

Obesity Insight

Un'analisi approfondita di un dataset sull'obesità per prevedere e comprendere i livelli di sovrappeso, supportando strategie mirate per la salute.

di:

Alessandro Pellegrino, 776460

Kevin Saracino, 776105

Link al progetto:

- [Colab](#)
- [Github](#)

Indice	
Indice	2
Introduzione	3
Analisi del Dataset	3
Pre-Elaborazione dei dati	6
Analisi e selezione delle Feature	6
Apprendimento Supervisionato	14
Alberi di Decisione (Decision Tree)	14
k-Nearest Neighbours	16
Belief Network	17
Sistema di raccomandazione tramite CSP	18
Conclusioni	19

Introduzione

Il progetto “**Obesity Insight**” si propone di costruire un modello predittivo per la classificazione dei livelli di obesità in individui provenienti da Colombia, Perù e Messico. Il modello si basa su dati raccolti tramite questionari sulle abitudini alimentari e le condizioni fisiche di un campione di individui.

Gli obiettivi principali del progetto sono:

- Analizzare il dataset per identificare le feature più rilevanti per la predizione del livello di obesità.
- Addestrare e valutare diversi modelli di Machine Learning per la classificazione del livello di obesità.
- Selezionare il modello migliore in base alle sue prestazioni e alla sua capacità di generalizzare a nuovi dati.
- Costruire una Belief Network per modellare le relazioni tra le feature e il livello di obesità.

La metodologia adottata si può definire tramite le seguenti fasi:

- **Data Collection and Preprocessing:** Raccolta del dataset e pulizia dei dati, gestendo i valori mancanti e convertendo le variabili categoriche in numeriche tramite tecniche di encoding.
- **Feature Engineering:** Creazione di nuove feature derivate da quelle esistenti per migliorare le prestazioni dei modelli.
- **Model Selection and Training:** Selezione di diversi modelli di Machine Learning, training e ottimizzazione degli iperparametri tramite tecniche di cross-validation.
- **Model Evaluation and Comparison:** Valutazione delle performance dei modelli addestrati tramite metriche appropriate e confronto dei risultati per selezionare il modello migliore.
- **Belief Network Construction:** Costruzione di una Belief Network per modellare le relazioni tra le feature e il livello di obesità, utilizzando algoritmi di apprendimento automatico.

Analisi del Dataset

Le informazioni riguardanti il dataset sono reperibili al seguente [link](#) in maniera dettagliata o riassuntiva su [kaggle](#).

Le informazioni principali del dataset sono:

- **Titolo:** Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico
- **Autori:** Fabio Mendoza Palechor e Alexis de la Hoz Manotas
- **Data di Pubblicazione:** 2019
- **Giornale:** *Data in Brief*

Il dataset è stato creato per stimare i livelli di obesità basandosi sugli abitudini alimentari e sulle condizioni fisiche degli individui provenienti dal Messico, Perù e Colombia. Contiene **17 attributi** e **2111 record**, etichettati con una variabile di classe chiamata **NObesity (Livello di Obesità)**.

I record sono classificati in sette categorie:

1. Peso Insufficiente;
2. Peso Normale;
3. Sovrappeso Livello I;
4. Sovrappeso Livello II;
5. Obesità Tipo I;
6. Obesità Tipo II;
7. Obesità Tipo III.

La fonte dei dati è "mista":

- 23% dei dati sono stati raccolti direttamente dagli utenti tramite un sondaggio online;
- 77% dei dati sono stati generati sinteticamente utilizzando lo strumento Weka e il filtro SMOTE (Synthetic Minority Over-sampling Technique).

Le feature del dataset possono essere suddivise in tre categorie principali

- **Abitudini:**
 - Consumo frequente di cibi ad alto contenuto calorico (FAVC);
 - Frequenza di consumo di verdure (FCVC);
 - Numero di pasti principali giornalieri (NCP);
 - Consumo di cibo tra i pasti (CAEC);
 - Quantità d'acqua bevuta giornalmente (CH2O);
 - Consumo di alcol (CALC);
 - Consumo di tabacco (SMOKE).
- **Condizioni Fisiche:**
 - Monitoraggio del consumo di calorie (SCC);
 - Frequenza di attività fisica (FAF);
 - Tempo trascorso nell'uso di dispositivi tecnologici (TUE);
 - Mezzo di trasporto utilizzato (MTRANS).
- **Generalità:**
 - Storico familiare riguardante casi di obesità in famiglia (Family_History);
 - Sesso;
 - Età;
 - Altezza;
 - Peso.

In totale, il dataset contiene 17 feature, ognuna delle quali rappresenta una caratteristica rilevante per la predizione del livello di obesità. Secondo le informazioni disponibili, il valore delle feature rappresentano le seguenti informazioni:

Feature	Valore
Gender	Female / Male
Age	Numeric Value (in years)
Height	Numeric Value (in metres)
Weight	Numeric Value (in kgs)
Family_History	Yes / No

"Has a family member suffered or suffers from overweight?"	
FAVC "Do you eat high caloric food frequently?"	Yes / No
FCVC "Do you usually eat vegetables in your meals?"	Never: 1 / Sometimes: 2 / Always: 3
NCP "How many main meals do you have daily?"	One / Two / Three / More than three
CAEC "Do you eat any food between meals?"	No / Sometimes / Frequently / Always
SMOKE "Do you smoke?"	Yes / No
CH2O "How much water do you drink daily?"	Less than a liter: 1 / Between 1 and 2 L: 2 / More than 2 L: 3
SCC "Do you monitor the calories you eat daily?"	Yes / No
FAF "How often do you have physical activity?"	I do not have: 0 / 1 or 2 days: 1 / 2 or 4 days: 2, 4 / 5 days: 3
TUE "How much time do you use technological devices such as cell phone, videogames, television, computer and others?"	Between 0 and 2 hours: 0 / Between 3 and 5 hours: 1 / More than 5 hours: 2
CALC "How often do you drink alcohol?"	No / Sometimes / Frequently / Always
MTRANS "Which transportation do you usually use?"	Automobile / Motorbike / Bike / Public_Transportation / Walking
Obesity_Level (Feature Target)	Insufficient_Weight / Normal_Weight, Overweight_Level_I / Overweight_Level_II, Obesity_Type_I / Obesity_Type_II / Obesity_Type_III

Osservazioni: dal dataset che si trova su Kaggle, rispetto alle informazioni del paper, possiamo osservare che:

- le feature "Age", "FCVC", "NCP", "CH2O", "FAF" e "TUE" risultano essere float invece che interi, dato che il 77% dei dati è stato generato sinteticamente utilizzando lo strumento Weka e il filtro SMOTE, quindi è necessario una operazione di arrotondamento per difetto dei valori di queste colonne;

- le feature "FCVC", "NCP", "CH2O", "FAF" e "TUE" secondo il paper non dovrebbero essere valori interi ma categorici, quindi è necessario trattarli come tali;
- le restanti colonne sono categoriche, per cui sono necessarie operazioni di Encoding per mapparle a valori interi.

Pre-Elaborazione dei dati

Analisi e selezione delle Feature

La selezione delle feature è un passaggio fondamentale per migliorare la qualità del dataset e garantire che i modelli siano efficienti e precisi. In questa fase:

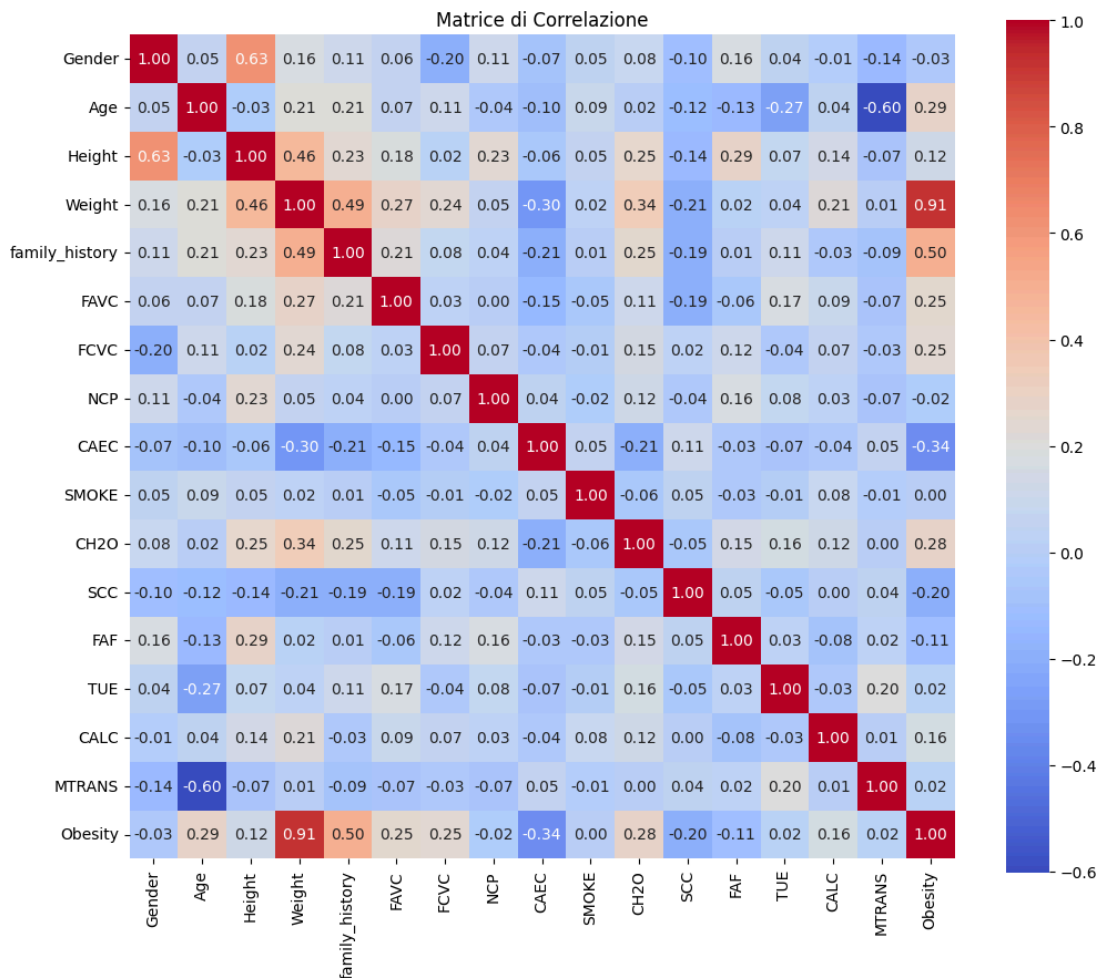
1. Elimineremo le feature che risultano poco informative o ridondanti;
2. Aggiungeremo nuove feature derivate dalla conoscenza del dominio per arricchire il dataset.

L'obiettivo è creare un **dataset ottimizzato** per la fase di apprendimento supervisionato.

