



UNIVERSITÀ DEGLI STUDI
DI SALERNO

CORSO DI LAUREA IN INFORMATICA

Unisa Mortgage-AI

Autore:

Antonio Clavino

Docenti:

Prof. Polese Giuseppe

Prof.ssa Caruccio Loredana

Contents

1	Introduzione al problema	4
1.1	Il Machine Learning e l'accesso al credito	4
1.2	Il Contesto Italiano dei Mutui	4
1.3	Il problema degli approcci IA in ambito finanziario	4
1.4	Il problema affrontato	5
1.4.1	Il problema dell'explainability nei modelli di Machine Learning	5
1.4.2	Ambito creditizio.....	5
1.4.3	Il problema dei dati mancanti	5
2	Data Understanding and Data Preparation	6
2.1	Data Gathering	6
2.2	Data Examination	6
2.3	Data Cleaning	7
2.3.1	Verifica dei valori nulli.....	7
2.3.2	Verifica dei duplicati	7
2.3.3	Imputazione e gestione dei valori nulli	8
2.3.4	Creazione delle variabili “Dummy”	8
2.4	Data Exploration	9
2.5	Analisi della variabile dipendente	10
2.5.1	Variabile dipendente.....	10
2.5.2	Distribuzione della variabile dipendente rispetto alcune feature	10
2.6	Matrice di correlazione delle variabili	10
2.7	Bilanciamento delle classi	12
2.7.1	Oversampling e Undersampling	12
2.8	Normalizzazione del dataset.....	13
2.8.1	Encoding delle variabili categoriche	13
2.8.2	Normalizzazione delle variabili continue	13
2.9	Data Splitting	13
3	Sviluppo del modello.....	13
3.1	Tipi di Classificatori.....	13
3.2	Valutazione dei modelli	14
3.3	Come migliorare l'explainability.....	14
3.4	Implementazione.....	14
3.5	Dataset e Addestramento.....	14

4 Training e Valutazione	15
4.1 Metrica per il Training	15
4.2 Decision Tree	15
4.2.1 Valutazione del decision tree	16
4.3 Random Forest	16
4.3.1 Valutazione del Random Forest.....	17
5 Explainability dei modelli implementati	18
5.1 Explainability	18
5.2 Predizione dei modelli implementati	18
5.2.1 Feature Importance	18
5.2.2 Permutation Importance.....	18
5.3 Applicazione delle metriche ai modelli sviluppati.....	18
5.3.1 Feature Importance: Decision Tree	19
5.3.2 Feature Importance: Random Forest.....	19
5.3.3 Permutation Importance: Decision Tree.....	20
5.3.4 Permutation Importance: Random Forest	21
5.4 Interfaccia Grafica del Modello	21
6 Conclusioni.....	22
6.1 Problemi Principali.....	22
6.2 Soluzione ai problemi riscontrati	22
6.3 Risultati	22

1 Introduzione al problema

1.1 Il Machine Learning e l'accesso al credito

Il tema della questione finanziaria legata a mutui con l'ausilio dell'intelligenza artificiale è un tema che negli ultimi e nei prossimi anni potrà rivestire un ruolo sempre più importante. Lo scopo principale di molti studi è l'applicazione di metodi di Machine Learning per definire un modello computazionale di classificazione che selezioni una richiesta di mutuo accettabile o non accettabile basandosi su dati storici.

1.2 Il Contesto Italiano dei Mutui

Nel contesto italiano, la concessione di un mutuo rappresenta un processo articolato che tiene conto del reddito del richiedente, della stabilità lavorativa, degli impegni finanziari in corso, dell'importo richiesto e dello storico creditizio. L'obiettivo è garantire la sostenibilità del finanziamento nel tempo, nel rispetto delle normative vigenti.

Nel sistema bancario italiano, la valutazione di una richiesta di mutuo si basa su diversi fattori, tra cui il reddito annuo del richiedente, la stabilità e la tipologia del contratto lavorativo, la presenza di impegni finanziari pregressi, l'importo del mutuo richiesto, la durata del finanziamento e lo storico creditizio.

In questo scenario, la possibilità di disporre di strumenti intelligenti in grado di fornire una valutazione preliminare e personalizzata della richiesta di mutuo risulta particolarmente rilevante, sia per gli istituti bancari sia per i potenziali clienti.

1.3 Il problema degli approcci IA in ambito finanziario

Come qualsiasi approccio di Machine Learning e IA, il problema principale individuato è la reperibilità dei dati. La discrepanza tra i dati rende molti dei modelli inutilizzabili nell'effettivo, soprattutto perché c'è la necessità di poter utilizzare dati reali e non solo dati di sintesi. Inoltre, l'uso di modelli di IA poco interpretabili può ridurre la fiducia degli utenti e rendere difficile la giustificazione delle decisioni.

Inoltre, l'utilizzo di dati storici può introdurre bias sistematici, legati a pratiche decisionali passate o a squilibri presenti nel dataset. Tali bias possono tradursi in decisioni discriminatorie, con conseguenze rilevanti sia dal punto di vista etico sia normativo.

Infine, la presenza di dati incompleti o mancanti rappresenta un ulteriore problema, in quanto una gestione non adeguata di tali informazioni può compromettere l'affidabilità delle predizioni.

1.4 Il problema affrontato

Partendo da una curiosità personale, l'intenzione di questo progetto è quella di definire un modello che analizzi i dati per classificare correttamente se un mutuo può essere approvato da una banca oppure no, cercando di porre maggiore attenzione sui dati presenti nel dataset, prestando maggiore attenzione sull'explainability, cercando di trovare un approccio migliore per affrontare il problema utilizzando approcci studiati.

1.4.1 Il problema dell'explainability nei modelli di Machine Learning

Uno dei problemi principali del Machine Learning, è l'explainability del modello, ossia quanto il processo decisionale su cui si basa il modello sia comprensibile a chi lo utilizza. Oltre a migliorare la fiducia verso un approccio di IA, migliorare l'explainability di un modello porta benefici in ambito legale e di miglioramento del modello stesso. La capacità di spiegare le decisioni automatiche è fondamentale non solo per garantire la fiducia degli utenti finali, ma anche per rispettare i requisiti normativi in materia di trasparenza e accountability, migliorando l'explainability del modello si può capire il processo decisionale che sta dietro una decisione, rendendo il processo stesso trasparente e affidabile, permettendo di modificare decisioni che portano in errore il modello stesso.

1.4.2 Ambito creditizio

In ambito creditizio, la mancanza di spiegazioni chiare può generare diffidenza nei confronti dei sistemi automatici e limitare la loro adozione. Ad esempio se 2 clienti chiedono un mutuo e ad uno con le stesse caratteristiche dell'altro viene accettato il mutuo, l'altro vorrà capire le motivazioni del declino, quindi il sistema di valutazione del mutuo deve essere in grado di fornire indicazioni comprensibili sui fattori che hanno influenzato l'esito della richiesta.

1.4.3 Il problema dei dati mancanti

I dataset finanziari sono spesso caratterizzati dalla presenza di dati mancanti, dovuti a informazioni non dichiarate o non disponibili. La gestione di tali valori rappresenta una fase critica del processo di preprocessing, in quanto scelte errate possono introdurre bias o ridurre le prestazioni del modello.

2 Data Understanding and Data Preparation

2.1 Data Gathering

Il dataset utilizzato e le risorse necessarie per proseguire con il completamento del progetto sono stati raccolti dallo studio fatto su dati di prestiti e mutui in America ovvero “**Loan-Approval-Prediction-Dataset**” e poi valutandone i dati ho applicato dei parametri “italianizzati” per adattare il caso di studio a ciò che serviva per lo sviluppo del progetto.

