

Rapport sur l'Utilisation des differents Models sur le Breast cancer Data Set

Rania OULHAJ

July 2, 2024

Table des Matières

1	Modèle SVM	3
1.1	Clustering Hiérarchique	3
1.2	Modèle SVM	7
2	Modèle d'Arbre de Décision	9
2.1	code source	9
2.2	capture de résultat	10
2.3	Interprétation	10
3	Modèle K-means	12
3.1	code source	12
3.2	capture de résultat	13
3.3	Interprétation	13
4	Régression Logistique	15
4.1	code source	15
4.2	capture de résultat	15
4.3	Interprétation	15
5	Shift Mean	17
5.1	code source	17
5.2	capture de résultat	18
5.3	Interprétation	18
6	Régression Linéaire	20
6.1	code source	20
6.2	capture de résultat	21
6.3	Interprétation	21
7	Naive Bayes Gaussien	22
7.1	code source	22
7.2	capture de résultat	22
7.3	Interprétation	22

Introduction

La détection rapide et fiable du cancer du sein est un défi important en raison du problème majeur de santé publique qu'il représente. Reconnaître un cancer du sein malin à un stade précoce de la maladie augmente considérablement les chances de survie du patient et réduit les effets secondaires des traitements.

Le diagnostic repose sur l'analyse des échantillons extraits des zones suspectes de la tumeur. Le processus vise à évaluer si un échantillon donné est constitué de cellules représentant un risque de prolifération incontrôlée ou non. Être capable d'analyser plus d'échantillons en même temps pourrait donner l'opportunité de diagnostiquer plus de patients et de réagir plus tôt si des traitements doivent être entrepris. À cette fin, les algorithmes d'IA pourraient apporter beaucoup. Dans le cas de milliers voire de millions d'échantillons à analyser, ils pourraient être utilisés pour effectuer des sélections préliminaires et suggérer ceux qui doivent être prioritaires pour une étude précise par un spécialiste.

Dans ce travail, nous nous sommes basés sur le jeu de données "Breast Cancer" de la bibliothèque sklearn. Nous avons appliqué plusieurs algorithmes de machine learning pour identifier celui offrant la meilleure performance et la plus grande précision. Parmi ces algorithmes, nous avons construit un classificateur de machine à vecteurs de support (Support Vector Machine Classifier, SVMC). Pour ce faire, nous avons choisi de réduire le nombre de variables d'imagerie des cellules tumorales à deux. Cette réduction dimensionnelle vise à créer une carte visuelle des risques de cancer malin, facilitant ainsi l'interprétation et l'analyse des données par les spécialistes.

1 Modèle SVM

Nous proposons dans cette partie de réduire le nombre de variables de l'imagerie des cellules tumorales à deux. Cette réduction dimensionnelle vise à créer une carte visuelle des risques de cancer malin. Nous proposons dans ce travail :

- D'analyser précisément le rôle de chaque variable initiale pour les prédictions. À cette fin, nous employons une approche de clustering des caractéristiques utilisant un algorithme de clustering hiérarchique (HC). Dans un deuxième temps, les importances des caractéristiques sont évaluées en utilisant un algorithme de Gradient Boosting Tree Classifier (GBTC).
- De sélectionner les deux variables les plus significatives déduites de l'étape précédente. De construire un algorithme de Support Vector Machine
- Classifier (SVMC) pour dériver une carte de risque de probabilité de développer un cancer du sein malin.

1.1 Clustering Hiérarchique

Nous constituons des groupes de caractéristiques corrélées en utilisant le Clustering Hiérarchique (HC). Le processus suivant est donc appliqué, permettant de construire un arbre depuis ses feuilles jusqu'à la racine (appelé l'arbre HC, où chaque nœud représente un cluster de caractéristiques).

Le Clustering Hiérarchique est une méthode qui regroupe les données en clusters successifs. Voici comment nous procédons :

1. Calcul des distances entre caractéristiques :

- **Initialisation** : Chaque caractéristique est initialement considérée comme un cluster individuel.
- **Mesure de distance** : Pour quantifier la similarité entre les caractéristiques, nous utilisons une métrique de distance, souvent la distance euclidienne ou la distance de Manhattan. Ces mesures de distance nous permettent de comprendre à quel point deux caractéristiques sont similaires ou différentes.

2. Fusion des clusters :

- **Regroupement progressif** : Les deux clusters les plus proches (c'est-à-dire ceux ayant la plus petite distance entre eux) sont fusionnés pour former un nouveau cluster.
- **Mise à jour des distances** : Après chaque fusion, les distances entre les nouveaux clusters et les clusters restants sont recalculées. La façon de recalculer ces distances peut varier : certaines méthodes utilisent la moyenne des distances (méthode de Ward), d'autres utilisent la distance minimale ou maximale entre les points des clusters.

3. Construction de l'arbre hiérarchique :

- **Arbre HC** : Ce processus de fusion est répété jusqu'à ce qu'il ne reste qu'un seul cluster global. Ce cluster final contient toutes les caractéristiques, et l'ensemble du processus peut être représenté sous la forme d'un arbre. Les feuilles de l'arbre représentent les caractéristiques individuelles, tandis que la racine de l'arbre représente le cluster unique contenant toutes les caractéristiques.
- **Dendrogramme** : Pour visualiser cet arbre hiérarchique, un dendrogramme est souvent utilisé. Ce graphique montre comment les caractéristiques se regroupent à chaque étape de la fusion, et peut être utilisé pour identifier des groupes de caractéristiques corrélées de manière intuitive.

