PredizioneFilm

Gruppo di lavoro

- Francesco Antonelli, 776450, f.antonelli4@studenti.uniba.it

Link github: https://github.com/AntonelliFrancesco/ICON-24-25

AA 2024-2025

Indice

Introduzione

Nel progetto sviluppato, viene utilizzata una rete bayesiana per inferire la probabilità di successo di un film prima della sua uscita nelle sale. Il modello si basa su diverse variabili osservabili, come la popolarità, il budget di produzione oppure eventuali nomination ai premi cinematografici per determinare la probabilità che un film raggiunga il successo, sia economico che di popolarità (successo = 1). L'obiettivo principale è capire in che modo queste variabili influenzano il successo del film e come la rete bayesiana possa essere utilizzata per predire il risultato.

Nel corso di questo progetto, sono state utilizzate diverse librerie Python(versione 3.13.2) per facilitare la gestione dei dati, la creazione della rete bayesiana e l'addestramento dei modelli. Le principali librerie utilizzate includono:

- **pgmpy**: per la costruzione e l'inferenza nella rete bayesiana.
- **scikit-learn**: per il preprocessing dei dati, la creazione e la valutazione dei modelli di apprendimento supervisionato.
- **pandas**: per la gestione, la pulizia e la manipolazione dei dati.
- **numpy**: per le operazioni numeriche avanzate e la gestione di array, supportando il calcolo delle probabilità e le metriche di valutazione.
- **networkx**: per la visualizzazione della rete bayesiana.
- **matplotlib**: per la generazione di grafici e la visualizzazione dei risultati.
- **pickle**: per la serializzazione e deserializzazione dei modelli addestrati.

1)Creazione e preprocessing del dataset

Il dataset utilizzato in questo progetto è stato costruito unendo due set di dati scaricati dalla piattaforma Kaggle, contenenti informazioni relative ai film. Il primo dataset include variabili generali, come la popolarità e il genere dei film, mentre il secondo contiene informazioni più dettagliate sul successo economico dei film, come il box office, il budget di produzione, e il punteggio IMDb.

Dettaglio dei Dataset:

1. **Primo dataset:**
    - **movie_id**: Identificativo unico per ogni film.
    - **title**: Titolo del film.
    - **release_date**: Data di uscita del film).
    - **genre**: Categoria/i del film (comedy, action, etc.).
    - **overview**: Breve descrizione o riassunto del film.
    - **popularity**: Indice di popolarità del film, che riflette l'interesse del pubblico basato sulle interazioni con il film su TMDB. Questo dato è pre-release, poiché la popolarità può essere monitorata anche prima che il film esca nelle sale (tramite trailer, recensioni, articoli e interazioni generali).
    - **vote_average**: Punteggio medio del film basato sulle valutazioni degli utenti di TMDB. Può essere disponibile anche prima dell'uscita, se il film è già stato recensito o ha ricevuto valutazioni preliminari.
    - **vote_count**: Numero di voti ricevuti dagli utenti su TMDB, che è un altro indicatore dell'interesse del pubblico, anche prima dell'uscita del film.
2. **Secondo dataset:**
    - **movie**: Titolo del film.
    - **director**: Nome del regista del film.
    - **running_time**: Durata del film in minuti.
    - **actor_1, actor_2, actor_3**: I principali attori protagonisti del film.
    - **budget**: Il budget di produzione del film.
    - **box_office**: Gli incassi al botteghino, che rappresentano il successo economico del film.
    - **actors_box_office_percentage**: Percentuale che riflette quante volte gli attori hanno gestito un film che ha raddoppiato il budget in altri progetti.

Questo dato è utile per predire la probabilità di successo di un film sulla base della carriera dell'attore.

- **director_box_office_percentage**: Percentuale che indica quante volte il regista ha ottenuto il successo economico con altri film.
- **earnings**: Differenza tra incassi al box office e budget, che indica la redditività del film.
- **oscars_and_golden_globes_nominations**: Numero di nomination ricevute dal film agli Oscar o ai Golden Globe, che può essere annunciato anche prima dell'uscita del film. La presenza di queste informazioni può aumentare la visibilità e l'interesse del pubblico, rendendola utile come variabile predittiva.
- **oscars_and_golden_globes_awards**: Numero di premi vinti agli Oscar e ai Golden Globe.
- **imdb_score**: Punteggio IMDb, che viene aggiornato regolarmente dagli utenti e potrebbe essere disponibile anche prima dell'uscita.

