CLASSIFICAÇÃO DA EMISSÃO NUCLEAR DE GALÁXAIS COM *MACHINE LEARNING* APLICADO A DADOS ESPECTROSCÓPICOS

Arthur Gama Ruiz[1]; Roberto Bertoldo Menezes [2]
[1] Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);
[2] Professor do Instituto Mauá de Tecnologia (IMT).

Resumo. *Os núcleos de galáxias possuem grande importância, já que podem fornecer informações sobre a formação e evolução dessas estruturas. Nesse contexto, a classificação da emissão nuclear de galáxias, que pode ser devida às estrelas ali presentes ou à acreção de matéria em um buraco negro central supermassivo, é uma das análises de maior relevância. Atualmente, essa classificação é feita, com base em dados espectroscópicos, utilizando-se diagramas de diagnóstico, que nada mais são do que gráficos relacionando duas razões de linhas de emissão. Embora muito útil, esse método pode resultar em classificações imprecisas ou dúbias. Nesse trabalho, desenvolvemos uma metodologia, envolvendo técnicas de machine learning, a ser aplicada a dados espectroscópicos, a fim de melhorar a precisão e ainda automatizar, na medida do possível, a classificação da emissão nuclear de galáxias. Ao final, constatou-se que as técnicas de machine learning Random Forest, Decision Trees, Support Vector Machine e XGBOOST são as mais eficientes para esse tipo de análise e podem ser aplicadas utilizando-se não apenas razões de linhas de emissão tradicionais (como [OIII]5007/Hβ, [NII]6583/Hα e [SII](6716+6731)/Hα), mas também outros parâmetros espectroscópicos menos frequentes.*

Introdução.

Galáxias são grandes estruturas gravitacionalmente ligadas, formadas por estrelas, planetas, gás e poeira. As regiões nucleares de algumas galáxias apresentam uma emissão de radiação eletromagnética que não pode ser atribuída apenas às estrelas ali presentes. Os núcleos de tais galáxias são denominados Núcleos Ativos de Galáxias (*Active Galactic Nuclei* – AGNs). O modelo mais aceito atualmente assume que a energia emitida pelos AGNs vem da acreção de matéria em um buraco negro central supermassivo presente no núcleo de cada uma dessas galáxias. Nesse caso, o material espirala e cai em direção ao buraco negro central, dando origem a uma estrutura chamada de disco de acreção. Nesse processo, ocorre a conversão da energia potencial gravitacional do material sendo acretado em energia térmica e energia radiativa, que é emitida, o que dá origem a todos os fenômenos observados nos AGNs. De acordo com as suas propriedades, os AGNs são classificados em diferentes categorias, como *Quasi Stellar Radio Sources* (quasares), rádio-galáxias, galáxias Seyfert, *Low Ionization Nuclear Emission-Line Regions* (LINERs), etc (para maiores detalhes, ver Netzer 2013). Em certas galáxias, a emissão nuclear vem de gás foto ionizado por estrelas quentes e jovens e é classificada como sendo característica de regiões H II. A análise da emissão nuclear de galáxias é de grande importância para estudos focados na formação e evolução de galáxias. Atualmente, a forma mais utilizada para classificar a emissão nuclear de galáxias nas categorias mencionadas anteriormente envolve os chamados diagramas de diagnóstico (ver figura 01 do apêndice), que nada mais são do que gráficos relacionando duas razões de linhas de emissão dos objetos observados. Apesar de ser o mais utilizado, esse método pode, em certos casos, resultar em classificações dúbias ou imprecisas. Isso pode ocorrer, por exemplo, quando diagramas de diagnóstico diferentes resultam em classificações diferentes para um mesmo objeto. Dessa forma, metodologias para