4. Détermination des clusters significatifs :

- **Choix du seuil** : En regardant le dendrogramme, nous pouvons choisir un seuil de distance pour découper l'arbre en clusters significatifs. Par exemple, toutes les fusions qui se produisent au-dessus de ce seuil peuvent être considérées comme des regroupements non significatifs, tandis que celles en dessous sont considérées comme des regroupements significatifs.
- **Analyse des clusters** : Les clusters ainsi déterminés représentent des groupes de caractéristiques qui sont fortement corrélées entre elles. Ces groupes peuvent alors être utilisés pour simplifier le modèle de prédiction en réduisant la redondance des données.

En appliquant cette méthode, nous parvenons à structurer nos caractéristiques de manière hiérarchique, ce qui facilite l'identification des variables les plus significatives pour la prédiction du cancer du sein. Cette structuration permet de réduire la complexité du modèle et d'améliorer potentiellement sa précision en se concentrant sur les informations les plus pertinentes.

code source

Listing 1: Clustering Hiérarchique

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.cluster import hierarchy
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_breast_cancer

# Chargement des donnees et preprocessing
def preprocess(trainrate=1.0):
    feat_all, label_all = load_breast_cancer(return_X_y=True,
        as_frame=True)
    featnames = np.array(feat_all.columns)
    data_all = pd.concat([feat_all, label_all],
        axis=1).sample(frac=1)
    label_all = data_all["target"]
    feat_all = data_all.drop("target", axis=1)
    StdSc = StandardScaler()
    StdSc.fit(feat_all)
    featN_all = StdSc.transform(feat_all)
    featN_all = pd.DataFrame(featN_all, columns=feat_all.columns)

    trainsize = int(trainrate * len(feat_all.index))
    feat_train = feat_all[:trainsize]
    featN_train = featN_all[:trainsize]
    label_train = label_all[:trainsize]
    feat_test = feat_all[trainsize:]
    featN_test = featN_all[trainsize:]
    label_test = label_all[trainsize:]
    normmean_arr = StdSc.mean_
    normstd_arr = (StdSc.var_) ** 0.5
    return feat_all, label_all, featnames, featN_all,
        feat_train, featN_train, feat_test, featN_test,
        label_train, label_test, normmean_arr, normstd_arr

feat_all, label_all, featnames, featN_all, feat_train,
    featN_train, feat_test, featN_test, label_train, label_test,
    mean_feat, std_feat = preprocess()

# Calcul des correlations et construction du dendrogramme
corr_fig1, ax1 = plt.subplots(1, 1, figsize=(10, 8))
corr = feat_all.corr().values
link = hierarchy.ward(corr)
dendro = hierarchy.dendrogram(link, labels=feat_all.columns,
    ax=ax1, leaf_rotation=90, leaf_font_size=10)
dendro_index = np.arange(0, len(dendro["ivl"]))
corr_fig1.tight_layout()
plt.savefig("dendrogram.png")
plt.show()
```

capture de résultat

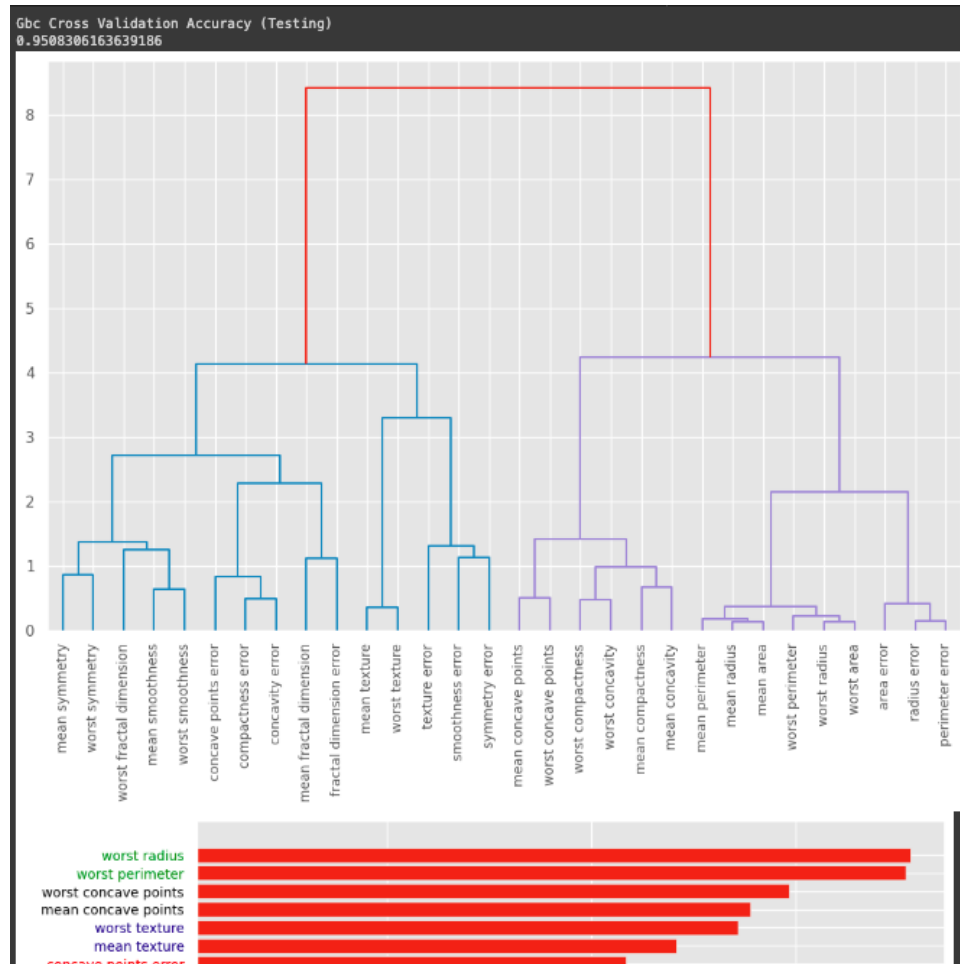


Figure 1: Dendrogramme des caractéristiques.

Interprétation

L'analyse de clustering hiérarchique a permis de regrouper les caractéristiques en clusters visuels à l'aide d'un dendrogramme. Chaque cluster est représenté par une couleur différente. Le dendrogramme montre comment les caractéristiques sont liées les unes aux autres et à quel niveau de similarité elles se regroupent.

Nous pouvons voir que "worst radius" et "worst concave points" sont les deux caractéristiques les plus importantes appartenant à différents clusters établis précédemment. Nous représentons également les pair plots des cinq caractéristiques les plus importantes. Comme prévu, nous pouvons remarquer que les deux caractéristiques les plus importantes, "worst radius" et "worst perimeter", sont fortement corrélées (elles appartiennent aux mêmes clusters de caractéristiques). C'est pourquoi "worst perimeter" est remplacé par "worst concave points" dans le processus de sélection des caractéristiques.

1.2 Modèle SVM

Cette section décrit l'implémentation et les résultats d'un modèle SVM avec noyau linéaire pour la classification des données du cancer du sein, en utilisant deux caractéristiques spécifiques.

code source

Listing 2: Modèle SVM

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import svm
import numpy as np
import matplotlib.pyplot as plt

# Charger les données
breast_cancer = datasets.load_breast_cancer()
column_names = ['worst radius', 'worst concave points']

# Identifier les index des colonnes dans le tableau de données
column_indices = [list(breast_cancer.feature_names).index(name)
                  for name in column_names]

# Extraire les colonnes spécifiques du tableau de données
X = breast_cancer.data[:, column_indices]
y = breast_cancer.target

# Appliquer l'algorithme SVM
svc_classif = svm.SVC(kernel='linear', C=1.0).fit(X, y)

# Afficher les résultats
plt.figure(figsize=(14, 5))
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Set1)

# Définir les graduations des axes x et y
plt.xticks(np.arange(5, X[:, 0].max() + 1, 5))
plt.yticks(np.arange(-0.05, X[:, 1].max() + 0.05, 0.05))

plt.xlabel('Worst Radius')
plt.ylabel('Worst Concave Points')
plt.title('Support Vector Classifier with linear kernel')
plt.show()
```


