



POLITECNICO
MILANO 1863

Progetto di Ingegneria Informatica 2023-2024

Titolare del corso: Prof.ssa Mariagrazia Fugini

Responsabili: Prof.ssa Letizia Tanca,

Dott.ssa Chiara Criscuolo, Dott. Mathyas Giudici

**STUDIARE LA FAIRNESS NEL
MACHINE LEARNING: UNA
SOLUZIONE BASATA SU CHATBOT**

Mattia Vicenzotto (Codice persona 10722332 - Matricola 956134)

1 Introduzione

Negli ultimi anni, l'uso Big Data e algoritmi basati su machine learning è diventato parte della nostra vita quotidiana. Queste tecnologie, basate sui dati, possono essere affidabili solo se i dati su cui sono costruite sono privi di bias e disparità. Tuttavia, spesso accade che gli algoritmi sviluppati in vari ambiti, inclusi quelli medici e giudiziari, siano addestrati su dati che possono contenere disparità tra individui.

Questo progetto analizza la fairness (equità) nei modelli di machine learning, concentrandosi su come questi modelli, quando addestrati su un dataset di donne con possibili diagnosi di diabete, possano comportarsi in modo equo o ingiusto. Nell'analisi sono stati presi in considerazione due modelli addestrati rispettivamente con tecniche di bagging e random forest. Le metriche di fairness considerate includono Group Fairness, Predictive Parity, Predictive Equality, Equal Opportunity, Equalized Odds, Conditional Use Accuracy Equality, Overall Accuracy Equality, Treatment Equality e FOR Parity, con un focus particolare sulla categoria d'età (donne sotto e sopra i 25 anni).

I risultati rivelano differenze significative tra i modelli in termini di equità, con alcuni indicatori che suggeriscono una discriminazione verso le donne giovani. Per rendere i risultati accessibili, è stata sviluppata una pagina web con un chatbot generativo basato sul modello GPT3.5 di OpenAI, che guida l'utente nell'interpretazione delle metriche di fairness.

La relazione si articola come segue: nella sezione 2 viene descritto il dataset, e come sono stati addestrati i modelli di ML utilizzati; nella sezione 3 sono descritte le metriche di fairness prese in considerazione; nella sezione 4 è spiegato come sono state calcolate le metriche di fairness ed una loro analisi; nella sezione 5 verrà trattato come è stato sviluppato il chatbot usando le API di Open-AI, nella sezione 6 le conclusioni e le implicazioni.

2 Dataset e modelli Machine Learning

2.1 Dataset

Il dataset¹ usato proviene originariamente dal National Institute of Diabetes and Digestive and Kidney Diseases (Istituto Nazionale di Diabete e Malattie Digestive e Renali). L'obiettivo del dataset è prevedere, basandosi su misurazioni diagnostiche, se un paziente ha il diabete o meno.

Sono state poste diverse restrizioni nella selezione di questi casi da un dataset più grande. In particolare, tutti i pazienti in questo dataset sono donne di almeno 21 anni di età e di origine Pima (una tribù nativa americana).

2.2 Caratteristiche del Dataset:

Le variabili incluse nel dataset sono le seguenti:

- Pregnancies: Numero di gravidanze
- Glucose: Concentrazione plasmatica di glucosio 2 ore dopo un test di tolleranza al glucosio orale
- BloodPressure: Pressione sanguigna diastolica (mm Hg)
- SkinThickness: Spessore della piega cutanea del tricipite (mm)
- Insulin: Insulina sierica a 2 ore (μ U/ml)
- BMI: Indice di massa corporea (peso in kg/(altezza in m)²)
- DiabetesPedigreeFunction: Funzione pedigree del diabete (una misura che rappresenta la probabilità di ereditarietà del diabete)
- AgeCategory: Variabile che può assumere valori 0 o 1, dove 0 indica che la persona ha 25 anni o meno e 1 indica che ne ha più di 25
- Outcome: Variabile che può assumere valori 0 o 1, dove 0 indica l'assenza di diabete e 1 indica la presenza di diabete

