

Progetto di Machine Learning: Predizione del Cancro

Dataset: Breast Cancer Wisconsin

Sara Gravante (886191)
Alessio Tosato (886081)

Università degli studi di Milano-Bicocca
Corso di laurea Magistrale in Informatica

Anno Accademico: 2024/2025

- 1 Introduzione
- 2 Analisi Esplorativa dei Dati (EDA)
- 3 Preprocessing
- 4 Modelli
- 5 Valutazione Modelli
- 6 Conclusioni

- **Nome:** Breast Cancer Wisconsin (Diagnostic) Dataset
- **Fonte:** Kaggle (<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>)
- **Contenuto:** Caratteristiche computate da un'immagine digitalizzata di un agoaspirato (FNA) di una massa mammaria. Descrivono le caratteristiche dei nuclei cellulari.
- **Numero di Istanze:** 569 osservazioni.
- **Attributi:**
 - ID
 - Diagnosi (M = maligno, B = benigno)
 - 30 caratteristiche a valore reale per ogni nucleo cellulare (es. raggio, texture, perimetro, area, ecc.)
Di ciascuna caratteristica: media, errore standard, valore peggiore.

- **Scopo:** Identificare pattern, anomalie, relazioni tra le variabili e preparare i dati per la modellazione.
- **Struttura:**
 - Presenza di valori nulli
 - Analisi bilanciamento variabile target
 - Studio correlazioni tra caratteristiche
 - Visualizzazione di distribuzioni e boxplot delle caratteristiche

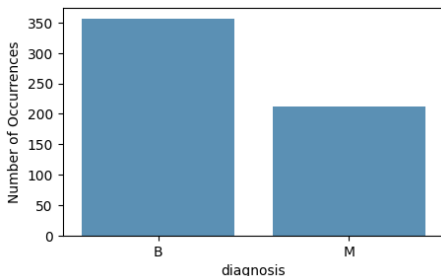
Viene rilevata la presenza di una colonna, `Unnamed: 32`, contenente esclusivamente valori nulli.

26	<code>smoothness_worst</code>	569 non-null	float64
27	<code>compactness_worst</code>	569 non-null	float64
28	<code>concavity_worst</code>	569 non-null	float64
29	<code>concave points_worst</code>	569 non-null	float64
30	<code>symmetry_worst</code>	569 non-null	float64
31	<code>fractal_dimension_worst</code>	569 non-null	float64
32	<code>Unnamed: 32</code>	0 non-null	float64

EDA: Bilanciamento del Target

Distribuzione delle classi della **variabile target** diagnosis:

- Benigno (B): 357 occorrenze ($\approx 62.74\%$)
- Maligno (M): 212 occorrenze ($\approx 37.26\%$)



Osservazione: Presenza di un leggero sbilanciamento delle classi.

Soluzione: verrà utilizzato il parametro `class_weight = 'balanced'` nei modelli per bilanciare i pesi delle classi e mitigarne l'impatto.

EDA: Matrice di Correlazione - 1

La **matrice di correlazione** permette di visualizzare graficamente le relazioni lineari tra le variabili numeriche.

Ogni elemento della matrice rappresenta il *coefficiente di correlazione* tra due variabili, indicando la forza e la direzione della loro relazione.

Dalla visualizzazione sono stati esclusi gli attributi:

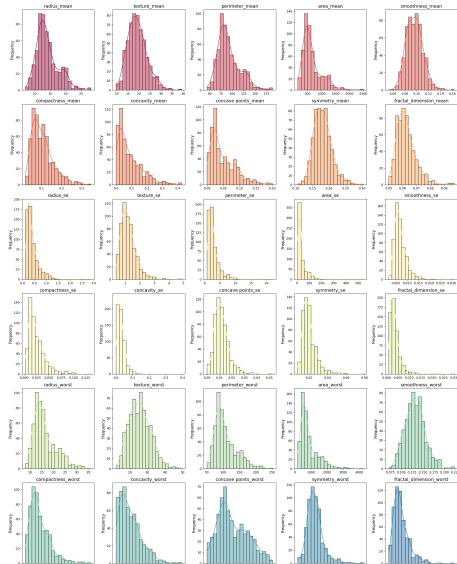
- id
- diagnosis
- Unnamed: 32

EDA: Matrice di Correlazione - 2

	radius_mean	-1	0.32	1	0.99	0.17	0.51	0.68	0.82	0.15	-0.31	0.68	-0.097	0.6	0.74	-0.22	0.21	0.19	0.38	-0.1	-0.048	0.97	0.3	0.97	0.94	0.12	0.41	0.53	0.74	0.16	0.0071	
	texture_mean	-0.32	1	0.33	0.32	-0.023	0.24	0.3	0.29	0.071	-0.076	0.28	0.39	0.28	0.26	0.066	0.19	0.14	0.16	0.09	0.054	0.95	0.91	0.36	0.34	0.078	0.28	0.3	0.3	0.11	0.12	
	perimeter_mean	1	0.33	1	0.99	0.21	0.56	0.72	0.85	0.18	-0.26	0.69	-0.087	0.69	0.74	-0.2	0.25	0.23	0.41	-0.082	0.005	0.97	0.3	0.97	0.94	0.15	0.46	0.56	0.77	0.19	0.051	
	area_mean	-0.99	0.32	0.99	1	0.18	0.5	0.69	0.82	0.15	-0.28	0.73	-0.066	0.73	0.8	-0.17	0.21	0.21	0.37	-0.072	-0.02	0.96	0.29	0.96	0.96	0.12	0.39	0.51	0.72	0.14	0.0037	
	smoothness_mean	-0.17	-0.023	0.21	0.18	1	0.66	0.52	0.55	0.56	0.58	0.3	0.068	0.3	0.25	0.33	0.32	0.25	0.38	0.2	0.28	0.21	0.036	0.24	0.21	0.81	0.47	0.43	0.5	0.39	0.5	
	compactness_mean	-0.51	0.24	0.56	0.5	0.66	1	0.88	0.83	0.6	0.57	0.5	0.046	0.55	0.46	0.14	0.74	0.57	0.64	0.23	0.51	0.54	0.25	0.59	0.51	0.57	0.87	0.82	0.82	0.51	0.69	
	concavity_mean	-0.68	0.3	0.72	0.69	0.52	0.88	1	0.92	0.5	0.34	0.63	0.076	0.66	0.62	0.099	0.67	0.69	0.68	0.18	0.45	0.69	0.3	0.73	0.68	0.45	0.75	0.88	0.86	0.41	0.95	
	concave points_mean	-0.82	0.29	0.85	0.82	0.55	0.83	0.92	1	0.46	0.17	0.7	0.021	0.71	0.69	0.028	0.49	0.44	0.62	0.095	0.26	0.83	0.29	0.86	0.81	0.45	0.67	0.75	0.91	0.38	0.37	
	symmetry_mean	-0.15	0.071	0.18	0.15	0.56	0.6	0.5	0.46	1	0.48	0.3	0.13	0.31	0.22	0.19	0.42	0.34	0.39	0.45	0.33	0.19	0.091	0.22	0.18	0.43	0.47	0.43	0.43	0.7	0.44	
	fractal_dimension_mean	-0.31	-0.076	-0.26	-0.28	0.58	0.57	0.34	0.17	0.48	1	0.0011	0.16	0.04	-0.22	0.4	0.56	0.45	0.34	0.35	0.69	0.25	0.051	0.21	-0.23	0.5	0.46	0.35	0.18	0.33	0.77	
	radius_se	-0.68	0.28	0.69	0.73	0.3	0.5	0.63	0.7	0.3	0.0001	1	0.21	0.97	0.95	0.16	0.36	0.33	0.51	0.24	0.23	0.72	0.19	0.72	0.75	0.14	0.29	0.38	0.53	0.095	0.05	
	texture_se	-0.097	0.39	-0.087	0.066	0.068	0.046	0.076	0.021	0.13	0.16	0.21	1	0.22	0.11	0.4	0.23	0.19	0.23	0.41	0.28	-0.11	0.41	-0.1	-0.083	0.074	0.092	-0.069	-0.12	-0.13	-0.048	
