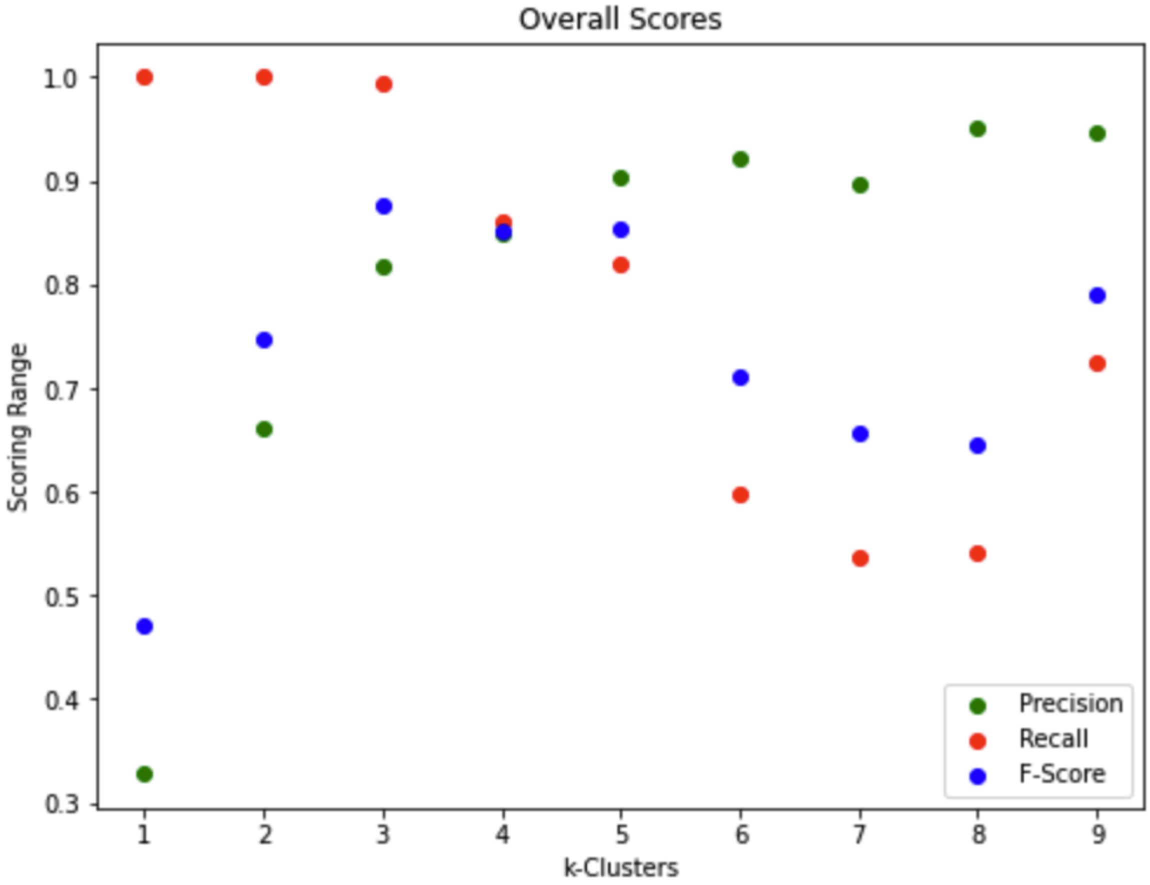
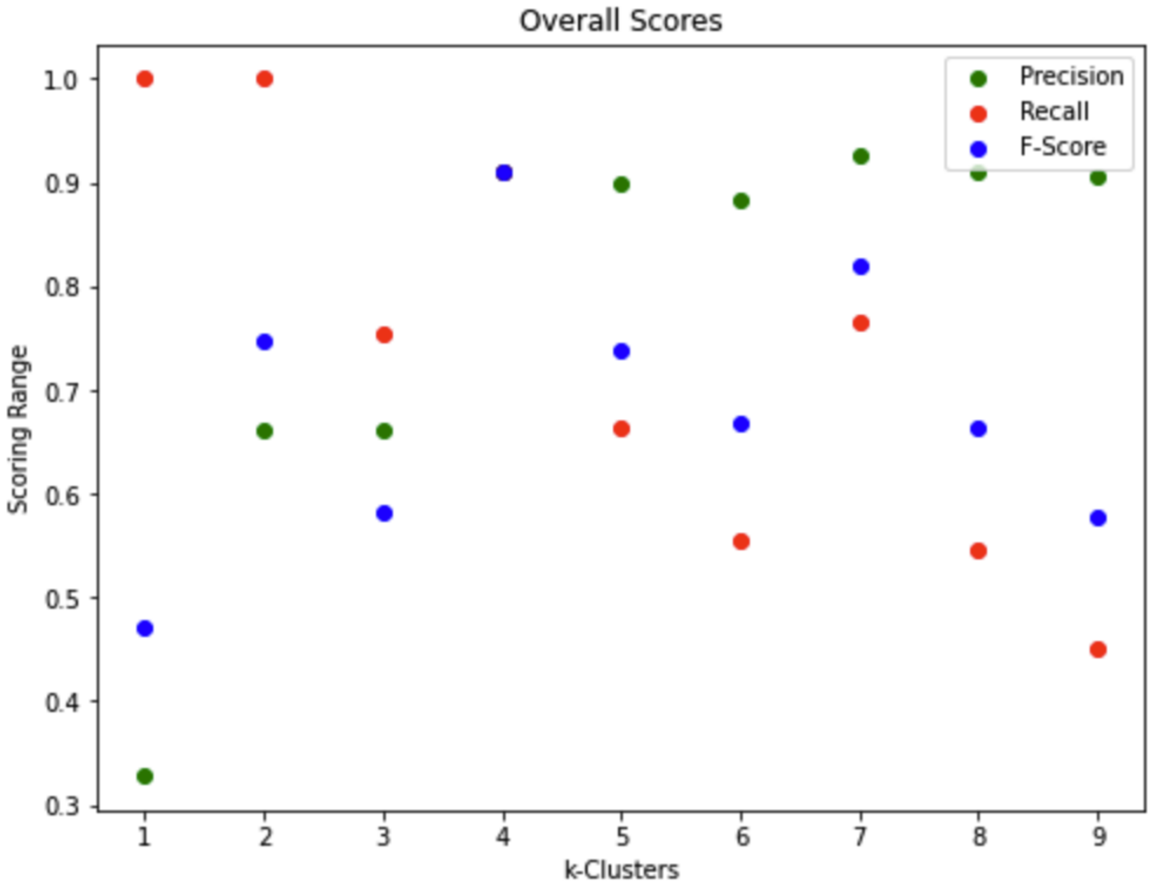
Michael Ross
COMP 527
CA Assignment 2 – Implementing the k-means and k-medians clustering algorithm
1 April 2022