2.2 Data Examination

Il primo passo svolto è stato convertire e formattare correttamente il dataset per l'analisi. Il dataset reperito è composto da diverse centinaia di righe e 18 colonne, contenente informazioni generali del richiedente e gli indicatori finanziari rilevati. Le variabili inerenti alle informazioni del richiedente sono divise tra categoriche come il livello di istruzione, il tipo di lavoro e continue come il reddito annuo e il patrimonio. Le variabili inerenti alla situazione finanziaria sono per la maggior parte variabili continue come l'importo del mutuo, liquidità, score CRIF, tranne per alcuni indicatori binari o categorici. In questa pagina verranno elencate e approfondite tutte le feature presenti.

Feature Presenti:

- **mutuo_approvato**: feature target utilizzata per dare il risultato della predizione, i valori che può assumere sono [0,Non Approvato; 1, Approvato].
- **id_mutuo**: identificativo univoco della pratica.
- **familiari_a_carico**: feature numerica che indica il numero di persone a carico del richiedente.
- **livello_istruzione**: feature categorica che indica il titolo di studio.
- **tipo_lavoro**: feature categorica che indica la tipologia contrattuale.
- **reddito_annuo**: feature continua utilizzata per indicare il reddito lordo annuale del richiedente.
- **importo_mutuo**: feature continua che indica la somma richiesta alla banca.
- **durata_mutuo_anni**: feature numerica che indica l'estensione temporale del finanziamento.
- **score_crif**: feature utilizzata per indicare il punteggio di affidabilità creditizia, i valori oscillano in un range predefinito.
- **patrimonio_residenziale**: feature continua che indica il valore degli immobili posseduti.
- **patrimonio_commerciale**: feature continua che indica il valore di asset commerciali.
- **beni_lusso**: feature continua che stima il valore di beni di lusso posseduti.
- **liquidita_bancaria**: feature continua che indica i risparmi o la liquidità disponibile.
- **rata_mensile**: feature calcolata che indica l'esborso mensile previsto.
- **rapporto_rata_reddito**: indicatore finanziario cruciale che rapporta la rata al reddito mensile.
- **crif_negativo**: feature binaria che indica la presenza di segnalazioni negative in passato [0, 1].

2.3 Data Cleaning

Ora passiamo alla fase di data cleaning per vedere se nel nostro dataset è possibile migliorare la qualità dei dati.

2.3.1 Verifica dei valori nulli

Un'operazione importante di data cleaning è la verifica della presenza o meno di valori nulli all'interno del dataset, risolvibile tramite imputazione o cancellazione delle colonne/righe, ma non senza effetti collaterali sul dataset, dato che un'imputazione o una cancellazione potrebbero portare a dati poco rappresentativi o addirittura alla perdita di informazioni importanti. Di seguito verrà riportata l'analisi del dataset e dei valori nulli.

Tabella 1: Analisi Valori Nulli

Column	Non-Null Count	Null Count	Dtype
id_mutuo	4269	0	int64
familiari_a_carico	4269	0	int64
livello_istruzione	4269	0	object
tipo_lavoro	4269	0	object
reddito_annuo	4269	0	int64
importo_mutuo	4269	0	int64
durata_mutuo_anni	4269	0	int64
score_crif	4269	0	int64
patrimonio_residenziale	4269	0	int64
patrimonio_commerciale	4269	0	int64
beni_lusso	4269	0	int64
liquidita_bancaria	4269	0	int64
rata_mensile	4269	0	object

Come possiamo osservare, in questo specifico dataset non sono stati riscontrati valori nulli, il che indica una buona qualità iniziale dei dati raccolti.

2.3.2 Verifica dei duplicati

Un'operazione molto importante di data cleaning, è la verifica dei duplicati. I duplicati sono record in un dataset che si ripetono più volte, portando quindi potenziali bias.

Di seguito saranno riportate le informazioni inerenti ai duplicati.

Numero di duplicati nel dataset: 0. Non essendo presenti record duplicati, non è stato necessario eliminare alcuna riga, mantenendo la grandezza del dataset originale.

Grandezza dataset finale: 4269 records.

2.3.3 *Imputazione e gestione dei valori nulli*

Dalle analisi svolte sul dataset è emerso che non sono presenti valori mancanti. Tuttavia, in uno scenario reale o in caso di dataset più sporchi, la strategia adottata sarebbe stata la seguente:

Per feature numeriche come **importo_mutuo** o **durata_mutuo_anni** con basse percentuali di nulli, si sarebbe proceduto all'imputazione per media o mediana. Per feature categoriche come **tipo_lavoro**, si sarebbe proceduto all'imputazione per moda. Dato che il dataset attuale risulta completo, non sono state necessarie operazioni di imputazione.

2.3.4 *Creazione delle variabili "Dummy"*

Anche se non vi erano valori mancanti, le feature categoriche devono essere convertite in formato numerico per essere elaborate dai modelli di Machine Learning. È stato quindi deciso che in fase di codifica delle variabili categoriche la tecnica utilizzata sarebbe stata quella della codifica "one-hot", trasformando ad esempio *Laurea* in 1 e *Diploma_o_inferiore* in 0 e *Dipendente* in 0 e *Autonomo* in 1. Questa operazione prepara il dataset per la fase successiva.

Colonna	Valori Nulli
id_mutuo	0.0
mutuo_approvato	0.0
rapporto_rata_reddito	0.0
score_crif_norm	0.0
crif_negativo	0.0
familiari_a_carico	0.0
livello_istruzione	0.0
tipo_lavoro	0.0
reddito_annuo	0.0
importo_mutuo	0.0
durata_mutuo_anni	0.0
score_crif	0.0
patrimonio_residenziale	0.0
patrimonio_commerciale	0.0
beni_lusso	0.0

liquidita_bancaria	0.0
rata_mensile	0.0
banca_consigliata	0.0

Tabella 2: Percentuale dati nulli

Come possiamo osservare dai risultati, nessuna colonna risulta avere dati con valore nullo.

2.4 Data Exploration

Una volta terminata l'analisi dei dati e la pulizia del dataset, è stato ritenuto opportuno rimuovere la colonna **id_mutuo**, in quanto non utilizzata se non per indicare l'identificativo del record nel dataset originale e priva di valore predittivo. È opportuno ora quindi capire la struttura assunta dal dataset. Di seguito verranno quindi mostrate le informazioni inerenti al dataset pronto per l'analisi.

La nuova dimensionalità del dataset: (**4269 records, 17 colonne/features**).

Colonna	Valori Nulli
mutuo_approvato	0.0
rapporto_rata_reddito	0.0
score_crif_norm	0.0
crif_negativo	0.0
familiari_a_carico	0.0
livello_istruzione	0.0
tipo_lavoro	0.0
reddito_annuo	0.0
importo_mutuo	0.0
durata_mutuo_anni	0.0
score_crif	0.0
patrimonio_residenziale	0.0
patrimonio_commerciale	0.0
beni_lusso	0.0
liquidita_bancaria	0.0
rata_mensile	0.0

banca_consigliata	0.0
--------------------------	-----

Tabella 3: Analisi Dataset Finale

Come possiamo osservare la maggior parte delle colonne rappresenta valori "int64", il che facilita l'elaborazione numerica.

