**Relatório**

**Sistema de recomendação**

**Capstone Project**

Tiago Pincowscy   
Março 05th, 2019

**I. Definition**

**Project Overview**

Hoje no mundo são produzidos uma quantidade incalculável de produção audiovisuais como filmes, séries, novelas, desenhos, vídeos para youtube , entre outros. Devido essa grande quantidade de produções algumas plataformas foram criadas, com o intuito de conter uma grande coletânea de produções, como por exemplo: Netflix, Youtube, Crunchyroll, entre outros.

Contudo o grande número de produções e o tempo limitado para consumi-las cria um problema, selecionar um produto para consumir que irá agradar. Para solucionar este problema as plataformas como Netflix e Youtube utilizam de algoritimos próprios para sugerir produções aos usuários.

A própria Netflix realizou uma competição para melhorar o seu algoritmo de recomendação, http://netflixprize.com/index.html. Ainda inspirada pela Netflix, segue o artigo cientifico: “Recommendatio System for Netflix” por Leidy Esperanza Molina Fernádez, https://beta.vu.nl/nl/Images/werkstuk-fernandez\_tcm235-874624.pdf.

Neste projeto foi criado 2 modelos de recomendação utilizando algoritmos de machine learning não supervisionado, GMM e Kmeans, para criar um agrupador baseado na similaridade das produções, Filtro Demográfico, outro baseado na similaridade dos “gostos” dos usuários, Filtro Colaborativo.

**Problem Statement**

Nosso objetivo é criar 2 modelos de sugestões, com sugestões baseadas em diferentes critérios, neste projeto iremos criar o Filtro Demográfico e o Colaborativo. Serão realizadas as seguintes atividades:

1. Download das 3 Tabelas( Anime, Anime\_list e users).
2. Limpar as tabelas
3. Modificar tabela para aplicar PCA ( Redução de dimensões para visualizações)
4. Aplicar métricas para modelo 1 – GMM
5. Aplicar métricas para modelo 2 – Kmeans
6. Analisar e escolher o melhor modelo
7. Explorar Agrupadores
8. Realizar Passos 3 a 7 no outro Filtro.

**Metrics**

Como os modelos utilizados neste trabalho são o GMM e o KMeans, serão utilizadas as seguintes métricas para descobrir qual é o melhor modelo e o melhor número de agrupadores:

* Silhouette Score : Ele é calculado utilizando a média da distância intra-cluster (a) e a média da distância do cluster mais perto (b), isto para cada dado. Ele será utilizado no:
  + GMM
  + Kmeans
* BIC - Bayesian Information Criterion : nos dá uma maneira de escolher entre dois diferentes modelos com diferentes números de parâmetros, selecionando o que nos dá a menor pontuação BIC. Entretando quanto menor o BIC, maior o “overfitting” do modelo. Ele será utilizado no:
  + GMM
* Elbow Method: Esse método examina a porcentagem de variação explicada como uma função do número de clusters. Traçar a porcentagem de variância explicada pelos clusters em relação ao número de clusters, os primeiros clusters adicionarão muita informação (explicam muita variação), mas em algum ponto o ganho marginal cairá, dando um ângulo no gráfico. O número de clusters é escolhido neste ponto, daí o "critério do cotovelo". Esse "cotovelo" nem sempre pode ser identificado sem ambigüidade.Ele será utilizado no:
  + Kmeans

**II. Analysis**

**Data Exploration**

**Tabela Anime:**

Tabela que possui os dados de cada produção, como: título, estúdio, produtora, mas também possui dados de popularidade, como: pontuação média, quantidade de membros na comunidade, quantidade de pontuações recebidas, etc. Possui inicialmente 14 474 registros.

* Foram retirados dados nulos:
  + Retirar Gender NaN
  + Retirar Rating = None
  + Retirar Type = Unknown
  + Retirar Source = Unknown
* Foram retiradas animes que, na época, ainda não tinham sido lançados;
* Foram retiradas animes que receberam notas de 0 pessoas.
* Devido ao grande número de dados nulos de producer e studio, não serão utilizados no modelo, pois a outra opção seria retiras os dados nulos que seria 40% da tabela a menos.
* As colunas que não serão utilizadas para classificação, retirada de informações ou possui dados irrelevantes para este projeto foram descartadas para facilitar o processamento dos dados. Foram elas:
  + 'title\_english', 'title\_japanese', 'title\_synonyms', 'image\_url', 'broadcast', 'related', 'licensor', 'opening\_theme', 'ending\_theme', 'background', 'premiered','aired\_string', 'aired' ,'airing', 'duration', 'rank', 'popularity', 'status'.

**Tabela Users:**

Tabela onde possui informações pessoais de cada usuário, como : apelido, ID, gênero, data de nascimento, localização e também possui alguns dados sobre suas atividades, como : última vez online, quantos episódios assistidos, etc.

* Foram retirados dados nulos
* Coluna birth\_date alterada para age, onde possui a idade do usuário
* A coluna location, embora eu queira usar, ela não possui dados confiáveis, pois os usuários podem preencher com qualquer informação, verdadeira ou não.
* As colunas que não serão utilizadas para classificação, retirada de informações ou possui dados irrelevantes para este projeto foram descartadas para facilitar o processamento dos dados. Foram elas:
  + 'user\_watching', 'user\_completed', 'user\_onhold', 'user\_dropped', 'user\_plantowatch', 'user\_days\_spent\_watching', 'join\_date', ’last\_online', 'stats\_rewatched', 'stats\_episodes','location'

**Tabela Anime List:**

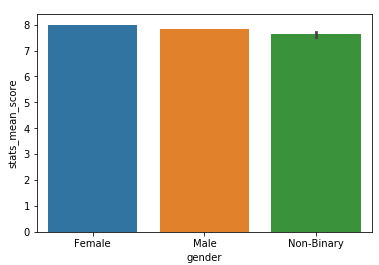
Tabela onde possui informação sobre cada pontuação dos usuários, como: qual pontuação, apelido do usuário, ID do anime, entre outros.

* Foram inseridos na tabela o título do anime e o id do usuário.
* Foram retirados todos os usuários que não estavam na tabela “Users”;
* Foram retirados usuários que assistiram 0 episódios, mas pontuarão
* Foram retirados usuários que deram notas a menos de 20 animes ou para mais de 300 animes.
  + Menores que 20 pontuações dificulta a identificação do perfil do usuário.
  + Maior que 300 pontuações, começa a criar uma dúvida sobre a veracidade das informações, pois o site possibilita você a pontuar qualquer anime independentemente se você assistiu ou não.
  + Devido à grande quantidade de dados nesta tabela, 24 milhões, o computador não estava suportando o seu processamento, logo este corte dos usuários foi fundamental para a continuidade do projeto.
* As colunas que não serão utilizadas para classificação, retirada de informações ou possui dados irrelevantes para este projeto foram descartadas para facilitar o processamento dos dados. Foram elas:
  + 'my\_start\_date', 'my\_finish\_date', 'my\_rewatching', 'my\_rewatching\_ep', 'my\_last\_updated', 'my\_tags', 'my\_status', 'my\_watched\_episodes'

