Projeto de Clusterização

Integrantes

Lucas Henrique - lhns3

Introdução

-> O presente projeto tem como objetivo explorar dados do dataset a partir de várias visualizações, preprocessar os dados, agrupá-los em clusters a partir dos modelos de K-means, DBSCAN e Fuzz cMeans e interpretar, discutir e analisar sobre os resultados obtidos.

-> O dataset escolhido foi o **Wholesame customers** (clique no link), o qual mostra os clientes de uma distribuidora varejista, inclui diversos gastos anuais em diversas categorias de produtos, e também.

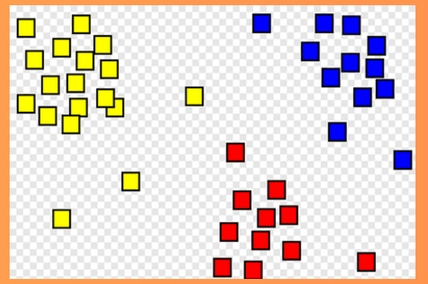
- **Características do Dataset:** Multivariado
- **Área:** Negócios
- **Tarefas associadas:** Clustering, Classificação
- **Tipo dos atributos:** Inteiro
- **Instâncias:** 440
- **Atributos:** 8

*Channel e Region são atributos categóricos

Channel: 1 - Retail, 2 - Hotel, Restaurnt, Cafe

Region: 1 - Oporto, 2 - Lisbon, 3 - Others

Fundamentos



-> Agrupamento

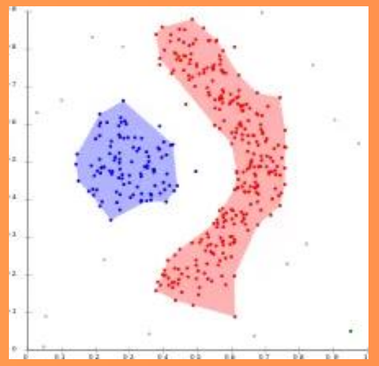
O agrupamento é uma técnica não-supervisionada de machine learning que busca agrupar pontos de dados em clusters, de forma que em um mesmo clusters estejam pontos bastante semelhantes e em clusters distintos os pontos tenham grandes diferenças. Existem diversos diversos tipos de técnicas de agrupamento, nesse projeto iremos utilizar K-Means, DBSCAN e Fuzzy C-Means.

1. K-Means

K-Means é uma das técnicas mais conhecidas e utilizadas de agrupamento não-supervisionado. É uma técnica baseada em centroides que necessita que seja setado um número de clusters em que os dados serão divididos, e então aleatoriamente - ou com técnicas mais inteligentes - os centroides serão posicionados para o início do agrupamento. A qual irá iterar diversas vezes e em cada iteração será calculado as distâncias dos pontos aos centroides, cada centroide formará um cluster e o centroide mais próximo de um ponto incluirá ele em seu cluster. A cada iteração os centroides serão