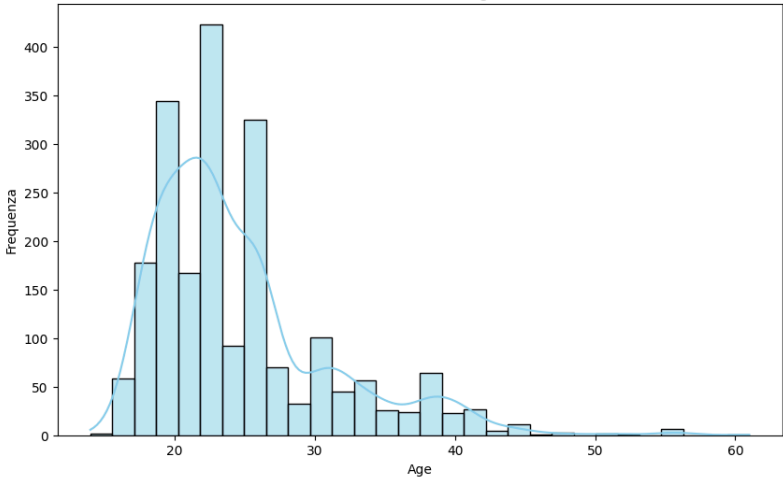
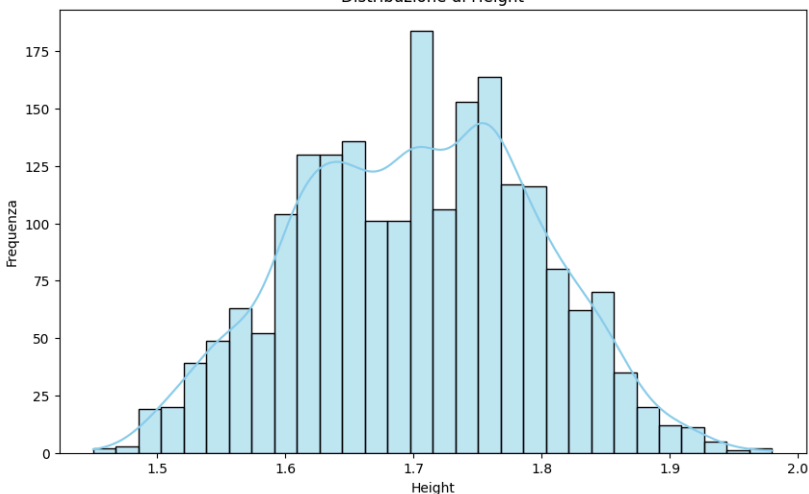
Prima di procedere con la modellazione, è fondamentale comprendere le relazioni tra le feature numeriche del dataset, tramite questa analisi riusciremo ad identificare:

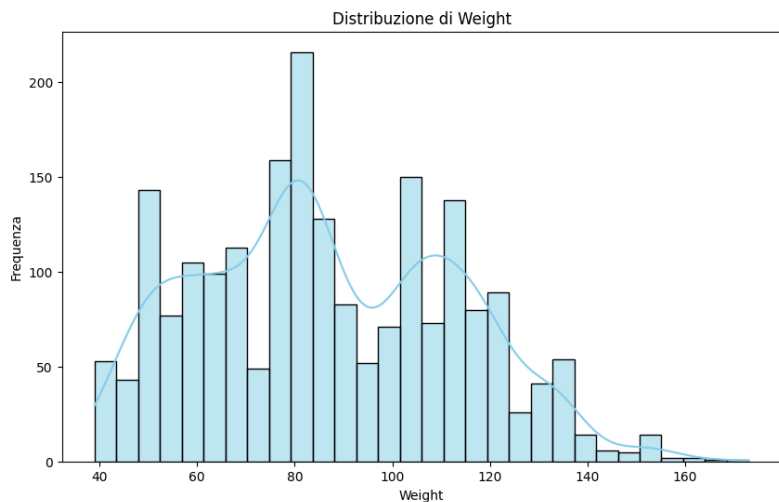
- Le feature più correlate con il target (**Obesity_level**), che saranno probabilmente le più rilevanti per la predizione;
- Le feature altamente correlate tra loro (**multicollinearità**), che potrebbero essere ridondanti e influenzare negativamente i modelli di apprendimento automatico.

Per questo scopo, calcoliamo la matrice di correlazione e visualizziamo i risultati tramite una heatmap:



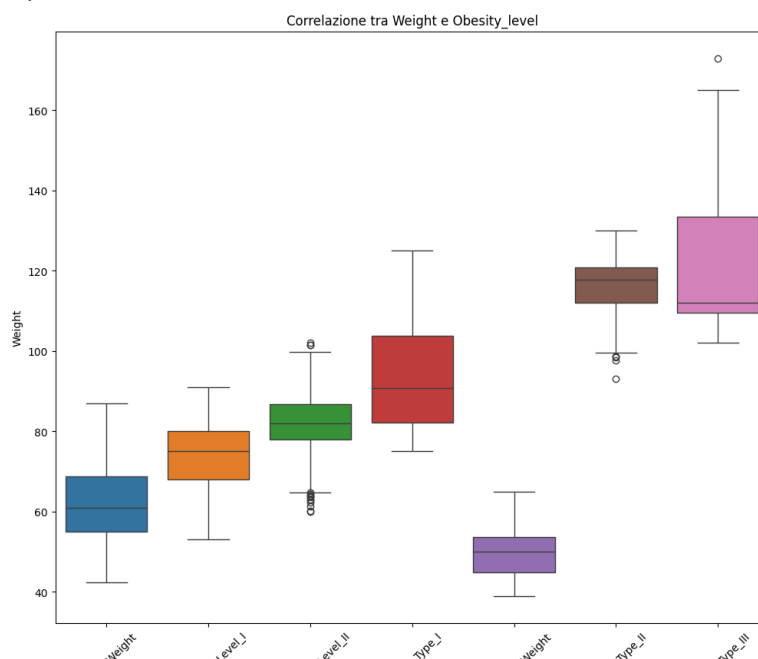
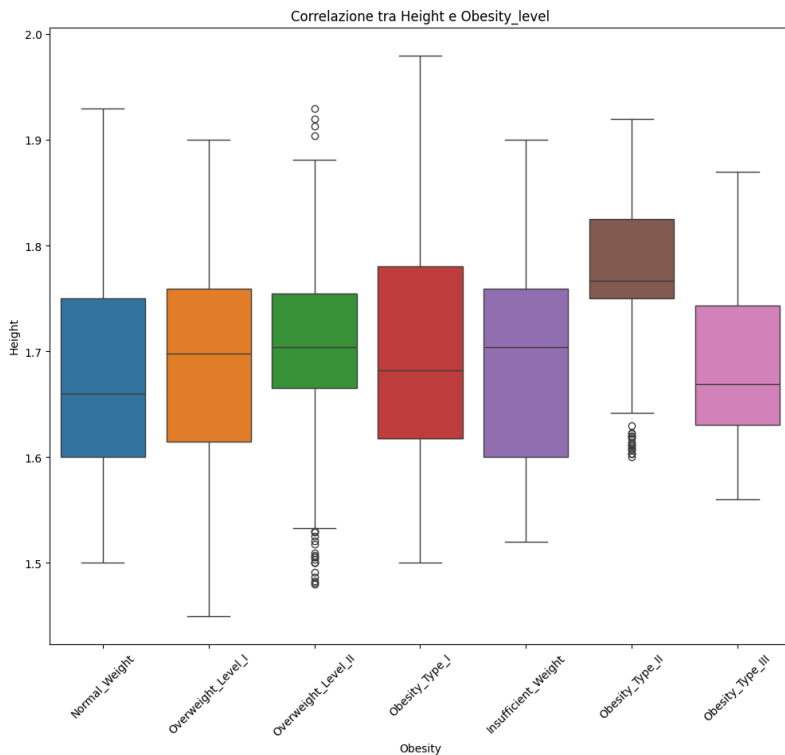
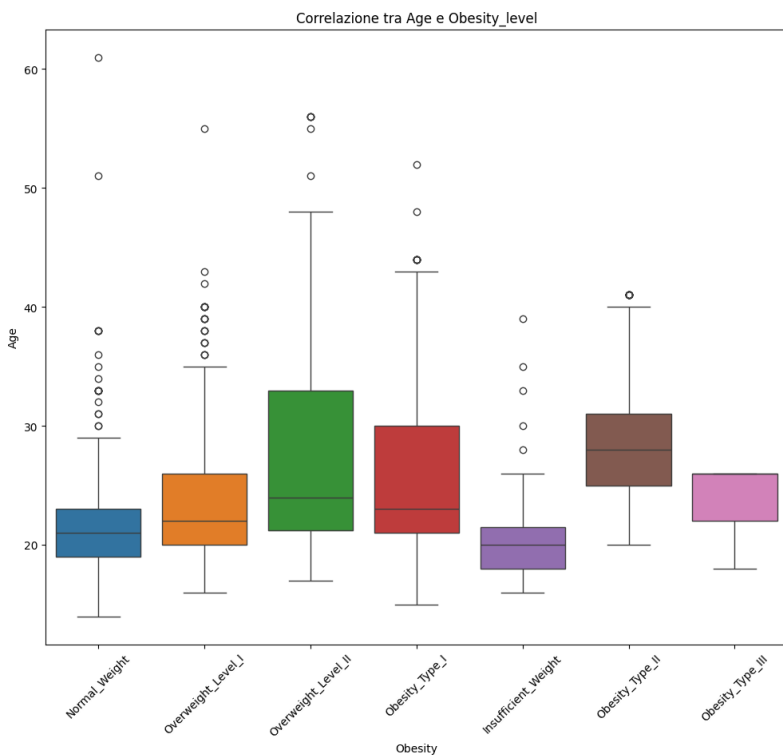
Dalla matrice di correlazione si può notare che la feature “**Weight**” è **altamente correlata** alla feature target “**Obesity**”. Adesso analizziamo le **distribuzioni delle feature numeriche**, ci permettono di capire come i dati sono strutturati e se esistono **anomalie** o **skewness**. Utilizziamo istogrammi e KDE plots per visualizzare queste distribuzioni:

Istogramma	Risultato
<p data-bbox="453 443 612 465">Distribuzione di Age</p> 	<p>La distribuzione è asimmetrica positivamente (right-skewed), con una coda lunga a destra. Ciò significa che la maggior parte dei dati è concentrata su valori più bassi di età, mentre alcuni individui hanno età significativamente più elevate.</p> <p>La maggior parte delle osservazioni si trova tra 15 e 30 anni, mostrando una prevalenza giovanile.</p> <p>L'intervallo della distribuzione va da circa 10 a 60 anni, ma la densità si riduce drasticamente dopo i 40 anni, con pochi individui in quella fascia.</p> <p>Non ci sono outlier evidenti.</p>
<p data-bbox="437 994 628 1016">Distribuzione di Height</p> 	<p>La distribuzione è asimmetrica positivamente (right-skewed), presenta una coda più lunga a destra. Suggerisce che la maggior parte delle osservazioni si trova sulla sinistra, mentre alcuni valori più elevati sono meno frequenti.</p> <p>La maggior parte dei dati sembra concentrarsi intorno a 1.7 metri, il che potrebbe rappresentare un valore medio o mediano.</p> <p>L'intervallo dei dati si estende da circa 1.4 a 2.0 metri, suggerendo una discreta variabilità nelle altezze.</p> <p>Non ci sono evidenti outlier.</p>



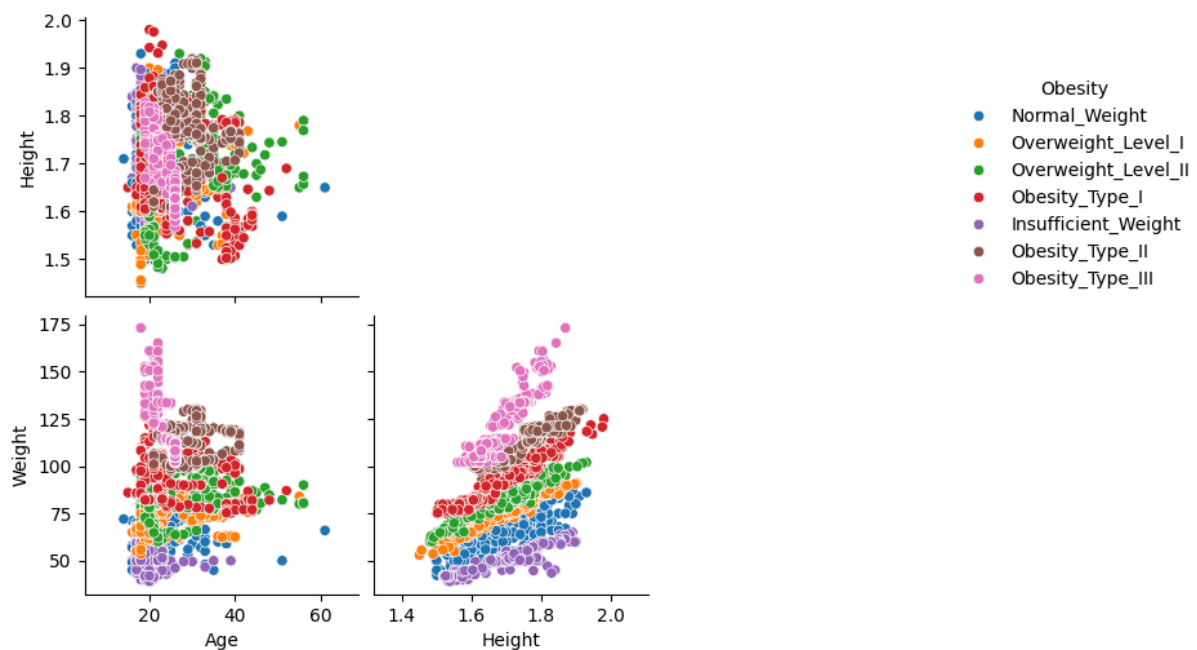
La distribuzione è leggermente **asimmetrica positivamente (right-skewed)**, presenta una coda più lunga a destra. Il **centro della distribuzione** può essere stimato intorno ai 80 kg, dove si trova il picco principale della frequenza. L'intervallo dei dati si estende **da 40 a circa 180 kg**, suggerendo una variabilità tra i pesi. Non ci sono evidenti outlier.