	perimeter_se	-0.67	0.28	0.69	0.73	0.3	0.55	0.66	0.71	0.31	0.04	0.97	0.22	1	0.94	0.15	0.42	0.36	0.56	0.27	0.24	0.7	0.2	0.72	0.73	0.13	0.34	0.42	0.55	0.11	0.085	
	area_se	-0.74	0.26	0.74	0.8	0.25	0.46	0.62	0.69	0.22	-0.09	0.95	0.11	0.94	1	0.075	0.28	0.27	0.42	0.13	0.13	0.76	0.2	0.76	0.81	0.13	0.28	0.39	0.54	0.074	0.018	
	smoothness_se	-0.22	0.066	-0.2	-0.17	0.33	0.14	0.099	0.028	0.19	0.4	0.16	0.4	0.15	0.075	1	0.34	0.27	0.33	0.41	0.43	-0.23	0.075	-0.22	-0.18	0.31	0.056	0.058	-0.1	-0.11	0.1	
	compactness_se	0.21	0.19	0.25	0.21	0.32	0.74	0.67	0.49	0.42	0.56	0.36	0.23	0.42	0.28	0.34	1	0.8	0.74	0.39	0.8	0.2	0.14	0.26	0.2	0.23	0.68	0.64	0.48	0.28	0.59	
	concavity_se	-0.19	0.14	0.23	0.21	0.25	0.57	0.69	0.44	0.34	0.45	0.33	0.19	0.36	0.27	0.27	0.8	1	0.77	0.31	0.73	0.19	0.1	0.23	0.19	0.17	0.48	0.66	0.44	0.2	0.44	
	concave points_se	-0.38	0.16	0.41	0.37	0.38	0.64	0.68	0.62	0.39	0.34	0.51	0.23	0.56	0.42	0.33	0.74	0.77	1	0.31	0.61	0.36	0.087	0.39	0.34	0.22	0.45	0.55	0.6	0.14	0.31	
	symmetry_se	-0.1	0.009	0.082	0.072	0.2	0.23	0.18	0.095	0.45	0.35	0.24	0.41	0.27	0.13	0.41	0.39	0.31	0.31	1	0.37	-0.13	-0.077	-0.1	-0.11	-0.013	0.06	0.037	-0.03	0.39	0.078	
	fractal_dimension_se	-0.043	0.054	0.005	0.02	0.28	0.51	0.45	0.26	0.33	0.69	0.23	0.28	0.24	0.13	0.43	0.8	0.73	0.61	0.37	1	0.037	0.003	0.001	-0.023	0.17	0.39	0.38	0.22	0.11	0.59	
	radius_worst	-0.97	0.35	0.97	0.96	0.21	0.54	0.69	0.83	0.19	-0.25	0.72	-0.11	0.7	0.76	-0.23	0.2	0.19	0.36	-0.13	-0.037	1	0.36	0.99	0.98	0.22	0.48	0.57	0.79	0.24	0.93	
	texture_worst	-0.3	0.91	0.3	0.29	0.036	0.25	0.3	0.29	0.091	-0.051	0.19	0.41	0.2	0.2	-0.075	0.14	0.1	0.087	-0.077	0.003	0.36	1	0.37	0.35	0.23	0.36	0.37	0.36	0.23	0.092	
	perimeter_worst	-0.97	0.36	0.97	0.96	0.24	0.59	0.73	0.86	0.22	-0.21	0.72	-0.1	0.72	0.76	-0.22	0.26	0.23	0.39	-0.1	-0.001	0.99	0.37	1	0.98	0.24	0.53	0.62	0.82	0.27	0.14	
	area_worst	-0.94	0.34	0.94	0.96	0.21	0.51	0.68	0.81	0.18	-0.23	0.75	-0.083	0.73	0.81	-0.18	0.2	0.19	0.34	-0.11	-0.028	0.98	0.35	0.98	1	0.21	0.44	0.54	0.75	0.21	0.68	
	smoothness_worst	-0.12	0.078	0.15	0.12	0.81	0.57	0.45	0.45	0.43	0.5	0.14	-0.074	0.13	0.13	0.31	0.23	0.17	0.22	-0.013	0.17	0.22	0.23	0.24	0.21	1	0.57	0.52	0.55	0.49	0.62	