capture de résultat

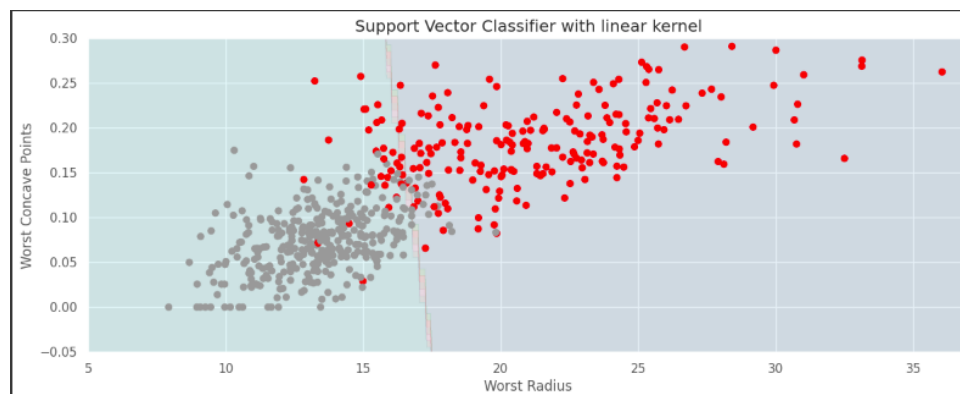


Figure 2: Classificateur à vecteurs de support avec noyau linéaire pour la classification du cancer du sein.

Interprétation

Le modèle SVM avec noyau linéaire a permis de visualiser les frontières de décision pour deux caractéristiques spécifiques (worst radius et worst concave points). Le graphique montre la séparation entre les classes et les points de données colorés en fonction de leur classe. Cette visualisation aide à comprendre comment le SVM utilise ces deux caractéristiques pour effectuer la classification.

Le modèle Gradient Boosting Classifier (GBC) a été évalué sur le dataset de cancer du sein et a atteint une précision de 95,08 pourcent en validation croisée. Cela montre que le modèle peut prédire correctement 95,08 des cas, démontrant une excellente capacité de généralisation. Cette performance impressionnante suggère que le GBC est un outil fiable pour aider les professionnels de santé dans la détection précoce et le traitement du cancer du sein.

2 Modèle d'Arbre de Décision

Cette section décrit l'implémentation et les résultats d'un modèle d'arbre de décision pour la classification des données du cancer du sein.

