Obesity Predictor

Francesco Ferrara 0512109789

23 gennaio 2024

Repository Github

Indice

1 Introduzione

L'obesità è una malattia che si caratterizza per un accumulo patologico di grasso corporeo con conseguenze anche importanti per lo stato di salute e la qualità di vita. Essa rappresenta uno dei maggiori problemi di salute pubblica a livello mondiale: secondo l'OMS i tassi di obesità sono triplicati dal 1975 ad oggi.
Nel 2016 più di 1,9 miliardi di adulti erano in sovrappeso, e tra questi più di 650 milioni erano obesi. Questa malattia incide in maniera decisa sulla durata della vita perché può comportare l'insorgenza di pressione alta, diabete mellito, apnee notturne e patologie cardiovascolari. Inoltre, elevati livelli di BMI indicano un maggiore fattore di rischio per l'insorgenza di tumori, come quello al seno e alla prostata.
Ci sono diversi fattori che incidono sul peso di una persona e che possono condurla all'obesità: la genetica e la dieta sono senza dubbio fra i più importanti. Tuttavia, una corretta alimentazione e una vita sana possono contrastare una genetica sfavorevole e portare una persona ad una condizione corporea meno rischiosa per la salute. Nei casi più gravi, però, un valido aiuto può derivare dalla chirurgia.

1.1 Obiettivi

Obesity Predictor nasce con lo scopo di fornire un aiuto alla categorizzazione della condizione fisica di una persona.
Gli obiettivi di questo progetto sono:

- L'analisi dei dati contenuti in un dataset.
- L'identificazione delle feature più rilevanti del problema.
- L'implementazione di un modello di Machine Learning che sia in grado di classificare correttamente la condizione corporea di una persona.

2 Specifica PEAS

Performance	La misura di performance dell'agente è la sua capacità di categorizzare correttamente lo stato del paziente. Questa verrà valutata tramite le metriche di valutazione quali precisione, accuratezza e recall.
Environment	L'ambiente in cui l'agente opera è quello dei dati clinici riguardo l'obesità, inclusi parametri come il peso e l'età. L'ambiente è: • **Completamente osservabile:** in quanto si ha accesso a tutte le informazioni in ogni momento; • **Stocastico:** in quanto le diagnosi possono essere influenzate da fattori con una certa variabilità; • **Sequenziale:** in quanto le azioni che la persona compie come la dieta o lo sport, hanno ripercussioni nel tempo; • **Dinamico:** in quanto la persona può evolvere durante il tempo; • **Discreto:**le variabili assumono valori in un determinato intervallo; • **A singolo agente:** in quanto l'unico agente che opera in questo ambiente è quello in oggetto.
Actuators	Gli attuatori dell'agente sono i modelli di machine learning addestrati.
Sensors	Il sensore dell'agente è rappresentato dalla tastiera (tramite essa l'agente riceve gli input) e dal dataset che contiene le informazioni relative alle caratteristiche.

3 CRISP-DM

Il modello che seguirò durante lo sviluppo del progetto è il CRISP-DM. Il CRISP-DM (Cross-Industry Standard Process for Data Mining) è un modello non sequenziale che rappresenta il ciclo di vita di progetti basati su intelligenza artificiale e data science.

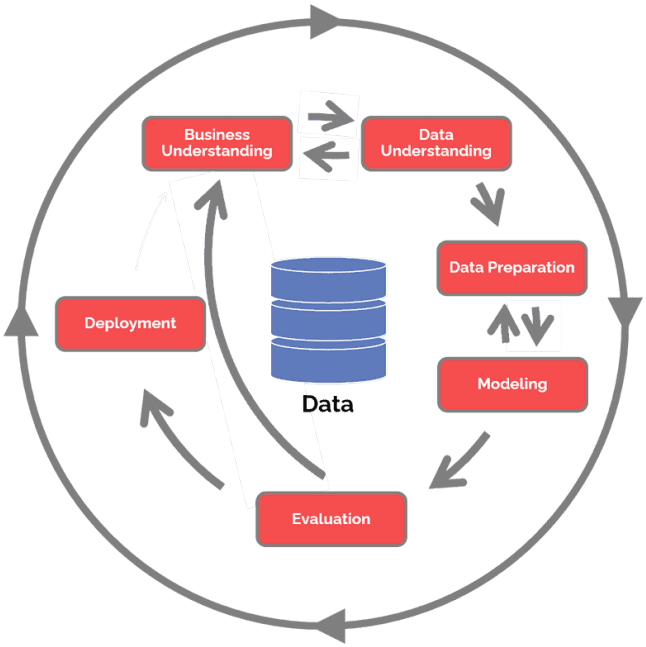

Figura 1: Modello CRISP-DM

Di seguito vengono spiegate nel dettaglio le varie fasi.