a classificação mais precisa da emissão nuclear de galáxias são muito desejáveis. Esse é o ponto no qual técnicas de *machine learning* se tornam uma opção bastante adequada. Algoritmos de *machine learning* têm sido cada vez mais utilizados na astronomia, devido ao volume e complexidade crescentes dos dados disponíveis. Uma das sub-áreas da astronomia na qual técnicas de *machine learning* têm sido utilizadas envolve a classificação da emissão nuclear de galáxias. Um exemplo é o trabalho de Zhang et al. (2020), que utilizaram o método t-SNE para diferenciar galáxias Seyfert, LINERs e regiões H II. Apesar da aplicação bem-sucedida desse método por esses autores, muitos aprimoramentos ainda são necessários para uma classificação mais precisa da emissão nuclear de galáxias. Nesse projeto, será proposto avaliar os algoritmos de *machine learning* mais eficazes e os parâmetros mais adequados para classificar a emissão nuclear de galáxias em Seyferts, LINERs ou regiões H II, utilizando-se dados espectroscópicos. Pretende-se utilizar os dados públicos disponíveis do survey *Mapping Nearby Galaxies at Apache Point Observatory* (MaNGA – Bundy et al. 2015), o qual está sendo conduzido com o *Sloan Digital Sky Survey* (SDSS), utilizando Unidades de Campo Integral (*Integral Field Units* – IFUs), com diâmetros que variam de 12" a 32" e com uma cobertura espectral de 3600 Å a 10300 Å (com uma resolução espectral de R~2000).

Material e Métodos.

Inicialmente, foram obtidos, a partir do banco de dados citado anteriormente, dados espectroscópicos de uma amostra de galáxias observadas com o survey MaNGA. Nessa etapa inicial, foram coletados apenas os seguintes dados espectroscópicos (todos razões de linhas): [OIII]5007/Hβ, [NII]6583/Hα e [SII](6716+6731)/Hα. Obteve-se um total de 10415 galáxias.

O tratamento desses dados foi feito por meio do software Excel. Em posse dos dados devidamente tratados (ao todo, 5143 objetos), sem linhas nulas ou informações inadequadas (razões de linha com valores tão baixos que o software aproxima para valores nulos), foi possível classificar os núcleos de emissão das galáxias em três tipos, regiões HII, LINER e Seyfert, por meio dos seguintes diagramas de diagnóstico: [OIII]5007/Hβ x [NII]6583/Hα e [OIII]5007/Hβ x [SII](6716+6731)/Hα. Essa classificação inicial foi feita utilizando-se programas escritos na linguagem Python. Estes, por sua vez, consideraram que a razão de linha deveria estar em determinado intervalo real para que o núcleo de emissão obtivesse uma classificação correspondente. Os objetos que possuíram as mesmas classificações segundo estes scripts, foram considerados como tendo "classificação clara". Esta etapa foi muito importante para o projeto, pois, para posterior aplicação do aprendizado de máquina, devem ser fornecidos dados exatos para correto treinamento do modelo.

Em posse dos objetos classificados como tendo "classificação clara", determinaram-se os melhores modelos de *machine learning* a serem aplicados. Dessa forma, a fim de selecioná-los, avaliou-se um parâmetro de validação chamado acurácia. Este, por sua vez, varia de 0 a 1 e mede a assertividade do método, considerando a razão entre os dados previstos corretamente pelo modelo e os dados totais passados como entrada. Ela pode ser calculada da seguinte maneira:

$$Acurácia = \frac{Previsões\ corretas}{Total\ de\ previsões} = \frac{PositivosVerdadeiros + NegativosVerdadeiros}{Total\ de\ previsões},$$

em que:

- Positivos Verdadeiros: o modelo prevê uma classe positiva (refere-se a uma categoria ou a um rótulo atribuído a exemplos com características específicas) e a previsão é correta, ou seja, o exemplo realmente pertence à classe positiva.;
- Negativos Verdadeiros: modelo prevê uma classe negativa (refere-se a uma categoria ou a um rótulo atribuído a exemplos que não possuem uma característica específica).

A seguir, tem-se a lista dos modelos considerados mais adequados nessa etapa do trabalho:

1. *Random Forest:* combina múltiplas árvores de decisão para melhorar a precisão (ver Breiman 2001).
2. *Decision Tree:* utiliza uma estrutura de árvore para tomar decisões. Possui raiz (atributo de entrada), nó (decisão a ser tomada baseada em um atributo) e folha (classe ou valor previsto). Para mais detalhes, ver Quinlan 1986;
3. *Gaussian Mixture Model:* utilizado para representar distribuições de dados complexas, compostas por múltiplas subclasses ou *clusters* (McLachlan 2000).
4. *Support Vector Machine:* busca o hiperplano que melhor separa as classes (Cortes 1995);
5. *XGBOOST (Extreme Gradient Boostin):* combina diversas árvores de decisão com *gradient boosting*, com objetivo de melhorá-las (Chen 2016).

