Reduction of dimensionality for Clustering

Using K-means clustering

Problem Statement

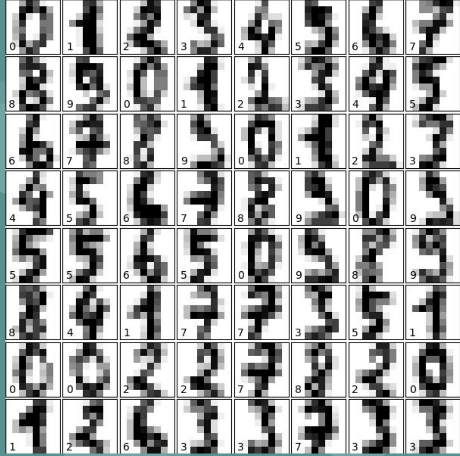
- Optimize reduction of dimensionality for K-means and K-median Clustering.

- Implement and evaluate the performance of PCA, SVD and Factor Analysis methods with each other as well as the results over the original dataset.

- Focus on the cost preservation as the target function using WCSS

Datasets Used

- Small real-world: Credit Card Information
- Library-provided: Digits(Sklearn)
- Large real-world: House Prices
  (required One-Hot Encoding)



Sklearn digits dataset

Factor Analysis

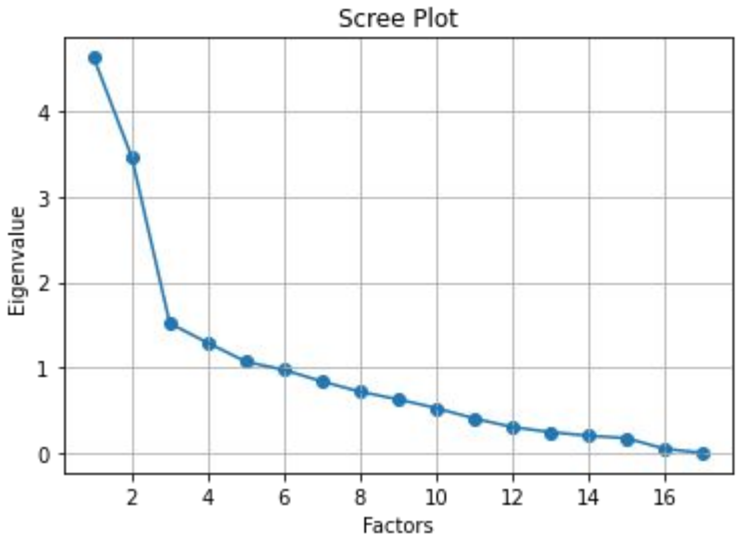
- An exploratory method that groups similar variables into dimensions
- Identifies correlated values in dataset
- Different rotation techniques to transform factor pattern.

Adequacy Test

- Calculated the eigenvalues for the columns of the dataset

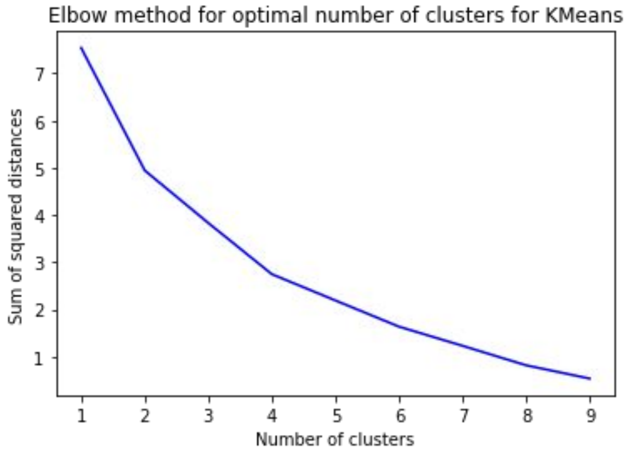
- Registered the columns with values greater than 1.

