Ar3_con_Val_Sorted_Name = setSortIndex(Sandstone_Ar3_con_Val_Sorted_Name, Ar3_index)
```

In [8]:

```
# Does what it says on the tin

def meanRockArrangements(Ar1, Ar2, Ar3):
    """Returns the mean of the three rock arrangements"""
    return (Ar1 + Ar2 + Ar3) / 3
```

In [9]:

```
# Dictionary of binary and continuous sets

arrangements_bi = ["Ar1_bi_Val_Sorted_Name", "Ar2_bi_Val_Sorted_Name", "Ar3_bi_Val_Sorted_Name"]

mean_accuracy_results_bi = {}

for rock_name in rockNamesTen:
    accuracy_list = []
    for arrangement in arrangements_bi:
        dataFrame_name = f"sorted_1to12_{rock_name}_{arrangement}"
        accuracy_list.append(eval(dataFrame_name))

    mean_accuracy = meanRockArrangements(*accuracy_list)
    mean_accuracy_results_bi[rock_name] = mean_accuracy

arrangements_con = ["Ar1_con_Val_Sorted_Name", "Ar2_con_Val_Sorted_Name", "Ar3_con_Val_Sorted_Name"]

mean_accuracy_results_con = {}

for rock_name in rockNamesTen:
    accuracy_list = []
    for arrangement in arrangements_con:
        dataFrame_name = f"sorted_1to12_{rock_name}_{arrangement}"
        accuracy_list.append(eval(dataFrame_name))

    mean_accuracy = meanRockArrangements(*accuracy_list)
    mean_accuracy_results_con[rock_name] = mean_accuracy
```

In [10]:

```
## Binary crystal rating by rock name

mean_accuracy_results_bi_key = next(iter(mean_accuracy_results_bi))

# Granite mean confidence
Granite_bi_mean_accuracy_results_key = list(mean_accuracy_results_bi.keys())[0]
Granite_bi_mean_accuracy_results = mean_accuracy_results_bi[Granite_bi_mean_accuracy_results_key].rename(index=dict(enumerate(rockNamesTen)))
# Obsidian mean confidence
Obsidian_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[1]
Obsidian_bi_mean_accuracy_results = mean_accuracy_results_bi[Obsidian_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Pegmatite mean confidence
Pegmatite_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[2]
Pegmatite_bi_mean_accuracy_results = mean_accuracy_results_bi[Pegmatite_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Pumice mean confidence
Pumice_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[3]
Pumice_bi_mean_accuracy_results = mean_accuracy_results_bi[Pumice_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Gneiss mean confidence
Gneiss_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[4]
Gneiss_bi_mean_accuracy_results = mean_accuracy_results_bi[Gneiss_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Marble mean confidence
Marble_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[5]
Marble_bi_mean_accuracy_results = mean_accuracy_results_bi[Marble_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Slate mean confidence
Slate_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[6]
Slate_bi_mean_accuracy_results = mean_accuracy_results_bi[Slate_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Breccia mean confidence
Breccia_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[7]
Breccia_bi_mean_accuracy_results = mean_accuracy_results_bi[Breccia_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Conglomerate mean confidence
Conglomerate_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[8]
Conglomerate_bi_mean_accuracy_results = mean_accuracy_results_bi[Conglomerate_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Sandstone mean confidence
Sandstone_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[9]
Sandstone_bi_mean_accuracy_results = mean_accuracy_results_bi[Sandstone_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))

## Continuous crystal rating by rock name

mean_accuracy_results_con_key = next(iter(mean_accuracy_results_con))