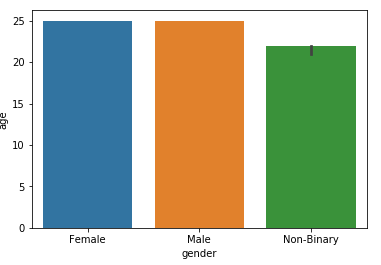
**Exploratory Visualization**

**Usuários :**

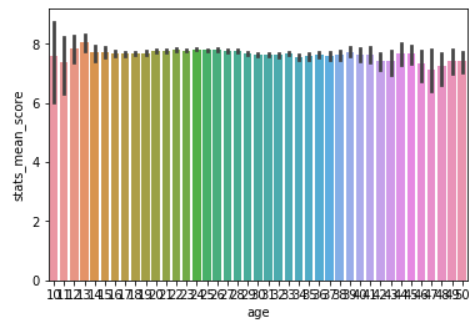
Pontuação média X Gênero

****

Idade média X Gênero

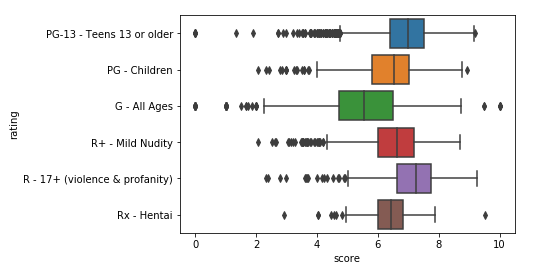
****

Pontuação média X Idade

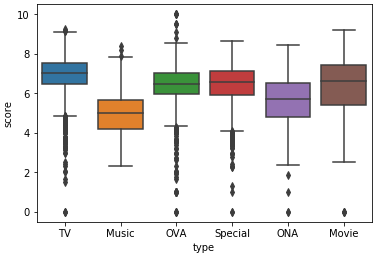
****

**Animes :**

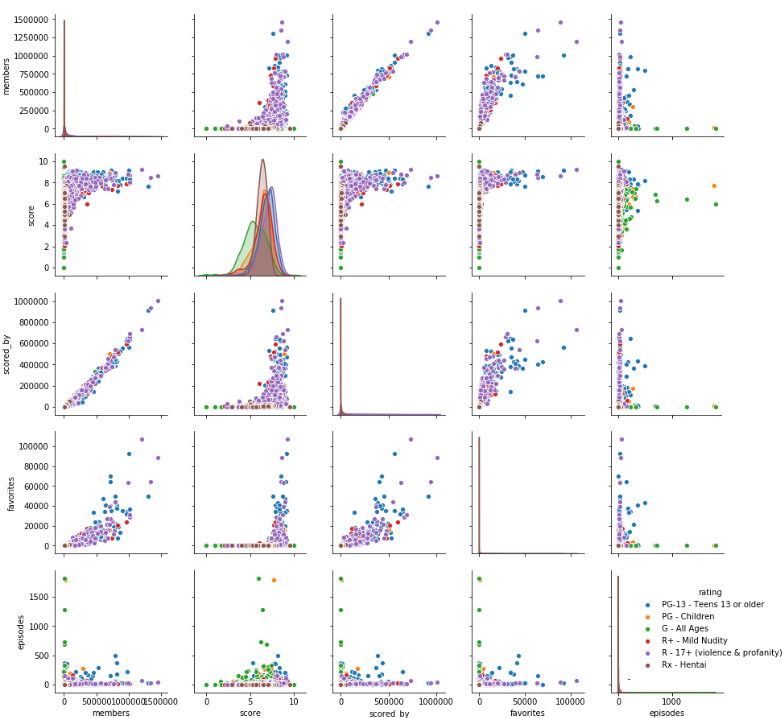
Pontuação média X Faixa Etária recomendada

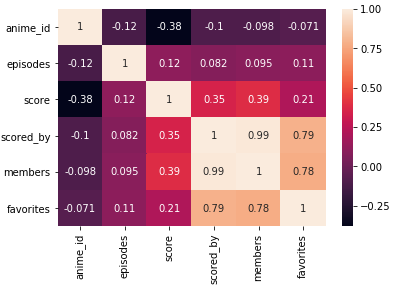


Pontuação média X Tipo de Produção

****

Relação geral entre as colunas.

****

****

**Algorithms and Techniques**

Este trabalho é baseado nos modelos não supervisionados do Machine Learning, logo será utilizados 2 algoritmos que utilizam cálculos diferenciados para criar suas “formas” de divisão entre kluster, são eles GMM – Gaussian Mixture e o Kmeans.

* GMM – Gaussian Mixture : Possui mais parâmetros, mais flexível, melhor para data sets uniformes, consegue assumir outras formas além do circulo/esfera, pois calcula a variância de cada ponto.
* Kmeans : Computacionalmente eficiente, muito bom com grandes data sets, é melhor para dados claramente separados.

**Benchmark**

O benchmark utilizado para este projeto se encontra no link: <https://www.kaggle.com/tanetboss/user-clustering-for-anime-recommendation/notebook>, nele o analista “Lastnight” utiliza as seguintes atividades:

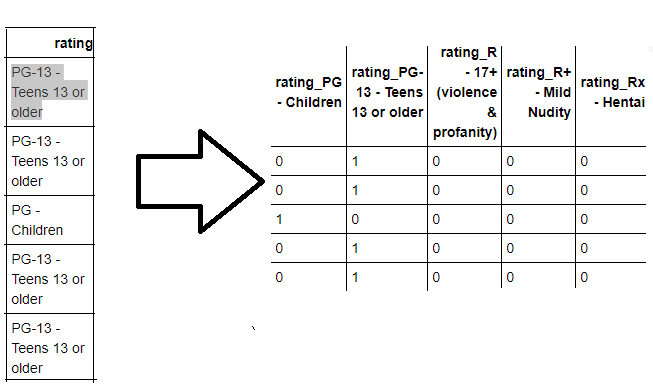
1. Transforma o sistema de pontuação.
   1. Ele transforma do sistema de 1 a 10, para gostou ou não, utilizando o cálculo : se a pontuação atribuída for menor que sua média geral, sinalizado como “não gostei”, caso contrário sinaliza “gostei”.
2. Modela a tabela para implementar PCA.
   1. Cria a tabela conjunta e em cima dela cria uma crosstab do “user\_id” com “name”
3. Calcula o número ideal de cluster pelo “elbow method” e pelo silhouette score
4. Implementa apenas o Kmeans
5. Retirar informações dos Clusters obtidos.

Podemos realizar comparativos principalmente em relação ao modelo utilizado e a quantidade de clusters observado como ideal. Entretanto a base de dados utilizada pelo modelo benchmark é diferente, com uma quantidade bem menor de dados. Podemos também comparar os gêneros principais de cada cluster. Outro ponto interessante seria o “como” ele tratou os dados antes de implementar no modelo, o sistema binário “gostei” ou “não gostei”.

**III. Methodology**

**Data Preprocessing**

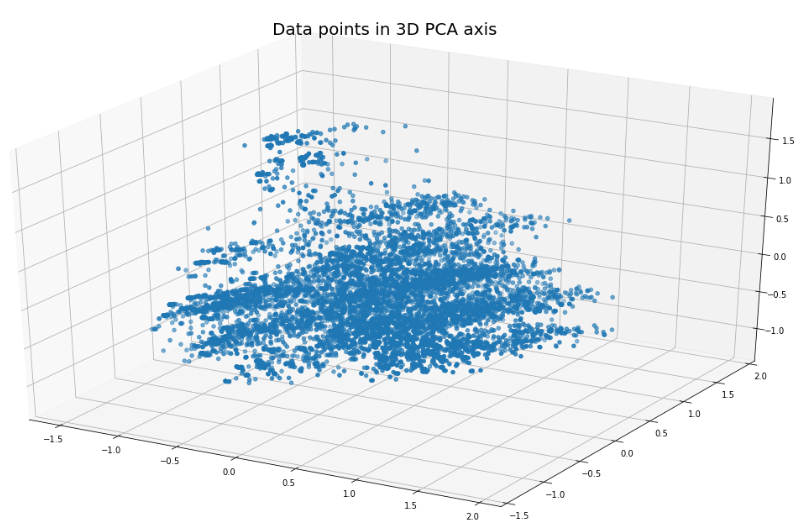
**Adequar Tabela para Modelo do Filtro Demográfico:**

O filtro demográfico utilizará apenas a tabela “Anime”, será necessário criar colunas “dummies”, ou seja, para cada classificação possível da coluna original possuirá sua própria coluna separada como por exemplo a coluna original “rating”: 

Esta modelagem irá acontecer também com as colunas: “type” , “source” e “genre”.