Fundamentos

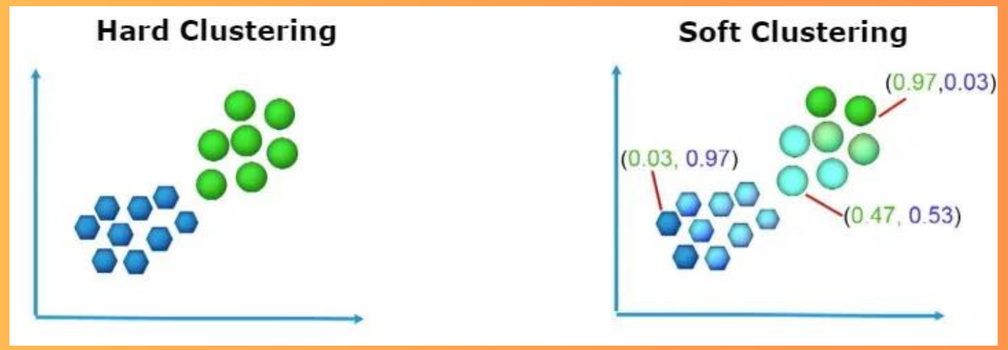
2. DBSCAN



DBSCAN é uma técnica de agrupamento baseada em densidade, na qual agrupa regiões densa em pontos de dados como clusters e trata regiões de pouca densidade como ruído. Essa técnica trabalha com dois parâmetros, epsilon, que diz a distância máxima para que dois pontos sejam considerados vizinhos e pontos mínimos, que diz o mínimo número de pontos vizinhos que é necessário para um ponto ser considerado ponto de dado principal. O DBSCAN funciona calculando a distância de um ponto a todos os outros pontos, caso tenha o número suficiente de vizinhos será marcado como ponto de dado principal, senão será ponto de borda ou de ruído, e ele continua classificando todos os outros pontos. Essa técnica é muito boa em trabalhar com datasets com muito ruído, identifica ruídos facilmente e os clusters podem assumir diversos formatos, já no K-Means os clusters são basicamente esféricos.

Fundamentos

3. Fuzzy C-Means (FCM)



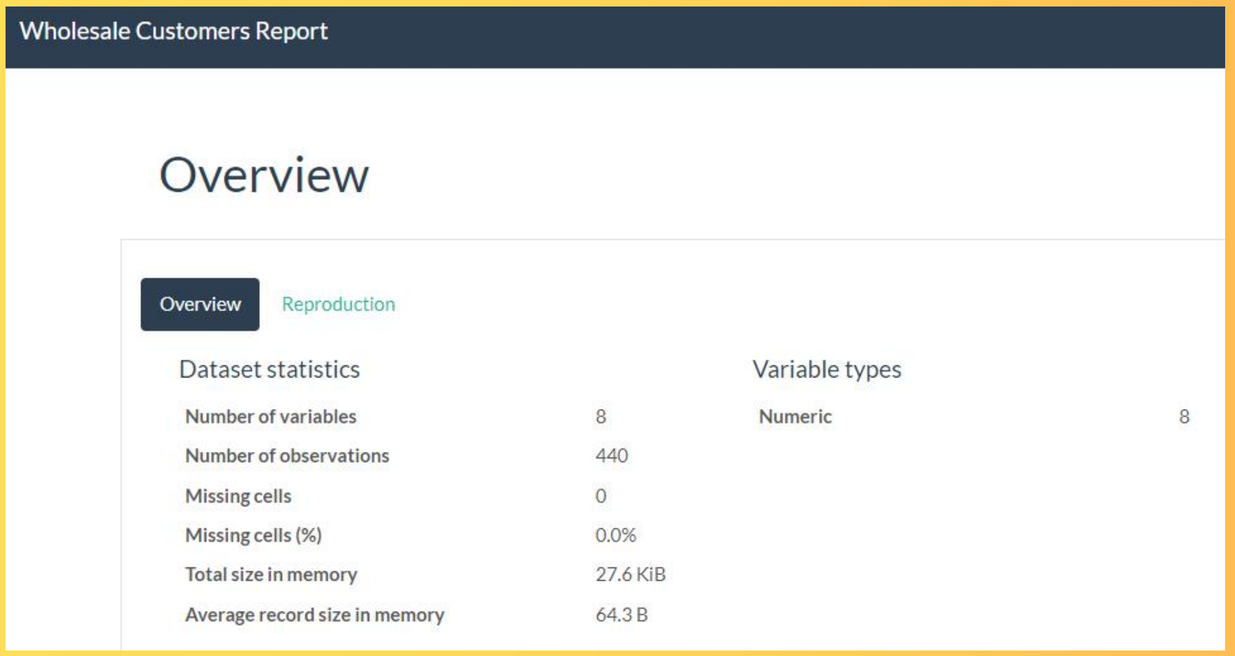
O Fuzzy C-Means é um algoritmo de agrupamento que classifica cada ponto a um mais clusters, dependendo da sua distância dos centroides. Diferen dos algoritmos de agrupamento tradicional, que só associa um ponto a um cluster. Isso permite que o FCM seja muito útil em datasets com sobreposição nos dados. Esse algoritmo trabalha atribuindo aleatoriamente o grau de filiação dos dados aos clusters, e então a posição dos centroides é atualizada de acordo com as filiações dos pontos e então o grau de filiação de cada ponto é atualizado de acordo com a distância a cada cluster. As iterações continuam até um número máximo pré-estabelecido ou até a convergência das posições dos centroides.

