Usairim Isani
Student number 2206784
February, 6, 2023

Exercise 2 | TKO_7092 Evaluation of Machine Learning Methods 2023

Prediction of the metal ion content from multi-parameter data

Use K-Nearest Neighbor Regression with euclidean distance to predict total metal concentration (c_total), concentration of Cadmium (Cd) and concentration of Lead (Pb), for each sample using number of neighbors k = 3.

```
- You may use Nearest Neighbor Regression from
https://scikit-learn.org/stable/modules/neighbors.html
- The data should be standarized using z-score. (Using
sklearn.preprocessing.StandardScaler is allowed)
- Implement your own Leave-One-Out cross-validation and
calculate the C-index for each output (c_total, Cd, Pb).
- Implement your own Leave-Replicas-Out cross-validation
and calculate the C-index for each output (c_total, Cd,
Pb).
- Return your solution as a Jupyter Notebook .ipynb
notebook and as a PDF-file made from it.
- Submit to moodle your solution on ** Wednesday 8 of
February ** at the latest.
```

Import libraries

In [ ]:
```
#In this cell import all libraries you need. For example:
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsRegressor
```

Read and visualize the dataset

In [ ]:
```
#In this cell read the file Water_data.csv
#Print the dataset dimesions (i.e. number of rows and columns)
#Print the first 5 rows of the dataset

water_df = pd.read_csv("./Water_data.csv")
print(f'Rows     : {water_df.shape[0]}\nColumns  : {water_df.shape[1]}\n'

features = ['Mod1', 'Mod2', 'Mod3']
labels = ['c_total', 'Cd', 'Pb']

water_df.head(5)
```

```
Rows      : 225
Columns   : 6
```

Out[ ]:

	c_total	Cd	Pb	Mod1	Mod2	Mod3
0	0	0.0	0.0	9945	119	72335
1	0	0.0	0.0	10786	117	82977
2	0	0.0	0.0	10812	120	98594
3	14	0.0	14.0	9742	127	154323
4	14	0.0	14.0	8495	120	131672

To show understanding of the data, answer the following questions:

- How many different mixtures of Cadmium (Cd) and Lead (Pb) were measured?
- How many total concentrations (c_total) were measured?
- How many mixtures have less than 4 replicas?
- Make plots of Lead (Pb) and Cadmium (Cd) mixtures for low and high concentrations.

Where low concentrations are those with c_total <= 100, while in high concentration c_total > 100.
Hint: plots are similar to the ones presented in the video lecture.

In [ ]:
```
# In this cell write the code to answer the previous questions and print

mixtures = water_df.groupby(["Cd", "Pb"]).count()

print(mixtures)
print()
print(f'There are {mixtures.shape[0]} Mixtures of Cd and Pb\n\n')
```

```
               c_total  Mod1  Mod2  Mod3
Cd     Pb
0.0    0.0           3     3     3     3
       14.0          3     3     3     3
       20.0          3     3     3     3
       35.0          3     3     3     3
       50.0          4     4     4     4
...                ...   ...   ...   ...
2000.0 0.0           3     3     3     3
       3000.0        3     3     3     3
3000.0 2000.0        3     3     3     3
4000.0 1000.0        3     3     3     3
5000.0 0.0           3     3     3     3

[67 rows x 4 columns]

There are 67 Mixtures of Cd and Pb
```

In [ ]:
```
c_total = water_df.groupby(["c_total"])["c_total"].count()

print(c_total)
```

```
print()
print(f'There are {c_total.shape[0]} concentrations\n\n')
```

```
c_total
0        3
14      18
20      18
35      18
50      24
70      24
100     24
200     24
500     18
1000    18
2000    18
5000    18
Name: c_total, dtype: int64

There are 12 concentrations
```

