Greedy MaxMin Pruning with top-1 and one by one view execution

Configurations

1. Dataset = one example subset of Heart disease dataset
2. 4 views have importance score = √2 because they don't have its bar pair
3. Total views = 180 – 4 = 176 views
4. Diversity function = Weighted Jaccard A:M:F (3:2:1)
5. This experiment using maximum bound = actual maximum importance score
6. Using constant of λ value = 0.5

Running steps of Greedy MaxMin

Step 1: Get two most distant views
S = 2 views
X = 174 views
Actual maximum importance score i_{actual} = 0.86060086581876793

Step 2: Execute query of S
Importance score s1 =0.34320761722812221, s2 =0.65856257658897521=
Max of S = 0.65856257658897521, this will be used as the minimum bound i_{min}

Step 3: Calculate the diversity score of all views in X to S and then sort it as descending

100 views have same diversity score to S, please see the figure below. This is the different of Greedy to Swap. Greedy only has two views as the initialization in the first iteration, hence, there will be so many views that have same diversity score.

Step 4: Calculate Umax and Umin using actual importance score as maximum bound and i_{min} as the minimum bound.
$Umax = \big((1-\lambda) * i_{actual}\big) + (\lambda * div)$
$Umin = \big((1-\lambda) * i_{min}\big) + (\lambda * div)$

	Attributes	Meassure	Function	div	Umax	Umin
0	restecg	oldpeak	avg	0.5	0.6803	0.579281
1	fbs	restbp	avg	0.5	0.6803	0.579281
2	thal	oldpeak	stddev	0.5	0.6803	0.579281
3	thal	oldpeak	avg	0.5	0.6803	0.579281
4	thal	oldpeak	max	0.5	0.6803	0.579281
5	fbs	restbp	variance	0.5	0.6803	0.579281
6	fbs	restbp	stddev	0.5	0.6803	0.579281
7	fbs	restbp	max	0.5	0.6803	0.579281
8	num	restbp	variance	0.5	0.6803	0.579281
9	slope	oldpeak	max	0.5	0.6803	0.579281
10	num	restbp	stddev	0.5	0.6803	0.579281
11	num	restbp	avg	0.5	0.6803	0.579281
12	num	restbp	max	0.5	0.6803	0.579281
13	restecg	restbp	variance	0.5	0.6803	0.579281
14	restecg	restbp	stddev	0.5	0.6803	0.579281
15	restecg	restbp	avg	0.5	0.6803	0.579281
16	thal	oldpeak	sum	0.5	0.6803	0.579281
17	slope	oldpeak	avg	0.5	0.6803	0.579281
18	sex	restbp	variance	0.5	0.6803	0.579281
19	restecg	oldpeak	variance	0.5	0.6803	0.579281
20	fbs	oldpeak	stddev	0.5	0.6803	0.579281
21	fbs	oldpeak	max	0.5	0.6803	0.579281
22	num	oldpeak	max	0.5	0.6803	0.579281
23	num	oldpeak	variance	0.5	0.6803	0.579281
24	num	oldpeak	avg	0.5	0.6803	0.579281
25	num	oldpeak	stddev	0.5	0.6803	0.579281
26	restecg	oldpeak	stddev	0.5	0.6803	0.579281
27	slope	oldpeak	stddev	0.5	0.6803	0.579281
28	restecg	oldpeak	max	0.5	0.6803	0.579281
29	sex	oldpeak	variance	0.5	0.6803	0.579281
...	...	...	...	...	...	...
70	fbs	oldpeak	variance	0.5	0.6803	0.579281
71	thal	thalach	avg	0.5	0.6803	0.579281
72	num	age	avg	0.5	0.6803	0.579281
73	sex	age	max	0.5	0.6803	0.579281
74	fbs	age	stddev	0.5	0.6803	0.579281
75	fbs	age	max	0.5	0.6803	0.579281
76	fbs	age	avg	0.5	0.6803	0.579281
77	num	age	variance	0.5	0.6803	0.579281
78	num	age	stddev	0.5	0.6803	0.579281
79	thal	age	max	0.5	0.6803	0.579281
80	thal	age	stddev	0.5	0.6803	0.579281
81	thal	age	avg	0.5	0.6803	0.579281
82	thal	age	sum	0.5	0.6803	0.579281
83	slope	age	avg	0.5	0.6803	0.579281
84	slope	age	max	0.5	0.6803	0.579281
85	slope	age	stddev	0.5	0.6803	0.579281
86	slope	age	variance	0.5	0.6803	0.579281
87	sex	age	avg	0.5	0.6803	0.579281
88	num	age	max	0.5	0.6803	0.579281
89	restecg	age	avg	0.5	0.6803	0.579281
90	restecg	age	variance	0.5	0.6803	0.579281
91	restecg	age	stddev	0.5	0.6803	0.579281
92	restecg	age	max	0.5	0.6803	0.579281
93	sex	age	stddev	0.5	0.6803	0.579281
94	thal	chol	sum	0.5	0.6803	0.579281
95	sex	age	variance	0.5	0.6803	0.579281
96	thal	chol	max	0.5	0.6803	0.579281
97	thal	chol	stddev	0.5	0.6803	0.579281
98	thal	chol	avg	0.5	0.6803	0.579281
99	fbs	age	variance	0.5	0.6803	0.579281

Step 5: Get the maximum of Umin
Maximum of Umin = 0.57928128829448755
Minimum of Umax = 0.59696709957605065

Step 6: There is no pruning at all.

Step 7: Execute the query of first view and compare the importance score to i_{min}, if the new importance score > i_{min}, the i_{min} will be changed to the new one.

Unfortunately the importance score of the first view = 0.1854775102197988

Step 8: Execute the query again till found the higher importance score

Unfortunately after 100 times view execution there are no importance score of view that larger than i_{min}. In 112 times of view execution, there is a view has importance score = 0.73240885499, then i_{min} is updated and Umin and Umax are recalculated.

Using i_{min}= 0.73240885499, **5 views can be pruned**.

Maximum Umin = 0.61620442749500004

Minimum Umax = 0.59696709957605065

Step 8: Execute the query again

Finally at the number 134, there is a view which has importance score = 0.86060086581876793 which is the actual maximum importance score.

Update the value of i_{min} and recalculate the Umin and Umax

Due to $i_{min} = i_{actual}$ then all remaining views are pruned.

Total pruned queries = 35 views.

The view that has the maximum real utility (calculated by the real diversity score and actual importance score) will be added to S.

Step 9: Now the view in S = 3 views, need to recalculate the diversity score of all views in X to S. Recalculate Umax and Umin again. However mostly views have been executed and actual importance score has been known. The chance to get more pruned queries seems hard.