

- Oleksandra Golub (856706)
- Giovanni Bishara (869532)
- Federico Brambilla (886046)



Progetto Diabetes — Prediction

Machine Learning

- Features & Target
- Casting & Controllo & Distribuzione
- Statistiche del dataset
- Statistiche del dataset: classe "Sani"
- Statistiche del dataset: classe "Diabetici"
- Statistiche del dataset: Asimmetria e Curtosi
- Preprocessing: Gestione dei valori errati
- Preprocessing: Gestione degli outlier - Assimetria
- Preprocessing: Gestione degli outlier - Curtosi
- Preprocessing: Standardizzazione delle feature
- Preprocessing: Bilanciamento del dataset
- Esplorazione grafica dei dati
- Matrice di correlazione
- PCA: Grafici della varianza spiegata
- PCA: Grafici delle componenti principali
- Training, Validation e Test Set
- Modello baseline
- Modelli di apprendimento: Introduzione
- Modelli di apprendimento: Decision Tree
- Modelli di apprendimento: Random Forest
- Modelli di apprendimento: SVM
- Modelli di apprendimento: Rete Neurale
- Valutazione: Metriche di Performance
- Valutazione: Intervalli di Confidenza
- Valutazione: Curve ROC
- Valutazione: Tempi di Training

Features & Target

- **Tipologia:** un problema di classificazione binaria
- **Obiettivo:** predire se un paziente è diabetico o meno
- **Numero di istanze:** 768 pazienti
- **Composizione:** 8 features numeriche + 1 target categorica

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

[1] Lara311. "Diabetes Dataset Using Many Medical Metrics." Kaggle.

Disponibile all'indirizzo: <https://www.kaggle.com/datasets/lara311/diabetes-dataset-using-many-medical-metrics> (Ultimo accesso: 21 febbraio 2025).



Casting dei dati

- Target diventa categorica

Pregnancies	int64
Glucose	int64
BloodPressure	int64
SkinThickness	int64
Insulin	int64
BMI	float64
DiabetesPedigreeFunction	float64
Age	int64
Outcome	category

Controllo iniziale

- Nessun valore nullo

Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

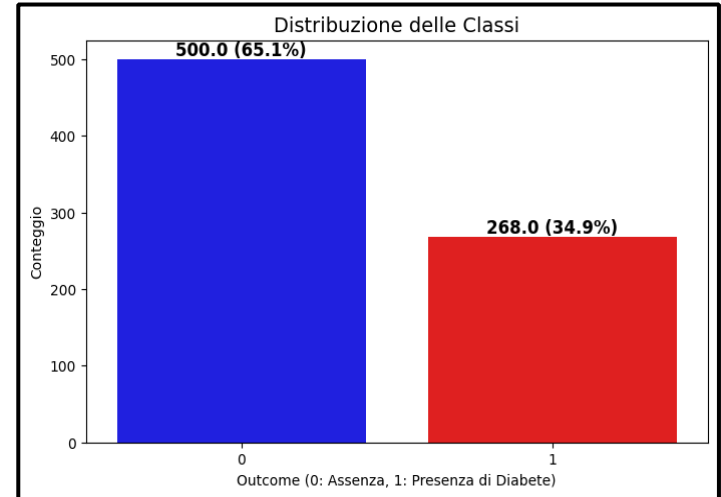
- Nessun duplicato

```
df.duplicated().sum()
```

0

Distribuzione delle classi

- Dataset non bilanciato



Statistiche del dataset



Evidente anomalia nei dati:

- I valori 0 in Glucose, BloodPressure e BMI non sono fisiologicamente possibili.
- Altre features, come Insulin e SkinThickness, potrebbero contenere veri valori nulli, ma bisogna verificare più dettagliatamente le colonne.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000

In rosso:

- valori chiaramente impossibili.
- da correggere nel preprocessing.

In blu:

- valori potenzialmente nulli.
- da analizzare prima di correggerli nel preprocessing.

Statistiche del dataset: classe "Sani"



Evidente anomalia nei dati:

- L'insulina a digiuno non dovrebbe **mai essere zero** in un paziente sano.
- Lo spessore cutaneo non dovrebbe **mai essere zero** in un paziente sano.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	500.000000	500.0000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	3.298000	109.9800	68.184000	19.664000	68.792000	30.304200	0.429734	31.190000
std	3.017185	26.1412	18.063075	14.889947	98.865289	7.689855	0.299085	11.667655
min	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000
25%	1.000000	93.0000	62.000000	0.000000	0.000000	25.400000	0.229750	23.000000
50%	2.000000	107.0000	70.000000	21.000000	39.000000	30.050000	0.336000	27.000000
75%	5.000000	125.0000	78.000000	31.000000	105.000000	35.300000	0.561750	37.000000
max	13.000000	197.0000	122.000000	60.000000	744.000000	57.300000	2.329000	81.000000

In **rosso**:

- valori chiaramente impossibili.
- da correggere nel preprocessing.

In **blu**:

- valori potenzialmente nulli.
- da analizzare prima di correggerli nel preprocessing.

Statistiche del dataset: classe "Diabetici"

Evidente anomalia nei dati:

- I **valori 0** in Insulin e SkinThickness nei diabetici potrebbero essere **errori o dati mancanti**.
- Tuttavia, **sono plausibili** in alcuni casi (es. diabete di tipo 1 e/o distrofia lipoatrofica), quindi **non verranno modificati** nel preprocessing.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	268.000000	268.000000	268.000000	268.000000	268.000000	268.000000	268.000000	268.000000
mean	4.865672	141.257463	70.824627	22.164179	100.335821	35.142537	0.550500	37.067164
std	3.741239	31.939622	21.491812	17.679711	138.689125	7.262967	0.372354	10.968254
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.088000	21.000000
25%	1.750000	119.000000	66.000000	0.000000	0.000000	30.800000	0.262500	28.000000
50%	4.000000	140.000000	74.000000	27.000000	0.000000	34.250000	0.449000	36.000000
75%	8.000000	167.000000	82.000000	36.000000	167.250000	38.775000	0.728000	44.000000
max	17.000000	199.000000	114.000000	99.000000	846.000000	67.100000	2.420000	70.000000

In **rosso**:

- valori chiaramente impossibili.
- da correggere nel preprocessing.

In **blu**:

- valori potenzialmente nulli.
- da analizzare prima di correggerli nel preprocessing.

Statistiche del dataset: Asimmetria e Curtosi



Evidente anomalia nei dati:

- Lavorare con le variabili con **fortemente asimmetriche**, applicando una trasformazione.
- Lavorare con le **variabili leptocurtiche**, per controllare e trattare eventuali outlier.

Skewness (Asimmetria):

Pregnancies	0.901674
Glucose	0.173754
BloodPressure	-1.843608
SkinThickness	0.109372
Insulin	2.272251
BMI	-0.428982
DiabetesPedigreeFunction	1.919911
Age	1.129597

Kurtosis (Curtosi):

Pregnancies	0.159220
Glucose	0.640780
BloodPressure	5.180157
SkinThickness	-0.520072
Insulin	7.214260
BMI	3.290443
DiabetesPedigreeFunction	5.594954
Age	0.643159

In **rosso**:

- valori sconsigliati.
- da correggere nel preprocessing.

In **blu**:

- valori potenzialmente anomali.
- da analizzare prima di correggerli nel preprocessing.

Preprocessing: Gestione dei valori errati

- Sostituiti i valori 0 con NaN per Glucose, BloodPressure, BMI, Insulin (solo nei pazienti sani) e SkinThickness (solo nei pazienti sani).

Valori nulli dopo aver sostituito 0 con NaN:

Pregnancies	0
Glucose	5
BloodPressure	35
SkinThickness	139
Insulin	236
BMI	11
DiabetesPedigreeFunction	0
Age	0
Outcome	0

dtype: int64

Numero totale di righe con almeno un valore mancante: 257

- Per l'imputazione, viene utilizzato **IterativeImputer**, che **stima i valori NaN** in base alle altre feature con un modello iterativo.

