Tesi Triennale

Alessandro Cometa

5 dicembre 2024

Indice

Capitolo 1

Introduzione: conseguenze e considerazioni dal caso "Loomis vs Wisconsin"

1.1 I soggetti, l'imputazione e la confutazione

Il giudizio di primo grado In sintesi, Eric L. Loomis nel 2013 viene fermato dalla polizia mentre era alla guida di un'automobile precedentemente usata per una sparatoria nello stato del Wisconsin, USA. Allo stato di fermo, al sig. Loomis vengono imputati vari capi d'accusa, tutti in recidiva, e con questi si va per i vari gradi di giudizio previsti dall'ordinamento giudiziario statunitense. All'attenzione del giudice, viene presentato anche il parere di un'intelligenza artificiale, il software COMPAS (Correctional offender management profiling for alternative sanctions), perché la produzione di un PSI (Presentence Investigation Report) era stata richiesta dalla Corte stessa, dopo che questa aveva recepito dal sig. Loomis l'ammissione di colpevolezza.[1]

Il giudizio di secondo grado L'uso di COMPAS nella determinazione della pena - nel giudizio di primo grado - violava il diritto all'equo processo sotto tre profili, secondo la difesa del sig. Loomis:

1. il diritto ad essere condannato ad una determinata pena sulla base di informazioni accurate, delle quali non si poteva disporre in quanto coperte da diritti di proprietà industriale
2. il diritto di essere condannato ad una pena individualizzata
3. l'uso improprio del dato di genere nella determinazione della pena.

Nessuno dei tre punti viene accettato dall'analogo USA della nostra Corte d'Appello:

1. pur vigendo il segreto industriale sul funzionamento di COMPAS, al sig. Loomis è stata data la possibilità di verificare i dati forniti in input, e di contestare il risultato finale del calcolo - cosa che il sig. Loomis non ha fatto;
2. sul diritto del reo ad una pena individualizzata, questo non è stato violato da COMPAS, perché se anche COMPAS non dovesse tener conto delle peculiarità dell'individuo - cosa di cui non si può sapere, poiché il funzionamento interno è secretato - al giudice è comunque dato di scegliere in totale discrezionalità dati puri o aggregati provenienti anche da fonti diverse;
3.

La sentenza della Corte Suprema La Corte Suprema (che è l'analogo USA della nostra Corte di Cassazione) arriva nel 2016 a sostenere, in una discussa sentenza, quali limiti e cautele devono essere applicati quando un giudice ha a che fare con pareri provenienti da software simili a COMPAS, nonché da COMPAS stesso.

[1] Un racconto molto migliore e molto più dettagliato dei fatti del caso è presente in Giurisprudenza Penale, ad opera di Stefania Carrer[13].

Si è infatti stabilito che tali software possono essere considerati fattori rilevanti in questioni quali

1. la comminazione di misure alternative alla detenzione per gli individui a basso rischio di recidiva;
2. la valutazione della possibilità di controllare un criminale in modo sicuro all'interno della società, anche con l'affidamento in prova;
3. l'imposizione di termini e condizioni per la libertà vigilata, la supervisione e per le eventuali sanzioni alle violazioni delle regole previste dai regimi alternativi alla detenzione.

Ribadendo la necessità che il giudice applichi i risultati COMPAS facendo esercizio della propria discrezionalità sulla base del bilanciamento con altri fattori, **la Corte ha confermato che l'uso dello strumento non può riguardare il grado di severità della pena sulla base di circostanze attenuanti od aggravanti, né la decisione sull'incarcerazione dell'imputato.** Ha specificato infatti che lo scopo di COMPAS è quello di individuare le esigenze del soggetto che deve scontare la pena e di valutare il rischio di reiterazione del reato.[13]

1.2 Il ruolo del software COMPAS: la stima della probabile recidività

COMPAS La sentenza ha suscitato vasto interesse accademico e giornalistico sulla possibilità di un software in generale e di un'intelligenza artificiale in particolare, notoriamente non umani, di interferire nella decisione dosimetrica, tutta umana, del giudice che sta per pronunciarsi su un caso penale. [4] [3] [10] [9] [5]
Difficile non citare Ugo Pagallo, Luciano Floridi e Giovanni Sartori come persone che in Italia si sono spese sul tema più ampio dell'interazione tra uomo e intelligenza artificiale, sia sul particolare piano giuridico, sia in vari altri campi del vivere per come lo conosciamo oggi.
La sentenza arriva anche all'attenzione di autori di un articolo su Science[1], dove si puntualizza... Negli USA, nel frattempo, ProPublica, una testata giornalistica indipendente, scrive una serie di articoli mirati a criticare l'efficacia statistica del software COMPAS.[12]
BLABLABLA

1.3 Il libero convincimento del giudice in Italia

La sentenza della Corte Suprema ha avuto origine, tra le altre cose, dalla supposizione che il parere di COMPAS non fosse stato generato secondo la Costitu-

zione effettivamente statuita[2]. La Corte Suprema, dunque, procede stabilendo che questo parere è legittimo, anche se è utilizzabile solo in alcuni casi e per alcuni scopi. Potrebbe essere interessante esplorare, a questo punto, se nell'ordinamento italiano potrebbe accadere qualcosa del genere: in Italia, una profilazione preliminare come quella offerta dal software COMPAS verrebbe presa in considerazione dal giudice italiano di primo grado? La normativa di riferimento si ritrova al libro primo (dedicato alle disposizioni generali), titolo V ("Dei poteri del giudice") del codice di procedura civile, agli artt. 115 e 116.

Sull'art. 115 c.p.c.

> Salvi i casi previsti dalla legge[3], il giudice deve porre a fondamento della decisione le prove proposte dalle parti o dal pubblico ministero nonché i fatti non specificatamente contestati dalla parte costituita[4].
>
> Il giudice può tuttavia, senza bisogno di prova, porre a fondamento della decisione le nozioni di fatto che rientrano nella comune esperienza[5].

Sentenze di Cassazione rilevanti sono:

- Cassazione civile, Sez. VI, ordinanza n. 3126 del 1 febbraio 2019

  > L'onere di contestazione riguarda le allegazioni delle parti e non le prove assunte, la cui valutazione opera in un momento successivo alla definizione dei fatti controversi ed è rimessa all'apprezzamento del giudice.

  Qui già si apre a quanto disciplinato dall'articolo seguente: la valutazione delle prove e il libero apprezzamento delle stesse prove da parte del giudice.

[2] Citando la formula di Hans Kelsen.

[3] In virtù del principio dispositivo sono le parti a proporre al giudice gli elementi di prova su cui basare il proprio convincimento. E' bene precisare che la legge prevede ipotesi eccezionali, in cui il giudice dispone ex officio mezzi di prova, come nel caso degli artt. 117 (interrogatorio non formale delle parti), 118 (ispezione di persone e di cose), 213 (richiesta di informazioni alla P.A.), 257 (assunzione di nuovi testimoni), 421, 442 (poteri istruttori del giudice in controversie di lavoro e di previdenza e di assistenza obbligatorie), 714 (poteri istruttori nei procedimenti di interdizione o inabilitazione).

[4] L'articolo in esame è stato aggiornato con le modifiche introdotte dalla Legge 18 giugno 2009, n. 69. Con tale nuova formulazione viene conferito al giudice il potere di ritenere provati, accanto ai fatti notori, anche quelli che non sono stati specificamente contestati dalla controparte nè direttamente nè indirettamente. La ratio di tale riforma è dettata dall'esigenza di attuare il disposto normativo di cui agli artt.2697,2698 c.c. che impongono a chi voglia far valere un fatto in giudizio di provarne i fatti che ne sono a fondamento: è il principio dell'onere della prova.

[5] L'ultimo comma rappresenta una deroga al principio dispositivo ed al contraddittorio in quanto introduce nel processo civile prove non fornite dalle parti e relative a fatti dalle stesse non vagliati nè controllati. Tuttavia, la possibilità per il giudice di ricorrere ai c.d. fatti notori sussiste solo ed esclusivamente nel caso in cui si tratti di fatti acquisiti alle conoscenze della collettività, in un dato tempo e luogo, con tale grado di certezza da apparire indubitabile ed incontestabile. Tipico esempio di fatto notorio è la svalutazione monetaria.

Sull'art. 116 c.p.c.

> Il giudice deve valutare le prove secondo il suo prudente apprezzamento[6], salvo che la legge disponga altrimenti[7].
>
> Il giudice può desumere argomenti di prova dalle risposte che le parti gli danno a norma dell'articolo seguente, dal loro rifiuto ingiustificato a consentire le ispezioni che egli ha ordinate e, in generale, dal contegno delle parti stesse nel processo[8].