Dentre os modelos supracitados, encontra-se um classificado como não-supervisionado (*Gaussian Mixture Model*), enquanto os demais são supervisionados. Seguem algumas diferenças entre eles:

1. Aprendizado Supervisionado: O algoritmo é treinado com dados rotulados (entrada e saída esperada), com o objetivo de prever a saída a partir de novas entradas. O modelo aprende a mapear entradas para as saídas, baseado nos dados de treinamento. Este tipo tende a ser o mais frequentemente utilizado na astronomia.

2. Aprendizado Não-Supervisionado: O algoritmo usa dados não rotulados, com objetivo de descobrir padrões. Utiliza uma grande variedade de ferramentas estatísticas para diferentes tipos de análises de dados, como *clustering* (que consiste na divisão dos dados analisados em diferentes grupos, com base nas propriedades observadas), redução da dimensionalidade, detecção de *outliers*, entre outros. A partir destas diferenças, nota-se que o algoritmo do aprendizado não-supervisionado não espera por uma saída em si. Dessa maneira, a acurácia não é uma métrica ideal para avaliá-lo, mas foi feita mesmo assim para efeitos comparativos.

É possível notar que não basta utilizar a função accuracy_score(), presente na biblioteca, para os modelos não supervisionados, pois um de seus parâmetros é a previsão do método. Sob esse viés, o processo de validação pode ser feito por outra maneira. É possível visualizar esta parte na Figura 02 do Apêndice, ao final do documento.

Como citado anteriormente, a acurácia é uma razão entre o número de objetos classificados assertivamente e o número total. Portanto, deve-se, como primeiro passo, determinar quantos

objetos o *Gaussian Mixture Model* classificou corretamente. Para isso, é preciso avaliar qual o rótulo representativo de cada um dos *clusters* obtidos com o método. Assim, realiza-se o mapeamento entre os *clusters* (um grupo de objetos de dados que são mais semelhantes entre si do que com objetos de outros *clusters*) e rótulos (uma categoria ou valor alvo que é atribuído a um grupo de dados para fins de classificação ou regressão, ou seja, é o "resultado" que se deseja prever para determinado objeto), salvam-se os rótulos mais frequentes em um dicionário para, posteriormente, verificar se é o rótulo mais comum deste *cluster* (de acordo com a entrada). Em outras palavras, o rótulo mais frequente do *cluster* é tomado como sendo o representativo. Incrementa-se a variável "correct_count", caso a entrada e a predição do modelo possuam o mesmo rótulo. Por fim, concluiu-se que o *Gaussian Mixture Model* classificou corretamente 1280 objetos, de um total de 2721. Enfim, a acurácia é dada por:

$$\frac{1280}{2721} = 0{,}470$$

Além disso, é possível visualizar a disposição dos *clusters* em um gráfico 3-D, em que cada ponto corresponde a uma galáxia. As cores representam o *cluster* ao qual aquelas galáxias foram atribuídas pelo modelo não-supervisionado. O Eixo X representa o primeiro componente principal (*principal component*) dos dados escalados. Este componente é uma combinação linear dos parâmetros originais (razões de linha) que captura a maior parte da variância nos dados. O Eixo Y apresenta o segundo componente principal dos dados escalados. Este componente é ortogonal ao primeiro componente principal e captura a segunda maior parte da variância nos dados. Já o Eixo Z representa o terceiro componente principal dos dados escalados. Este componente é ortogonal aos dois primeiros componentes principais e captura a terceira maior parte da variância nos dados.

Em resumo, os eixos X, Y e Z no gráfico 3D representam os três primeiros componentes principais dos dados de teste escalados. Esses componentes principais são combinações lineares dos parâmetros originais que capturam a maior parte da variância nos dados. O gráfico mostra como os dados estão distribuídos em um espaço tridimensional definido por esses componentes principais, e os *clusters* identificados pelo *Gaussian Mixture Model* são representados por cores diferentes. Vê-se tal gráfico na Figura 03 do Apêndice.