4 Business Understanding

La prima fase è quella di raccolta dei requisiti e di definizione degli obiettivi di business che si vogliono raggiungere (ossia cosa deve fare il machine learner) e dei cosiddetti business success criteria, ovvero i criteri secondo i quali potremo accertare che il sistema costruito è in linea con gli obiettivi di business. Inoltre bisogna determinare la disponibilità delle risorse, stimare i rischi e condurre una analisi costi-benefici. Oltre a ciò, vanno definiti gli obiettivi tecnici che si intendono raggiungere e le tecnologie ed i tool necessari agli obiettivi.

Andiamo quindi a definire i vari step:

- **Obiettivi di business**: Il machine learner da progettare avrà l'obiettivo di stimare o predire il valore della variabile dipendente: nel nostro caso dovrà predire lo stato di salute della persona, ossia se è obesa oppure no.
- **Risorse disponibili**: Per raggiungere l'obiettivo ci serviremo di un dataset contenente i dati relativi all'obesità.
- **Stima dei rischi**:Il problema in questione non è di facile risoluzione ma il dataset preso in esame risulta essere abbastanza completo.
- **Tecnologie e tool**: utilizzerò Kaggle per l'acquisizione del dataset, Python come linguaggio in quanto è il più adatto per l'analisi di dataset e Google Colab perchè ha integrato Jupyter Notebook per la creazione e visualizzazione delle azioni svolte sul dataset.

5 Data Understanding

La seconda fase consiste nell'identificazione, collezione e analisi dei dataset che possono portare al raggiungimento degli obiettivi definiti precedentemente. Inoltre andremo a descrivere e visualizzare i dati e ne controlleremo la qualità.

5.1 Identificazione del dataset

Per affrontare il problema in esame userò Kaggle, una piattaforma online che contiene migliaia di dataset su diversi ambiti.
In particolare, il dataset scelto è il seguente: https://www.kaggle.com/datasets/tathagatbanerjee/obesity-dataset-uci-ml?select=ObesityDataSet_raw_and_data_sinthetic.csv
Questo dataset è stato scelto rispetto ad altri due, che riguardavano la stessa analisi, in quanto più completo.

5.2 Analisi del dataset

Il dataset in questione si presenta ben strutturato ed è formato da 2111 righe e 17 colonne. A questo punto si può passare alla sua descrizione dettagliata.

5.2.1 Descrizione dei dati

I dati presentano le seguenti caratteristiche:

1. **Gender**: indica se il soggetto è maschio(Male) o femmina(Female);
2. **Age**: indica l'età dei partecipanti;
3. **Height**: indica l'altezza dei partecipanti in metri;
4. **Weight**: indica il peso dei partecipanti in kg;
5. **family_history_with_overweight**: è yes se un familiare del partecipante era o è obeso, no altrimenti;
6. **FAVC**: è yes se il soggetto mangia frequentemente cibi ad alto quantitativo calorico, no altrimenti;
7. **FCVC**: indica se il soggetto mangia verdure durante i pasti (1 sta per Never, 2 per Sometimes, 3 per Always);
8. **NCP**: rappresenta quanti pasti principali il soggetto ha durante la giornata, in una scala tra 1 e 4;
9. **CAEC**: indica se il soggetto mangia tra i pasti (no, Sometimes, Frequently, Always);
10. **SMOKE**: yes se il soggetto fuma, no altrimenti;
11. **CH2O**: indica quanta acqua beve il soggetto in una scala da 1 a 3;
12. **SCC**: yes se il soggetto monitora le calorie, no altrimenti;
13. **FAF**: indica quanto spesso il soggetto svolge attività fisica in una scala da 0 a 3.
14. **TUE**: indica quanto tempo il soggetto passa davanti a device elettronici in una scala da 0 a 2.
15. **CALC**: indica quanto spesso il soggetto beve (no, Sometimes, Frequently, Always);
16. **MTRANS**: indica quale mezzo di trasporto viene usato di solito dal soggetto(Automobile, Motorbike, Public_Transportation, Walking, Bike);
17. **NObeyesdad**: indica la condizione del soggetto (Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_Type_I, Insufficient_Weight, Obesity_Type_II, Obesity_Type_III);

La variabile dipendente scelta è **NObeyesdad**.

5.2.2 Data Quality

In questa sezione verranno esaminate le questioni relative alla presenza di dati duplicati e dati mancanti.

Grazie al metodo duplicated() di Python è possibile vedere che il dataset contiene 24 duplicati, ossia l'1,14% del totale. Dato che il numero è molto basso ho deciso di rimuoverli.
Per quanto riguarda i dati mancanti, grazie a Kaggle è possibile constatare che non ce ne sono.
Oltre a questo ho notato che alcune colonne presentano dati non interi: ad esempio nella colonna dell'età alcune celle sono 20.9, 22.5 ecc.
Credo quindi che sia opportuno correggere questi valori e approssimarli ad un intero.