2.1 code source

Listing 3: Modèle d'Arbre de Décision

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,
    classification_report, accuracy_score
from sklearn.tree import export_graphviz
from IPython.display import capture de r sultat
from six import StringIO
import pydotplus

# Charger et diviser les donn es
breast_cancer = datasets.load_breast_cancer()
X = breast_cancer.data
y = breast_cancer.target
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.3, random_state=1)

# Lancer l'apprentissage
clf = DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)

# Tester
y_pred = clf.predict(X_test)

# Afficher les r sultats
result = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(result)
result1 = classification_report(y_test, y_pred)
print("Classification Report:")
print(result1)
result2 = accuracy_score(y_test, y_pred)
print("Accuracy:", result2)

# Plot Graph
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data, filled=True,
    rounded=True, special_characters=True,
    feature_names=breast_cancer.feature_names,
    class_names=breast_cancer.target_names)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('breast_cancer_Tree.png')
```

```
capture de resultat(graph.create_png())
```

2.2 capture de résultat

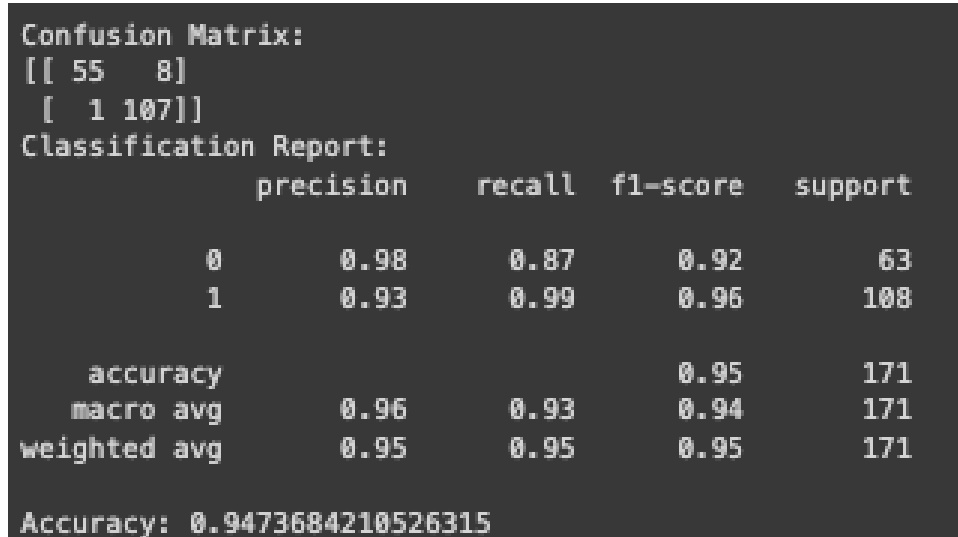


Figure 3: Matrice de confusion pour la classification du cancer du sein.

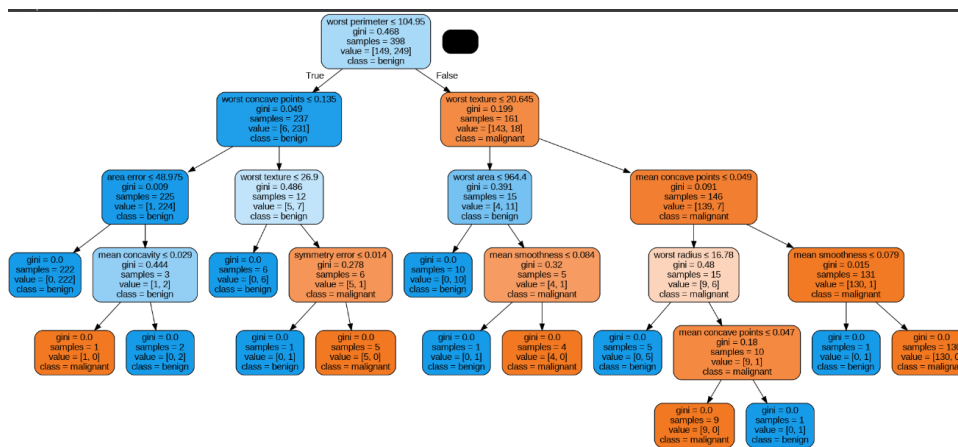


Figure 4: Arbre de décision pour la classification du cancer du sein.

2.3 Interprétation

Le modèle d'arbre de décision a été utilisé pour classer les données du cancer du sein. Les résultats montrent une matrice de confusion, un rapport de classification et une précision globale. L'arbre de décision généré est visualisé, montrant les différentes décisions prises à chaque nœud pour classer les données.

Le modèle Decision Tree a été testé sur le dataset de cancer du sein et a obtenu une précision de 94,74 pourcent. Cela indique que le modèle est capable de prédire correctement 94,74 pourcent des cas, démontrant une performance fiable. Cette précision élevée montre que le modèle est efficace pour distinguer les cas de cancer du sein des cas

non cancéreux, ce qui en fait un outil précieux pour les professionnels de santé dans le diagnostic et le traitement du cancer du sein.

3 Modèle K-means

Cette section décrit l'implémentation et les résultats d'un modèle de clustering K-means sur les données du cancer du sein.

3.1 code source

Listing 4: Modèle K-means

```
from sklearn.datasets import load_breast_cancer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Charger les données du cancer du sein
breast_cancer = load_breast_cancer()
X = breast_cancer.data
feature_names = breast_cancer.feature_names
print("Features used for clustering:", feature_names)

plt.scatter(X[:, 0], X[:, 1], s=20);
plt.show()