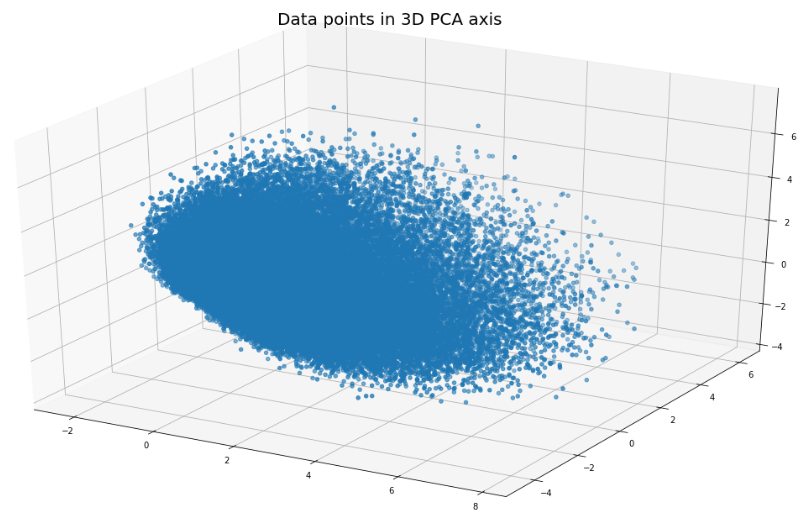
As colunas “anime\_id” e “title” serão deletadas, pois não será possível colocas no modelo de redução de dimensões, PCA. Mas antes de implementar o PCA é necessário normalizar os valores da tabela, para este projeto foi utilizado o MinMaxScaler, ele transforma todos os valores em números entre 0 e 1.

Inicialmente eu iria utilizar o np.log, mas ele criaria valores nulos que impossibilita sua utilização.

O PCA escolhido foi 3 para conseguirmos visualizar os resultados. Segue abaixo a imagens dos dados :

**Adequar Tabela para Modelo do Filtro Colaborativo:**

O filtro colaborativo utilizará as tabelas “Anime List” e “Users”, foi necessário um merge das duas tabelas, para então criar uma crosstab utilizando as colunas “user\_id” e “title” da tabela conjunta. Todos os valores nulos foram alterados para zero. Foi utilizado o MinMaxScaler para normalizar a base, depois foi implementado o PCA = 3 para visualização. Segue abaixo a imagem dos dados:

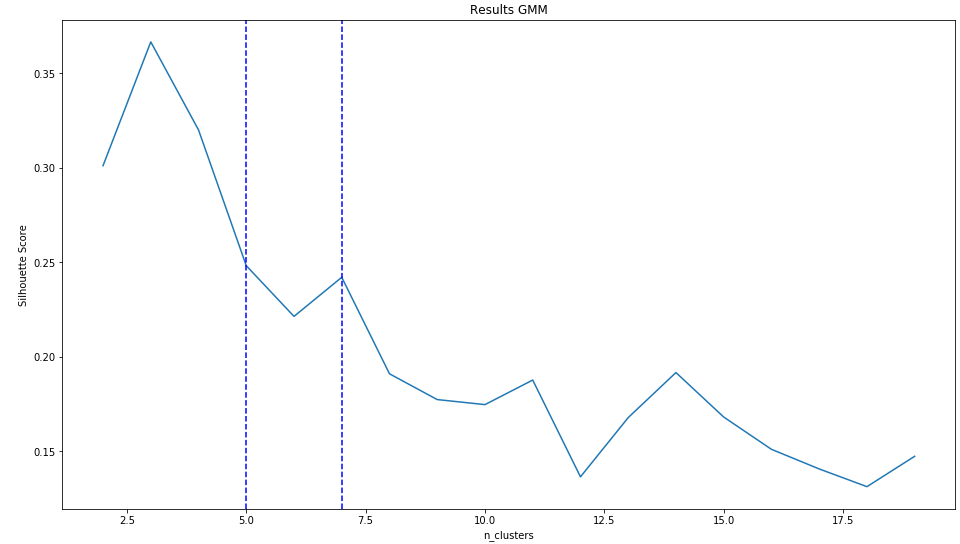


**Implementation**

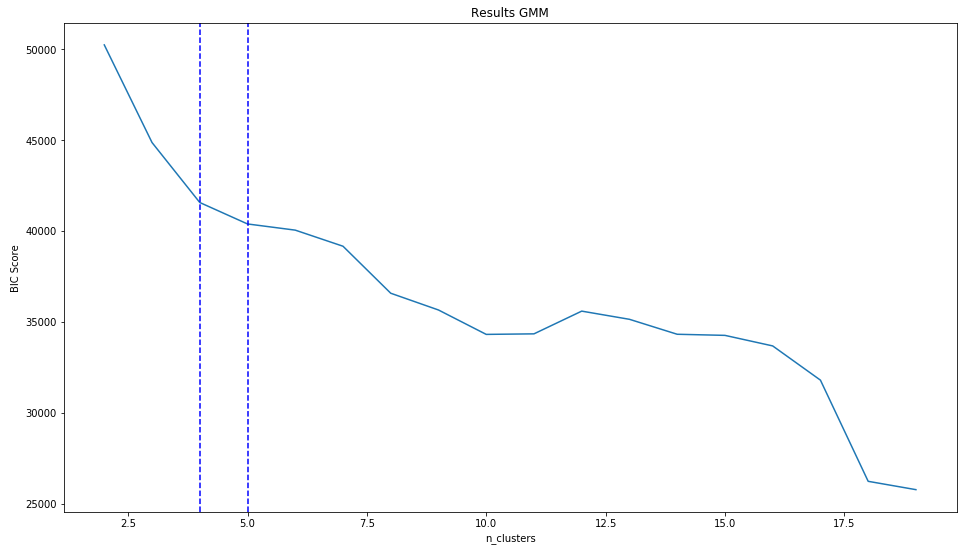
**Modelo do Filtro Demográfico**

**Resultado da métrica - GMM**

Silhouette score = Resultado 5 e 7

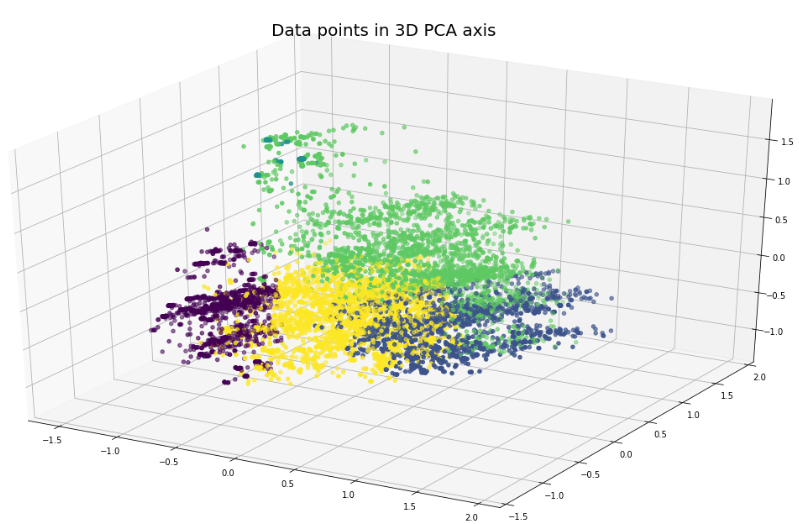


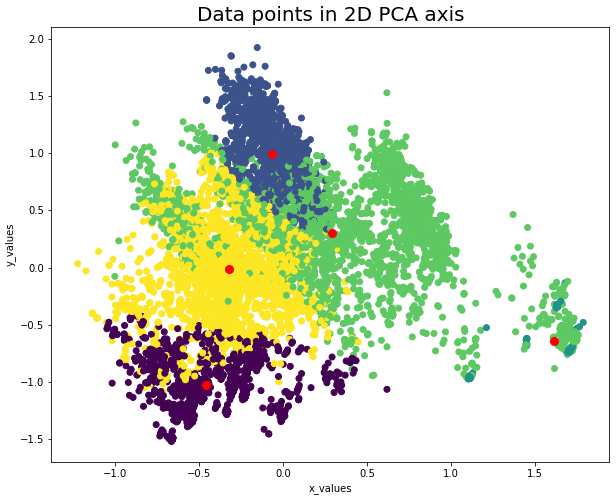
BIC score = Resultado 4 e 5



Podemos observar nos gráficos acima os resultados do BIC score e do Silhouette score no modelo GMM entre 2 e 20 clusters. O Bic Score aponta o K =4 ou 5 como número ideal de clusters e o Silhouette score indica o 5 ou 7. Foi escolhido no caso o K=5 pois aparece nas 2 métricas.

Importante desacatar o porquê o K =3 do Silhouette score foi ignorado. Este tema foi escolhido por alguns motivos, meu interesse em sistemas de recomendação e meu conhecimento sobre o mundo dos Anime(Otakus), então eu consigo deduzir que um Cluster com apenas 3 grupos não irá identificar a grande quantidade de nichos de usuários que existe no universo dos animes. A partir de 5 já começa a ficar mais plausível.

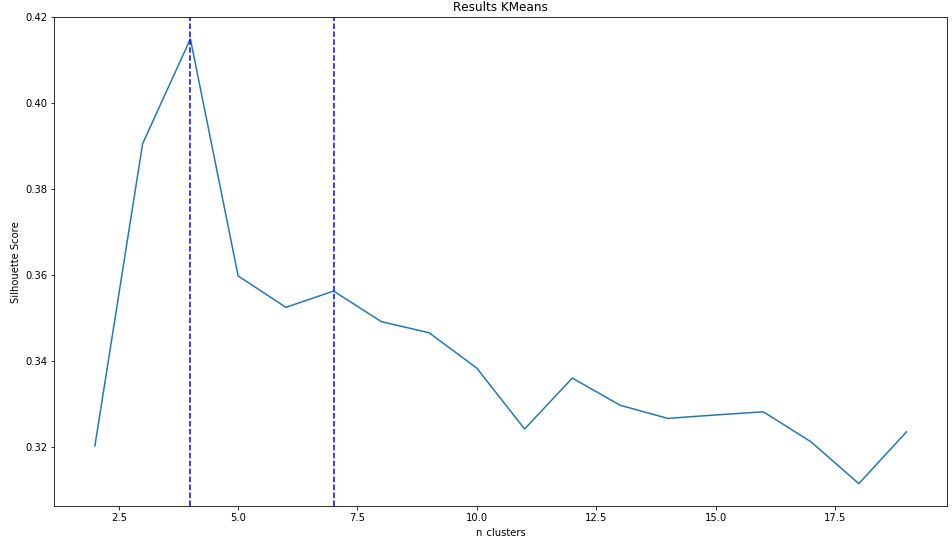
**Implementação – GMM** 



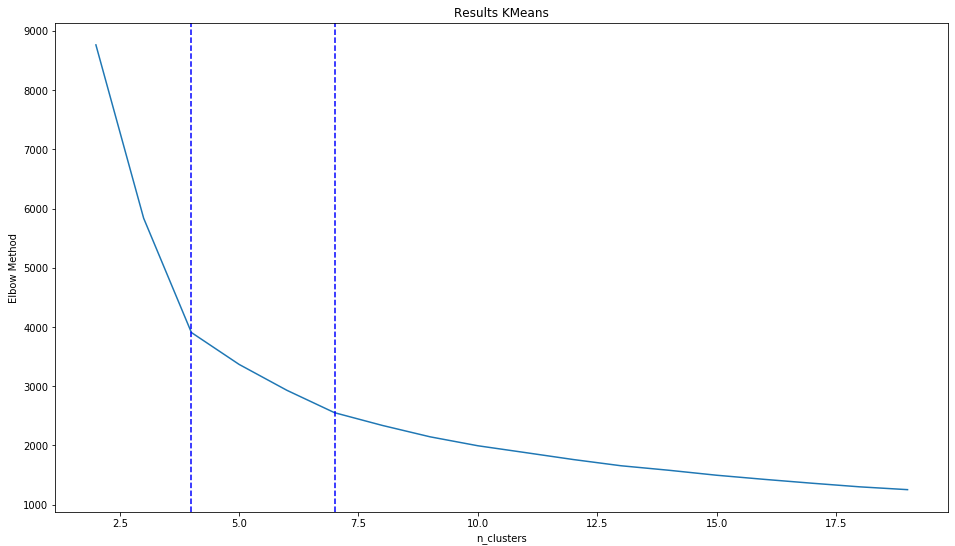
Nas imagens acimas conseguimos visualizar como foram divididos os 5 grupos e onde se encontra o centro de cada grupo. Podemos observar agora que este data set não é muito uniforme e que o grupo do centro mais à direita (Imagem 2D) ou mais a cima (Imagem 3D) representado pela cor verde escura, não possui um grupo muito bem definido, embora possamos visualizar claramente que este pedaço dos dados seria um grupo próprio.