1. See accompanying .py file
2. See accompanying .py file
3. Kmeans w/o normalization:



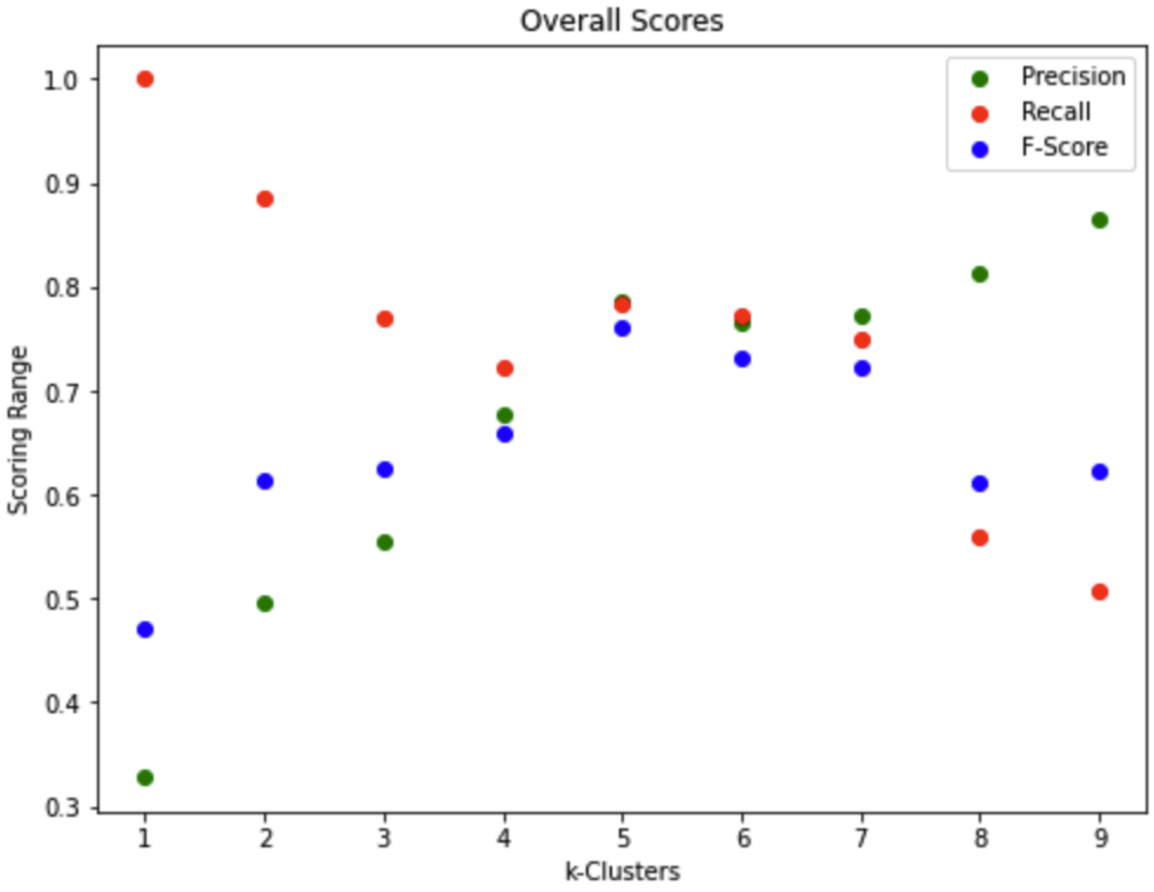
k	Precision Average	Recall Average	F-Score Average
1	0.32871344537029257	1.0	0.4723145424389056
2	0.6623558453999435	1.0	0.746848858469652
3	0.8181760352059978	0.9939892439101549	0.8775968384431971
4	0.8486542976559149	0.8615417324103357	0.850916485028214
5	0.9040044335191596	0.8189391804752947	0.8537236608814964
6	0.9220549946631208	0.5973964262953689	0.7102900973598365
7	0.8960334592552621	0.5373000524639368	0.657722049501045
8	0.9506038459544913	0.5405449306801363	0.6447857988236054
9	0.9472267773690477	0.724433222445133	0.7903750386561594