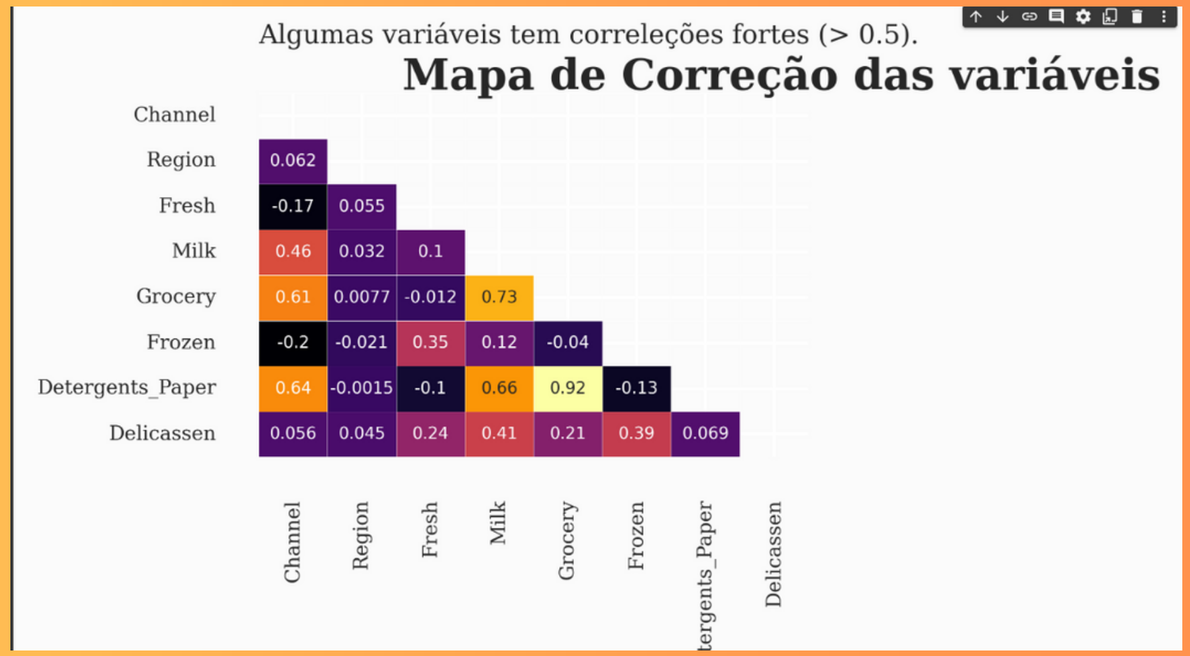
.

Metodologia

Visualização inicial dos dados

- Podemos perceber que a base de dados já estava bastante limpa, sem valores nulos ou indefinidos;
- Algumas variáveis tem forte relação, como Grosseries e Detergents_Paper e Grosseries e Milk;
- Podemos perceber que os produtos da varejista são vendidos uniformemente entre as regiões;
- Varejistas e atacadistas que compram leite também compram mantimentos em geral e papel detergente;





Metodologia

Pré-Processamento

- Como todas as variáveis são relevantes para a clusterização não há necessidade de remover colunas;
- Como a base de dados não tem valores faltosos a imputação não é necessária também;
- Foi realizado a normalização dos valores contínuos da base de dados para as variáveis com grandes valores não enviesarem a clusterização.