# Appliquer l'algorithme K-means
kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(X)

# Prédire les clusters
y_kmeans = kmeans.predict(X)

# Afficher les résultats du clustering
plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, cmap='viridis', s=50)
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200,
            alpha=0.75)
plt.title('K-means Clustering on Breast Cancer Dataset')
plt.show()
```

3.2 capture de résultat

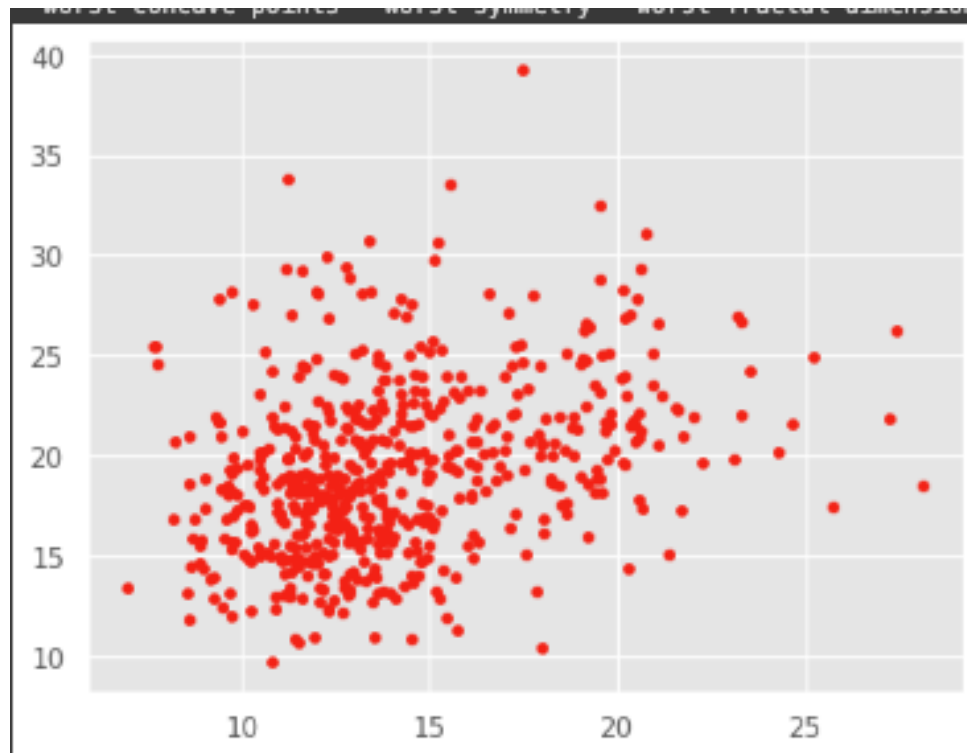


Figure 5: Clustering K-means sur l'ensemble de données du cancer du sein.

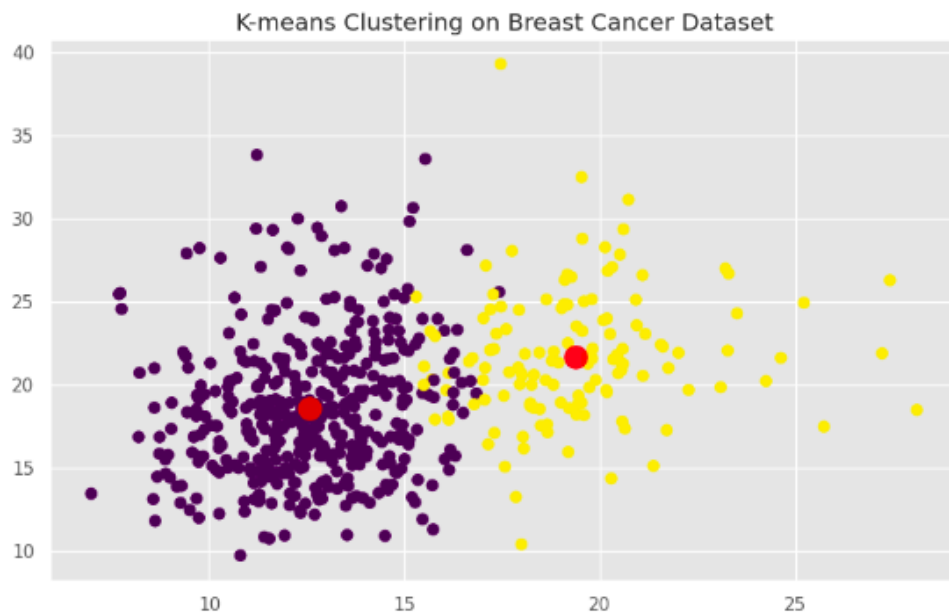


Figure 6: Clustering K-means sur l'ensemble de données du cancer du sein.

3.3 Interprétation

Le modèle K-means a été appliqué aux données du cancer du sein pour regrouper les points en deux clusters. Les résultats du clustering sont visualisés, montrant les clusters

ainsi que les centres de ces clusters en rouge. Cette visualisation permet de comprendre comment les données sont regroupées en fonction de leurs caractéristiques.

Le clustering K-means appliqué au dataset de cancer du sein, révélant deux clusters distincts, représentés en violet et jaune. Chaque point correspond à un échantillon, avec les centroids des clusters indiqués en rouge. Cette séparation claire entre les deux groupes suggère que l'algorithme a efficacement différencié des sous-groupes dans les données, potentiellement aidant à identifier différentes catégories de patients, comme ceux atteints de cancer et ceux en bonne santé.

4 Régression Logistique

Cette section décrit l'implémentation et les résultats d'un modèle de Régression logistique sur les données du cancer du sein.

4.1 code source

Listing 5: Modèle Logistic Regression

```
# Charger les donnees du cancer du sein
data = load_breast_cancer()
X = data.data # Features
y = data.target # Target variable