# Granite mean confidence
Granite_con_mean_accuracy_results_key = list(mean_accuracy_results_con.keys())[0]
Granite_con_mean_accuracy_results = mean_accuracy_results_con[Granite_con_mean_accuracy_results_key].rename(index=dict(enumerate(rockNamesTen)))
# Obsidian mean confidence
Obsidian_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[1]
Obsidian_con_mean_accuracy_results = mean_accuracy_results_con[Obsidian_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Pegmatite mean confidence
Pegmatite_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[2]
Pegmatite_con_mean_accuracy_results = mean_accuracy_results_con[Pegmatite_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Pumice mean confidence
Pumice_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[3]
Pumice_con_mean_accuracy_results = mean_accuracy_results_con[Pumice_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Gneiss mean confidence
Gneiss_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[4]
Gneiss_con_mean_accuracy_results = mean_accuracy_results_con[Gneiss_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Marble mean confidence
Marble_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[5]
Marble_con_mean_accuracy_results = mean_accuracy_results_con[Marble_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Slate mean confidence
Slate_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[6]
Slate_con_mean_accuracy_results = mean_accuracy_results_con[Slate_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Breccia mean confidence
Breccia_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[7]
Breccia_con_mean_accuracy_results = mean_accuracy_results_con[Breccia_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Conglomerate mean confidence
Conglomerate_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[8]
Conglomerate_con_mean_accuracy_results = mean_accuracy_results_con[Conglomerate_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Sandstone mean confidence
Sandstone_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[9]
Sandstone_con_mean_accuracy_results = mean_accuracy_results_con[Sandstone_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
```

In [11]:

```
Feature_Analysis_Path_12 = root + "/Cardiff/Dissertation/XAI&I - Dissertation/Continuous vs Binary Crystals/12 Features - No Brightness/"
#Feature_Analysis_Path_13 = "C:/Users/c21012241/Dropbox/Cardiff/Dissertation/XAI&I - Dissertation/Continuous vs Binary Crystals/13 Features"

model_parameter_name = "C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024"

filePath = Feature_Analysis_Path_12 + model_parameter_name + "/"
#filePath = Feature_Analysis_Path_13 + model_parameter_name + "/"

Granite_Comparison = Granite_con_mean_accuracy_results.compare(Granite_bi_mean_accuracy_results)
Granite_Comparison = Granite_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Granite_Comparison.to_csv(filePath + "Granite_Comparison" + ".csv")

Obsidian_Comparison = Obsidian_con_mean_accuracy_results.compare(Obsidian_bi_mean_accuracy_results)
Obsidian_Comparison = Obsidian_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Obsidian_Comparison.to_csv(filePath + "Obsidian_Comparison" + ".csv")

Pegmatite_Comparison = Pegmatite_con_mean_accuracy_results.compare(Pegmatite_bi_mean_accuracy_results)
Pegmatite_Comparison = Pegmatite_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Pegmatite_Comparison.to_csv(filePath + "Pegmatite_Comparison" + ".csv")

Pumice_Comparison = Pumice_con_mean_accuracy_results.compare(Pumice_bi_mean_accuracy_results)
Pumice_Comparison = Pumice_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Pumice_Comparison.to_csv(filePath + "Pumice_Comparison" + ".csv")

Gneiss_Comparison = Gneiss_con_mean_accuracy_results.compare(Gneiss_bi_mean_accuracy_results)
Gneiss_Comparison = Gneiss_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Gneiss_Comparison.to_csv(filePath + "Gneiss_Comparison" + ".csv")

Marble_Comparison = Marble_con_mean_accuracy_results.compare(Marble_bi_mean_accuracy_results)
Marble_Comparison = Marble_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Marble_Comparison.to_csv(filePath + "Marble_Comparison" + ".csv")

Slate_Comparison = Slate_con_mean_accuracy_results.compare(Slate_bi_mean_accuracy_results)
Slate_Comparison = Slate_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Slate_Comparison.to_csv(filePath + "Slate_Comparison" + ".csv")

Breccia_Comparison = Breccia_con_mean_accuracy_results.compare(Breccia_bi_mean_accuracy_results)
Breccia_Comparison = Breccia_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Breccia_Comparison.to_csv(filePath + "Breccia_Comparison" + ".csv")

