**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 = 0.86060086581876793

**Step 2:** Execute query of S

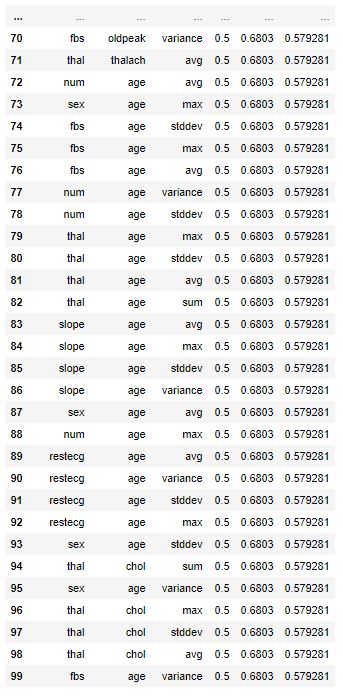
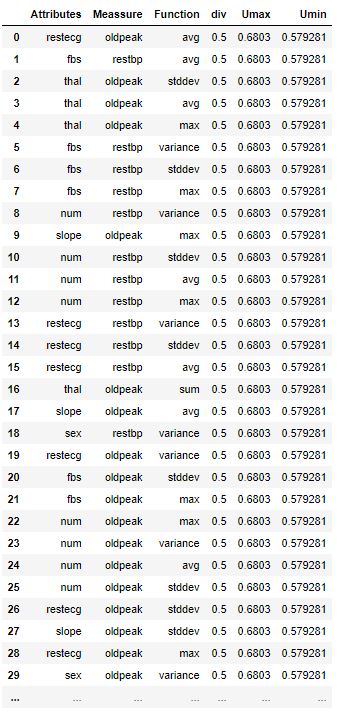
Importance score s1 =0.34320761722812221, s2 =0.65856257658897521=

Max of S = 0.65856257658897521, this will be used as the minimum bound

**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 as the minimum bound.



**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 , if the new importance score > , the 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 In 112 times of view execution, there is a view has importance score = 0.73240885499, then is updated and Umin and Umax are recalculated.

Using = 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 and recalculate the Umin and Umax

Due to 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.

There are couples of issues:

1. One of the problem in Greedy is the initialization. There are only two views in the set S as the first initialization. That’s why there are a lot of views in X have same diversity score while adding the third view. If mostly views have same diversity score, it means the chance of pruning is low.
2. The attached document is just one of the example. It may in some case the high importance score is founded in the top list and it can prune a lot of views in advanced.
3. In the attached document, I used hypothetical as the maximum bound. In here, I only change the minimum bound and I am not sure when should I change the maximum bound while I used static bound as the first.