Statistiche nuove del dataset

	Pregnancies	Glucose	BloodPressure	kinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.612547	72.434965	24.611359	109.476318	32.435347	0.471876	33.240885
std	3.369578	30.453089	12.150487	12.921637	104.379368	6.880494	0.331329	11.760232
min	0.000000	44.000000	24.000000	0.000000	0.000000	18.200000	0.078000	21.000000
25%	1.000000	99.750000	64.000000	18.000000	50.000000	27.500000	0.243750	24.000000
50%	3.000000	117.000000	72.000000	25.000000	92.000000	32.021014	0.372500	29.000000
75%	6.000000	140.250000	80.000000	33.000000	140.000000	36.600000	0.626250	41.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000

Statistiche nuove del dataset: classe "Sani"

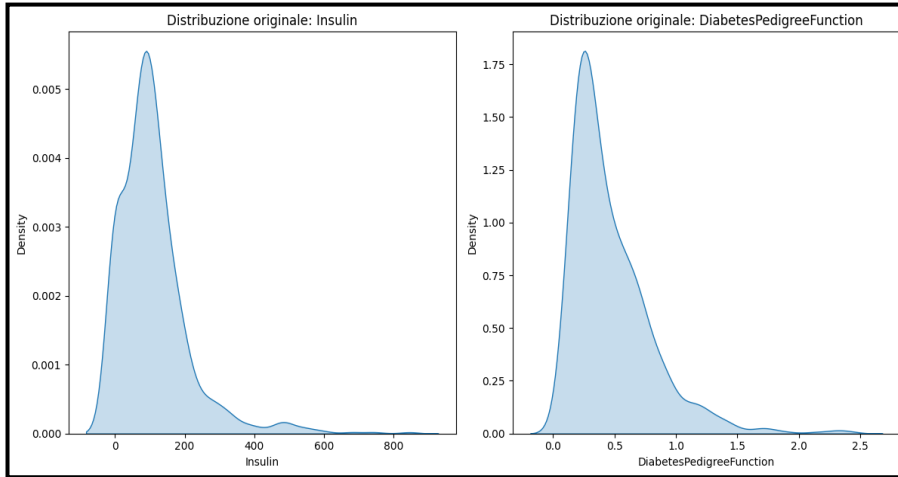
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	3.298000	110.644242	70.814448	25.923047	114.375625	30.868812	0.429734	31.190000
std	3.017185	24.705049	11.942228	9.189934	79.910456	6.503771	0.299085	11.667655
min	0.000000	44.000000	24.000000	7.000000	1.902988	18.200000	0.078000	21.000000
25%	1.000000	93.000000	63.500000	19.000000	69.740667	25.750000	0.229750	23.000000
50%	2.000000	107.000000	70.000000	25.000000	95.045366	30.342732	0.336000	27.000000
75%	5.000000	125.000000	78.000000	31.250000	132.002979	35.300000	0.561750	37.000000
max	13.000000	197.000000	122.000000	60.000000	744.000000	57.300000	2.329000	81.000000

Preprocessing: Gestione degli outlier - Assimetria

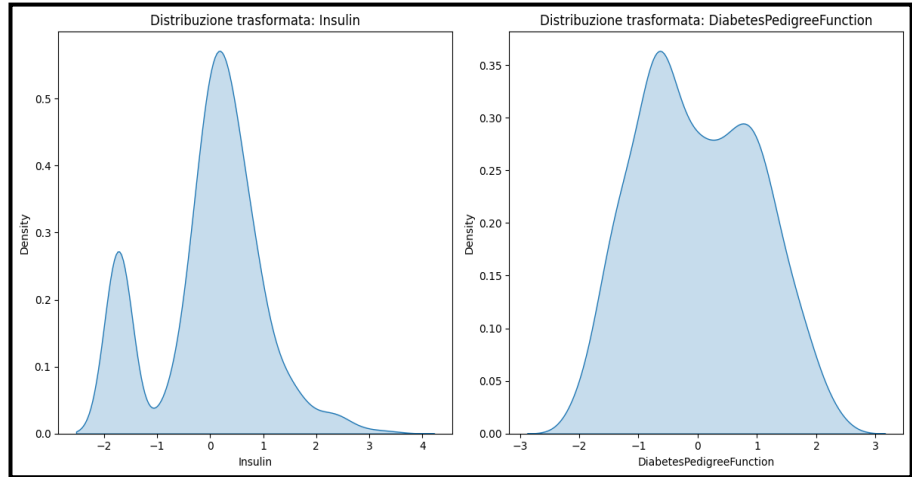


- Applicato **PowerTransformer** per rendere le variabili Insulin e DiabetesPedigreeFunction **più simmetriche e vicine a una distribuzione normale**.

Distribuzione originale



Distribuzione trasformata



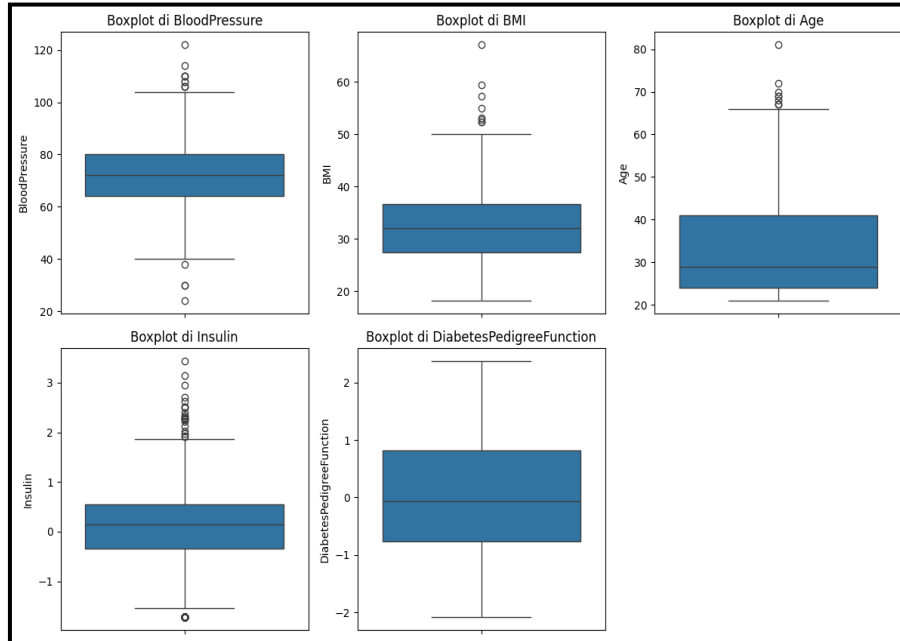
- Questo **migliora la stabilità della varianza e le prestazioni di algoritmi sensibili alla distribuzione dei dati**, come MLP, regressione lineare e SVM. Dopo la trasformazione, entrambe le distribuzioni risultano **più bilanciate**.

Preprocessing: Gestione degli outlier - Curtosi

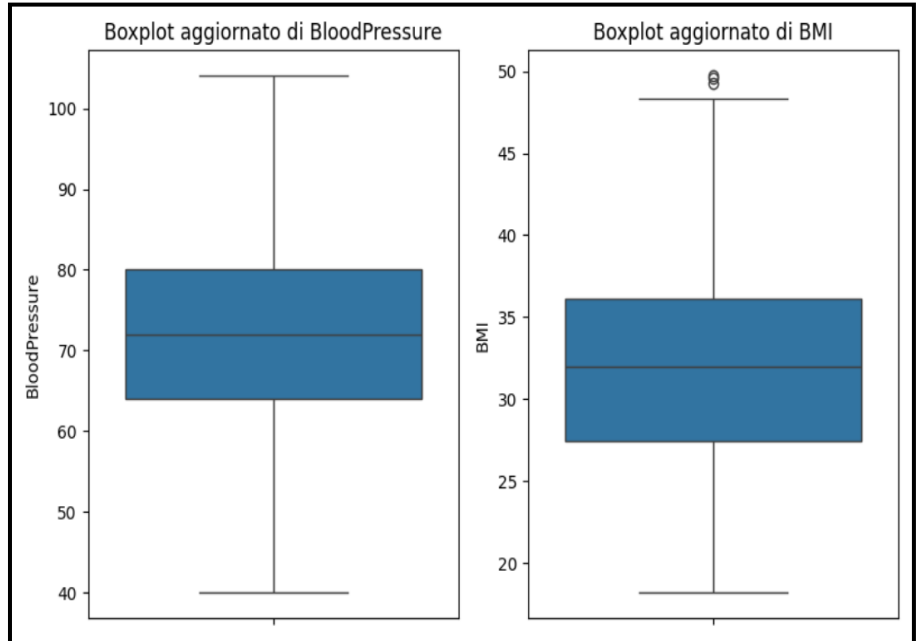


- Applicato il **metodo IQR** (Interquartile Range) per identificare e rimuovere i **valori estremi** in BloodPressure e BMI, mantenendo solo quelli **entro il range accettabile**.
- Per **altre features** gli outlier sono **plausibili** e non sono stati rimossi.