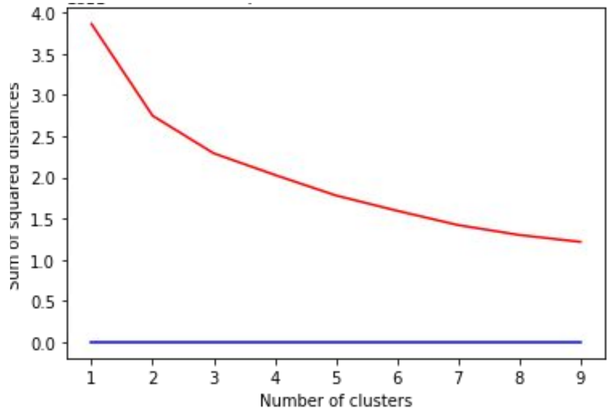
- kmo_model value = 0.645



K Means Clustering Implementation

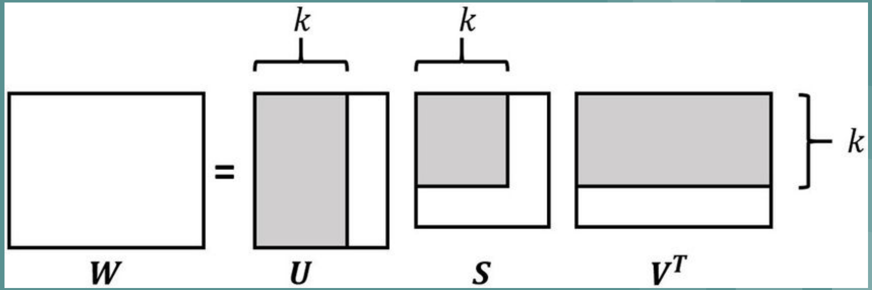
- Optimal clusters - 6(over the reduced dataset)
- WCSS reduced from 159517814576 to 1.637





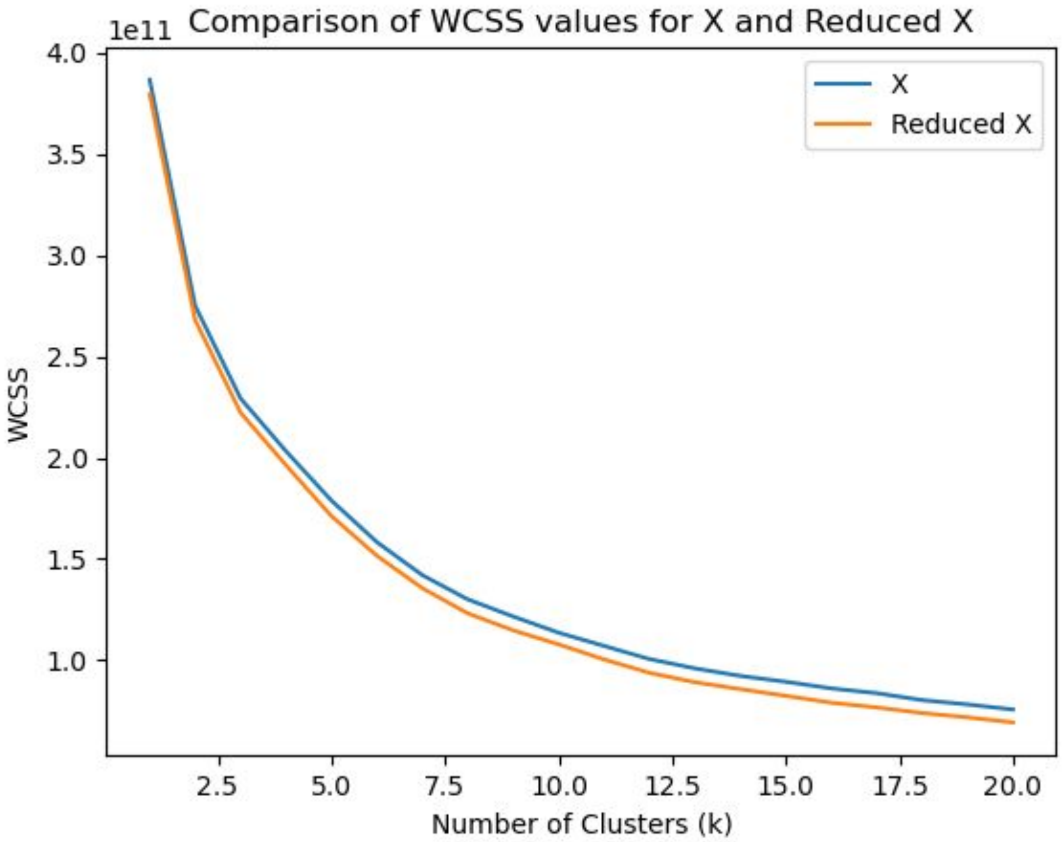
Truncated SVD

- Matrix factorization technique that decomposes a matrix into three parts: U, ∑, and V
- Approximate the original matrix using only a subset of its singular values and vectors
- Preserves pairwise distances



