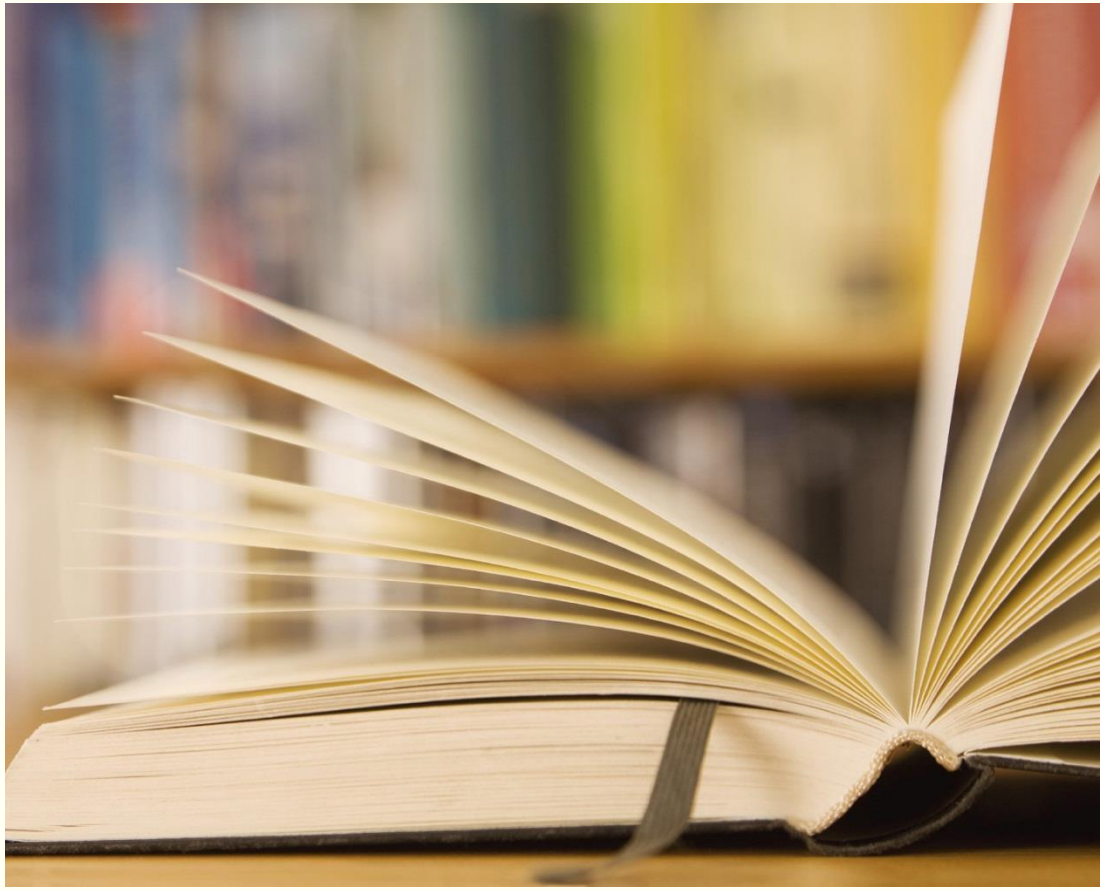
ANÁLISE DE DADOS USANDO APRENDIZAGEM MÁQUINA

Curricular unit: Intelligent Systems for Bioinformatics



DADOS E PROBLEMÁTICA

Dados

- Conjunto de dados GDSC1;
- Conjunto de dados contém o IC50 de laboratório para 208 fármacos em 1000 linhas de células cancerígenas;
- Utilizado para conceber modelos capazes de prever a resposta aos fármacos
- Conceber um modelo que possa prever a resposta ao fármaco e encontrar o melhor fármaco para tratar determinado doente;
- Expressão genética normalizada RMD foi utilizada para as linhas de cancro:
- SMILES utilizadas para os fármacos;
- Y é o IC50 normalizado por log.

Problemática

Estudos recentes demonstraram que as alterações nos genomas do cancro influenciam a resposta clínica às terapias anticancerígenas;

Hoje em dia, as alterações genómicas são utilizadas como biomarcadores moleculares para identificar os doentes com maior probabilidade de beneficiar de um tratamento;

No entanto, muitos medicamentos contra o cancro em desenvolvimento ou já em uso não foram associados a um marcador genómico;

Avanços na sequenciação de ADN permitiram associar a complexidade genómica do cancro à sensibilidade a medicamentos, utilizando linhas celulares cancerígenas como plataformas para descoberta de biomarcadores e desenvolvimento de novas terapias.

Genomics of Drug sensitivity in Cancer (GDSC)

Base de dados concebida para facilitar o estudo e a compreensão das caraterísticas moleculares que influenciam a resposta aos medicamentos em linhas celulares de cancro;

Contém conjuntos de dados sobre a sensibilidade aos medicamentos em células cancerosas;

Associa esses dados a informações genómicas pormenorizadas para facilitar a descoberta de biomarcadores moleculares da resposta aos medicamentos.

Exploração inicial do dataset

- Visualização do dataset

	Drug_ID	Drug	Cell Line_ID	Cell Line	Y
0	Erlotinib	COCCOC1=C(C=C2C(=C1)C(=NC=N2)NC3=CC=CC(=C3)C#C...	MC-CAR	[3.23827250519154, 2.98225419469807, 10.235490...	2.395685
1	Erlotinib	COCCOC1=C(C=C2C(=C1)C(=NC=N2)NC3=CC=CC(=C3)C#C...	ES3	[8.690197905033282, 3.0914731119366, 9.9924871...	3.140923