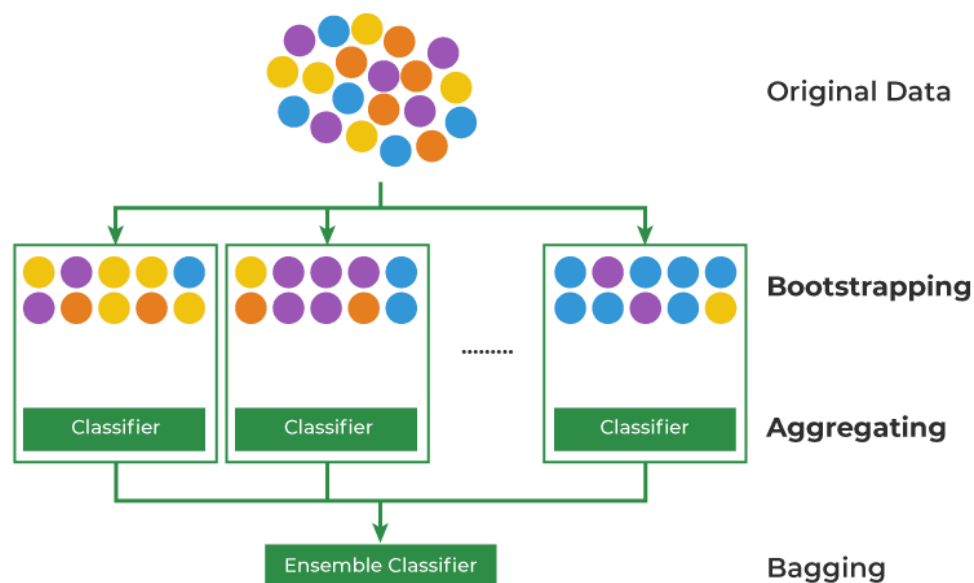
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Outcome	AgeCategory
0	6	148	72	35	0	33.6	0.627	1	1
1	1	85	66	29	0	26.6	0.351	0	1
2	8	183	64	0	0	23.3	0.672	1	1
3	1	89	66	23	94	28.1	0.167	0	0
4	0	137	40	35	168	43.1	2.288	1	1
5	5	116	74	0	0	25.6	0.201	0	1
6	3	78	50	32	88	31.0	0.248	1	1
7	4	110	92	0	0	37.6	0.191	0	1
8	10	168	74	0	0	38.0	0.537	1	1
9	10	139	80	0	0	27.1	1.441	0	1

2.3 Modelli Machine learning

Sia il BaggingClassifier che il RandomForestClassifier sono tecniche di ensemble che utilizzano il metodo del bagging (Bootstrap Aggregating) per migliorare la stabilità e l'accuratezza dei modelli di machine learning. Entrambi creano multiple versioni del dataset originale tramite campionamento con sostituzione e addestrano un modello su ciascuno di questi sottoinsiemi. Le previsioni finali sono ottenute aggregando le previsioni dei singoli modelli, riducendo la varianza e aumentando la robustezza rispetto a un singolo modello.

BaggingClassifier:

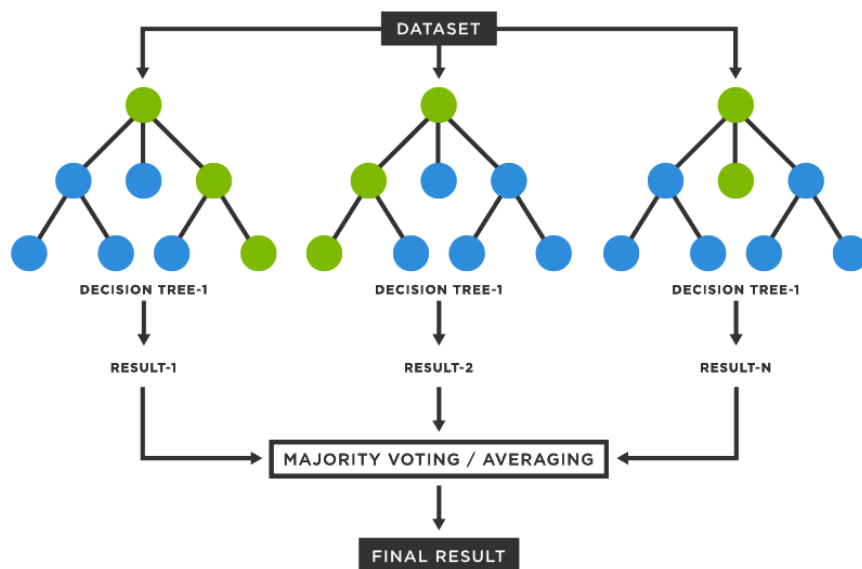
Il BaggingClassifier² è una tecnica di ensemble generica che può essere applicata con qualsiasi classificatore come modello base. Funziona creando diversi sottoinsiemi del dataset originale attraverso campionamento bootstrap, addestrando un modello su ciascuno di questi sottoinsiemi e combinando le previsioni dei modelli attraverso il voto di maggioranza per i problemi di classificazione o la media per i problemi di regressione.



² <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html>
<https://www.geeksforgeeks.org/ml-bagging-classifier/>

RandomForestClassifier:

Il RandomForestClassifier³ è una tecnica di ensemble specializzata che utilizza alberi decisionali come modelli base e introduce ulteriori casualizzazioni per migliorare la diversità tra gli alberi. Oltre al campionamento bootstrap per creare sottoinsiemi del dataset, il RandomForestClassifier randomizza anche la selezione delle feature a ogni split nei nodi degli alberi. Questa casualizzazione aggiuntiva riduce la correlazione tra gli alberi, migliorando la robustezza e l'accuratezza del modello finale.



2.4 Train-Test Split (Validazione modelli)

Per la validazione dei modelli è stata usata la tecnica Train-Test Split⁴. Consiste nel dividere il dataset originale in due insiemi: uno per l'addestramento (train set) e uno per il test (test set). Nel caso di questo progetto è stata scelta una suddivisione 70-30, ovvero un 70% dei dati viene usato per l'addestramento e 30% per il test.

³ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
<https://medium.com/@dishantkharkar9/about-random-forest-algorithms-62163357db25>

⁴ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Codice:

```
[ ] # Define train/test split ratio
test_size = 0.3 # You can adjust this ratio based on your preference

Y = df[target_variable]
X = df.drop(target_variable, axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=1)

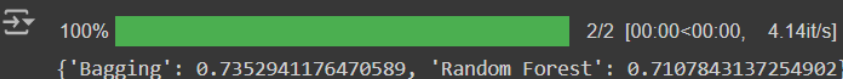
models = {'Bagging': BaggingClassifier(estimator=DecisionTreeClassifier()).fit(X_train, y_train),
          'Random Forest': RandomForestClassifier().fit(X_train, y_train)}

scores = {}
predicted_values = {}

for model_name in tqdm(models):
    clf = models[model_name]
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    predicted_values[model_name] = y_pred

    score = clf.score(X_test, y_test)
    scores[model_name] = score

print(scores)
```



{'Bagging': 0.7352941176470589, 'Random Forest': 0.7107843137254902}

Il codice in questione rappresenta la parte di addestramento dei modelli, con le modalità descritte fino a questo momento.

Gli “scores” rappresentati sotto indicano la precisione ottenuta nelle predizioni per ogni modello addestrato:

Bagging: 73.53%

Random Forest: 71.08%

2.5 Confusion Matrices

Infine, dai risultati ottenuti sono state generate le Confusion Matrices, che rappresentano l’accuratezza delle previsioni raggruppandole in 4 possibili gruppi: True Positive (TP), True Negative (TN), False Positive (FP) e False Negative (FN). Per ogni modello sono state generate 2 Confusion Matrices: una per il gruppo discriminato (donne con 25 anni o meno) ed una per il gruppo privilegiato (donne con più di 25 anni).

Codice:

```
confusion_matrices = {}
for model_name in predicted_values.keys():
    temp_dict = {}
    #print(model_name)
    #print(classification_report(df[target_variable], predicted_values[model_name]))

    temporary = list()
    for idx, i in enumerate(y_test.index):
        temporary.append((i, predicted_values[model_name][idx]))
    #print(temporary)

    y_pred_df = pd.DataFrame(temporary, columns = ['index', 'y_pred'])
    #print(y_pred_df)
    y_pred_df.set_index(keys=y_pred_df['index'], inplace = True)
    y_pred_df.drop(columns='index', axis=1, inplace= True)
    #print(y_pred_df)

    y_pred_discriminated = y_pred_df.loc[y_pred_df.index.intersection(discriminated)]
    y_pred_privileged = y_pred_df.loc[y_pred_df.index.intersection(privileged)]

    cm_discriminated = confusion_matrix(y_test_discriminated, y_pred_discriminated)
    cm_privileged = confusion_matrix(y_test_privileged, y_pred_privileged)
    temp_dict['discriminated'] = cm_discriminated
    temp_dict['privileged'] = cm_privileged
    confusion_matrices[model_name] = temp_dict

    print(model_name)
    print(confusion_matrices[model_name]['discriminated'])
    print(confusion_matrices[model_name]['privileged'])
```

```
↔ Bagging
[[61  0]
 [ 8  4]]
[[49 22]
 [24 36]]
Random Forest
[[61  0]
 [ 8  4]]
[[45 26]
 [25 35]]
```

Per ogni modello la prima matrice rappresenta il gruppo discriminato, la seconda quello privilegiato.

3 Metriche di fairness

Di seguito verranno spiegate le metriche usate nell'analisi. Queste metriche di fairness⁵ sono utilizzate per valutare il comportamento dei modelli di machine learning in termini di fairness tra diversi gruppi. Ogni metrica mira a misurare un aspetto specifico nel processo decisionale del modello.

Terminologia:

- d è il valore predetto
- Y è il valore effettivo nel dataset
- G è l'attributo protetto, priv=gruppo privilegiato, discr=gruppo discriminato

Group Fairness: $(d=1|G=priv) = (d=1|G=discr)$

(a.k.a. statistical parity, equal acceptance rate, benchmarking).

Un classificatore soddisfa questa definizione se i soggetti nei gruppi protetti e non protetti hanno la stessa probabilità di essere assegnati alla classe predetta positiva.

Predictive Parity: $(Y=1|d=1, G=priv) = (Y=1|d=1, G=discr)$

(a.k.a. outcome test).

Un classificatore soddisfa questa definizione se sia i gruppi protetti che non protetti hanno lo stesso valore predittivo positivo (PPV) - la probabilità che un soggetto con un valore predittivo positivo appartenga veramente alla classe positiva.

Predictive Equality: $(d=1|Y=0, G=priv) = (d=1|Y=0, G=discr)$

(a.k.a. False positive error rate balance).

Un classificatore soddisfa questa definizione se sia i gruppi protetti che non protetti hanno lo stesso tasso di errore falso positivo (FPR) - la probabilità che un soggetto nella classe negativa abbia un valore predittivo positivo.

Equal Opportunity: $(d=0|Y=1, G=priv) = (d=0|Y=1, G=discr)$

(a.k.a. False negative error rate balance).