	compactness_worst	-0.41	0.28	0.46	0.39	0.47	0.87	0.75	0.67	0.47	0.46	0.29	-0.092	0.34	0.28	-0.056	0.68	0.48	0.45	0.06	0.39	0.48	0.36	0.53	0.44	0.57	1	0.89	0.8	0.61	0.81	
	concavity_worst	-0.53	0.3	0.56	0.51	0.43	0.82	0.88	0.75	0.43	0.35	0.38	-0.069	0.42	0.39	-0.058	0.64	0.66	0.55	0.037	0.38	0.57	0.37	0.62	0.54	0.52	0.89	1	0.86	0.53	0.69	
	concave points_worst	-0.74	0.3	0.77	0.72	0.5	0.82	0.86	0.91	0.43	0.18	0.53	-0.12	0.55	0.54	-0.1	0.48	0.44	0.6	-0.03	0.22	0.79	0.36	0.82	0.75	0.55	0.8	0.86	1	0.5	0.51	
	symmetry_worst	-0.16	0.11	0.19	0.14	0.39	0.51	0.41	0.38	0.7	0.33	0.095	-0.13	0.11	0.074	-0.11	0.28	0.2	0.14	0.39	0.11	0.24	0.23	0.27	0.21	0.49	0.61	0.53	0.5	1	0.54	
	fractal_dimension_worst	-0.0071	0.12	0.051	0.0037	0.5	0.69	0.51	0.37	0.44	0.77	0.05	-0.046	0.085	0.018	0.1	0.59	0.44	0.31	0.078	0.59	0.093	0.22	0.14	0.08	0.62	0.81	0.69	0.51	0.54	1	
	radius_mean																															
	texture_mean																															
	perimeter_mean																															
	area_mean																															
	smoothness_mean																															
	compactness_mean																															
	concavity_mean																															
	concave points_mean																															
	symmetry_mean																															
	fractal_dimension_mean																															
	radius_se																															
	texture_se																															
	perimeter_se																															
	area_se																															
	smoothness_se																															
	compactness_se																															
	concavity_se																															
	concave points_se																															
	symmetry_se																															
	fractal_dimension_se																															
	radius_worst																															
	texture_worst																															
	perimeter_worst																															
	area_worst																															
	smoothness_worst																															
	compactness_worst																															
	concavity_worst																															
	concave points_worst																															