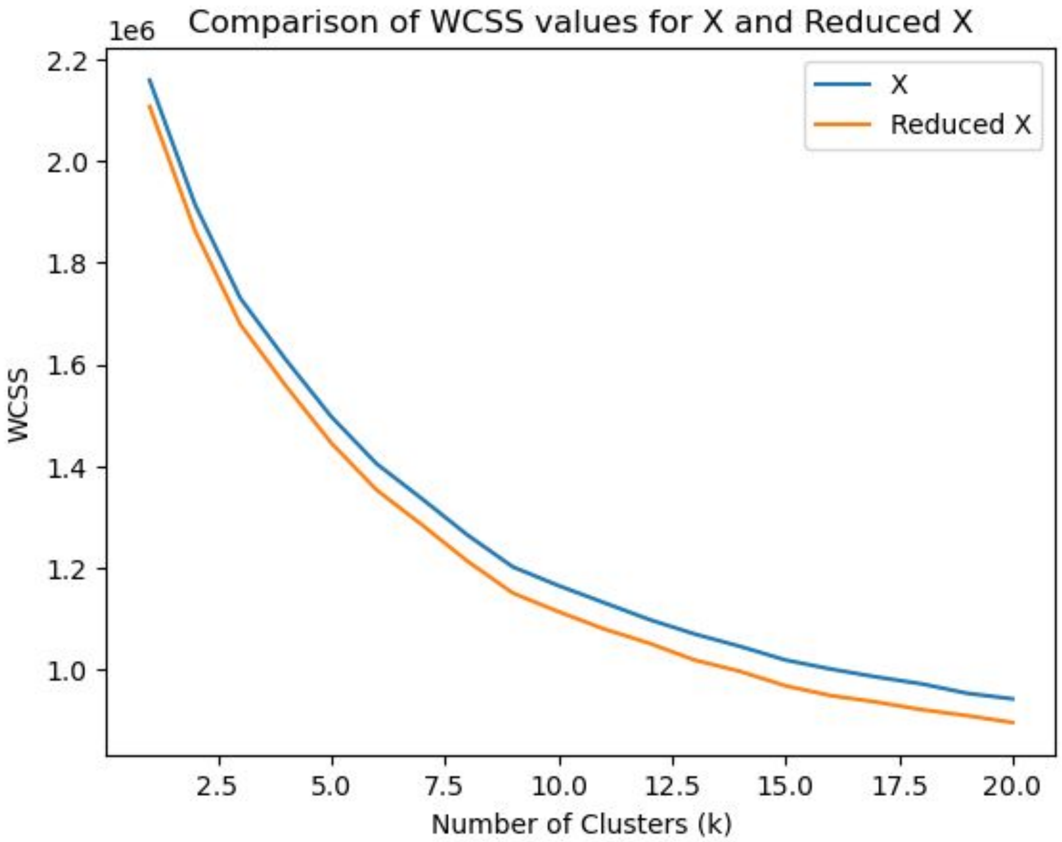
KMeans Clustering with Dataset of dimensions 17