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185
...	...	...	...	...	...	...	...	...
435	1	3	29703	12051	16027	13135	182	2204
436	1	3	39228	1431	764	4510	93	2346
437	2	3	14531	15488	30243	437	14841	1867
438	1	3	10290	1981	2232	1038	168	2125
439	1	3	2787	1698	2510	65	477	52

440 rows × 8 columns

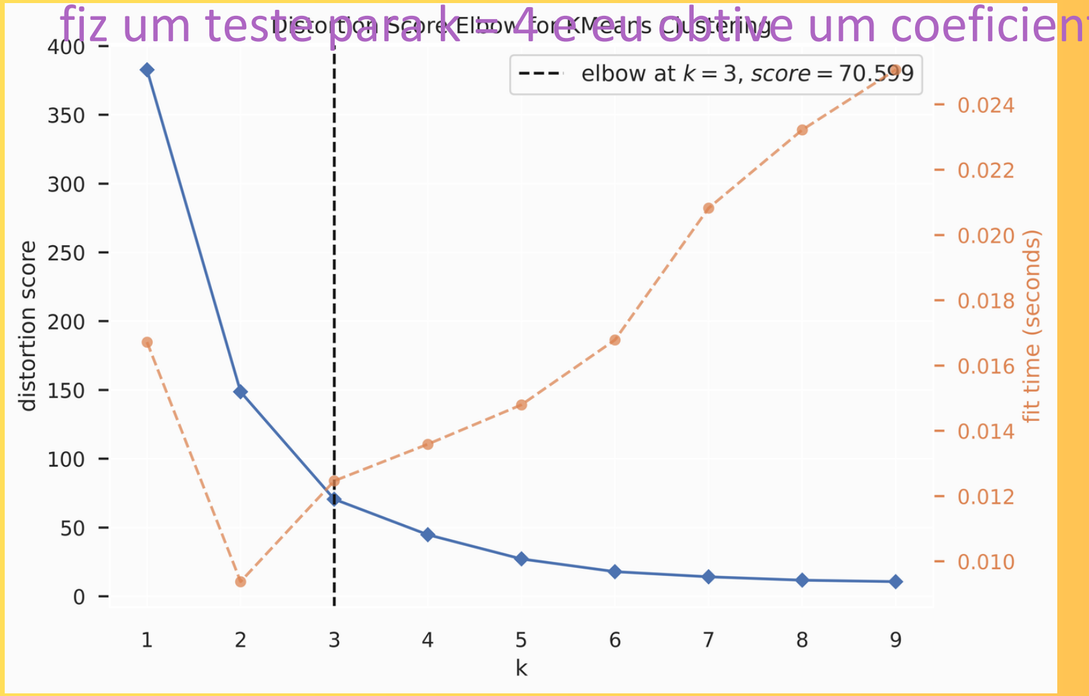
	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	0.112964	0.131378	0.081494	0.003516	0.065496	0.027908
1	2	3	0.062924	0.133473	0.103126	0.028947	0.080657	0.037044
2	2	3	0.056647	0.119840	0.082820	0.039511	0.086119	0.163611
3	1	3	0.118278	0.016273	0.045495	0.105210	0.012418	0.037294
4	2	3	0.201648	0.073607	0.077581	0.064318	0.043525	0.108149
...	...	...	...	...	...	...	...	...
435	1	3	0.264848	0.163964	0.172742	0.215791	0.004458	0.045971
436	1	3	0.349778	0.019470	0.008235	0.074094	0.002278	0.048933
437	2	3	0.129566	0.210727	0.325965	0.007179	0.363509	0.038942
438	1	3	0.091751	0.026953	0.024057	0.017053	0.004115	0.044323
439	1	3	0.024850	0.023103	0.027053	0.001068	0.011683	0.001085