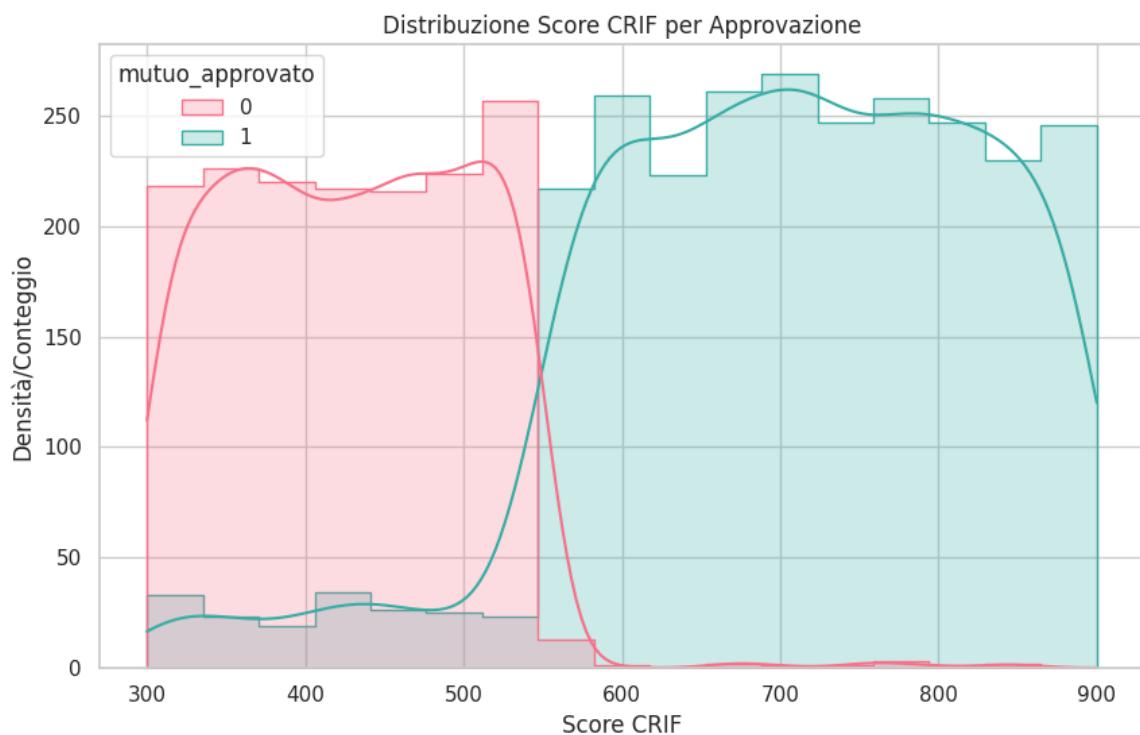
2.5 Analisi della variabile dipendente

2.5.1 *Variabile dipendente*

La variabile dipendente del dataset, ossia la variabile da predire, è riportata all'interno del dataset stesso con la label **mutuo_approvato**. Il valore assunto dalla variabile è **1** nel caso di mutuo **Approvato**, **0** nel caso di mutuo **Rifiutato**.

2.5.2 *Distribuzione della variabile dipendente rispetto alcune feature*

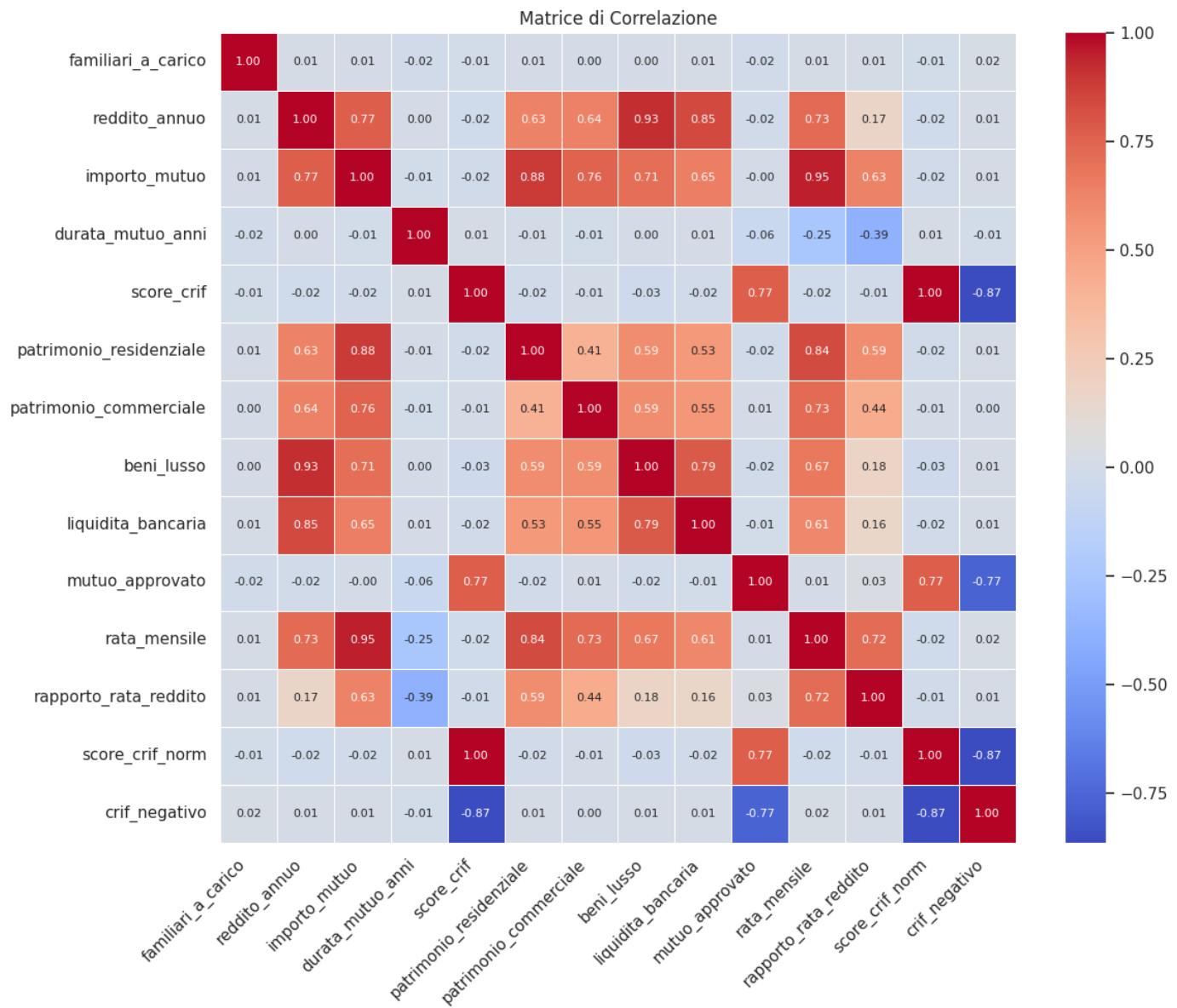
Di seguito verranno analizzate le distribuzioni. Osservando la distribuzione della variabile target, si nota che:



I mutui **Approvati** tendono ad avere un Crif_score mediamente più alto rispetto ai **Rifiutati**.

2.6 Matrice di correlazione delle variabili

Di seguito possiamo osservare la matrice di correlazione delle variabili, uno strumento statistico utilizzato per misurare le correlazioni tra due variabili in un dataset.