Boxplot originali



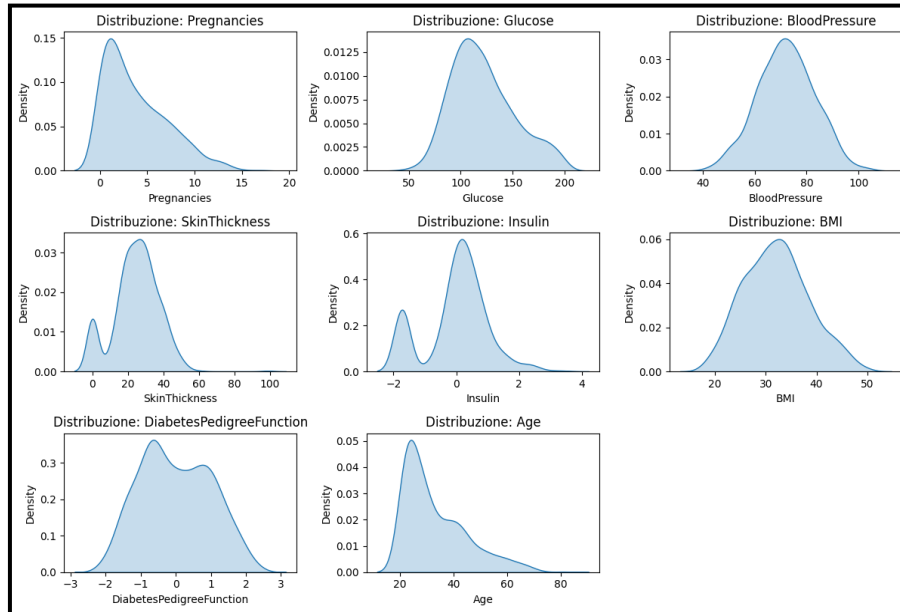
Boxplot trasformati



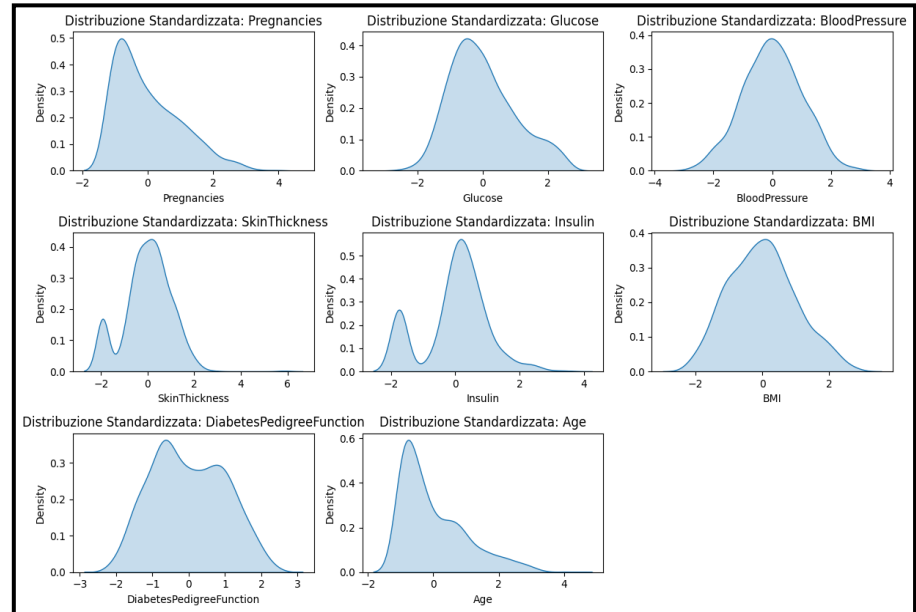
Preprocessing: Standardizzazione delle feature

- Applicato **StandardScaler** per trasformare i dati in modo che ogni feature abbia **media 0** e **deviazione standard 1**. Viene applicato per eliminare il problema delle **scale diverse**.

Distribuzione originale



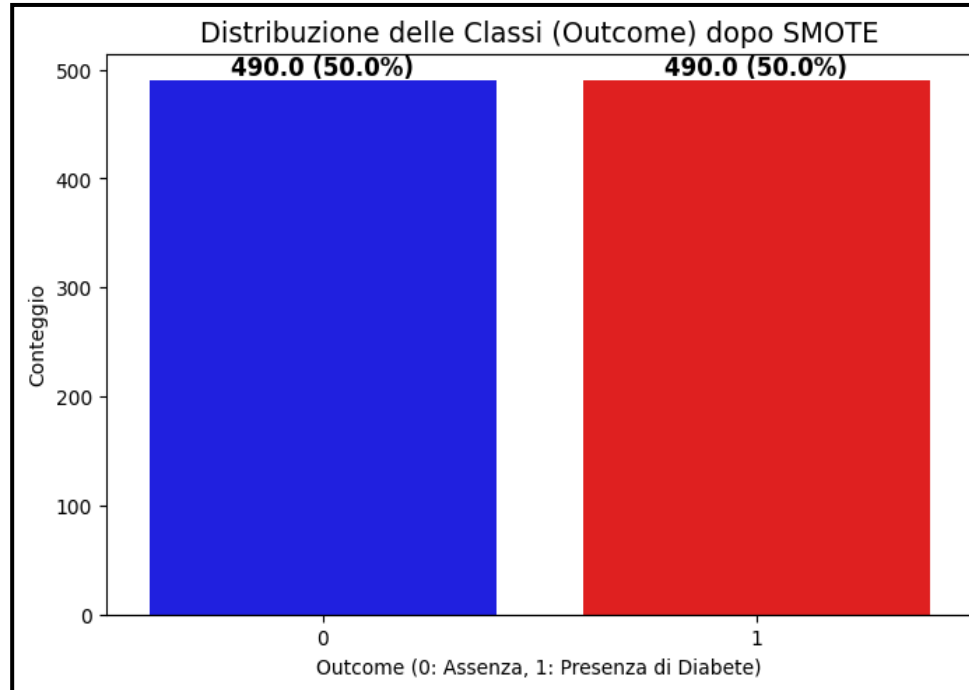
Distribuzione standardizzata



Preprocessing: Bilanciamento del dataset



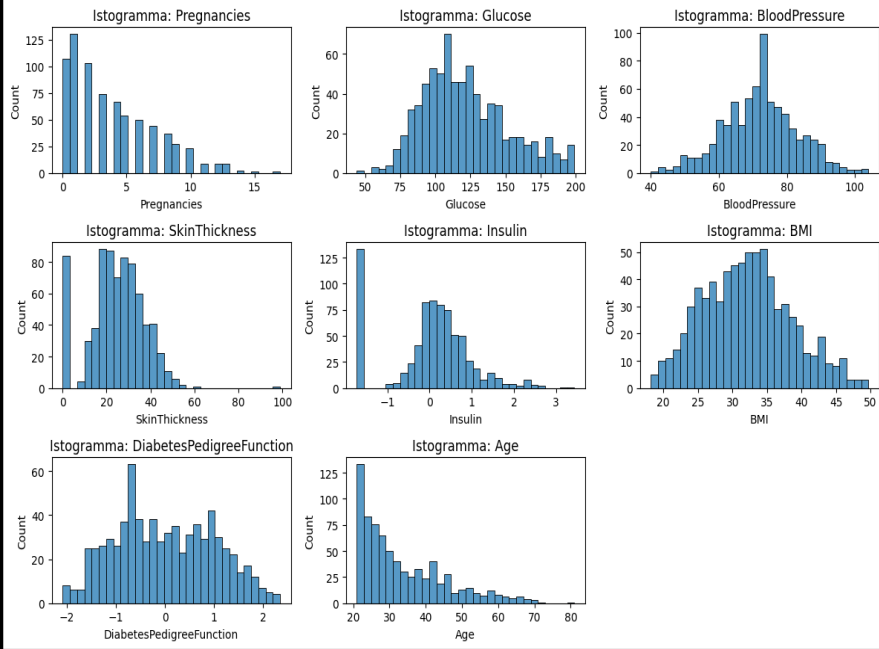
- Applicato **SMOTE** per bilanciare il dataset sbilanciato.
- Genera campioni sintetici per la classe minoritaria (nel nostro caso, i pazienti diabetici).
- Aiuta a **evitare problemi di overfitting** che possono verificarsi quando si duplicano i dati reali.



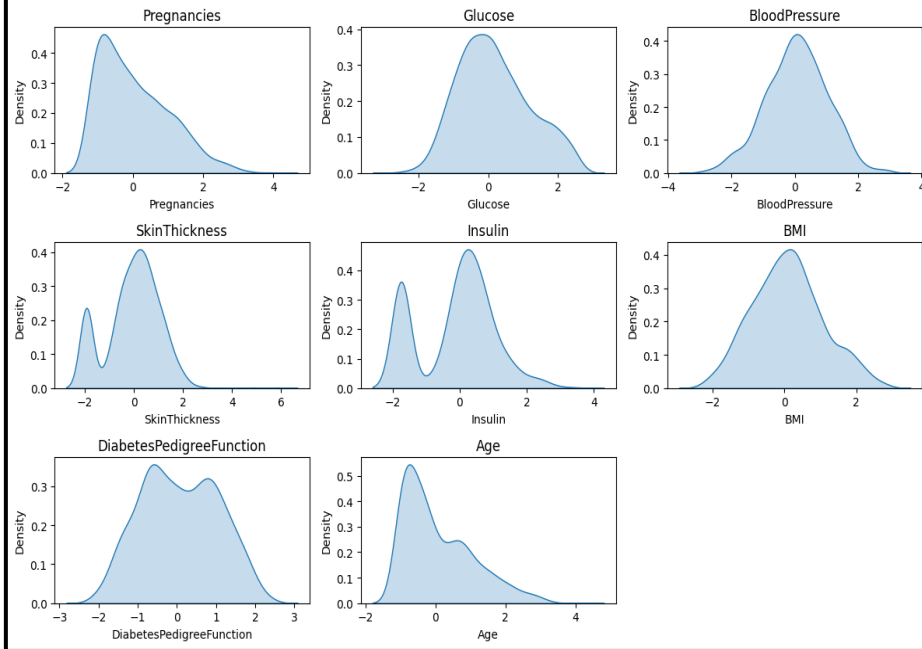
Esplorazione grafica dei dati

- **Istogrammi delle feature originali** mostrano la distribuzione delle feature dopo la gestione dei valori errati e/o anomali, ma prima della standardizzazione e del bilanciamento del dataset.
- **KDE (Kernel Density Estimation)** fornisce una rappresentazione continua della densità dei dati, mostrando come la standardizzazione uniformi le distribuzioni e SMOTE riequilibri le classi.