440 rows × 8 columns

Resultados

K-Means

Como citado anteriormente precisamos dizer o valor k, da quantidade de clusters do K-Means, podemos obter o número ótimo de cluster com o método de Elbow, e obtemos o valor de 3. E utilizando esse número de clusters conseguimos um bom valor de Coefieiente de Silhueta, de 0,69 aproximadamente, entretanto eu fiz um teste para k = 4 e eu obtive um coeficiente ainda melhor de 0,72.



k = 3

```
silhouette_avg = metrics.silhouette_score(features, cluster_labels)
print ('silhouette coefficient for the above clutering = ', silhouette_avg)
```

```
silhouette coefficient for the above clutering =  0.6888178169537851
```

```
silhouette_avg = metrics.silhouette_score(features, cluster_labels)
print ('silhouette coefficient for the above clutering = ', silhouette_avg)
```

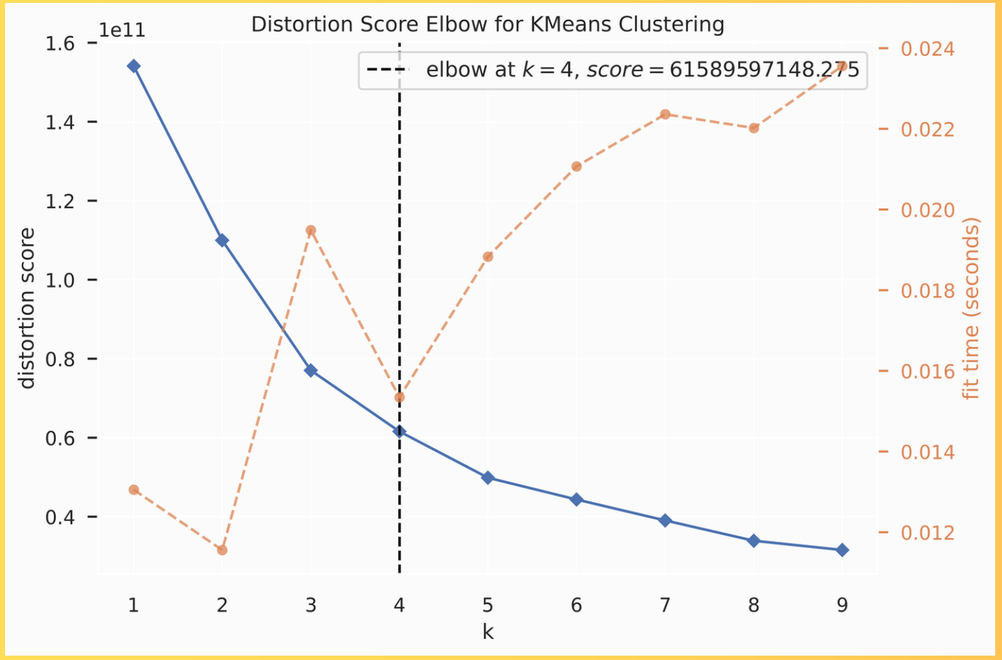
```
silhouette coefficient for the above clutering =  0.7224460716379005
```

k = 4

Resultados

K-Means

Algo interessante é notar a diferença muito profunda no coeficiente de silhueta quando os dados não são normalizados, cerca de 0,24 pontos percentuais de diferença.



k = 3

```
silhouette_avg = metrics.silhouette_score(features_no_normalized, cluster_labels)
print ('silhouette coefficient for the above clutering = ', silhouette_avg)
```

```
silhouette coefficient for the above clutering =  0.4809514242942262
```

```
silhouette_avg = metrics.silhouette_score(features_no_normalized, cluster_labels)
print ('silhouette coefficient for the above clutering = ', silhouette_avg)
```