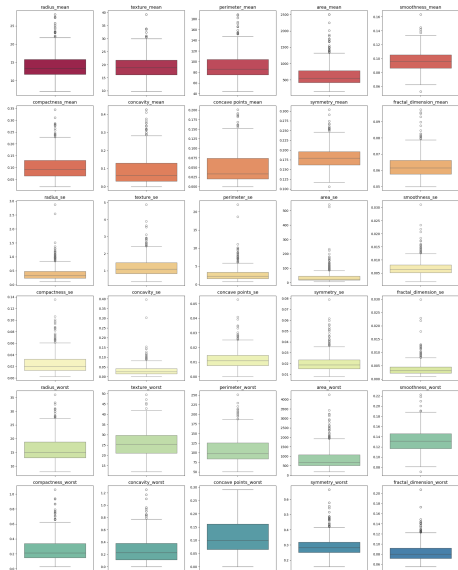
EDA: Distribuzione delle Caratteristiche - 1

Istogrammi con curve di densità: visualizzazione che aiuta a comprendere la distribuzione delle variabili e a individuare eventuali asimmetrie o anomalie nei dati.



EDA: Distribuzione delle Caratteristiche - 2

Boxplot: grafici utili per visualizzare la distribuzione dei dati attraverso quartili, mediana e identificare la presenza di potenziali *outlier*.



Scopo: Trasformare i dati grezzi in un formato pulito e ottimale per gli algoritmi di machine learning.

Fasi:

- Eliminazione valori nulli
- Codifica di dati categorici
- Suddivisione del dataset in set di training e test
- Standardizzazione delle features

Preprocessing: Eliminazione valori nulli

Viene eliminata la colonna `Unnamed:32`, composta da solo valori nulli (NaN), tramite il comando `.drop()`.

24	<code>perimeter_worst</code>	569 non-null	float64
25	<code>area_worst</code>	569 non-null	float64
26	<code>smoothness_worst</code>	569 non-null	float64
27	<code>compactness_worst</code>	569 non-null	float64
28	<code>concavity_worst</code>	569 non-null	float64
29	<code>concave points_worst</code>	569 non-null	float64
30	<code>symmetry_worst</code>	569 non-null	float64
31	<code>fractal_dimension_worst</code>	569 non-null	float64

Preprocessing: Codifica Dati Categorici

Gli algoritmi di Machine Learning operano esclusivamente su valori numerici, quindi la variabile target `diagnosis` è stata **codificata** numericamente utilizzando `LabelEncoder()`:

- maligno (**M**) è stato mappato a 1.
- benigno (**B**) è stato mappato a 0.

```
LEncoder = LabelEncoder()  
  
df['diagnosis'] = LEncoder.fit_transform(df['diagnosis'])
```

Preprocessing: Suddivisione Train/Test

Scopo: garantire che l'addestramento dell'algoritmo di classificazione possa essere generalizzato efficacemente a nuovi dati.

Il dataset è stato suddiviso in:

- **Training Set:** 80% delle istanze, usato per addestrare i modelli.
- **Test Set:** 20% delle istanze, usato per valutare le performance finali.

È stato impostato `random_state=42` per garantire che la suddivisione sia la stessa in esecuzioni successive.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,  
                                                    test_size=0.2, random_state=42)
```

Preprocessing: Standardizzazione dei Dati

- Al dataset viene applicata la **standardizzazione** tramite `StandardScaler`, una tecnica che trasforma le variabili in modo da avere media pari a 0 e deviazione standard pari a 1.
- Questo tipo di trasformazione è particolarmente indicato per algoritmi sensibili alla scala dei dati, come SVM.

```
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Viene applicato il metodo `.fit_transform()` sul training set e `.transform()` sul test set per evitare **data leakage**.

Algoritmi scelti per classificare il dataset in esame:

Support Vector Classifier (SVC):

- potente algoritmo di classificazione binaria appartenente alle SVM (Support Vector Machines)
- ottime prestazioni anche con dataset di dimensioni moderate

Decision Tree Classifier (DT):

- modello ad albero, quindi interpretabile e versatile
- presenta il rischio di **overfitting** con dataset di dimensioni ridotte, ma è possibile *contenere* questo rischio grazie alla possibilità di regolare parametri come `max_depth`, `min_samples_split` e `min_samples_leaf`.

- Viene implementata la `GridSearchCV()` con **5-fold cross-validation**. Questo approccio permette di esplorare diverse combinazioni di iperparametri e trovare quella che massimizza le performance del modello.
- **"Media punteggi CV=5"** rappresenta la media dei punteggi di cross-validation (in questo caso, a 5 fold) ottenuti dal modello con la combinazione di iperparametri ottimale, secondo la metrica "accuracy".
- Il parametro `class_weight='balanced'` è stato utilizzato in entrambi i modelli per compensare il leggero sbilanciamento della variabile target.

SVC (Support Vector Classifier) - 1

Principio: Cerca l'iperpiano ottimo che separa le classi massimizzando il margine tra i punti più vicini di ciascuna classe (support vectors).