**Resultado da métrica – Kmeans**

Silhouette score = Resultado 4 e 7

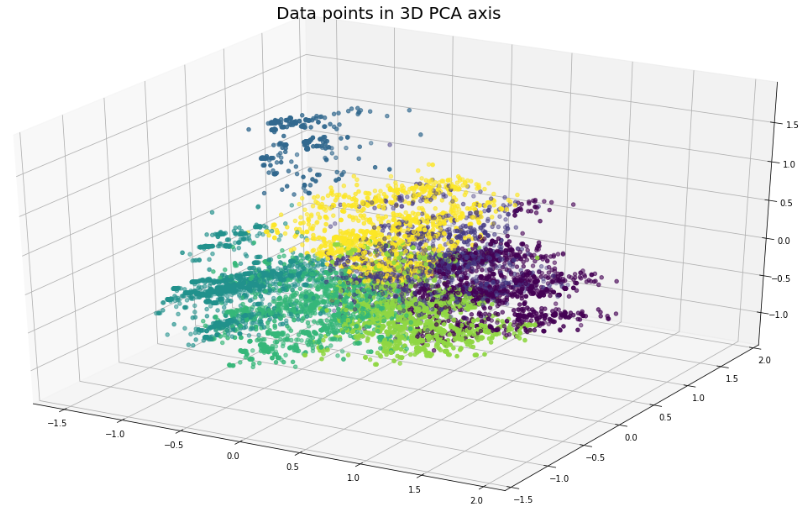
****

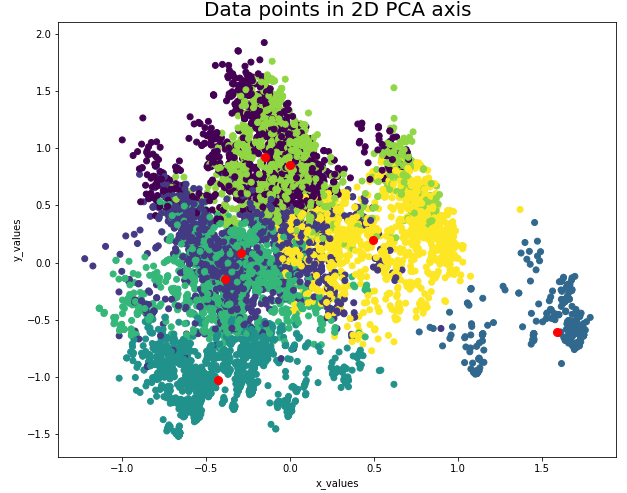
Elbow Method = Resultado 4 e 7



Podemos observar nos gráficos acima os resultados do Elbow method e do Silhouette score no modelo Kmeans entre 2 e 20 clusters, onde ambos indicam os mesmos resultados K= 4 ou 7. Assim como explicado anteriormente iremos escolher o K=7 devido ao número grande de nichos culturais dentro do universo dos animes.

**Implementação – Kmeans**

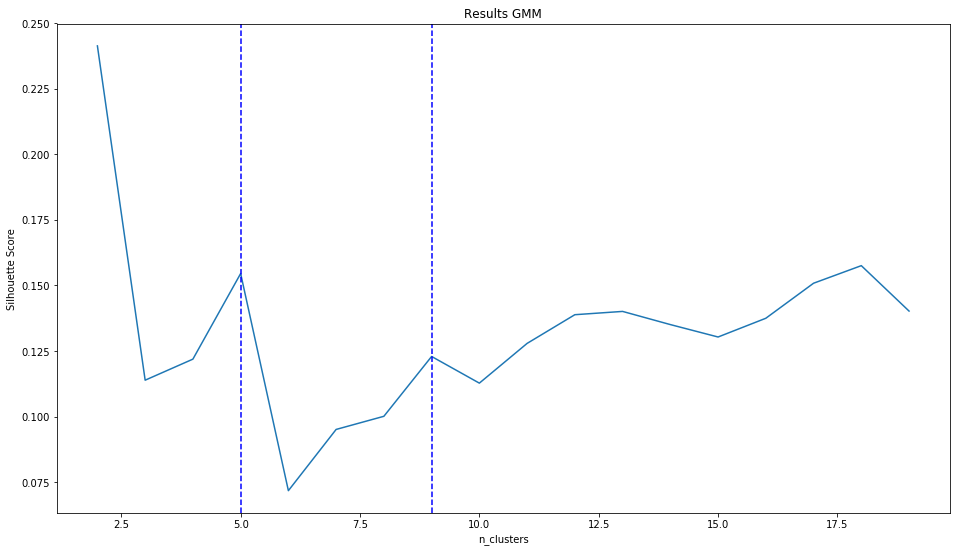


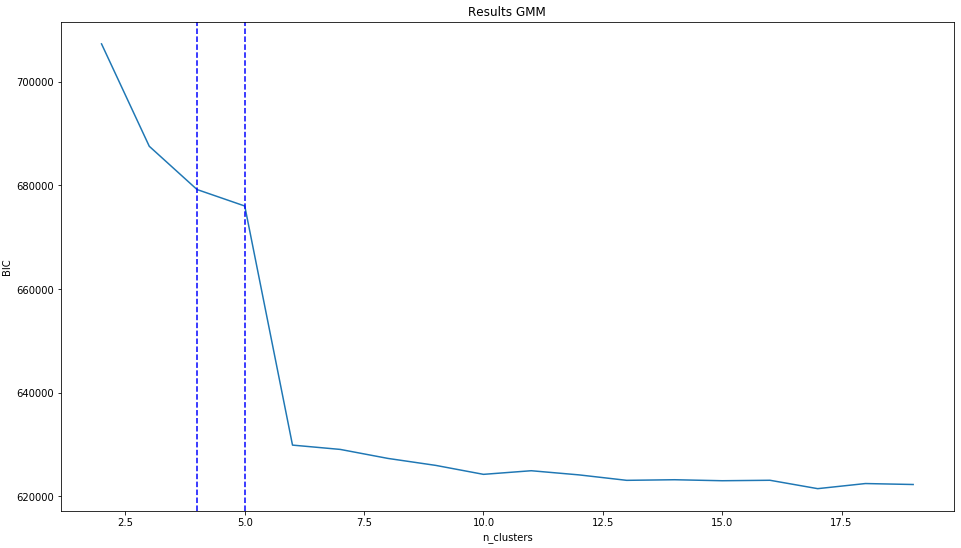


Interessante identificar que os centros tanto no Kmeans quanto no GMM são extremamente similares, mas a divisão criada por cada modelo se diferencia bastante, como o grupo mais à direita (Imagem 2D) ou mais a cima (Imagem 3D), representado por azul, ele agora está claramente representado.