Analizzando poi la variabile dipendente, essa presenta la seguente distribuzione delle etichette:

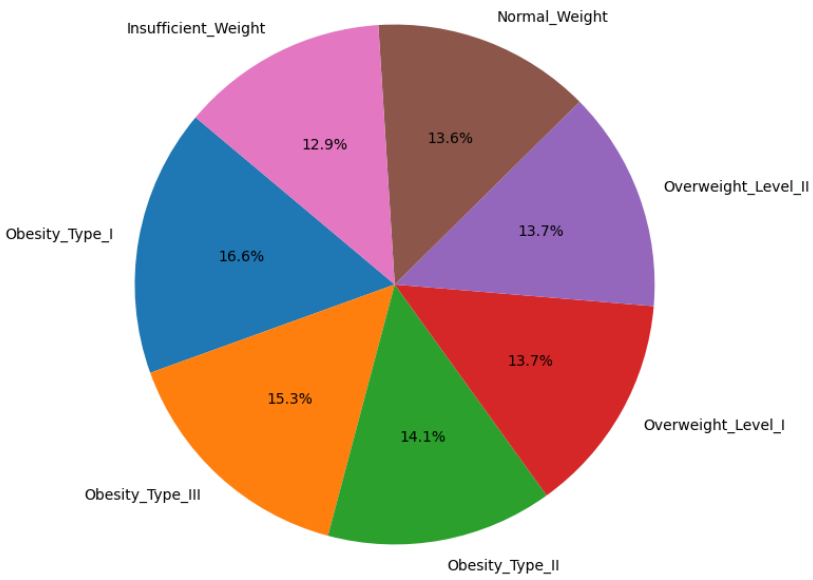

Figura 2: Distribuzione delle etichette di NObeyesdad

Sebbene la distribuzione sia già piuttosto bilanciata, credo sia opportuno cercare di migliorare la situazione tramite tecniche di undersampling o oversampling nelle fasi successive.
Andrà fatta un'analisi riguardo il loro numero: si può decidere di accorpare le etichette dell'Obesità e quelle relative all'Overweight, e quindi considerare solo 4 etichette totali.

6 Data Preparation

In questa fase del CRISP-DM verranno preparati i dati in modo che possano essere utilizzati da un algoritmo di Machine Learning. Le quattro operazioni che saranno effettuate sono:

1. **Data Cleaning:** verrà effettuata la pulizia dei dati, ossia rimozione di valori nulli e di duplicati ed eventuale data imputation;
2. **Feature Scaling:** verrà effettuata la trasformazione, normalizzazione e scalatura dei dati;
3. **Feature Selection:** verranno selezionate le caratteristiche più rilevanti del problema in esame;
4. **Data Balancing:** verranno applicate tecniche di oversampling o undersampling se necessario al bilanciamento del dataset.

6.1 Data Cleaning

In fase di Data Understanding è stata riscontrata la presenza di alcuni dati duplicati, quindi si procederà alla loro rimozione. Non sarà poi necessaria la rimozione di valori null in quanto è già stata verificata la loro assenza su Kaggle.

6.2 Feature Scaling

Alcune colonne come Age, Weight e Height presentano valori in R: sebbene questa scelta sia corretta per variabili come il peso e l'altezza, perde la sua rilevanza per un concetto come l'età.
Ho quindi ritenuto opportuno rendere interi i valori di Age e Weight, e approssimare alla seconda cifra decimale i valori di Height.
C'è poi lo stesso problema con le colonne CH2O, FAF, TUE, FCVC e NCP: siccome rappresentano valori in scala andrò a renderle intere.

A questo punto tutte le variabili numeriche sono intere, ma non tutte le feature del dataset hanno lo stesso dominio di valori: alcune colonne presentano dati numerici, mentre altre variabili categoriche. Dato che alcuni modelli di apprendimento potrebbero incontrare problemi con variabili categoriche, trasformo tutte le feature in variabili numeriche.
In particolare, le feature **Gender, family_history_with_overweight, FAVC, SMOKE ed SCC** avranno valori binari 0 ed 1, mentre le feature **CAEC, CALC ed MTRANS** avranno valori da 0 a 3 per le prime due e da 0 a 4 per MTRANS.

L'ultima analisi da fare riguarda il range di valori che assumono le variabili Age, Height Weight: esse assumono valori in un intervallo molto più ampio rispetto alle altre feature in gioco e questo potrebbe far pensare ad un machine learner che abbiano una rilevanza molto superiore.
Height è già riportata in metri, quindi non sarà necessario compiere un'ulteriore conversione. Per equilibrare le feature va compiuta la normalizzazione: ho usato la min-max normalization con la quale vengono identificati il valore massimo e il valore minimo della distribuzione, viene assegnato 1 al massimo e 0 al minimo e il nuovo valore sarà compreso nel nuovo range.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Figura 3: Min-Max normalization

Fatto ciò si può passare allo step successivo.