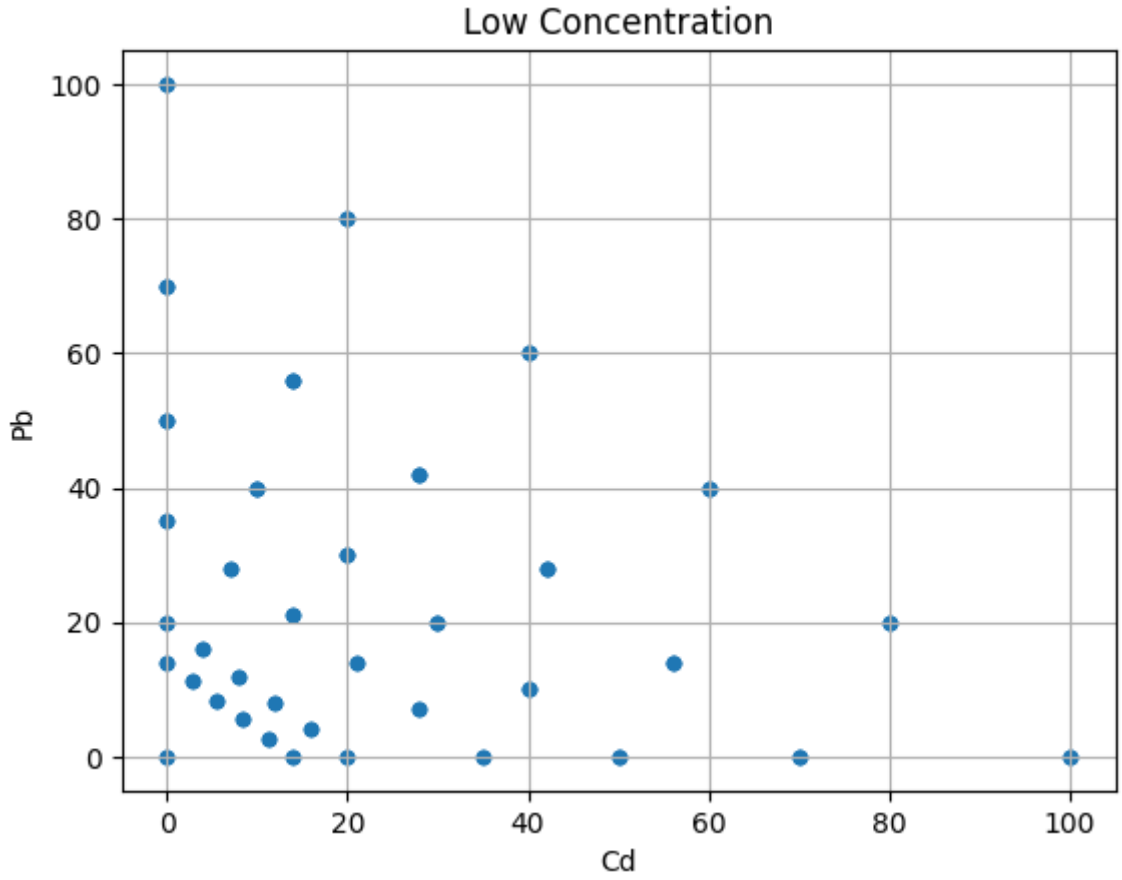
In [ ]:
```
replicas_less_4 = mixtures[mixtures["c_total"] < 4].shape[0]
print(f'Mixtures with less than 4 replicas are : {replicas_less_4}\n\n')
```

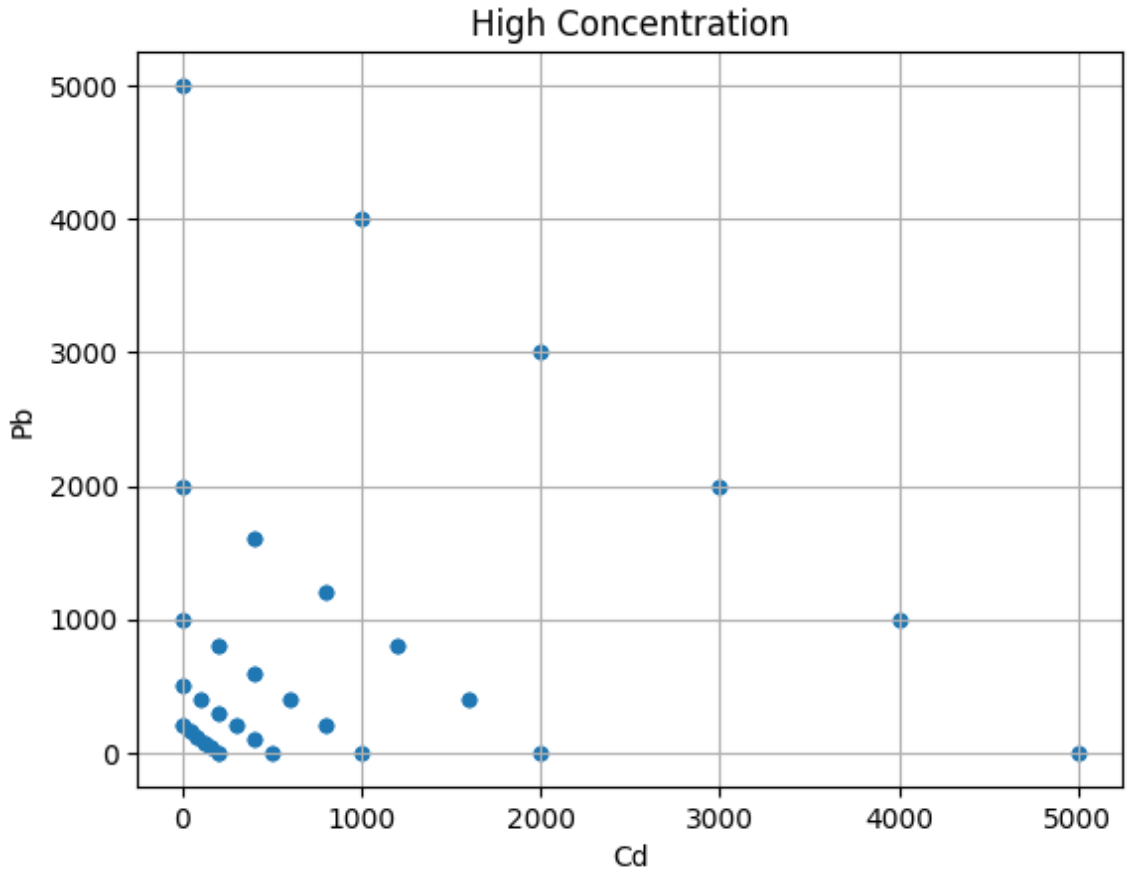
```
Mixtures with less than 4 replicas are : 43
```