Outras maneiras de validação de um modelo não-supervisionado são: *Silhouette Score* e *Davies-Bouldin Index*. O primeiro varia de -1 a 1 e indica a qualidade de atribuição de determinada amostra a um *cluster*. Já o segundo avalia a qualidade de agrupamento de dados, ou seja, os algoritmos de agrupamento. Ele é calculado a partir da média da razão entre as distâncias dos pontos dentro de um *cluster* e as distâncias dos pontos aos outros *clusters*. *Clusters* bons são aqueles com baixa variação interna e alta separação entre si. Sob essa óptica, quanto mais próximo de zero estiver esse índice, melhor a qualidade dos agrupamentos. Na Figura 04 do apêndice, para melhor compreensão, explicita-se a aplicação destas duas métricas

No geral, os demais modelos (todos supervisionados) não diferem essencialmente no que se refere às suas aplicações, mas sim em seus hiper parâmetros. Ao serem manipulados, os hiper parâmetros são capazes de otimizar o aprendizado, combinando-os para obter uma maior acurácia. A figura 05, vista no apêndice, exemplifica uma busca por melhores hiper parâmetros, aplicada ao aprendizado *Decision Tree*.

Após a utilização dos modelos de *machine learning* tomando-se as razões de linhas de emissão como parâmetros, para meios comparativos, torna-se viável a aplicação deles a novos

parâmetros. Com isso, houve uma nova etapa de coleta, a qual consistiu na obtenção dos valores dos seguintes parâmetros*: MaNGAId* (ID do MaNGA, identificador), *RAJ2000* (ascensão reta do objeto em coordenadas equatoriais), *DEJ2000* (declinação do objeto em coordenadas equatoriais), *EWHac* (largura equivalente de Hα), *OH-O3N2c* (abundância de oxigênio usando o calibrador O3N2 na abertura central), *logsigMc* (densidade superficial de massa estelar na abertura central), *VdispHac* (dispersão da velocidade Hα) e *Vdispsc* (dispersão de velocidades, uma medida da largura da linha de emissão). O cálculo de *EWHac* é dado por:

$$W_\lambda = \int \left(1 - \frac{F_\lambda}{F_0}\right) d\lambda,$$

em que:
-F_0: fluxo do contínuo, representa o fluxo de radiação eletromagnética que incide sobre o instrumento de medida, fora de qualquer linha espectral específica;
- F_λ representa o fluxo através de toda a faixa de comprimento de onda em que se deseja calcular a largura equivalente.

Vale ressaltar que as duas etapas de aquisição de dados foram feitas com dois diferentes catálogos do site do VizieR, mas ambos advêm do survey MaNGA.

Nesta etapa, os mesmos modelos de aprendizado de máquina citados anteriormente foram aplicados aos novos parâmetros, agrupados três a três. Após isso, estes parâmetros foram misturados com as razões de linhas de emissão.

Com intuito de finalizar os testes com *machine learning*, testou-se uma combinação de dois parâmetros adquiridos na segunda coleta de dados com uma razão de linha de emissão.

Resultado e Discussão

Nessa seção, são abordados os modelos de aprendizado de máquina, assim como suas respectivas acurácias e seus dados de entrada. A lista a seguir mostra a acurácia de cada método, considerando apenas as razões de linha [OIII]5007/Hβ, [NII]6583/Hα e [SII](6716+6731)/Hα:

1. *Random Forest*: 0,970
2. *Decision Tree*: 0,954
3. *Gaussian Mixture Model*: 0,470
4. *Support Vector Machine*: 0,975
5. *XGBOOST*: 0,980

Aplicou-se também os modelos aos parâmetros da segunda coleta, agrupados três a três. Em suma, não houve diferença significativa das acurácias entre os modelos, estando em torno de 0,905. Os demais testes desconsideraram o modelo *Gaussian Mixture Model,* já que este não apresentou bons resultados com as razões de linha.