Analizziamo ora la correlazione tra le feature precedenti ("Age", "Weight" e "Height") e la feature target ("Obesity_Level"):



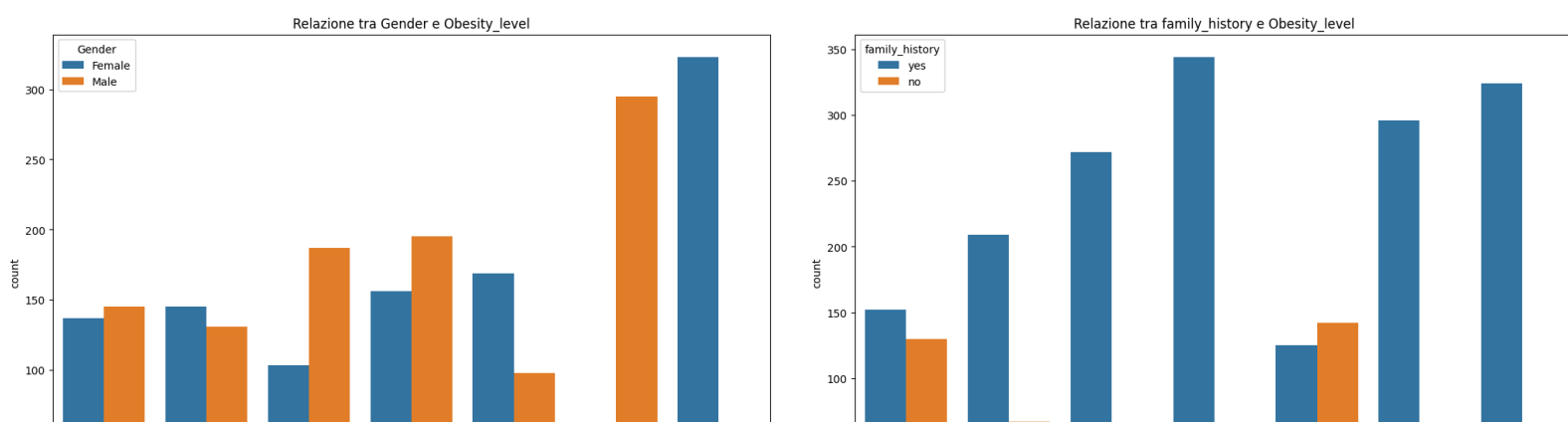
Possiamo intuire, quindi, che con le feature “**Age**” e “**Weight**” si riesce a **classificare in maniera** relativamente **semplice** i livelli di obesità, mentre attraverso la feature “**Height**” **non si riescono a classificare** le classi di obesità, perché i boxplot tra i vari livelli di obesità si sovrappongono.

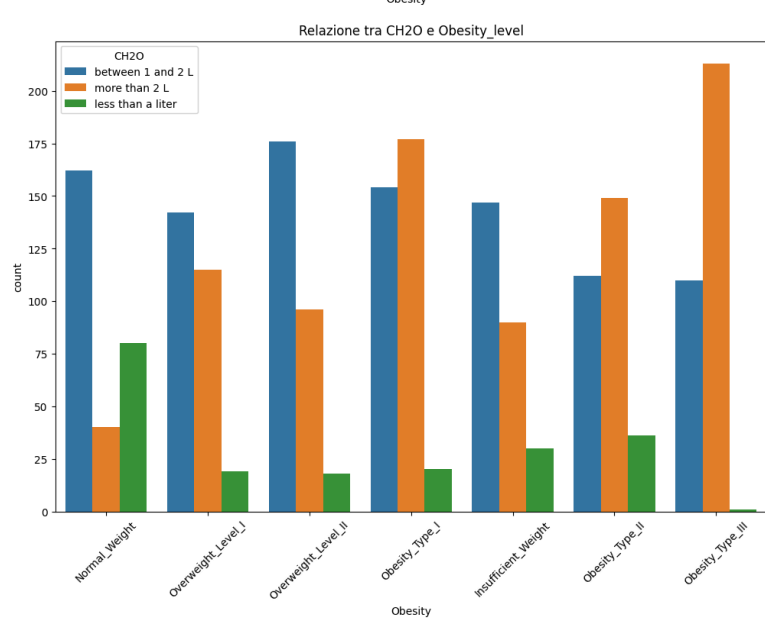
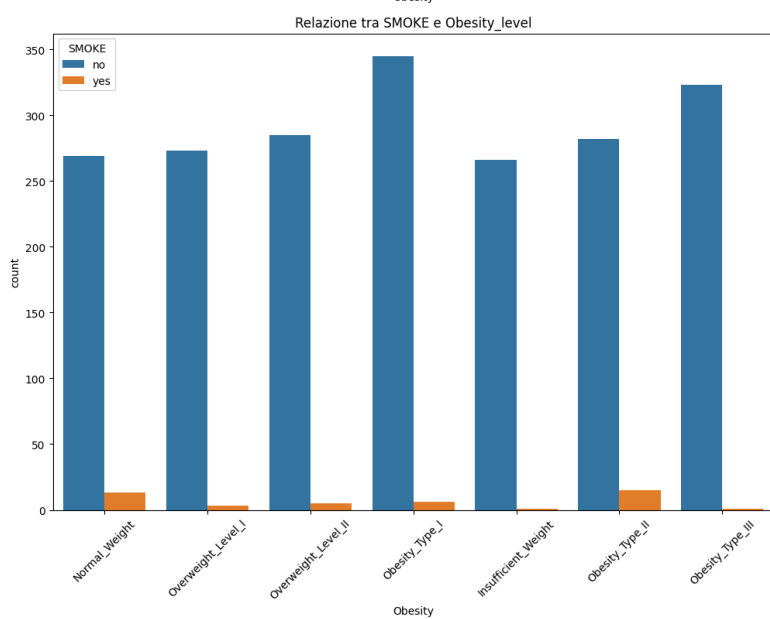
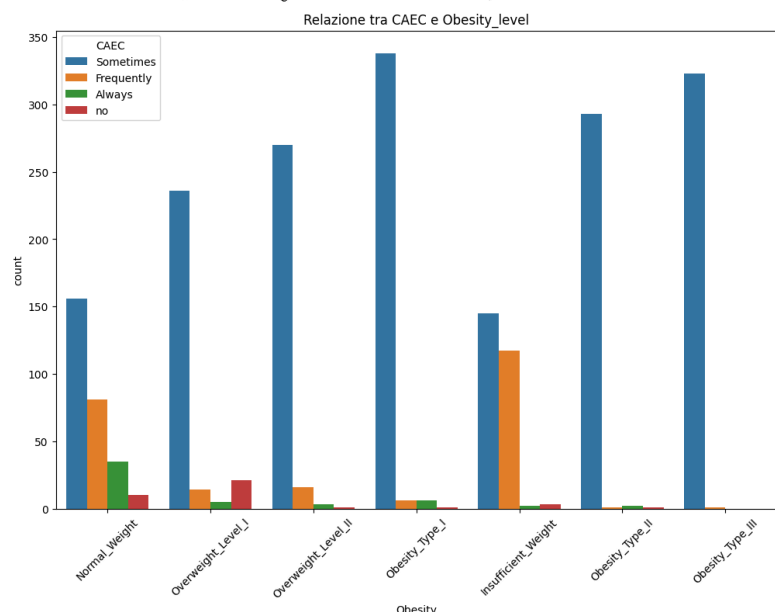
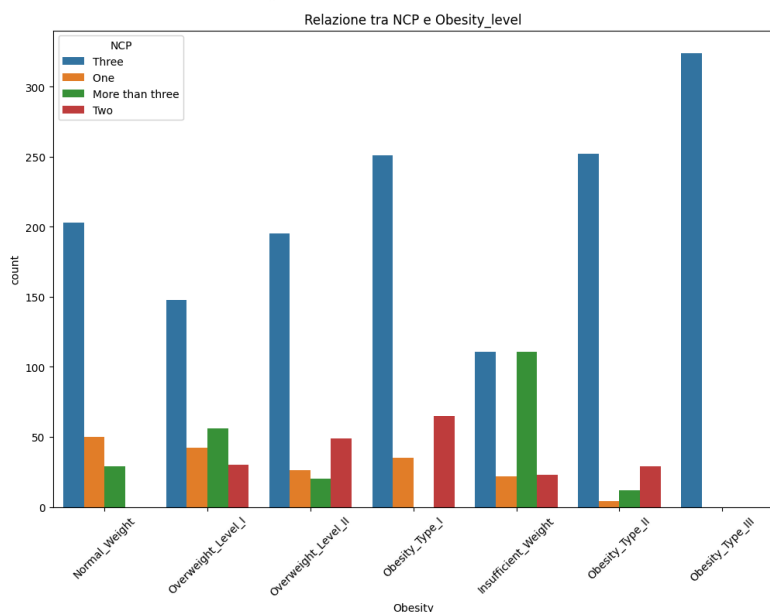
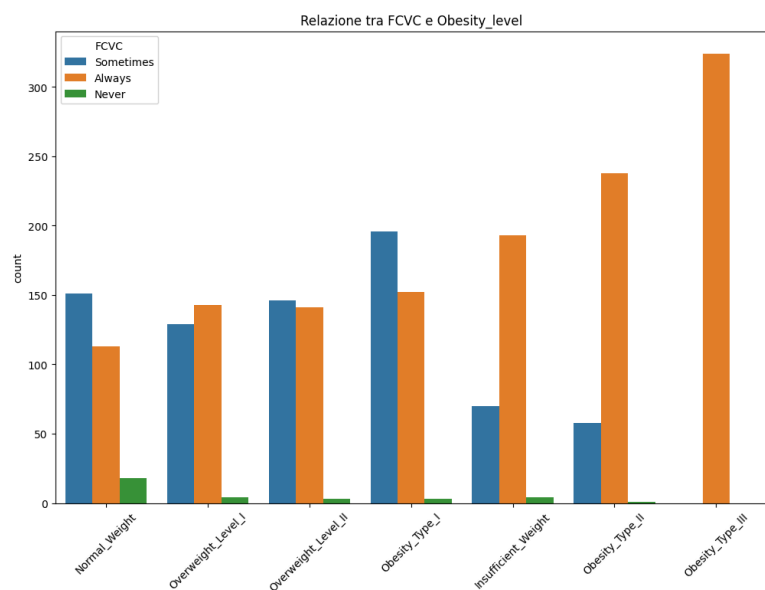
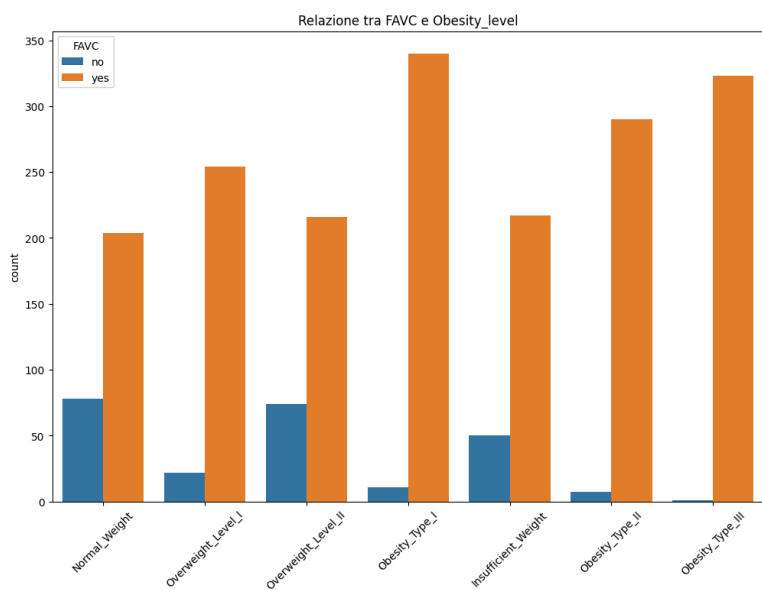
Le **relazioni tra feature numeriche** possono essere analizzate tramite scatter plots o pair plots, questi grafici ci permettono di vedere se esistono pattern lineari o non lineari tra le variabili:

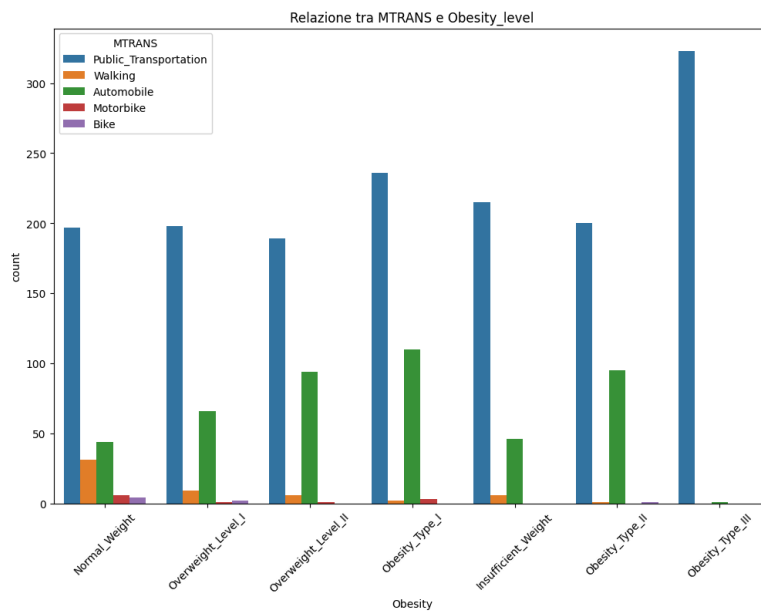
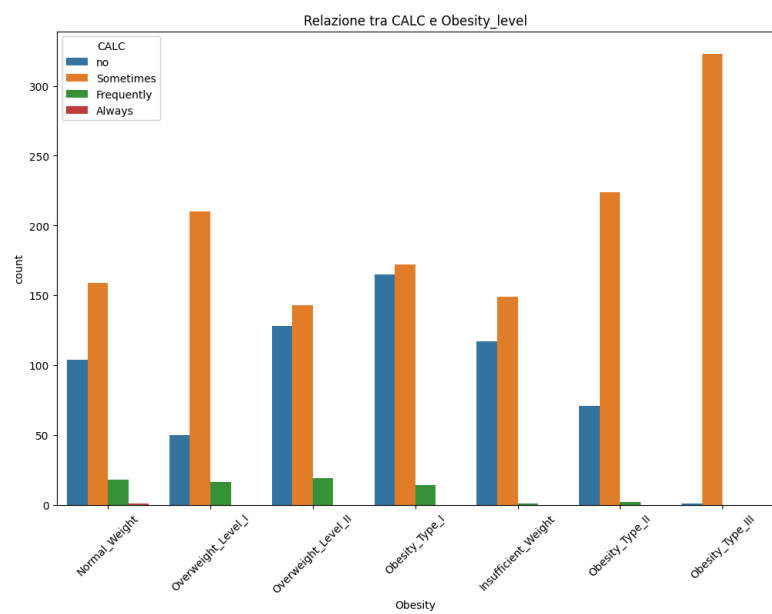
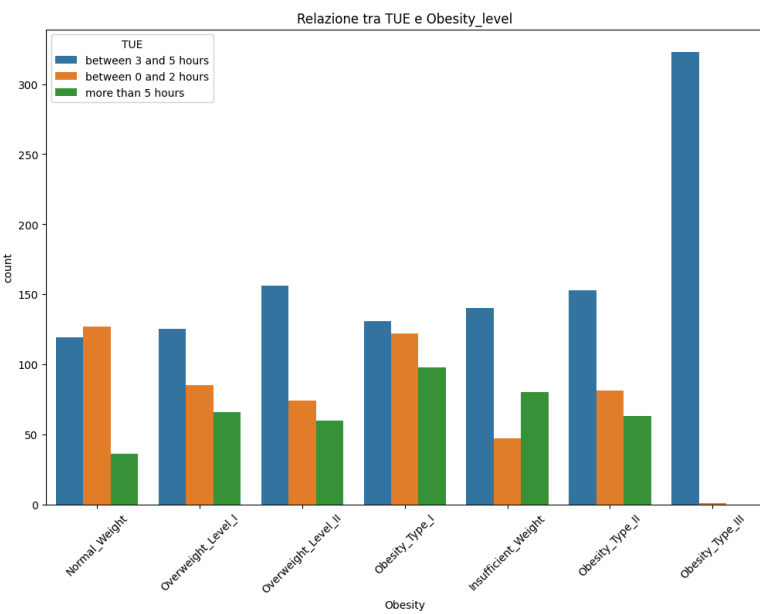
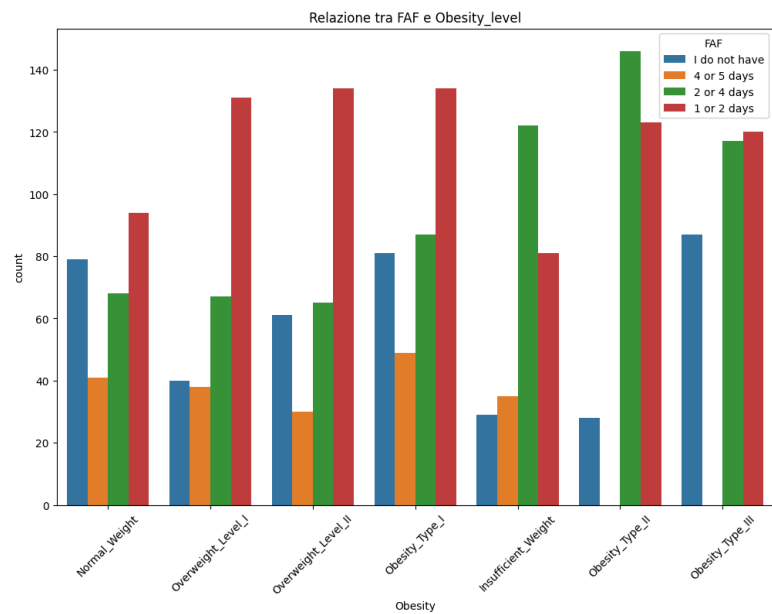
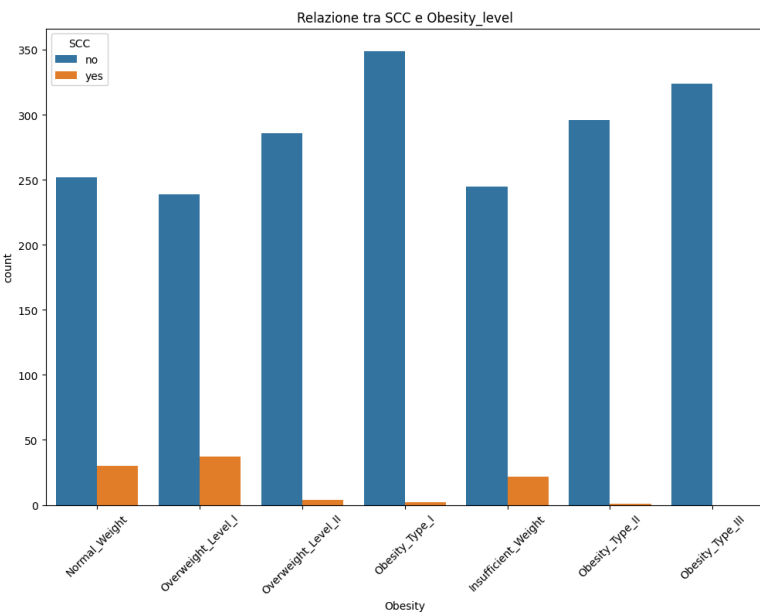


Dallo scatter plot tra “Height” e “Weight”, possiamo osservare una chiara relazione positiva tra altezza e peso; mentre il pair plot, che offre una visione completa delle relazioni tra tutte le feature numeriche mostra che non sembrano esserci relazioni non lineari particolarmente complesse tra le feature numeriche.

Infine, analizziamo tutte le feature in correlazione con la feature target:







Da una attenta analisi degli istogrammi precedenti si possono notare le seguenti osservazioni:

- **Feature** che sembrano avere una **correlazione bassa/nulla** con il livello di obesità:
 - **Gender**: il livello di obesità è quasi bilanciato tranne per i casi di **Obesity_Type_II** e **Obesity_Type_III** i cui casi si trovano principalmente in quello maschile e femminile rispettivamente;
 - **SMOKE**: non sembra essere direttamente correlato ai livelli di obesità in questo dataset;
 - **SCC**: il monitoraggio delle calorie mangiate dall'istogramma sembra che non influisce molto sull'obesità;
 - **CALC**: il consumo di bevande alcoliche sembra non avere nessuna relazione con l'obesità;
 - **MTRANS**: usare un mezzo di trasporto rispetto ad un altro non risulta essere importante per il livello di obesità.

Quindi queste feature verranno eliminate, perché non hanno una correlazione significativa né un impatto diretto con il target, eliminandole il dataset sarà più semplice e migliorerà le prestazioni dei modelli.

- **Feature** che sembrano avere una **correlazione alta** con il livello di obesità:
 - **FAVC**: individui che consumano regolarmente cibi ad alto contenuto calorico sono molto più numerosi nelle categorie di sovrappeso e obesità;
 - **family_history**: se l'obesità è presente nella storia familiare, è più probabile essere in sovrappeso;
 - **CAEC**: mangiare tra i pasti frequentemente è correlato a un aumento del numero di individui nelle categorie di obesità.
- Le **feature rimanenti** ("FCVC", "NCP", "CH2O", "FAF", "TUE") **non sembrano avere correlazioni** dal punto di vista grafico (i grafici si dimostrano per lo più bilanciati in tutti i casi), ma dal punto di vista logico (in un'analisi realistica) sono fattori estremamente correlati al nostro caso di studio. Come dimostrano i seguenti studi:
 - **FCVC**: La consumazione frequente di verdure comporta una perdita di peso e un rischio minore di obesità ([paper correlato](#));
 - **NCP**: Il numero di pasti principali giornalieri incide sul rischio di obesità, chi consuma più di 3 pasti al giorno ha il 32% di probabilità di avere livelli alti di obesità, rispetto a chi ne consuma meno di 3 ([paper correlato](#));
 - **CH2O**: La quantità d'acqua bevuta giornalmente è un fattore correlato al controllo del proprio peso, quindi rilevante al rischio di obesità ([paper correlato](#));
 - **FAF**: La frequenza di attività fisica è un fattore importante per il rischio di obesità, si dimostra che gli adulti dovrebbero accumulare circa 60 minuti di attività fisica di intensità moderata al giorno per prevenire l'aumento di peso non salutare ([paper correlato](#));
 - **TUE**: Il tempo trascorso nell'uso di dispositivi tecnologici sembra influire sul rischio di obesità, usare dispositivi tra 1 e 3 ore al giorno, aumenta del 40% il rischio di obesità rispetto ad usarli per meno di un'ora ([paper correlato](#));

Dopo l'eliminazione delle feature non correlate e basandosi sulla conoscenza del dominio, possiamo **creare nuove feature** che catturano relazioni nascoste nei dati.

Ad esempio:

- **BMI (Body Mass Index, o IMC, Indice di Massa Corporea)**: l'indicatore di riferimento per studi epidemiologici e di screening di obesità, tuttavia è utile sottolineare che il BMI in quanto indicatore di studi di popolazione, non è in grado di valutare la reale composizione corporea, così come non permette di conoscere la distribuzione del grasso corporeo nell'individuo. Esso viene calcolato come peso diviso l'altezza al quadrato:

$$BMI = \frac{weight}{height^2}$$

- **Indice di Famiglia e Storia Personale**: la presenza di obesità nella storia familiare (**family_history**) e il consumo di cibi ad alto contenuto calorico (**FAVC**) possono essere combinati per creare un indice che misuri il **rischio genetico e comportamentale** (def. **Genetic and Behavioral Risk, o GBR**), quindi la presenza di entrambi i fattori fa aumentare significativamente il rischio di obesità. Verrà quindi calcolato come una media tra family_history e FAVC:

$$GBR = \frac{family_history + FAVC}{2}$$

Apprendimento Supervisionato

L'**apprendimento supervisionato** è una **branca del machine learning** in cui un modello viene addestrato su un set di **dati etichettati**, ovvero dati in cui ogni esempio ha un'etichetta o un valore di output corrispondente. L'obiettivo è quello di imparare una funzione che mappi gli input agli output desiderati, in questo modo, il modello può prevedere l'output per nuovi esempi che non ha mai visto prima.

Nel nostro caso, può essere utilizzato per prevedere il livello di obesità di una persona in base alle sue caratteristiche (età, peso, altezza, abitudini alimentari, ecc.), il dataset contiene esempi etichettati, in cui ogni esempio rappresenta una persona con le sue caratteristiche e il livello di obesità corrispondente. Il modello verrà addestrato su questo dataset per imparare a prevedere il livello di obesità per nuove persone.