4. Kmeans with l2 normalisation



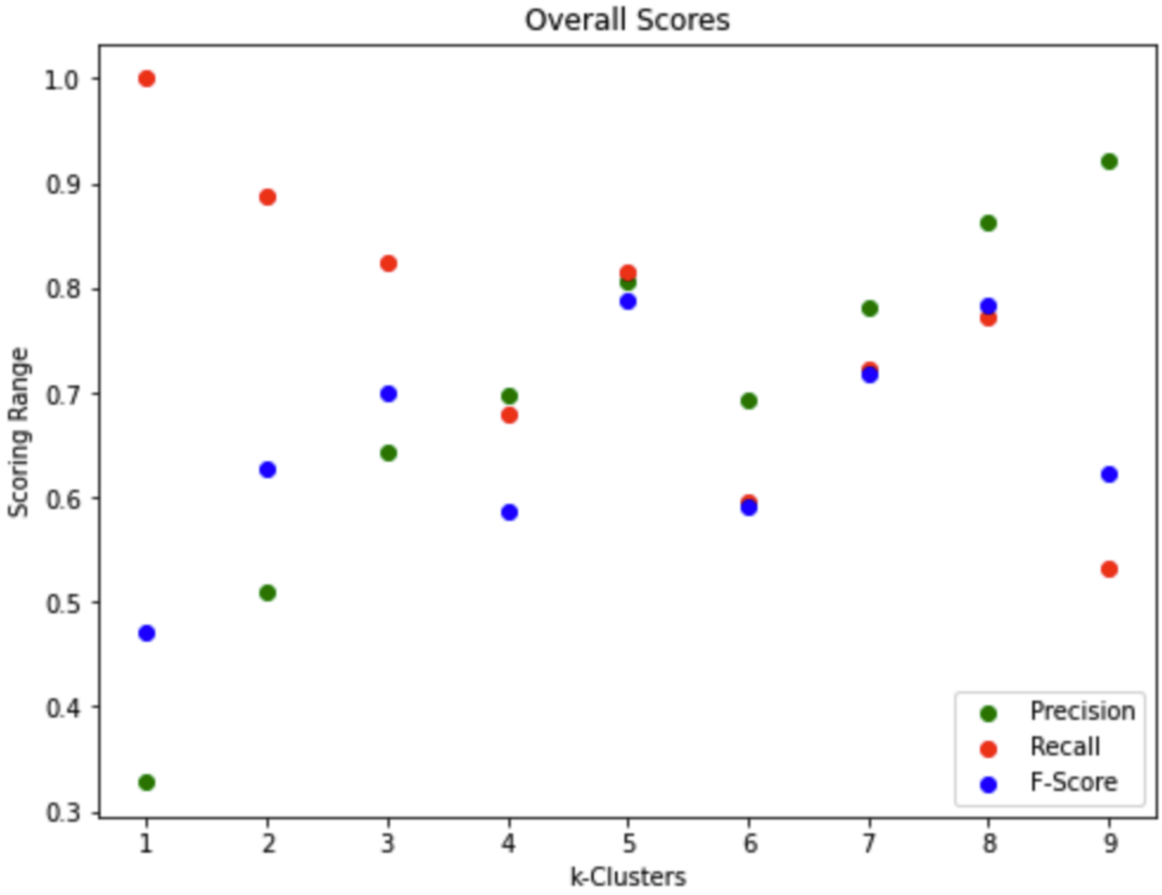
k	Precision Average	Recall Average	F-Score Average
1	0.32871344537029257	1.0	0.4723145424389054
2	0.6623558453999489	1.0	0.7468488584696534
3	0.6623558453999489	0.7542879936178699	0.5830064407686294
4	0.9113624179874183	0.910623325296725	0.9096606293354556
5	0.8996359400029114	0.6628291358907723	0.737454191065606 6
6	0.8840193747650552	0.554273348253788	0.6681690016144556
7	0.9261805847490484	0.7656565164337872	0.819310629680877
8	0.911023097168557	0.5453255285856424	0.6627788132240527
9	0.9070848875894748	0.4517703402558239	0.5778716466513745