- Reduced the dimensions to 6
- Loss is almost same



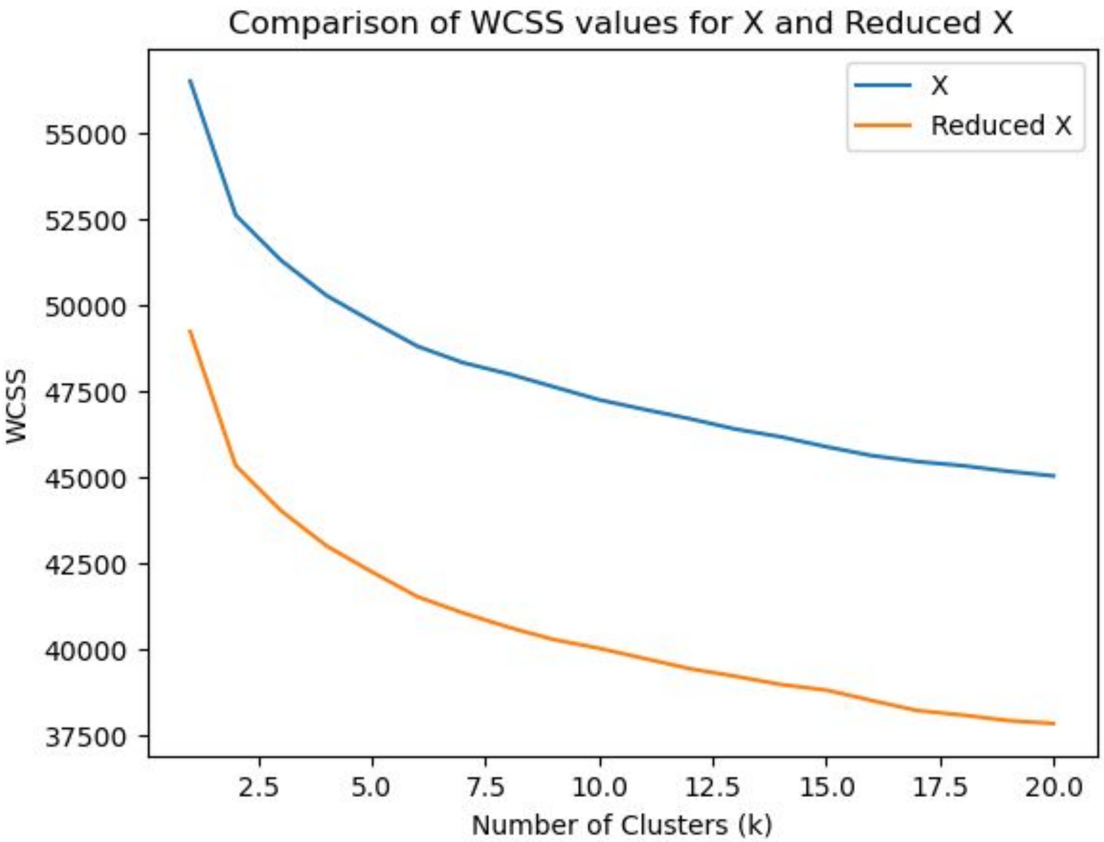
KMeans Clustering with Dataset of dimensions 64