Conglomerate_Comparison = Conglomerate_con_mean_accuracy_results.compare(Conglomerate_bi_mean_accuracy_results)
Conglomerate_Comparison = Conglomerate_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Conglomerate_Comparison.to_csv(filePath + "Conglomerate_Comparison" + ".csv")

Sandstone_Comparison = Sandstone_con_mean_accuracy_results.compare(Sandstone_bi_mean_accuracy_results)
Sandstone_Comparison = Sandstone_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Sandstone_Comparison.to_csv(filePath + "Sandstone_Comparison" + ".csv")
```

In [12]:

```
print(Granite_Comparison)
```

```
                1                2                3                4   \
           Continuous Binary Continuous Binary Continuous Binary Continuous
Granite         99.96% 99.95%     99.82% 99.49%     91.86% 93.96%     99.48%
Obsidian         0.00%  0.00%      0.00%  0.00%      0.07%  0.02%      0.00%
Pegmatite        0.03%  0.05%      0.03%  0.02%      0.86%  0.53%      0.01%
Pumice           0.00%  0.00%      0.00%  0.00%      0.00%  0.00%      0.00%
Gneiss           0.00%  0.00%      0.03%  0.09%      5.66%  4.54%      0.51%
Marble           0.00%  0.00%      0.11%  0.39%      0.03%  0.27%      0.00%
Slate            0.00%  0.00%      0.00%  0.00%      0.00%  0.01%      0.00%
Breccia          0.00%  0.00%      0.00%  0.00%      0.60%  0.26%      0.00%
Conglomerate     0.00%  0.00%      0.00%  0.00%      0.92%  0.41%      0.00%
Sandstone        0.00%  0.00%      0.00%  0.00%      0.00%  0.00%      0.00%

                    5          ...         8                9   \
           Binary Continuous Binary  ... Continuous Binary Continuous
Granite    98.28%     84.00% 91.13%  ...     66.05% 60.80%     89.49%
Obsidian    0.00%      0.01%  0.03%  ...      0.01%  0.10%      0.04%
Pegmatite   0.00%      8.75%  3.77%  ...     14.66% 19.95%      3.00%
Pumice      0.00%      0.00%  0.02%  ...      0.00%  0.00%      0.00%
Gneiss      1.71%      5.93%  4.52%  ...     11.12% 12.73%      0.01%
Marble      0.00%      0.11%  0.02%  ...      0.05%  1.36%      2.00%
Slate       0.00%      0.00%  0.01%  ...      0.00%  0.00%      0.00%
Breccia     0.00%      0.30%  0.14%  ...      7.21%  3.82%      2.50%
Conglomerate 0.00%     0.87%  0.34%  ...      0.89%  1.23%      2.96%
Sandstone   0.00%      0.03%  0.01%  ...      0.00%  0.01%      0.00%

                   10               11               12
           Binary Continuous Binary Continuous Binary Continuous Binary
Granite    83.42%     94.81% 90.78%     70.77% 78.28%     77.48% 66.93%
Obsidian    0.54%      0.00%  0.00%      0.01%  0.02%      0.07%  0.10%
Pegmatite   9.00%      5.19%  9.22%     27.91% 19.20%      9.25% 17.80%
Pumice      0.00%      0.00%  0.00%      0.00%  0.00%      0.00%  0.00%
Gneiss      0.00%      0.00%  0.00%      0.12%  0.19%      0.12%  0.02%
Marble      5.34%      0.00%  0.00%      1.10%  1.86%      3.63%  5.31%
Slate       0.00%      0.00%  0.00%      0.00%  0.00%      0.02%  0.05%
Breccia     0.36%      0.00%  0.00%      0.01%  0.02%      6.95%  7.14%
Conglomerate 1.33%     0.00%  0.00%      0.08%  0.42%      2.48%  2.64%
Sandstone   0.00%      0.00%  0.00%      0.00%  0.00%      0.00%  0.01%

[10 rows x 24 columns]
```