Torna molto pratico menzionare anche alcune sentenze della Corte di Cassazione:

- Cassazione civile, Sez. I, sentenza n. 17392 del 1 settembre 2015

  > Nell'ordinamento processuale vigente manca una norma di chiusura sulla tassatività tipologica dei mezzi di prova, sicché il giudice può legittimamente porre a base del proprio convincimento anche prove cd. atipiche, quali le dichiarazioni scritte provenienti da terzi, della cui utilizzazione fornisca adeguata motivazione e che siano idonee ad offrire elementi di giudizio sufficienti, non smentiti dal raffronto critico con le altre risultanze istruttorie, senza che ne derivi la violazione del principio di cui all'art. 101 c.p.c., atteso che, sebbene raccolte al di fuori del processo, il contraddittorio si instaura con la produzione in giudizio.

- Cassazione civile, Sez. III, sentenza n. 13229 del 26 giugno 2015

  > Nel vigente ordinamento processuale, improntato al principio del libero convincimento del giudice e in assenza di una norma di chiusura sulla tassatività tipologica dei mezzi di prova, questi può porre a fondamento della decisione anche prove atipiche, non espressamente previste dal codice di rito, della cui utilizzazione fornisca adeguata motivazione e che siano idonee ad offrire elementi di giudizio sufficienti, non smentiti dal raffronto critico con le altre risultanze del processo.

[6] Per "prudente apprezzamento" si intende il compito del giudice tenuto a valutare la attendibilità di ogni circostanza posta alla sua attenzione, ma non necessariamente ad utilizzarla e che può poi anche considerare tutti gli elementi con efficacia probatoria emersi nel corso del giudizio.

[7] Il riferimento è quello alle c.d. prove legali, quali le prove documentali (atto pubblico e scrittura privata autenticata o riconosciuta) o quelle assunte nel processo come la confessione (v. 228 e ss.), il giuramento (v. 233 e ss.) e la testimonianza (v. 244 e ss.). Si tratta di prove la cui efficacia è predeterminata dalla legge e di fronte alle quali al giudice è impedita ogni valutazione sul contenuto della stessa, dovendosi semplicemente attenere alle risultanze della prova offerta, così come legalmente stabilito.

[8] La norma si può anche considerare riferita all'ipotesi in cui il giudice possa valutare, per la formazione del suo convincimento, anche prove formate in un diverso processo. Tali prove possono essere utilizzate dal giudice come semplici indizi idonei a fornire utili e concorrenti elementi di giudizio oppure possono avere valore di prova esclusiva come ad es. accade nel caso di una perizia svolta in sede penale o di una consulenza tecnica svolta in altra sede civile.

- Cassazione civile, Sez. Lavoro, sentenza n. 13054 del 10 giugno 2014

> In tema di procedimento civile, sono riservate al giudice del merito l'interpretazione e la valutazione del materiale probatorio, nonché la scelta delle prove ritenute idonee alla formazione del proprio convincimento, con la conseguenza che è insindacabile, in sede di legittimità, il "peso probatorio" di alcune testimonianze rispetto ad altre, in base al quale il giudice di secondo grado sia pervenuto ad un giudizio logicamente motivato, diverso da quello formulato dal primo giudice.

- Cassazione civile, Sez. VI, sentenza n. 26550 del 12 dicembre 2011

> Il giudice del merito può porre a fondamento della propria decisione una perizia stragiudiziale, anche se contestata dalla controparte, purché fornisca adeguata motivazione di tale sua valutazione, attesa l'esistenza, nel vigente ordinamento, del principio del libero convincimento del giudice.

Quindi, la possibilità che un report proveniente da un software come COMPAS possa essere utilizzato nel processo italiano è rimessa alla sensibilità del giudice sul tema.

Sul giusto processo Si applicano, in ogni caso, le norme sul giusto processo, ex art. 111 della Costituzione.

> La giurisdizione si attua mediante il giusto processo regolato dalla legge.
>
> Ogni processo si svolge nel contraddittorio tra le parti, in condizioni di parità, davanti a giudice terzo e imparziale. La legge ne assicura la ragionevole durata.
>
> Nel processo penale, la legge assicura che la persona accusata di un reato sia, nel più breve tempo possibile, informata riservatamente della natura e dei motivi dell'accusa elevata a suo carico; disponga del tempo e delle condizioni necessari per preparare la sua difesa; abbia la facoltà, davanti al giudice, di interrogare o di far interrogare le persone che rendono dichiarazioni a suo carico, di ottenere la convocazione e l'interrogatorio di persone a sua difesa nelle stesse condizioni dell'accusa e l'acquisizione di ogni altro mezzo di prova a suo favore; sia assistita da un interprete se non comprende o non parla la lingua impiegata nel processo.
>
> Il processo penale è regolato dal principio del contraddittorio nella formazione della prova. La colpevolezza dell'imputato non può essere provata sulla base di dichiarazioni rese da chi, per libera scelta, si è sempre volontariamente sottratto all'interrogatorio da parte dell'imputato o del suo difensore.
>
> La legge regola i casi in cui la formazione della prova non ha luogo in contraddittorio per consenso dell'imputato o per accertata

impossibilità di natura oggettiva o per effetto di provata condotta illecita.

Tutti i provvedimenti giurisdizionali devono essere motivati.

Contro le sentenze e contro i provvedimenti sulla libertà personale, pronunciati dagli organi giurisdizionali ordinari o speciali, è sempre ammesso ricorso in Cassazione per violazione di legge. Si può derogare a tale norma soltanto per le sentenze dei tribunali militari in tempo di guerra.

Contro le decisioni del Consiglio di Stato e della Corte dei conti il ricorso in Cassazione è ammesso per i soli motivi inerenti alla giurisdizione.

Sulla figura dell'amicus curiae A p. 42, Simone Paduanelli riferisce nella sua tesi di laurea magistrale all'Università di Milano-Bicocca [6] che si stanno tentando vari approcci alla tecnologia: sia tramite lavori preparatori di creazione di banche dati pubbliche ministeriali sulle sentenze civili; sia tramite partenariati interuniversitari che, per esempio, radunano le università locali lombarde sotto il progetto "Predictive Justice" (che dovrebbe terminare nel 2026), voluto dal presidente della Corte d'Appello di Brescia, il quale auspica una maggior diffusione della consapevolezza circa il rispetto degli specifici diritti rilevati dalle sentenze pregresse (il "case-law"). Anche l'università Sant'Anna di Pisa si sta attivando, in collaborazione con le corti di Genova e Pisa, per la creazione di una AI che si occupi di apporre annotazioni semantiche alle decisioni giudiziali, per favorire un recupero più rapido delle informazioni: il progetto attualmente sta restituendo dati soddisfacenti negli ambiti del diritto di famiglia e della responsabilità da danno personale.

Nulla di poi così simile ad un software di profilazione come COMPAS, ma dei tentativi di indagare le interazioni tra giurisprudenza e intelligenza artificiale sono in atto. Ciononostante, alcune considerazioni possono essere fatte sull'ingegneria dietro un software come COMPAS - se serve, immaginandocelo come strumento a disposizione del diritto vigente in Italia[9].

[9] Questo diritto, come da art. ¡boh, 80?¿ Cost., estende il diritto vigente al diritto comunitario e al diritto internazionale.

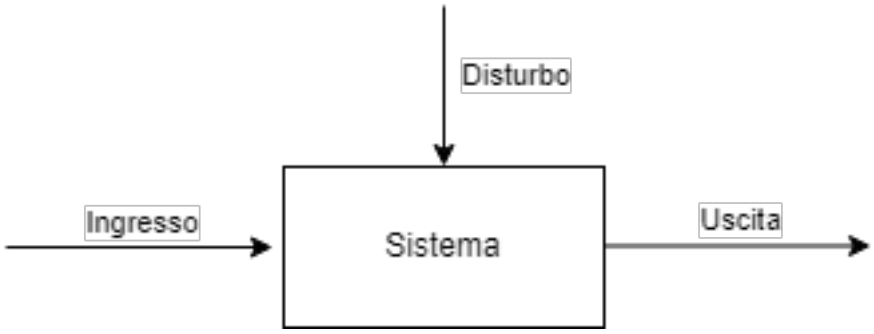
Capitolo 2

Metodologia: come migliorare una AI come COMPAS?

2.1 Struttura di una AI

Per iniziare, una modellazione classica di un qualsiasi sistema è quella della "black-box", in cui si prevedono:

- un ingresso;
- un'elaborazione;
- un eventuale disturbo (anche detto "ingresso non controllabile");
- un'uscita.



Questa concettualizzazione, poiché generalissima, si applica a qualsiasi sistema, e quindi anche a qualsiasi intelligenza artificiale; queste proprietà apparterranno anche a COMPAS.

I dati in ingresso - L'input ProPublica è riuscita ad estrarre dei dati su alcuni casi su cui è stato prodotto il report PSI di cui si è parlato prima[1][7]. Sono state raccolte le 137 domande sulla base delle quali il sistema COMPAS era chiamato a produrre una stima numerica della probabilità che il soggetto si rendesse nuovamente reo.[8]

I dati in uscita - L'output L'output corrisponde ad un punteggio da 0.0 a 10.0, corrispondente alla probabilità stimata di recidività. Punteggi più alti corrispondono ad un reo a più probabile recidività.