Creazione del Dataset Finale

Per ottenere il dataset finale da utilizzare nel modello, è stato necessario unire due set di dati provenienti da fonti diverse. La chiave comune tra i due dataset era il **titolo del film**, che è stato utilizzato per combinare le informazioni relative ai film, come il budget, il box office e le nomination, con i dati più generali, come la popolarità e il punteggio IMDb. L'unione di queste informazioni ha consentito di disporre di un set di dati completo, utile per costruire e allenare il modello predittivo del successo cinematografico.

Il processo di unione è stato effettuato utilizzando la funzione merge di **pandas**, che ha combinato i due DataFrame in base alla colonna del titolo del film. Il codice utilizzato per eseguire l'operazione di merge è stato il seguente:

df_merged = pd.merge(df1, df2, how='inner', left_on='original_title', right_on='Movie')

In questo caso, la funzione merge è stata impostata con l'argomento how='inner', che ha garantito che solo i film presenti in entrambi i dataset venissero mantenuti nel dataset finale. L'uso di una **join interna** assicura che vengano eliminati i film per i quali non c'era una corrispondenza tra i due set di dati, riducendo quindi la possibilità di inserire dati incompleti o incongruenti.

Una volta uniti i dataset, sono state eseguite altre trasformazioni per preparare i dati per la modellazione, come la discretizzazione delle variabili e la creazione di nuove caratteristiche, che sono state poi utilizzate per addestrare il modello di rete bayesiana.

Discretizzazione dei Generi

Uno degli aspetti importanti nella preparazione dei dati è stata la gestione della colonna "genre", che nel dataset originale conteneva i generi di ciascun film come una lista di etichette. Per rendere questi dati utilizzabili da un modello di machine learning, ogni genere

di film è stato trasformato in una colonna binaria (0 o 1) che indica se un determinato genere è presente nel film. Questo è stato ottenuto utilizzando una tecnica di one-hot encoding per ogni genere presente nel dataset.

Introduzione delle Nuove Feature

Per rendere il dataset più informativo e utile per la classificazione del successo del film, sono state introdotte alcune nuove feature che considerano gli aspetti economici e di performance del film:

1. **ROI (Return on Investment):**

La feature ROI è stata calcolata come la differenza tra box office e budget divisa per il budget del film:

ROI = (Box office – Budget)/Budget

Questa feature misura il ritorno economico di un film rispetto alla sua spesa iniziale e serve come un indicatore importante del suo successo commerciale.

2. **Successo del Film (Target):**

La variabile target "successo" è stata introdotta per etichettare i film come successo (1) o insuccesso (0) sulla base di una serie di criteri.

Il film è stato etichettato come successo se soddisfaceva le seguenti condizioni:

- **ROI >= 0.8**
- **Punteggio IMDb >= 7**

Il 28,17% dei film nel dataset è stato considerato come un successo secondo queste metriche.

Rimozione di Colonne Testuali e Ridondanti

Oltre alla normalizzazione delle feature numeriche, sono state rimosse le colonne testuali e quelle ridondanti per semplificare il dataset e ridurre la complessità del modello. Le colonne rimosse includono:

- Titolo del film
- Nomi degli attori
- Nome del regista
- Overview del film
- Data e anno di uscita del film

La rimozione di queste colonne ha consentito di concentrarsi sulle variabili numeriche, che sono quelle realmente rilevanti per l'analisi del successo di un film, migliorando l'efficienza del modello.

2) Rete bayesiana

Una rete bayesiana è un modello grafico che rappresenta le relazioni probabilistiche tra variabili. Ogni nodo nel grafo rappresenta una variabile, mentre gli archi indicano dipendenze condizionate tra di esse. Questo approccio consente di inferire probabilità e relazioni causali, ed è utile in contesti incerti o complessi, come in questo caso per predire il successo di un film prima della sua uscita.