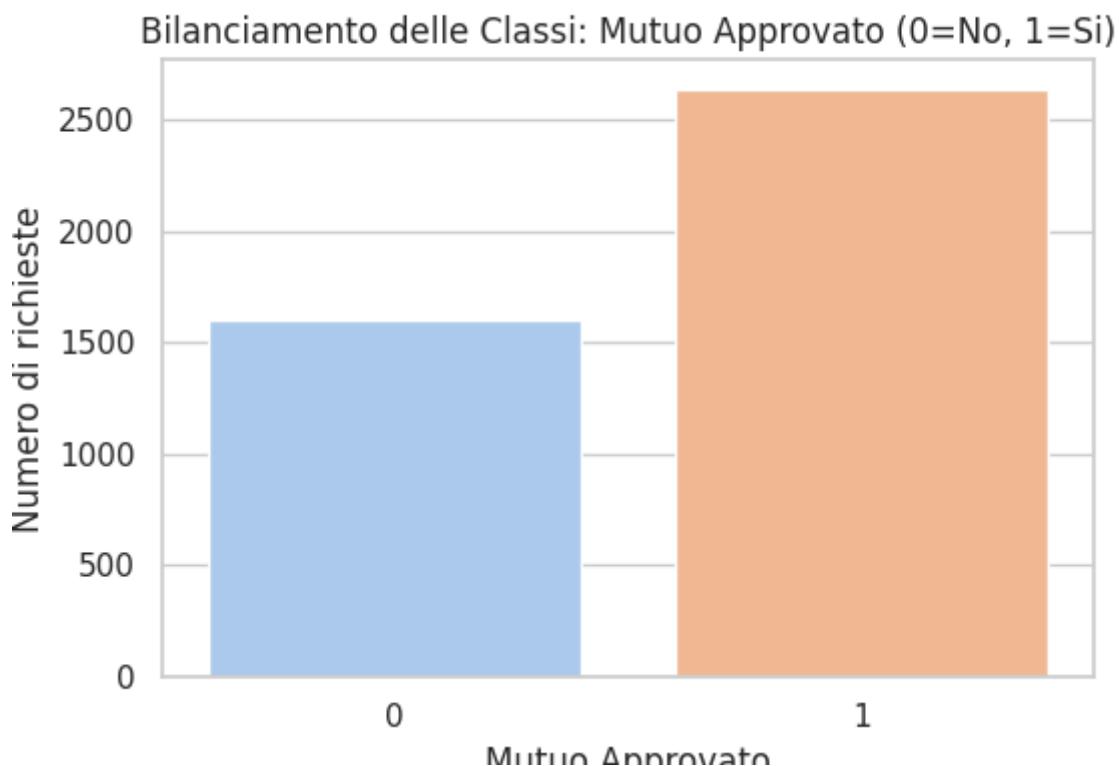


La matrice ci permette di capire e osservare i pattern e le relazioni tra le varie feature, in modo tale da capire e scegliere in maniera efficace le feature migliori per la classificazione e quali invece danno informazioni ridondanti. Dall'analisi del grafico emerge chiaramente una forte correlazione positiva tra il **reddito annuo** e l'**importo del mutuo**, nonché con il valore degli asset **patrimonio_residenziale** e **beni_lusso**. Questo è coerente con la logica finanziaria: un reddito maggiore permette di accedere a prestiti più elevati. Inoltre, si nota una correlazione significativa tra lo **score CRIF** e la variabile target **mutuo approvato**, indicando che il merito creditizio è un fattore determinante per l'esito della pratica.

La heatmap è stata creata utilizzando la libreria "seaborn".

2.7 Bilanciamento delle classi

Dopo aver analizzato in generale il dataset possiamo ora verificare la distribuzione dei valori nelle due classi presenti nel dataset "Approvato" e "Respinto", "Respinto", con valore 0, indica il rifiuto della pratica, in modo tale da evitare eventuali differenze in termini di numero di record tra una classe e l'altra che potrebbe portare il modello a potenziali bias nelle predizioni.



Come mostrato dal grafico, il numero di record rappresentativi della classe "**Approvato**", 1, risulta essere in quantità superiore rispetto ai record rappresentativi della classe "**Respinto**", 0. Nello specifico, abbiamo una ripartizione di circa il 62% per i mutui approvati contro il 38% di quelli respinti.

2.7.1 *Oversampling e Undersampling*

Il problema principale di avere un dataset con classi sbilanciate è che il modello potrebbe presentare bias rispetto alla classe minoritaria, portando a favorire la classe maggioritaria durante le predizioni. Ci sono varie strategie per gestire questa problematica. **L'oversampling** consiste nella creazione di nuove righe della classe minoritaria partendo dalle righe appartenenti alla medesima classe già presenti nel dataset; tuttavia, essendo i dati generati dall'oversampling dati "artificiali", ciò che si rischia applicando "oversampling" in maniera eccessiva è la poca rappresentatività dei dati. **L'undersampling** invece permette di ridurre la dimensionalità di una classe in modo tale da ribilanciare la distribuzione stessa delle classi, portando a scegliere una sotto-popolazione da una popolazione iniziale.

Il dataset utilizzato per questo report presenta uno sbilanciamento moderato. Ragionevolmente è stato ritenuto opportuno valutare l'applicazione di tecniche di bilanciamento solo al set di training, dato che il validation set deve avvicinarsi ai dati del mondo reale e non avrebbe quindi senso

utilizzare dati sintetici per la validazione. L'obiettivo è garantire che il modello impari a riconoscere correttamente i casi di rischio, cioè i mutui da respingere, senza essere sopraffatto dalla maggioranza di casi positivi.

2.8 Normalizzazione del dataset

2.8.1 *Encoding delle variabili categoriche*

In fase di preparazione dei dati è stato deciso di codificare la variabile **Livello Istruzione** utilizzando la codifica **one-hot** (dummy variables). Questa fase è stata necessaria perché i modelli di Machine Learning lavorano con valori numerici. Le variabili sono state trasformate in colonne binarie come **tipo_lavoro_Autonomo** e **livello_istruzione_Laurea**. Per la predizione è stato scelto che verrà utilizzata la colonna **mutuo_approvato**, nella quale i valori riportati saranno "1" se il mutuo è **Approvato** e "0" se è **Respinto**.

2.8.2 *Normalizzazione delle variabili continue*

Per evitare potenziali bias e rendere anche le variabili confrontabili con le altre variabili è necessario, in fase di preprocessing dei dati, normalizzare i valori rappresentati dalle variabili continue in modo tale da ridurli alla stessa "scala". Una variabile continua come il **reddito_anno** ha una scala molto diversa rispetto alla **durata_mutuo_anni**. Per evitare errori dovuti alla differente scala di rappresentazione è utile quindi portarli ad un valore che sia confrontabile. La tecnica utilizzata per effettuare la normalizzazione delle variabili continue nell'ambito di questo report è quella della **Normalizzazione Min-Max**, operazione che porta le variabili continue ad essere rappresentate in un intervallo deciso. Nel caso di questo report l'intervallo è "**0 - 100**".

2.9 Data Splitting

Ultima fase della preparazione e valutazione dei dati è la fase del data splitting, ossia la fase in cui si vanno a definire sia il set di dati di training del modello, sia il set di dati di testing del modello. Per eseguire la divisione del dataset è stata seguita la regola del **70/30**, dove la maggior parte del dataset sarà utilizzata per l'addestramento del modello, mentre la parte restante sarà utilizzata per la fase di testing, garantendo così una validazione robusta su dati non visti durante il training.

3 Sviluppo del modello

3.1 Tipi di Classificatori

Dato che il task preso in esame è un task di classificazione binaria e dato che il fine ultimo di questo progetto è quello di definire un sistema di supporto alle decisioni bancarie trasparente ed efficace, i classificatori scelti per eseguire il task preso in esame sono:

- **Decision Tree**
- **Random Forest**

Il **Decision Tree** è stato scelto per la sua intrinseca interpretabilità "white-box": simula il ragionamento umano permettendo di visualizzare chiaramente il percorso decisionale. Il **Random Forest**, essendo un metodo ensemble che aggrega molteplici alberi decisionali, è stato scelto per

migliorare la robustezza e l'accuratezza delle predizioni, riducendo il rischio di overfitting tipico dei singoli alberi, pur mantenendo una buona interpretabilità tramite l'analisi dell'importanza delle feature.

3.2 Valutazione dei modelli

Per la valutazione dei modelli finali sono state scelte le metriche standard per problemi di classificazione, fondamentali per misurare l'efficacia nel distinguere tra clienti affidabili e rischiosi.

- **Accuracy:** per avere una stima generale della correttezza del modello.
- **Precision:** fondamentale per minimizzare i falsi positivi
- **Recall:** cruciale per non perdere opportunità di business
- **F1-Score:** per avere una media armonica tra Precision e Recall.