In [ ]:
```
low_conc = water_df[water_df["c_total"]<= 100]
high_conc = water_df[water_df["c_total"] > 100]


ax = low_conc.plot.scatter(x="Cd", y="Pb")
ax.set_title("Low Concentration")
ax.grid()

ax = high_conc.plot.scatter(x="Cd", y="Pb")
ax.set_title("High Concentration")
ax.grid()
```





Standardization of the dataset

In [ ]:
```
#In this cell standardize the dataset features by removing the mean and s
#In other words, use z-score to scale the dataset features (Mod1, Mod2, M
```

```
#Print the 5 first samples (i.e. rows) of the scaled dataset
ss = StandardScaler()
water_df[["Mod1", "Mod2", "Mod3"]] = ss.fit_transform(water_df[["Mod1", "
water_df.head(5)
```

Out[ ]:

	c_total	Cd	Pb	Mod1	Mod2	Mod3
0	0	0.0	0.0	-0.999216	-0.714208	-0.414911
1	0	0.0	0.0	-0.990800	-0.714373	-0.238335
2	0	0.0	0.0	-0.990539	-0.714125	0.020788
3	14	0.0	14.0	-1.001247	-0.713546	0.945465
4	14	0.0	14.0	-1.013727	-0.714125	0.569631

C-index code

In [ ]:

```
def cindex(true_labels, pred_labels):
    """Returns C-index between true labels and predicted labels"""
    n = 0
    h_num = 0
    for i in range(0, len(true_labels)):
        t = true_labels[i]
        p = pred_labels[i]
        for j in range(i+1, len(true_labels)):
            nt = true_labels[j]
            np = pred_labels[j]
            if (t != nt):
                n = n + 1
                if (p < np and t < nt) or (p > np and t > nt):
                    h_num += 1
                elif (p == np):
                    h_num += 0.5
    return h_num/n
```

In [ ]:

```
#test cindex function with following values
true_labels = [-1, 1, 1, -1, 1]
predictions = [0.60, 0.80, 0.75, 0.75, 0.70]
cindx = cindex(true_labels, predictions)
print(cindx)
```

```
0.75
```

Functions

Include here all the functions that you need to run in the data analysis part.

Note: using a leave-one-out and leave-replicas-out cross-validation from an already made package (e.g. Scikit-learn) is not accepted.

In [ ]:

```
def LeaveOneOut(data):
    data = [*range(data)]
    for i in data:
        test_index = i
        train_index = []

        for j in data:
```

```
            if j != test_index:
                train_index.append(j)
        yield train_index, [test_index]

def LeaveReplicasOut(df_grouped, df_rows):
    # Data frame with grouped data
    # This group the data into group where replicate are present
    for _, group in df_grouped:
        print(group)
        # Indices for the grouped data which will be used to remove from
        test_index = list(group.index)
        train_index = []

        # loop over the rows and add only those which are not in the grou
        for j in range(df_rows):
            if j not in test_index:
                train_index.append(j)
        yield train_index, test_index
```

Results for Leave-One-Out cross-validation

In [ ]:
```
# In this cell run your script for Leave-One-Out cross-validation and pri

knn = KNeighborsRegressor(n_neighbors=3)

c_total_pred, c_total_true = [], []
Cd_pred, Cd_true = [], []
Pb_pred, Pb_true = [], []

for _, (train_index, test_index) in enumerate(LeaveOneOut(water_df.shape[

    X_train, y_train = water_df.loc[train_index,
                                    features].values, water_df.loc[train_
    X_test, y_test = water_df.loc[test_index,
                                  features].values, water_df.loc[test_ind

    knn.fit(X_train, y_train)

    prediction = knn.predict(X_test)[0]
    true_label = y_test[0]

    c_total_pred.append(prediction[0])
    c_total_true.append(true_label[0])

    Cd_pred.append(prediction[1])
    Cd_true.append(true_label[1])

    Pb_pred.append(prediction[2])
    Pb_true.append(true_label[2])
```