Utilizzando una rete bayesiana, si analizzano le probabilità condizionate di variabili per inferire sulla probabilità che un film abbia successo o meno.

Per costruire la rete bayesiana, sono state selezionate esclusivamente variabili numeriche e osservabili prima dell'uscita del film. Le variabili scelte includono:

- **Budget:** Il budget di produzione del film, che può influenzare le aspettative di successo.
- **Durata:** La lunghezza del film, che può avere un impatto sulla ricezione del pubblico.
- **Valutazione IMDb pre-release:** La valutazione media su IMDb prima del rilascio, che
- **popolarità:** Una misura dell'interesse pubblico, disponibile tramite la piattaforma TMDb, che indica il livello di attenzione e discussione sul film prima del rilascio.
- **Genere:** Le colonne che indicano i vari generi (e.g., azione, commedia, dramma) del film.
- **Risultati al box office di attori e registi**: Indicatori delle performance precedenti di attori e registi coinvolti nel film, che potrebbero predire il successo del film.
- **Eventuali nomination ai premi**: Se il film o il cast sono già stati nominati a premi prestigiosi, questo potrebbe contribuire ad aumentare l'interesse e il potenziale di successo.

Queste variabili sono state selezionate perché sono tutte disponibili prima dell'uscita del film e riflettono fattori che influenzano il successo, ma che non dipendono da dati post-lancio come incassi o recensioni finali.

Creazione della rete Bayesiana

La rete bayesiana è stata progettata per includere tutte le variabili selezionate come nodi genitori del nodo target “successo”. In questo modo, il successo di un film dipende direttamente da ciascuna variabile osservabile pre-release, come budget, popolarità, durata, performance storiche di attori e registi e nomination a premi. Questa configurazione consente di rappresentare in modo chiaro e interpretabile la relazione causale tra i fattori indipendenti e il risultato finale, facilitando l’analisi delle probabilità condizionate.

Per evitare di includere variabili poco informative, è stato calcolato il valore di **informazione mutua** tra ciascuna variabile e il nodo “successo”. Sono state selezionate le sei variabili con il punteggio più elevato tra quelle pre-release rimaste dopo l’esclusione delle variabili post-release (incassi, recensioni, premi vinti), non disponibili al momento della previsione.

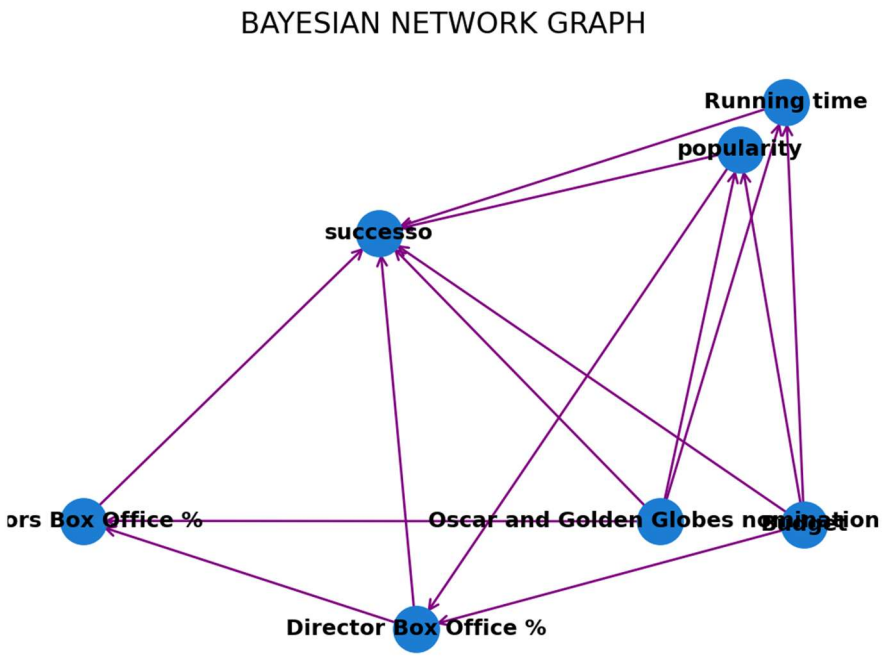
Inoltre sono state escluse anche variabili dei generi cinematografici che, pur essendo pre-release, generavano un numero eccessivo di configurazioni tra i nodi genitori, causando un appiattimento delle distribuzioni di probabilità condizionata (CPD) e riducendo la capacità discriminativa del modello.