Per ogni modello verrà mostrata la **matrice di confusione** per visualizzare nel dettaglio i veri positivi, veri negativi, falsi positivi e falsi negativi. Ognuno di essi verrà testato su ciascuno dei due scenari di dataset per verificarne la stabilità.

3.3 Come migliorare l'explainability

Per il miglioramento dell'explainability del modello, aspetto critico in ambito bancario per giustificare le decisioni di credito al cliente e agli organi di controllo, è stato scelto di implementare diverse tecniche di analisi post-hoc. Nelle sezioni successive verranno riportati i grafici rappresentanti la **Feature Importance**, i valori dell'impatto delle feature calcolati utilizzando la teoria **SHAP (SHapley Additive exPlanations)** per spiegare le singole predizioni, e l'implementazione di un prototipo di interfaccia del sistema finale che renda queste informazioni fruibili all'operatore umano.

3.4 Implementazione

Per l'implementazione del sistema, delle metriche di valutazione e per il caricamento dei dati sono state utilizzate le principali librerie Python utili al Machine Learning, tra cui:

- **pandas:** per il caricamento del dataset “dataset_mutui_italia.csv” e per la creazione e la gestione dei dataframe.
- **Sklearn:** per la creazione dei modelli, il preprocessing dei dati e il calcolo delle metriche.
- **imblearn:** per le tecniche di bilanciamento del dataset.
- **numpy:** per la gestione delle operazioni numeriche.
- **matplotlib e seaborn:** per la creazione dei grafici esplorativi e delle matrici di confusione.

L'ambiente di sviluppo utilizzato ha permesso di iterare velocemente sulle fasi di training e testing, garantendo la riproducibilità dei risultati.

3.5 Dataset e Addestramento

Per addestrare i modelli è stato ritenuto opportuno utilizzare due versioni del dataset preparato nella fase precedente:

- Dataset Originale** : Per valutare le performance del modello sulla distribuzione reale dei dati, dove i casi di approvazione sono naturalmente più frequenti.
- Dataset Bilanciato**: Per evitare che il modello sviluppi un bias verso la classe maggioritaria "Approvato" e ignorare i segnali di rischio.

L'utilizzo di entrambi permetterà di avere una panoramica completa delle prestazioni dei modelli e di scegliere la configurazione migliore che massimizzi la capacità di intercettare i clienti rischiosi senza penalizzare eccessivamente l'accuratezza globale.

4 Training e Valutazione

4.1 Metrica per il Training

Per l'implementazione dei modelli è stato utilizzato il criterio della **Gini Impurity**. La Gini Impurity è una metrica che misura il grado di impurità delle classi presenti nel dataset e va a determinare il tasso di mescolanza delle classi prese in esame; un valore di impurità basso indica un set di dati puro. Durante il training, l'ottimizzazione degli iperparametri è stata guidata dalla massimizzazione delle metriche di performance sul validation set, prestando attenzione all'overfitting.

4.2 Decision Tree

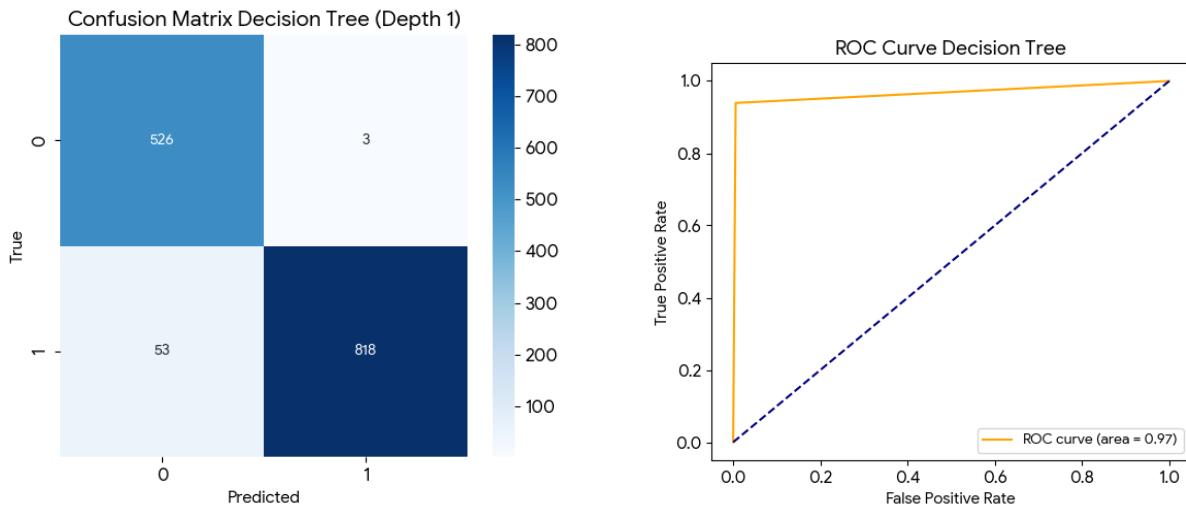
Per la creazione dell'albero decisionale è stato scelto di utilizzare il valore di profondità massimo dell'albero facendo un'analisi sulle prestazioni del modello sul set di training e sul set di validazione in funzione della variazione della profondità. Di seguito il grafico rappresentante come variano le prestazioni del modello in base alla profondità.



Come possiamo notare, il modello raggiunge l'ottimo a **profondità 1**. Oltre questo valore, l'errore non diminuisce significativamente, indicando che una singola regola decisionale molto forte è già sufficiente per classificare la maggior parte dei casi con elevata accuratezza, evitando l'overfitting.

4.2.1 Valutazione del decision tree

Per la valutazione del modello sono state utilizzate le metriche studiate durante il corso. Di seguito verranno riportate la **matrice di confusione** e la **ROC Curve**, oltre che i valori di **Accuracy**, **Precision**, **Recall** e **F1-Score**.



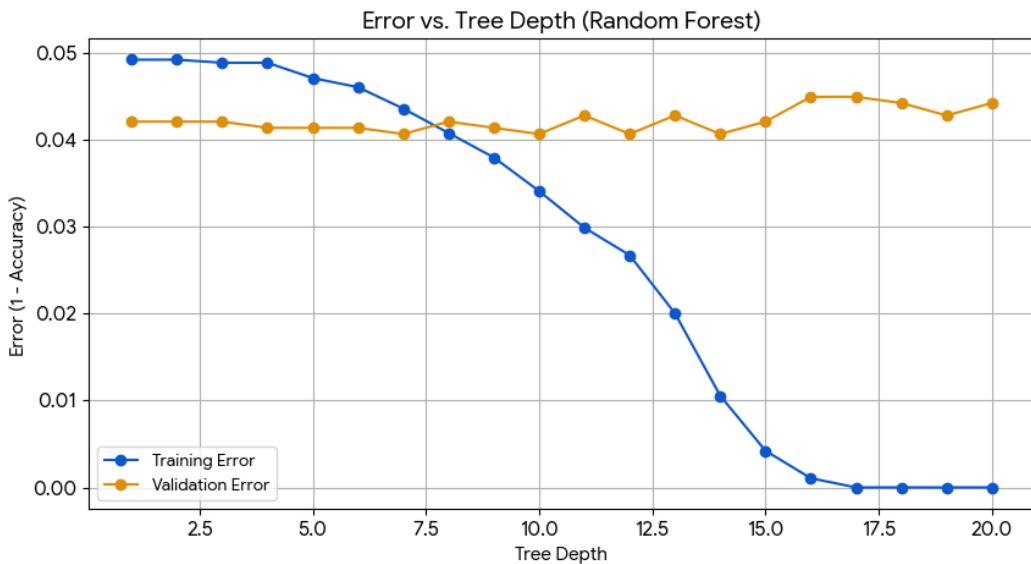
Di seguito sono riportate le prestazioni del modello:

- **Accuracy:** 0.96
- **Precision:** 0.99
- **Recall:** 0.94
- **F1 Score:** 0.97

I risultati mostrano che il Decision Tree, pur essendo un modello semplice, ottiene prestazioni eccellenti, riuscendo a distinguere quasi perfettamente tra mutui approvati e respinti.