In [ ]:
```
print(f'The cindex for c_total is {cindex(c_total_true, c_total_pred)}\n'
print(f'The cindex for Cd is {cindex(Cd_true, Cd_pred)}\n')
print(f'The cindex for Pb is {cindex(Pb_true, Pb_pred)}\n')
```

```
The cindex for c_total is 0.9141907740422205

The cindex for Cd is 0.8995907629348144

The cindex for Pb is 0.8744519146448407
```

Results for Leave-Replicas-Out cross-validation

In [ ]:
```
# In this cell run your script for Leave-Replicas-Out cross-validation ar
knn = KNeighborsRegressor(n_neighbors=3)
df_grouped = water_df.groupby(['c_total', 'Cd', 'Pb'])

c_total_pred, c_total_true = [], []
Cd_pred, Cd_true = [], []
Pb_pred, Pb_true = [], []

for _, (train_index, test_index) in enumerate(LeaveReplicasOut(df_grouped

    X_train, y_train = water_df.loc[train_index,
                                    features].values, water_df.loc[train_
    X_test, y_test = water_df.loc[test_index,
                                  features].values, water_df.loc[test_ind

    knn.fit(X_train, y_train)

    prediction = knn.predict(X_test)[0]
    true_label = y_test[0]

    c_total_pred.append(prediction[0])
    c_total_true.append(true_label[0])

    Cd_pred.append(prediction[1])
    Cd_true.append(true_label[1])

    Pb_pred.append(prediction[2])
    Pb_true.append(true_label[2])
```

	c_total	Cd	Pb	Mod1	Mod2	Mod3
0	0	0.0	0.0	-0.999216	-0.714208	-0.414911
1	0	0.0	0.0	-0.990800	-0.714373	-0.238335
2	0	0.0	0.0	-0.990539	-0.714125	0.020788
	c_total	Cd	Pb	Mod1	Mod2	Mod3
3	14	0.0	14.0	-1.001247	-0.713546	0.945465
4	14	0.0	14.0	-1.013727	-0.714125	0.569631
5	14	0.0	14.0	-0.998816	-0.713629	0.691867
	c_total	Cd	Pb	Mod1	Mod2	Mod3
6	14	2.8	11.2	-1.015698	-0.714704	0.031490
7	14	2.8	11.2	-1.013046	-0.713298	0.276062
8	14	2.8	11.2	-1.015348	-0.715034	0.450498
	c_total	Cd	Pb	Mod1	Mod2	Mod3
9	14	5.6	8.4	-0.999876	-0.713298	-0.156950
10	14	5.6	8.4	-0.994542	-0.715696	-0.030300
11	14	5.6	8.4	-0.988768	-0.714208	0.792201
	c_total	Cd	Pb	Mod1	Mod2	Mod3
12	14	8.4	5.6	-0.903684	-0.713050	1.476272
13	14	8.4	5.6	-0.910860	-0.714290	0.389355
14	14	8.4	5.6	-0.909058	-0.714125	0.858455
	c_total	Cd	Pb	Mod1	Mod2	Mod3
15	14	11.2	2.8	-0.935218	-0.713381	0.255056
16	14	11.2	2.8	-0.929043	-0.712637	0.937766
17	14	11.2	2.8	-0.911220	-0.712306	1.070107
	c_total	Cd	Pb	Mod1	Mod2	Mod3
18	14	14.0	0.0	-0.864275	-0.712802	-0.579259
19	14	14.0	0.0	-0.749019	-0.701145	-0.328598
20	14	14.0	0.0	-0.817740	-0.689570	-0.178785
	c_total	Cd	Pb	Mod1	Mod2	Mod3
21	20	0.0	20.0	-0.991740	-0.711562	0.648611
22	20	0.0	20.0	-0.982333	-0.712637	0.507061
23	20	0.0	20.0	-0.981893	-0.712719	0.749194
	c_total	Cd	Pb	Mod1	Mod2	Mod3
24	20	4.0	16.0	-0.999156	-0.712967	0.882497
25	20	4.0	16.0	-1.000817	-0.712637	1.032260
26	20	4.0	16.0	-0.989368	-0.713215	1.358981
	c_total	Cd	Pb	Mod1	Mod2	Mod3
27	20	8.0	12.0	-0.969253	-0.713050	0.189881
28	20	8.0	12.0	-0.935969	-0.708751	0.507277
29	20	8.0	12.0	-0.930595	-0.707924	0.517697
	c_total	Cd	Pb	Mod1	Mod2	Mod3
30	20	12.0	8.0	-1.019491	-0.707263	0.055400
31	20	12.0	8.0	-1.008353	-0.710900	0.505949
32	20	12.0	8.0	-1.011385	-0.711397	0.316581
	c_total	Cd	Pb	Mod1	Mod2	Mod3
33	20	16.0	4.0	-0.974297	-0.711645	0.731490
34	20	16.0	4.0	-0.993902	-0.711975	0.949331
35	20	16.0	4.0	-0.976159	-0.713298	1.280930
	c_total	Cd	Pb	Mod1	Mod2	Mod3
36	20	20.0	0.0	-0.945566	-0.708668	0.868692
37	20	20.0	0.0	-0.918536	-0.703956	1.196442
38	20	20.0	0.0	-0.921808	-0.703956	1.465039
	c_total	Cd	Pb	Mod1	Mod2	Mod3
39	35	0.0	35.0	-0.986176	-0.700649	0.257960
40	35	0.0	35.0	-0.980322	-0.693290	0.437074
41	35	0.0	35.0	-0.986616	-0.698830	0.747269
	c_total	Cd	Pb	Mod1	Mod2	Mod3
42	35	7.0	28.0	-0.960407	-0.711397	0.452190
43	35	7.0	28.0	-0.963609	-0.711479	0.763115
44	35	7.0	28.0	-0.943684	-0.710487	0.891407