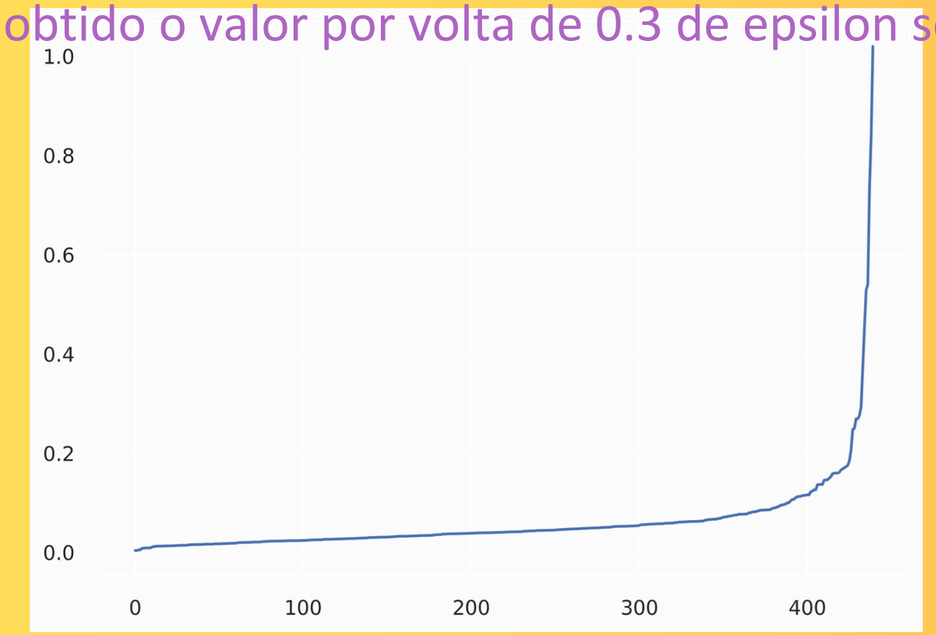
```
silhouette coefficient for the above clutering =  0.4000757807628987
```

k = 4

Resultados

DBSCAN

Assim como o K-Means iremos utilizar uma técnica para obter um valor ótimo do parâmetro do algoritmo, agora precisamos achar o valor para o epsilon, aquele que determina a distância máxima entre vizinhos, e utilizaremos min_samples = 4 como padrão. Usaremos o Knee Method para encontrar o epsilon ideal, o qual tenta procurar a média das distâncias para todo ponto dos seus min_samples vizinhos, no nosso caso 4, e selecionar a distância na qual ocorre a curvatura máxima ou uma mudança brusca. De acordo com o gráfico obtido o valor por volta de 0.3 de epsilon seria muito bom.