- Reduced the dimensions to 35
- Loss is almost same



KMeans Clustering with Dataset of dimensions 9k+

- Reduced the dimensions to 400
- Loss is almost same

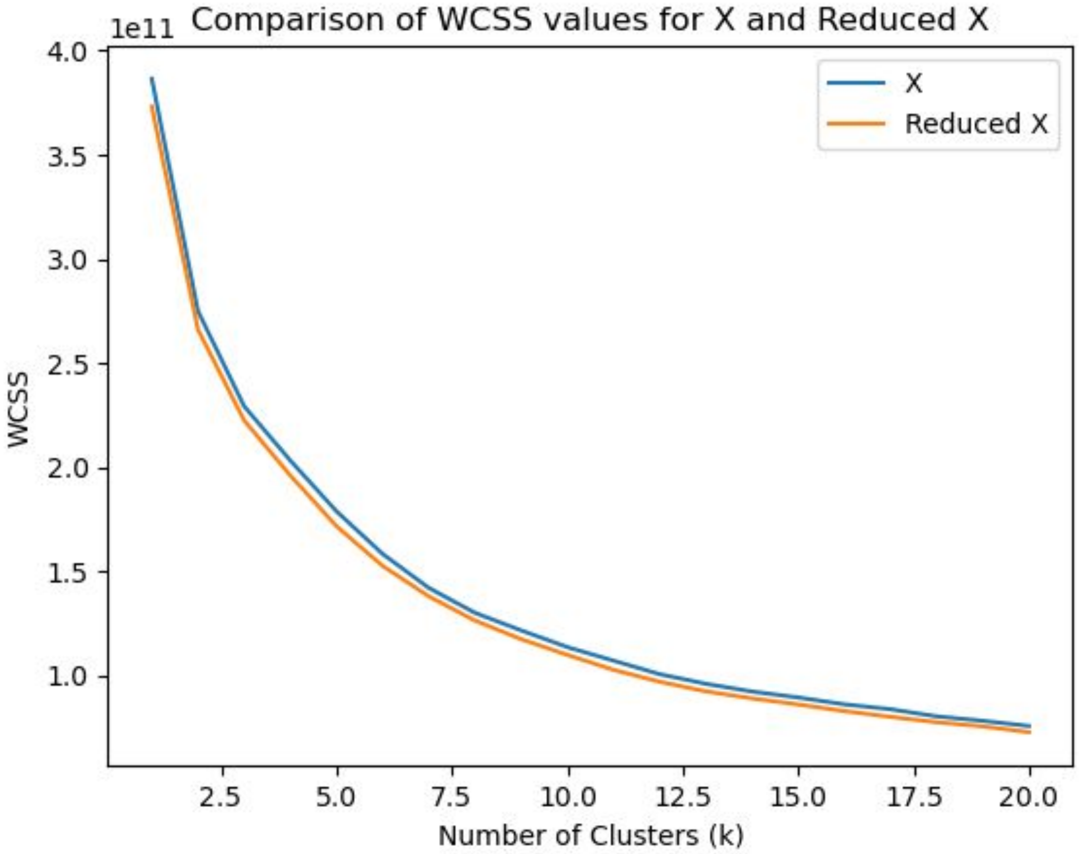


Johnson - Lindenstrauss Lemma

- Johnson-Lindenstrauss lemma maps high-dimensional data to lower dimensions while approximately preserving pairwise distances.
- It uses a random projection matrix to achieve this, with distortion in pairwise distances no more than $(1 \pm \epsilon)$ with **δ** probability.
- The lemma is useful for reducing the dimensionality of high-dimensional data and maintaining its structure in a lower-dimensional space.

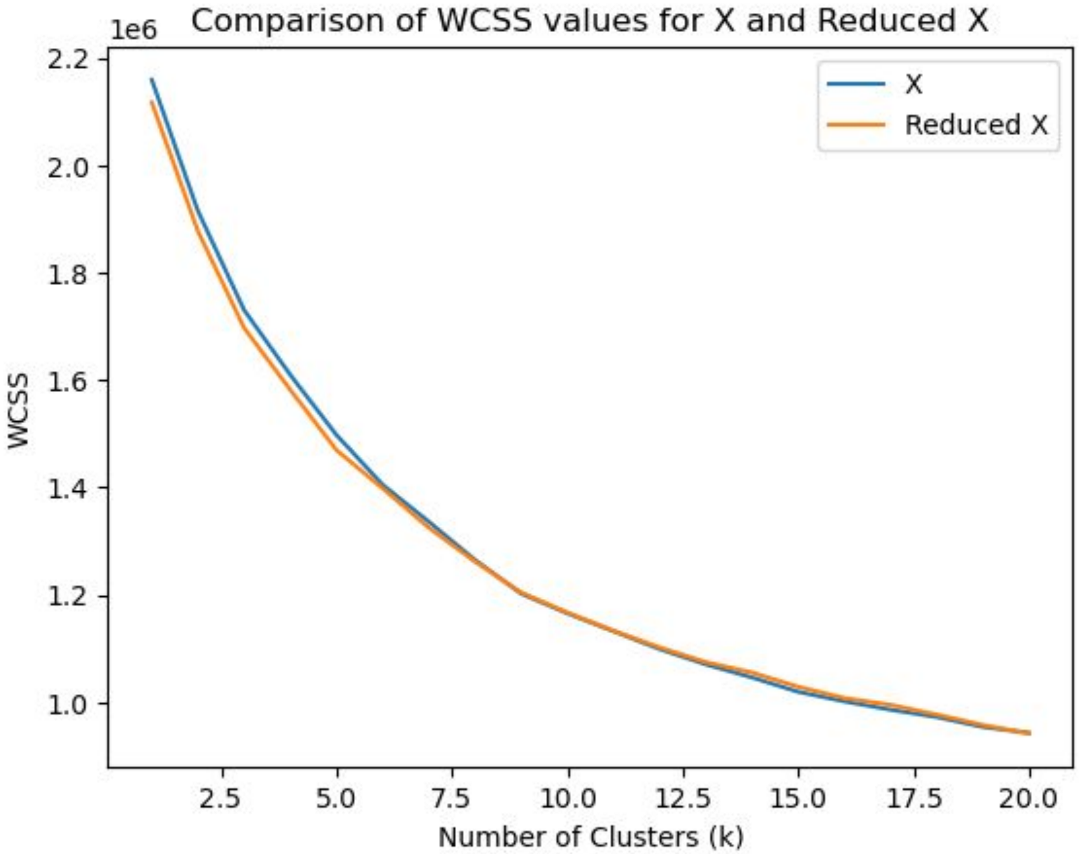
KMeans Clustering with Dataset of dimensions 17