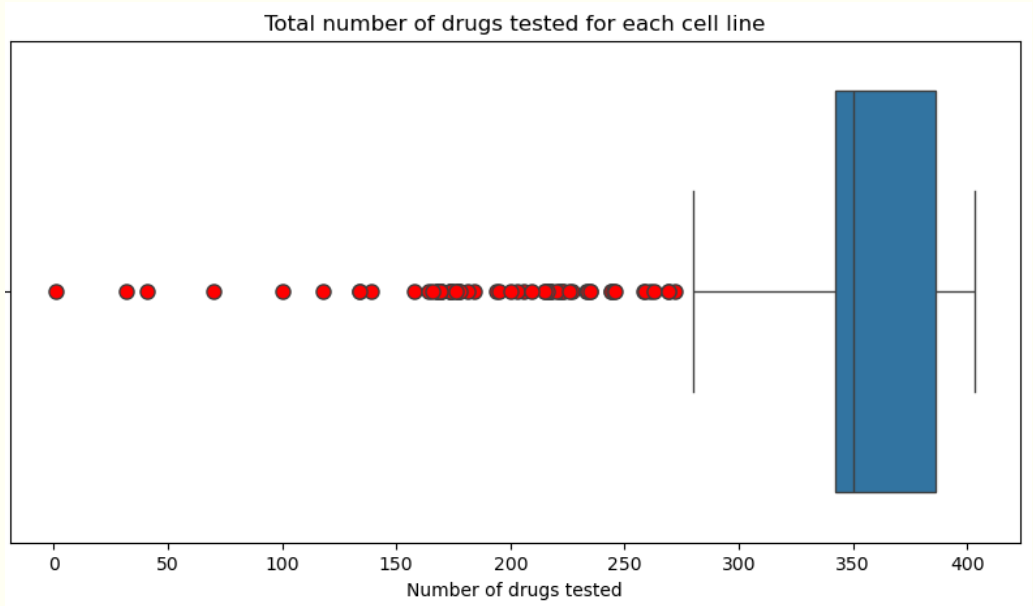
- Informações sobre a linha celular

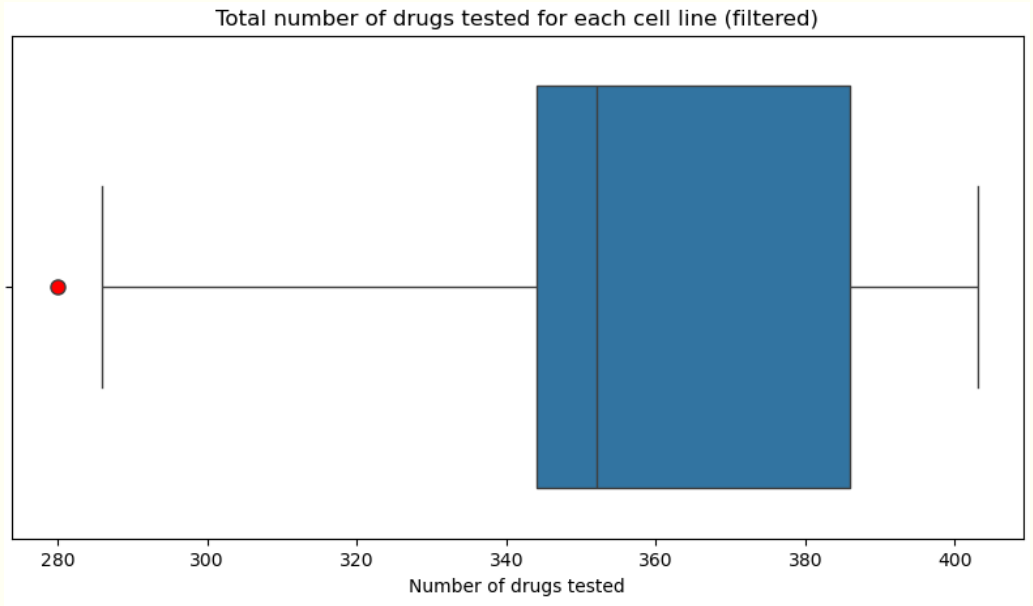
	Cell line Name	Model ID	COSMIC ID	TCGA Classfication	Tissue	Tissue sub-type	Datasets	number of drugs
0	22RV1	SIDM00499	924100	PRAD	urogenital_system	prostate	GDSC1	353
2	23132-87	SIDM00980	910924	STAD	digestive_system	stomach	GDSC1	344

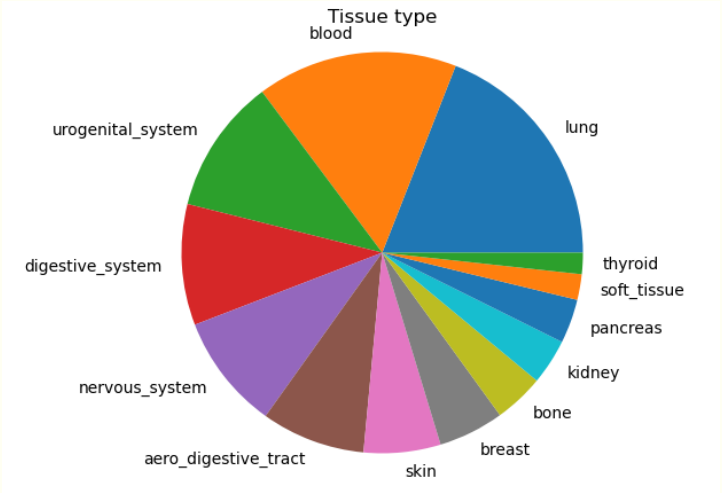
- Informação sobre medicamentos

	Name	Synonyms	Targets	Target pathway	PubCHEM
0	(5Z)-7-Oxozeaenol	5Z-7-Oxozeaenol, LL-Z1640-2	TAK1	Other, kinases	9863776
4	5-Fluorouracil	5-FU	Antimetabolite (DNA & RNA)	Other	3385

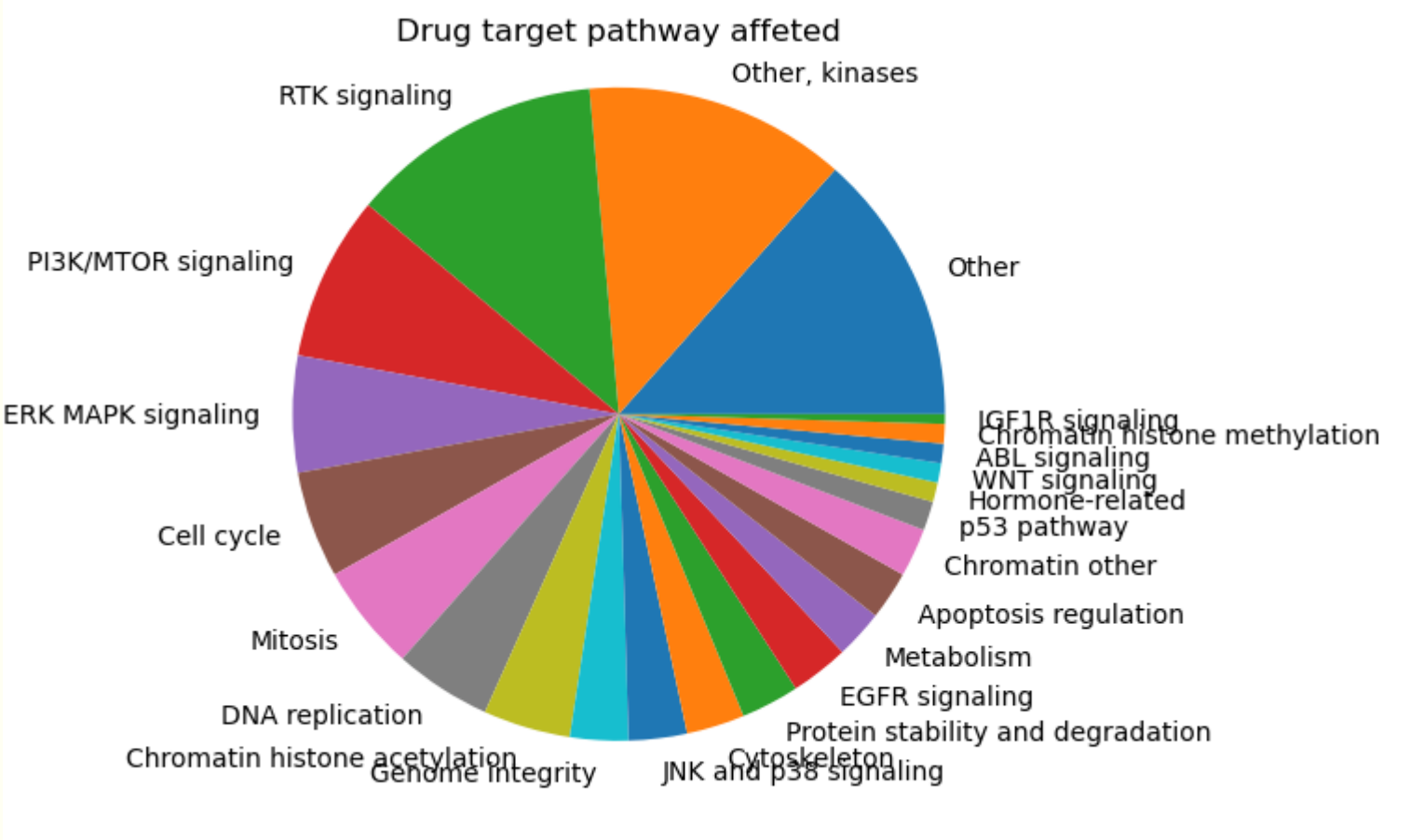
Exploração e preparação do dataset: Exploração de dados de linhas celulares