Parametri di SVC:

- **C – Regularizzazione:** controlla il compromesso tra margine ampio e classificazione corretta dei punti.
- **kernel – Tipo di kernel.**
- **gamma – Coefficiente del kernel:** quanto ogni punto impatta sulla curva di decisione.
- **n_jobs=-1:** Indica al processo di utilizzare tutti i core disponibili per velocizzare.

SVC (Support Vector Classifier) - 2

```
param_grid_svc = {  
    'C': [0.1, 1, 10, 100],  
    'kernel': ['linear', 'rbf', 'poly'],  
    'gamma': ['scale', 'auto', 1, 0.1, 0.01],  
}  
grid_search_svc = GridSearchCV(svc, param_grid_svc, cv=5,  
    n_jobs=-1)  
grid_search_svc.fit(X_train, y_train)
```

Risultati ottimali da GridSearch:

Parametri migliori: {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}

Media punteggi CV=5: 0.9802197802197803

SVC Train Accuracy: 0.9846

SVC Test Accuracy: 0.9649

Decision Tree Classifier - 1

Principio: Costruisce un modello predittivo con una struttura ad albero, dove ogni *nodo* interno rappresenta un test su un attributo, ogni *ramo* rappresenta l'esito del test, e ogni nodo *foglia* rappresenta una decisione.

! Facilmente interpretabile e visualizzabile, ma può essere soggetto a overfitting.

Iperparametri del DT Classifier:

- `criterion` – **Criterio di divisione.**
- `max_depth` – **Massima profondità** dell'albero.
Se troppo grande → overfitting. Se troppo piccolo → underfitting.
- `min_samples_split` – Minimo N° di campioni per dividere un **nodo.**
- `min_samples_leaf` – Minimo N° di campioni in una **foglia.**
- `max_features`: N° di **features** che il DT considera a ogni **split**.
Il valore giusto aiuta a prevenire overfitting e migliorare la generalizzazione.

Decision Tree Classifier - 2

```
param_grid_dt = {  
    'criterion': ['gini', 'entropy'],  
    'max_depth': [2, 3, 4, 5, 6],  
    'min_samples_leaf': [2, 3, 4, 5, 6],  
    'min_samples_split': [5, 10, 15],  
    'max_features': ['sqrt', 'log2']  
}  
  
grid_search_dt = GridSearchCV(dt, param_grid_dt, cv=5)  
grid_search_dt.fit(X_train, y_train)
```

Risultati ottimali da GridSearch:

Parametri migliori: {'criterion': 'gini', 'max_depth': 4,
 'max_features': 'log2', 'min_samples_leaf': 6,
 'min_samples_split': 2}

Media punteggi CV=5: 0.9516483516483516

DT Train Accuracy: 0.967

DT Test Accuracy: 0.9737

- **Matrice di Confusione:** tabella che permette di visualizzare le performance dell'algoritmo di classificazione. Mostra i Veri Positivi (TP), Falsi Positivi (FP), Veri Negativi (TN) e Falsi Negativi (FN).
- **Report di Classificazione:** include le metriche Accuratezza, Precisione, Recall, F1-score.
- **Punteggio ROC AUC** (Area Under the Curve): misura la capacità del modello di distinguere tra le classi. Un valore più alto indica una migliore performance.

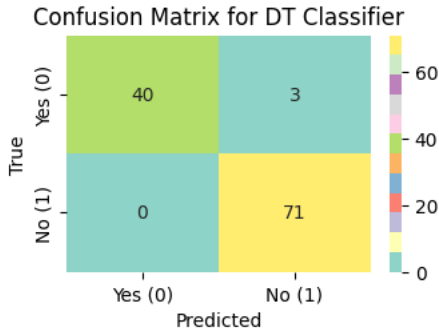
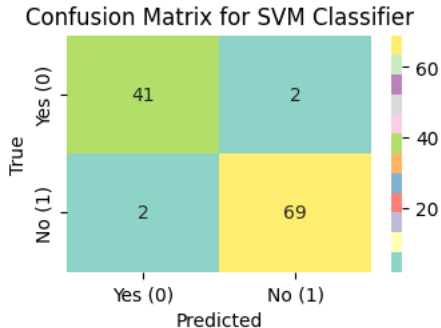
Valutazione: Matrice di Confusione - 1

Una **matrice di confusione** è una tabella nella quale ogni riga rappresenta le istanze della classe *reale*, mentre ogni colonna rappresenta le istanze della classe *predetta*.