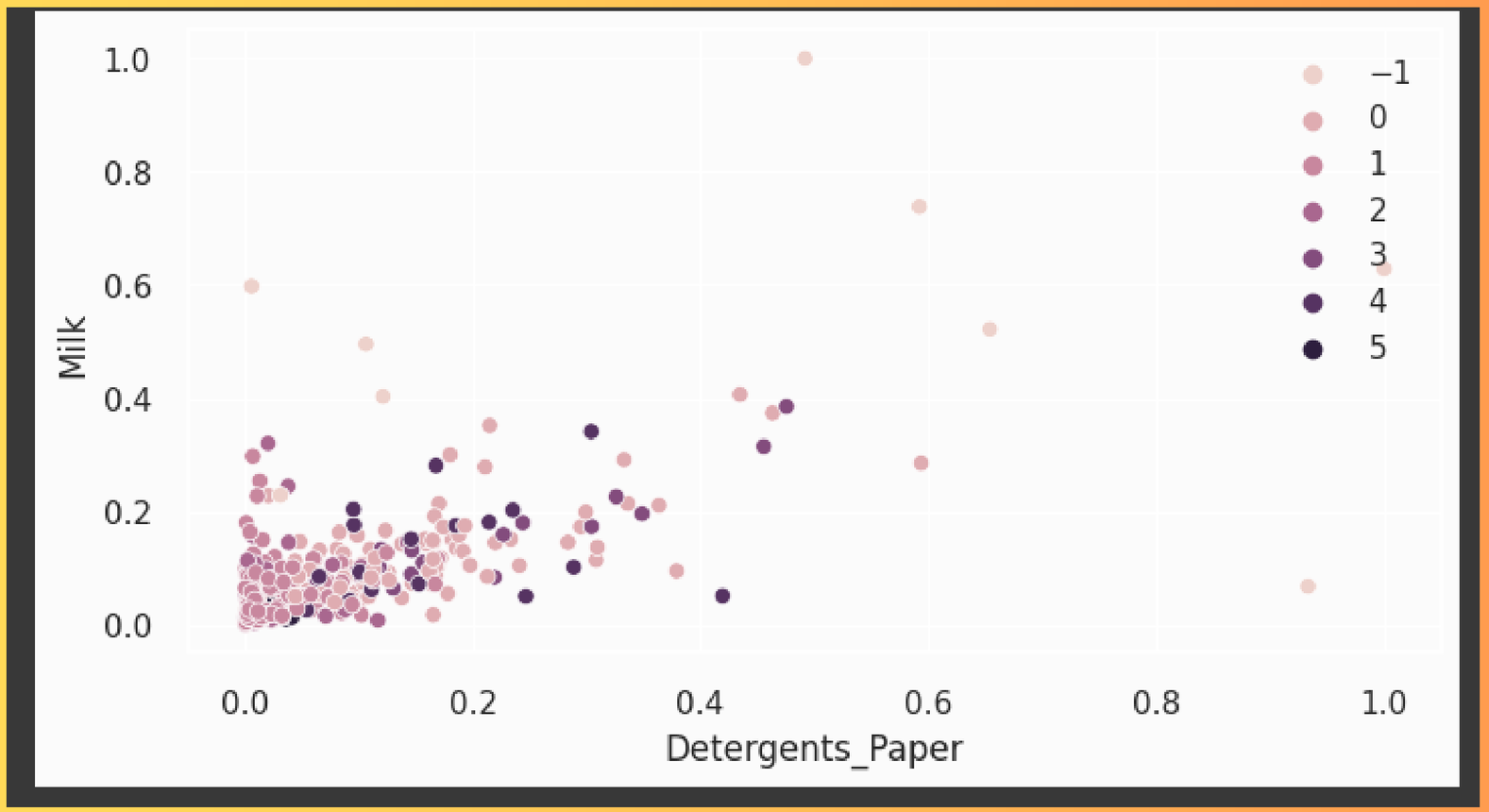


Com 0.3 de epsilon e min_samples = 4 conseguimos um coeficiente de Silhueta muito bom, acima até que o obtido no K-Means, o número de noise points também é lógico dado o dataset escolhido, entretanto houve um aumento no número de clusters.

```
Estimated number of clusters: 6
Estimated number of noise points: 9
Silhouette Coefficient for the Iris Dataset Clusters: 0.79
```

Resultados

DBSCAN



Conclusões

O projeto foi importante para entender na prática as técnicas de visualização de dados, exploração de dados e de clusterização. Percebe-se como é importante tomar algumas decisões desde o início da análise dos dados, como a normalização dos dados, que quando não feita em determinados datasets podem enviesar muito os resultados finais, outro ponto importante é o uso de algoritmos para encontrar valores ótimos de parâmetros, como o Knee Method e Elbow Method, que é um dos principais focos da área de aprendizado de máquina e todos os algoritmos procuram pelos valores que melhor aumente a eficiência dos resultados. Finalmente, é importante notar como diferentes algoritmos para resolução de problemas similares como o de clusterização não estão aí por acaso, e tem efeitos realmente diferentes, cada um podendo ser mais eficiente em contextos mais específicos, tal como foi visto nesse projeto, em que o DBSCAN pareceu ter se saído melhor de acordo com o coeficiente da silhueta, entretando a divisão em 6 clusters pode não ter sido tão bem acertada dado o formato dos pontos de dados.