Exploração e preparação do dataset: Exploração de dados sobre medicamentos



Análise não supervisionada

Redução de dimensionalidade:

Principal Component Analysis (PCA);

t-SNE;

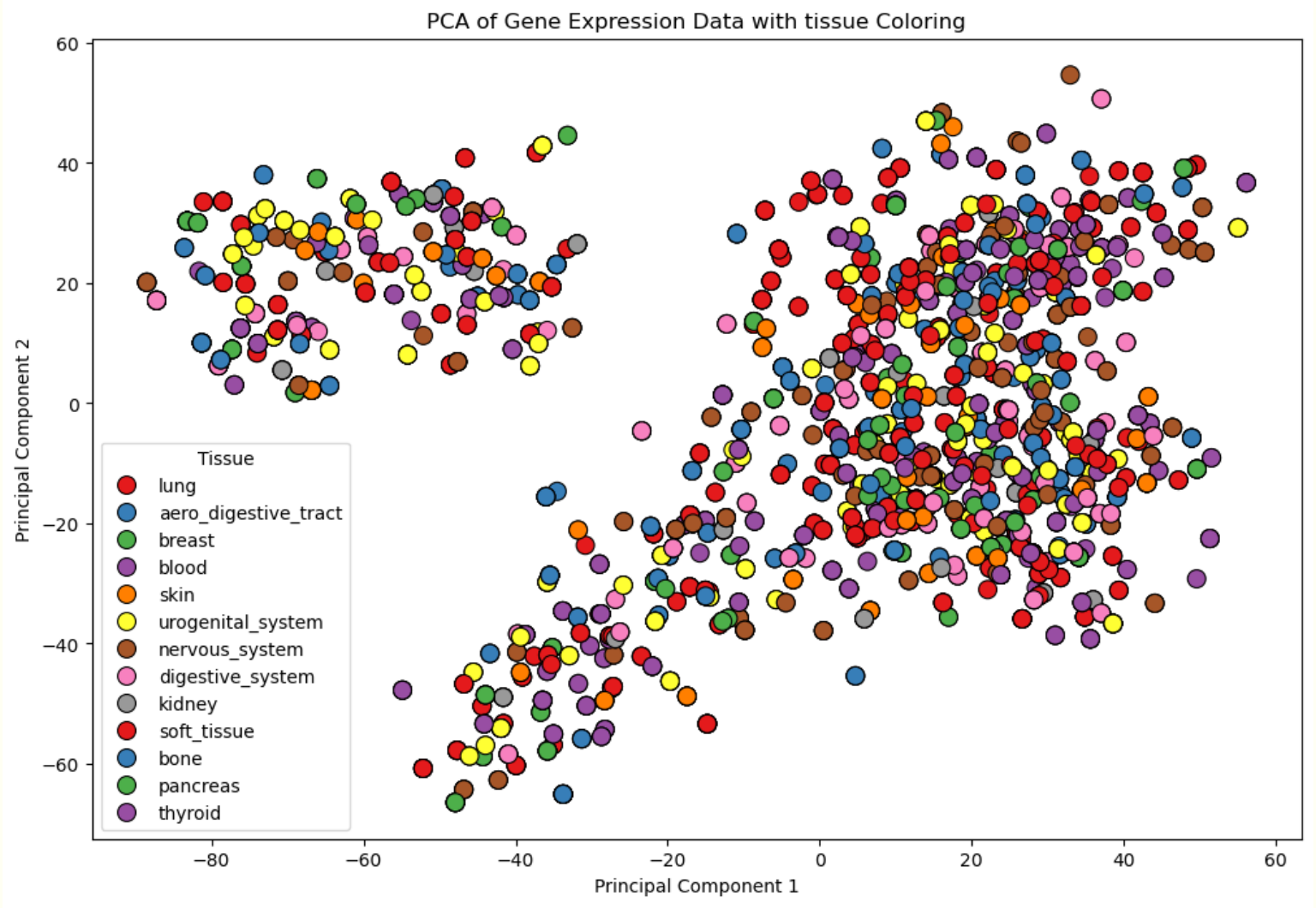
Autoencoders (Unsupervised Neural Networks).

Métodos de clustering:

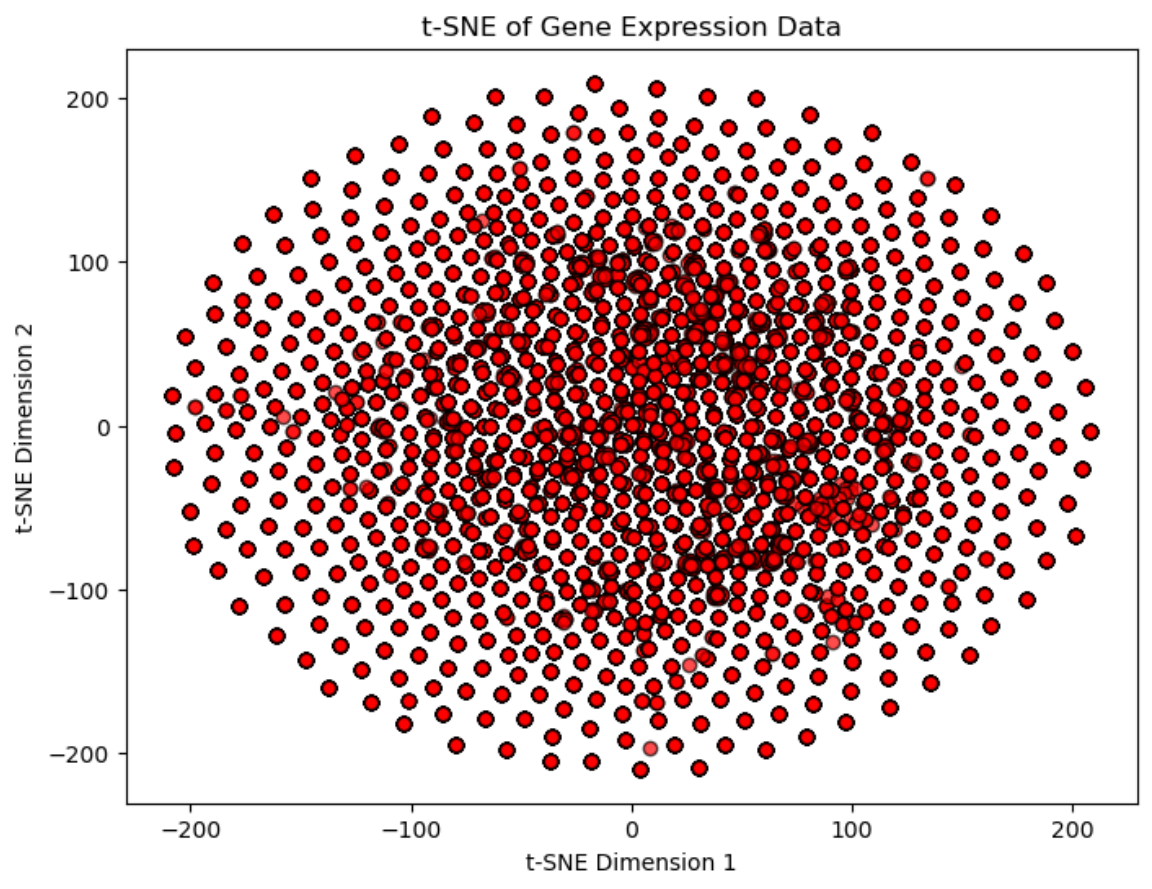
K-means;

Hierárquico.