Un classificatore soddisfa questa definizione se sia i gruppi protetti che non protetti hanno lo stesso tasso di errore falso negativo (FNR) - la probabilità che un soggetto nella classe positiva abbia un valore predittivo negativo.

Equalized Odds: $(d=1|Y=i, G=priv) = (d=1|Y=i, G=discr), i \in \{0, 1\}$

(a.k.a. conditional procedure accuracy equality and disparate mistreatment).

Questa definizione combina le due precedenti: un classificatore soddisfa la definizione se i gruppi protetti e non protetti hanno lo stesso TPR e lo stesso FPR. Matematicamente, è equivalente alla congiunzione delle condizioni per il bilancio del tasso di errore falso positivo e del tasso di errore falso negativo.

⁵ <https://dl.acm.org/doi/10.1145/3194770.3194776>

Conditional Use Accuracy Equality: $(Y=1|d=1, G=priv) = (Y=1|d=1, G=discr)$ and $(Y=0|d=0, G=priv) = (Y=0|d=0, G=discr)$

Questa definizione congiunge due condizioni: equal PPV e NPV - la probabilità che i soggetti con valore predittivo positivo appartengano veramente alla classe positiva e la probabilità che i soggetti con valore predittivo negativo appartengano veramente alla classe negativa.

Overall Accuracy Equality: $(d=Y, G=priv) = (d=Y, G=discr)$

Un classificatore soddisfa questa definizione se sia i gruppi protetti che non protetti hanno la stessa accuratezza di predizione - la probabilità che un soggetto sia assegnato alla sua classe rispettiva, sia positiva che negativa. La definizione presuppone che i veri negativi siano desiderabili quanto i veri positivi.

Treatment Equality: $(Y=1, d=0, G=priv)/(Y=0, d=1, G=priv) = (Y=1, d=0, G=discr)/(Y=0, d=1, G=discr)$

Questa definizione guarda il rapporto degli errori che il classificatore compie piuttosto che la sua accuratezza. Un classificatore soddisfa questa definizione se sia i gruppi protetti che non protetti hanno lo stesso rapporto di falsi negativi e falsi positivi.

FOR Parity: $(Y=1|d=0, G=priv) = (Y=1|d=0, G=discr)$

Confronta i tassi di omissione falsi (FOR) tra diversi gruppi demografici.

4 Calcolo ed analisi delle metriche

Sono state calcolate le metriche per entrambi i modelli. Sono state calcolate in 2 modi diversi (divisione e sottrazione), seguendo però la stessa definizione per ciascuna metrica.

4.1 Calcolo delle metriche

prendiamo come esempio il calcolo della metrica “**Group Fairness**”. Il calcolo delle altre metriche avviene secondo lo stesso principio ma semplicemente con formule diverse. Diventerebbe ridondante includere tutte le metriche.

Codice:

calcolo tramite divisione:

```
# Fairness metrics computed using division operator
def fairness_metrics_division(confusion_matrix, discriminated, privileged, threshold = 0.15):

    TP_priv, TN_priv, FP_priv, FN_priv = retrieve_values(confusion_matrix['privileged'])
    TP_discr, TN_discr, FP_discr, FN_discr = retrieve_values(confusion_matrix['discriminated'])

    GroupFairness_discr = (TP_discr+FP_discr)/len(discriminated)
    GroupFairness_priv = (TP_priv+FP_priv)/len(privileged)
    GroupFairness = GroupFairness_discr/GroupFairness_priv
```

Per calcolare la metrica di Group Fairness vengono sommati i True Positive ed i False Positive (per ottenere il numero totale di predizioni positive) e rispettivamente vengono divisi per il gruppo di appartenenza (discriminato o privilegiato). Infine viene fatto il rapporto tra i due valori trovati.

Se i membri appartenenti ai diversi gruppi hanno la stessa probabilità di essere positivi (secondo il nostro modello) il valore della metrica dovrebbe essere 1. Se devia da 1 allora secondo questa metrica il modello viola il principio di fairness.

calcolo tramite sottrazione:

```
# Fairness metrics computed using subtraction operator
def fairness_metrics_subtraction(confusion_matrix, discriminated, privileged, threshold = 0.15):

    TP_priv, TN_priv, FP_priv, FN_priv = retrieve_values(confusion_matrix['privileged'])
    TP_discr, TN_discr, FP_discr, FN_discr = retrieve_values(confusion_matrix['discriminated'])

    GroupFairness_discr = (TP_discr+FP_discr)/len(discriminated)
    GroupFairness_priv = (TP_priv+FP_priv)/len(privileged)
    GroupFairness = GroupFairness_priv-GroupFairness_discr
```

Viene applicato lo stesso procedimento a meno dell'ultima operazione. al posto della divisione viene fatta la sottrazione tra i due valori trovati. Quindi in questo caso se i due gruppi hanno la stessa probabilità di avere dei valori previsti positivi la metrica dovrebbe valere 0. Se devia da zero significativamente viola il principio di fairness.