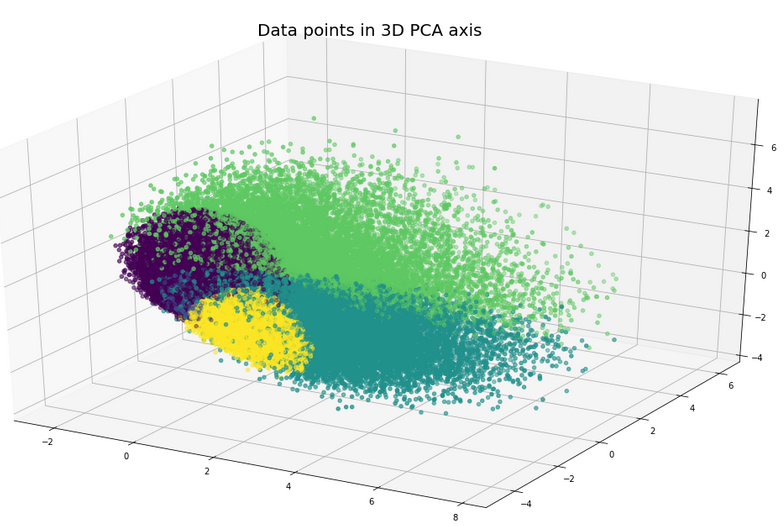
**Modelo do Filtro Colaborativo**

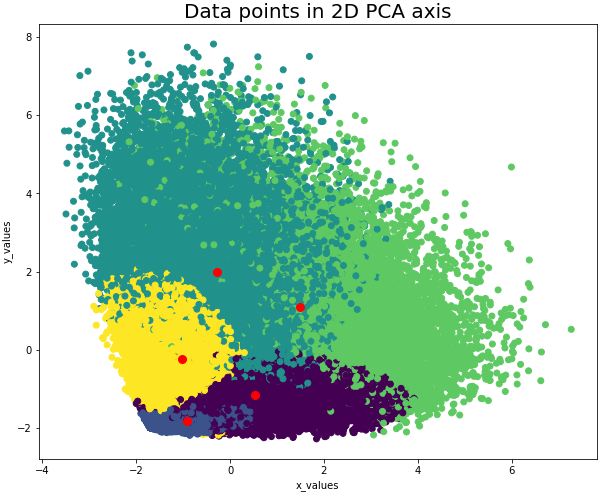
**Resultado da métrica – GMM**

Silhouette Score, K= 5 ou 10

BIC Score, K= 4 ou 5

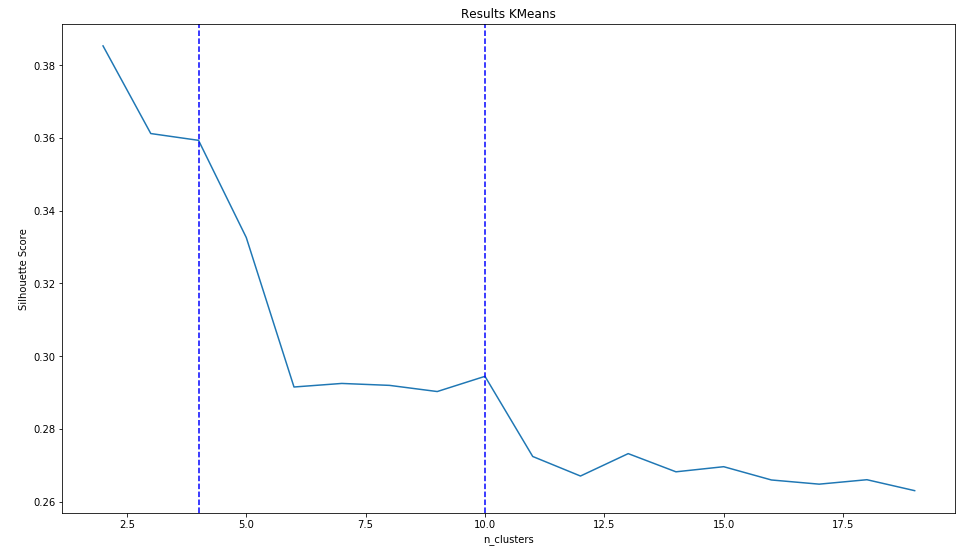
Foi o escolhido o K=5 para ser implementado, pois aparece como ideal nas 2 métricas.

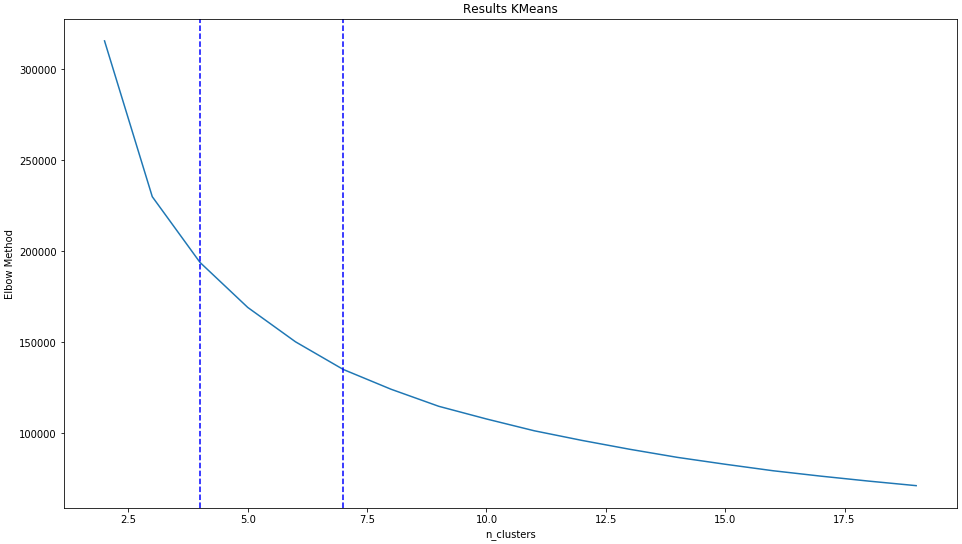
**Implementação – GMM**



Embora os dados estejam uniformes, o GMM não obteve uma boa nota nas métricas. Deve ser pelo fato da concentração de pontos do lado esquerdo (3D), embaixo (2D).

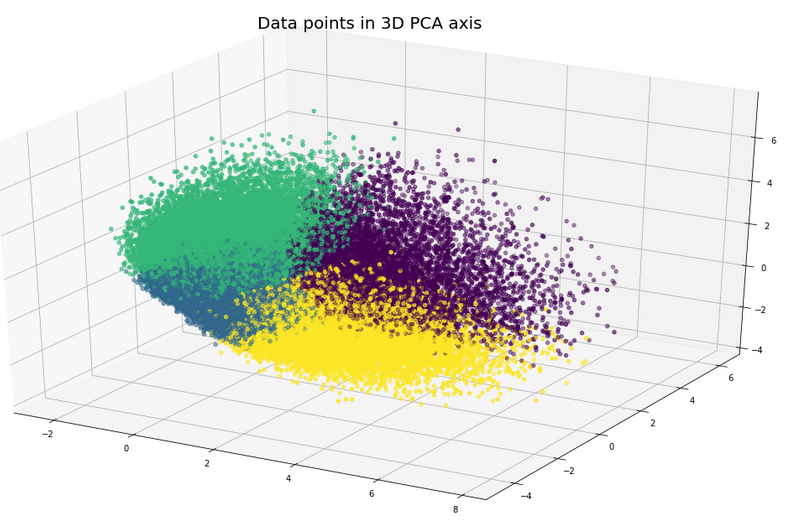
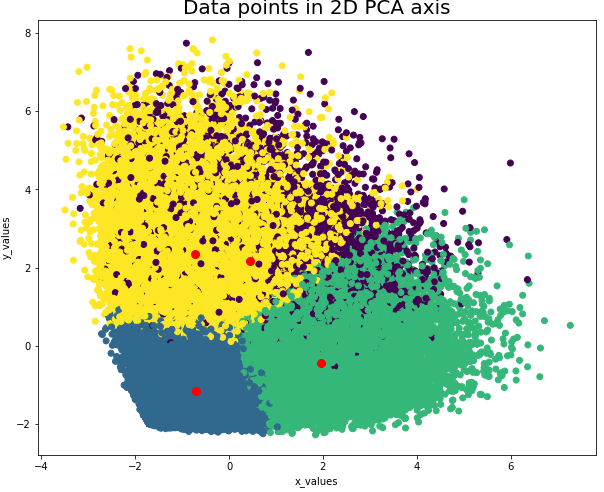
**Resultado da métrica – Kmeans**