Distribuzione delle Feature Originali



Distribuzione delle Feature Preprocessate



Matrice di correlazione

Correlazioni forti:

- Pregnancies & Age (0.52)
- SkinThickness & Insulin (0.53)

Correlazioni moderate:

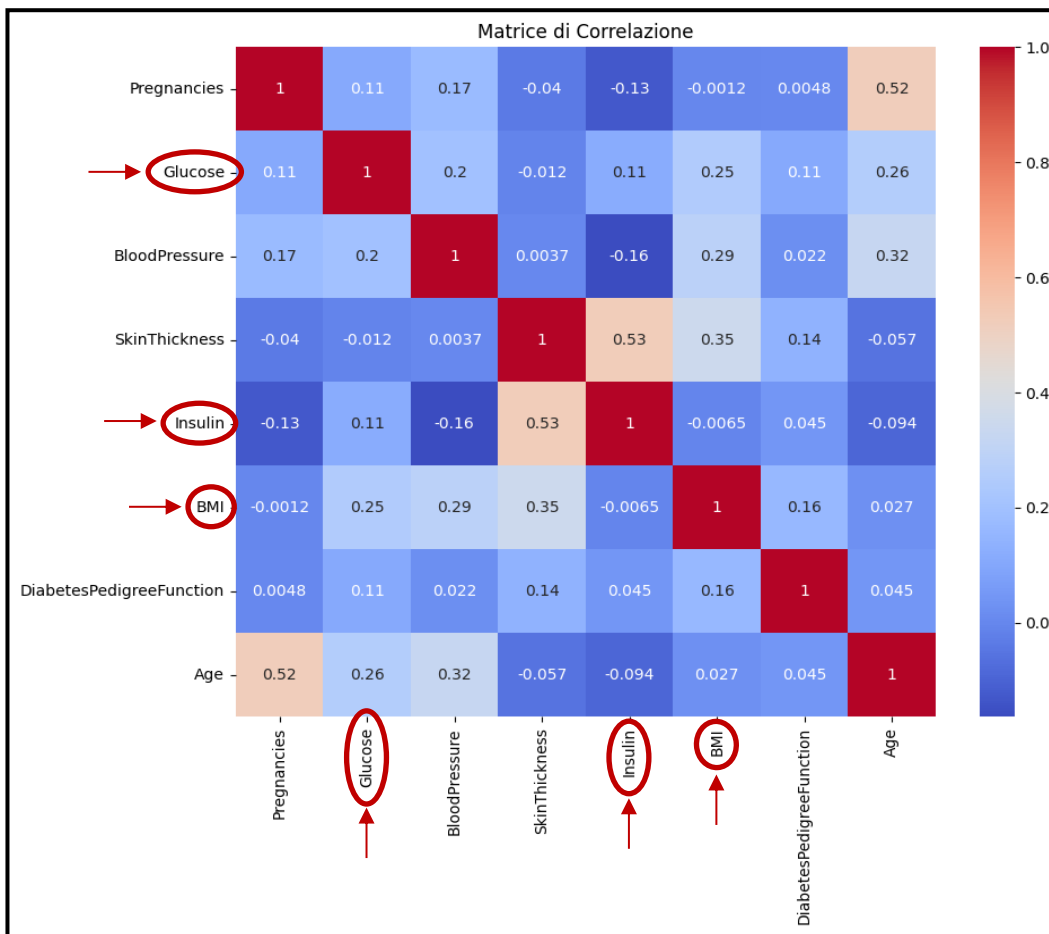
- BMI & SkinThickness (0.35)
- BloodPressure & BMI (0.29)

Correlazioni deboli:

- Glucose & BMI (0.25)

Osservazioni principali:

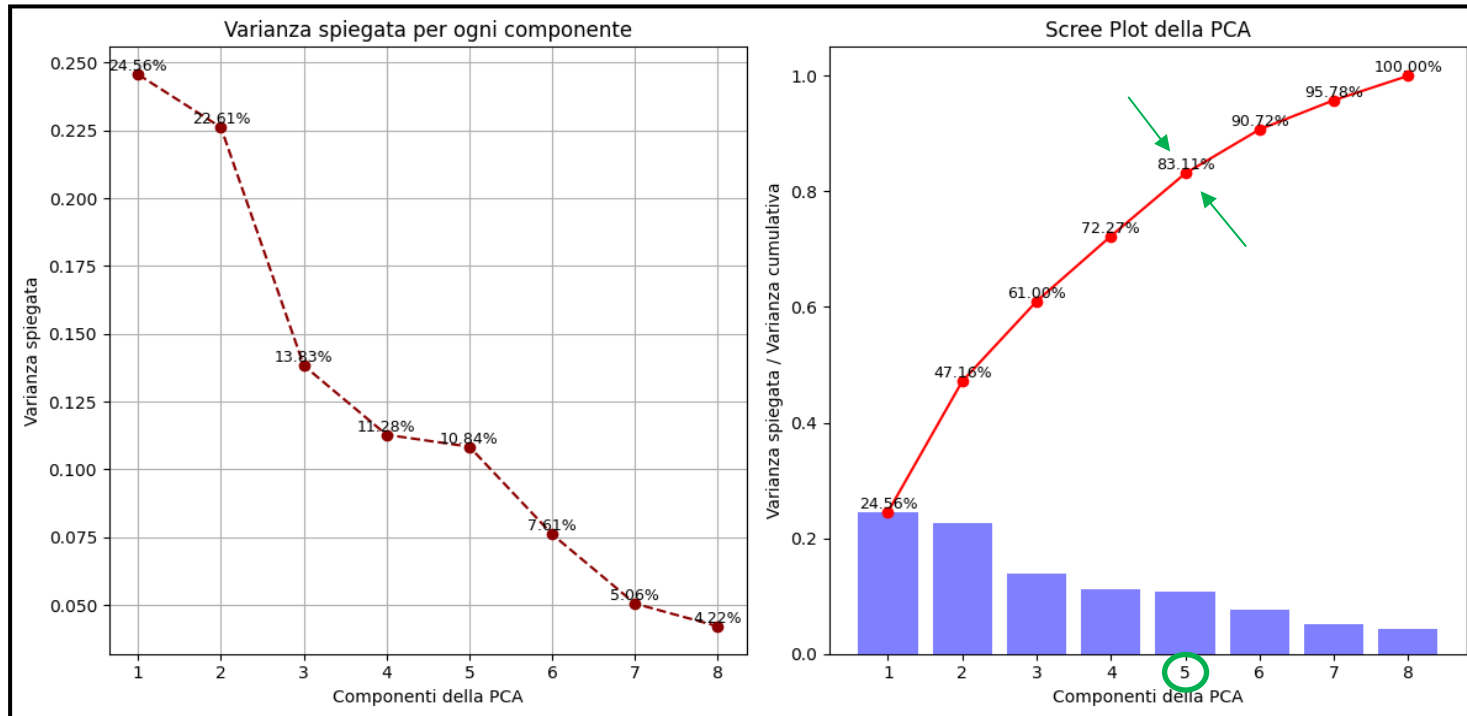
- **Glucose**, **BMI** e **Insulin** mostrano correlazioni significative con altre feature, suggerendo un impatto rilevante sulla classificazione del diabete.