- Loss is almost same
- **δ** = 0.9 and ϵ = 0.5



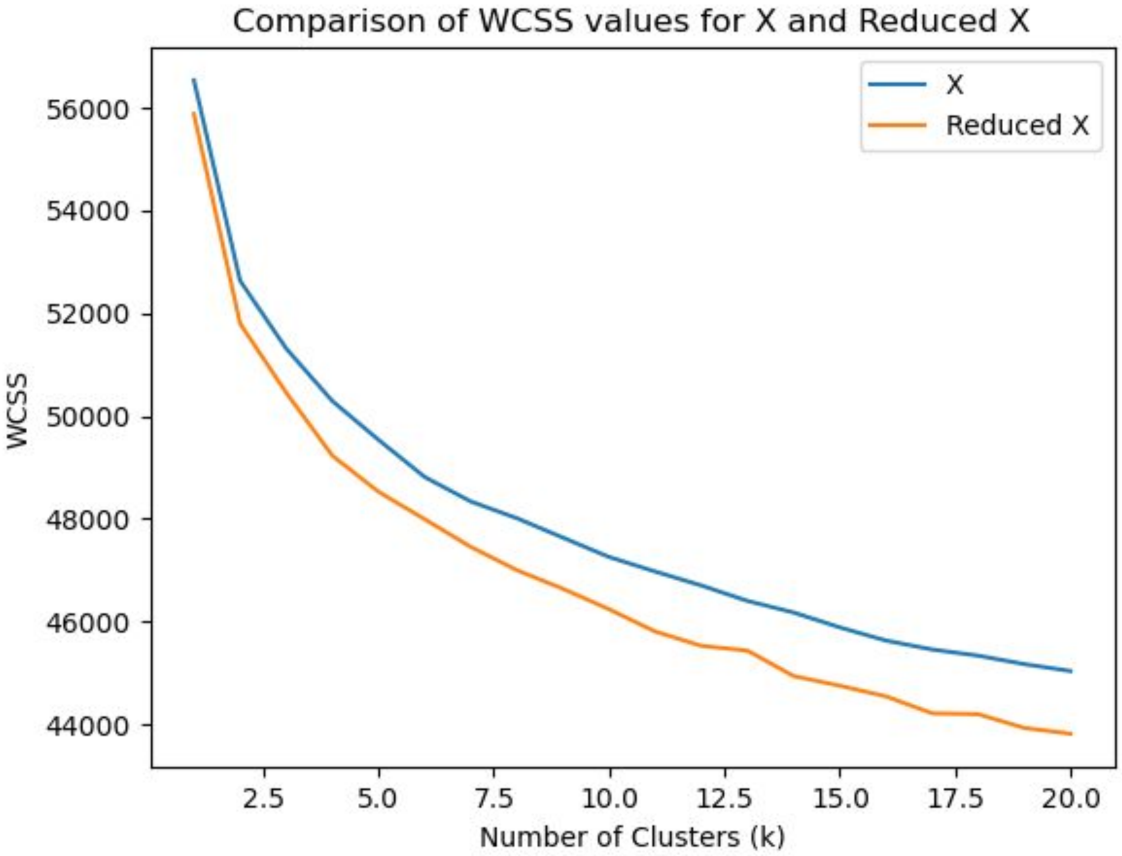
KMeans Clustering with Dataset of dimensions 64

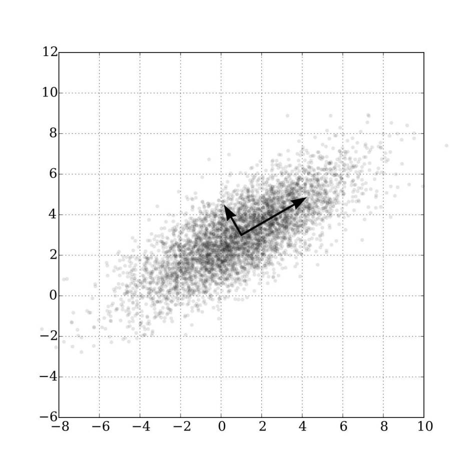
- Loss is almost same
- **δ** = 0.9 and ϵ = 0.5