I dati nel mentre - L'elaborazione Come le 137 domande portano ad un punteggio? Il materiale di partenza non permette di rispondere a questa domanda precisamente, tuttavia il file `"compas.db"` contiene la tabella "people", che sarà oggetto di indagine di questa tesi. La mancanza di competenze giuridiche adeguate sul processo penale statunitense obbliga a proiettare i dati su una restrizione di colonne.

[1] Vatti a vedere la reference.

2.2 L'importanza dei dati

Sulla normalizzazione delle tabelle Il file che ospita l'input, "compas.db", possiede numerose tabelle. L'attenzione gravita naturalmente, trattandosi di un'elaborazione statistica, sulla tabella "people", la quale è, abbastanza evidentemente, non normalizzata, se non nella cosiddetta "Prima forma normale".

La normalizzazione algebrica dei dati è una sequenza di accorgimenti volti al minimizzare i dati ridondanti, che evitano sia la perdita di dati rilevanti, sia eventuali confusioni nell'accesso ai dati.

Ispezione preliminare Il file che era stato originariamente preso in considerazione, `compas-scores.csv`, presentava colonne duplicate.

```
id,
name,
first,
last,
compas_screening_date,
sex,
dob,
age,
age_cat,
race,
juv_fel_count,
decile_score,
juv_misd_count,
juv_other_count,
priors_count,
days_b_screening_arrest,
c_jail_in,
c_jail_out,
c_case_number,
c_offense_date,
c_arrest_date,
c_days_from_compas,
c_charge_degree,
c_charge_desc,
is_recid,
num_r_cases,
r_case_number,
r_charge_degree,
r_days_from_arrest,
r_offense_date,
r_charge_desc,
r_jail_in,
r_jail_out,
```

```
is_violent_recid,
num_vr_cases,
vr_case_number,
vr_charge_degree,
vr_offense_date,
vr_charge_desc,
v_type_of_assessment,
v_decile_score,
v_score_text,
v_screening_date,
type_of_assessment,
decile_score,
score_text,
screening_date
```

Già ad una prima ispezione, ci si accorge che alcune colonne sono duplicate, infatti la colonna `decile_score` figura due volte. Con un'ispezione dei dati contenuti nel dataset, anche altre colonne risultano contenere, in ordine, gli stessi dati: hanno soltanto nomi diversi, e sono state eliminate in quanto il seguente blocco di codice non ha sollevato alcun AssertionError:

```
def eq_col(data, feat1, feat2):
    for x in range(0, 11757):
        assert data[feat1].iloc[x] == data[feat2].iloc[x]
    print(f"{feat1} coincide con {feat2}")
eq_col(data, "decile_score", "decile_score.1")
eq_col(data, "compas_screening_date", "v_screening_date")
eq_col(data, "compas_screening_date", "screening_date")
```

Sono, evidentemente, figlie di un qualche join infelice.

Sull'algebra relazionale

Le forme normali Qualche definizione informale:

1. Prima forma normale (1NF):

   Si richiede che nessuna riga sia uguale alle altre, e che nessuna colonna contenga dati compositi.

2. Seconda forma normale (2NF):

   In aggiunta a quanto previsto dalla 1NF, si richiede che ciascun attributo non-chiave dipenda sull'intera chiave - ovvero che ogni attributo non-chiave abbia una piena dipendenza funzionale su ciascuna chiave candidata.

   Esempio:

   Inventario = {Prodotto, Magazzino, Prezzo, NomeProdotto, IndirizzoMagazzino};

   La tabella non è in 2NF perché, anche ammesso che sia in 1NF, alcuni attributi non sono funzionalmente dipendenti dall'interezza della chiave primaria.

   La decomposizione della tabella che produce tabelle in 2NF porterà alla seguente situazione:

   Inventario = {Prodotto, Magazzino, Prezzo}

   Prodotti = {Prodotto, NomeProdotto}

   Magazzini = {Magazzino, IndirizzoMagazzino}

3. Terza forma normale:

   Oltre a quanto previsto dalla 2NF, non devono essere presenti dipendenze funzionali transitive tra gli attributi della tabella, ovvero gli attributi devono tutti dipendere dalla chiave primaria della tabella.

   Esempio:

   ESEMPIO DI 3NF

Per "chiave" si intende, come nell'informatica delle basi di dati, quell'insieme di informazioni che identificano completamente l'elemento presente nella base di dati. Tramite questo recupero dell'oggetto, quindi, è possibile conoscerne anche le altre informazioni.

Per quanto riguarda la tabella esposta dal file `compas-scores.csv`, la tabella non è nemmeno in 1NF, perché i dati contenuti nella colonna `"name"` non sono atomici.

Sul preprocessing dei dati

Gestione dei dati non numerici Alcuni dati non sono in forma numerica, mentre ciascuno dei modelli prende in ingresso solo dati numerici, pertanto è stato necessario individuare strategie per gestirli. Il primo problema è stato sui nomi dei condannati: risolto esportando quei dati in una tabella a parte, con l'id a fungere sia da chiave primaria che da chiave esterna. Dopodiché, le date: tutte convertite nella differenza tra la data in questione e una data di riferimento. Per la data di riferimento, tre opzioni:

- La scelta convenzionale, il 01/01/1970, ovvero il cosiddetto "Unix time".
- La scelta più logica, la data più antica presente nel dataset: 14/10/1919, che garantirebbe l'assenza di valori negativi.
- La scelta effettivamente selezionata: anche per mostrare che una data vale l'altra, la data selezionata è il 24/6/1936. E' sufficientemente buona, solo 5 date nell'intero dataset sono anteriori a questa.

Infine, un altro problema è quello delle colonne che ospitano dati categorici. Si è proceduto con una tecnica, chiamata One-hot-encoding, che genera una colonna per ciascuna categoria, assegnando 1 nella nuova cella appartenente alla riga dove il dato originario corrisponde alla colonna della cella, e 0 alle celle della riga in tutte le altre nuove colonne. La colonna "race" è stata trattata così. Ad una più attenta analisi, la colonna "sex" contiene solo dati "Male" e "Female" (è stato verificato), pertanto è stata trattenuta solo la colonna "is_male": la colonna "is_female" sarebbe stata ridondante, in quanto calcolabile all'occorrenza negando il dato di fatto booleano della colonna "is_male".

Ora, ciascuna di queste colonne presenta dei massimi e dei minimi che variano da una colonna all'altra. Questo non è positivo per i modelli, perché span ($= x_{max}–x_{min}$) più elevati faranno credere al modello che un dato sia più importante di altri, mentre magari è solo un risultato indesiderato delle codifiche di cui sopra. Pertanto, tre tecniche di riscalatura:

- Standard scaling: si calcola, per ogni dato in input, un punteggio

$$z = \frac{x–\mu}{\sigma} \tag{2.1}$$

  Questa tecnica ha il compito di rendere unitaria la varianza, e nulla la media, della collezione di dati che gli si dà in input.

- Minmax scaling

$$z = \frac{x–min}{max–min} \tag{2.2}$$

  Questa tecnica garantisce che tutti i dati siano mappati nell'intervallo $[0, 1]$.

- Quantile transformation: una funzione di distribuzione cumulativa mappa i valori originali ad una distribuzione uniforme. Questa tecnica, poiché distribuisce in maniera uniforme i dati, è particolarmente efficiente nella reiezione dei disturbi provocati da dati isolati (outliers).

2.3 Possibili fonti dell'errore e dell'affidabilità di una AI

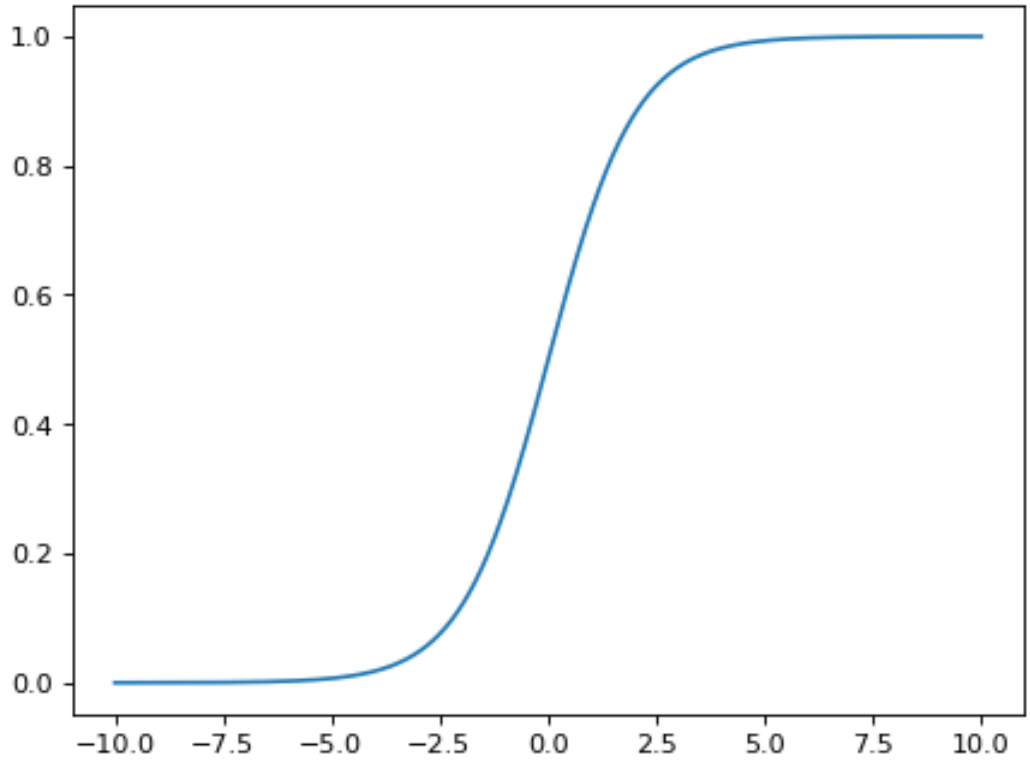
Sistema, modello, elaborazione

La pipeline La libreria selezionata è scikit-learn (versione 1.5.2). Scikit-learn espone vari strumenti per l'analisi dei dati, tra cui:

1. preprocessing.StandardScaler
2. preprocessing.MinMaxScaler
3. preprocessing.QuantileTransformer
4. preprocessing.scale()
5. preprocessing.minmax_scale()
6. preprocessing.quantile_transform()
7. linear_model.LogisticRegression
8. neighbors.KNearestNeighbors
9. ensemble.RandomForestRegressor
10. svm.SVR
11. model_selection.train_test_split()
12. pipeline.make_pipeline()
13. decomposition.PCA
14. metrics.r2_score()
15. metrics.mean_squared_error()
16. metrics.make_scorer()
17. tree.plot_tree()

Le prime 6 risorse sono state spiegate nel paragrafo precedente, dedicato al preprocessing dei dati. Ora ci si concentra sul resto, ovvero sui modelli resi disponibili dalla libreria. La regressione logistica è un processo di apprendimento automatico per cui i dati in input vengono scrutinati da una funzione detta logistica, o sigmoide:

$$f(x) = \frac{1}{1 + e^x} \tag{2.3}$$



Come da foto, la funzione è monotona crescente, e ha valori nell'intervallo $[0, 1]$. In particolare, i valori estremali 0 e 1 sono asintoti per la funzione. Output ragionevolmente vicini a questi due valori, in base a delle soglie di tolleranza predeterminate, permettono di effettuare una classificazione o una regressione. La funzione è interessante per il fatto di essere derivabile.

$$f'(x) = f(x)(1 - f(x)) \tag{2.4}$$

L'errore: patologia dell'apprendimento automatico ma non solo

Teorema 1 (L'errore non è mai del tutto eliminabile)

$$
\begin{aligned}
MSE(X,y,D) &= E_{X,y,D}[[h_D(X)-y]^2] \\
&= E_{X,y,D}[[h_D(X)+(\overline{h}_D(X)-\overline{h}_D(X))-y]^2] \\
&= E_{X,y,D}[[(h_D(X)-\overline{h}_D(X))+(\overline{h}_D(X))-y)]^2] \\
&= E_{X,y,D}[[(h_D(X)-\overline{h}_D(X))^2+(\overline{h}_D(X))-y)^2 \\
&\qquad + 2(h_D(X)-\overline{h}_D(X))(\overline{h}_D(X))-y)]] \\
&= E_{X,D}[(h_D(X)-\overline{h}_D(X))^2] \\
&\quad +E_{X,y}[(\overline{h}_D(X))-y)^2] \\
&+2E_{X,y,D}[(h_D(X)-\overline{h}_D(X))(\overline{h}_D(X))-y)]
\end{aligned}
\tag{2.5}
$$

Addendo #3 in (2.5): nullo.

$$
\begin{aligned}
&E_{X,y,D}[(h_D(X)-\overline{h}_D(X))(\overline{h}_D(X))-y)] \\
&= E_{X,y}[E_D[h_D(X)-\overline{h}(X)](\overline{h}(X)-y)] \\
&= E_{X,y}[(E_D[h_D(X)]-\overline{h}(X))(\overline{h}(X)-y)] \\
&= E_{X,y}[(\overline{h}(X)-\overline{h}(X))(\overline{h}(X)-y)] \\
&= E_{X,y}[0*(\overline{h}(X)-y)] \\
&= E_{X,y}[0]=0
\end{aligned}
\tag{2.6}
$$

Addendo #3 in (2.5): scomponibile.

$$
\begin{aligned}
&E_{X,y}[(\overline{h}_D(X))-y)^2] \\
&= E_{X,y}[(\overline{h}(X)-\overline{y}(X)+\overline{y}(X)-y))^2] \\
&= E_{X,y}[(\overline{y}(X)-y)^2] \\
&\quad +E_X[(\overline{h}(X)-\overline{y}(X))^2] \\
&+2E_{X,y}[(\overline{h}(X)-\overline{y}(X))(\overline{y}(X)-y)]
\end{aligned}
\tag{2.7}
$$

Addendo #3 nell'eq. precedente: nullo.

$$
\begin{aligned}
&E_{X,y}[(\overline{h}(X)-\overline{y}(X))(\overline{y}(X)-y)] \\
&= E_X[E_{y|X}[\overline{y}(X)-y](\overline{h}(X)-\overline{y}(X))] \\
&= E_X[(\overline{y}(X)-E_{y|X}[y])(\overline{h}(X)-\overline{y}(X))] \\
&= E_X[(\overline{y}(X)-\overline{y}(X))(\overline{h}(X)-\overline{y}(X))] \\
&= E_X[0]=0
\end{aligned}
\tag{2.8}
$$

$$
\begin{aligned}
&MSE(X,y,D) = E_{X,y,D}[[h_D(X)-y]^2] \\
&= E_{X,D}[(h_D(X)-\overline{h}_D(X))^2] \\
&+ E_X[(\overline{h}(X)-\overline{y}(X))^2] \\
&+ E_{X,y}[(\overline{y}(X)-y)^2]
\end{aligned}
\tag{2.9}
$$

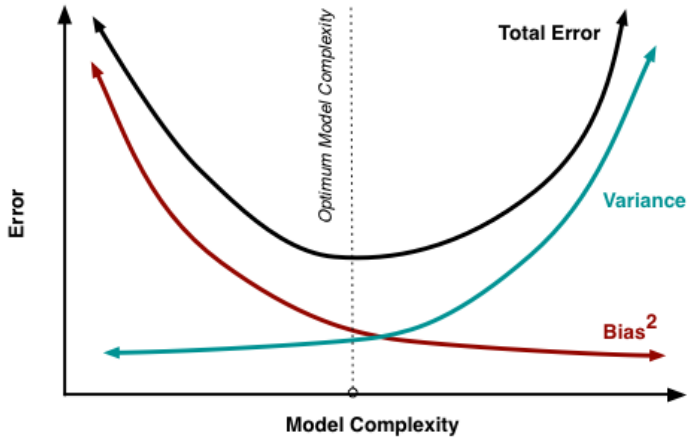


Figura 2.1: da [14]

Bias-variance tradeoff *Quindi, l'errore quadratico medio è scomponibile in tre contributi additivi:*

- *La varianza (o errore casuale):* $E_{X,D}[(h_D(X) - \overline{h}_D(X))^2]$*;*
- *Il quadrato del bias (o errore sistematico):* $E_X[(\overline{h}(X) - \overline{y}(X))^2]$*;*
- *Il rumore (ineliminabile) dei dati:* $E_{X,y}[(\overline{y}(X) - y)^2]$*;*

La qualità di un modello è valutabile dalla sua capacità di minimizzare l'errore complessivo. Un primo problema risiede nel fatto che, man mano che cresce la complessità del modello, benché cali l'errore sistematico, cresce l'errore casuale. Il modello, quindi, benché interessato programmaticamente a minimizzare l'errore, non potrà mai annullarlo.

Overfitting e underfitting

- Il fenomeno dell'overfitting si manifesta quando il sistema intelligente non riesce a generalizzare, e quindi ad apprendere alcunché dai dati. Ciò implica che crea un modello troppo più complesso e rigido del necessario sui dati presenti nel training set, talmente rigido da non comportarsi correttamente coi dati nel test set.
- Il fenomeno dell'underfitting, dualmente, coincide con modelli semplicistici, altrettanto inadeguati, con la conseguenza che il modello espone errore anche nei dati del training set, diminuendo la varianza in favore dell'errore sistematico (e del suo quadrato).

Altri indici di affidabilità

Matrici di confusione La matrice di confusione, detto di un sistema di classificazione binaria (esempi: vero/falso, alto/basso, spam/non-spam) è una matrice con due righe (positivo attuale e negativo attuale) e due colonne (positivo predetto e negativo predetto). Ciascuna delle celle contiene il conteggio dei risultati:

- davvero positivi (TP);
- davvero negativi (TN);
- erroneamente rilevati come positivi (FP);
- erroneamente rilevati come negativi (FN).