Inizialmente si suddivide il dataset iniziale in due parti una per l'allenamento dei modelli (**training set**) e l'altra per il testing dei modelli allenati (**test set**); sono state create le funzioni che permettono di allenare e valutare un modello e successivamente visualizzare le sue informazioni (tramite **tuning** degli **iperparametri**) e prestazioni (attraverso **F1** e ed **Accuratezza**).

Per trovare la migliore combinazione di questi parametri, è stata utilizzata la tecnica di Grid Search con Cross-Validation a 5 fold. La **Grid Search** è una tecnica di ottimizzazione degli iperparametri, funziona valutando un modello su tutte le possibili combinazioni di iperparametri specificate in una griglia, per ogni combinazione, il modello viene addestrato e valutato utilizzando la **cross-validation**. La combinazione di iperparametri che produce il punteggio migliore (ad esempio attraverso accuratezza, F1-score) viene quindi selezionata come la migliore. Questo processo prevede la valutazione di tutte le possibili combinazioni di valori specificate per i parametri, misurando le prestazioni del modello su diversi sottoinsiemi dei dati di addestramento. Infine, vengono scelti i valori che massimizzano le prestazioni del modello sulla base di una metrica di valutazione (ad esempio, l'accuratezza). In seguito sono stati riportati i modelli di apprendimento supervisionato utilizzati.

Alberi di Decisione (Decision Tree)

Gli **alberi di decisione** sono un tipo di algoritmo di apprendimento supervisionato utilizzato sia per la classificazione (come nel nostro caso) che per la regressione. Sono modelli predittivi che utilizzano una struttura ad albero per rappresentare una serie di decisioni basate su feature (caratteristiche) dei dati, che portano a una previsione o a una classificazione finale. La loro struttura prevede:

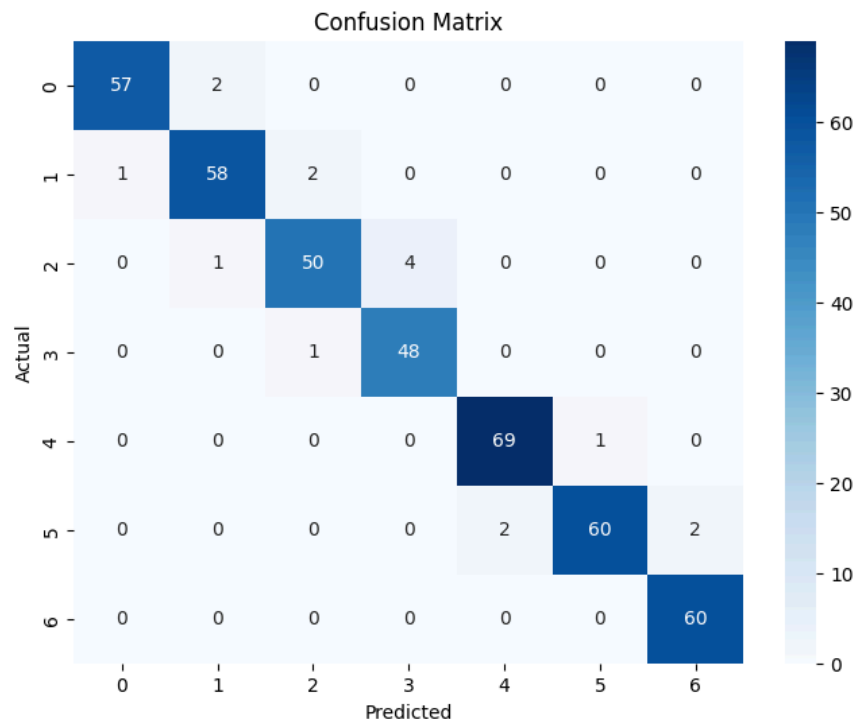
- **Radice:** L'albero inizia con un nodo radice che rappresenta l'intero dataset;
- **Nodi interni:** Ogni nodo interno rappresenta una feature dei dati e si divide in rami in base ai possibili valori di quella feature;
- **Rami:** I rami rappresentano le possibili decisioni prese in base al valore della feature nel nodo interno;
- **Nodi foglia:** I nodi foglia rappresentano il risultato finale, ovvero la previsione o la classificazione.

Durante la fase di addestramento e ottimizzazione del modello sono stati presi in considerazione:

1. **criterion**: parametro che definisce la funzione di impurità utilizzata per valutare la qualità di una suddivisione (**split**) nell'albero, sono state scelte **log_loss** e **entropy**, non scegliendo, invece, il criterio **gini**:
 - **log_loss**: noto come **Logarithmic Loss** o **Cross-Entropy Loss**, quantifica la differenza tra le probabilità predette dal modello e le etichette vere;
 - **entropy**: misura l'impurità di un nodo in base all'incertezza associata alla classificazione delle istanze nel nodo;
 - **gini**: misura l'impurità di un nodo in base alla probabilità di classificare erroneamente un'istanza se la sua classe venisse assegnata casualmente.
2. **max_depth**: parametro che controlla la profondità massima dell'albero decisionale. Un valore maggiore consente di creare alberi più complessi e potenzialmente più accurati, ma aumenta anche il rischio di **overfitting** (adattamento eccessivo ai dati di addestramento). Sono stati considerati i valori 5, 10 e 15 per esplorare diverse complessità del modello.
3. **random_state**: parametro che controlla il generatore di numeri casuali utilizzato durante la costruzione dell'albero. Impostando un valore specifico, come [42](#) in questo caso, si assicura che i risultati siano riproducibili.

Infine, il modello è stato valutato tramite le metriche di: Precision, Recall ed F1.

Si riporta la matrice di confusione generata dalla valutazione dell'Albero di Decisione:



k-Nearest Neighbours

k-NN è un algoritmo di apprendimento automatico supervisionato utilizzato per la **classificazione** (come nel nostro caso) e la **regressione**. È un algoritmo non parametrico basato su istanze, esso non fa ipotesi sulla distribuzione sottostante dei dati e memorizza tutti gli esempi di addestramento per effettuare previsioni.

Il suo funzionamento si può dividere in:

1. **Fase di addestramento:** memorizza semplicemente tutti gli esempi di addestramento con le relative etichette;
2. **Fase di previsione:** quando viene presentato un nuovo esempio, il modello k-NN calcola la distanza tra il nuovo esempio e tutti gli esempi di addestramento;
3. **Selezione dei vicini:** Il modello seleziona i *k* esempi di addestramento più vicini al nuovo esempio;
4. **Previsione:** Per la classificazione, il modello assegna al nuovo esempio l'etichetta più frequente tra i k vicini. Per la regressione, il modello prevede il valore medio dell'output tra i k vicini.

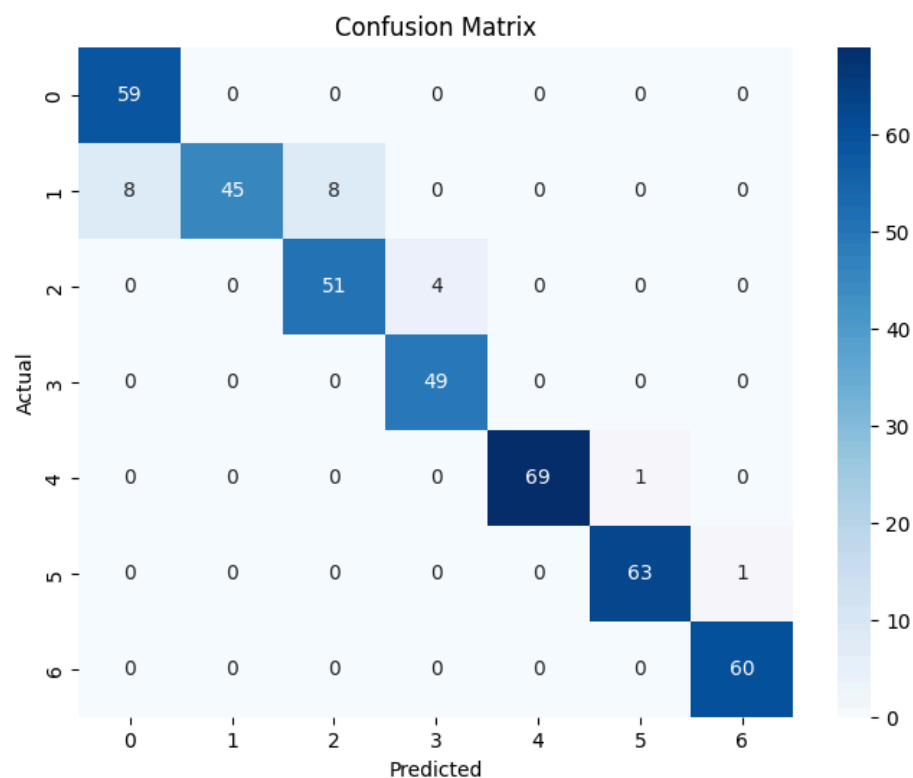
Nel nostro caso, il k-NN è stato utilizzato per prevedere il livello di obesità di una persona in base alle sue caratteristiche. Il dataset è stato diviso in un set di addestramento e un set di test. Il modello k-NN è stato addestrato sul set di addestramento e valutato sul set di test.

I principali **iperparametri** del k-NN sono:

- **k:** il numero di vicini da considerare.
- **Metrica di distanza:** la funzione utilizzata per calcolare la distanza tra gli esempi.
- **Pesi:** se dare a tutti i vicini lo stesso peso o dare più peso ai vicini più vicini.

Infine, il modello è stato valutato tramite le metriche di: **Precision**, **Recall** ed **F1**.

Si riporta la matrice di confusione generata dalla valutazione del k-NN:



Regressione Logistica

La **Regressione Logistica** è un algoritmo di apprendimento supervisionato utilizzato per la classificazione, è un modello lineare che prevede la probabilità che un esempio appartenga a una determinata classe.

Utilizza una **funzione logistica** per mappare l'input a un valore compreso tra 0 e 1, che rappresenta la probabilità che l'esempio appartenga a una determinata classe. La funzione logistica è definita come segue:

$$\hat{p}(X_i) = \text{expit}(X_i w + w_0) = \frac{1}{1 + \exp(-X_i w - w_0)}$$

Nel nostro caso, la regressione logistica è stata utilizzata per prevedere il livello di obesità di una persona in base alle sue caratteristiche, il dataset è stato diviso in un set di addestramento e un set di test. Il modello di regressione logistica è stato addestrato sul set di addestramento e valutato sul set di test.

I **principali iperparametri** della regressione logistica sono:

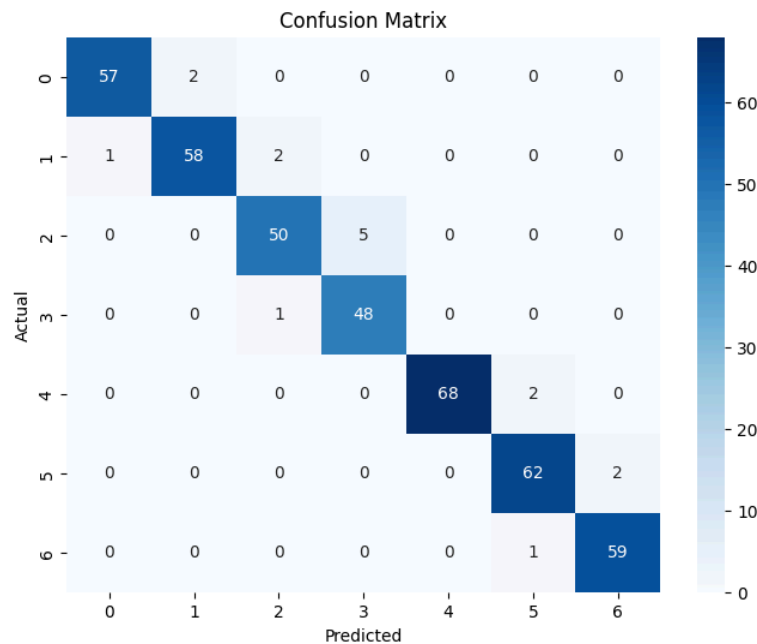
1. **C**: il parametro di regolarizzazione inversa, impostato a 0.1, 1 e 10.
2. **random_state**: parametro che controlla il generatore di numeri casuali utilizzato durante la costruzione dell'albero. Impostando un valore specifico, come [42](#) in questo caso, si assicura che i risultati siano riproducibili.
3. **solver**: l'algoritmo utilizzato per ottimizzare i pesi del modello, sono stati utilizzati "lbfgs", "newton-cg" e "sag". Scelti seguendo la tabella:

solver	penalty	multinomial multiclass
'lbfgs'	'l2', None	yes
'liblinear'	'l1', 'l2'	no
'newton-cg'	'l2', None	yes
'newton-cholesky'	'l2', None	no
'sag'	'l2', None	yes
'saga'	'elasticnet', 'l1', 'l2', None	yes

Gli **iperparametri** della regressione logistica sono stati ottimizzati utilizzando la ricerca a griglia con cross-validation a 5 fold, essa ha valutato diverse combinazioni di valori per gli iperparametri e ha selezionato la combinazione che ha prodotto le migliori prestazioni sul set di convalida.

Infine, il modello è stato valutato tramite le metriche di: **Precision**, **Recall** ed **F1**

Si riporta la matrice di confusione generata dalla valutazione della Regressione Logistica:



Random Forest

Random Forest è un algoritmo di apprendimento automatico supervisionato utilizzato per la classificazione e la regressione, è un **metodo di ensemble** che combina più alberi decisionali per creare un modello più robusto e preciso. Il suo funziona si basa sulla costruzione di un insieme di alberi decisionali durante l'addestramento, producendo l'output in base alla classe (classificazione) o alla previsione media (regressione) dei singoli alberi. Segue le seguenti fasi:

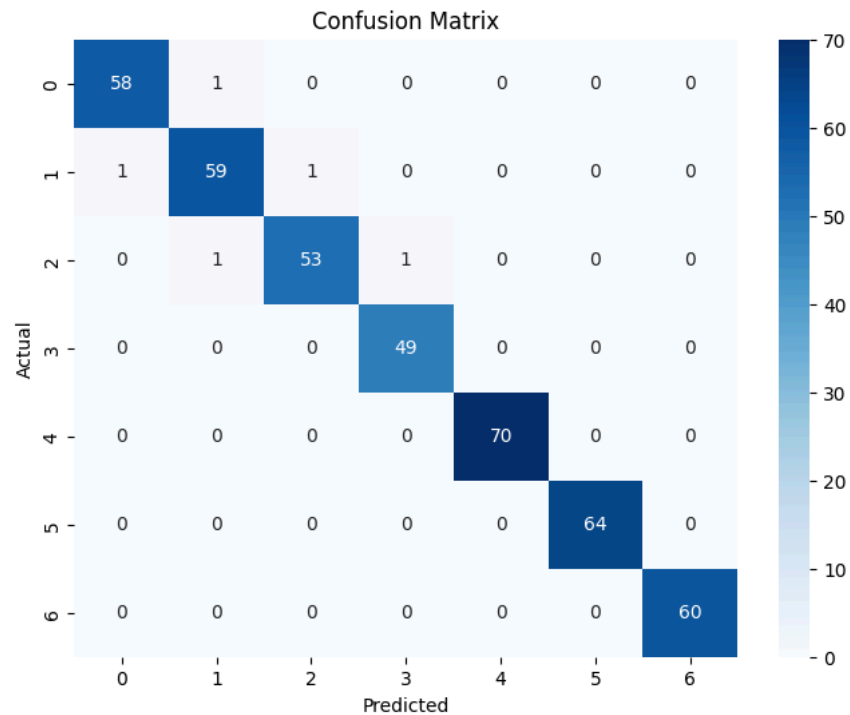
1. **Campionamento Bootstrap:** crea più sottoinsiemi di dati dal set di dati originale utilizzando il campionamento con sostituzione.
2. **Costruzione di alberi decisionali:** Per ogni sottoinsieme di dati, viene costruito un albero decisionale. Durante la costruzione di ciascun albero, viene selezionato un sottoinsieme casuale di attributi per ogni nodo per determinare la migliore suddivisione.
3. **Aggregazione:** aggrega le previsioni di tutti gli alberi decisionali per effettuare la previsione finale. Per la classificazione, la previsione finale è la classe che ottiene il maggior numero di voti dagli alberi. Per la regressione, la previsione finale è la media delle previsioni di tutti gli alberi.

Nel nostro caso, Random Forest è stato utilizzato per prevedere il livello di obesità di una persona in base alle sue caratteristiche. Il dataset è stato diviso in un set di addestramento e un set di test, il modello è stato addestrato sul set di addestramento e valutato sul set di test.

I principali iperparametri di Random Forest sono:

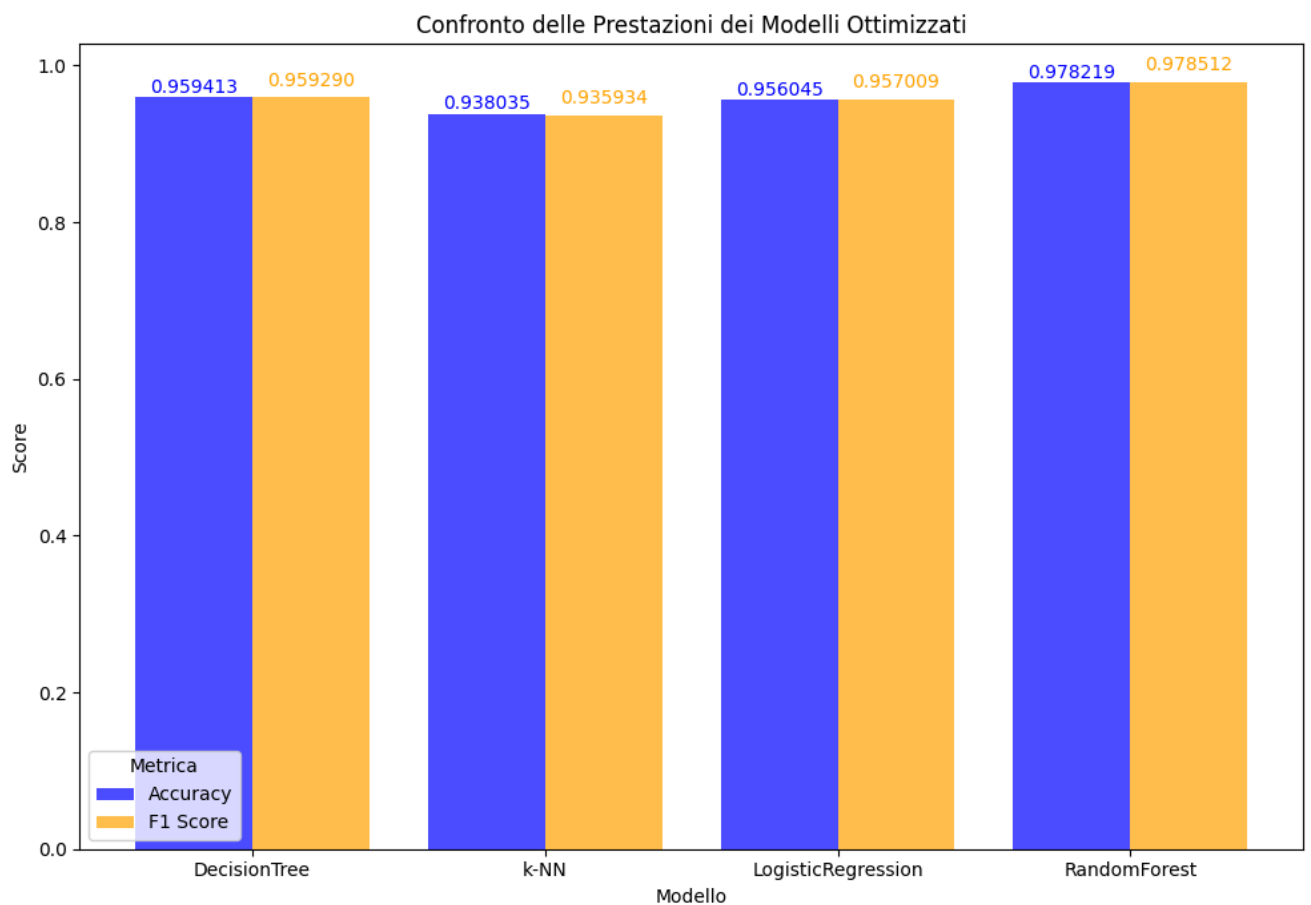
- **n_estimators:** il numero di alberi decisionali nella foresta.
- **max_depth:** la profondità massima di ogni albero decisionale.
- **min_samples_split:** il numero minimo di campioni richiesti per dividere un nodo interno.
- **min_samples_leaf:** il numero minimo di campioni richiesti in un nodo foglia.

Infine, il modello è stato valutato tramite le metriche di: **Precision, Recall** ed **F1**
 Si riporta la matrice di confusione generata dalla valutazione della Random Forest:



Confronto tra i modelli

Analizzando le metriche di ogni modello è possibile classificare i modelli in un grafico:



Dal grafico precedente possiamo notare che **tutti i modelli** hanno effettuato **ottime performance**, hanno accuracy e F1 score molto alti ($\approx 0.93 - 0.98$), indicando una buona capacità di classificazione.

Il Random Forest è il migliore, ha il punteggio più alto su entrambe le metriche:

- **Accuracy** = 0.9782
- **F1 Score** = 0.9785

Esso tende a performare meglio grazie alla combinazione di più alberi decisionali, riducendo l'**overfitting** (si tratta di **correlazioni spurie** nei dati che non si riflettono sull'intero dominio del problema; si verifica un **eccesso di fiducia** nelle predizioni del modello rispetto a quanto autorizzino i dati a disposizione) e aumentando la generalizzazione. Anche se leggermente inferiore, il Decision Tree ha ottenuto performance molto simili.

Il k-NN risulta essere il meno performante, ha i valori più bassi tra tutti i modelli:

- **Accuracy** = 0.9380
- **F1 Score** = 0.9359

Non performa bene come gli altri modelli in questo caso probabilmente perché:

- KNN calcola la distanza tra i punti dati, e feature con scale diverse possono influenzare eccessivamente il risultato. Nel dataset, alcune feature hanno range molto diversi (es. età vs. peso), e questo può penalizzare KNN.
- Il dataset ha molte feature e una complessità intrinseca. KNN, essendo un algoritmo "lazy" (anche detto **algoritmo di apprendimento basato su istanze**, un tipo di algoritmo di apprendimento automatico che rimanda la costruzione del modello fino a quando non viene richiesta una previsione) può avere difficoltà a catturare relazioni complesse tra le feature e il target. Gli altri modelli, come Decision Tree e Random Forest, sono in grado di modellare meglio queste relazioni.

Mentre la Logistic Regression il Decision Tree sono in una posizione intermedia, in quanto hanno prestazioni leggermente inferiori a Random Forest, ma migliori della k-NN.

Belief Network

Una **Belief Network** o (**rete bayesiana**) è un modello (a grafo) che evidenzia la **dipendenza condizionata** fra variabili. Definito un ordinamento totale sull'insieme delle sue variabili $\{X_1, \dots, X_n\}$, la distribuzione di probabilità congiunta potrà essere decomposta in termini di probabilità condizionate, tramite la **chain rule** (permette di calcolare la probabilità dell'intersezione di eventi non necessariamente indipendenti o la distribuzione congiunta di variabili casuali, rispettivamente, utilizzando le probabilità condizionali):

$$P(X_1 = v_1 \wedge X_2 = v_2 \wedge \dots \wedge X_n = v_n) = \prod_{i=1}^n P(X_i = v_i \wedge \dots \wedge X_{i-1} = v_{i-1})$$

quindi,

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1})$$

Formalmente, una BN è rappresentata da un **grafo aciclico orientato (DAG)** con una variabile, X, (con un suo dominio) per ciascun nodo, archi verso X da ciascuno dei nodi relativi ai suoi genitori, abbiamo quindi:

1. **Nodi**: Rappresentano le variabili casuali (discrete o continue).
2. **Archii diretti**: Indicano relazioni di dipendenza tra le variabili. Se esiste un arco da (X) a (Y), significa che (X) influenza direttamente (Y).
3. **Distribuzioni di probabilità**: Ogni nodo ha associata una probabilità condizionata rispetto ai suoi genitori nel grafo, rappresentata tramite la **Tabella di Probabilità Condizionata (CPT)**.

Alcuni vantaggi della BN sono:

- In grado di gestione l'incertezza, quindi è ideale per problemi in cui i dati sono **incompleti** o **rumorosi**;
- Assume un **ragionamento causale**, permettendo di modellare cause ed effetti;
- Ha un'**inferenza efficiente**, è possibile calcolare la probabilità di un evento dato un'evidenza osservata;
- E' facile da interpretare rispetto ad altri modelli probabilistici.

La fase iniziale di creazione della BN prevedeva una **discretizzazione delle feature**¹, ossia il **processo di conversione** di attributi, caratteristiche o variabili continue in attributi discretizzati, quindi definite come segue:

- la feature **"Age"** è stata suddivisa negli intervalli:
[14, 20] [20, 30] [30, 40] [40, 50] [50, 61]
- la feature **"Weight"** in intervalli, secondo la sua deviazione standard e arrotondati per difetto, in quanto la sua distribuzione si avvicina alla **distribuzione normale** (dato che il massimo 173 kg, è stato deciso di estendere invece creare un intervallo breve):
(39, 65] (65, 91] (91, 117] (117, 143] (143, 173]
- la feature **"Height"** è stata suddivisa negli intervalli:
[1.45, 1.54] [1.54, 1.63] [1.63, 1.72] [1.72, 1.81] [1.81, 1.90] [1.90, 1.99]
- la feature **"BMI"** è stata suddivisa negli intervalli presenti sul [paper](#):
[0, 18.5] (18.5, 24.9] (25, 34.9] (35, 39.9] (40, 60]

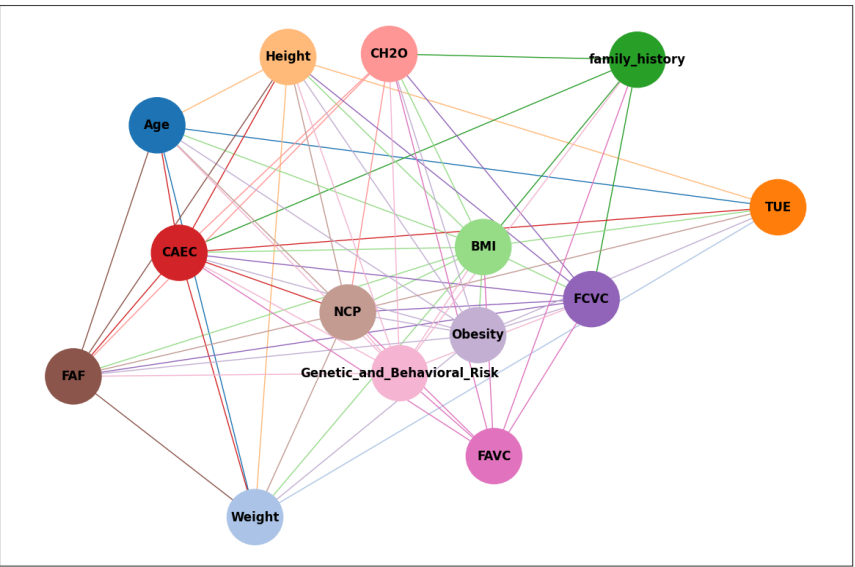
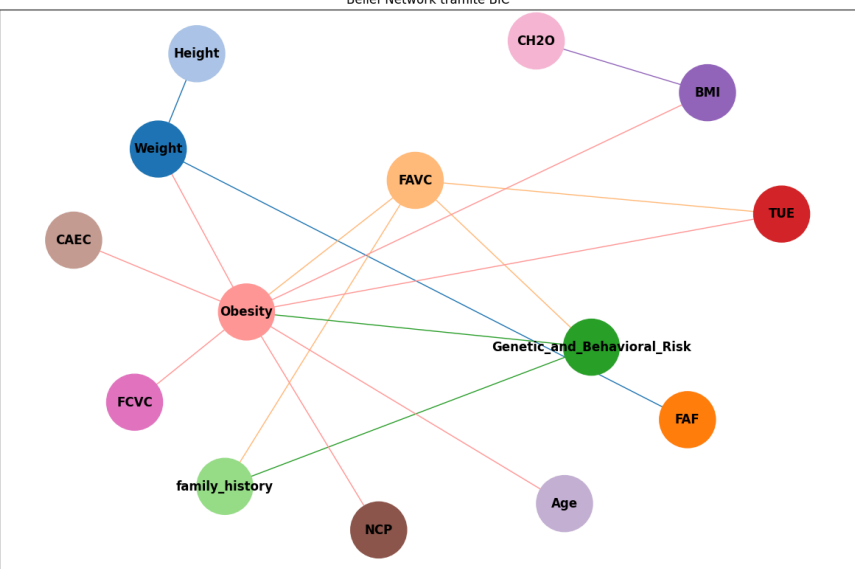
¹ rif. [Discretization of continuous features - Wikipedia](#)

Nella fase successiva, ci occuperemo di costruire tre Belief Network che si distinguono in base a come viene appreso il DAG di essa:

- Due Belief Network sono create tramite **Hill Climb Search**², un algoritmo di ricerca locale per massimizzare una funzione obiettivo utilizzata per valutare la bontà di una particolare struttura (insieme di dipendenze tra variabili) in una rete bayesiana. Per la creazione delle BN sono state decise le segue funzioni obiettivo:
 - **K2Score**, misura utilizzata per valutare la qualità di una struttura di rete bayesiana. Si basa sulla probabilità condizionata dei dati dati una specifica configurazione della rete e incorpora un termine che penalizza la complessità della struttura.
Questa misura favorisce modelli più semplici con meno dipendenze tra variabili, ma premia allo stesso tempo l'accuratezza del modello nel descrivere i dati osservati.
 - **BICScore (Bayesian Information Criterion)**, misura utilizzata per selezionare il modello migliore tra vari modelli concorrenti, bilanciando la bontà dell'adattamento del modello ai dati con la complessità del modello stesso. Più basso è il BIC, migliore è il modello, in quanto penalizza modelli troppo complessi che potrebbero sovradattare ai dati.
 -
- La terza Belief Network è stata creata attraverso una struttura scelta da noi dopo aver effettuato una serie di analisi approfondite sul dataset, definita come **Expert_Structure**. Le scelte strutturali sono state guidate da osservazioni empiriche riguardo le relazioni tra le variabili, l'intuizione sul dominio del problema e il comportamento dei dati stessi.

² rif. [Hill Climbing in Artificial Intelligence](#)

Di seguito ad un'analisi sono state osservate le Belief Network e considerate le seguenti:

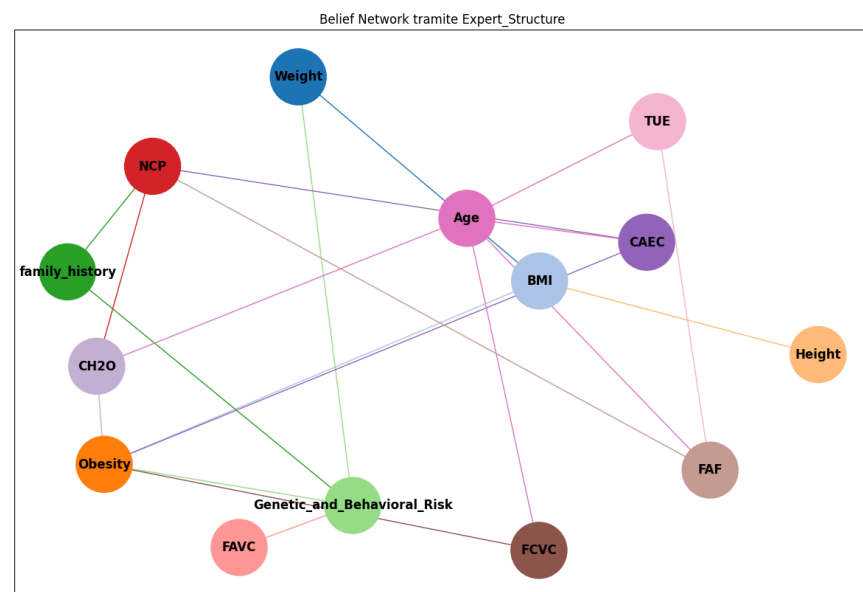
Analisi Metodo	Grafo Risultante
<p>BN_K2: La rete appare densamente connessa, con numerosi archi che collegano i nodi, potrebbe esserci un'elevata complessità nel grafo, suggerendo che molte variabili sono interdipendenti.</p> <p>I nodi come Obesity, BMI, family_history, e altri sembrano avere molteplici connessioni dirette, il che potrebbe indicare una modellazione eccessivamente dettagliata o poco parsimoniosa. L'alta densità di connessioni può portare a un modello che sovra-adatta i dati di addestramento, riducendo la sua capacità di generalizzare su dati nuovi. L'accuratezza relativamente bassa pari a 0.6866 supporta questa ipotesi, suggerendo che la rete potrebbe essere troppo complessa per catturare efficacemente le relazioni sottostanti.</p> <p>Il metodo K2, noto per costruire reti basandosi su un ordinamento dei nodi e una funzione di scoring, potrebbe aver generato una struttura che riflette più il rumore nei dati che le vere dipendenze causali.</p>	<p>Belief Network tramite K2</p> 
<p>BN_BIC: La rete appare meno densa rispetto a BN_K2, con un numero inferiore di archi tra i nodi. Le connessioni sembrano più selettive e mirate, suggerendo una migliore identificazione delle relazioni causali principali.</p> <p>Il nodo Obesity è probabilmente centrale, connesso a variabili chiave come BMI, NCP, FCVC, e altre correlate al comportamento e alla genetica. La maggiore accuratezza pari a 0.8971 indica che la rete BIC ha trovato un buon equilibrio tra complessità del modello e capacità predittiva. L'utilizzo del criterio BIC ha probabilmente penalizzato modelli troppo complessi, favorendo una struttura più parsimoniosa che si concentra sulle relazioni più significative.</p>	<p>Belief Network tramite BIC</p> 

BN_EXP: Alcune **connessioni** sono **simili** a quelle di BN_BIC, ma ci sono anche alcune aree con maggiore densità di archi, suggerendo che alcune relazioni potrebbero essere state modellate in modo più dettagliato.

I nodi centrali come Obesity e BMI sono ancora ben rappresentati, ma potrebbero esserci alcune differenze nelle connessioni secondarie rispetto a BN_BIC.

L'accuratezza 0.7990 è inferiore a quella di BN_BIC ma superiore a quella di BN_K2, suggerendo che la rete BN_EXP offre una soluzione di compromesso tra complessità e precisione.

La struttura visiva riflette questo compromesso, con alcune parti della rete che appaiono più dettagliate rispetto ad altre.



Inferenza sulle Belief Network

L'**inferenza in una Bayesian Network** consiste nel calcolare la probabilità di una o più variabili di interesse, dato che abbiamo osservato alcune evidenze. In pratica, ci permette di aggiornare le nostre credenze su un evento alla luce di nuove informazioni.

Ci sono diversi due tipi di inferenza:

- **Inferenza esatta:** Tale forma di inferenza richiede che le probabilità vengano calcolate precisamente, fondamentalmente gli approcci relativi a tale forma si dividono in due modalità principali:
 1. Enumerazione dei mondi consistenti (coerenti) con l'evidenza;
 2. Sfruttamento della struttura della rete: ad esempio l'algoritmo di eliminazione delle variabili, basato sulla programmazione dinamica, sfrutta le relazioni di indipendenza condizionata.

Prettamente utile nelle reti di piccole dimensioni.

- **Inferenza approssimata:** Esistono diversi metodi per stimare le probabilità in base al tipo di approssimazione:
 - metodi che forniscono **limiti garantiti** $[l, u]$ di variazione entro i quali ricada la probabilità esatta p , ad esempio un algoritmo anytime garantisce che l e u tendano ad avvicinarsi reciprocamente col passare del tempo (o con più spazio a disposizione);
 - metodi che forniscono **limiti probabilistici** sull'errore garantendo un errore contenuto in un'alta percentuale di casi, ovvero stime di probabilità che convergono nel tempo verso quella esatta e una certa velocità di convergenza;
 - metodi di **inferenza variazionale** capaci di fornire generalmente buone approssimazioni.

Prettamente utile nelle reti di grandi dimensioni.

Sistema di raccomandazione di abitudini per un livello di obesità normopeso tramite CSP

Un **CSP**, ossia, un **problema di soddisfacimento di vincoli** è definito da un insieme di variabili, ognuna con un proprio dominio, e un insieme di vincoli, una sua soluzione è un'assegnazione totale che soddisfa tutti i vincoli.

Per risolvere il **CSP**, utilizziamo un approccio basato su **backtracking** o **forward checking**, che sono tecniche comuni per risolvere problemi di soddisfacimento di vincoli. Nel nostro caso, utilizziamo la libreria **Python constraint** per implementare il CSP.

I passi per risolvere il CSP sono i seguenti:

1. **Definizione del Problema:** Creiamo un oggetto Problem dalla libreria constraint e aggiungiamo le variabili con i relativi loro domini e i vincoli da rispettare tra le variabili;
2. **Risoluzione del Problema:** Utilizziamo il metodo getSolutions() della libreria constraint per trovare tutte le soluzioni valide che soddisfano i vincoli. Se esistono soluzioni, mostriamo le 5 possibili soluzioni che permettono di avere un livello di obesità da normo peso;
3. **Conversione della Soluzione in Raccomandazioni:** Convertiamo le soluzioni in maniera di facile lettura da parte dell'utente.

Nel contesto del nostro progetto, il CSP viene utilizzato per generare raccomandazioni su abitudini di vita che permettono di avere un livello di obesità normale, le abitudini della persona che possono essere modificate per migliorare il livello di obesità. Nel nostro caso, le variabili con relativi domini sono:

- **family_history** (storia familiare di obesità): 0 = no, 1 = yes;
- **FAVC** (consumo frequente di cibi ad alto contenuto calorico): 0 = no, 1 = yes;
- **FCVC** (frequenza di consumo di verdure nei pasti): 1 = mai, 2 = a volte, 3 = sempre;
- **NCP** (numero di pasti principali giornalieri): 1 = uno, 2 = due, 3 = tre, 4 = più di tre;
- **CAEC** (consumo di snack tra i pasti): 0 = no, 1 = a volte, 2 = frequentemente, 3 = sempre;
- **CH2O** (consumo di acqua giornaliero): 1 = meno di un litro, 2 = tra 1 e 2 litri, 3 = più di 2 litri;
- **FAF** (frequenza dell'attività fisica settimanale): 0 = mai, 1 = 1 o 2 giorni, 2 = tra 2 e 4 giorni, 3 = 4 o 5 giorni;
- **TUE** (tempo di utilizzo di dispositivi tecnologici): 0 = meno di 2 ore, 1 = tra le 3 e 5 ore, 2 = più di 5 ore.

I vincoli definiti nel progetto (parte della definizione del problema) rappresentano una **serie di relazioni logiche** tra variabili correlate a comportamenti alimentari, stile di vita e fattori familiari che possono influenzare lo stato di salute, in particolare l'obesità. Ogni vincolo è espresso come una funzione lambda con due parametri (le variabili coinvolte) e una condizione che deve essere soddisfatta affinché il vincolo sia rispettato. Di seguito, analizziamo ciascun vincolo rilevato nel dettaglio:

- $FAVC = 1 \Rightarrow FCVC \in \{2, 3\}$

Se il consumo frequente di cibi ad alto contenuto calorico (**FAVC**) è presente, allora la frequenza di consumo di verdure (**FCVC**) deve essere **moderata** o **alta** (quindi, valori 2 o 3).

Una persona che mangia spesso cibi calorici dovrebbe compensare consumando abbastanza verdure per mantenere un equilibrio nutrizionale.

- $NCP \in \{3, 4\} \Rightarrow CH2O \in \{2, 3\} \text{ o } CH2O > 1$
Se il numero di pasti principali giornalieri (**NCP**) è 3 o 4, allora la quantità d'acqua bevuta giornalmente (**CH2O**) deve essere moderata o alta (valori 2 o 3). Altrimenti, se NCP è inferiore la quantità d'acqua consumata giornalmente deve essere uguale o maggiore di un litro.
Mangiare più pasti al giorno richiede un maggior apporto idrico per favorire la digestione e mantenere l'idratazione.
- $NCP \in \{3, 4\} \Rightarrow CAEC \in \{0, 1\} \text{ o } NCP = 1 \Rightarrow CAEC \in \{2, 3\} \text{ o } CAEC \in \{0, 1, 2\}$
Se il numero di pasti principali giornalieri (**NCP**) è 3 o 4, il consumo di cibo tra i pasti (**CAEC**) deve essere basso o nullo (valori 0 o 1). Se invece NCP è 1, il consumo di cibo tra i pasti può essere moderato o alto (valori 2 o 3) altrimenti il consumo tra pasti può essere tra nullo e moderato.
Una persona che mangia regolarmente tre o quattro pasti al giorno non ha bisogno di snacking³ frequenti; al contrario, chi mangia meno pasti potrebbe avere bisogno di integrare con snack.
- $TUE > 1 \Rightarrow FAF > 2 \text{ o } FAF > 0$
Se il tempo trascorso nell'uso di dispositivi tecnologici (**TUE**) è elevato (valori 2 o 3), allora la frequenza di attività fisica (**FAF**) deve essere moderata o alta (valori 2 o 3). Altrimenti per valori bassi di **TUE** l'attività fisica settimanale può essere anche bassa. Un'elevata esposizione ai dispositivi tecnologici può limitare il tempo dedicato all'attività fisica.
- $CAEC \in \{2, 3\} \Rightarrow CH2O \in \{2, 3\}, \text{ o } CAEC \notin \{2, 3\} \Rightarrow CH2O \geq 1$
Se il consumo di cibo tra i pasti (**CAEC**) è moderato o alto (valori 2 o 3), allora la quantità d'acqua bevuta giornalmente (**CH2O**) deve essere moderata o alta (valori 2 o 3). Altrimenti, se invece CAEC è basso o nullo (valori 0 o 1), la quantità d'acqua deve essere almeno moderata (≥ 1).
Chi mangia spesso tra i pasti potrebbe bere meno acqua, probabilmente sostituendola con altre bevande.
- $FAF \in \{2, 3\} \Rightarrow CH2O \in \{2, 3\}$
Se la frequenza di attività fisica (**FAF**) è moderata o alta (valori 2 o 3), allora la quantità d'acqua bevuta giornalmente (**CH2O**) deve essere moderata o alta (valori 2 o 3). Altrimenti, se FAF è bassa o nulla, la quantità d'acqua può essere bassa (1).
Una maggiore attività fisica aumenta il fabbisogno idrico.
- $family_history = 1 \Rightarrow FAF \in \{2, 3\}$
Se esiste uno storico familiare riguardante casi di obesità (quindi, **family_history** = 1), allora la frequenza di attività fisica (**FAF**) deve essere moderata o alta (valori 2 o 3). Altrimenti, se non c'è alcuna storia familiare, la frequenza di attività fisica può essere bassa o nulla (valori 0 o 1).
Una predisposizione genetica all'obesità richiede uno stile di vita più attivo per prevenire problemi di salute.

³ L'abitudine di fare spuntini al di fuori (o in sostituzione) dei pasti principali.

- $NCP > 1$

Il numero di pasti principali deve essere maggiore di 1 per garantire una corretta distribuzione dei nutrienti durante la giornata

Per la **risoluzione del problema** invece, la funzione `getSolutions()` della libreria "Problem" trova e restituisce tutte le soluzioni del problema nella forma di lista di dizionari che effettuano un mapping tra variabili e valori.

Queste soluzioni vengono, infine, convertite in suggerimenti più leggibili dall'utente ricevente.

Conclusioni

Il progetto "Obesity Insight" ha analizzato in maniera esauriente il dataset attraverso diversi campi.

Ci sono diversi punti migliorabili nel progetto, ad esempio:

- **Dataset:** Il dataset originale contiene una porzione significativa di dati generati sinteticamente, acquisire più dati reali potrebbe migliorare la generalizzazione dei modelli.
- **Feature Engineering:** Si potrebbero esplorare nuove feature derivate da quelle esistenti o integrare dati esterni, come informazioni socio-economiche o ambientali, per migliorare la capacità predittiva dei modelli.
- **Modelli:** Si potrebbero sperimentare altri algoritmi di apprendimento automatico, come **Support Vector Machines** o **Reti Neurali**, e ottimizzare ulteriormente gli iperparametri dei modelli esistenti per ottenere performance ancora migliori.
- **Valutazione:** Si potrebbero utilizzare metriche di valutazione più specifiche per il problema dell'obesità, come la sensibilità e la specificità per ogni classe di obesità.

Sviluppi futuri

Si potrebbe continuare a sviluppare il progetto concentrandosi sul miglioramento del sistema di raccomandazioni, passando da un approccio basato su valori ottimali generali ad uno personalizzato in base alle abitudini dell'utente e allo storico delle sue interazioni.

L'obiettivo è quello di fornire raccomandazioni più specifiche e rilevanti per ciascun utente, tenendo conto del suo **profilo** e delle **sue preferenze**, al fine di migliorare l'efficacia degli interventi di prevenzione e trattamento dell'obesità.

Un possibile approccio potrebbe dividersi in fasi del tipo:

1. **Raccolta Dati Storici:** Integrare il sistema con un database che memorizzi lo storico delle interazioni dell'utente, come:
 - Dati inseriti: età, peso, altezza, abitudini alimentari, attività fisica, ecc.
 - Raccomandazioni ricevute: tipo di raccomandazione, data, feedback dell'utente.
 - Risultati ottenuti: cambiamenti nel peso, nelle abitudini, ecc.
2. **Profilazione Utente:** Creare un profilo per ciascun utente, che includa:
 - Caratteristiche demografiche: età, genere, ecc.
 - Stile di vita: abitudini alimentari, livello di attività fisica, ecc.
 - Obiettivi: perdita di peso, mantenimento del peso, miglioramento della salute, ecc.
 - Preferenze: tipo di alimenti preferiti, attività fisica praticata, ecc.
3. **Sistema di Raccomandazioni Ibrido:** Sviluppare un sistema di raccomandazioni che combini:
 - Regole basate sulla conoscenza: linee guida generali per la prevenzione e il trattamento dell'obesità, basate su evidenze scientifiche.
 - Apprendimento automatico: modelli predittivi personalizzati, addestrati sui dati storici dell'utente, per prevedere l'efficacia di diverse raccomandazioni.
 - Filtraggio collaborativo: raccomandazioni basate sulle preferenze di utenti simili, per scoprire nuove opzioni potenzialmente interessanti.

4. **Feedback e Adattamento:** Integrare un meccanismo di feedback che permetta all'utente di valutare le raccomandazioni ricevute. Utilizzare questo feedback per:
 - Adattare il profilo dell'utente: aggiornare le preferenze e gli obiettivi in base alle risposte dell'utente.
 - Migliorare il sistema di raccomandazioni: perfezionare i modelli predittivi e le regole basate sulla conoscenza.

Un altro possibile sviluppo futuro si concentra sull'**arricchimento del dataset originale integrando** informazioni e **dati esterni** per ottenere predizioni più accurate e una comprensione più approfondita delle relazioni tra le feature. Così facendo è possibile migliorare la capacità predittiva e comprendere meglio i fattori che influenzano l'obesità, fornendo informazioni su relazioni più complesse tra le variabili.

Un approccio possibile potrebbe essere effettuato tramite:

1. **Identificazione di Dati Esterni Rilevanti:**
 - Dati Socioeconomici: Livello di istruzione, reddito, occupazione, accesso ai servizi sanitari, ecc.
 - Dati Ambientali: Densità di popolazione, disponibilità di spazi verdi, sicurezza alimentare, inquinamento atmosferico, ecc.
 - Dati Genetici: Predisposizione genetica all'obesità, varianti genetiche associate al metabolismo, ecc.
 - Dati Comportamentali: Utilizzo di social media, abitudini del sonno, stress, ecc.
 - Dati Nutrizionali: Composizione dettagliata degli alimenti consumati, micronutrienti, ecc.
2. **Acquisizione e Integrazione dei Dati:**
 - Fonti di Dati: Database pubblici (es. ISTAT, Eurostat), API di servizi web (es. Google Maps, OpenWeatherMap), studi scientifici, ecc.
 - Tecniche di Integrazione: Data linkage, data fusion, data augmentation.
3. **Analisi delle Relazioni tra Feature:**
 - Tecniche di Machine Learning: Feature selection, feature extraction, dimensionality reduction.
 - Modelli Statistici: Regressione multipla, analisi della varianza, analisi fattoriale.
 - Visualizzazione dei Dati: Grafici interattivi, mappe di calore, network graphs.
4. **Sviluppo di Modelli Predittivi Avanzati:**
 - Deep Learning: Reti neurali profonde per catturare relazioni non lineari complesse.
 - Ensemble Methods: Combinazione di modelli diversi per migliorare la robustezza e la generalizzazione.
 - Modelli Causali: Bayesian networks per comprendere le relazioni di causa-effetto tra le variabili.
5. **Valutazione e Validazione:**
 - Metriche di Performance: Accuratezza, F1-score, AUC, ecc.
 - Cross-Validation: Valutazione della generalizzazione dei modelli su nuovi dati.
 - Studi di Validazione Esterna: Confronto dei modelli con dati provenienti da fonti indipendenti.