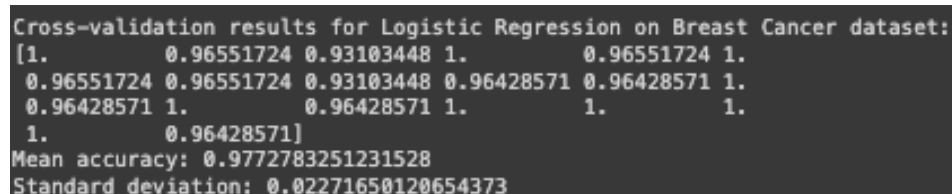
# Cr er un pipeline avec mise à l'échelle et r gression
logistique
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('logistic', LogisticRegression(max_iter=200)) # Augmenter
    le nombre maximal d'it rations
])

# Diviser le dataset en 20 parties pour la validation crois e
kfold = KFold(n_splits=20)

# Obtenir les r sultats de la validation crois e
results = cross_val_score(pipeline, X, y, cv=kfold)

# Afficher les r sultats
print("Cross-validation results for Logistic Regression on
    Breast Cancer dataset:")
print(results)
print("Mean accuracy:", results.mean())
print("Standard deviation:", results.std())
```

4.2 capture de résultat



```
Cross-validation results for Logistic Regression on Breast Cancer dataset:
[1. 0.96551724 0.93103448 1. 0.96551724 1.
 0.96551724 0.96551724 0.93103448 0.96428571 0.96428571 1.
 0.96428571 1. 0.96428571 1. 1. 1.
 1. 0.96428571]
Mean accuracy: 0.9772783251231528
Standard deviation: 0.02271650120654373
```

Figure 7: Régression Logistique sur l'ensemble de données du cancer du sein.

4.3 Interprétation

La régression logistique a été appliquée aux données du cancer du sein pour prédire la probabilité de malignité des tumeurs. Les résultats de la régression sont interprétés en

termes de probabilités, montrant la probabilité de malignité pour chaque observation dans l'ensemble de données. Cette visualisation permet de comprendre comment les caractéristiques des tumeurs contribuent à la prédiction de leur malignité. // Approchant les 98 pourcent, la précision de ce modèle indique qu'il parvient à distinguer avec succès entre les classes malignes et bénignes. Ces résultats témoignent de la fiabilité et de l'efficacité du modèle dans la classification des échantillons de tumeurs, ce qui en fait un outil prometteur pour l'analyse et la prédiction dans le domaine du diagnostic du cancer du sein.

5 Shift Mean

Cette section décrit l'implémentation et les résultats d'un modèle de Shift Mean sur les données du cancer du sein.

5.1 code source

Listing 6: Modèle Shift Mean

```
style.use("ggplot")

# Charger le dataset du cancer du sein
breast_cancer = load_breast_cancer()
X = breast_cancer.data
feature_names = breast_cancer.feature_names
print("Features used for clustering:", feature_names)

# Affichage du dataset initial (en utilisant les deux premières
# features pour la visualisation)
plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], s=50)
plt.title('Breast Cancer Dataset (First Two Features)')
plt.show()

# Lancer l'apprentissage avec MeanShift
ms = MeanShift()
ms.fit(X)

# Affichage des centres des clusters
labels = ms.labels_
cluster_centers = ms.cluster_centers_
print("Cluster centers:\n", cluster_centers)