4.2 Rescaling delle metriche

Per confrontare più facilmente le metriche tra di loro (quelle calcolate tramite divisione e sottrazione), viene fatto uno shift delle metriche calcolate tramite la divisione pari a -1. Ciò significa che una metrica al posto di essere equa(fair) quando è pari ad 1, ora lo sarà quando si avvicina a 0. vengono inoltre standardizzati i valori: se una metrica risulta maggiore di 1 viene posta ad 1, in caso contrario se minore di -1 viene posta a -1.

4.3 Scelta delle metriche

Dati i risultati delle metriche, andiamo ad analizzare le più significative:

Group Fairness: Fornisce una visione macro del bilanciamento delle previsioni tra i gruppi.

Equal Opportunity: Garantisce equità nell'identificazione dei veri positivi.

Predictive Equality: Garantisce equità nell'evitare falsi positivi.

Overall Accuracy Equality: Verifica la performance uniforme, assicurando che nessun gruppo soffra costantemente di una precisione inferiore.

Nel contesto sanitario, una visione d'insieme della fairness implica valutare come vari aspetti della predizione del modello (come i tassi di veri positivi, falsi positivi, accuratezza complessiva, ecc.) influenzino differenti gruppi di pazienti. Ad esempio, non è sufficiente garantire che i tassi di diagnosi corretta siano equi (**Equal Opportunity**), ma è anche importante che i tassi di falsi positivi non siano sbilanciati (**Predictive Equality**) e che l'accuratezza complessiva del modello sia uniforme tra i gruppi (**Overall Accuracy Equality**). Questo approccio assicura che tutte le possibili dimensioni della fairness siano considerate e che nessun gruppo sia sistematicamente svantaggiato o avvantaggiato.

4.4 Risultati ed analisi delle metriche

Bagging:			Random Forest:		
Metric	Division	Subtraction	Metric	Division	Subtraction
GroupFairness	-0.77447	0.10186	GroupFairness	-0.87113	0.11457
PredictiveParity	0.15079	-0.0936	PredictiveParity	0.61111	-0.37931
PredictiveEquality	-0.89419	0.27707	PredictiveEquality	-1	0.30986
EqualOpportunity	-0.31429	-0.18333	EqualOpportunity	-0.4	-0.26667
EqualizedOdds	-0.89419	0.18333	EqualizedOdds	-1	0.26667
ConditionalUseAccuracyEquality	0.33179	-0.22271	ConditionalUseAccuracyEquality	0.61111	-0.37931
OverallAccuracyEquality	0.20408	1	OverallAccuracyEquality	0.2449	1
TreatmentEquality	-0.68831	-1	TreatmentEquality	-1	-1
FORParity	1	0.22271	FORParity	1	0.21283
FN	0.83479	0.02476	FN	0.60544	0.02052
FP	-0.83012	-0.04141	FP	-1	-0.04989

Analizziamo il modello generato con Bagging, in particolare le metriche calcolate usando il metodo della divisione:

Un risultato equo sarebbe un valore della metrica vicino a zero "0". Un numero negativo significa che il gruppo discriminato è effettivamente discriminato, un valore positivo significa che il gruppo privilegiato è discriminato. Consideriamo come soglia per la discriminazione una deviazione da 0 di 0,15. Ciò significa che per una metrica compresa tra i valori di -0.15 e 0.15 il modello non sembra presentare discriminazione.

Group Fairness: -0.77447, significa che c'è un significativo grado di ingiustizia. La percentuale totale di positivi per il gruppo discriminato è molto più bassa rispetto alla percentuale di positivi per il gruppo privilegiato.

Equal Opportunity: -0.31429, considerando questa metrica, potrebbe non sembrare così ingiusto, ma c'è ancora un bias significativo. Questo risultato implica che un membro del gruppo discriminato ha maggiori probabilità di ricevere un valore predetto negativo erroneamente (falso negativo).

Predictive Equality : -0.89419, questa metrica ci dice che il modello favorisce altamente il gruppo privilegiato. Un membro con un valore effettivamente negativo del gruppo privilegiato ha molte più probabilità di ottenere un valore predetto positivo (falso positivo).

Overall Accuracy Equality: 0.20408, secondo i nostri standard siamo vicini all'equità secondo questa metrica, in generale ci dice che i valori predetti per il gruppo privilegiato sono meno accurati

5 Chatbot

Data la complessità dei risultati, un utente inesperto nel campo del Machine Learning, in particolare riguardo alla fairness, avrebbe difficoltà a comprenderli appieno. Per questo motivo, si è deciso di sviluppare un chatbot che possa guidare l'utente nell'analisi e nella descrizione delle metriche di fairness.

5.1 Tecnologie e metodologie usate

- **Flask**⁶: framework per lo sviluppo web in Python. È stato utilizzato per creare la pagina web. Leggero e semplice, consente di gestire facilmente le richieste HTTP e le API.
- **Azure OpenAI**⁷ (**gpt-35-turbo**): attraverso la sottoscrizione ad “Azure for students” è stato possibile accedere al servizio di “Azure OpenAI”, grazie al quale è stato implementato il modello gpt-35-turbo, necessario per il funzionamento del chatbot.
- **Prompt Engineering**: per realizzare e definire il comportamento di interazione con l'utente del chatbot è necessario usare una serie di prompt iniziali⁸ (invisibili all'utente) per indirizzare la conversazione nello stile desiderato, oltre a fornire al LLM tutte le informazioni sulle metriche ed i risultati ottenuti.