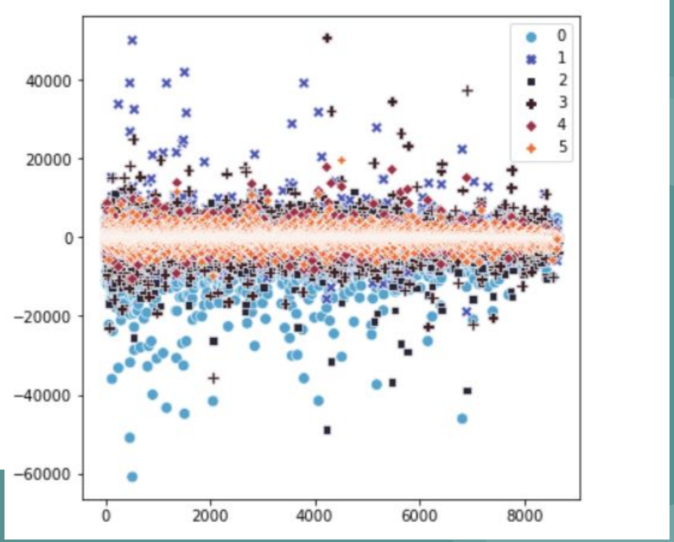


KMeans Clustering with Dataset of dimensions 9k+

- Reduced the dimensions to 650
- Loss is similar a little off
- **δ** = 0.9 and ϵ = 0.5







(Seaborn graph of distribution of data on Credit card dataset when running (PCA, num_components=6)

PCA

- Implemented from scratch
- Simple transformation of data
- Better-suited to low dimensionality

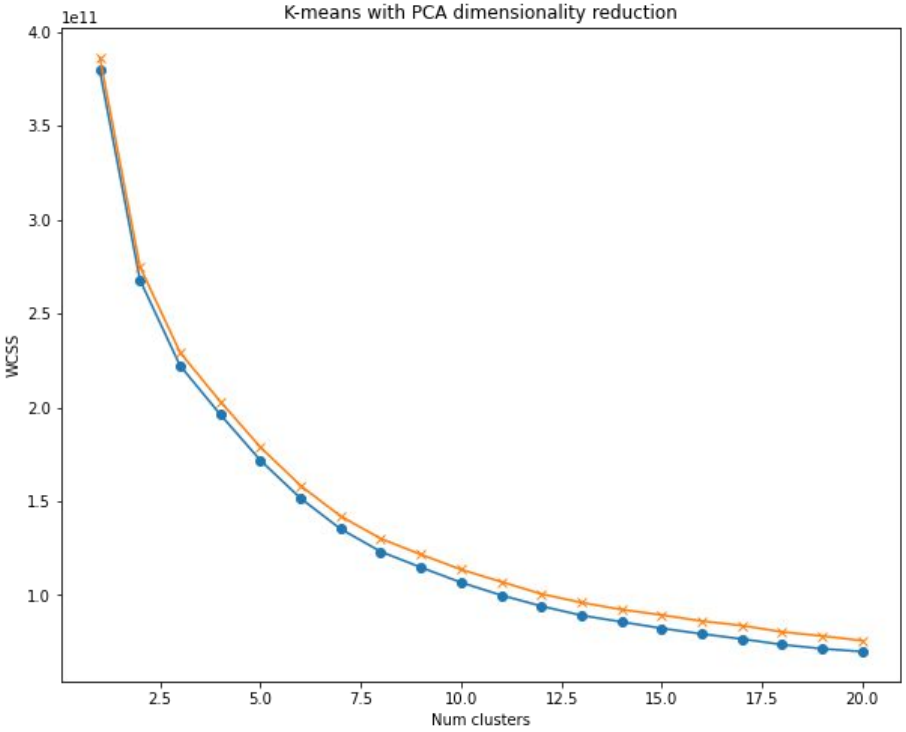
PCA - KMeans implementation Dimensionality=17

- reduced to 6 components
- Loss function: WCSS

- High preservation of cost at low dimensionality

- Perfect cost preservation all the way down to 7 components (41% of original)