Utilizando os parâmetros *EWHac*, *OH-O3N2c*, *Vdispsc*:

1. *Random Forest*: 0,905
2. *Decision Tree*: 0,904
3. *Support Vector Machine*: 0,906
4. *XGBOOST:* 0,878

Utilizando os parâmetros *logsigMc*, *VdispHac*, *Vdispsc*:

1. *Random Forest*: 0,903
2. *Decision Tree*: 0,906
3. *Support Vector Machine*: 0,902
4. *XGBOOST*: 0,878

Utilizando os parâmetros *EWHac*, *VdispHac*, *Vdispsc*:

1. *Random Forest*: 0,905
2. *Decision Tree*: 0,904
3. *Support Vector Machine*: 0,901
4. *XGBOOST*: 0,900

Com os seguintes parâmetros: *OH-O3N2c*, *VdispHac*, *Vdispsc*:

1. *Random Forest*: 0,906
2. *Decision Tree*: 0,903
3. *Support Vector Machine*: 0,901
4. *XGBOOST*: 0,880

Ademais, os mesmos testes foram feitos com dois parâmetros (*EWHac* e *OH-O_3N_2*) e duas razões de linha ([NII]6583/Hα e [OIII]5007/Hβ). Novamente, as acurácias dos métodos não destoaram consideravelmente entre si, podendo ser vistas a seguir:

1. *Random Forest*: 0,906
2. *Decision Tree*: 0,903
3. *Support Vector Machine*: 0,907
4. *XGBOOST*: 0,893

Posteriormente, com todos os parâmetros da segunda coleta e uma das razões de linhas de emissão, obteve-se a seguinte lista de resultados.

Utilizando os parâmetros *EWHac*, *OH-O_3N_2*, *logsigMc*, *VdispHac*, *Vdispsc* e [NII]6583/Hα:

1. *Random Forest*: 0,900
2. *Decision Tree*: 0,897
3. *Support Vector Machine*: 0,899
4. *XGBOOST*: 0,896

Utilizando os parâmetros *EWHac*, *OH-O_3N_2*, *logsigMc*, *VdispHac*, *Vdispsc* e [OIII]5007/Hβ:

1. *Random Forest*: 0,901
2. *Decision Tree*: 0,888
3. *Support Vector Machine*: 0,900

4. *XGBOOST*: 0,898

Utilizando os parâmetros *EWHac*, *OH-O_3N_2*, *logsigMc*, *VdispHac*, *Vdispsc* e [SII](6716+6731)/Hα:

1. *Random Forest*: 0,899
2. *Decision Tree*: 0,894
3. *Support Vector Machine*: 0,869
4. *XBGOOST*: 0,898

Alguns modelos obtiveram os mesmos valores de acurácia por conta de arredondamento para três algarismos significativos.

Conclusões

Em suma, pode-se concluir que os métodos de *machine learning Random Forest*, *Decision Trees*, *Support Vector Machine* e *XGBOOST*, aplicados tomando-se como parâmetros apenas as razões de linhas de emissão [OIII]5007/Hβ, [NII]6583/Hα e [SII](6716+6731)/Hα, apresentaram acurácias extremamente elevadas (superiores a 95%). Isso está de acordo com o esperado fisicamente, já que a análise de diagramas de diagnóstico com razões de linhas de emissão representa a principal forma para classificar a emissão nuclear de galáxias. No entanto, esses resultados indicam que o uso de uma técnica de *machine learning* com as três razões de linhas simultaneamente mostrou-se um método eficaz e muito mais prático para esse tipo de análise. Por outro lado, o uso das técnicas das *machine learning* mencionadas tomando-se os parâmetros espectroscópicos *EWHac*, *OH-O_3N_2*, *logsigMc*, *VdispHac* e *Vdispsc* (que usualmente não são utilizados para a classificação da emissão nuclear de galáxias), ou mesmo combinações desses parâmetros com uma ou duas razões de linhas de emissão, também se mostrou um método eficaz (com acurácias próximas a 90%) para essa análise. Assim, a principal conclusão desse trabalho é que as técnicas de *machine learning* citadas têm o potencial de contribuir significativamente em estudos astronômicos envolvendo a classificação da emissão nuclear de galáxias, os quais podem ser realizados não apenas com as tradicionais razões de linhas de emissão, mas também com parâmetros espectroscópicos menos usuais, como *EWHac*, *OH-O_3N_2*, *logsigMc*, *VdispHac* e *Vdispsc*.