Precision e recall La bontà di un modello emerge, ad un rapido calcolo, da alcuni indici di semplice computazione, tra cui "precision" e "recall".

$$Precision = TP/(TP + FP) \tag{2.10}$$

$$Recall = TPR = TP/(TP + FN) \tag{2.11}$$

Più sono alti i valori di questi indici, migliore è il modello:

- "precision" si occupa di verificare quante, tra le predizioni positive, sono corrette;
- "recall" tiene traccia di quanti, tra i veri positivi, sono stati predetti correttamente.

Curva ROC Una rappresentazione più interessante, specie nell'ottica di comparare più predittori, richiede di considerare un ulteriore indice:

$$FPR = FP/(TN + FP) \tag{2.12}$$

La curva ROC (Receiver Operating Characteristic, o anche Relative Operating Characteristic) esiste in uno spazio cartesiano dove su un asse è rappresentato il TPR, e sull'altro il FPR.

Un predittore binario è tanto più efficace, quanto più la sua caratteristica operativa si distanzia da quella di un decisore casuale, che avrebbe invece un tasso di veri positivi identica al tasso di falsi positivi.

2.4 Funzionamento di una AI: i modelli

La rete neurale: il ruolo della funzione di attivazione La funzione di attivazione $\phi(x)$ è, data una rete neurale, quella funzione che trasforma tutti gli input di un percettrone in un dato numerico di output. Il percettrone è ispirato al neurone, componente elementare del cervello umano, che riceve più segnali elettrochimici dai collegamenti sinaptici con altri neuroni. Si è detto prima che, per un funzionamento ottimale del modello, l'errore è necessario, e per evitare un errore casuale troppo alto (che pregiudicherebbe la ripetibilità dei risultati), viene preso in considerazione un errore sistematico fisso adeguato, il "bias".

$$f(x) = b + w_1x_1 + w_2x_2 + ... + w_nx_n \tag{2.13}$$

$$g(x) = \phi(f(x)) \tag{2.14}$$

Funzioni di attivazione alternative: descrizione analitica Tra le funzioni di attivazione $\phi(x)$ che si possono utilizzare si segnalano:

- **Sigmoide** (che è stata già spiegata)
- **ReLU**
  La Rectified Linear Unit ha una espressione molto semplice:

  $$\phi(x) = max(0, x)$$

  Non derivabile in 0, altrove espone una dinamica (e, quindi, una derivata) ancora più semplice: è il gradino di Heaviside. La non derivabilità in 0 è un problema che si tenta di aggirare scegliendo una funzione concettualmente simile...
- ...**la funzione softplus**,

  $$\phi(x) = ln(1 + e^x)$$

  la cui derivata esiste ed è pari alla funzione logistica.

La regressione logistica: la rete neurale minima Dato un primo strato di percettroni, che si occupano meramente dell'acquisizione dei dati, tutti connessi ad strato mono-percettrone che emette l'output, se la funzione di attivazione del percettrone di output è l'ormai famosa sigmoide, il compito svolto da questa rete neurale minima nient'altro è che la regressione o classificazione logistica.

K-nearest neighbors: sulle metriche di prossimità Un approccio alternativo all'interpretazione di dati multidimensionali si avvale delle nozioni di metrica di prossimità tra punti immersi in $\mathbb{R}^n$. Come descritto dal famoso eserciziario Marcellini-Sbordone [2]:

> "Sia X un insieme e $d : XxX \to [0, +\inf)$ una funzione. Si dice che d è una distanza o metrica su X, se sono verificate le condizioni:

1. $d(x, y) = 0 \iff x = y$
2. $d(x, y) = d(y, x) \forall x, y \in X$ — disuguaglianza triangolare
3. $d(x, y) \leq d(x, z) + d(z, y) \forall x, y, z \in X$

Se d è una distanza su X si dice che (X, d) è uno spazio metrico [...]."

Le metriche di prossimità più comuni

- Distanza di Minkowski: famiglia di distanze continue

$$d(x, y) = \left(\sum_{i=1}^{n} |y_i - x_i|^p \right)^{\frac{1}{p}} \tag{2.15}$$

Da notare che, per valori particolari di p, si ottengono molte delle distanze più famose:

1. p=2: la distanza euclidea

$$d = \left(\sum_{i=1}^{n} |y_i - x_i|^2 \right)^{\frac{1}{2}} = \sqrt{(y - x)^2} \tag{2.16}$$

2. p=1: la distanza in modulo assoluto:

$$d = \sum_{i=1}^{n} |y_i - x_i| \tag{2.17}$$

3. $p \to \infty$: la distanza di Chebyshev

$$d = \lim_{p \to \inf} \left(\sum_{i=1}^{n} |y_i - x_i|^p \right)^{\frac{1}{p}} = \max_i \{|y_i - x_i|\} \tag{2.18}$$

- Distanze discrete Esistono varie nozioni discrete di distanza.

1. Distanza discreta - per antonomasia

$$d(x, y) = (x == y) \tag{2.19}$$

2. Distanza di Hamming
Su un insieme $\mathbf{F^n}$ di stringhe di lunghezza n su un alfabeto **F**, d(x, y) = cardinalità({ i : $x_i \neq y_i$)

Random forest: il modello esplicabile Foresta casuale, dall'inglese, di alberi. L'albero decisionale, in particolare, ovvero una rete di selettori che categorizzano in maniera via via più raffinata i dati di allenamento in ingresso, fino a degli insiemi con una cardinalità minima preimpostata chiamati "foglie". Dopodiché, tocca ai dati di test, per cui ogni albero compirà un errore minimo. La tecnica della foresta casuale è un modello di apprendimento ensemble, poiché inizializza molteplici alberi, tra i quali è possibile riconoscere l'albero migliore (cioè, quello che performa meglio, col minimo errore). La tecnica è di interesse nell'ambito delle intelligenze artificiali spiegabili, ovvero di cui è facile comprendere il processo matematico che mappa gli input negli output. Infatti, basterà ripercorrere il percorso decisionale (la sequenza ottimale di decisioni binarie) per capire la categorizzazione compiuta dal modello. Come con tutti gli alberi, alla struttura può essere richiesto di arrestarsi a varie profondità.2.2

Support Vector Machines: il metodo kernelizzato L'idea di base è l'identificazione di un'unico iperpiano nell'iperspazio in cui sono immersi i dati, dal quale è massima la distanza dei punti più prossimi. La ricerca è del più robusto vincolo monolatero (scleronomo, ovviamente) che separi linearmente (in uno spazio eventualmente aumentato di dimensioni extra, se non fossero nativamente linearmente separabili) la collezione di dati che gli arrivano in input nel modo più esatto possibile.

I kernel I tre kernel più famosi sono:

- Il kernel lineare
$$K(x, y) = x^T y \tag{2.20}$$
- Il kernel polinomiale
$$K(x, y) = (x^T y + r)^n \tag{2.21}$$
- Il kernel gaussiano (o radiale, o RBF)
$$K(x, y) = e^{-\frac{|x-y|^2}{2\sigma^2}} \tag{2.22}$$

Ma perché il kernel è così importante per questi modelli? Il kernel genera uno spazio implicito in cui non è necessario calcolare esplicitamente nuovi parametri e nuove coordinate. L'approccio, infatti, è quello:

1. di associare l'i-esimo record dei dati di allenamento (x_i, y_i), imparandone un peso scalare w_i
2. e di applicare ai dati di verifica x' una funzione di similarità k (appunto, il kernel)