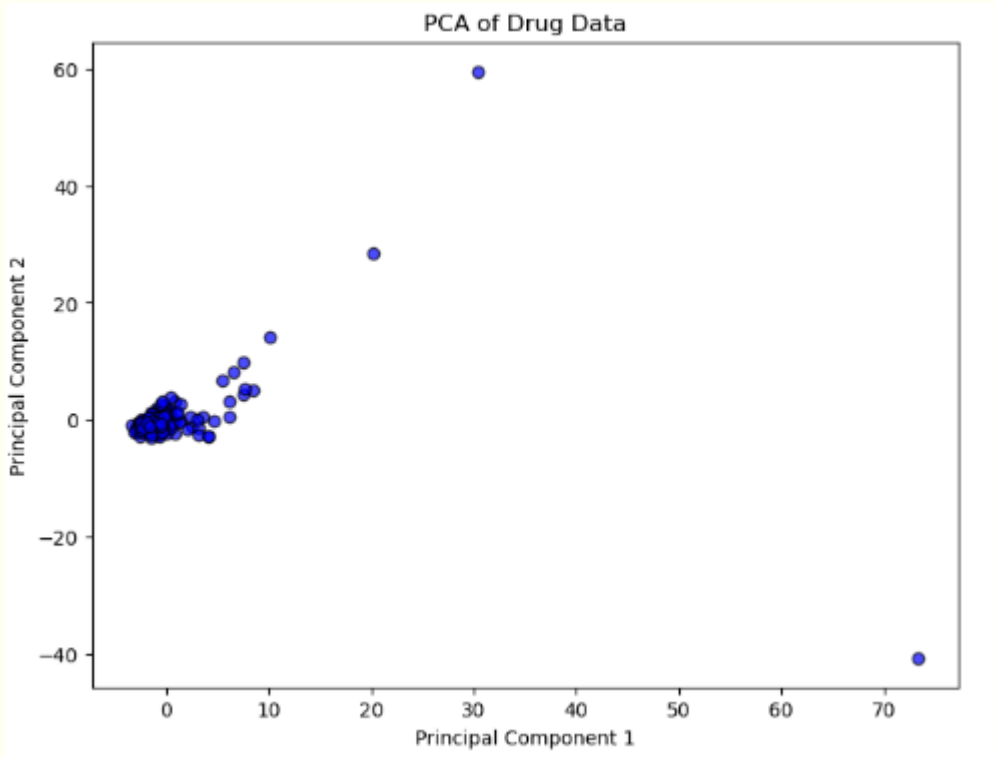
Redução de dimensionalidade: PCA

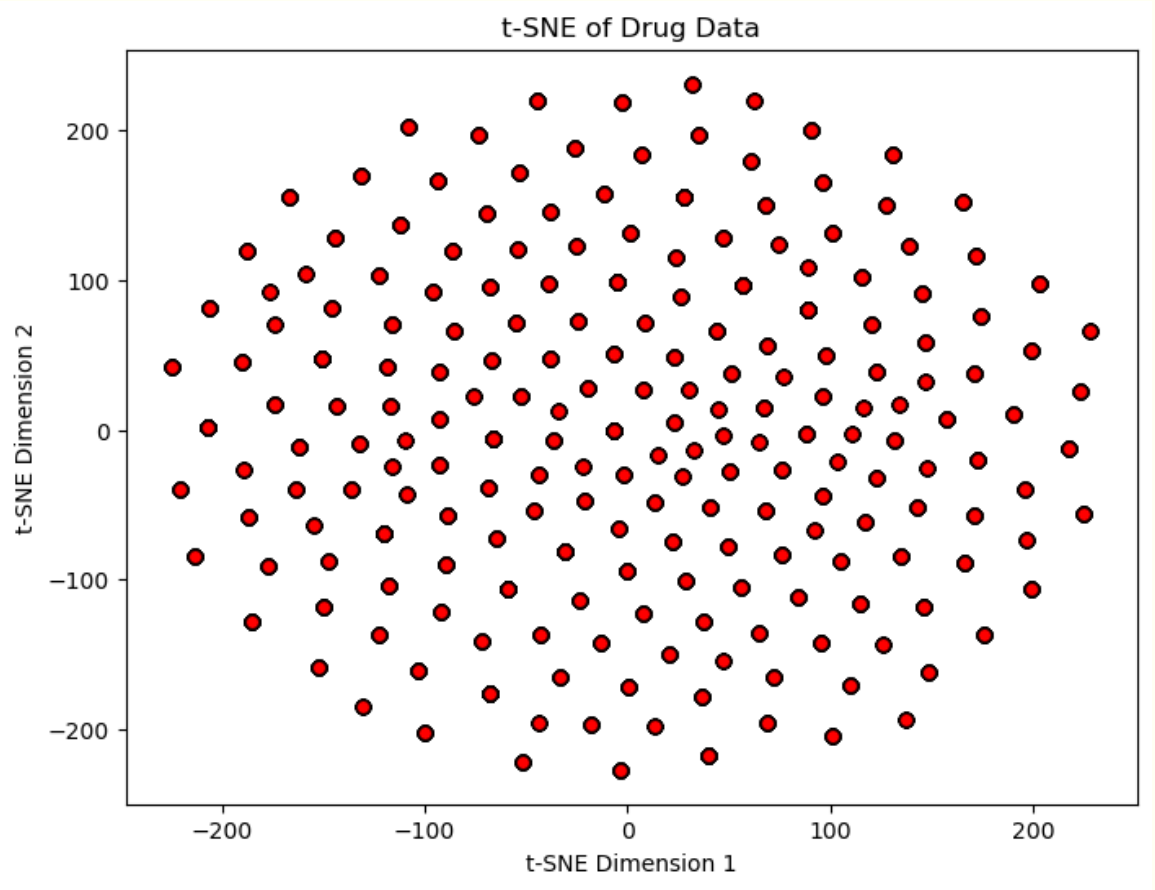


Redução de dimensionalidade: t-SNE