PCA: Grafici della varianza spiegata

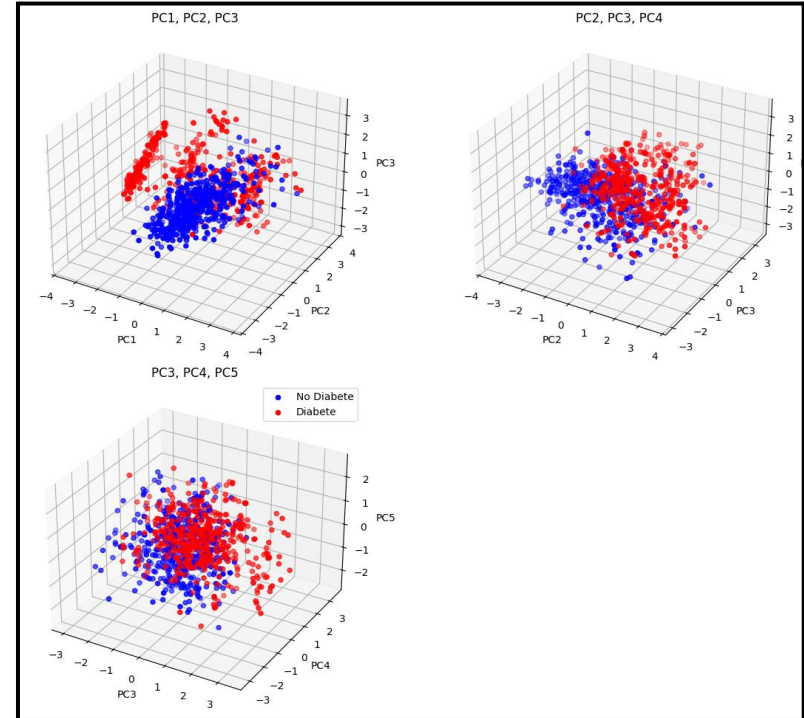
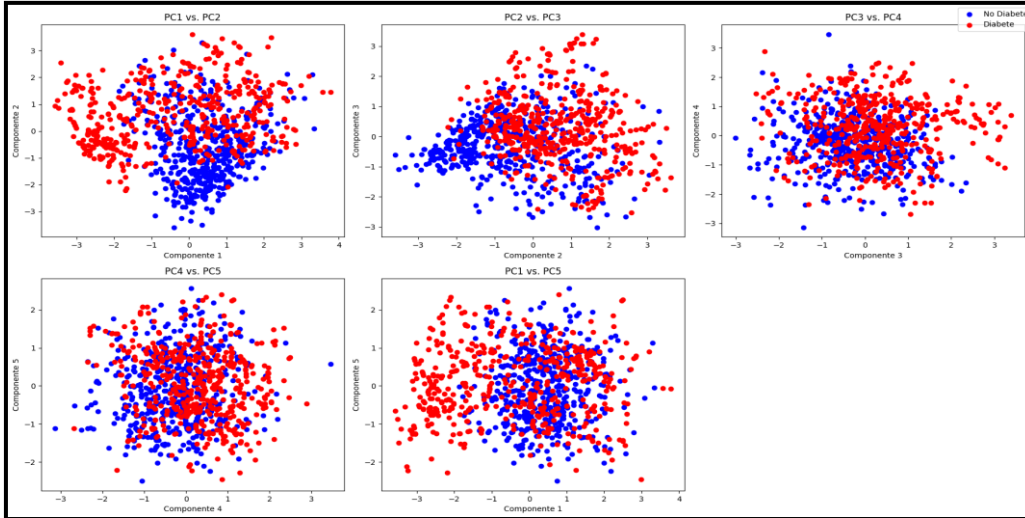
Questi due grafici aiutano a determinare quante componenti principali mantenere.

- Grafico a sinistra mostra quanto ciascuna componente contribuisce alla varianza totale.
- Grafico a destra indica quanta informazione viene mantenuta sommando le componenti principali.



PCA: Grafici delle componenti principali (2D e 3D)

- Le classi sono spesso **sovrapposte**, suggerendo che PCA **non fornisce una chiara separazione**.
- L'informazione utile alla classificazione non è concentrata solo nelle prime componenti, ma **distribuita tra più dimensioni**.



Decisione finale:

- Utilizzare **tutte le feature originali** per ottenere la massima accuratezza.

Esempio: PCA migliora un Decision Tree? **No!**

Accuratezza Validation Set senza PCA: 0.8844
 Accuratezza Validation Set con PCA: 0.7891
 Accuratezza Test Set senza PCA: 0.8707
 Accuratezza Test Set con PCA: 0.8163

Training, Validation e Test Set

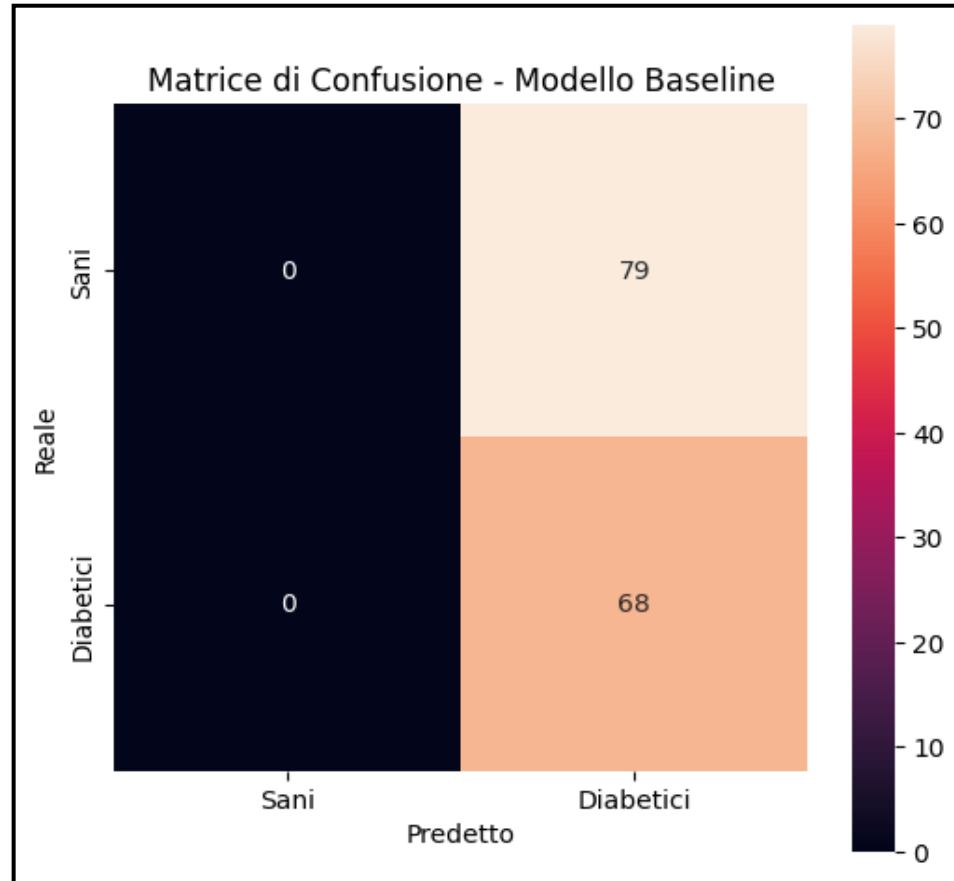
Suddivisione del dataset	Metriche di performance	Cross-validation per una valutazione avanzata
<div> <p>Train Set: (686, 8) Validation Set: (147, 8) Test Set: (147, 8)</p> </div> <ul style="list-style-type: none"> ➤ 70% Training Set, utilizzato per l'addestramento del modello. ➤ 15% Validation Set, usato per la selezione del modello e l'ottimizzazione degli iperparametri. ➤ 15% Test Set, utilizzato per la valutazione finale delle prestazioni su dati non visti. 	<ul style="list-style-type: none"> ➤ Accuracy, ovvero la percentuale di predizioni corrette. ➤ Precision, ovvero la percentuale di istanze classificate come positive che sono effettivamente tali. ➤ Recall, ovvero la capacità del modello di identificare tutte le istanze positive. ➤ F1-score, ovvero la media armonica tra Precision e Recall. 	<p>Obiettivo: Migliorare la generalizzazione del modello e ridurre il rischio di overfitting.</p> <ol style="list-style-type: none"> 1. Il dataset viene suddiviso in 10 fold, alternando training e test set. 2. Per ogni fold, vengono calcolate le metriche di performance. 3. Viene stimata la media delle metriche su tutti i fold per una valutazione più stabile. 4. Si calcola un intervallo di confidenza al 95% con la distribuzione t di Student per stimare l'incertezza sulle performance.

Modello baseline

- Il modello non distingue i pazienti sani, portando a un **alto numero di falsi positivi**.
- La bassa accuratezza (46.26%) riflette che il modello baseline non è utile in quanto **non riesce a fare predizioni corrette per la classe più rara** (ovvero i pazienti sani).

Accuracy del modello baseline: 0.46258503401360546

- L'utilizzo di un modello più avanzato è **necessario** per una classificazione più accurata.



Modelli di apprendimento: Introduzione

Modelli:

Modelli considerati:

- **Decision Tree** (DT)
- **Random Forest** (RF)
- **Support Vector Machine** (SVM)
- **Reti Neurali** (MLP - Multi-Layer Perceptron)

Modelli considerati, ma scartati:

- **Regression Logistic** (RL)
- **Naive Bayes** (NB)

Perché alcuni modelli sono stati scartati?