	c_total	Cd	Pb	Mod1	Mod2	Mod3
45	35	14.0	21.0	-0.892616	-0.701641	0.685861
46	35	14.0	21.0	-0.852896	-0.701806	1.065760
47	35	14.0	21.0	-0.913592	-0.706932	1.443088
	c_total	Cd	Pb	Mod1	Mod2	Mod3
48	35	21.0	14.0	-0.808073	-0.686593	1.036557
49	35	21.0	14.0	-0.793822	-0.672786	0.911451
50	35	21.0	14.0	-0.786557	-0.694034	1.271240
	c_total	Cd	Pb	Mod1	Mod2	Mod3
51	35	28.0	7.0	-0.441039	-0.688826	0.580615
52	35	28.0	7.0	-0.436665	-0.679814	1.104487
53	35	28.0	7.0	-0.346168	-0.695523	1.169927
	c_total	Cd	Pb	Mod1	Mod2	Mod3
54	35	35.0	0.0	-0.040629	-0.643767	0.584863
55	35	35.0	0.0	-0.050647	-0.616401	1.168749
56	35	35.0	0.0	-0.053569	-0.558031	1.565606
	c_total	Cd	Pb	Mod1	Mod2	Mod3
57	50	0.0	50.0	-0.959706	-0.668074	-0.077489
58	50	0.0	50.0	-0.933817	-0.698003	-0.089966
59	50	0.0	50.0	-0.956934	-0.697838	0.377690
60	50	0.0	50.0	-0.926602	-0.699822	0.351225
	c_total	Cd	Pb	Mod1	Mod2	Mod3
61	50	10.0	40.0	-0.801748	0.031537	-0.169908
62	50	10.0	40.0	-0.806512	-0.500488	0.513615
63	50	10.0	40.0	-0.769134	-0.663279	0.683405
64	50	10.0	40.0	-0.765851	-0.678243	0.394167
	c_total	Cd	Pb	Mod1	Mod2	Mod3
65	50	20.0	30.0	-0.725711	-0.480976	0.174434
66	50	20.0	30.0	-0.740663	-0.418142	0.808528
67	50	20.0	30.0	-0.835654	-0.315540	0.872177
68	50	20.0	30.0	-0.740102	-0.637235	1.415295
	c_total	Cd	Pb	Mod1	Mod2	Mod3
69	50	30.0	20.0	-0.539212	-0.352993	1.028776
70	50	30.0	20.0	-0.654168	-0.480398	1.680060
71	50	30.0	20.0	-0.603170	-0.537114	1.873760
72	50	30.0	20.0	-0.490025	-0.451213	2.102618
	c_total	Cd	Pb	Mod1	Mod2	Mod3
73	50	40.0	10.0	0.544198	0.334879	0.584614
74	50	40.0	10.0	0.657312	-0.143324	0.694356
75	50	40.0	10.0	0.448056	-0.235839	0.229985
76	50	40.0	10.0	0.507370	0.268241	0.607146
	c_total	Cd	Pb	Mod1	Mod2	Mod3
77	50	50.0	0.0	1.140543	1.334855	0.810320
78	50	50.0	0.0	1.138011	1.206375	0.640331
79	50	50.0	0.0	1.065307	1.219024	0.471288
80	50	50.0	0.0	1.038457	1.185210	0.622328
	c_total	Cd	Pb	Mod1	Mod2	Mod3
81	70	0.0	70.0	-0.936589	-0.668570	-0.047141
82	70	0.0	70.0	-0.938701	-0.706271	-0.467310
83	70	0.0	70.0	-0.922519	-0.705692	0.222319
84	70	0.0	70.0	-0.916894	-0.695688	0.310408
	c_total	Cd	Pb	Mod1	Mod2	Mod3
85	70	14.0	56.0	-0.680207	-0.533889	-0.148803
86	70	14.0	56.0	-0.654188	-0.612515	0.753558
87	70	14.0	56.0	-0.667498	-0.658153	0.830115
88	70	14.0	56.0	-0.682449	-0.652531	0.585211
	c_total	Cd	Pb	Mod1	Mod2	Mod3
89	70	28.0	42.0	-0.122361	-0.386063	0.606815
90	70	28.0	42.0	-0.088475	-0.406815	0.449900
91	70	28.0	42.0	-0.224477	-0.513220	0.290298

```
92       70  28.0  42.0 -0.265558 -0.531574  0.383813
    c_total    Cd    Pb      Mod1      Mod2      Mod3
93       70  42.0  28.0  0.446935  0.350753  0.848118
94       70  42.0  28.0  0.498353  0.474354  0.799303
95       70  42.0  28.0  0.636417  0.911384  0.607959
96       70  42.0  28.0  0.588631  0.502382  1.006011
     c_total    Cd    Pb      Mod1      Mod2      Mod3
97        70  56.0  14.0  1.462474  2.692325  1.053050
98        70  56.0  14.0  1.557985  2.081344  1.226540
99        70  56.0  14.0  1.495749  2.377657  1.457141
100       70  56.0  14.0  1.380433  2.175513  1.293441
     c_total    Cd   Pb      Mod1      Mod2      Mod3
101       70  70.0  0.0  1.910999  2.569633  1.096406
102       70  70.0  0.0  1.860521  2.545905  1.236778
103       70  70.0  0.0  1.812015  2.622133  1.111223
104       70  70.0  0.0  1.702443  2.159060 -0.017043