6.3 Feature Selection

A questo punto ciò che va fatto è analizzare il dataset per capire quali sono le feature più rilevanti e capire quali variabili hanno la correlazione maggiore con la variabile target.
Verranno analizzate la varianza delle feature e la correlazione delle variabili indipendenti con la variabile target. La decisione riguardo l'eliminazione o meno di feature avverrà in base alla combinazione di questi due aspetti.
Iniziamo con la varianza:

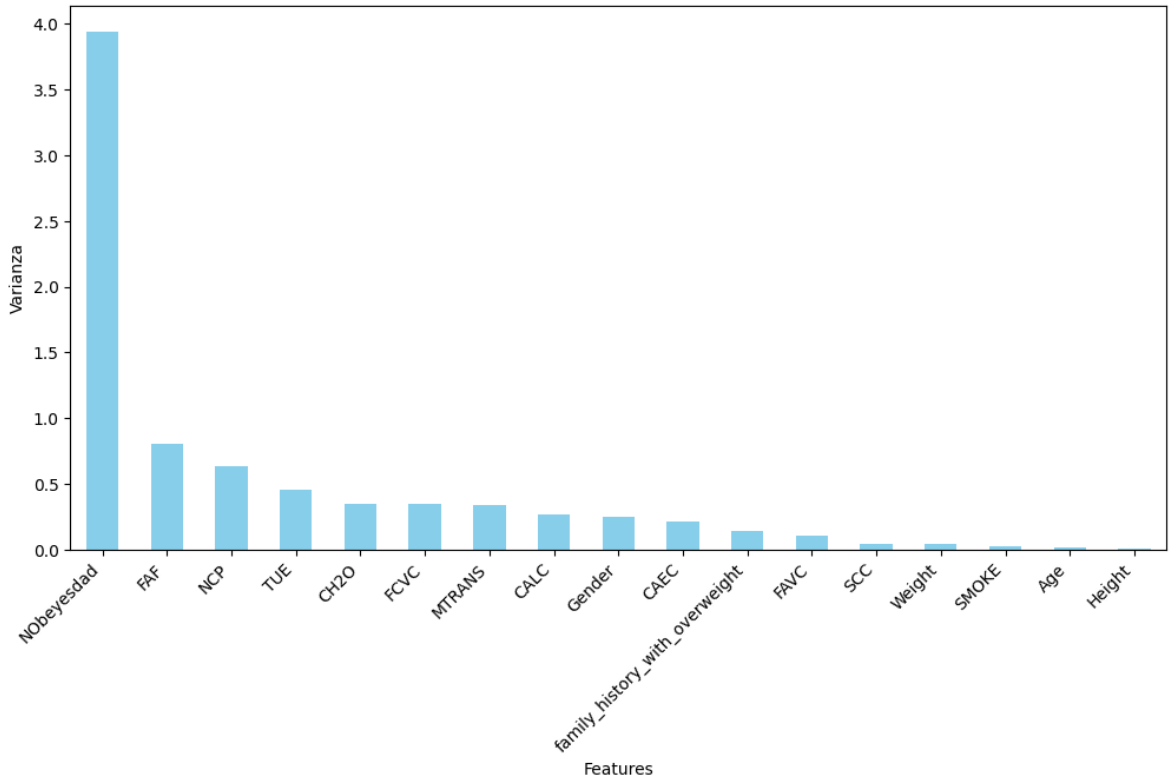

Figura 4: Varianza delle features

Come si vede dal grafico, le variabili con una varianza più bassa sono SCC, Weight, SMOKE, Age e Height.

La seconda analisi è stata compiuta tramite una heatmap:

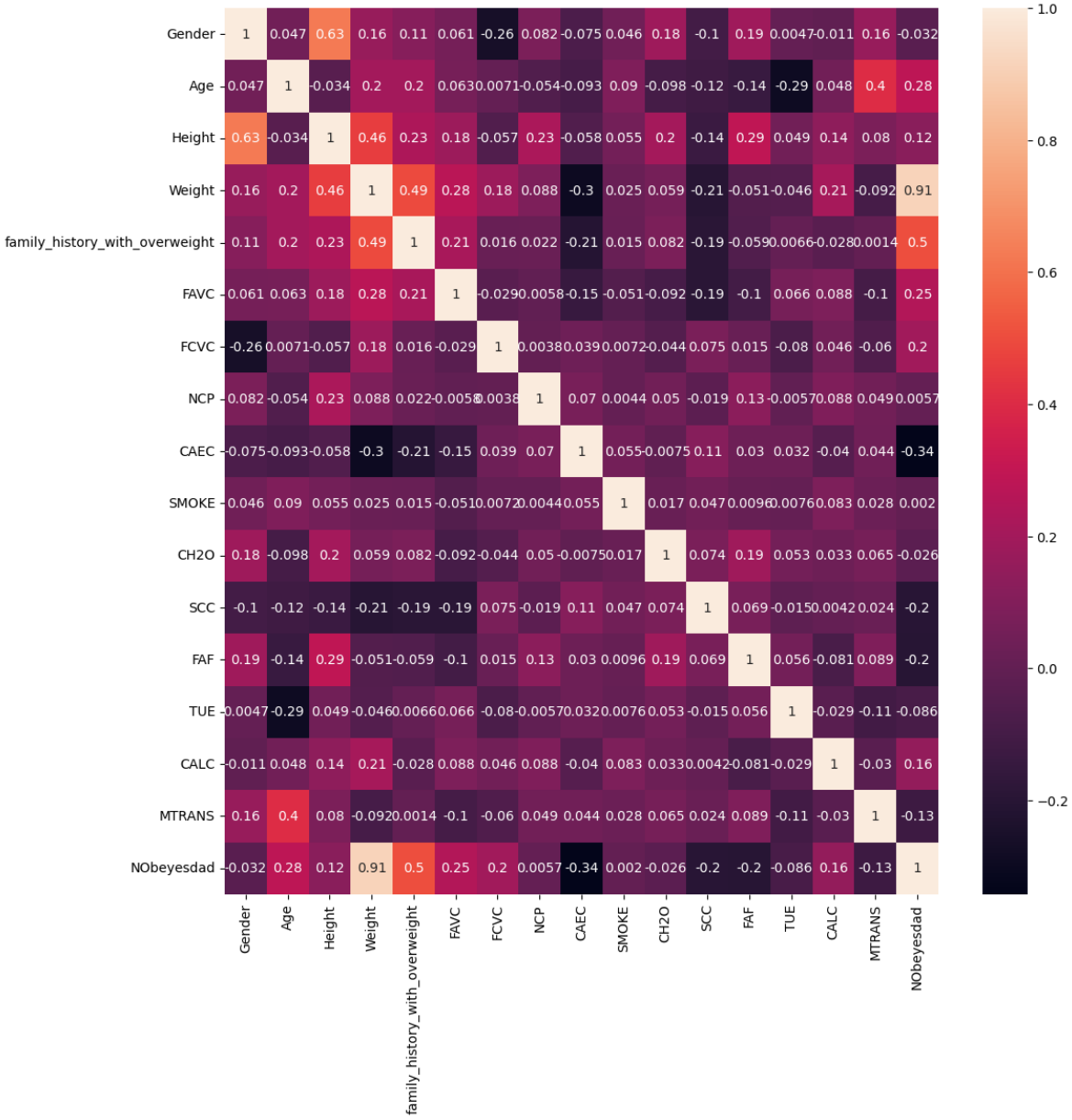

Figura 5: Matrice di correlazione

Nella heatmap, i colori più scuri rappresentano una correlazione minore tra variabile dipendente e indipendente. Quindi analizzando la riga NObeyesdad abbiamo che la variabile CAEC presenta un colore più scuro delle altre.

Le variabili con una correlazione negativa con la colonna target sono le seguenti:

	NObeyesdad
Gender	-0.032
CAEC	-0.342
CH2O	-0.026
SCC	-0.198
FAF	-0.203
TUE	-0.086
MTRANS	-0.127

Figura 6: Variabili con correlazione negativa

Le correlazioni peggiori riguardano CAEC, SCC, FAF e MTRANS. CAEC indica se il soggetto mangia tra i pasti, SCC se monitora le calorie, FAF se svolge attività fisica e MTRANS come si sposta.
La colonna SCC presenta valori bassi sia per la varianza che per la correlazione. CAEC invece ha una varianza maggiore ma è comunque la meno correlata a NObeyesdad.
Ho quindi deciso di eliminare queste due feature.

6.4 Data Balancing

In fase di Data Understanding ho scoperto che il dataset di partenza era già abbastanza bilanciato ma ritengo che si possa comunque migliorare il rapporto tra le etichette.

Le strade percorribili sono due:

1. **Undersampling:** il bilanciamento avviene eliminando un certo numero di istanze dal dataset;
2. **Oversampling:** il bilanciamento avviene andando ad aggiungere nuove istanze al dataset.