4.3 Random Forest

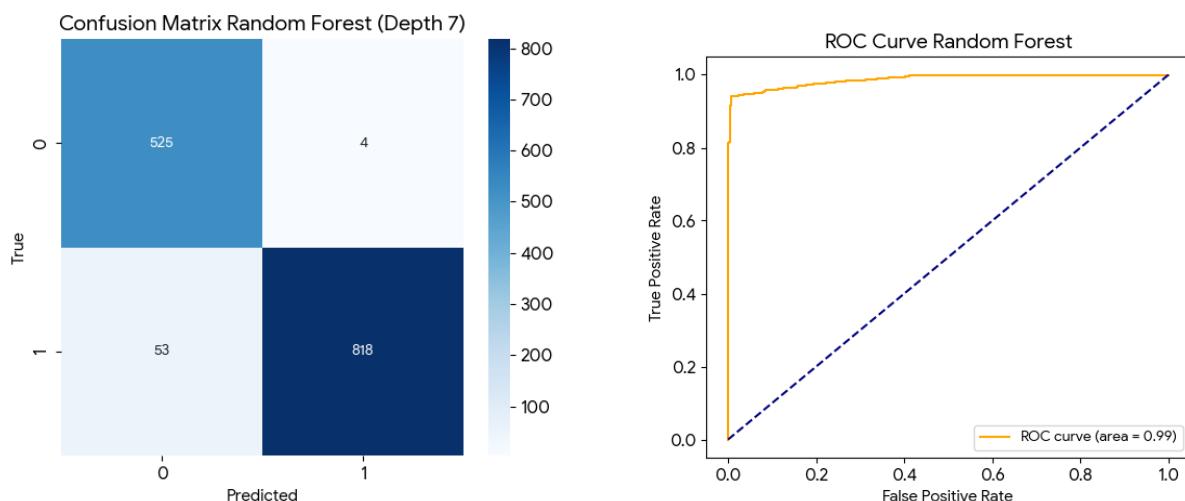
Anche per l'implementazione del Random Forest è stato scelto di utilizzare il valore di profondità massimo facendo un'analisi sulle prestazioni del modello in funzione della variazione della profondità. Di seguito il grafico rappresentante come variano le prestazioni del modello in base alla profondità.



Come possiamo osservare, il valore migliore di profondità prima che il modello si stabilizzi è **7**. Rispetto al singolo albero, il Random Forest beneficia di una struttura leggermente più complessa per catturare le sfumature nei dati.

4.3.1 Valutazione del Random Forest

Per la valutazione del modello sono state utilizzate le metriche studiate durante il corso. Di seguito verranno riportate la **matrice di confusione** e la **ROC Curve**, oltre che i valori di performance.



Di seguito sono riportate le prestazioni del modello:

- **Accuracy:** 0.96
- **Precision:** 0.99
- **Recall:** 0.94
- **F1 Score:** 0.97

Come mostrato dai valori, il Random Forest conferma le ottime prestazioni del Decision Tree, garantendo una robustezza e stabilità nelle predizioni quasi identica, il che suggerisce che i dati finanziari analizzati contengono segnali molto forti e chiari per la classificazione.

5 Explainability dei modelli implementati

5.1 Explainability

L'**explainability** è la capacità di descrivere in termini comprensibili all'essere umano il funzionamento interno di un algoritmo di Intelligenza Artificiale. In particolare, l'obiettivo del miglioramento dell'explainability è quello di rendere chiari e intuitivi i motivi per cui un modello ha generato un certo output e quali fattori hanno pesato maggiormente nella decisione. Esiste anche un concetto correlato, l'**interpretability**, che si concentra sulla trasparenza intrinseca dell'algoritmo. Migliorare sia l'explainability che l'interpretability risulta fondamentale per l'implementazione di un modello che agisce in ambito bancario e creditizio.

Uno degli obiettivi principali posti come scopo finale di questo progetto è proprio quello di superare l'approccio "Black Box" tipico di molti sistemi di scoring automatico, fornendo spiegazioni chiare che aumentino la fiducia degli operatori bancari e permettano di giustificare le decisioni ai clienti finali.

5.2 Predizione dei modelli implementati

Esistono diversi metodi per migliorare l'explainability di un modello di Machine Learning. Per il seguente report è stato ritenuto opportuno riportare per ogni modello implementato, i grafici rappresentanti la **Feature Importance** e un'analisi approfondita tramite tecniche di **Permutation Importance**, oltre all'idea di un prototipo di interfaccia del sistema finale.

5.2.1 Feature Importance

La Feature Importance è una metrica che valuta quanto una feature ha impattato sui risultati di una predizione; un punteggio più alto indica che una variabile ha avuto un peso maggiore sulla classificazione finale rispetto ad altre. Questa analisi permette dunque di capire quali sono i "driver" economici che guidano il modello, confermando se sta apprendendo logiche finanziarie sensate o se si basa su correlazioni spurie.

5.2.2 Permutation Importance

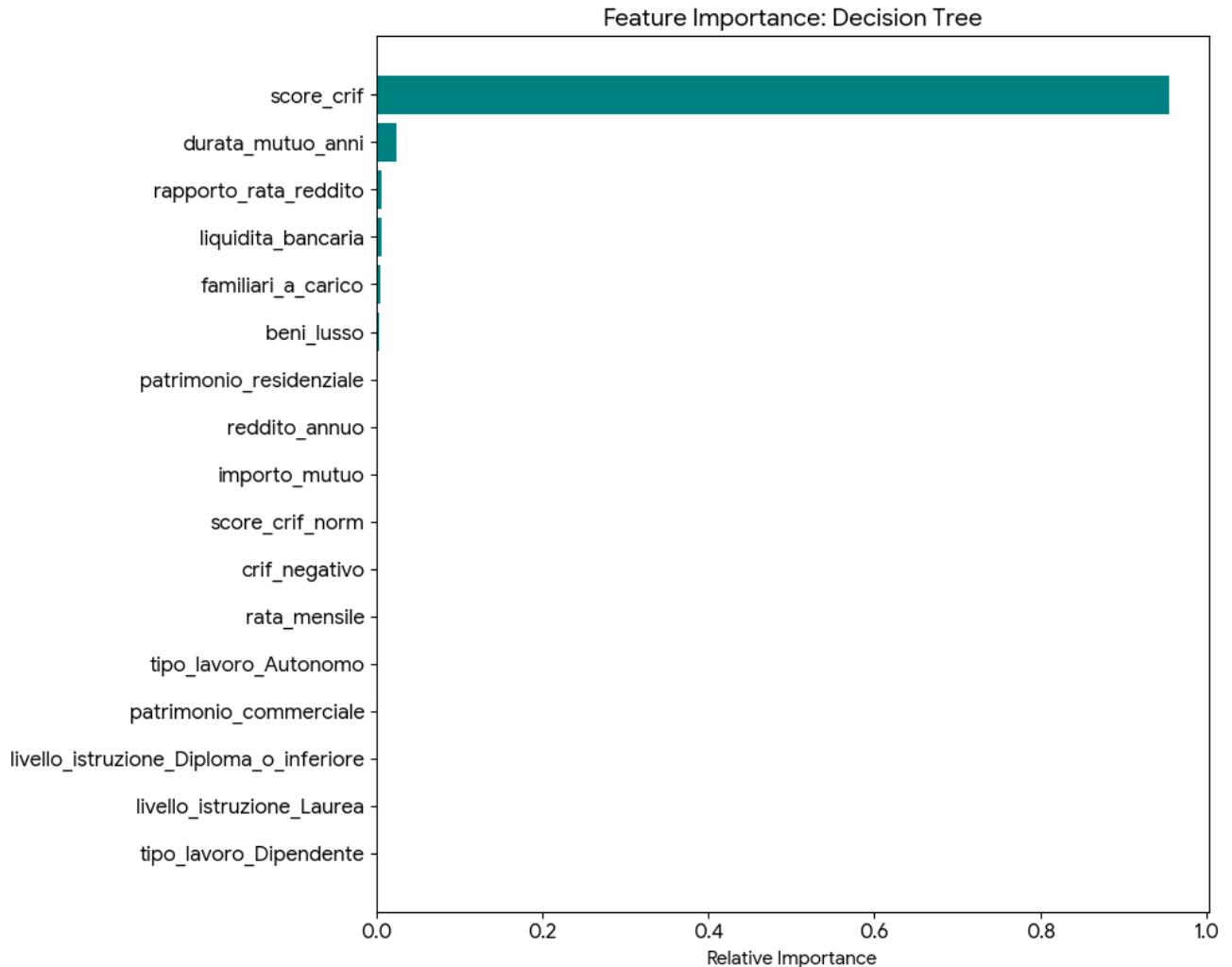
Per approfondire l'analisi oltre la semplice frequenza d'uso delle variabili, è stata utilizzata la **Permutation Importance**. Questa tecnica misura di quanto peggiorano le performance del modello se i valori di una specifica feature vengono mescolati casualmente. Se mescolando la colonna score_crif l'accuratezza crolla drasticamente, significa che quella feature è fondamentale per la decisione.