- **TP** – True Positive: un'osservazione della classe positiva (1) è correttamente classificata come positiva (1) dal modello.
- **FP** – False Positive: un'osservazione della classe negativa (0) è erroneamente classificata come positiva (1).
- **TN** – True Negative: un'osservazione della classe negativa (0) è correttamente classificata come negativa (0).
- **FN** – False Negative: un'osservazione della classe positiva (1) è erroneamente classificata come negativa (0).

Valutazione: Matrice di Confusione - 2

In questo caso, ogni volta che il modello predice **Yes** (0), indica l'*assenza* (classe negativa) di cellule tumorali (**Healthy**), mentre quando predice **No** (1), indica la *presenza* (classe positiva) di cellule tumorali (**Cancer**).



Il **classification report** permette di visualizzare una panoramica delle metriche di ciascun modello.

- **Accuracy**: previsioni corrette sul totale.
- **Precision**: veri positivi tra tutti i positivi predetti.
- **Recall** (Sensibilità): veri positivi tra tutti i positivi reali (importante in ambito medico).
- **F1-Score**: Media armonica pesata tra precisione e recall, utile per *dataset sbilanciati*.

Valutazione: Classification Report - 2

Classification Report del modello SVC:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	71
1	0.95	0.95	0.95	43
accuracy			0.96	114
macro avg	0.96	0.96	0.96	114
weighted avg	0.96	0.96	0.96	114

Classification Report del modello DecisionTreeClassifier:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	71
1	1.00	0.93	0.96	43
accuracy			0.97	114
macro avg	0.98	0.97	0.97	114
weighted avg	0.97	0.97	0.97	114

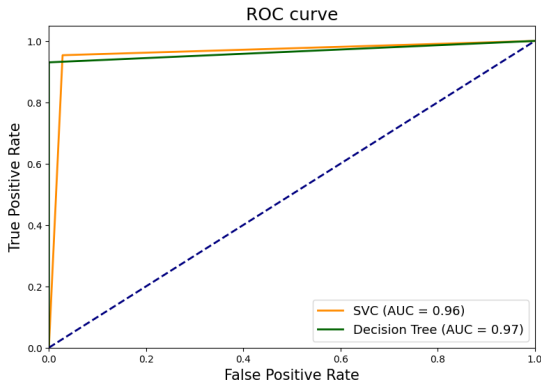
Valutazione: ROC Curve e AUC

- **ROC Curve:** rappresenta la capacità di un classificatore binario di discriminare tra classi al variare della soglia.
- **AUC:** misura l'area sotto la curva ROC. Un valore vicino a 1 indica un'eccellente capacità di separazione delle classi.

● **SVC AUC:** 0.9627

● **DT AUC:** 0.9651

Il Decision Tree mostra un AUC leggermente superiore, suggerendo una capacità discriminativa marginalmente migliore.



- Entrambi i modelli (SVC e Decision Tree) hanno raggiunto accuratèzze e valori AUC **molto elevati**.
- **Decision Tree leggermente superiore**: ha mostrato una recall perfetta per la classe "maligno" e un AUC marginalmente più alto. Questo lo rende potenzialmente più adatto in contesti clinici dove minimizzare i falsi negativi è fondamentale.
- Tuttavia, i risultati eccezionalmente alti su un dataset di sole 569 istanze suggeriscono un potenziale rischio di **overfitting**. I modelli potrebbero aver imparato troppo bene i dati di training, compromettendo la *generalizzazione* su nuovi dati.

Grazie per l'attenzione

Appendici: Decision Tree