Referências bibliográficas

Bundy, K., Bershady, M. A., Law, D. R., Yan, R., Drory, N., et al. 2015, ApJ, 798, 7
Ishida, E. E. O. 2019, NatAs, 3, 680
Netzer, H. 2013, The Physics and Evolution of Active Galactic Nuclei, Cambridge, UK: Cambridge University Press
Zhang, X., Feng, Y., Chen, H. & Yuan, Q. 2020, ApJ, 905, 97m
Breiman, L. (2001). "Random Forests". Machine Learning, 45(1), 5-32.
Quinlan, J. R. (1986). "Induction of Decision Trees". Machine Learning, 1(1), 81-106.
Cortes, C., & Vapnik, V. (1995). "Support-Vector Networks". Machine Learning, 20(3), 273-297.
Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System". Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
McLachlan, G. J., & Peel, D. (2000). "Finite Mixture Models."

Apêndice

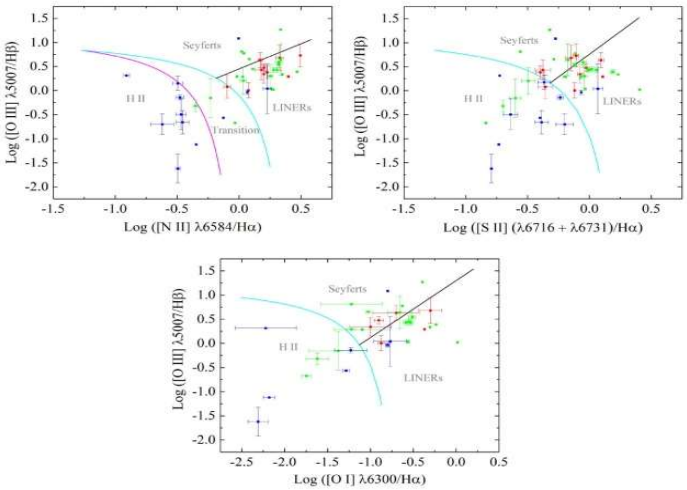


Figura 01-Exemplo de diagrama de diagnóstico

```python
from collections import Counter
from collections import defaultdict

# Encontre o mapeamento entre clusters e rótulos
cluster_mapping = defaultdict(list)
for i, label in enumerate(Y_test):
  cluster_mapping[Y_pred[i]].append(label)

# Dicionário para armazenar o rótulo mais comum para cada cluster
cluster_labels = {}

# Iteração sobre os rótulos de teste
for cluster, labels in cluster_mapping.items():
  cluster_labels[cluster] = Counter(labels).most_common(1)[0][0]

# Conte quantos objetos foram classificados corretamente
# Verifica se o rótulo corresponde ao rótulo mais comum do cluster
# ao qual foi atribuído. Em caso afirmatuvo, incrementa a variável correct_count

correct_count = 0
for i, label in enumerate(Y_test):
  if label == cluster_labels[Y_pred[i]]:
    correct_count += 1

print(f"Número de objetos classificados corretamente: {correct_count}")
```

```
Número de objetos classificados corretamente: 1280
```

Figura 02-Determinação da métrica acurácia para um modelo de aprendizado não-supervisionado