- RL e NB funzionano bene se le relazioni tra le feature e l'output sono **lineari**.
- Nel nostro caso le relazioni nei dati **non sono lineari** e per questo altri modelli **funzionano meglio!**

Per ogni modello si esegue:

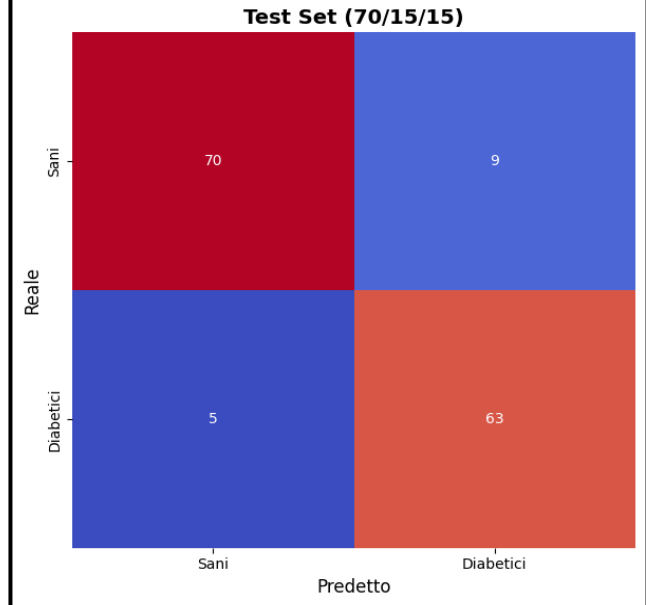
1. **Addestramento e ottimizzazione** con la scelta dei migliori iperparametri trovati tramite **RandomizedSearchCV**.
2. **Cross-validation** per ottenere una stima più robusta delle performance del modello su **dati non visti**.
3. **Controllo dell'overfitting** analizzando il **gap** tra training e test.
4. **Analisi delle performance** tramite **matrici di confusione** e **curve ROC**.

Modelli di apprendimento: Decision Tree

```
DecisionTreeClassifier
DecisionTreeClassifier(ccp_alpha=0.01, max_depth=15, min_samples_split=20,
random_state=42)
```

Dopo l'addestramento del modello con **migliori iperparametri** trovati si ottengono le seguenti conclusioni:

- Con il 95% di probabilità, la vera accuratezza del modello sta tra 80.7% e 87.7% (con la media a 84.2%).
- La differenza tra training, validation e test è minima, indicando che il modello **non soffre di overfitting** e generalizza bene sui dati nuovi.



	Media (70/15/15)	95% CI (70/15/15)
Accuracy	0.842477	(0.807875, 0.877078)
Precision	0.846883	(0.814754, 0.879011)
Recall	0.842512	(0.808361, 0.876663)
F1 Score	0.841760	(0.806647, 0.876872)

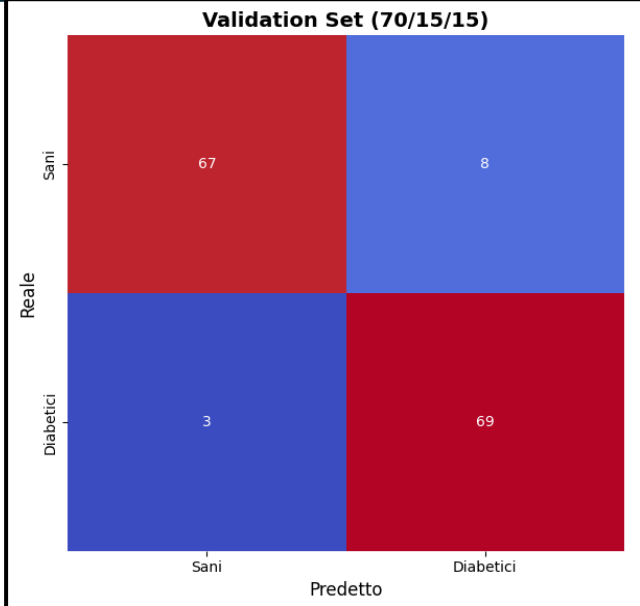
	Accuracy (70/15/15)	Precision (70/15/15)	Recall (70/15/15)	F1 Score (70/15/15)	Accuracy Gap (Train - Test)	F1 Score Gap (Train - Test)
Train	0.890671	0.893526	0.889821	0.890279	-0.014091	-0.014267
Validation	0.897959	0.903199	0.899167	0.897789	-0.014091	-0.014267
Test	0.904762	0.904167	0.906273	0.904545	-0.014091	-0.014267

Modelli di apprendimento: Random Forest

```
RandomForestClassifier
RandomForestClassifier(class_weight='balanced', criterion='entropy',
max_depth=10, max_features=0.8, min_samples_leaf=2,
min_samples_split=10, n_estimators=200, n_jobs=-1,
oob_score=True, random_state=42)
```

Dopo l'addestramento del modello con **migliori iperparametri** trovati si ottengono le seguenti conclusioni:

- Con il 95% di probabilità, la vera accuratezza del modello sta tra 85.8% e 90.5% (con la media a 88.2%).
- La differenza tra training, validation e test è minima, indicando che il modello **non soffre di overfitting** e generalizza bene sui dati nuovi.



	Accuracy (70/15/15)	Precision (70/15/15)	Recall (70/15/15)	F1 Score (70/15/15)	Accuracy Gap (Train - Test)	F1 Score Gap (Train - Test)
Train	0.972303	0.972269	0.972321	0.972293	0.081147	0.08175
Validation	0.925170	0.926623	0.925833	0.925156	0.081147	0.08175
Test	0.891156	0.890544	0.890544	0.890544	0.081147	0.08175

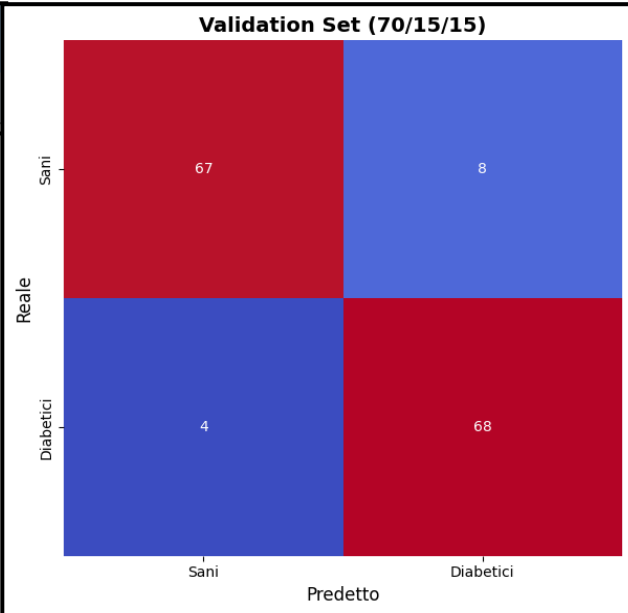
	Media (70/15/15)	95% CI (70/15/15)
Accuracy	0.881905	(0.858098, 0.905713)
Precision	0.885755	(0.863032, 0.908478)
Recall	0.881887	(0.85797, 0.905804)
F1 Score	0.881501	(0.857508, 0.905494)

Modelli di apprendimento: Support Vector Machine

```
SVC
SVC(C=1.4391207615728072, probability=True, random_state=42)
```

Dopo l'addestramento del modello con **migliori iperparametri** trovati si ottengono le seguenti conclusioni:

- Con il 95% di probabilità, la vera accuratezza del modello sta tra 81.4% e 87.4% (con la media a 84.4%).
- La differenza tra training, validation e test è minima, indicando che il modello **non soffre di overfitting** e generalizza bene sui dati nuovi.



	Media (70/15/15)	95% CI (70/15/15)
Accuracy	0.843990	(0.813625, 0.874354)
Precision	0.847340	(0.81724, 0.877441)
Recall	0.844237	(0.814103, 0.874372)
F1 Score	0.843612	(0.813181, 0.874043)

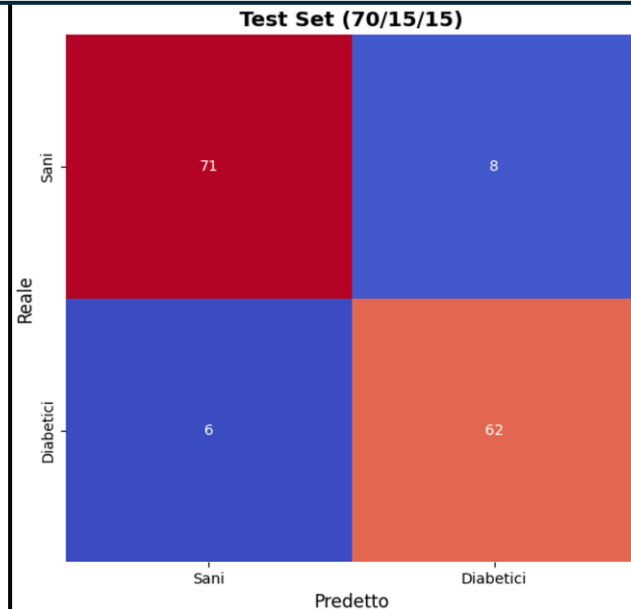
	Accuracy (70/15/15)	Precision (70/15/15)	Recall (70/15/15)	F1 Score (70/15/15)	Accuracy Gap (Train - Test)	F1 Score Gap (Train - Test)
Train	0.902332	0.903687	0.902976	0.902316	0.017979	0.018738
Validation	0.918367	0.919199	0.918889	0.918364	0.017979	0.018738
Test	0.884354	0.884049	0.883191	0.883578	0.017979	0.018738

Modelli di apprendimento: Rete Neurale

```
MLPClassifier(alpha=0.0330893825622149, batch_size=64, early_stopping=True,
              hidden_layer_sizes=(200, 100),
              learning_rate_init=0.013203823484477885, max_iter=1000,
              momentum=0.8331949117361643, n_iter_no_change=15,
              nesterovs_momentum=False, random_state=42, solver='lbfgs',
              tol=0.00026877998160001695,
              validation_fraction=0.1162522284353982)
```

Dopo l'addestramento del modello con migliori iperparametri trovati si ottengono le seguenti conclusioni:

- Con il 95% di probabilità, la vera accuratezza del modello sta tra 84.1% e 86.0% (con la media a 84.1%).
- La differenza tra training, validation e test è minima, indicando che il modello **non soffre di overfitting** e generalizza bene sui dati nuovi.