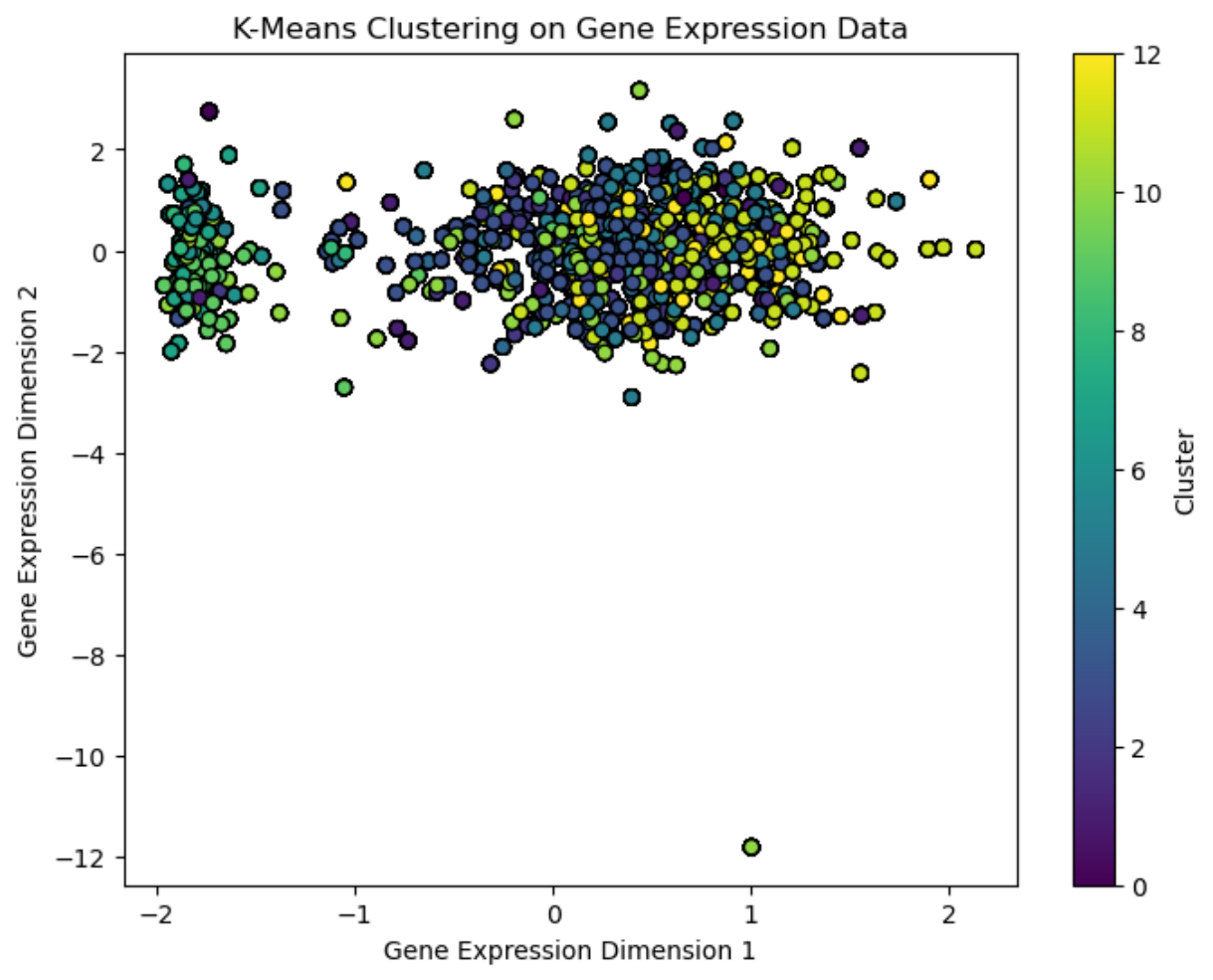


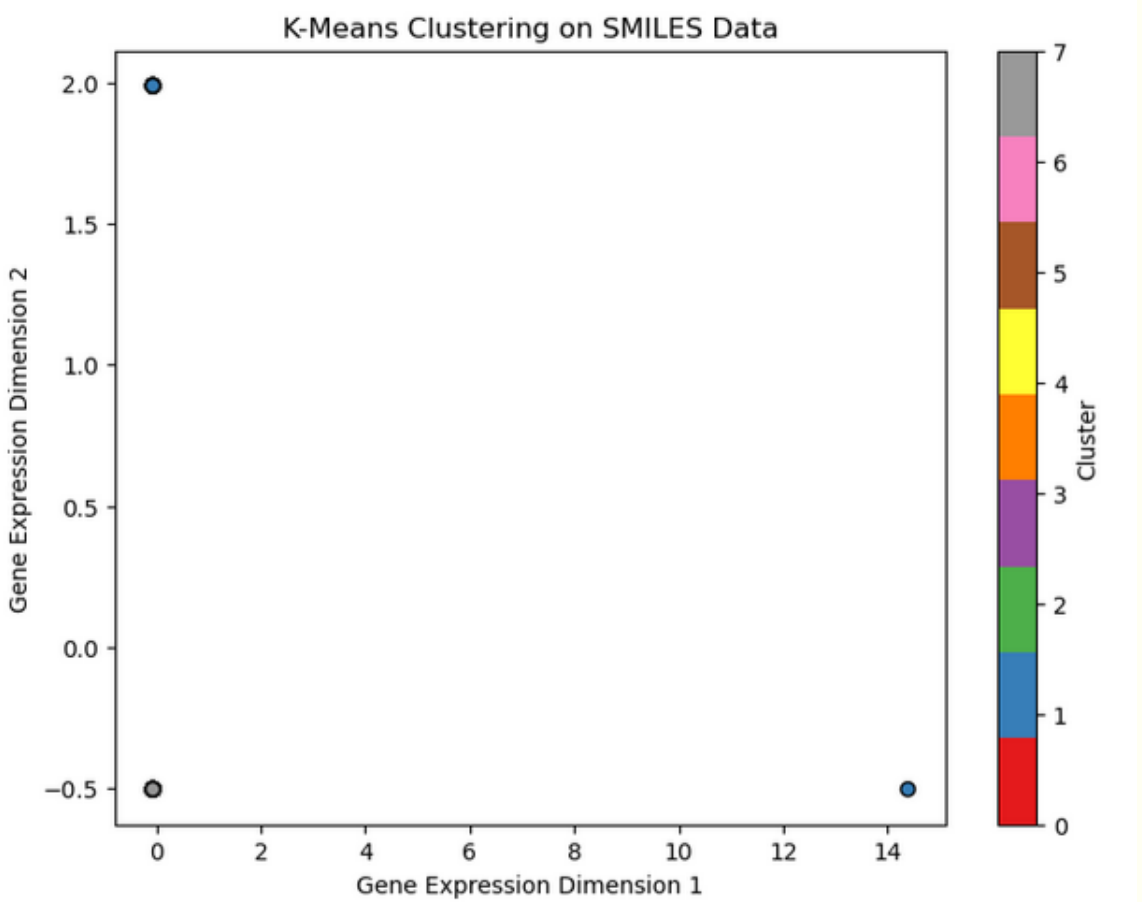
Redução de dimensionalidade: PCA e t-SNE