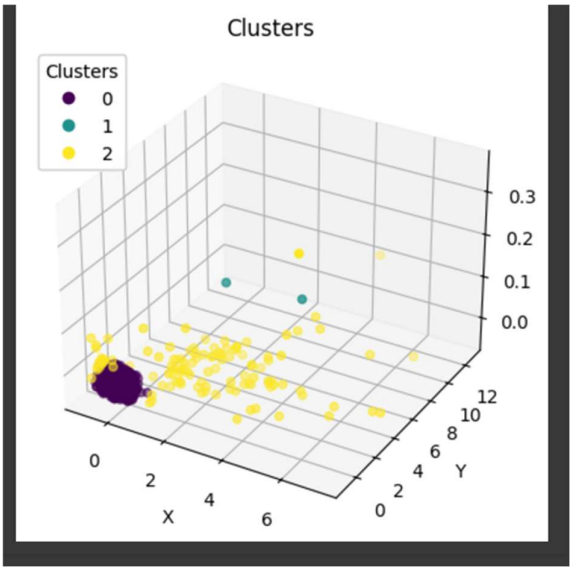
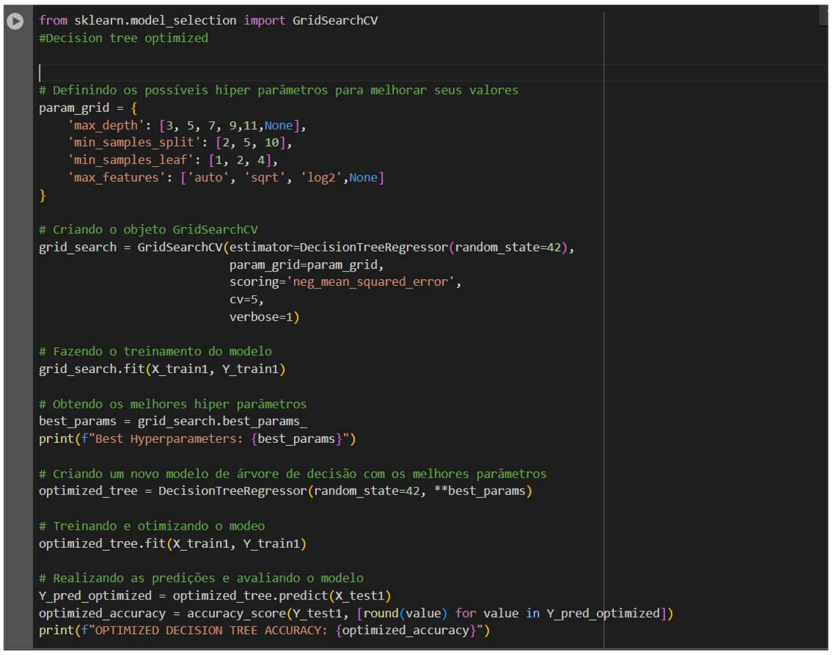


Figura 03 – Disposição dos clusters

```python
#Aplicação do método não supervisionado (Gaussian Mixture Model)
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score
from sklearn.metrics import davies_bouldin_score
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = GaussianMixture(n_components=3, random_state=0)
model.fit(X_train_scaled)                              #treinando o modelo
Y_pred = model.predict(X_test_scaled)                  #fazendo a predição

#Métricas como acurácia podem não ser adequadas para avaliar o modelo (não supervisionado). Utilizaremos outras métricas, tais quais:

# 1- Silhouette Score: varia e -1 a 1, sendo que valores mais próximos de 1 indicam amostras bem agrupadas nos seus respectivos clusters
#valores próximos de -1 podem indicar atribuições errônias da amostra ao cluster

silhouette_avg = silhouette_score(X_test_scaled, Y_pred)
print(f"Silhouette score: {silhouette_avg}")

# 2- Davies-Bouldin Index:
dbindex = davies_bouldin_score(X_test, Y_pred)
print(f"Davies-Bouldin Index: {dbindex}")
```

Figura 04 – Métrica de validação *Silhouette Score*



```python
from sklearn.model_selection import GridSearchCV
#Decision tree optimized


# Definindo os possíveis hiper parâmetros para melhorar seus valores
param_grid = {
    'max_depth': [3, 5, 7, 9,11,None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2',None]
}

# Criando o objeto GridSearchCV
grid_search = GridSearchCV(estimator=DecisionTreeRegressor(random_state=42),
                           param_grid=param_grid,
                           scoring='neg_mean_squared_error',
                           cv=5,
                           verbose=1)

# Fazendo o treinamento do modelo
grid_search.fit(X_train1, Y_train1)

# Obtendo os melhores hiper parâmetros
best_params = grid_search.best_params_
print(f"Best Hyperparameters: {best_params}")

# Criando um novo modelo de árvore de decisão com os melhores parâmetros
optimized_tree = DecisionTreeRegressor(random_state=42, **best_params)

# Treinando e otimizando o modeo
optimized_tree.fit(X_train1, Y_train1)

# Realizando as predições e avaliando o modelo
Y_pred_optimized = optimized_tree.predict(X_test1)
optimized_accuracy = accuracy_score(Y_test1, [round(value) for value in Y_pred_optimized])
print(f"OPTIMIZED DECISION TREE ACCURACY: {optimized_accuracy}")
```

Figura 05 – Otimização de hiper parâmetros