Il numero di variabili selezionate (sei) corrisponde quindi all'insieme completo dei predittori informativi residui, garantendo un set compatto ma rappresentativo:

1. Oscar and Golden Globes nominations
2. Director Box Office %
3. Actors Box Office %
4. Budget
5. Popularity
6. Running time

Tutte le variabili numeriche selezionate sono state discretizzate per consentire la stima delle distribuzioni di probabilità condizionata su insiemi di valori finiti, come richiesto dalla rete bayesiana. La discretizzazione riduce la sensibilità del modello a valori anomali e consente di catturare pattern non lineari nelle relazioni tra predittori e variabile target. È stata adottata la strategia dei **quantili** (*quantile strategy*), che suddivide i dati in intervalli contenenti approssimativamente lo stesso numero di osservazioni. Questa scelta garantisce una rappresentazione più equilibrata delle classi di valori, evitando che intervalli scarsamente popolati compromettano la stima delle probabilità. In una fase preliminare è stato testato un partizionamento in tre intervalli ($k = 3$), ma le prestazioni non erano ottimali: alcune variabili mostravano ancora distribuzioni troppo compresse in uno o due bin. L'aumento a quattro intervalli ($k = 4$) ha permesso di catturare una maggiore variabilità interna, migliorando la capacità discriminativa e la stabilità delle stime probabilistiche.

La struttura della rete è stata appresa tramite l'algoritmo **Hill Climb Search**, ottimizzando il punteggio **K2Score**. Questo ha permesso di catturare correttamente le relazioni probabilistiche tra le variabili selezionate, migliorando l'accuratezza e riducendo la complessità computazionale. La rete finale viene rappresentata nel grafico qui sotto.



Fitting del Modello

Una volta preparato il dataset, è stato eseguito il fitting della rete bayesiana stimando le distribuzioni di probabilità condizionata per ciascun nodo tramite il metodo BayesianEstimator. La struttura di partenza, appresa con l'algoritmo Hill Climb Search e punteggio K2Score, è stata integrata con archi diretti da tutte le variabili predittive selezionate verso il nodo target "successo", così da modellare esplicitamente la dipendenza diretta di quest'ultimo da ciascun predittore. Per la stima delle probabilità è stato utilizzato un prior BDeu (Bayesian Dirichlet equivalent uniform) con equivalent sample size pari a 20, valore che garantisce una regolarizzazione adeguata senza penalizzare l'informazione proveniente dai dati di training. Test preliminari con valori inferiori (ad esempio 10) non hanno evidenziato variazioni significative nelle metriche, confermando la stabilità del modello rispetto a questo iperparametro. L'uso del prior ha contribuito a ridurre il rischio di overfitting e a mantenere stime robuste anche in presenza di combinazioni di variabili meno frequenti.

Inferenze e interpretazione probabilistica

Una volta definite le variabili e discretizzate in fasce, il modello Bayesiano consente di effettuare inferenze per stimare la probabilità di successo di un film in base a specifici scenari. Un esempio pratico di questa capacità di inferenza è la funzione **predici_successo()**, che simula un esempio casuale di film utilizzando la funzione **simulate()** dalla libreria **pgmpy**, che consente di generare un insieme casuale di variabili (come budget, popolarità, ecc.). Le variabili vengono quindi discretizzate e utilizzate per calcolare la probabilità di

successo. Ad esempio, se tutte le variabili vengono impostate ai loro valori massimi, il modello restituisce una probabilità condizionata di successo pari a circa l'80%. Sebbene questa cifra non rappresenti una certezza assoluta, essa fornisce una stima molto realistica: anche in condizioni ottimali, il successo non è mai garantito, ma è certamente molto probabile. Inoltre, l'inferenza permette di esplorare l'effetto marginale di ciascuna variabile, osservando come la probabilità di successo vari al variare di un singolo fattore, mantenendo gli altri costanti. La funzione **prob_successo_per_variabile()** è progettata per eseguire questa analisi in modo preciso, consentendo di studiare come cambiamenti in una variabile, come il budget o la popolarità, influenzino la probabilità che un film abbia successo. Ad esempio, fissando tutte le altre variabili, la funzione esamina la probabilità di successo del film per diversi valori di una singola variabile, restituendo un quadro dettagliato dell'impatto di ciascun fattore.