     c_total   Cd     Pb      Mod1      Mod2      Mod3
105      100  0.0  100.0 -0.898020 -0.682708 -0.763717
106      100  0.0  100.0 -0.874563 -0.622932 -0.695339
107      100  0.0  100.0 -0.718066 -0.623098 -0.577666
108      100  0.0  100.0 -0.864165 -0.668074 -0.534808
     c_total    Cd    Pb      Mod1      Mod2      Mod3
109      100  20.0  80.0 -0.478667 -0.160356 -0.313698
110      100  20.0  80.0 -0.499873 -0.155395 -0.245221
111      100  20.0  80.0 -0.520568 -0.276268 -0.058789
112      100  20.0  80.0 -0.360188 -0.296359 -0.112548
     c_total    Cd    Pb      Mod1      Mod2      Mod3
113      100  40.0  60.0  0.734310  0.622677 -0.254115
114      100  40.0  60.0  0.609246  0.782739 -0.391848
115      100  40.0  60.0  0.649006  0.635822 -0.173791
116      100  40.0  60.0  0.604893  0.666991 -0.240227
     c_total    Cd    Pb      Mod1      Mod2      Mod3
117      100  60.0  40.0  1.443910  2.201969  0.640962
118      100  60.0  40.0  1.442039  2.094820  0.606748
119      100  60.0  40.0  1.489454  2.091926  0.408901
120      100  60.0  40.0  1.546917  2.208831  0.587352
     c_total    Cd    Pb      Mod1      Mod2      Mod3
121      100  80.0  20.0  1.717625  2.391878  0.709505
122      100  80.0  20.0  1.763589  2.711341  0.883542
123      100  80.0  20.0  1.759446  2.786908  0.924808
124      100  80.0  20.0  1.595063  1.988994  0.902972
     c_total     Cd   Pb      Mod1      Mod2      Mod3
125      100  100.0  0.0  1.791840  2.472322  0.146575
126      100  100.0  0.0  1.709038  2.525153  0.819496
127      100  100.0  0.0  1.825685  2.547641  0.924642
128      100  100.0  0.0  1.651645  2.122186  0.792002
     c_total   Cd     Pb      Mod1      Mod2      Mod3
129      200  0.0  200.0 -0.773097 -0.058746 -0.672691
130      200  0.0  200.0 -0.774848 -0.091320 -0.748750
131      200  0.0  200.0 -0.744696 -0.062797 -0.764331
132      200  0.0  200.0 -0.785916 -0.121994 -0.705029
     c_total    Cd     Pb      Mod1      Mod2      Mod3
133      200  40.0  160.0  0.536802  0.648885 -0.630613
134      200  40.0  160.0  0.522091  0.528094 -0.576770
135      200  40.0  160.0  0.618913  0.536775 -0.480584
136      200  40.0  160.0  0.453840  0.408130 -0.461619
     c_total    Cd     Pb      Mod1      Mod2      Mod3
137      200  80.0  120.0  1.671380  0.863349 -0.192557
138      200  80.0  120.0  1.652686  0.885093 -0.262975
139      200  80.0  120.0  1.711690  0.874924 -0.245039
```

```
140      200  80.0  120.0  1.664585  0.887242 -0.124644
     c_total     Cd    Pb      Mod1      Mod2      Mod3
141      200  120.0  80.0  1.661032  1.419598 -0.163022
142      200  120.0  80.0  1.630079  1.374622 -0.192142
143      200  120.0  80.0  1.798275  1.138580 -0.043574
144      200  120.0  80.0  1.748598  1.365114  0.008476
     c_total     Cd    Pb      Mod1      Mod2      Mod3
145      200  160.0  40.0  1.949548  2.072497  1.017310
146      200  160.0  40.0  1.903444  2.123509  1.117694
147      200  160.0  40.0  1.824454  2.014789  1.288795
148      200  160.0  40.0  1.798865  1.861506  1.198814
     c_total     Cd   Pb      Mod1      Mod2      Mod3
149      200  200.0  0.0  1.946165  2.139714  1.356741
150      200  200.0  0.0  1.878525  2.218091  1.671498
151      200  200.0  0.0  1.931074  2.258933  1.658556