5. Kmedians w/o normalisation



k	Precision Average	Recall Average	F-Score Average
1	0.3287134453702927	1.0	0.47231454243890575
2	0.4952435340291433	0.88563123418631	0.6128248688058111
3	0.5555699079208255	0.7694484867657927	0.6248883773985829
4	0.6770774845005934	0.7219817614128752	0.6588791284709303
5	0.7866157475564456	0.7828999261836959	0.7602611068272443
6	0.7652850804226948	0.7720257564533591	0.73267138091884
7	0.7730567973886681	0.7488136665612148	0.7230545428487023
8	0.8125523045277144	0.5603872244742619	0.6125288843344705
9	0.8660148373940253	0.5073964459447863	0.621947089941102

6. Kmedians with l2 normalisation



k	Precision Average	Recall Average	F-Score Average
1	0.3287134453702927	1.0	0.4723145424389056
2	0.5090528789645361	0.8880269956764714	0.6278130269582922
3	0.6441520947491295	0.8234610348822934	0.6989536350051273
4	0.6981302464954946	0.6797682678975048	0.5862216653909242
5	0.806754772063378	0.8147603786573258	0.7889150301859386
6	0.692794204561897	0.59622744329015	0.5910082433943842
7	0.7808468697459497	0.7216703574818074	0.7173378911258228
8	0.8621324512039763	0.7714183537327027	0.7830142960042753
9	0.922765890784318	0.533326233344662	0.6237971152487992

7. Comparison of results

When first attempting to implement the kmeans and kmedians algorithms, I initially selected random k-points as my starting centroids. However, this ultimately resulted in results that varied dramatically. I then decided to use the kmeans++ method to initialize my centroids, which resulted in much more consistent results. As seen in the charts above, my results vary, but I realize that these scores should be taken into context to truly understand what these scores represent.

Precision scores should increase the higher the k-clusters created as this will create a higher chance of the same labels being clustered together. So in that sense, precision scores should give a better indication of our model performance. For my results, I obtained the highest precision score of approximately 95% with the kmeans model without normalizing each instance. However, it should be noted that this was obtained when k=8 instead of 9, in which I observed a slight decrease.

Recall scores should be preferred in real applications if the number of labels (classes) are known and we hope our model accurately groups all instances of the same label in the correct cluster. In this example, we have four labels (animals, countries, fruits, veggies) and a good model prioritizing recall performance should have the highest recall score at k=4. This was achieved in my kmeans model with l2 normalisation with a score of approximately 91%.

Although my highest scores were achieved with the kmeans model, I did notice that the kmedians model achieved a more consistent and linear precision and recall scores from range k = 1-9. Overall, my kmeans model gave me the best outputs. I predict that this is the result of using the Euclidean distance instead of the Manhattan distance and using the mean of the distances to adjust centroids. The mean takes into account all distances calculated as opposed to the median, which selects the central distance calculated which may not be reflective of all instances.