CSE564: Visualization and Visual Analytics
Dimensionality Reduction: PCA, MDS and Scatterplot matrix
Project 2 Report

1) Dataset:

The dataset selected for analysis is an IMDb movie dataset. This dataset was also used by me in the 1st assignment. It contains various attributes related to popular movies. The total number of movies regarding which it contains information is 5000. The attributes selected for clustering are the following:

1) num_critics_for_reviews: Number of critics who reviewed the movie
2) duration: Duration of the movie in minutes
3) director_facebook_likes: Number of Facebook likes received by the director
4) actor1_facebook_likes: Number of Facebook likes received by actor1
5) actor2_facebook_likes: Number of Facebook likes received by actor2
6) actor3_facebook_likes: Number of Facebook likes received by actor3
7) gross: Box office gross of the movie in USD
8) num_voted_users: Number of users who voted for the review of movie on IMDb
9) cast_total_facebook_likes: Total Facebook likes received by the whole cast
10) facenumber_in_poster: Total number of faces on the poster of the movie
11) num_user_for_reviews: Number of users who reviewed the movie on IMDb
12) budget: Budget of the movie in USD
13) imdb_score: The IMDb score of the movie
14) movie_facebook_likes: Total number of Facebook likes received by the movie

2) Task 1: Data clustering and decimation

1) Random sampling:

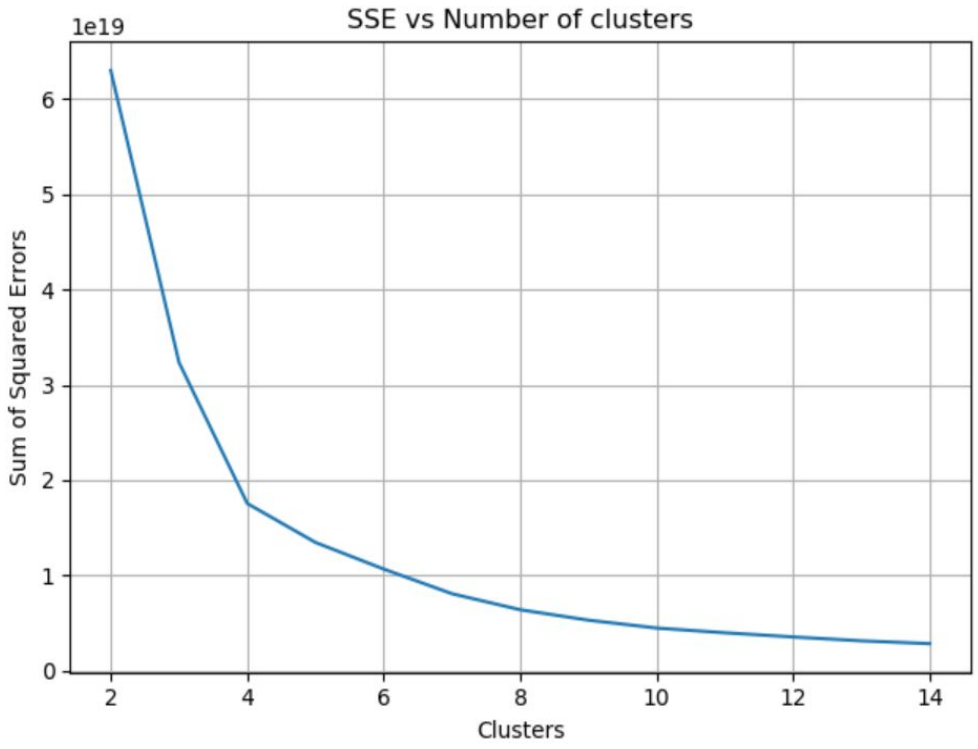
I performed random sampling by using the **random.sample()** function of python. I kept only **25 % of the data** as mentioned in the assignment instructions pdf. First, I converted the dataframe to a numpy array. The below **code snippet** shows random sample generation:

```
109     def random_sample_generation(data):
110         data_np = np.array(data)
111         return data_np[random.sample(range(len(data_np)), int(0.25 * len(data_np)))]
```