152      200  200.0  0.0  1.896498  2.208583  1.632805
     c_total   Cd     Pb      Mod1      Mod2      Mod3
153      500  0.0  500.0 -0.595294 -0.710404 -1.501331
154      500  0.0  500.0 -0.572597 -0.610779 -1.510888
155      500  0.0  500.0 -0.554193 -0.615987 -1.521375
     c_total     Cd     Pb      Mod1      Mod2      Mod3
156      500  100.0  400.0  1.186258 -0.470394 -1.429752
157      500  100.0  400.0  1.122030 -0.446665 -1.424027
158      500  100.0  400.0  1.140623 -0.498256 -1.407850
     c_total     Cd     Pb      Mod1      Mod2      Mod3
159      500  200.0  300.0  1.312442 -0.346461 -1.211728
160      500  200.0  300.0  1.109960 -0.280981 -1.131271
161      500  200.0  300.0  1.272472 -0.294292 -1.189378
     c_total     Cd     Pb      Mod1      Mod2      Mod3
162      500  300.0  200.0  1.091106 -0.349685 -1.187271
163      500  300.0  200.0  1.037676 -0.373000 -1.219526
164      500  300.0  200.0  1.102745 -0.383666 -1.195716
     c_total     Cd     Pb      Mod1      Mod2      Mod3
165      500  400.0  100.0  1.043971  0.057581 -1.040544
166      500  400.0  100.0  1.043971  0.164812 -0.823250
167      500  400.0  100.0  1.026108  0.202430 -0.763966
     c_total     Cd   Pb      Mod1      Mod2      Mod3
168      500  500.0  0.0  1.023036  0.534708  0.410825
169      500  500.0  0.0  0.919688  0.468484  0.456720
170      500  500.0  0.0  0.990771  0.660212  0.999407
     c_total   Cd      Pb      Mod1      Mod2      Mod3
171     1000  0.0  1000.0 -0.580283 -0.521405 -1.497747
172     1000  0.0  1000.0 -0.591091 -0.554476 -1.498710
173     1000  0.0  1000.0 -0.583625 -0.560594 -1.498494
     c_total     Cd     Pb      Mod1      Mod2      Mod3
174     1000  200.0  800.0  0.681150 -0.507515 -1.489833
175     1000  200.0  800.0  0.681150 -0.500157 -1.486016
176     1000  200.0  800.0  0.731328 -0.500157 -1.485751
     c_total     Cd     Pb      Mod1      Mod2      Mod3
177     1000  400.0  600.0  0.640710 -0.452701 -1.393331
178     1000  400.0  600.0  0.610617 -0.457248 -1.372060
179     1000  400.0  600.0  0.643912 -0.470394 -1.373902
     c_total     Cd     Pb      Mod1      Mod2      Mod3
180     1000  600.0  400.0  0.597287 -0.550673 -1.407286
181     1000  600.0  400.0  0.439479 -0.537114 -1.411633
182     1000  600.0  400.0  0.537343 -0.549267 -1.393398
     c_total     Cd     Pb      Mod1      Mod2      Mod3
183     1000  800.0  200.0  0.512364 -0.395571 -0.998267
184     1000  800.0  200.0  0.512364 -0.356630 -1.020086
185     1000  800.0  200.0  0.540065 -0.410122 -0.997188
```

```
     c_total      Cd   Pb      Mod1      Mod2      Mod3
186     1000  1000.0  0.0  0.464578  0.213261  0.549090
187     1000  1000.0  0.0  0.365734  0.380268  0.492941
188     1000  1000.0  0.0  0.445014  0.251292  0.656376
     c_total   Cd      Pb      Mod1      Mod2      Mod3
189     2000  0.0  2000.0 -0.667348 -0.437902 -1.525822
190     2000  0.0  2000.0 -0.645171 -0.495941 -1.530484
191     2000  0.0  2000.0 -0.643500 -0.429055 -1.531380
     c_total     Cd      Pb      Mod1      Mod2      Mod3
192     2000  400.0  1600.0  0.307010 -0.454768 -1.528161
193     2000  400.0  1600.0  0.307010 -0.451130 -1.537785
194     2000  400.0  1600.0  0.294221 -0.479323 -1.531679
     c_total     Cd      Pb      Mod1      Mod2      Mod3
195     2000  800.0  1200.0  0.187001 -0.537610 -1.494512