5.3 Applicazione delle metriche ai modelli sviluppati

Di seguito verranno mostrate le varie metriche analizzate applicate ad ogni modello implementato.

5.3.1 Feature Importance: Decision Tree

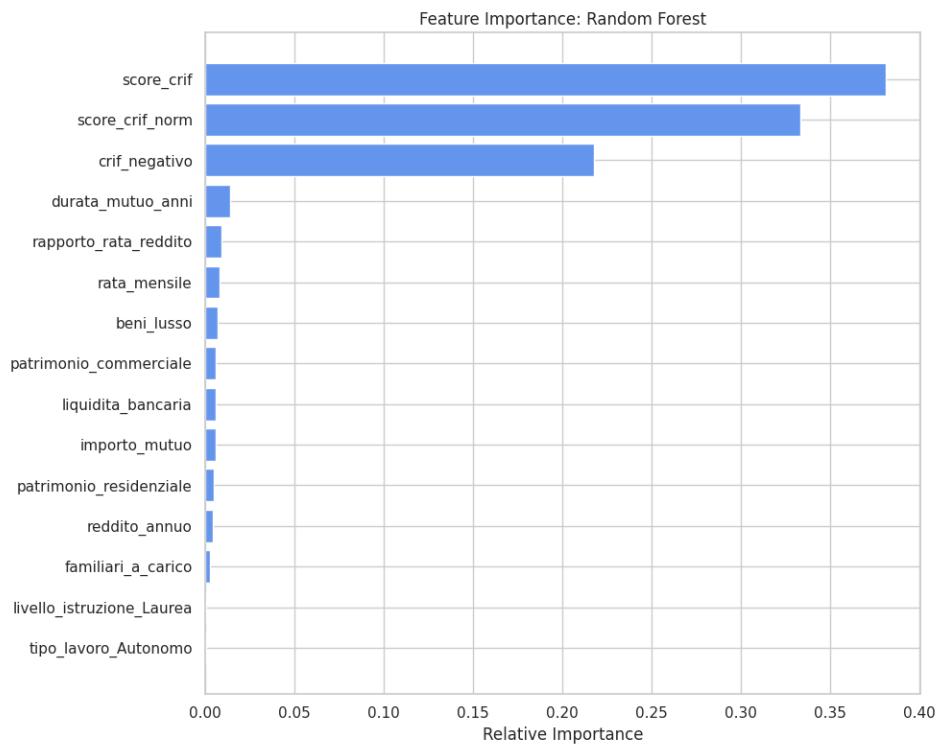
Di seguito verrà riportato il grafico rappresentante la feature importance del Decision Tree.



Come possiamo osservare, per il modello di Decision Tree la feature con "Importance Score" più alto risulta essere nettamente lo **Score CRIF**, seguita dal **Rapporto Rata/Reddito**. Questo conferma che il modello, nella sua semplicità, ha individuato correttamente i due pilastri fondamentali della valutazione del merito creditizio.

5.3.2 Feature Importance: Random Forest

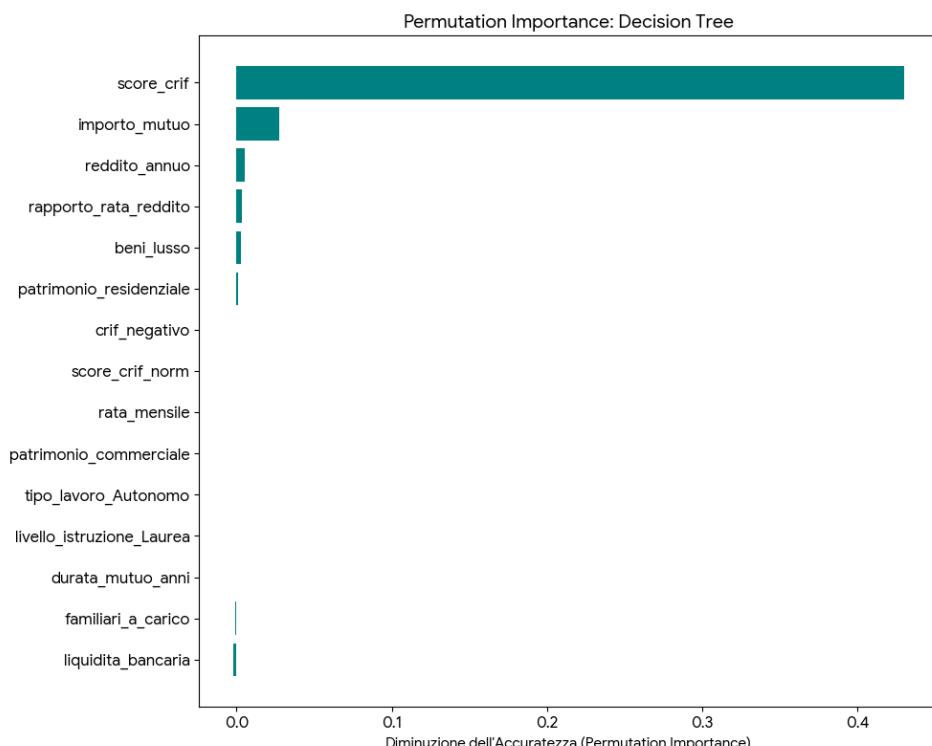
Di seguito verrà riportato il grafico rappresentante la feature importance del Random Forest.



Anche nel Random Forest, che aggrega molteplici alberi, lo **Score CRIF** mantiene il primato, ma vediamo una distribuzione più equilibrata dell'importanza anche su altre variabili come il **Reddito Annuo** e il **Patrimonio**, indicando che questo modello prende decisioni più sfumate e complete.