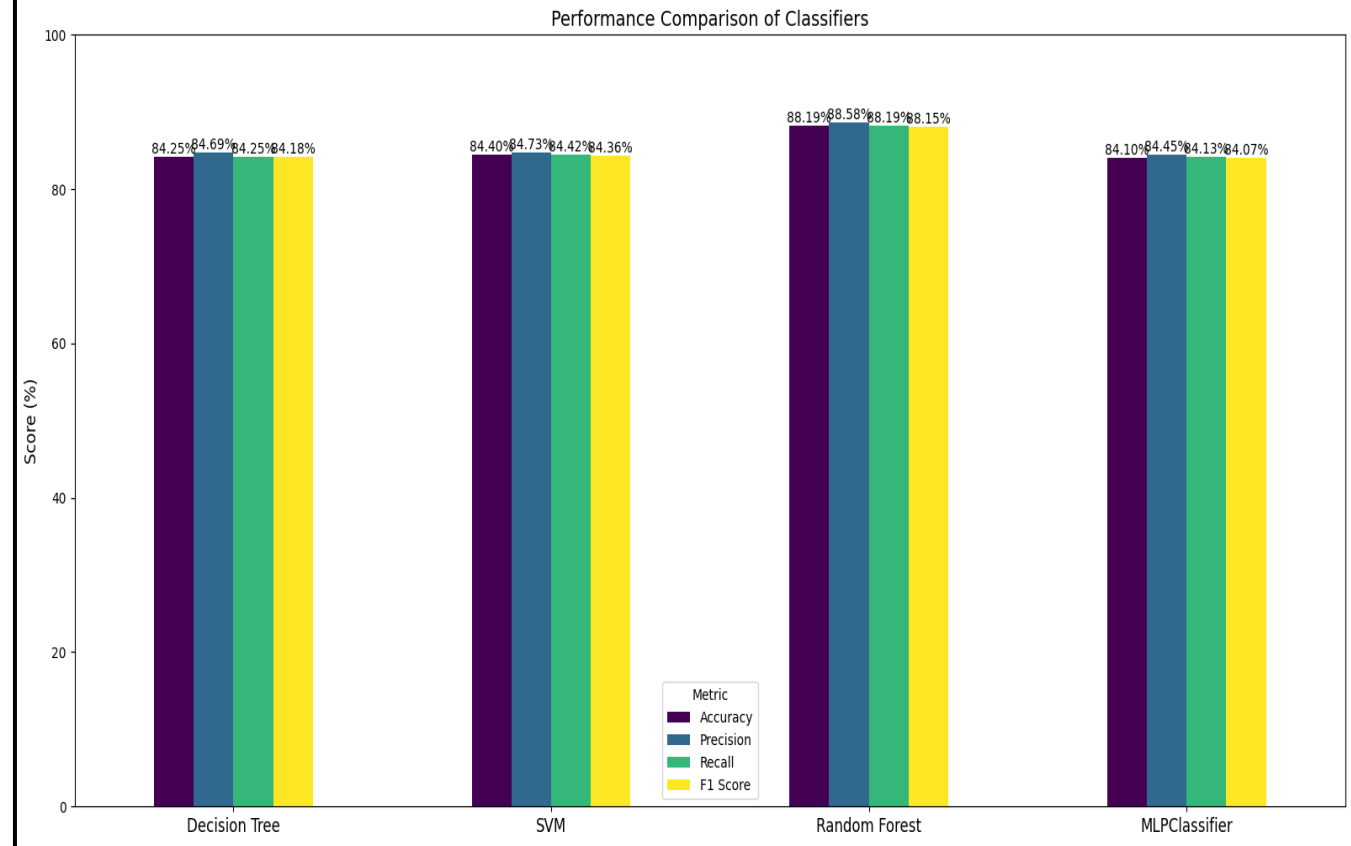


	Accuracy (70/15/15)	Precision (70/15/15)	Recall (70/15/15)	F1 Score (70/15/15)	Accuracy Gap (Train - Test)	F1 Score Gap (Train - Test)
Train	0.919825	0.919884	0.919702	0.919776	0.015063	0.015372
Validation	0.931973	0.932821	0.932500	0.931970	0.015063	0.015372
Test	0.904762	0.903896	0.905249	0.904404	0.015063	0.015372

	Media (70/15/15)	95% CI (70/15/15)
Accuracy	0.841049	(0.82218, 0.859917)
Precision	0.844491	(0.824264, 0.864718)
Recall	0.841331	(0.822262, 0.860399)
F1 Score	0.840689	(0.821831, 0.859546)

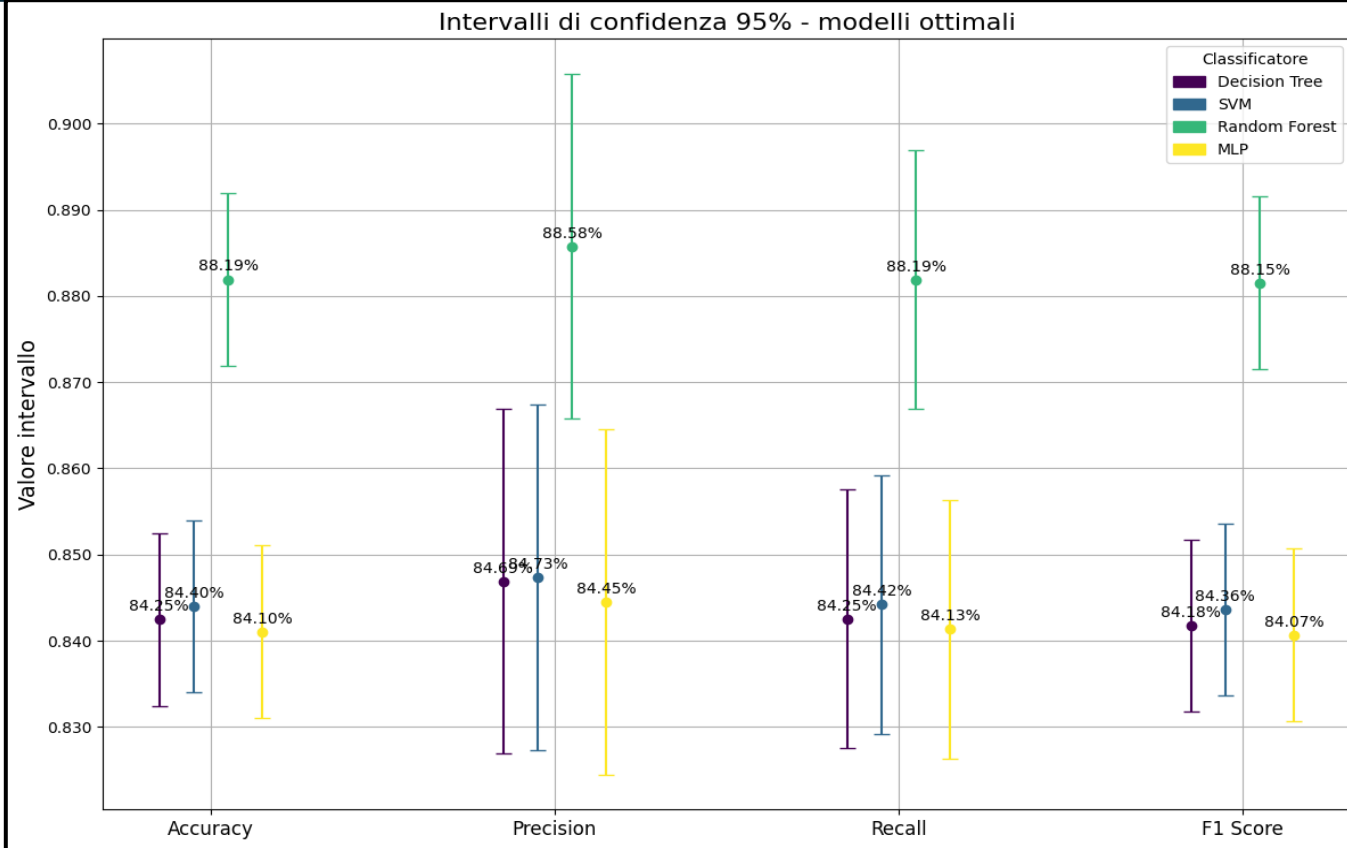
Valutazione: Metriche di Performance

- **Random Forest** ha le migliori prestazioni complessive, con **la più alta accuracy**, precision, recall e F1-score.
- **SVM, Decision Tree e MLP** hanno performance simili, con valori leggermente inferiori, intorno all'84%.