196     2000  800.0  1200.0  0.166505 -0.508756 -1.499041
197     2000  800.0  1200.0  0.174902 -0.521240 -1.492006
     c_total      Cd     Pb      Mod1      Mod2      Mod3
198     2000  1200.0  800.0 -0.248275 -0.598873 -1.522453
199     2000  1200.0  800.0 -0.255340 -0.606562 -1.515302
200     2000  1200.0  800.0 -0.210667 -0.600031 -1.501016
     c_total      Cd     Pb      Mod1      Mod2      Mod3
201     2000  1600.0  400.0 -0.346278 -0.505283 -1.284817
202     2000  1600.0  400.0 -0.346278 -0.505945 -1.300480
203     2000  1600.0  400.0 -0.358967 -0.498669 -1.280304
     c_total      Cd   Pb      Mod1      Mod2      Mod3
204     2000  2000.0  0.0 -0.443671 -0.207068  0.271134
205     2000  2000.0  0.0 -0.445252 -0.190285  0.217010
206     2000  2000.0  0.0 -0.462885 -0.206159  0.432495
     c_total   Cd      Pb      Mod1      Mod2      Mod3
207     5000  0.0  5000.0 -0.926211 -0.267835 -1.514091
208     5000  0.0  5000.0 -0.917855 -0.352744 -1.512365
209     5000  0.0  5000.0 -0.906467 -0.240800 -1.526402
     c_total      Cd      Pb      Mod1      Mod2      Mod3
210     5000  1000.0  4000.0 -0.810195 -0.252954 -1.519649
211     5000  1000.0  4000.0 -0.810195 -0.262957 -1.525540
212     5000  1000.0  4000.0 -0.798276 -0.352827 -1.522536
     c_total      Cd      Pb      Mod1      Mod2      Mod3
213     5000  2000.0  3000.0 -0.778261 -0.517519 -1.544986
214     5000  2000.0  3000.0 -0.773027 -0.518181 -1.547674
215     5000  2000.0  3000.0 -0.767202 -0.515452 -1.542149
     c_total      Cd      Pb      Mod1      Mod2      Mod3
216     5000  3000.0  2000.0 -0.893437 -0.634590 -1.579398
217     5000  3000.0  2000.0 -0.873292 -0.648810 -1.579050
218     5000  3000.0  2000.0 -0.866196 -0.644015 -1.579183
     c_total      Cd      Pb      Mod1      Mod2      Mod3
219     5000  4000.0  1000.0 -0.874613 -0.677499 -1.491442
220     5000  4000.0  1000.0 -0.874613 -0.584157 -1.515186
221     5000  4000.0  1000.0 -0.875954 -0.604495 -1.516845
     c_total      Cd   Pb      Mod1      Mod2      Mod3
222     5000  5000.0  0.0 -0.872241 -0.349768  0.499810
223     5000  5000.0  0.0 -0.872021 -0.354729  0.777301
224     5000  5000.0  0.0 -0.865256 -0.373414  0.712840
```

In [ ]:
```
print(f'The cindex for c_total is {cindex(c_total_true, c_total_pred)}\n'
print(f'The cindex for Cd is {cindex(Cd_true, Cd_pred)}\n')
print(f'The cindex for Pb is {cindex(Pb_true, Pb_pred)}\n')
```

```
The cindex for c_total is 0.833822091886608

The cindex for Cd is 0.7631826741996234

The cindex for Pb is 0.7688323917137476
```

Interpretation of results

Answer the following questions based on the results obtained

- Which cross-validation approach had more optimistic results?
- Which cross-validation generalize better on unseen data? Why?

The Leave One Out Cross Validation is more optimistic. Data from each group is dependent and one instance is used for testing and the remaining instances of the group are used for training leading to leakage, in turn leading the model to perform good.

The Leave Replica Out approach of cross validation has lower accuracy compared to the Leave One Out Cross Validation. But this lower accuracy is traded off with better generalization. Since the whole group is used for testing and removed in the training phases resulting in no leakage of replicas into the model.