5.2 Codice e sviluppo Chatbot

Link al repository su Github:

<https://github.com/Vice41/Progetto-Ingegneria-Informatica>

```
api_key = os.getenv("AZURE_OPENAI_API_KEY")
azure_endpoint = os.getenv("AZURE_OPENAI_ENDPOINT")

client = AzureOpenAI(
    api_key=api_key,
    api_version="2024-02-01",
    azure_endpoint=azure_endpoint
)
```

Per configurare le API, le chiavi "api_key" e l'endpoint "azure_endpoint" sono contenute nel file .env, dato che sono private e non dovrebbero essere condivise pubblicamente. Questo file .env deve essere incluso nel file .gitignore per evitare che venga caricato nei repository pubblici.

⁶ <https://flask.palletsprojects.com/en/3.0.x/>

⁷ <https://azure.microsoft.com/it-it/products/ai-services/openai-service>

⁸ <https://www.promptingguide.ai/>

```

messagesMemory = []

# Function to add a new message while maintaining the limit of 5 messages
def add_message(msgList, new_message):
    msgList.append(new_message)
    if len(msgList) > 5:
        del msgList[0] # Delete the oldest message if the limit is exceeded

@app.route('/')
def home():
    # just render the HTML homepage
    return render_template("index.html")

@app.route('/process', methods=['POST'])
def detect_intent():
    text=request.form["message"]
    add_message(messagesMemory,
                {
                    "role": "user",
                    "content": text,
                })
    currentMessages=[
        {
            "role": "system",
            "content": initialPrompt,
        },
    ]
    #add the up to the last 5 messages as history
    for msg in messagesMemory:
        currentMessages.append(msg)

    chat_completion = client.chat.completions.create(
        model="fairBot",
        messages=currentMessages,
    )

    add_message(messagesMemory,
                {
                    "role": "assistant",
                    "content": chat_completion.choices[0].message.content,
                })

    return str(chat_completion.choices[0].message.content)

```

add_message(msgList, new_message):
 aggiunge un nuovo messaggio alla lista msgList. Se la lista supera i 5 messaggi, il messaggio più vecchio viene rimosso per mantenere il limite di 5.

Mantenere in memoria i messaggi precedenti permette al Chatbot di rispondere conoscendo il contesto della chat. Salvare più di 5 messaggi diventa esponenzialmente più dispendioso dal punto di vista dei token usati per ottenere una risposta.

detect_intent(): gestisce l'elaborazione dei messaggi inviati dall'utente:

- Riceve un messaggio dall'utente tramite **request.form["message"]**
- Aggiunge il messaggio alla lista messagesMemory utilizzando **add_message**
- Crea una lista currentMessages che contiene un messaggio di sistema iniziale (initialPrompt) e gli ultimi 5 messaggi dalla lista messagesMemory
- Invia questa lista di messaggi al modello di OpenAI tramite **client.chat.completions.create**
- Aggiunge la risposta generata dal modello a messagesMemory e restituisce la risposta come stringa

5.3 Prompt iniziale

Il prompt iniziale dato al modello può essere diviso nelle seguenti sezioni.

Descrizione generale del comportamento:

You are a helpful assistant and should answer me clearly.

Specifically you are a chatbot. Your responses should be short and understandable (less than 800 characters!).

You should guide the user into asking meaningful questions, and navigate the results.

Informazioni sul dataset:

I have a dataset made of women that might have diabetes, with different data and the outcome (diabetes or not diabetes)

the columns are: Pregnancies Glucose BloodPressure SkinThickness Insulin BMI

DiabetesPedigreeFunction Outcome AgeCategory

I trained the 2 ML model one using bagging, the other with random forest. And I calculated the fairness metrics.

the sensitive attribute is AgeCategory.

age category = 1 means that the person is older than 25 years old

age category = 0 means that the person is younger than 25 years old

the discriminated group selected in the study is the one with value "0". so the younger group

Descrizione delle metriche di fairness:

Fairness Metrics List:

Group Fairness: $(d=1|G=priv) = (d=1|G=discr)$

(a.k.a. statistical parity, equal acceptance rate, benchmarking). A classifier satisfies this definition if subjects in both protected and unprotected groups have equal probability of being assigned to the positive predicted class.

Predictive Parity: $(Y=1|d=1, G=priv) = (Y=1|d=1, G=discr)$

(a.k.a. outcome test). A classifier satisfies this definition if both protected and unprotected groups have equal PPV – the probability of a subject with positive predictive value to truly belong to the positive class.

Predictive Equality: $(d=1|Y=0, G=priv) = (d=1|Y=0, G=discr)$

(a.k.a. False positive error rate balance). A classifier satisfies this definition if both protected and unprotected groups have equal FPR – the probability of a subject in the negative class to have a positive predictive value.