Original = Orange
PCA = Blue



Comparison of WCSS between original data and PCA data

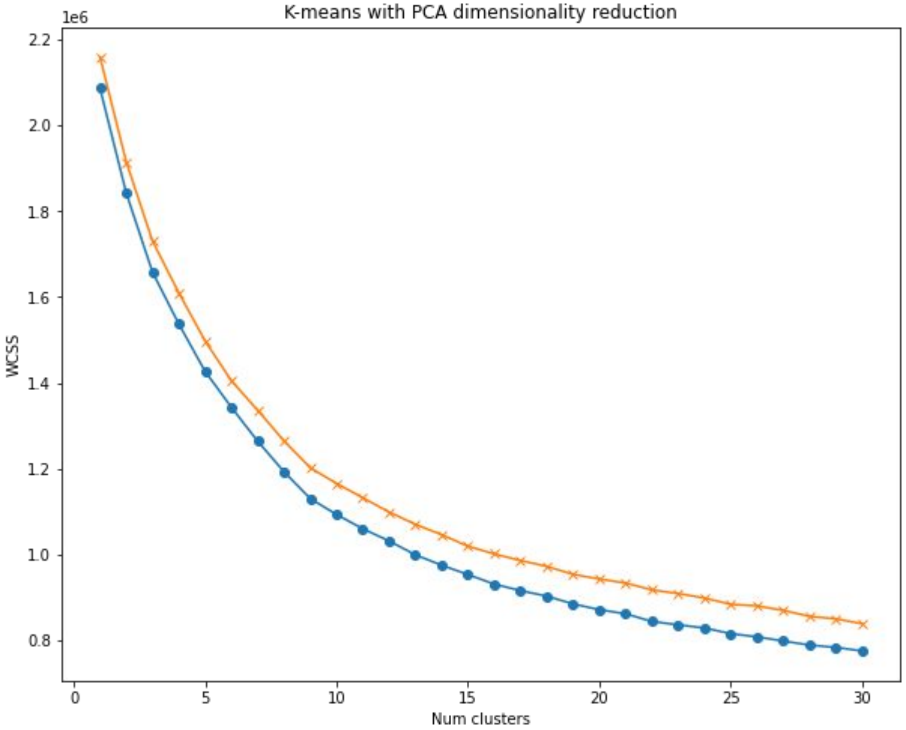
PCA - KMeans implementation Dimensionality=64

- reduced to 32 components

- Able to preserve cost, but less efficiently

- Cost preservation perfect only down to ~42/43 components (66% of original)
- Still pretty strong

Original = Orange
PCA = Blue



Comparison of WCSS between original data and PCA data

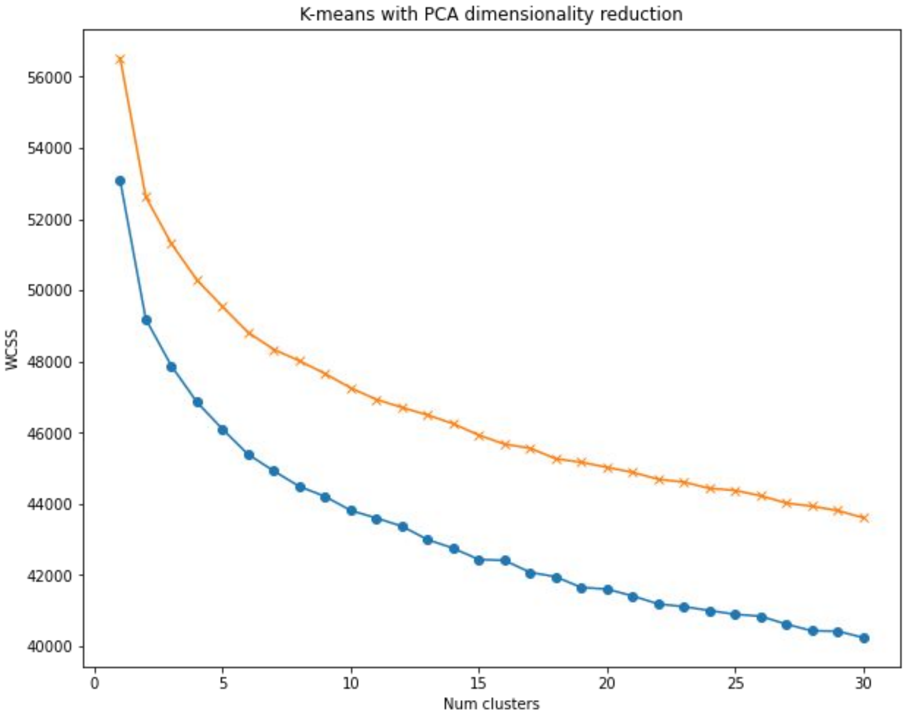
PCA - KMeans implementation
Dimensionality=9K+

- reduced to 900 components

- much harder time preserving cost

- Behavior remains the same
- Cost preservation only perfect down to 6K

- Higher values of n → longer compute time



Conclusion

- Dimension reduction:
  - PCA/SVD perform very well with low-dimensionality
    PCA requires fewer components
  - Struggles with bigger data
  - Factor Analysis not sufficient
- Projection:
  - Johnson-Lindenstrauss addresses weaknesses, remains very efficient
  - Very well-suited to high data

- Future work:
  - Testing on other clustering algs (K-Median? Hierarchical?)
  - Trying new projection algorithms, getting more datasets

Questions?