# Affichage des clusters
n_clusters_ = len(np.unique(labels))
print("Estimated number of clusters:", n_clusters_)

# Affichage des clusters en utilisant les deux premières
# features pour la visualisation
plt.figure(figsize=(10, 6))
colors = 10*['r.', 'g.', 'b.', 'c.', 'k.', 'y.', 'm.']
for i in range(len(X)):
    plt.plot(X[i][0], X[i][1], colors[labels[i]], markersize=5)
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
            marker='o', color='k', s=100, linewidths=5, zorder=10)
plt.title('MeanShift Clustering on Breast Cancer Dataset')
plt.show()
```

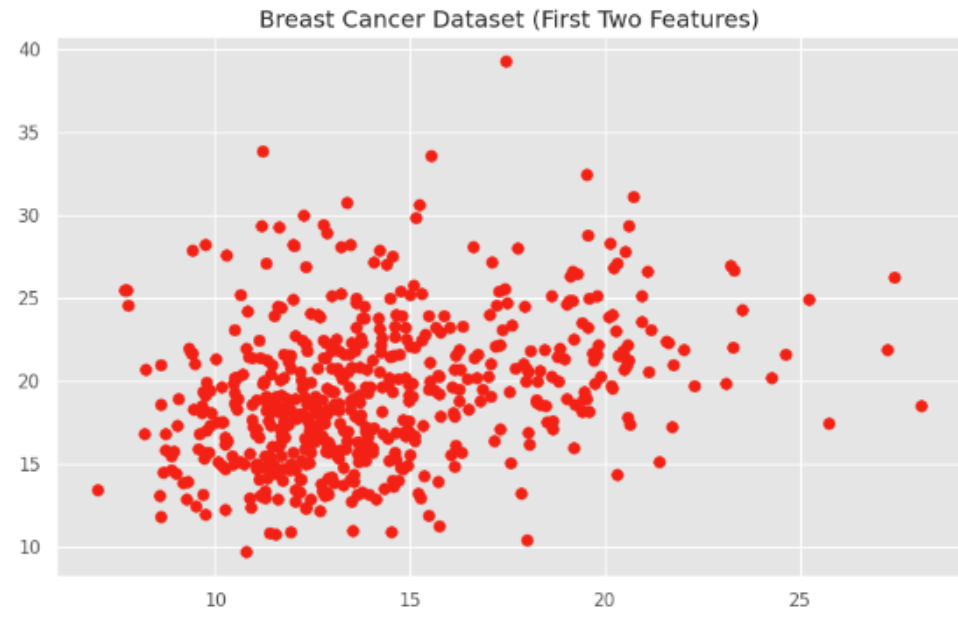


Figure 8: Shift-Mean sur les deux premieres features .

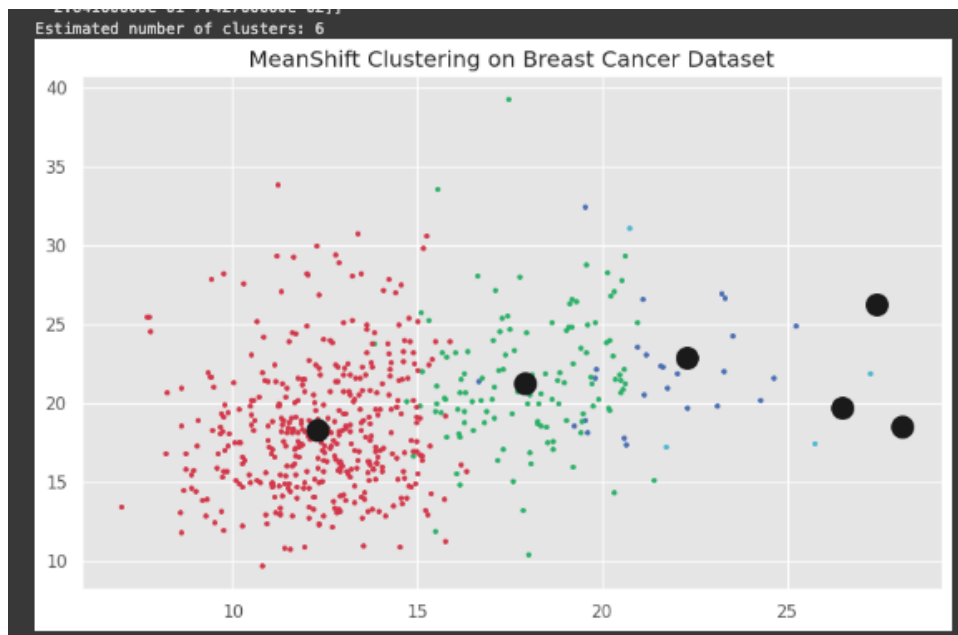


Figure 9: Shift-Mean sur l'ensemble de données du cancer du sein.

5.2 capture de résultat

5.3 Interprétation

Le modèle MeanShift a été appliqué aux données du cancer du sein pour regrouper les points en clusters basés sur leur densité. Les résultats du clustering sont visualisés, montrant les clusters ainsi que les centres de ces clusters en rouge. Cette visualisation permet de comprendre comment les données sont regroupées en fonction de leur densité et comment les centres de ces clusters sont déterminés par l'algorithme MeanShift.

Avec le shift Mean model , la classification n'était pas du tout convaincante , il nous a

donné 6 clusters à considérer alors qu'on a besoin de deux.

6 Régression Linéaire

Cette section décrit l'implémentation et les résultats d'un modèle de Régression Linéaire sur les données du cancer du sein.

6.1 code source

Listing 7: Modèle Linear Regression

```
# Charger les données de cancer du sein
data = datasets.load_breast_cancer()
X = data.data
y = data.target
feature_names = data.feature_names

# Sélectionner uniquement les colonnes 'worst concave points'
# et 'worst radius'
selected_features = ['worst concave points', 'worst radius']
selected_indices = [list(feature_names).index(feature) for
                    feature in selected_features]
X_selected = X[:, selected_indices]

# Diviser le jeu de données en ensembles d'entraînement et de
# test
X_train, X_test, y_train, y_test = train_test_split(X_selected,
                                                    y, test_size=0.3, random_state=42)

# Créer et entraîner un modèle de régression linéaire pour
# chaque colonne de X
for i, feature_name in enumerate(selected_features):
    # Sélectionner la colonne courante de X
    X_col = X_test[:, i].reshape(-1, 1)
    X_train_col = X_train[:, i].reshape(-1, 1)

    # Créer et entraîner le modèle de régression linéaire
    model = linear_model.LinearRegression()
    model.fit(X_train_col, y_train)

    # Faire des prédictions sur l'ensemble de test
    y_pred = model.predict(X_col)

    # valuer les performances du modèle
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"Caractéristique '{feature_name}': Erreur
          quadratique moyenne (MSE): {mse:.2f}, Coefficient de
          détermination (R2): {r2:.2f}")