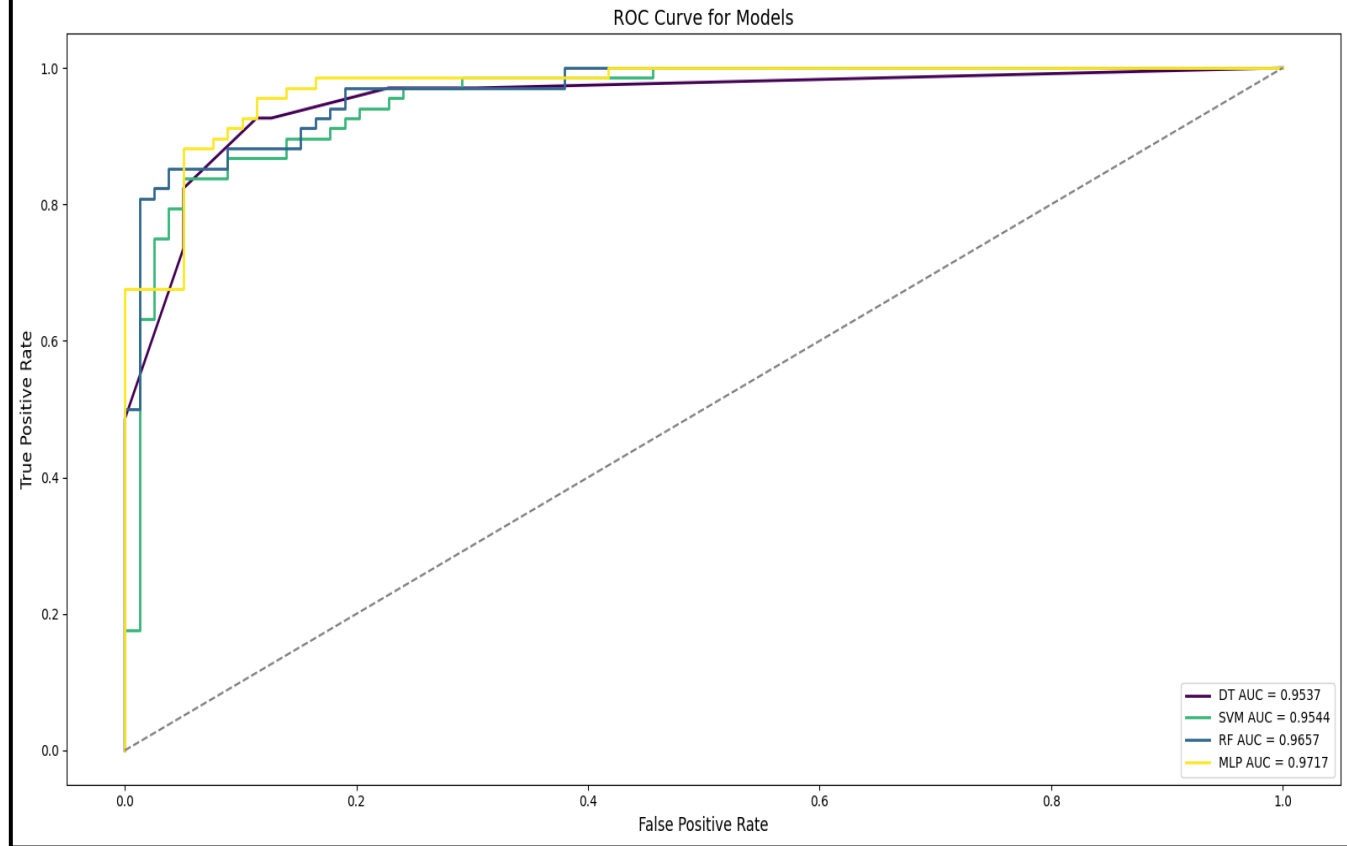
Valutazione: Intervalli di Confidenza

- **Random Forest** ha la **performance più alta**, ma con **maggiore variabilità**.
- **SVM, Decision Tree e MLP** sono **più stabili**, ma con **performance simili e leggermente inferiori**.



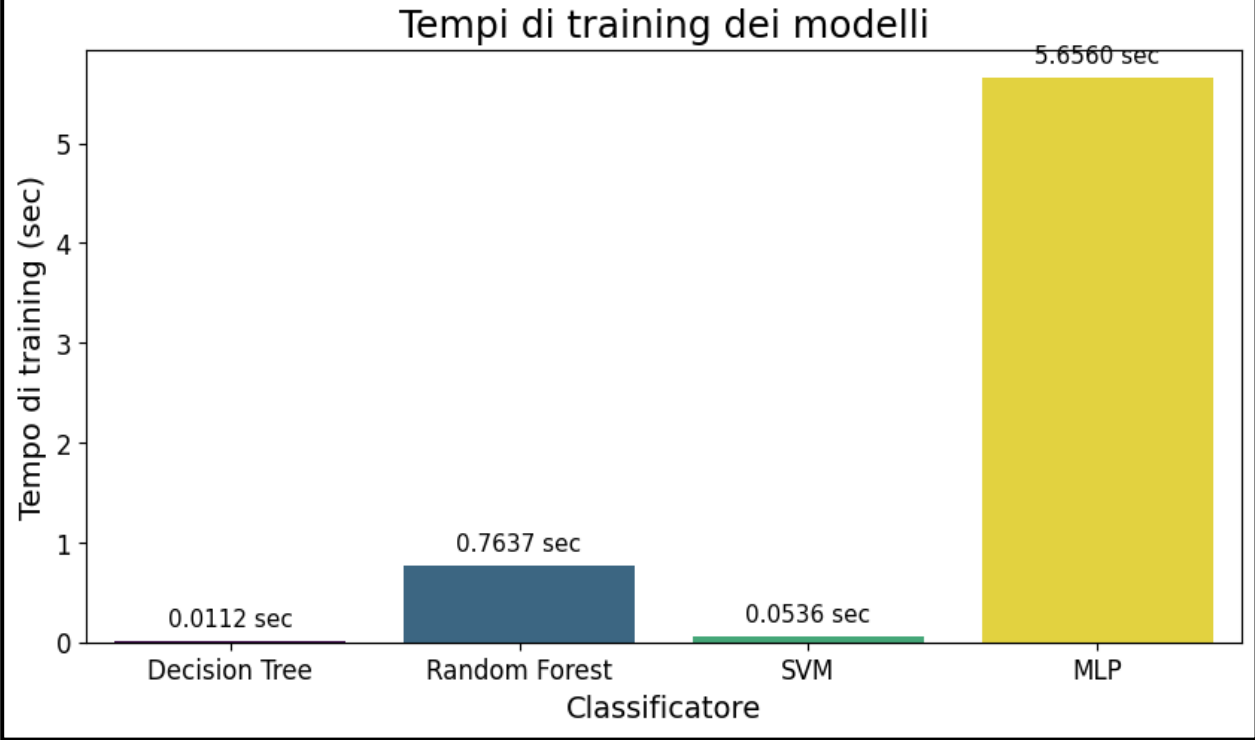
Valutazione: Curve ROC

- **MLP** ha l'**AUC più alto**, il che lo rende il modello **più efficace nella distinzione tra classi**.
- **Random Forest** è quasi altrettanto valido, suggerendo un ottimo **bilanciamento tra accuratezza e robustezza**.
- **SVM** e **Decision Tree** hanno valori **AUC inferiori**, il che indica che potrebbero essere **meno affidabili** per una classificazione precisa.



Valutazione: Tempi di Training

- **Decision Tree** è il modello **più veloce** da addestrare, perfetto per scenari con aggiornamenti rapidi.
- **SVM** offre un **buon equilibrio** tra tempo di training e prestazioni, **leggermente migliori** rispetto a quelle del Decision Tree.
- **Random Forest** è molto più lento, ma ottiene le **migliori performance**, quindi il tempo extra potrebbe essere giustificato.
- **MLP** è il **più** computazionalmente **costoso**, ma se fornisce la migliore **curva ROC**, può essere valutata come la **scelta migliore**.





Grazie per l'attenzione!