Questo approccio rende il modello interpretabile e utile non solo per generare scenari ipotetici, ma anche per supportare decisioni strategiche. L'implementazione di queste funzioni rende la rete bayesiana un potente strumento per esplorare rapidamente combinazioni plausibili di variabili e testare la sensibilità del modello a cambiamenti in fattori chiave. Inoltre, esse forniscono una panoramica più ampia del comportamento probabilistico del modello, rendendo il progetto più dinamico e interattivo per l'analisi predittiva del successo di un film.

Fase di Addestramento e Validazione

Il modello è stato addestrato utilizzando una **cross-validation** a 5 fold, con 5 ripetizioni, per garantire che il modello non soffrisse di **overfitting** e che i risultati fossero robusti e riproducibili. Per ogni fold, il modello è stato addestrato sui dati di addestramento e validato sui dati di test. Le metriche principali utilizzate per valutare la performance del sistema sono state:

- **Accuracy:** Percentuale di previsioni corrette sul totale delle osservazioni. Indica quanto il modello è globalmente preciso, ma può essere fuorviante in caso di classi sbilanciate.
- **Precision:** Rapporto tra i veri positivi e il totale dei positivi predetti. Misura quanto il modello è affidabile quando predice la classe positiva (successo).
- **Recall:** Rapporto tra i veri positivi e il totale dei positivi reali. Indica la capacità del modello di individuare correttamente tutti i casi positivi.
- **F1-score:** Media armonica tra precision e recall. È particolarmente utile quando le classi sono sbilanciate, poiché bilancia l'attenzione tra falsi positivi e falsi negativi.
- **ROC-AUC:** (Area sotto la curva ROC) Misura la capacità del modello di distinguere tra le classi. Un valore vicino a 1 indica un'eccellente separabilità tra "successo" e "insuccesso".
- **Train Error / Test Error:** Percentuale di errore commesso rispettivamente sui dati di addestramento e sui dati di test. Serve a valutare l'eventuale overfitting o underfitting del modello.

Valutazioni

In questa sezione si presentano e confrontano i risultati della rete bayesiana, per valutare l'efficacia delle scelte progettuali e misurare l'impatto di ciascuna configurazione sulle metriche di performance.

Tutte le metriche riportate nelle tabelle sono espresse come media ± deviazione standard, calcolate su 25 run (cross-validation a 5 fold con 5 ripetizioni). Tra parentesi è indicata la varianza.

Inizialmente si è scelto di discretizzare le variabili numeriche utilizzando *KBinsDiscretizer* con **k = 3** e strategia dei quantili, come descritto precedentemente. Questa configurazione ha rappresentato la base di partenza per la valutazione delle prestazioni del modello e si sono ottenuti i seguenti risultati:

Metrica	**Media ± Dev. Std**	**Varianza**
Accuracy	0.799 ± 0.011	0.00012
Precision	0.707 ± 0.033	0.00107
Recall	0.490 ± 0.022	0.00049
F1-score	0.579 ± 0.023	0.00052
ROC-AUC	0.856 ± 0.010	0.00010
Train Error	0.201 ± 0.003	0.00001
Test Error	0.201 ± 0.011	0.00012

In questa configurazione iniziale il modello raggiunge un'accuracy di circa 80% e un'AUC-ROC di 0.856, segno di una discreta capacità di distinguere tra successi e insuccessi. Il recall (0.49) rivela però una sensibilità limitata verso la classe "successo": quasi metà dei casi positivi non viene riconosciuta. La precision (0.707) indica che, quando il modello predice un successo, nella maggior parte dei casi è corretto, ma la copertura resta incompleta. L'F1-score (0.579) riflette lo squilibrio tra precision e recall. Gli errori di training e test, identici (0.201), mostrano un buon equilibrio tra apprendimento e generalizzazione, senza overfitting. La bassa varianza conferma la stabilità del modello, ma c'è margine di miglioramento soprattutto nella capacità di individuare i casi positivi.

Per questo ho scelto di passare alla configurazione a 4 bin, così da aumentare la granularità della classificazione e intercettare meglio le sfumature di performance, in particolare nei casi borderline. L'obiettivo è migliorare la sensibilità verso i successi senza compromettere la precisione complessiva.

Metrica	Media ± Dev. Std	Varianza
Accuracy	0.897 ± 0.010	0.00011
Precision	0.874 ± 0.021	0.00044
Recall	0.740 ± 0.035	0.00121
F1-score	0.801 ± 0.023	0.00051
ROC-AUC	0.967 ± 0.005	0.00002
Train Error	0.103 ± 0.003	0.00001
Test Error	0.103 ± 0.010	0.00011

Il passaggio da 3 a 4 bin ha prodotto un miglioramento netto in tutte le metriche. L'errore di training e di test si riduce di circa il 10%, senza alcun segnale di overfitting, mentre la varianza tra i fold cala ulteriormente, indicando una maggiore stabilità del modello. Quindi l'aumento della granularità nella discretizzazione ha avuto un impatto positivo sia sull'accuratezza sia sulla robustezza complessiva delle previsioni.

3) Apprendimento supervisionato

Oltre alla rete bayesiana, per il problema in esame sono stati considerati diversi modelli di **apprendimento supervisionato** con l'obiettivo di confrontarne le prestazioni predittive e individuare l'algoritmo più adatto al contesto applicativo. L'analisi si è concentrata su un compito di **classificazione binaria**, in cui la variabile target assume due possibili stati (*successo* / *insuccesso*). Il dataset presenta una moderata asimmetria nella distribuzione delle classi, con una percentuale di successi pari a circa **28%**, condizione che può influenzare le metriche di valutazione e richiede particolare attenzione nella scelta degli algoritmi e delle strategie di validazione.

L'obiettivo di questa fase è duplice:

- **Valutare** le prestazioni di modelli supervisionati eterogenei in termini di accuratezza, capacità di generalizzazione e robustezza.
- **Confrontare** i risultati con quelli ottenuti dalla rete bayesiana, per verificare se approcci discriminativi possano offrire un vantaggio in termini predittivi.

Modelli selezionati

Il **Decision Tree** è un classificatore strutturato ad albero, in cui ogni nodo interno rappresenta una condizione logica su una variabile di input e le foglie contengono la classe predetta o la distribuzione di probabilità sulle classi. È stato scelto per la sua semplicità, l'elevata interpretabilità e la capacità di fornire una rappresentazione chiara del processo decisionale, utile come baseline per confrontare modelli più complessi.

La **Random Forest** è un metodo ensemble che combina più alberi decisionali addestrati su sottoinsiemi differenti del dataset, secondo la tecnica del *bagging*. La predizione finale deriva dall'aggregazione delle uscite dei singoli alberi, riducendo la varianza e l'overfitting tipico di un albero singolo. È stata selezionata per la sua robustezza, la stabilità delle prestazioni anche in presenza di rumore nei dati e la capacità di stimare l'importanza delle variabili.

Il **Gradient Boosting** è un metodo ensemble sequenziale in cui ogni nuovo albero viene addestrato per correggere gli errori del modello precedente. La combinazione di tutti gli alberi produce un classificatore potente e flessibile, capace di modellare relazioni complesse e non lineari. È stato scelto per l'elevata capacità predittiva e per la possibilità di regolare finemente il compromesso tra bias e varianza, adattandosi bene a scenari in cui si ricerca la massima accuratezza mantenendo una buona generalizzazione.

Ricerca e ottimizzazione degli iperparametri

Per ciascun modello considerato è stata definita una griglia di iperparametri da esplorare, selezionata sulla base delle caratteristiche intrinseche di ciascun algoritmo, delle indicazioni presenti in letteratura e di valutazioni preliminari sui dati.

La ricerca è stata condotta tramite Grid Search integrata con Repeated Stratified K Fold Cross Validation (5 fold ripetuti 5 volte). In questo approccio, per ogni combinazione di iperparametri, il modello viene addestrato e validato su ciascun fold, calcolando la metrica di ottimizzazione scelta; il punteggio medio sui fold determina la bontà della combinazione. Al termine, la Grid Search restituisce il set di iperparametri che ha ottenuto il valore medio più alto della metrica selezionata. Con tali parametri, il modello viene quindi riaddestrato sull'intero training set e valutato nuovamente in cross validation.

La metrica di ottimizzazione adottata è stata l'F1 score nella sua forma standard, coerente con quella impiegata nella valutazione del classificatore Naïve Bayes. Nel contesto di un problema di classificazione binaria, questa metrica misura la media armonica tra precision e recall calcolate sulla classe positiva (etichetta 1), fornendo un indicatore sintetico della capacità del modello di individuare correttamente i positivi senza trascurare il controllo dei falsi positivi.

La scelta di ottimizzare per F1 score, anziché per la sola accuracy, è stata dettata dalla distribuzione non perfettamente bilanciata delle classi (la classe positiva rappresenta circa il 28% del dataset). In tali condizioni, l'accuracy può risultare fuorviante, premiando modelli che privilegiano la classe maggioritaria. L'F1 score, invece, penalizza in modo più equo le prestazioni insufficienti sulla classe minoritaria, incentivando la ricerca di un compromesso ottimale tra sensibilità e precisione.

Implicazioni sulla Complessità Computazionale

La ricerca degli iperparametri tramite Grid Search e Repeated Stratified K Fold Cross Validation comporta un notevole costo computazionale. In particolare, modelli complessi come Gradient Boosting richiedono diversi minuti per ogni ciclo di ottimizzazione, aumentando significativamente il tempo di addestramento.

Questo impatto è più evidente nei modelli come Random Forest e Gradient Boosting, che necessitano di maggiore potenza computazionale rispetto a modelli più semplici come Naïve Bayes. In scenari reali, dove i tempi di risposta sono critici, l'efficienza computazionale diventa un fattore limitante.

Per ottimizzare il processo, si potrebbero esplorare approcci alternativi come Random Search o ottimizzazione bayesiana (es. Optuna, Hyperopt), che sono più efficienti in termini di tempo. Inoltre, tecniche come early stopping o una riduzione dei fold di cross-validation potrebbero ridurre ulteriormente il costo computazionale senza compromettere troppo la qualità dei risultati.

Fase di Addestramento e Validazione

Per la valutazione delle prestazioni dei modelli sono state utilizzate le stesse metriche descritte nella sezione dedicata alla Rete Bayesiana (Accuracy, Precision, Recall, F1-score, ROC-AUC, Train Error e Test Error), alle quali si rimanda per la definizione. Tutte le metriche

riportate nelle tabelle sono espresse come media ± deviazione standard, calcolate su 25 run (cross-validation a 5 fold con 5 ripetizioni).

Random Forest

Metrica	**Media ± Dev. Std**	**Varianza**
Accuracy	0.843 ± 0.013	0.000180
Precision	0.847 ± 0.012	0.000146
Recall	0.843 ± 0.013	0.000180
F1-score	0.731 ± 0.021	0.000441
ROC-AUC	0.902 ± 0.010	0.000099
Train Error	0.082 ± 0.003	0.000009
Test Error	0.157 ± 0.013	0.000180

La Random Forest ottimizzata raggiunge un'accuracy media dell'84,3% e un ROC-AUC di 0.902, segno di una buona capacità di separare le classi. L'F1-score (0.731) indica un equilibrio discreto tra precision e recall, con una leggera prevalenza della precision. Il train error (0.082) è contenuto e il gap con il test error (0.157) è moderato, segnalando un rischio di overfitting limitato. La bassa varianza conferma la stabilità del modello nei diversi split.

Metrica	**Media ± Dev. Std**	**Varianza**
Accuracy	0.851 ± 0.013	0.000175
Precision	0.847 ± 0.014	0.000190
Recall	0.851 ± 0.013	0.000175
F1-score	0.717 ± 0.026	0.000660
ROC-AUC	0.902 ± 0.008	0.000070
Train Error	0.055 ± 0.002	0.000005
Test Error	0.149 ± 0.013	0.000175

Il Gradient Boosting mostra un'accuracy media dell'85,1% e un ROC-AUC di 0.902. L'F1-score (0.717) è leggermente inferiore a quello della Random Forest, ma con un recall più bilanciato. Il train error molto basso (0.055) e il gap con il test error (0.149) indicano una maggiore aderenza ai dati di training, con un potenziale rischio di overfitting mitigato dalla stabilità delle metriche.

Metrica	**Media ± Dev. Std**	**Varianza**
Accuracy	0.793 ± 0.018	0.000333
Precision	0.821 ± 0.013	0.000157
Recall	0.793 ± 0.018	0.000333
F1-score	0.683 ± 0.021	0.000438
ROC-AUC	0.859 ± 0.018	0.000307
Train Error	0.186 ± 0.011	0.000126
Test Error	0.207 ± 0.018	0.000333

Il Decision Tree, pur essendo il modello più semplice, ottiene un'accuracy media del 79,3% e un ROC-AUC di 0.859. L'F1-score (0.683) è inferiore rispetto agli altri modelli, soprattutto a

causa di un recall più basso. Il train error e il test error sono vicini, segno di un modello poco incline all'overfitting ma con capacità predittiva più limitata.

Analisi comparativa

Il confronto tra i tre modelli considerati e la Naive Bayes (BN) evidenzia differenze significative sia in termini di prestazioni medie sia di stabilità. I modelli ensemble, in particolare Random Forest e Gradient Boosting, mostrano valori medi di accuratezza e F1-score superiori, accompagnati da una minore varianza, segno di maggiore robustezza rispetto alle variazioni nei dati di addestramento. La Naive Bayes, pur risultando più semplice e veloce, presenta metriche inferiori e una maggiore sensibilità alla distribuzione dei dati, mentre il modello lineare (ad es. Logistic Regression) si colloca in una posizione intermedia, con buone prestazioni ma una leggera tendenza a soffrire in scenari con feature non linearmente separabili.

Osservazioni sull'overfitting

L'analisi del gap tra prestazioni in training e test set rivela che il Gradient Boosting, pur ottenendo punteggi molto elevati in addestramento, mostra un leggero calo in test, indice di un potenziale overfitting. La Random Forest mantiene un gap più contenuto, mostrando una migliore capacità di generalizzazione. La Naive Bayes, al contrario, presenta prestazioni simili tra train e test, ma a un livello complessivamente più basso, suggerendo che la sua semplicità riduce il rischio di overfitting ma limita anche la capacità predittiva.

Conclusioni

Il progetto ha mostrato come la Rete Bayesiana, grazie alla sua capacità di gestire l'incertezza e di fare inferenze anche con dati parziali, rappresenti una soluzione efficace e versatile. Per valutarne appieno le potenzialità, è stata messa a confronto con altri modelli come Random Forest, Gradient Boosting e Decision Tree, che hanno confermato buone prestazioni predittive ma senza offrire la stessa flessibilità inferenziale. Questo confronto ha permesso di evidenziare i punti di forza della BN e di capire meglio in quali contesti possa fare la differenza. Per motivi di tempo non è stato possibile approfondire alcune aree di interesse, come l'aggiunta di nuove variabili per arricchire il modello, l'espansione del dataset per aumentarne la robustezza e l'ottimizzazione del tempo di ricerca degli iperparametri per migliorare l'efficienza del processo di addestramento.

Riferimenti Bibliografici

[1] Koller, D., Friedman, N. (2009). Section 3.1–3.4: Representation of Bayesian Networks. In *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

[2] Margaritis, D. (2005). Learning dynamic Bayesian network models via cross-validation. *Pattern Recognition Letters*, 26(14), 2175–2184. https://doi.org/10.1016/j.patrec.2005.03.018

[3] Scutari, M. (2010). Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3), 1–22. https://doi.org/10.18637/jss.v035.i03

[4] Jensen, F.V., Nielsen, T.D. (2007). Chapter 2: Bayesian Networks. In *Bayesian Networks and Decision Graphs* (pp. 15–57). Springer.