Silhouette Score, K= 4 ou 10

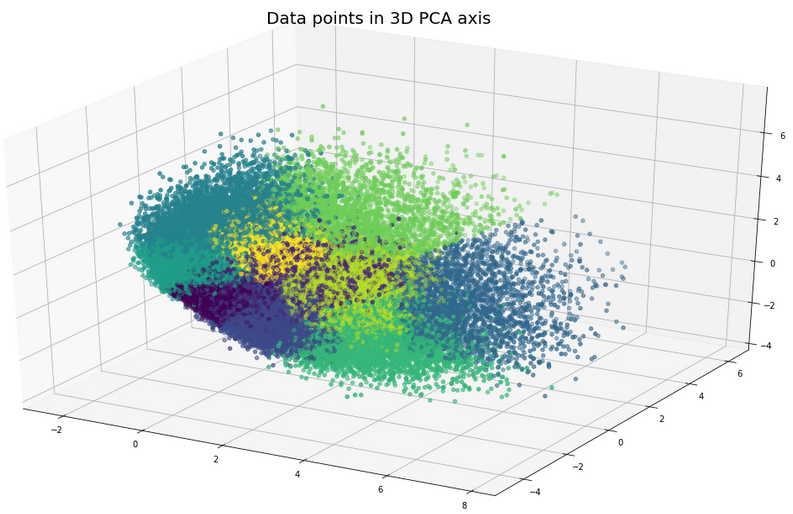
Elbow Method, K = 4 ou 7

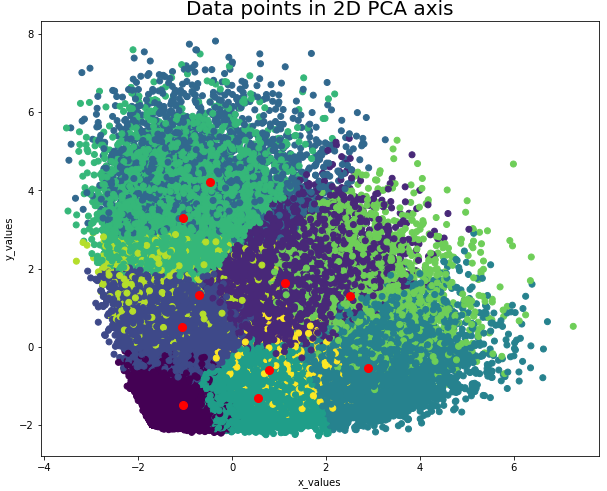
Neste modelo será observado o K=4 e o K=10, pois o benchmark para este projeto, indicou o K=4 no modelo Kmeans como ideal, mas tanto o Kmeans, quanto o GMM apontam o K=10 como uma opção viável.

**Implentação – Kmeans , K=4**

**Implentação – Kmeans , K= 10**





**Refinement**

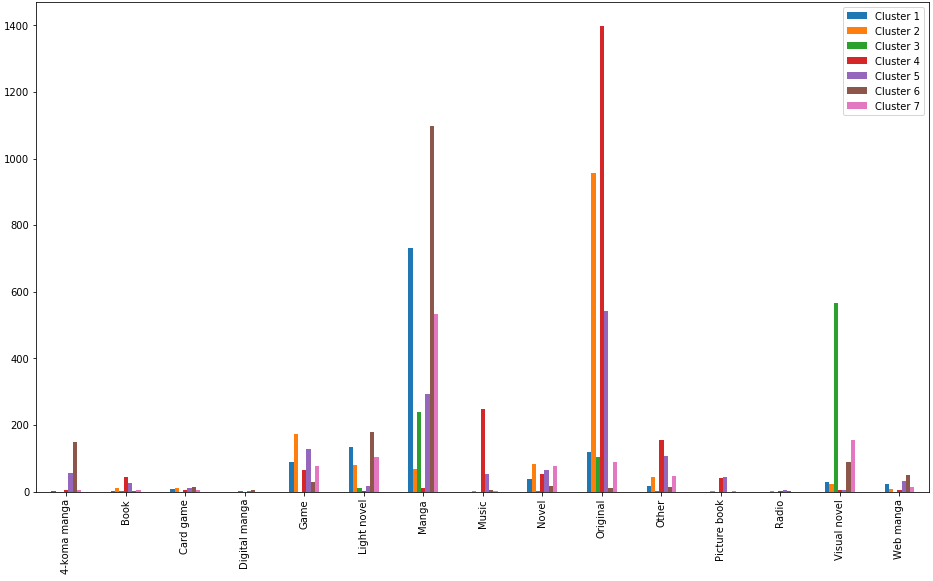
* Não foi usado nenhuma técnica de hyper parametrização, pelo fato do trabalho ter sido construído em cima de um tema subjetivo (impossibilitando qualquer meio de conferir os resultados) e também pelo peso processual que iria ter sobre o projeto, o qual eu não teria como arcar.
* Não senti a necessidade neste trabalho da hyper parametrização pelo fato de estar observando diversos Modelos e métricas com o intuito de encontrar o melhor modelo possível.

**IV. Results**

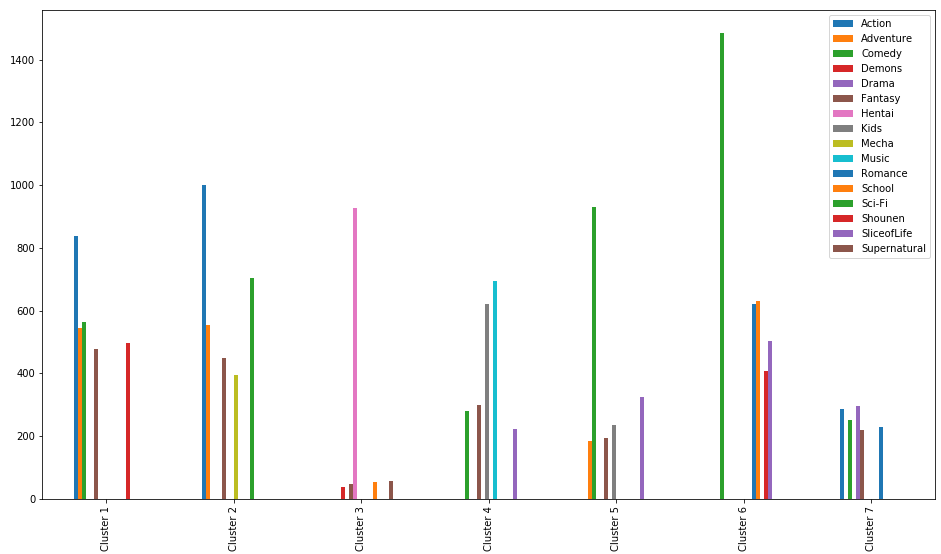
**Model Evaluation and Validation**

**Filtro Demográfico**

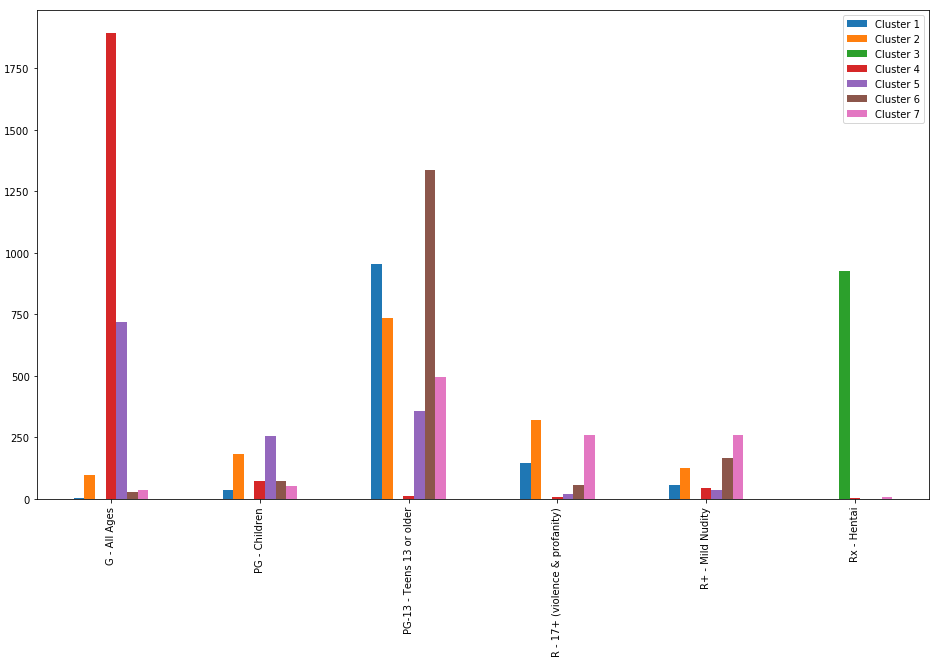
O filtro demográfico retirado a partir da tabela “anime”, teve o melhor resultado no Kmeans com K=7, assim como já explicado anteriormente o K=4 foi ignorado, pois ele não conseguirá identificar, logo recomendar para os vários nichos culturais existente na comunidade Otaku. Segue abaixo algumas visualizações dos dados retirados de cada cluster:

Origem dos Animes 

Gênero dos animes



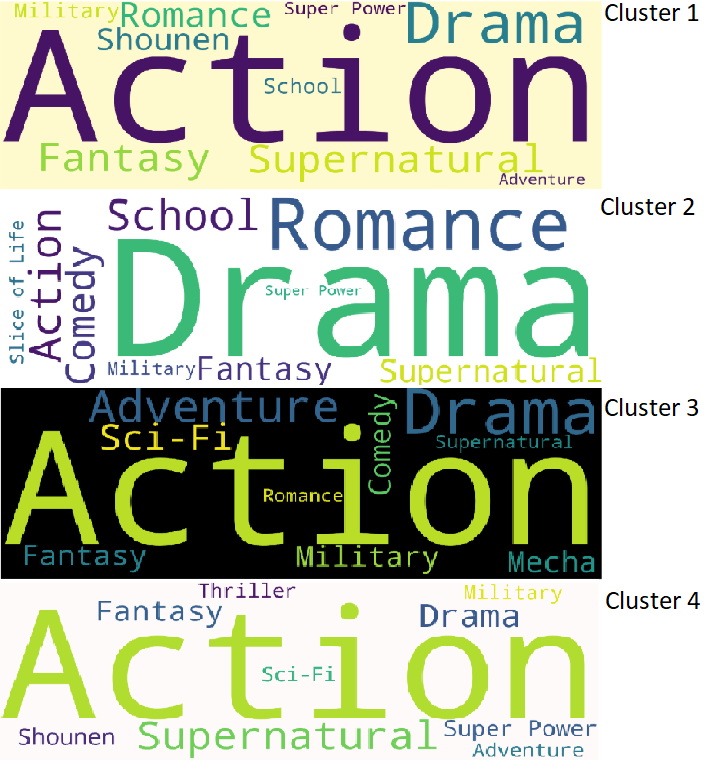
Classificação ( Faixa etária) dos animes



Com as visualizações acima, conseguimos visualizar alguns nichos, como por exemplo o Cluster 2, predomina animes de robôs/ Sci-fi; Cluster 3, onde predomina “Hentai”; Cluster 4 predomina animes para o público infantil.

**Filtro Colaborativo**

O filtro colaborativo retirado do merge da tabela “users” e “anime\_list”, obteve o melhor resultado através do Kmeans, o GMM neste data set obteve resultados nas métrica muito inferiores ao Kmeans, isto se deve ao fato do grande acumulo de dados do lado esquerdo (3D), embaixo (2D). A quantidade de cluster escolhido foi K=10, pois como o projeto benchmark utilizou o K=4, seus agrupadores ficaram sem diversidade, como podemos observar nas imagens abaixo :

Principais gêneros retirados dos Cluster do projeto Benchmark.

Podemos observar que o gênero ação prevaleceu em 3 dos 4, mas ainda aparece com uma participação considerável no cluster 2.

Ao utilizar o K=10, o gênero “Action” ainda prevalece em 6 dos 10 cluster, mas aparece em todos com diferentes participações. A maior diferença seria nos 3 cluster de dominância do gênero “Comedy” e o cluster 7 onde não possui apenas 1 gênero dominante.





Existe algumas similaridades entre o resultado do filtro demográfico e o filtro colaborativo:

* Cluster 7 de ambos, possui gêneros similares;
* Cluster 4 colaborativo com o Cluster 6 do demográfico, ambos possuem foco em animes de âmbito escolar(“Comedy”, “Slice of life”, “Romance” e “School”).
* Cluster 5 colaborativo com Cluster 2 demográfico, ambos possuem “Sci-Fi”, “Mecha” e “Action” como genêros dominantes.

**V. Conclusion**

Embora os 2 filtros possuam suas similaridades, não é interessante para o resultado deste projeto eles serem muito iguais, pois o objetivo do projeto é proporcionar 2 “departamentos” de recomendações diferentes, baseados em algoritmos diferentes para facilitar a escolha do Usuário do Sistema. Plataformas como Netflix, Youtube, entre outros possuem várias abas/departamentos diferentes com recomendações diferentes, pois se trata de um tema subjetivo, gostar ou não gostar de algo, não tem como aferir com 100% de certeza que o usuário irá escolher e gostar de algo recomendado no departamento, por isto é criado vários departamentos/abas com escopos diferentes.

**Reflection**

O problema inicial foi a limpeza dos dados, pois descobrir as peculiariedades de cada coluna dos data set, foi um processo muito oneroso existia colona onde valores eram nulos ou preenchidos com termos que representam valor nulo, como: “none” e “Unknown”, colunas onde valores não eram confiaveis, falta de dados preenchidos, entre outros problemas que me surprenderam.

Entretando o problema mais dificil encontrado foi de como criar e utilizar as informações no modelo de machine learning, muitas vezes encontrava erro de logica no merge, procurava maneiras novas de concatenar tabelas, procurava outros trabalhos de modelos não supervisionados a procura de outros metodos de preparar os dados. Não consegui utilizar o mesmo metodo para os 2 modelos, pois 1 utiliza mais de parte classficatória o outro da pontuação recebida.

Outro problema enfrentado foi o tamanho do data set, mais de 24 milhões de dados, fazendo o projeto ter erros constantes de memoria e longos tempos de processamento, principalmente no calculo das métricas. Isto fez com que eu buscasse formas inteligentes de diminuir o data set sem afetar na estrutura estatisca do modelo.

**Improvement**

Para próximos projetos podemos considerar a criação de outros “departamentos” de recomendação, como por exemplo um hibrido entre os 2 criados neste trabalho onde eu recolheria sua preferência em gênero/ popularidade de animes baseado em sua nota.

Departamento possível seria de tendências, onde seria recomendado baseado na quantidade de visualizações no último período de tempo (1 , 3 ou 7 dias por exemplo).

Outro departamento possível seria baseado em pessoas as quais o próprio usuário escolheu como “influenciadores”, pois ele considera as opiniões dessas pessoas valida, podendo possuir recomendações bem diferentes de outros departamentos.

Entretanto seria necessárias fontes de informações melhores e mais confiáveis, pois o muitas das informações deste data set são preenchida sem qualquer tipo de validação pelo site MyAnimeList.

No aspecto algoritmo e modelos, podemos utilizar alguma técnica de regressão para descobri e balancear melhor os parâmetros para cada departamento de recomendação.