2) **Stratified sampling:**

For **stratified sampling**, I first segregated the data into an **appropriate number of clusters**. The number of clusters was chosen by performing the **elbow method** by performing **k-means clustering** first. For k-means clustering, I used the sklearn library in python. I performed clustering for the number of clusters ranging from 2 to 14 (the maximum number of attributes of the data) iteratively. After performing k-means clustering for all these iterations, I plotted the graph for the **'Sum of Squared Errors'** vs **'Number of clusters'.** Using the **elbow method**, the **appropriate number of clusters** came out to be **4.** The below code snippet shows the plot for elbow plot generation.

```
147  def elbow_plot(sum_of_squared_errors):
148      fig = plt.figure()
149      ax = fig.add_subplot(111)
150      ax.plot(range(2, 15), sum_of_squared_errors)
151      plt.grid(True)
152      plt.title('SSE vs Number of clusters')
153      plt.xlabel('Clusters')
154      plt.ylabel('Sum of Squared Errors')
155      plt.xticks(range(2, 15, 2))
156      plt.show()
```



The figure above clearly shows that the elbow point of the k-means clustering occurs at **n = 4**. Hence, I select n = 4 for clustering and then based on that, I do stratified sampling.

3) Task 2: Dimension reduction on both org and 2 types of reduced data

a) **Intrinsic dimensionality of the data using PCA:**
The **intrinsic dimensionality** of the data is explained by the **variance ratio** extracted by the **PCA**. I have kept the **threshold** for the data for intrinsic dimensionality to be **90%.** This means that the intrinsic dimensionality of the data explains **90% variance** in the data. We use scree plot to explain the intrinsic dimensionality of the data. The intrinsic dimensionality of the data is **marked** by a **Factor Line** in the **Scree-plot**.

i) When there is no sampling, the intrinsic dimensionality of the data is 1.
ii) When there is random sampling, the intrinsic dimensionality of the data is increased to 2.
iii) When there is stratified sampling, the intrinsic dimensionality of the data is also increased to 2.

Hence, there is a **bias introduced** which can be explained by the changing of the intrinsic dimensionality from 1 to 2.

Three attributes with highest PCA loadings:

Comparison of Random Sampling and Stratified sampling:

Random sampling is completely random. Hence, it can introduce huge bias sometimes, whereas very low bias at other times. Stratified sampling is more representative of all the samples in the data and samples the data more equivalently from all types of samples from the data.

b) **Top three attributes with highest PCA loadings:**
i) When there is no sampling:
1) **facenumber_in_poster**
2) **num_critic_for_reviews**
3) **imdb_score**

ii) When there is random sampling:
1) **actor_3_facebook_likes**
2) **num_critics_for_reviews**
3) **imdb_score**

iii) When there is stratified sampling:

1) duration

2) director_facebook_likes

3) actor_1_facebook_likes

```
117  def pca_generation(data):
118      pca = PCA()
119      pca.fit(data)
120      pca_results = pca.explained_variance_ratio_
121      return pca_results
122
123
124  def pca_top_three_attr(data):
125      pca = PCA()
126      pca.fit(data)
127      loadings = np.sum(np.square(pca.components_), axis=0)
128      return loadings.argsort()[-3:][::-1]
```

The above code snippet shows the code for generating PCA (**pca_generation()**) for the given data and also for finding the top three attributes for the given data (**pca_top_three_attr()**).

4) Task 3: visualization of both original and 2 types of reduced data:

All the plots are displayed on the webpage and are explained in the video. The code for PCA for top two components, MDS with Euclidian distance, MDS with Correlation distance is given in app.py and the data is passed to the index.html file via d3_data

Learnings & Comparison between different methods:

All the methods of dimensionality reduction are important in their own sense.

1) Random sampling and stratified sampling can be applied to the data for quick reduction of data.

2) Scatterplot matrix helps us find relations between important components.
3) PCA helps us reconfigure the data into most important components and lets us figure out the role of each of the attributes in the data.
4) MDS is used to find the similarity or dissimilarity between data.