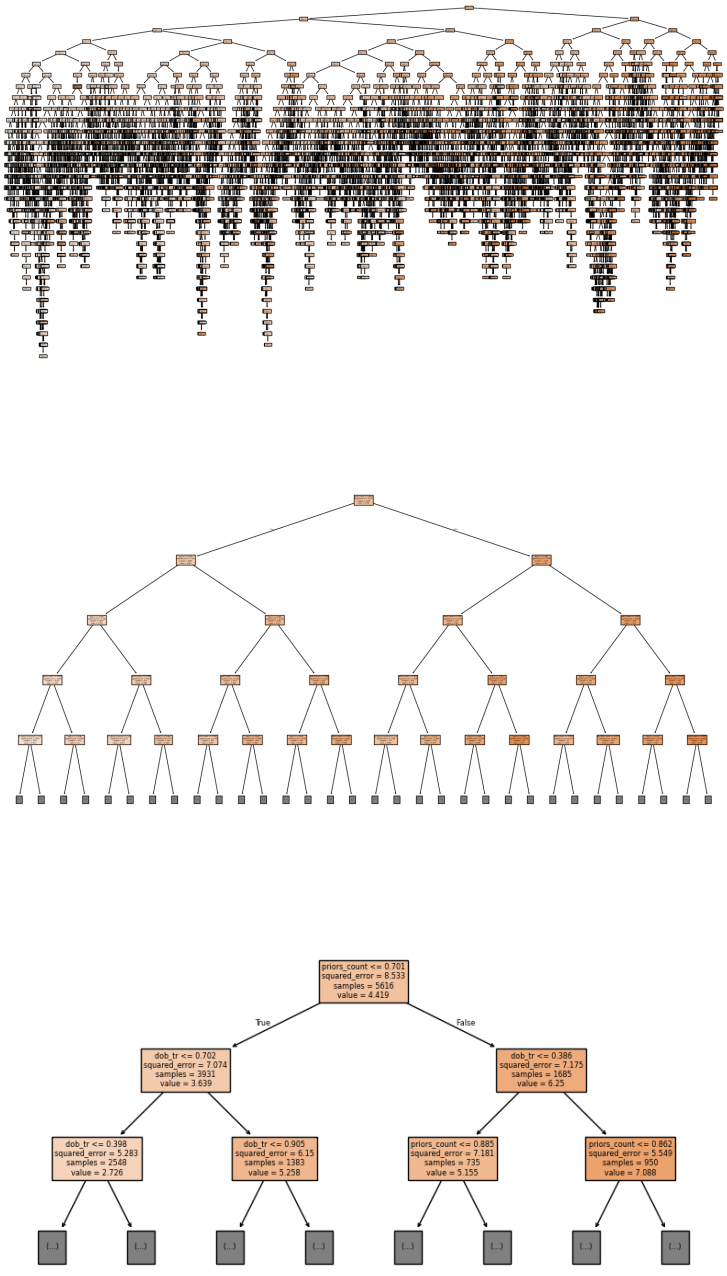


Figura 2.2: L'albero migliore della Foresta, generata con i dati di COMPAS, poi potato a varie profondita (depth=4, depth=2)

2.5 Le pipelines

Vista la nota best practice del preprocessing, gli sviluppatori della libreria scikit-learn hanno previsto la necessità di associare, in un unico processo, sia il preprocessore dei dati che il modello che si nutre direttamente dell'output del preprocessore.

2.6 Le metriche di performance

Poiché ciascun oggetto istanziato da una classe della libreria scikit-learn espone un metodo .fit() e un metodo .predict(), le pipelines, essendo di classe Pipeline, non fanno eccezione. Delle varie metriche, la metodologia selezionata per questa tesi prevede di comparare varie pipelines, composte ciascuna da un preprocessore seguito da un modello tra quelli esaminati, sulla base del mean_squared_error (MSE) summenzionato. Si mostrano anche i risultati del R2_score (cioè, il coefficiente di determinazione: $R^2 = 1 - \frac{RSS}{TSS}$, dove RSS sta per la somma residua dei quadrati, mentre TSS è la somma totale dei quadrati). R2 segnala, se è prossimo a 1, il buon comportamento predittivo del regressore, ma la sua espressività è limitata alle regressioni lineari. Le regressioni tra cui si vuole esplorare non lo sono, il che apre a R2_scores negativi, che quindi non sono significativi.

2.7 Un'ultima indagine sui dati: la PCA

La PCA, Principal Component Analysis, è una tecnica che permette di individuare gli autovettori della matrice di covarianza, o in altre parole quale vettore di pesi nella combinazione lineare dei dati spiega la maggior porzione di varianza possibile. Invece che prendere i dati così come sono, con la PCA è possibile generare un nuovo spazio, a dimensionalità ridotta, i cui dati sono calcolati con la combinando linearmente i dati usando come pesi le componenti dell'autovettore selezionato. Autovettore, della matrice delle covarianze.

Capitolo 3

Esempi: casi d'uso su restrizioni del dataset

3.1 Prendere meno colonne

Per l'elaborazione, è stato preso un sottoinsieme delle colonne presenti nella tabella `"people"` nel file `"compas.db"`.

Capitolo 4

Conclusioni: come cambiano le metriche al variare della tecnica?

4.1 Presentazione dei risultati

Model	Pre-scaler	mse_score	r2_score
logistic	standard	8.7636	-0.0945
logistic	minmax	8.9271	-0.1149
logistic	quantile	7.4837	0.0654
svr	standard	18.4113	-1.2994
svr	minmax	6.5447	0.1826
svr	quantile	4.6640	0.4175
knn	standard	6.4966	0.1886
knn	minmax	7.0803	0.1157
knn	quantile	5.0062	0.3748
random_forest	standard	6.0295	0.2470
random_forest	minmax	7.3607	0.0807
random_forest	quantile	5.1890	0.3519
voting	standard	5.5647	0.3050
voting	minmax	6.9249	0.1351
voting	quantile	4.8070	0.3996

Figura 4.1: Come già spiegato, mse_score coincide con una nozione classica di errore, pertanto di queste 15 tecniche di analisi dei dati risulta interessante cercare quella per cui è stato registrato il minimo errore quadratico medio. Questo primato sembra spettare alla pipeline costituita dal preprocessore che trasforma tramite quantili, fatto seguire dalla Support Vector Machine (svr, perché il suo compito è di regressione). Presento comunque r2_score, sapendo che non si applica per predittori non lineari.

Le features più rilevanti Ciascuna delle tre tabelle in figura 4.2 deriva da una PCA cui sono stati dati in input i dati preprocessati tramite una delle tre tecniche viste in precedenza. La terza tabella mostra risultati di qualità significativamente più bassa, visto che le prime cinque componenti spiegano, in somma, a malapena il 55% della varianza complessiva dei dati in input. Va meglio, ma non molto meglio, con le altre due PCA:

- Il quantile transformer spiega il 78%
- Il minmax scaler spiega il 95% (praticamente, tutto)

Ma il dato più rilevante, è sull'importanza delle features. Sempre restando sulle prime due tabelle, si evidenzia che la componente 1, che spiega rispettivamente il 30 e il 39 % della varianza, è importante per oltre il 70% essere afro-americani. E, per oltre il 60%, non essere caucasici. La recidività è importante solo ad un occhio più attento che si spinga alla seconda componente principale, che però è seconda proprio perché spiega una percentuale inferiore della varianza dei dati (il metodo restituisce le componenti in ordine decrescente). Il metodo rileva correttamente anche un altro fatto, molto meno grave: una correlazione tra la feature "dob_tr" e la feature "age", come prevedibile dalla semantica di queste features.

Quantile_transform

Feature	Component 1	Component 2	Component 3	Component 4	Component 5
Explained variance ratio	0.3038	0.1665	0.1241	0.1089	0.0928
age	-0.1321	-0.1150	-0.5275	-0.2959	0.1905
juv_fel_count	0.0453	0.0526	-0.0180	0.0192	-0.0363
juv_misd_count	0.0578	0.0843	0.0120	0.0198	-0.0661
juv_other_count	0.0476	0.1083	0.0778	0.0818	-0.0978
priors_count	0.1392	0.2584	-0.3885	-0.2017	-0.1010
is_male	0.0659	0.2166	-0.5228	0.7642	-0.2318
dob_tr	0.1322	0.1152	0.5276	0.2961	-0.1900
is_hispanic	-0.0441	-0.0566	0.0213	0.2716	0.6024
is_asian	-0.0020	-0.0032	-0.0015	0.0059	0.0086
is_african-american	0.7018	-0.1625	-0.0705	-0.2151	-0.3468
is_other	-0.0273	-0.0396	0.0168	0.0963	0.2131
is_caucasian	-0.6275	0.2621	0.0316	-0.1599	-0.4840
is_native american	-0.0009	-0.0002	0.0023	0.0011	0.0067
is_recid	0.2116	0.8593	0.0766	-0.1943	0.2923

Minmax_scale

Feature	Component 1	Component 2	Component 3	Component 4	Component 5
Explained variance ratio	0.3919	0.2077	0.1508	0.1238	0.0635
age	-0.0521	-0.0404	0.0321	-0.0229	-0.0007
juv_fel_count	0.0031	0.0035	0.0018	-0.0009	0.0004
juv_misd_count	0.0054	0.0066	0.0006	-0.0022	-0.0003
juv_other_count	0.0026	0.0060	0.0017	-0.0018	-0.0022
priors_count	0.0372	0.0506	0.0132	-0.0324	-0.0161
is_male	0.0569	0.2223	0.9449	-0.2286	0.0057
dob_tr	0.0518	0.0403	-0.0318	0.0229	0.0006
is_hispanic	-0.0407	-0.0502	0.1804	0.6706	-0.5110
is_asian	-0.0016	-0.0025	0.0055	0.0085	0.0090
is_african-american	0.7246	-0.0944	-0.1225	-0.4219	-0.1602
is_other	-0.0239	-0.0326	0.0603	0.2284	0.8266
is_caucasian	-0.6575	0.1801	-0.1229	-0.4935	-0.1708
is_native american	-0.0010	-0.0005	-0.0008	0.0079	0.0065
is_recid	0.1742	0.9485	-0.1967	0.1485	0.0173