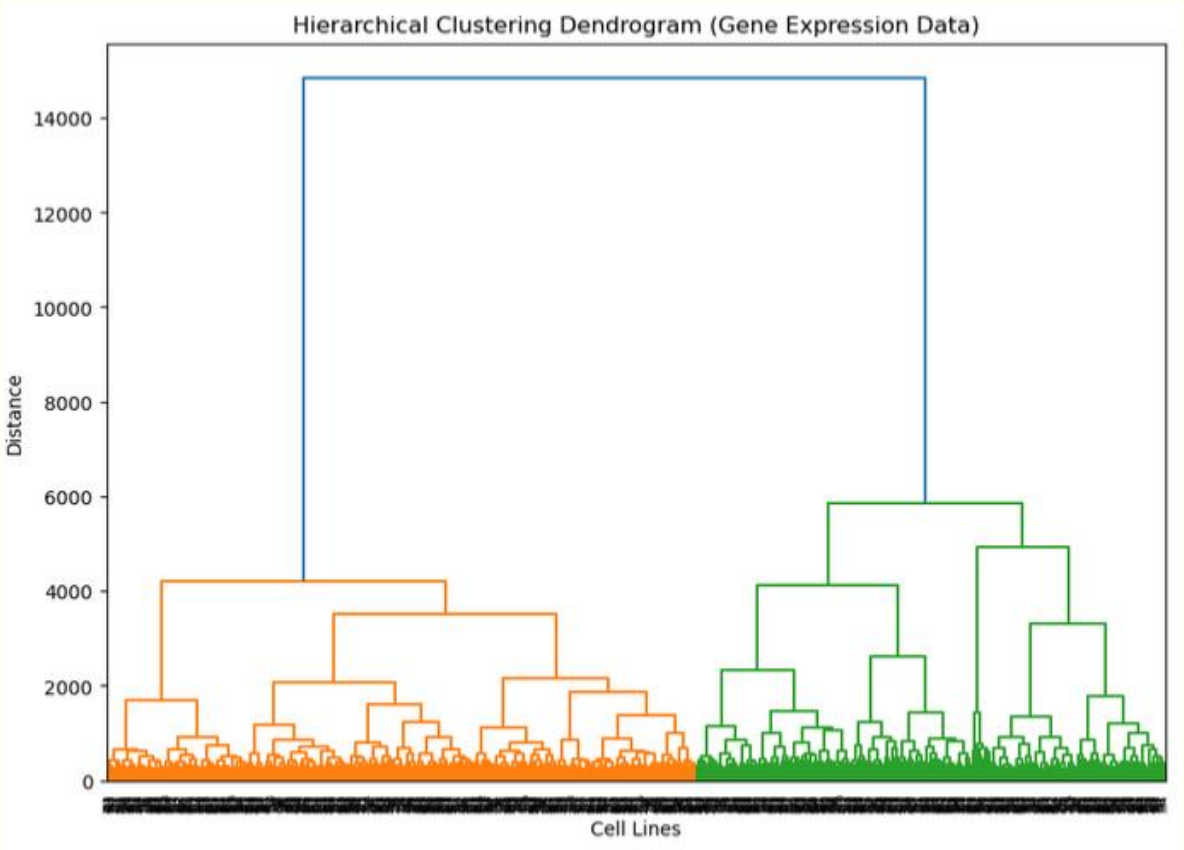


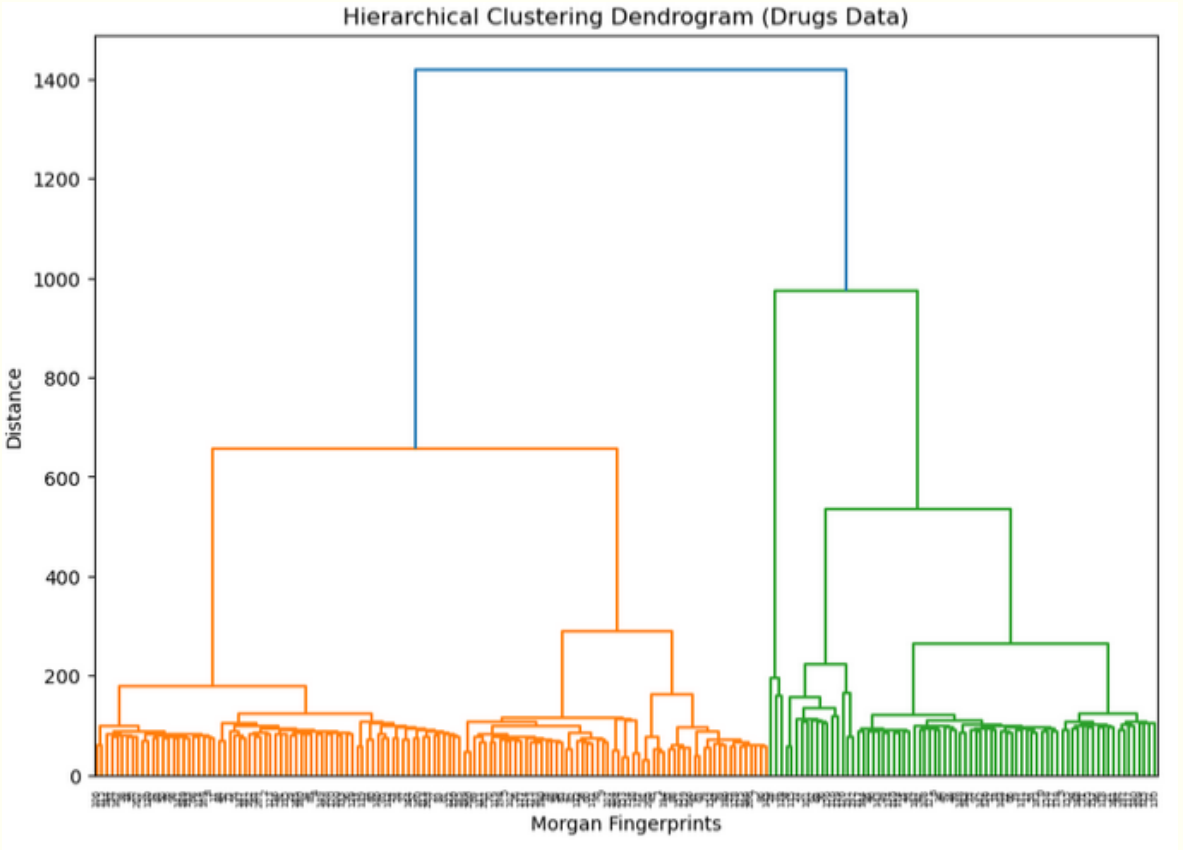
Métodos de clustering: K-means clustering



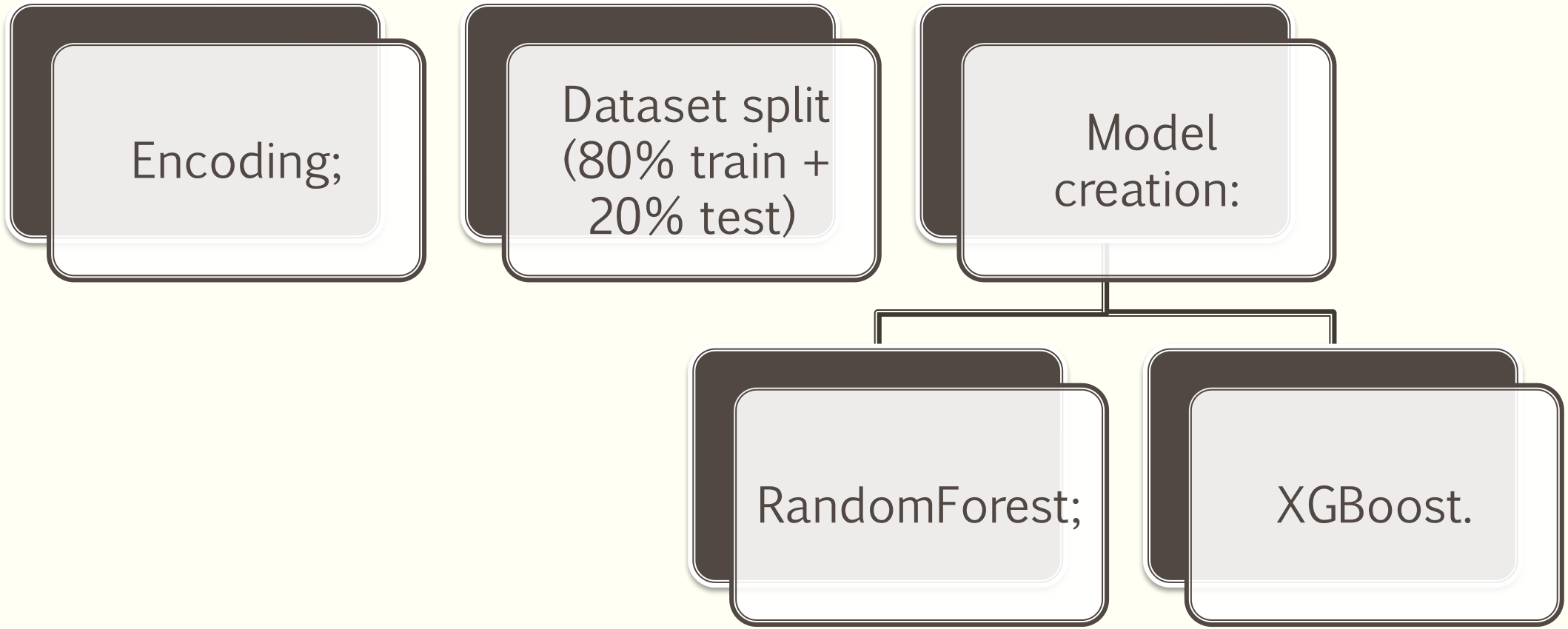


Métodos de clustering: Hierarchical Clustering





Aprendizagem máquina



Aprendizagem máquina

Modelo de Random Forest

```
# Modelo de Random Forest
rf = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
rf.fit(X_train, y_train)

# Predições e métricas
y_pred_rf = rf.predict(X_test)
print("Random Forest:")
print("MSE:", mean_squared_error(y_test, y_pred_rf))
print("R2 Score:", r2_score(y_test, y_pred_rf))
```

✓ 20m 28.6s

```
Random Forest:
MSE: 7.1242157008668014
R2 Score: -0.00513707069332936
```

Modelo de XGBoost

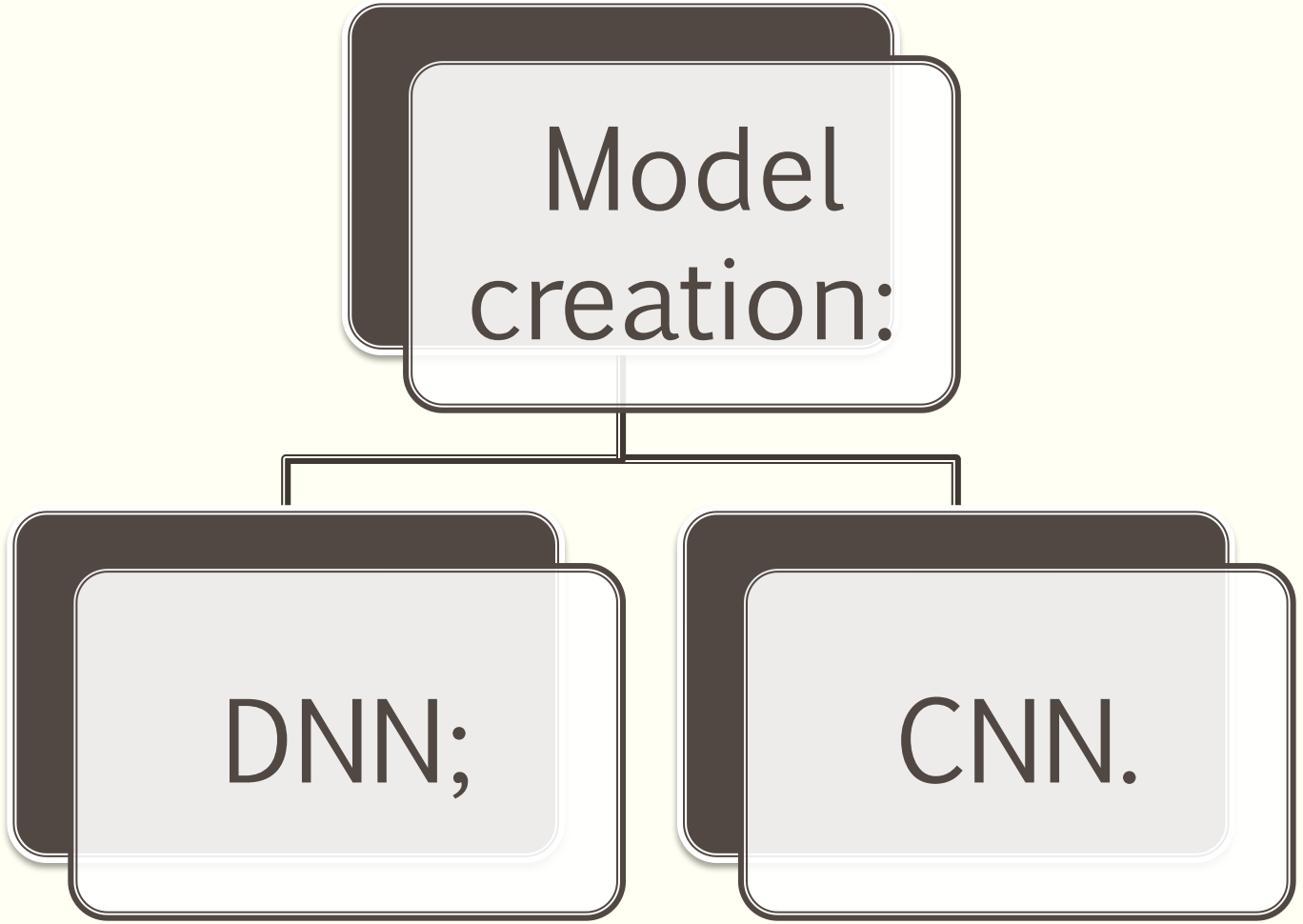
```
# Modelo de XGBoost
xgboost = xgb.XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=6, random_state=42)
xgboost.fit(X_train, y_train)

# Predições e métricas
y_pred_xgb = xgboost.predict(X_test)
print("\nXGBoost:")
print("MSE:", mean_squared_error(y_test, y_pred_xgb))
print("R2 Score:", r2_score(y_test, y_pred_xgb))
```

✓ 2m 56.6s

```
XGBoost:
MSE: 7.482964163241064
R2 Score: -0.05575195852480386
```

Deep learning



Deep learning: DNN

```python
def create_dnn(input_shape):
    model = Sequential([
        layers.Dense(128, activation='relu', input_shape=input_shape),
        layers.Dropout(0.3),
        layers.Dense(64, activation='relu'),
        layers.Dropout(0.3),
        layers.Dense(1, activation='linear')
    ])
    model.compile(optimizer='adam', loss='mse', metrics=['mae'])
    print("DNN Summary:")
    print(model.summary())
    return model
```

Deep learning: CNN

```
def create_cnn(input_shape):
    model = Sequential([
        layers.Reshape((input_shape[0], 1), input_shape=input_shape),
        layers.Conv1D(32, kernel_size=3, activation='relu'),
        layers.MaxPooling1D(pool_size=2),
        layers.Conv1D(64, kernel_size=3, activation='relu'),
        layers.MaxPooling1D(pool_size=2),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dropout(0.3),
        layers.Dense(1, activation='linear')
    ])
    model.compile(optimizer='adam', loss='mse', metrics=['mae'])
    print("CNN Summary:")
    print(model.summary())
    return model
```

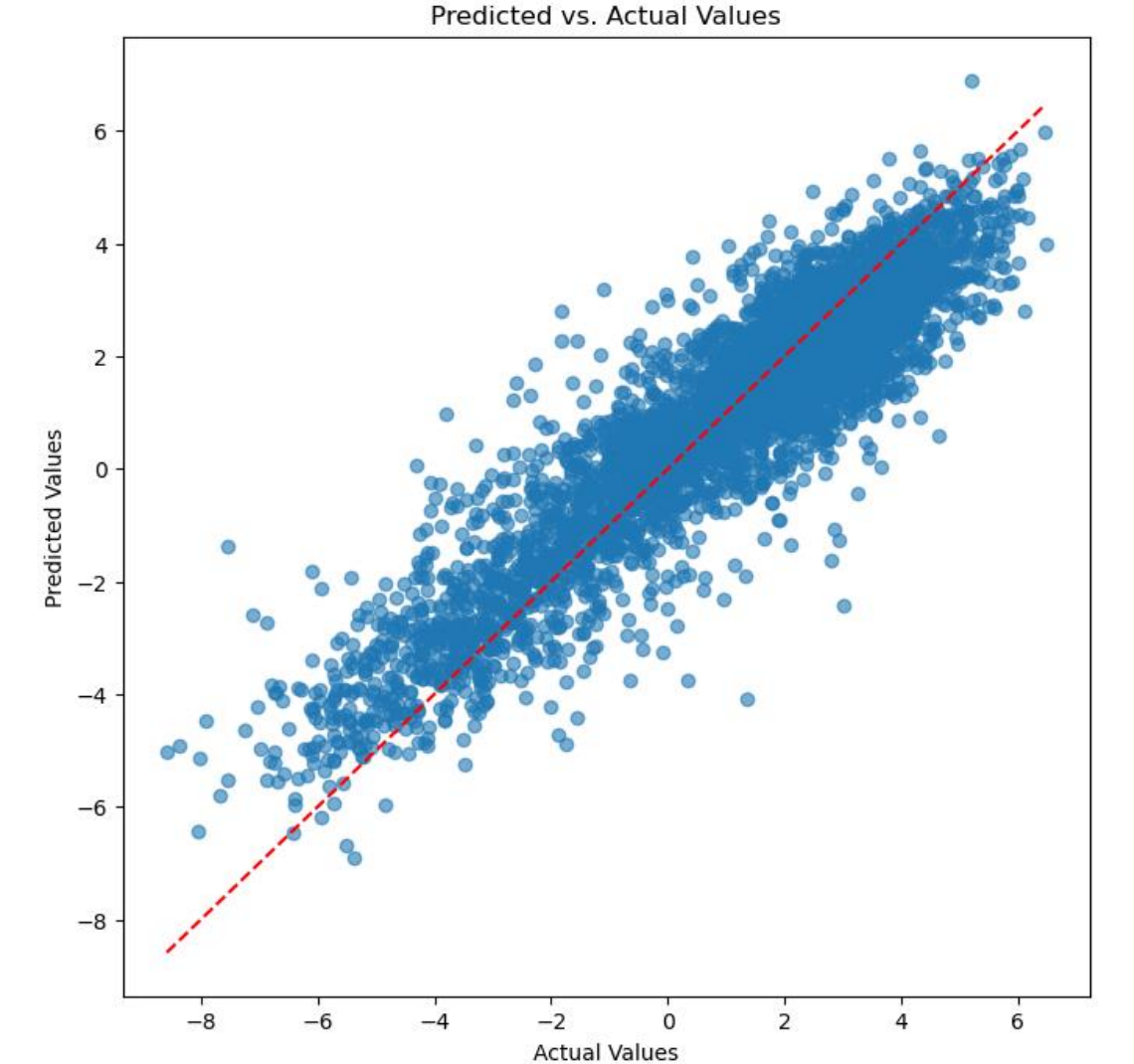
Deep learning: DNN e CNN

```
dnn_history = dnn_model.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=50,
    batch_size=32,
    verbose=1
)
```

```
cnn_history = cnn_model.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=50,
    batch_size=32,
    verbose=1
)
```

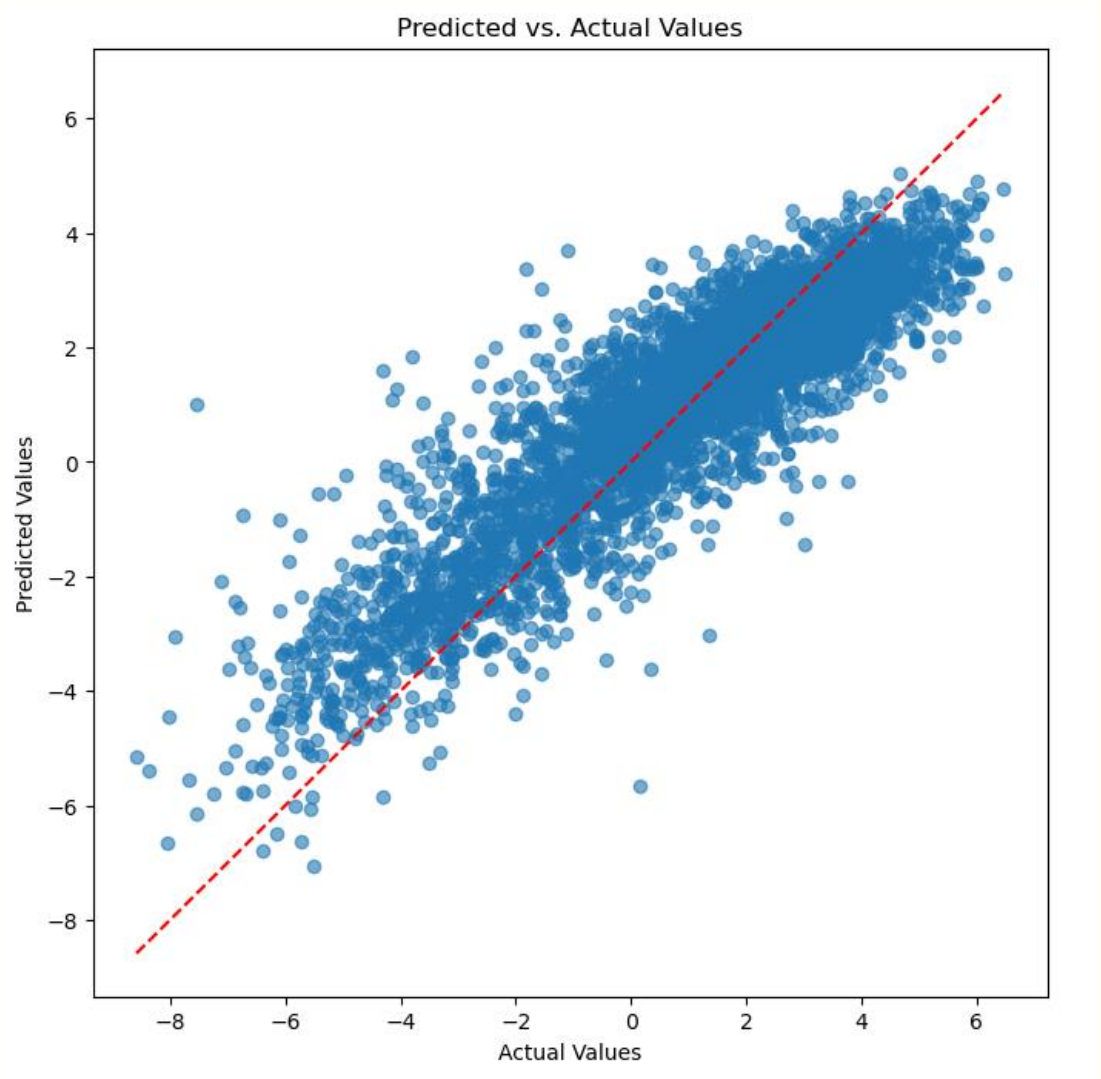
Deep learning: DNN

- MAE no conjunto de teste: 0.8769



Deep learning: CNN

- MAE no conjunto de teste: 0.9171



TRABALHO FUTURO

Este trabalho foi desenvolvido no âmbito da unidade curricular Sistemas Inteligentes para Bioinformática do Mestrado em Bioinformática, pelos alunos:

Beatriz Santos (pg46723)
Duarte Velho (pg53481)
Ricardo Oliveira (pg53501)
Rita Nóbrega (pg46733)
Rodrigo Esperança (pg50923)

Bibliografia

- [1] Chang, Y., Park, H., Yang, H.-J., Lee, S., Lee, K.-Y., Kim, T. S., Jung, J., & Shin, J.-M. (2018). Cancer Drug Response Profile scan (CDRscan): A Deep Learning Model That Predicts Drug Effectiveness from Cancer Genomic Signature. Scientific Reports, 8(1), 8857
- [2] Baptista, D., Ferreira, P. G., & Rocha, M. (2021). Deep Learning for Drug Response Prediction in Cancer. Briefings in Bioinformatics, 22(1), 360–379.