Dimensionality Reduction

Dimensionality Reduction

Low Variance Filter: Data columns with little changes in the data are unlikely to carry much information (*mathematical information*). Therefore attributes which fall below a certain threshold are removed. Requires normalization first.

--Variations of Numeric Attributes (no standardization/normalization beforehand)

Nscore	0.996215
Escore	0.994903
Oscore	0.992471
Ascore	0.994888
Cscore	0.995051
Imp	0.910945
SS	0.928720

High Correlation Filter: Data columns with very similar trends or distributions may tend to carry similar information and so only one of those attributes needs to be fed into the model. We only select an attribute from a group when the correlation coefficient exceeds a certain threshold. Requires normalization first as correlation is scale sensitive.

--Correlation between numeric attributes (no standardization/normalization beforehand)

	Nscore	Escore	Oscore	Ascore	Cscore	Imp	SS
Nscore	1.000000	-0.431051	0.010177	-0.216964	-0.391088	0.174399	0.079988
Escore	-0.431051	1.000000	0.245277	0.157336	0.308024	0.114151	0.210130
Oscore	0.010177	0.245277	1.000000	0.038516	-0.056811	0.277512	0.421709
Ascore	-0.216964	0.157336	0.038516	1.000000	0.247482	-0.229690	-0.208061
Cscore	-0.391088	0.308024	-0.056811	0.247482	1.000000	-0.335133	-0.229038
Imp	0.174399	0.114151	0.277512	-0.229690	-0.335133	1.000000	0.623120
SS	0.079988	0.210130	0.421709	-0.208061	-0.229038	0.623120	1.000000

- Only one pair to consider dropping one of the two → Imp & SS correlation = 0.623120

PCA: A few options on how we could proceed with PCA…

1) Use the resulting principal components as clustering indicators.

2) Perform PCA then right away cluster the reduced data.
3) Solely use it for visualization purposes.

- The issue with PCA is you lose all interpretability of your data as the principal components are linear combos of the previous feature space.

https://stats.stackexchange.com/questions/183236/what-is-the-relation-between-k-means-clustering-and-pca

http://ranger.uta.edu/~chqding/papers/KmeansPCA1.pdf

Random Forests / Ensemble Trees: Build a large set of 2000 shallow trees with each being trained on a small fraction of the total number of attributes. If an attribute is regularly split upon it is most likely a beneficial attribute to retain.

- Rylen's Results: From running my script a few times I'm getting that attributes we should certainly keep are:
    - Escore, Cscore, Oscore, Nscore & Ascore

  Other attributes which appear relatively often are:

    - Legalh, LSD & Coke

  And some slightly less common but still reoccuring attributes are:

    - Amph, Benz & Shrooms

- Emma's Results (only put demographic info and personality scores into RF to predict for each specific drug -- didn't use the other drugs data)

- Attributes in drugs top 5 most often: Oscore (19 times), Cscore (19), Nscore (15), Ascore (15), Escore (12), SS (11), Country_UK (2), Age_18-24 (1), Country_USA (1)
- Attributes in drugs top 10 most often: Oscore (19 times), Cscore (19), Nscore (19), Ascore (19), Escore (19), SS (19), Imp (19), Country_UK (16), Age_18-24 (15), Country_USA (14)

Chi-Squared Test:

Rylen: The chi-squared test only accepts positive values which represent non-negative features or class data. I dropped the score attributes as well as 'Imp' and 'SS'.

From there I randomly selected 5 attributes to be targets before using the SelectKBest and Chi2 functions in sklearn to get the best attributes…

```
[['Alc' 'Caff' 'Keta' 'Legalh' 'Meth']
 ['Amph' 'Benz' 'Coke' 'Crack' 'LSD']
 ['Amph' 'Coke' 'Heroin' 'Keta' 'Legalh']
 ['Amph' 'Benz' 'Coke' 'Ect' 'LSD']
 ['Amph' 'Coke' 'Ect' 'Keta' 'Legalh']]
```

Each row pertains to the five best attributes from that sample.

Manifold Projections (ISOMAP):

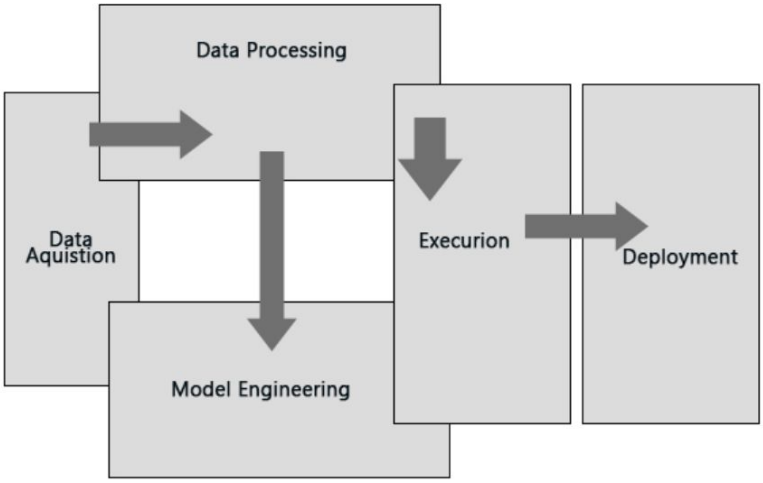
- Neighborhood Graph:
- Compute Graph Distances
- Embedding:

Not exactly sure how this method works…

References:

1) https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/
2) https://www.kdnuggets.com/2015/05/7-methods-data-dimensionality-reduction.html

Model Architecture's



Data Acquisition: Already done, using the drug consumption data from University of California Irvine https://archive.ics.uci.edu/ml/datasets/Drug+consumption+(quantified).

Data Processing:

- Data Cleaning: Done
- Encoding: Not needed
- Transformations: ???
- Normalization or Standardization: For our numerical data does it make sense to standardize or normalize? It may be already

Data Models:

1) DBSCAN
2) K-Modes and K-Prototypes Clustering
    - https://arxiv.org/ftp/cs/papers/0603/0603120.pdf (K-Modes)
    - http://www.cs.ust.hk/~qyang/Teaching/537/Papers/huang98extensions.pdf (K-Modes and K-Prototypes)
    - https://medium.com/datadriveninvestor/k-prototype-in-clustering-mixed-attributes-e6907db91914 (more k-prototypes and some python code to implement it)
3) Expectation Maximization (EM)
4) Hierarchical Clustering ← May be beneficial considering we have categorical data as well.
5) Mean Shift
6) Birch