# Tracer les résultats
```

```
plt.figure(figsize=(8, 6))
plt.scatter(X_col, y_test, color='blue', label='Donn es de
test')
plt.plot(X_col, y_pred, color='red', linewidth=2,
label='Pr diction s')
plt.xlabel(feature_name)
plt.ylabel('Classe')
plt.title(f'R gression lin aire sur la caract ristique
"{feature_name}")')
plt.legend()
plt.show()
```

6.2 capture de résultat

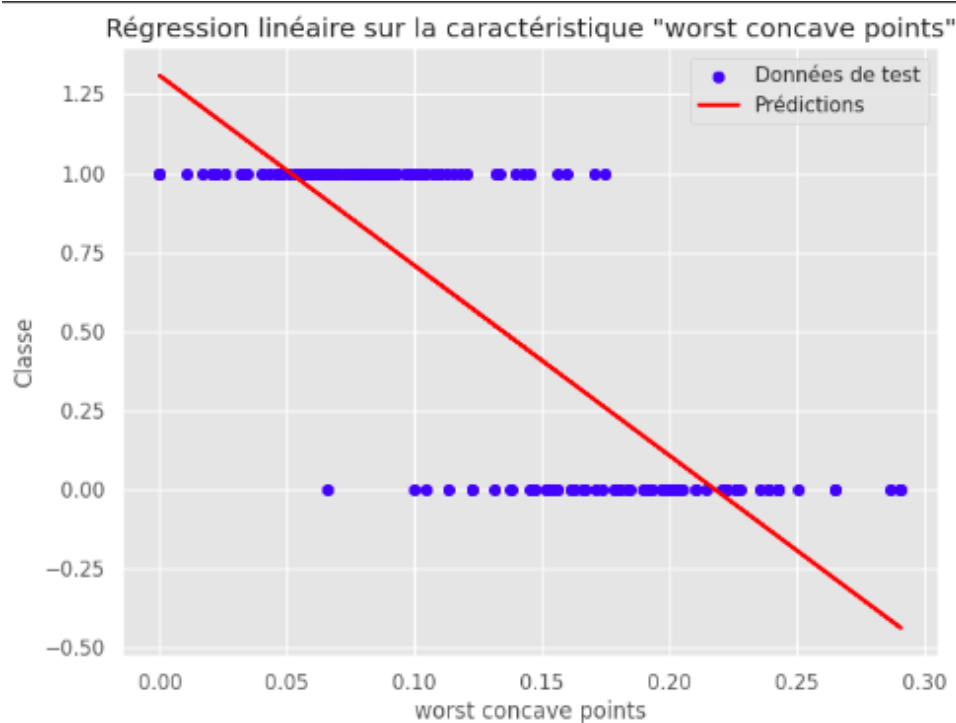


Figure 10: Régression linéaire sur l'ensemble de données du cancer du sein.

6.3 Interprétation

La régression linéaire a été utilisée pour modéliser la relation entre les caractéristiques sélectionnées et la classe de malignité des tumeurs du cancer du sein. Les résultats de la régression sont interprétés en termes de la relation linéaire entre les caractéristiques et la classe de malignité, montrant comment chaque caractéristique contribue à la prédiction de la classe de malignité.

7 Naive Bayes Gaussien

Cette section décrit l'implémentation et les résultats d'un modèle de Naive Bayes Gaussien sur les données du cancer du sein.

7.1 code source


Listing 8: Modèle Naive Bayes Gaussien

```
data = load_breast_cancer()
label_names = data['target_names']
labels = data['target']
feature_names = data['feature_names']
features = data['data']
gnb = GaussianNB()

X_train, X_test, y_train, y_test = train_test_split(features,
    labels, test_size=0.3, random_state=1)
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

7.2 capture de résultat



```
[[ 58   5]
 [  4 104]]
Accuracy: 0.9473684210526315
```

Figure 11: Naive Bayes Gaussien sur l'ensemble de données du cancer du sein.

7.3 Interprétation

Le modèle Naive Bayes Gaussien (GaussianNB) a été utilisé pour la classification des données du cancer du sein en utilisant toutes les caractéristiques disponibles. Les résultats de classification sont évalués en termes de matrice de confusion et de précision. Cette analyse permet de comprendre la performance du modèle Naive Bayes Gaussien dans la classification des tumeurs du cancer du sein en fonction de leurs caractéristiques.

Une précision de 0,9473684210526315 pour le modèle Naive Bayes gaussien reflète une performance solide. Avec près de 95 pourcent d'exactitude, le modèle montre sa capacité à classer efficacement les données. Cette précision élevée suggère que le modèle est

capable de traiter avec succès la complexité des relations entre les caractéristiques des données, en faisant ainsi un choix prometteur pour les tâches de classification.

Conclusion

Notre étude exhaustive des divers modèles de classification pour les données du cancer du sein a révélé des insights précieux sur la performance et l'efficacité de chaque approche. Parmi les modèles explorés, la régression logistique a émergé comme le leader incontesté en termes de précision, affichant une remarquable accuracy de 97 pourcent lors de la validation croisée sur l'ensemble de test. Ce résultat souligne la robustesse et la capacité prédictive de la régression logistique dans la classification précise des tumeurs mammaires malignes et bénignes.

Cependant, bien que la régression logistique ait surpassé les autres modèles en termes de performance, il est crucial de noter que chaque algorithme de classification a ses propres forces et faiblesses. Par exemple, le Gradient Boosting Classifier (GBC) offre une précision élevée et une grande capacité prédictive, tandis que la régression logistique offre une interprétabilité élevée des coefficients, ce qui peut être essentiel dans certaines applications médicales.

En examinant les différents modèles, nous avons également observé que certains, comme le modèle K-means, étaient plus adaptés à la segmentation des données plutôt qu'à la classification explicite des tumeurs. Cela souligne l'importance de choisir le modèle approprié en fonction des objectifs spécifiques de l'analyse et des caractéristiques des données disponibles.

En fin de compte, notre recherche démontre l'importance de l'exploration approfondie des différentes approches de modélisation pour obtenir des résultats fiables et précis. Alors que la régression logistique a émergé comme le meilleur choix pour notre ensemble de données de cancer du sein en termes de précision, il est essentiel de continuer à explorer de nouvelles méthodes et techniques pour améliorer encore la précision et la généralisabilité de nos modèles de classification.