Equal Opportunity: $(d=0|Y=1, G=priv) = (d=0|Y=1, G=discr)$

(a.k.a. False negative error rate balance). A classifier satisfies this definition if both protected and unprotected groups have equal FNR – the probability of a subject in a positive class to have a negative predictive value.

Equalized Odds: $(d=1|Y=i, G=priv) = (d=1|Y=i, G=discr)$, $i \in \{0, 1\}$

(a.k.a. conditional procedure accuracy equality and disparate mistreatment). This definition combines the previous two: a classifier satisfies the definition if protected and unprotected groups have equal TPR and equal FPR. Mathematically, it is equivalent to the conjunction of conditions for false positive error rate balance and false negative error rate balance definitions.

Conditional Use Accuracy Equality: $(Y=1|d=1, G=priv) = (Y=1|d=1, G=discr)$ and $(Y=0|d=0, G=priv) = (Y=0|d=0, G=discr)$

This definition conjuncts two conditions: equal PPV and NPV – the probability of subjects with positive predictive value to truly belong to the positive class and the probability of subjects with negative predictive value to truly belong to the negative class.

Overall Accuracy Equality: $(d=Y, G=priv) = (d=Y, G=discr)$

A classifier satisfies this definition if both protected and unprotected groups have equal prediction accuracy – the probability of a subject from either positive or negative class to be assigned to its respective class. The definition assumes that true negatives are as desirable as true positives.

Treatment Equality: $(Y=1, d=0, G=priv)/(Y=0, d=1, G=priv) = (Y=1, d=0, G=discr)/(Y=0, d=1, G=discr)$

This definition looks at the ratio of errors that the classifier makes rather than at its accuracy. A classifier satisfies this definition if both protected and unprotected groups have an equal ratio of false negatives and false positives.

FOR Parity: $(Y=1|d=0, G=priv) = (Y=1|d=0, G=discr)$

compares the False Omission Rates (FOR) across different demographic groups

Come interpretare le metriche:

the metrics are calculated in 2 ways: with division and subtraction. the

the metric calculated with the division is fair when is close to "1", but in this case we always subtract 1 to the metric calculated with the division so that its centered in "0", so that a value close to 0 means that its fair.

the metric calculated with the subtraction is fair when is close to "0".

the metrics have values from -1 to 1.

a value greater than zero means that the discriminated group is discriminated.

a value smaller than zero means that the privileged group is discriminated.

the threshold for discrimination is a deviation from 0 of 0.15. That means that a for a metric between the values of -0.15 and 0.15 the ml model, in relation to that metric, does not appear to have discrimination.

Valori delle metriche calcolate:

BAGGING

Metric	Division	Subtraction
GroupFairness	-0.77447	0.10186
PredictiveParity	0.15079	-0.0936
PredictiveEquality	-0.89419	0.27707
EqualOpportunity	-0.31429	0.18333
EqualizedOdds	-0.89419	0.18333
ConditionalUseAccuracyEquality	0.33179	-0.22271
OverallAccuracyEquality	0.20408	1
TreatmentEquality	-0.68831	-1
FORParity	1	0.22271

RANDOM FOREST

Metric	Division	Subtraction
GroupFairness	-0.87113	0.11457
PredictiveParity	0.61111	-0.37931
PredictiveEquality	-1	0.30986
EqualOpportunity	-0.4	-0.26667
EqualizedOdds	-1	0.26667
ConditionalUseAccuracyEquality	0.61111	-0.37931
OverallAccuracyEquality	0.2449	1
TreatmentEquality	-1	-1
FORParity	1	0.21283

Possibili domande di follow-up:

For a non-expert user possible routes to navigate could be:

What is fairness in the context of ML?

Why so many metrics and which one to use.

How to interpret the results.

....and so on.

So when the conversation gets stale and the user doesn't know what to ask, suggest these topics.

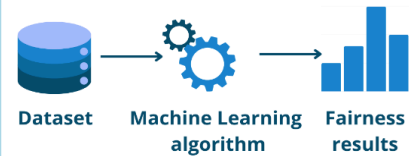
From the next message your conversation with the user shall begin.

Fairness assessment

Dataset

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Outcome	AgeCategory
0	6	148	72	35	0	33.6	0.627	1	1
1	1	85	66	29	0	26.6	0.351	0	1
2	8	183	64	0	0	23.3	0.672	1	1
3	1	89	66	23	94	28.1	0.167	0	0
4	0	137	40	35	168	43.1	2.288	1	1
5	5	116	74	0	0	25.6	0.201	0	1
6	3	78	50	32	88	31.0	0.248	1	1
7	4	110	92	0	0	37.6	0.191	0	1
8	10	168	74	0	0	38.0	0.537	1	1
9	10	139	80	0	0	27.1	1.441	0	1

Pipeline and Results



hello

Hello! How can I assist you with your dataset of women and ML models? Do you have any specific questions or concerns about fairness and the available metrics?

Type here!

Il nostro chatbot si presenta in questo modo. Sulla sinistra in alto un'immagine ci mostra degli esempi di record contenuti nel nostro dataset. Scrivendo nella casella di testo in fondo alla pagina e premendo invio è possibile interagire con il chatbot.

Pipeline and Results

Bagging:			Random Forest:		
Metric	Division	Subtraction	Metric	Division	Subtraction
GroupFairness	-0.77447	0.10186	GroupFairness	-0.87113	0.11457
PredictiveParity	0.15079	-0.0936	PredictiveParity	0.61111	-0.37931
PredictiveEquality	-0.89419	0.27707	PredictiveEquality	-1	0.30906
EqualOpportunity	-0.31429	-0.18333	EqualOpportunity	-0.4	-0.26667
EqualizedOdds	-0.89419	0.18333	EqualizedOdds	-1	0.26667
ConditionalUseAccuracyEquality	0.33179	-0.22271	ConditionalUseAccuracyEquality	0.61111	-0.37931
OverallAccuracyEquality	0.20408	1	OverallAccuracyEquality	0.2449	1
TreatmentEquality	-0.68831	-1	TreatmentEquality	-1	-1
FORParity	1	0.22271	FORParity	1	0.21283
FN	0.83479	0.02476	FN	0.60544	0.02052
FP	-0.83012	-0.04141	FP	-1	-0.04909

Passando con il mouse sopra all'immagine sulla destra è possibile visualizzare le metriche di fairness calcolate.

Un altro metodo per visualizzarle è semplicemente chiedendo al chatbot di elencare i risultati delle metriche di fairness.

what do the results imply?

The results we have show the performance of two different ML models trained on your dataset of women, one using bagging and the other using random forest. The fairness metrics were also calculated for each model. The metrics have values from -1 to 1. A value greater than zero means that the discriminated group is discriminated, while a value smaller than zero means that the privileged group is discriminated. A threshold of 0.15 is used to determine whether or not discrimination is present, meaning that a metric between the values of -0.15 and 0.15 indicates that the ML model doesn't appear to have discrimination with respect to that metric. Based on the results, we can see that some of the fairness metrics may indicate discrimination in one or both of the models, depending on the specific metric and calculation method used. It's important to note that none of these metrics are perfect, and they should be interpreted alongside each other as well as considering other factors that may influence the decision-making process.

what metrics should i consider?

The answer to this question depends on your specific values and priorities as a researcher, as different fairness metrics capture different aspects of fairness. That being said, it is generally recommended to consider a variety of metrics in order to gain the most comprehensive understanding possible of the performance of your ML model. Some of the most commonly used and widely accepted fairness metrics include: - Group Fairness (Statistical Parity) - Predictive Parity - Predictive Equality - Equal Opportunity - Equalized Odds - Conditional Use Accuracy Equality - Overall Accuracy Equality - FOR Parity - Treatment Equality Again, each of these metrics captures a different aspect of fairness, so it may be helpful to consider them all in relation to the specific needs of your research project. I should also note that different organizations or industries may prioritize different fairness metrics depending on their specific goals and values.

based on those metrics is there discrimination in the model?

Type here!



Parlando con il Chatbot è possibile fare domande generali sulla fairness/ML oppure entrare nello specifico nei risultati del caso analizzato. Questo permette ad utenti esperti e non di avere un'esperienza su misura.

6 Conclusioni

Il problema principale della fairness è che si tratta di un argomento complesso per il quale non esistono ancora metodologie “ufficiali” o “provate” utilizzate nel mondo del lavoro e dell'industria. La questione è aggravata dalla mancanza di conoscenza sulla fairness e sui bias tra i professionisti che lavorano con algoritmi di Machine Learning.

Questo progetto propone un metodo per rendere più accessibili i risultati delle analisi di questi algoritmi. Il focus principale è l'analisi e l'interpretazione delle metriche di fairness, indipendentemente dal dataset specifico utilizzato. Se i dati o il modello cambiassero, basterebbe aggiornare il chatbot con le nuove informazioni (nuove metriche e nuovo dataset) per ottenere un'analisi personalizzata ed esaustiva. La capacità del chatbot di fornire risposte chiare e contestualizzate su argomenti complessi come il bias e la fairness dimostra il potenziale dei Large Language Models nell'istruzione e nella comunicazione.

Il progetto ha inoltre sottolineato l'importanza cruciale della fairness nell'addestramento dei modelli di Machine Learning. Ignorare le metriche di fairness può portare a risultati discriminatori, evidenziando la necessità di integrare valutazioni di fairness in tutte le fasi del ciclo di vita del Machine Learning, dall'acquisizione dei dati alla valutazione finale del modello.

Dal punto di vista personale, il progetto ha richiesto l'acquisizione e l'approfondimento di competenze tecniche significative, quali la gestione dei dataset, l'addestramento dei modelli di Machine Learning, il calcolo delle metriche di fairness, la gestione del backend (chiamate alle API) ed il prompt engineering.

In conclusione, il progetto ha messo in luce l'importanza della fairness nel Machine Learning ed ha dimostrato come strumenti innovativi possano rendere queste conoscenze accessibili a un pubblico più ampio. Le competenze acquisite e le metodologie sviluppate durante il progetto rappresentano un solido punto di partenza per ulteriori ricerche e applicazioni pratiche.