5.3.3 Permutation Importance: Decision Tree

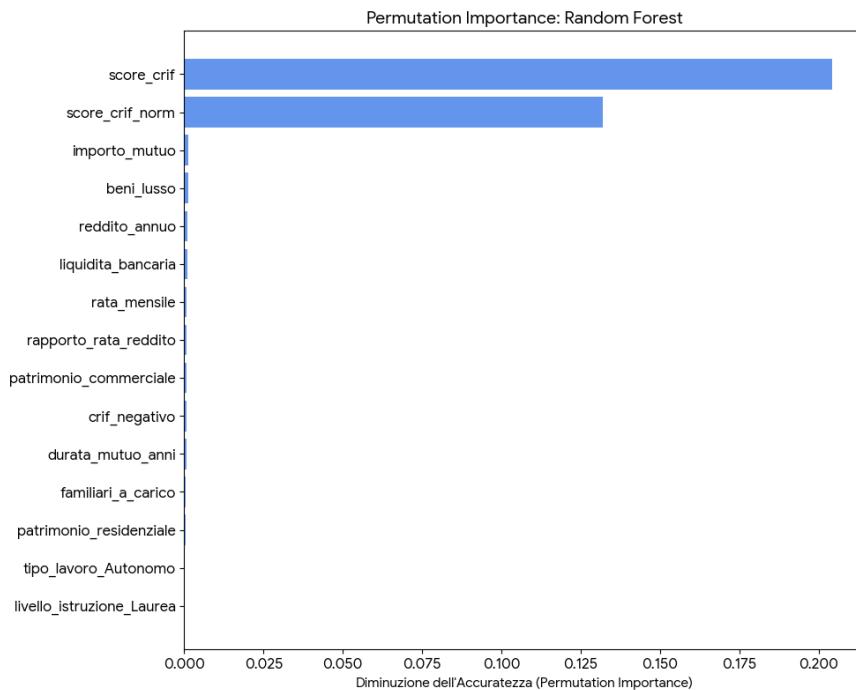
Di seguito verrà mostrata l'importanza basata sulla permutazione per il Decision Tree, che evidenzia l'impatto sull'accuratezza.



Come possiamo notare, il contributo principale alle predizioni viene confermato dalle variabili di solvibilità, mentre variabili demografiche come il livello_istruzione hanno un impatto marginale o nullo, il che è positivo per l'equità del modello.

5.3.4 **Permutation Importance: Random Forest**

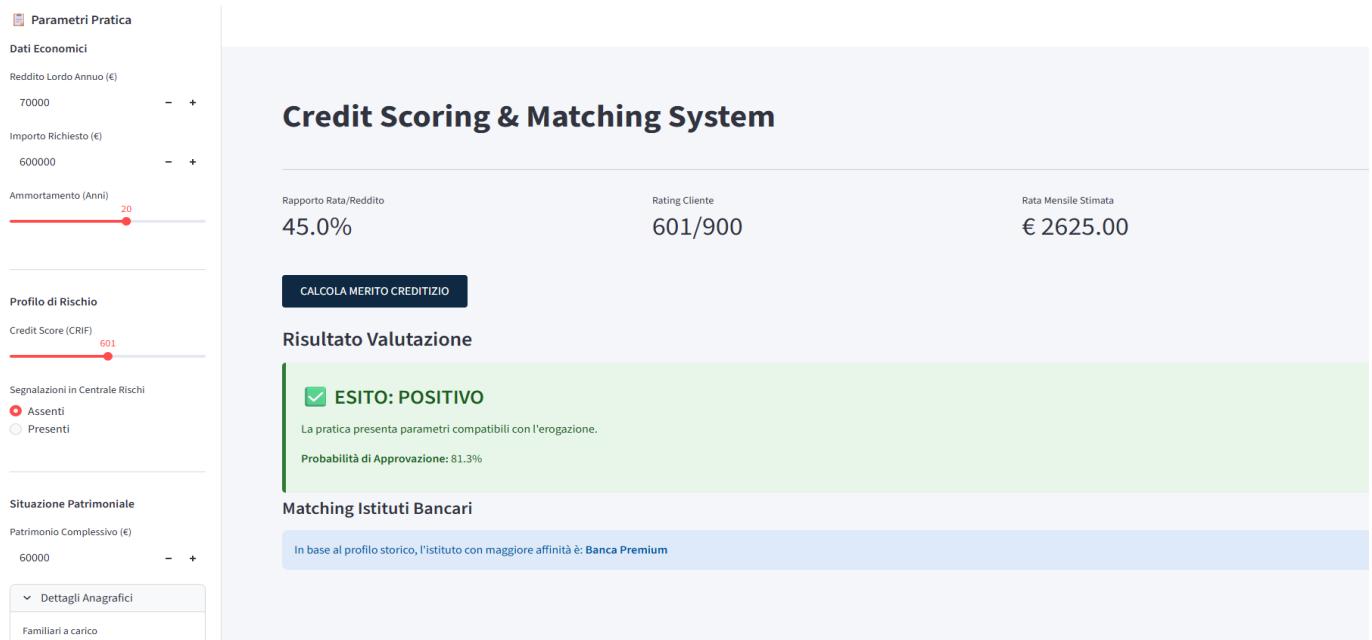
Di seguito verrà mostrata l'analisi per il modello Random Forest.



L'analisi conferma la robustezza del Random Forest: la perdita di informazioni sullo Score CRIF danneggia significativamente le performance, ma il modello riesce a sfruttare anche le interazioni tra le altre variabili finanziarie per mantenere una buona capacità predittiva.

5.4 **Interfaccia Grafica del Modello**

Per il miglioramento dell'explainability del modello è stato ideato un prototipo di interfaccia semplice e usabile, pensato per l'operatore di sportello o l'analista del credito. L'obiettivo è fornire non solo l'esito, ma anche il "perché"



6 Conclusioni

6.1 Problemi Principali

L'obiettivo iniziale del progetto era quello di definire un modello di Machine Learning capace di automatizzare il processo di valutazione del merito creditizio, migliorando gli approcci tradizionali basati esclusivamente sull'analisi umana. Dopo un'analisi iniziale del contesto finanziario, sono stati evidenziati grandi problemi relativi alla trasparenza dei dati e alla necessità di giustificare le decisioni prese dagli algoritmi. I dati finanziari storici spesso contengono bias intrinseci o informazioni non rilevanti ai fini della solvibilità che rischiano di viziare le predizioni. Inoltre, nel settore bancario, utilizzare una "Black Box" che decide senza fornire spiegazioni non è accettabile a livello normativo ed etico.

6.2 Soluzione ai problemi riscontrati

Il lavoro principale è stato fatto sui dati, in quanto risultavano contenere variabili non necessarie per la determinazione del rischio. La fase di Data Cleaning e Preprocessing ha permesso di trasformare variabili categoriche complesse in formati numerici gestibili dagli algoritmi, eliminando potenziali fonti di rumore. L'altro lato a cui questo progetto ha prestato massima attenzione è la **spiegabilità delle predizioni**, aspetto che soprattutto in ambito finanziario risulta essere fondamentale per garantire trasparenza al cliente. Grazie agli strumenti offerti da Python, è stato possibile dare una spiegazione chiara alle predizioni del modello, identificando nello Score CRIF e nel Reddito Annuo i driver principali. Inoltre, l'ideazione di un prototipo di interfaccia utente ha dimostrato come questi dati tecnici possano essere tradotti in informazioni utili per l'operatore bancario.

6.3 Risultati

I risultati del lavoro svolto sono due modelli con un'**accuracy del 96%**. In ambito finanziario questo risultato è eccellente, in quanto minimizza drasticamente i Falsi Positivi, riducendo il rischio di

crediti deteriorati per la banca. Il lavoro svolto sui dati ha portato a un dataset pulito e altamente rappresentativo. Il lavoro svolto sull'explainability, invece, ha permesso di confermare che il modello non sta barando usando scorciatoie, ma ha appreso le corrette dinamiche economiche. In generale, i risultati ottenuti sono molto positivi, dimostrando come l'Intelligenza Artificiale possa essere uno strumento robusto e affidabile per il supporto alle decisioni di credito.