Scale

Feature	Component 1	Component 2	Component 3	Component 4	Component 5
Explained variance ratio	0.1720	0.1254	0.1008	0.0848	0.0768
age	0.5211	0.4157	-0.0559	0.0384	0.0158
juv_fel_count	-0.1384	0.1813	0.1881	0.1116	0.0847
juv_misd_count	-0.2135	0.1941	0.4154	0.1507	0.1118
juv_other_count	-0.2086	0.0700	0.4289	0.1509	0.0548
priors_count	-0.1022	0.5278	0.2173	0.0126	-0.0124
is_male	-0.0509	0.1895	0.1241	0.3011	-0.0384
dob_tr	-0.5211	-0.4157	0.0559	-0.0381	-0.0159
is_hispanic	0.0597	-0.1354	-0.0880	0.7326	-0.4856
is_asian	0.0266	-0.0165	-0.0367	0.0828	0.0369
is_african-american	-0.3936	0.3769	-0.4169	-0.2667	-0.0517
is_other	0.0535	-0.1093	-0.0960	0.3350	0.8503
is_caucasian	0.3482	-0.2546	0.5446	-0.3461	-0.0773
is_native american	-0.0009	-0.0294	0.0047	0.0237	0.0359
is_recid	-0.2289	0.1622	0.2245	-0.0325	-0.0715

Figura 4.2: PCA con i tre preprocessori

4.2 Paragonare il bias statistico di COMPAS al bias sociale

Anche prima dell'avvento delle intelligenze artificiali, la sociologia della devianza si interroga da secoli su come capire chi compie devianza e perché. Non tutta la devianza è reato, ma tutto il reato è devianza. Nella storia del pensiero sociologico si annoverano 6 teorie principali:

- Teoria biologica dell'atavismo criminale (Cesare Lombroso, tardo Ottocento) Uno dei primi studiosi che ha dato una scientificità al desiderio di predire il criminale è stato il medico e psichiatra Cesare Lombroso (1835-1909). Sotto la spinta dell'allora recente teoria darwiniana dell'evoluzione, Lombroso ha ritenuto che il criminale ne fosse rimasto indietro, e ha cercato le prove analizzando le caratteristiche fisiche e biologiche dei criminali, concentrandosi soprattutto sulla forma cranica. Ad ogni modo, la teoria è stata ripetutamente oggetto di critiche, fino a decadere. Esiste il museo di Antropologia criminale, a Torino, dedicato a lui e alla sua teoria. Gli allestitori del museo hanno dichiarato che alla base del museo esiste:

  > una funzione educativa intesa a mostrare come la costruzione della conoscenza scientifica sia un processo che avanza grazie alla dimostrazione non tanto di verità, quanto della "falsificabilità" di dati e teorie che non resistono a una critica.
  >
  > Il nuovo allestimento vuole fornire al visitatore gli strumenti concettuali per comprendere come e perché questo personaggio così controverso formulò la teoria dell'atavismo criminale e quali furono gli errori di metodo scientifico che lo portarono a fondare una scienza poi risultata errata.[11]

- Teoria della tensione (Robert K. Merton, anni '60) Merton (1910-2003), sociologo, scompone la cultura in mete culturali (ovvero, le prosepttive sociali cui tendere, "gli scopi nella vita") e in mezzi istituzionalizzati (i processi sociali autorizzati dalla società e presenti nella società). Merton reinterpreta l'anomia di Émile Durkheim (1858-1917) come tensione (o distanza) tra la struttura sociale istituzionale e la struttura culturale della società. Dunque, individua 5 modi di adattamento alla società:

  1. Conformità : accettare entrambi gli aspetti, e dunque non deviare.
  2. Innovazione: accettare le mete, ma rifiutare i mezzi. Per esempio si vuole diventare più ricchi, ma magari lavorando in modi innovativi o cambiando carriera (eventualmente, passando ad una carriera criminale).
  3. Ritualismo : rifiutare le mete, ma accettando i mezzi. Per esempio, lavorare, ma senza l'ambizione comune di diventare più ricco, e quindi fare appena il necessario.

4. Rinuncia : abbandonare entrambi gli aspetti, e quindi ritirarsi dalla vita sociale in toto (aprendo quindi a fenomeni di marginalità sociale: clochardage, eremitismo - anche urbano, ecc.).
5. Ribellione : rifiutare, ma protestare per la sostituzione di uno o entrambi i valori della società con nuovi.

- Teoria della subcultura (dai critici di Merton) I vari critici della teoria della tensione ritenevano che non fosse sufficiente a spiegare il comportamento deviante, poiché lo ascriveva ad una mera intenzionalità individuale. Invece, soprattutto i membri più celebri della Scuola di Chicago (fondata da Robert Park (1864-1944)), Clifford Shaw e Henry McKay, hanno condotto mastodontiche ricerche sociologiche alla ricerca delle subculture dove si apprendono sia le competenze tecniche, sia le razionalizzazioni che favoriscono l'insorgenza di attività criminali. L'esempio scolastico è quello dello scasso: servono delle competenze specifiche per lo scasso, nonché dei valori più solidi di quelli che legano l'individuo alla società perché questo individuo inizi a delinquere; non è una cosa innata (Lombroso) o una "invenzione dell'attore" (Merton) ma, in soldoni, secondo questi studiosi la strada è la scuola dei devianti. Uno dei maggiori criminologi americani del Novecento, Edwin Sutherland (1883-1950), riprende questa teoria, e la espande notando, tra le altre cose, che chi sta deviando non sta veramente deviando, se si analizza l'individuo in relazione alla subcultura in cui è immerso: deviante è il gruppo rispetto alla società generale, e non l'individuo, essendo che i valori della subcultura si frappongono tra l'individuo e la società generale.

- Teoria del controllo sociale (Travis Hirschi) La teoria della tensione di Merton summenzionata, si basa sull'assunto che l'individuo sia naturalmente portato a seguire mete e mezzi socialmente determinati prima, e accettati poi. La teoria del controllo sociale muove le sue premesse da visione più pessimistica dell'individuo, moralmente molle/debole/flessibile. Pertanto, la prospettiva viene ribaltata: non ci si chiede più perché l'individuo compie reati, bensì come mai non li compia. La risposta è "perché sono frenate dal farlo" dal controllo sociale, cioè dalla sensibilità individuale:
  - al controllo esterno (cioè alla sorveglianza degli altri individui - siano essi rappresentanti di istituzioni pubbliche oppure membri del "gruppo dei pari", cioè del gruppo di amici - per la prevenzione dei comportamenti devianti)
  - al controllo interno diretto (cioè ai sentimenti di imbarazzo, colpa e vergogna interni al trasgressore)
  - al controllo interno indiretto (ovvero all'attaccamento psicologico ed emotivo agli altri, nonché al desiderio di non perdere la loro stima e il loro affetto)

  Il reo, dunque, secondo il sociologo Hirschi (1935-2017), massimo esponente di questa teoria, una persona compie reato quando i vincoli sociali

sono troppo deboli e cadono. Diremmo noi, "quando non ha più nulla da perdere".

- La teoria della scelta razionale (Hiroshi Tsutomi) Le teorie precedenti non prendono in nessuna considerazione l'intenzionalità del reo di violare le norme. Il criminologo giapponese Hiroshi Tsutomi rileva che le persone commettono reati non perché sono patologiche o malvagie, ma perché sono normali. Le motivazioni sono sempre quelle: la ricerca di potere, prestigio, profitto, piacere. Altri sostenitori della teoria sono Cesare Beccaria (tra le tante altre cose, l'autore del trattato "Dei delitti e delle pene" contro la pena di morte e la tortura) e Jeremy Bentham (uno dei più celebri filosofi utilitaristi). Infatti la teoria ha un chiaro sapore economico (sembra ricalcare abbastanza fedelmente le idee del "voto del portafogli" e del libero convincimento del consumatore in un mercato), ed è ripresa non solo da sociologi anche contemporanei, ma anche da economisti. Viene, appunto, individuata una classificazione dei costi del delinquere:
  - Costi esterni pubblici (le sanzioni legali comminate dallo Stato e le sanzioni sociali sulla reputazione)
  - Costi esterni privati (o "costi di attaccamento": le sanzioni informali dagli "altri significativi")
  - Costi interni (emergenti dalla coscienza, o dalle "norme interiorizzate", principalmente sotto forma di senso di colpa e vergogna)
- Teoria dell'etichettamento (dall'interazionismo simbolico)
  Premesse:
  - William Thomas: "Se gli uomini definiscono reali certe situazioni, esse saranno reali nelle sue conseguenze." (Teorema di Thomas; esempio: banca che viene realmente messa in crisi dalle malelingue sul presunto stato di crisi in cui verserebbe, anche fossero state false dicerie.)
  - George Herbert Mead: "I significati e le identità (personali e sociali) emergono tramite le interazioni sociali"
  - Charles Cooley (il teorico del "sé specchio"): "Il comportamento degli individui tiene molto conto di ciò che gli altri pensano, della nostra percezione della percezione che gli altri hanno di lui." - l'interazione è centrale e mentalistica.

  La teoria
  - La definizione basale
    Devianza : etichettamento, che resiste nel tempo tramite l'interazione.
    E' una definizione molto poco centrata sul deviante, che rovescia la prospettiva sul processo con cui identifichiamo ed etichettiamo il deviante, che quindi è un costrutto sociale.

- Howard Becker, "Outsiders. Saggi di sociologia della devianza", 1963
  I gruppi sociali creano la devianza, istituendo norme la cui infrazione costituisce la devianza stessa, applicando quelle norme a certe persone, e a quelle persone l'etichetta di deviante. La devianza non è una qualità dell'atto commesso, piuttosto una conseguenza dell'applicazione, da parte di altri, di normee sanzioni verso un "colpevole". Il comportamento deviante è il comportamento così etichettato.
- Edwin Lemert
  La devianza è comune, e spesso senza conseguenze. Due tipi:
  * Devianza primaria (poligenetica. Anche se socialmente sgradita, la devianza primaria presenta implicazioni marginali per lo status e per la struttura psichica della persona interessata)
  * Devianza secondaria (la scoperta del reato, tramite la stigmatizzazione, porta alla creazione sociale e psicologica del deviante. L'individuo inizia a percepirsi come deviante, fino a farla diventare la caratteristica principale - cioè, fino a sentirsi pi deviante che individuo; potrebbe quindi iniziare una "carriera criminale".)

Capitolo 5

Risorse impiegate per la stesura della tesi

Bibliography

[1] Hany Farid Julia Dressel. *The accuracy, fairness, and limits of predicting recidivism.* 1996.

[2] Carlo Sbordone Paolo Marcellini. *Esercizi di Matematica, Volume II, Tomo I.* Liguori editore, 2009. ISBN: 9788820746490.

[3] Giovanni Sartor. *Intelligenza artificiale e diritto. Un'introduzione.* Giuffrè editore, 1996. ISBN: 9788814053863.

[4] Giovanni Sartor, Francesca Lagioia et al. "Il sistema COMPAS: algoritmi, previsioni, iniquità". In: *XXVI Lezioni di Diritto dell'Intelligenza artficiale.* Giappichelli, 2021, pp. 226–244.

[5] Luigi Viola. *Interpretazione della legge con modelli matematici. Processo, a.d.r., giustizia predittiva. Volume I.* 2ª ed. Centro Studi Diritto Avanzato Edizioni, 2018. ISBN: 9788828388708.

Sitography

[6] *Artificial Intelligence: current Italian regulations and future perspectives.* URL: `https://www.brocardi.it/tesi-di-laurea/artificial-intelligence-current-italian-regulations-future/819.html`.

[7] *COMPAS analysis.* URL: `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

[8] *COMPAS analysis - GitHub resources.* URL: `https://github.com/propublica/compas-analysis/tree/master`.

[9] *Giustizia predittiva: dubbi tra innovazione e regresso*. URL: https://www.altalex.com/documents/2024/01/04/giustizia-predittiva-dubbi-innovazione-regresso.

[10] *Giustizia predittiva: quale futuro?* URL: https://www.altalex.com/documents/news/2023/03/15/giustizia-predittiva-quale-futuro.

[11] *Il museo di antropologia criminale*. URL: https://www.museolombroso.unito.it/museo/intro/.

[12] *Machine bias: risk assessments in criminal sentencing*. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[13] *Se l'amicus curiae è un algoritmo: il chiacchierato caso Loomis alla Corte Suprema del Wisconsin*. URL: https://www.giurisprudenzapenale.com/2019/04/24/lamicus-curiae-un-algoritmo-chiacchierato-caso-loomis-alla-corte-suprema-del-wisconsin/.

[14] *Understanding the Bias-Variance Tradeoff*. URL: https://scott.fortmann-roe.com/docs/BiasVariance.html.

Appendice A

test_sqlite.py

```
import sqlite3
from datetime import date

### Date handler (conversion to integers)

pot_dates = lambda giorno: (giorno-date(1936,6,24)).days # Ciao, nonna.
date_converter = lambda item: None if item == None or item == '' else pot_dates(

### Utilities and Loading data

file = "data\\compas-analysis\\compas.db"
con = sqlite3.connect(file)
cur = con.cursor()
gen = lambda col: cur.execute(f"SELECT {col} FROM people WHERE is_recid != -1")
gen_list = lambda gen: [val for tup in gen for val in tup]
date_int = lambda lst: list(map(date_converter, lst))
table_gen = lambda data: {col : gen_list(gen(col)) for col in data}
lindt = [*cur.execute("PRAGMA table_info('people')")]
columns = [lindt[_][1] for _ in range(len(lindt))]
data = {col : gen_list(gen(col)) for col in columns}

none_number = lambda caller, data, columns: print(*[f"{caller}" + f"{i}:" + f"N
people = ('id', 'first', 'last')
people_data = table_gen(people)

### Proiezione algebrica dei dati

short_columns = ('id', 'sex', 'race', 'dob', 'age', 'juv_fel_count', 'juv_misd_c
short_data = table_gen(short_columns)
# none_number('projected_data: ', short_data, short_columns) # projected data st
```

```
assert data['is_recid'].count(-1) == 0
#print("\'decile_score\': ", data['decile_score'].count(-1)) # Output: 15

race_uniques = [*set(short_data['race'])]
race_list = [f"is_{i.lower()}" for i in race_uniques]
transformed = ('is_male', 'dob_tr', *race_list)
# data['id'] = [data['id'][i] -1 for i in range(len(data['id']))]
short_data['id'] = [item-1 for item in short_data['id']]
assert (short_data['id'][i] == i for i in range(len(short_data['id'])))
short_data['is_male'] = [int(item == 'Male') for item in short_data['sex']]
short_data['dob_tr'] = [date_converter(item) for item in short_data['dob']]
for i in range(len(race_uniques)):
    short_data[race_list[i]] = [int(item == race_uniques[i]) for item in short_d
# none_number('transformed_data: ', short_data, transformed) # revised columns
kys = [*short_data.keys()]
#print(kys)
kys.remove('sex')
kys.remove('race')
kys.remove('dob')
kys.append(kys.pop(6))
kys.append(kys.pop(6))
#print(*kys, sep="\n")
ultimate_data_dict = {i : short_data[i] for i in kys}
for key, values in ultimate_data_dict.items():
    for value in values:
        assert type(value) == int
ultimate_data_list = [*ultimate_data_dict.values()]
# print(ultimate_data_list)
```

Appendice B

test_sklearn.py

```
from test_sqlite import *
from prettytable import PrettyTable
import matplotlib.pyplot as plt

from sklearn.preprocessing import *
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor, VotingRegressor
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.decomposition import PCA
from sklearn.metrics import r2_score, mean_squared_error, make_scorer
from sklearn.tree import plot_tree


Z = ultimate_data_dict
Z.pop('id')
y = Z.pop('decile_score')
X = list(map(list, zip(*Z.values()))) # Transpose the list
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.2, random_sta

scalers={'standard': StandardScaler(), 'minmax': MinMaxScaler(), 'quantile': Qua
models = {
    'logistic': LogisticRegression(max_iter=1000),
    'svr': SVR(),
    'knn': KNeighborsRegressor(),
    'random_forest': RandomForestRegressor(),
    'voting': VotingRegressor(estimators=[
        ('knn', KNeighborsRegressor()),
```

```
        ('random_forest', RandomForestRegressor()),
        ])
}
pipelines = {(scaler, estimator): make_pipeline(scalers[scaler], models[estimato
scorers = {'mse': make_scorer(mean_squared_error), 'r2': make_scorer(r2_score)}
t = PrettyTable(["Model", "Pre-scaler", "mse_score", "r2_score"])
t.add_rows([[key[1], key[0], f"{scorers['mse'](pipeline, X_test, y_test):.4f}",
print(t)

def PCA_shower(n_components, preprocessor):
    print(f"{preprocessor.__name__}".capitalize())
    pca = PCA(n_components=n_components)
    pca.fit(preprocessor(X_train))
    t = PrettyTable(["Feature"] + [f"Component {i+1}" for i in range(len(pca.com
    t.add_row(["Explained variance ratio"] + [f"{pca.explained_variance_ratio_[i
    t.add_rows([[feature] + [f"{pca.components_[j][i]:.4f}" for j in range(len(p
    return t

print(PCA_shower(5, quantile_transform))
print(PCA_shower(5, minmax_scale))
print(PCA_shower(5, scale))

get_rfr = lambda pipeline: pipeline.named_steps['randomforestregressor']
random_forest=get_rfr(pipelines[('standard', 'random_forest')])
########print(get_rfr(pipelines[('standard', 'random_forest')]))
# Qui esce la foresta della pipeline allenata alla linea 34, se la pipeline ne h

tree_scores=[(i, mean_squared_error(y_test, tree.predict(X_test))) for i, tree i
tree_scores_sorted=sorted(tree_scores, key=lambda x: x[1])
best_tree_index, best_tree_mse=tree_scores_sorted[0]
best_tree=random_forest.estimators_[best_tree_index]
print(f"best tree index: {best_tree_index}")
print(f"MSE best tree: {best_tree_mse:.4f}")

plt.figure(figsize=(10,5))
plot_tree(best_tree, max_depth=2, feature_names=list(Z.keys()), filled=True)
plt.show()
```