Ritengo che usare la tecnica dell'Undersampling sia eccessivo in quanto non c'è una grande sperequazione tra le etichette e inoltre andare a ridurre il numero di istanze vorrebbe dire ridurre l'addestramento del modello, e questa non è un'opzione. Andrò quindi ad utilizzare l'Oversampling ed in particolare la tecnica ADASYN (Adaptive Synthetic Sampling) per la generazione di nuove istanze.
Prima di fare Oversampling, però, il dataset verrà diviso in Training Set e Test Set con un rapporto di 67/33. Questa decisione è stata presa sulla base del fatto che i problemi reali molto spesso tendono ad essere sbilanciati e quindi è meglio addestrare il modello con una conoscenza uniforme di tutte le classi da predire ma testarlo su un dataset che sia il più vicino alla realtà, in modo da non avere problemi di data leakage.
Effettuata la divisione si può fare Oversampling e dopo ciò il Training Set avrà 2458 righe senza duplicati e il rapporto tra le classi sarà il seguente:

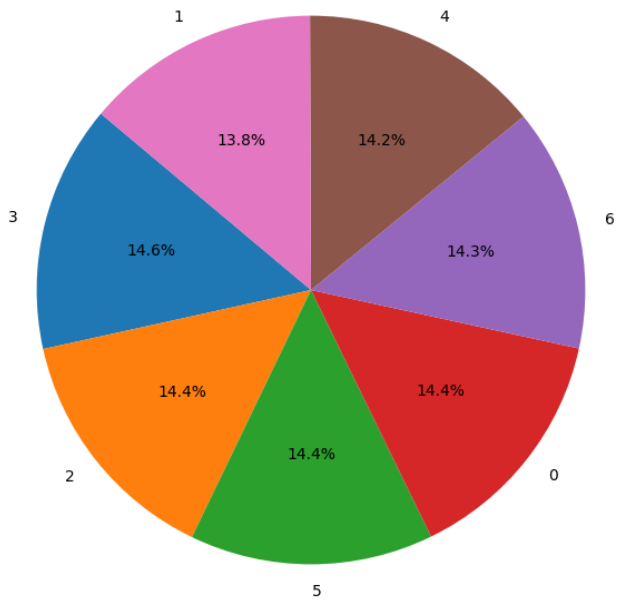

Figura 7: Distribuzione delle etichette dopo l'Oversampling.

7 Data Modeling

Dopo aver completato la fase di Data Preparation si può passare al Data Modeling, ossia la fase in cui vengono scelti gli algoritmi di Machine Learning da usare per l'addestramento del modello.
Essendo questo un problema di classificazione ho optato per Decision Tree e Random Forest.

7.1 Decision Tree

Il Decision Tree è un classificatore che mira a creare una struttura ad albero: ogni nodo rappresenta un sottoinsieme delle caratteristiche e le ramificazioni sono i valori che vengono assunti da esse. La radice dell'albero sarà costituita dalla caratteristica che divide meglio il dataset: vengono utilizzati i concetti di **Entropia** ed **Information Gain** per costruire l'albero decisionale.

Dopo aver usato il Decision Tree, la matrice di confusione risultante è stata:

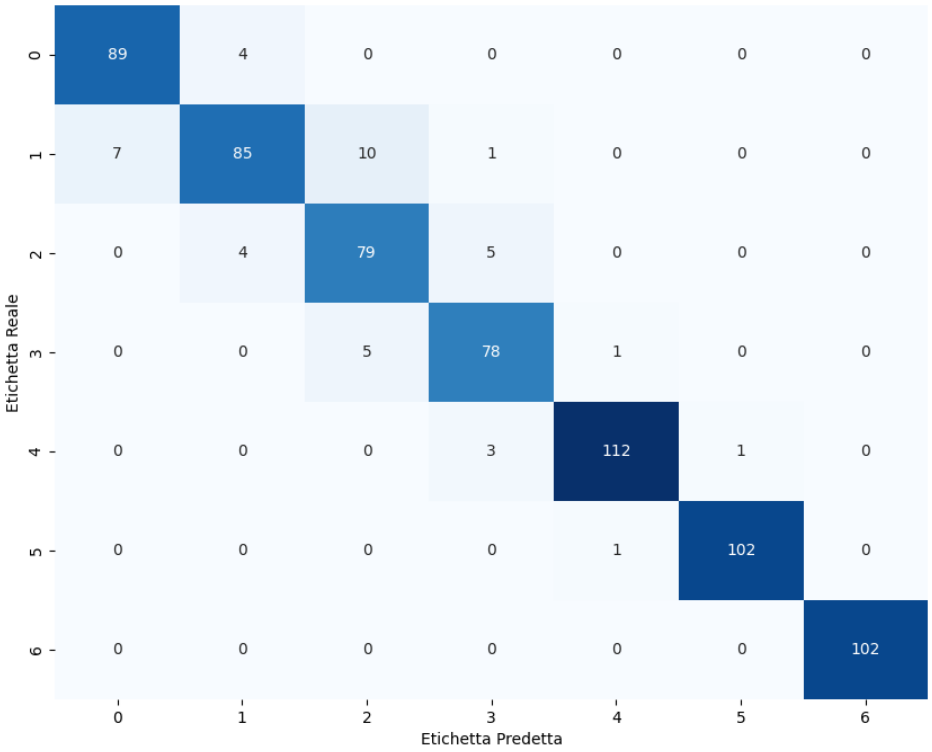

Figura 8: Matrice di Confusione Decision Tree

A seguire c'è il peso che ogni feature ha avuto per il DT:

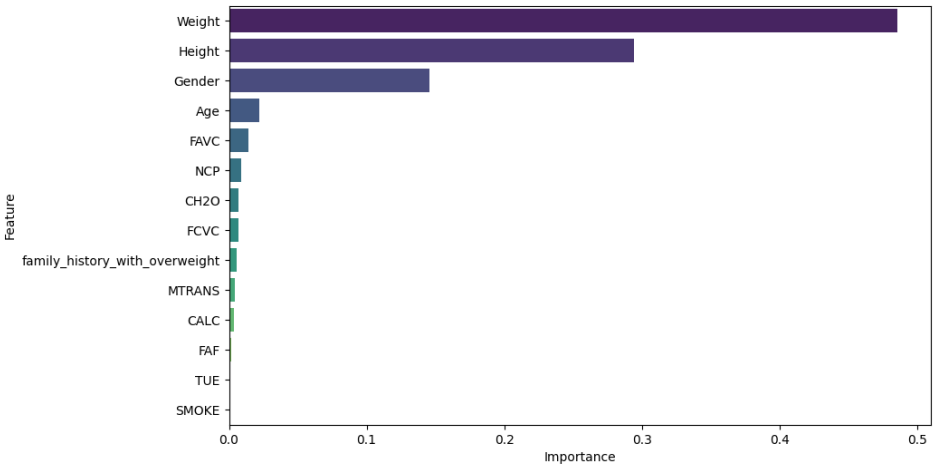

Figura 9: Feature Importance del Decision Tree

7.2 Random Forest

Il secondo algoritmo utilizzato è il Random Forest: esso mira a creare una foresta casuale che coinvolge la costruzione di un insieme di alberi decisionali, ognuno dei quali viene addestrato su un sottoinsieme casuale dei dati e delle feature. Le previsioni di ciascun albero vengono quindi combinate attraverso un processo di votazione (nel caso della classificazione) o di media (nel caso della regressione) per ottenere la previsione finale del modello. Essendo il problema in analisi di classificazione, verrà usato il processo di votazione.
Dopo aver usato il Random Forest, la matrice di confusione indica i seguenti risultati:

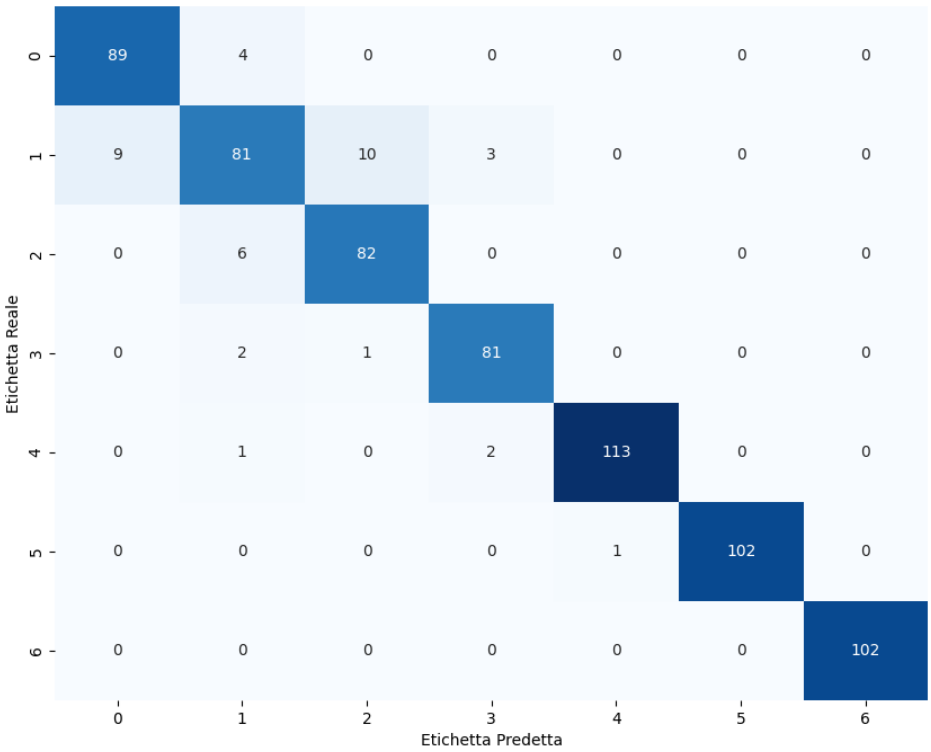

Figura 10: Matrice di Confusione Random Forest

A seguire c'è il peso che ogni feature ha avuto per il RF:

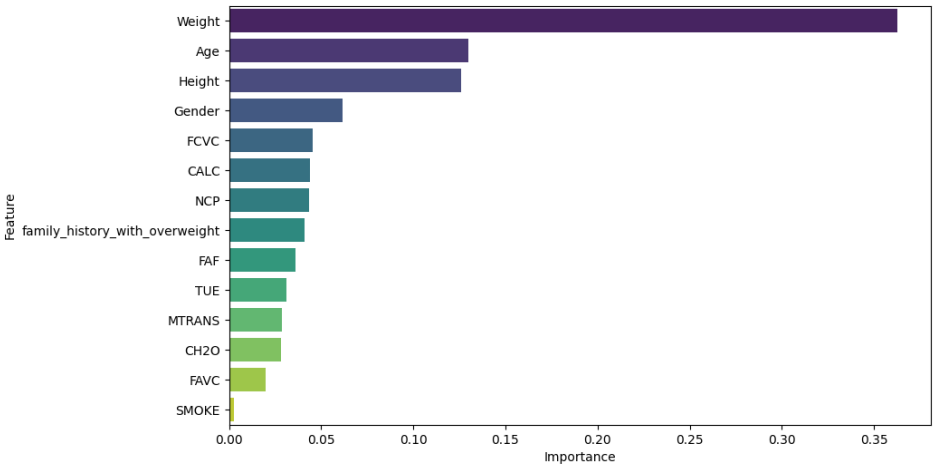

Figura 11: Feature Importance del Random Forest

8 Evaluation

In questa fase verranno valutati gli algoritmi secondo le metriche di Accuracy, Precision, Recall ed F1-Score. I due algoritmi hanno avuto prestazioni praticamente identiche.
Di seguito i valori delle metriche per il DT:

```
Decision Tree Accuracy: 0.939

Classification Report:
              precision    recall  f1-score   support

           0      0.927     0.957     0.942        93
           1      0.914     0.825     0.867       103
           2      0.840     0.898     0.868        88
           3      0.897     0.929     0.912        84
           4      0.982     0.966     0.974       116
           5      0.990     0.990     0.990       103
           6      1.000     1.000     1.000       102

    accuracy                          0.939       689
   macro avg      0.936     0.938     0.936       689
weighted avg      0.940     0.939     0.939       689
```

Figura 12: Report Decision Tree

e per il RF:

```
Random Forest Accuracy: 0.943

Random Forest Classification Report:
              precision    recall  f1-score   support

           0      0.908     0.957     0.932        93
           1      0.862     0.786     0.822       103
           2      0.882     0.932     0.906        88
           3      0.942     0.964     0.953        84
           4      0.991     0.974     0.983       116
           5      1.000     0.990     0.995       103
           6      1.000     1.000     1.000       102

    accuracy                          0.943       689
   macro avg      0.941     0.943     0.942       689
weighted avg      0.943     0.943     0.943       689
```

Figura 13: Report Random Forest

Analizzando più nel dettaglio i valori risultanti, si può notare che la media di precision, recall ed f1-score tra le 7 classi è sempre maggiore nel Random Forest.
Quindi si può concludere che l'algoritmo che ha performato meglio è stato il Random Forest.

9 Deployment

Dopo aver ottenuto il modello di Machine Learning finale, lo si può utilizzare per costruire la Demo. Viene scaricato tramite la funzione dump di Python il .joblib del modello e dello scaler in modo da poter essere usati nella Demo.
Il risultato sarà il seguente:

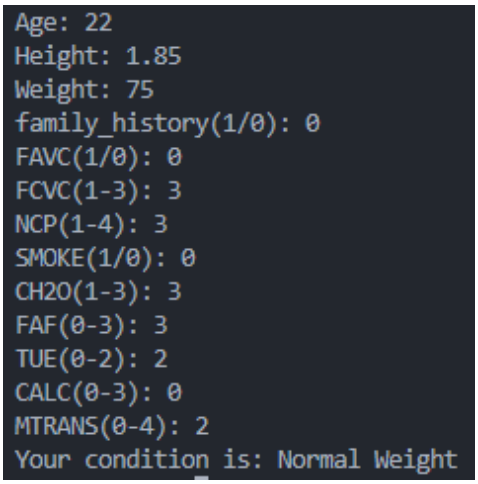

Figura 14: Deploy del tool

10 Conclusioni

Giunto alla fine di questo progetto posso ritenermi soddisfatto del lavoro svolto: ho potuto utilizzare nuove tecnologie e cimentarmi per la prima volta nell'analisi di un dataset.
Nonostante ciò, credo che ObesityPredictor possa essere migliorato con l'aggiunta di feature come il BMI e l'analisi della bodyfat dell'individuo, in modo da rendere la predizione ancora più accurata.

11 Bibliografia

https://www.sciencedirect.com/science/article/pii/S2352340919306985
https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight