UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO | DIPARTIMENTO DI INFORMATICA

E-Commerce_Fraud_Detection

Documentazione sul caso di studio

Ingegneria della Conoscenza AA 2025-2026

Realizzato da:

Paolantonio Di Grassi, 758320, p.digrassi2@studenti.uniba.it

Giovanna Antida De Pascale, 758317, g.depascale4@studenti.uniba.it

Repository:

E-Commerce_Fraud_Detection

Indice

Introduzione

Nel contesto attuale del commercio elettronico, la sicurezza delle transazioni online costituisce un aspetto cruciale per aziende e consumatori. L'elevato numero di operazioni digitali, unito alla varietà dei metodi di pagamento e all'assenza di interazione fisica tra le parti, rende i sistemi e-commerce particolarmente esposti a fenomeni di frode. Diventa quindi fondamentale individuare in modo tempestivo transazioni sospette, al fine di ridurre le perdite economiche e garantire l'affidabilità delle piattaforme online.

Il progetto "*E-Commerce Fraud Detection*" ha come obiettivo la realizzazione di un sistema di analisi dei dati per la rilevazione automatica delle frodi nelle transazioni di e-commerce. Il sistema si basa su una pipeline strutturata di preprocessing, analisi dei dati ed addestramento di modelli di *Machine Learning supervisionato* su un dataset reale di transazioni, con lo scopo di individuare pattern sospetti e stimare la probabilità di frode di ogni transazione.

Il progetto presenta un modulo di ragionamento logico basato su Prolog, utilizzato per applicare regole esperte e fornire spiegazioni interpretabili delle decisioni prese dal sistema.

L'obiettivo finale del progetto è dimostrare come un approccio integrato, che combina tecniche statistiche, modelli di apprendimento automatico e ragionamento logico, possa migliorare l'efficacia e l'affidabilità dei sistemi di rilevazione delle frodi nel dominio e-commerce.

Requisiti funzionali

Per poter eseguire il progetto è necessario che siano installati Python e SWI-Prolog e che siano disponibili le seguenti librerie:

- *Pandas*: gestione dei dataset
- *Numpy*: elaborazione numerica
- *Scikit-learn*: modelli di Machine Learning e preprocessing
- *Imbalanced-learn*: gestione dello sbilanciamento delle classi
- *PySWIP*: integrazione di Prolog all'interno di un software Python
- *Matplotlib*: visualizzazione grafica delle prestazioni dei modelli di ML

Installazione e avvio

Nella directory principale del progetto è presente un file *requirements.txt*, contenente l'elenco completo delle dipendenze necessarie per l'esecuzione del progetto. L'installazione può essere effettuata eseguendo il seguente comando dal prompt: *pip install -r requirements.txt*.

Dopo essersi assicurati di aver configurato correttamente Python, SWI-Prolog e le librerie necessarie, è sufficiente aprire il progetto con l'IDE e avviare il software dal file main.py.

Dataset

Il dataset utilizzato è un dataset sintetico di transazioni e-commerce progettato per simulare in modo realistico scenari di frode, mantenendo al contempo la completa anonimizzazione dei dati e l'assenza di informazioni sensibili reali. Il dataset è composto da circa 300.000 transazioni effettuate da 6.000 utenti univoci, ciascuno dei quali compie mediamente tra 40 e 60 operazioni, e presenta un forte sbilanciamento delle classi (circa il 2% di transazioni fraudolente).

Le feature presenti nel dataset sono le seguenti:

- *transaction_id*: Identificativo univoco della transazione. Viene utilizzato esclusivamente per fini di tracciamento e non contiene informazione predittiva.
- *user_id*: Identificativo univoco dell'utente che ha effettuato la transazione.
- *account_age_days*: Età dell'account utente in giorni.
- *total_transactions_user*: Numero totale di transazioni effettuate dall'utente fino a quel momento.
- *avg_amount_user*: Importo medio delle transazioni precedenti dell'utente.
- *amount*: Importo della transazione. È una delle variabili più informative, poiché transazioni fraudolente tendono spesso a presentare importi anomali o elevati.
- *country*: Paese in cui è registrato il commerciante o in cui avviene la transazione.
- *bin_country*: Paese di emissione della carta di pagamento, ricavato dal BIN (Bank Identification Number). Rende sospetta una transazione se è avvenuta in un paese diverso da quello di emissione della carta.
- *channel*: Canale attraverso cui viene effettuata la transazione.
- *merchant_category*: Categoria delle merci del commerciante.
- *promo_used*: Indica se è stato utilizzato un codice promozionale.
- *avs_match*: Esito del controllo AVS sull'indirizzo di fatturazione.
- *cvv_result*: Esito del controllo CVV della carta.
- *three_ds_flag*: Indicatore dell'utilizzo del protocollo 3-D Secure.
- *transaction_time*: Timestamp completo della transazione.
- *shipping_distance_km*: Distanza geografica tra il luogo del commerciante e l'indirizzo di spedizione. Distanze molto alte possono essere sospette.
- *is_fraud*: Indica se la transazione è una frode o meno.

Preprocessing

La fase di preprocessing consente di trasformare il dataset grezzo *ecommerce_fraud.csv* di transazioni e-commerce in una forma strutturata e coerente con le esigenze sia dei modelli di Machine Learning sia dei successivi controlli rule-based.

- Pulizia e trasformazione delle variabili temporali:
  Il dataset originale include la variabile *transaction_time*, che rappresenta il momento in cui la transazione è stata effettuata. L'uso diretto del timestamp completo risulta poco efficace per modelli supervisionati su dati tabellari. Per questo l'informazione temporale viene trasformata estraendo l'ora della transazione, dando origine alla feature *transaction_hour*. Questa trasformazione consente di catturare pattern temporali rilevanti, come la maggiore probabilità di frode in determinate fasce orarie.

- Creazione feature *high_amount*:
  A partire dalla variabile numerica *amount* viene introdotta la feature binaria *high_amount*, che segnala le transazioni caratterizzate da un importo particolarmente elevato. La soglia utilizzata non è definita a priori, ma viene definita in modo adattivo sulla base della distribuzione dei dati, utilizzando un percentile elevato. Questo permette di rendere la feature robusta in presenza di outlier, evitando assunzioni arbitrarie sugli importi considerati "anomali".

- Eliminazione delle variabili non predittive:
  La feature *transaction_id* viene esclusa poiché non contiene informazioni utili ai fini della classificazione, prevenendo l'introduzione di rumore. Analogamente, la variabile *user_id* e il timestamp originale *transaction_time* non vengono utilizzati nelle fasi successive di addestramento, in quanto l'informazione rilevante è già stata catturata attraverso feature più informative.

- Creazione di un dataset ridotto:
  Considerando il forte sbilanciamento tra la classe maggioritaria e quella minoritaria (*is_fraud*), viene costruita una versione ridotta del dataset. In questa fase vengono mantenute tutte le transazioni fraudolente (*is_fraud = 1*) e solo un sottoinsieme di transazioni legittime (*is_fraud = 0*) pari a 25000. Questa operazione consente di ridurre il costo computazionale delle fasi di addestramento e sperimentazione, preservando l'informazione critica legata alla classe di interesse.

- Preprocessing differenziato delle feature:
  Prima della fase di addestramento, le feature vengono preprocessate in modo distinto in base alla loro tipologia. Le variabili numeriche, come *amount* e *transaction_hour*, vengono normalizzate per eliminare effetti dovuti a scale diverse e favorire la stabilità dei modelli. Le variabili categoriche vengono trasformate tramite *one-hot encoding*, generando una colonna binaria (colonna dummy) per ciascuna categoria osservata nel training. L'opzione *handle_unknown="ignore"*, garantisce robustezza operativa durante il testing, dato che in presenza di categorie non viste durante il training, evita errori, codificandole implicitamente come vettori nulli nel blocco one-hot.

- Suddivisione dei dati e prevenzione del data leakage:
  Il dataset ridotto viene suddiviso in un insieme di training e uno di test, mediante split stratificato, preservando la distribuzione della variabile target *is_fraud*. Il test set viene mantenuto completamente separato e utilizzato esclusivamente per la valutazione finale delle prestazioni e per l'integrazione con il modulo rule-based. Le trasformazioni di preprocessing non vengono applicate globalmente prima dello split, ma sono apprese esclusivamente sui dati di training all'interno della pipeline, garantendo una corretta separazione tra dati di addestramento e dati di valutazione.

- Applicazione dell'oversampling nella pipeline di addestramento:
  Per gestire lo sbilanciamento delle classi, viene applicata la tecnica dell'oversampling casuale (*Random Oversampling*). È stata utilizzata questa tecnica dato che è coerente con il problema e con il progetto, in quanto over-campiona la classe minoritaria scegliendo campioni a caso con ripetizione, generando così un dataset più bilanciato dal punto di vista della distribuzione delle classi [1]. In questo progetto l'oversampling è integrato direttamente nella pipeline di addestramento ed eseguito all'interno di ciasun fold della cross validation. Questa scelta progettuale consente di preservare la correttezza metodologica del progetto, evitando che informazioni provenienti dai dati di validazione influenzino la fase di addestramento.
  È stato utilizzato *RandomOverSampler* in questo modo per diversi motivi:
    - *Sbilanciamento marcato*: con una percentuale di frodi molto bassa un modello addestrato senza correzioni tende a privilegiare *is_fraud=0*, ottenendo una accuracy apparentemente alta ma scarsa capacità di intercettare frodi.
    - *Semplicità e controllo*: il random oversampling non introduce punti sintetici nello spazio delle feature ma replica esempi reali, risultando più prevedibile come comportamento (utile soprattutto quando le feature includono diverse categoriche codificate con one-hot).

    - *Valutazione corretta*: mantenere il test set non alterato permette di misurare metriche in condizioni che riflettono l'operatività reale del sistema, evitando stime ottimistiche.

- Preparazione dei dati per l'integrazione con il sistema rule-based:
  In parallelo alla pipeline di preprocessing utilizzata dai modelli di Machine Learning, viene mantenuta una rappresentazione *raw* del test set, contenente le feature originali della transazione. Questo permette di collegare le predizioni del modello a informazioni direttamente interpretabili e di applicare regole logiche basate su attributi semantici della transazione, permettendo di affiancare alle predizioni del modello di Machine Learning controlli basati su regole esplicite.

Apprendimento supervisionato

L'apprendimento supervisionato è una tecnica di Machine Learning in cui un modello viene addestrato utilizzando un insieme di dati etichettati, nei quali a ogni osservazione è associato un valore noto della variabile target. Questo paradigma comprende principalmente due tipologie di problemi: la regressione, utilizzata per la previsione di valori continui, e la classificazione, impiegata per l'assegnazione di categorie discrete. [2]

Nel progetto sviluppato, il problema di rilevazione delle frodi è formulato come un *compito di classificazione supervisionata binaria*, in cui l'obiettivo è distinguere tra transazioni fraudolente e non fraudolente. Non sono state considerate tecniche di regressione, in quanto la variabile target *is_fraud* assume esclusivamente valori discreti e non continui.

Classificazione

La classificazione è il processo di identificazione della categoria a cui appartiene una nuova osservazione, basandosi su un insieme di dati di addestramento che contengono osservazioni già etichettate.

In questo progetto sono stati confrontati cinque modelli di classificazione per valutare quale sia il più efficiente per il problema in esame. I modelli considerati sono:

- Logistic Regression.
- Decision Tree.
- Random Forest.
- K-Nearest Neighbors (KNN).
- Multi-Layer Perceptron (MLP).

Per ciascun modello, la fase di addestramento e selezione degli iperparametri è stata condotta mediante cross validation stratificata a 10 fold, utilizzando come metrica di ottimizzazione la ROC-AUC. Tale approccio consente di ottenere una stima più robusta delle prestazioni, riducendo la dipendenza da un singolo split dei dati. Per tutti i modelli, tranne MLP, è stata usata come ottimizzazione degli iperparametri la *GridSearchCV*, per MLP, caratterizzato da una maggiore complessità e dimensionalità degli iperparametri, è stata usata la *RandomizedSearchCV.* Le curve ROC mediate sui 10 fold forniscono inoltre una rappresentazione grafica (realizzata con l'uso di Matplotlib) della stabilità delle prestazioni rispetto alle diverse partizioni dei dati, evidenziando la variabilità inter-fold e la robustezza dei modelli.

Tutti i modelli producono in output una stima probabilistica dell'appartenenza alla classe fraudolenta. Tuttavia, la decisione finale non è basata unicamente sulla soglia standard pari a 0.5. In particolare, per il modello che ha ottenuto le migliori prestazioni in termini di ROC-AUC, ovvero il Random Forest, è stata condotta una fase di tuning esplicito della soglia decisionale, con l'obiettivo di massimizzare l'F1-score della classe minoritaria e ad analizzare il compromesso tra capacità di intercettare frodi e contenimento dei falsi positivi.

Infine, l'output probabilistico del Random Forest è stato integrato all'interno di un modulo di ragionamento logico sviluppato in Prolog. Tale modulo combina la probabilità stimata del modello con regole esperte basate su attributi semantici delle transazioni. Questo permette alla probabilità ML di non rappresentare una decisione finale, ma di contribuire alla costruzione di uno score di rischio pesato, dal quale vengono derivati livelli di rischio discreti e azioni operative, realizzando un approccio ibrido che unisce apprendimento statistico e conoscenza simbolica.

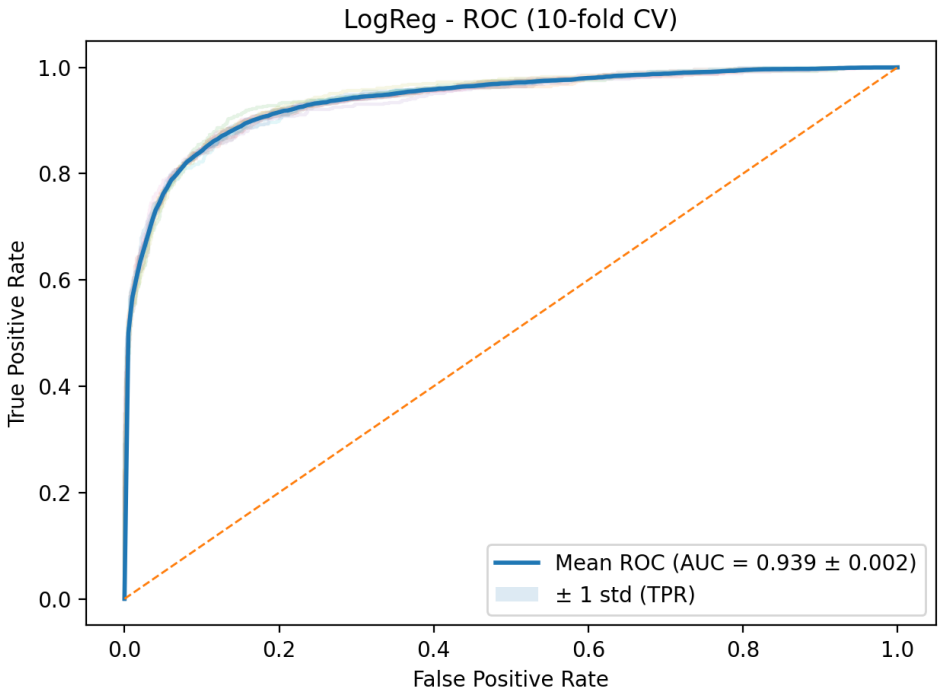
Logistic Regression

La Logistic Regression è un modello che stima la probabilità di appartenenza a una classe mediante una funzione logistica applicata a una combinazione lineare delle feature. Rappresenta un baseline lineare per il problema di classificazione binaria.

Iperparametri migliori rilevati con la Cross Validation:

- *C* = 0.01 - Parametro che controlla il compromesso tra adattamento ai dati e penalizzazione dei pesi. Un valore basso impone una penalizzazione più forte sui coefficienti, riducendo il rischio di overfitting in presenza di feature correlate e rumorose.
- *class_weight* = None - Indica che, nel contesto del dataset ridotto e dell'oversampling applicato nella pipeline, il modello riesce a gestire lo sbilanciamento senza ulteriori correzioni esplicite.

Curva ROC-AUC:



La curva ROC media mostra una buona capacità discriminativa, con ROC-AUC = 0.939±0.002 e una bassa variabilità tra i fold, a conferma della stabilità del modello.

Metriche risultato sul test set ottenute con i migliori iperparametri:

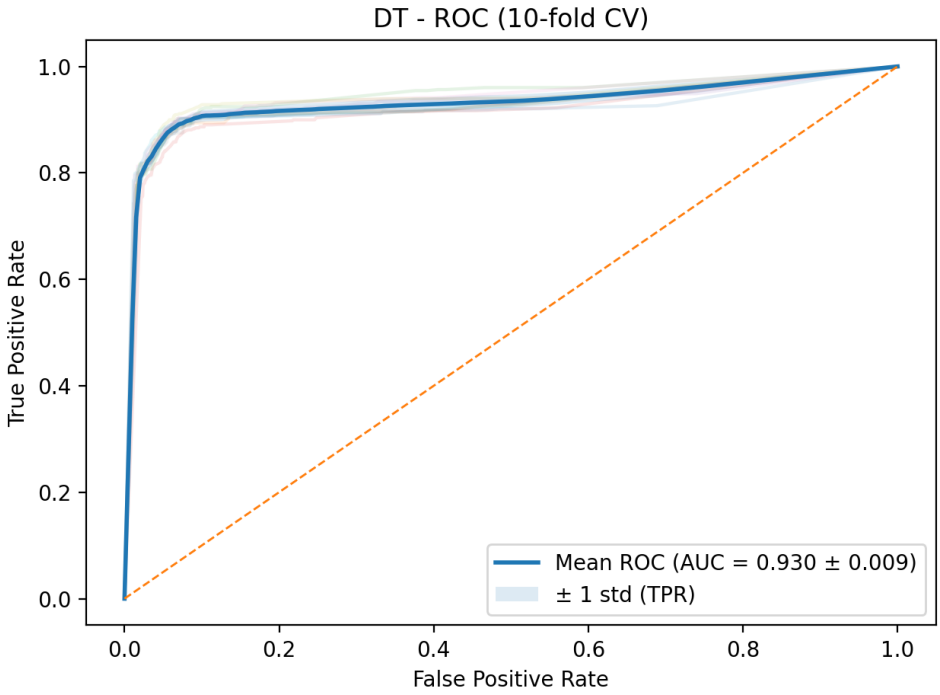
- *Accuracy* = 0.8845 ± 0.0124
- *Precision* = 0.6792 ± 0.0267
- *Recall* = 0.8496 ± 0.0317
- *F1-score* = 0.7549 ± 0.0256
- *ROC AUC* = 0.9403 ± 0.0148

Decision Tree

Il Decision Tree è un modello che utilizza una struttura ad albero per fare previsioni. L'algoritmo costruisce l'albero suddividendo i dati in base a condizioni su variabili, in modo da massimizzare l'informazione guadagnata ad ogni livello.
Iperparametri migliori rilevati con la Cross Validation:

- *max_depth* = 15 – Questo iperparametro limita la profondità massima dell'albero.
- *min_samples_leaf* = 5 – Il minimo numero di campioni che devono essere presenti in un nodo foglia.
- *class_weight* = None – In questo caso, il modello non applica pesi particolari per le classi.

Curva ROC-AUC:



Il modello ha ottenuto un AUC di 0.930 ± 0.009, il che indica che il modello è in grado di distinguere efficacemente tra le classi (frodi e non frodi). La curva si avvicina rapidamente all'angolo superiore sinistro, che rappresenta il miglior comportamento del modello.

Metriche risultato sul test set ottenute con i migliori iperparametri:

- *Accuracy* = 0.9225 ± 0.0107
- *Precision* = 0.7831 ± 0.0294
- *Recall* = 0.8707 ± 0.0315
- *F1-score* = 0.8246 ± 0.0238
- *ROC AUC* = 0.9283 ± 0.0119

Random Forest

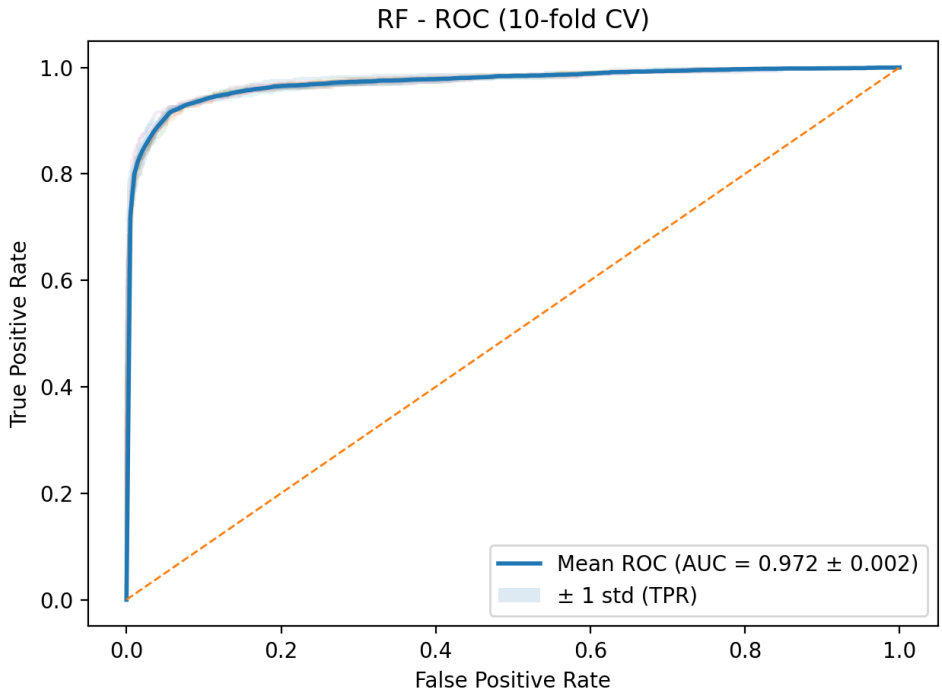
La Random Forest è un ensemble di alberi decisionali, dove ogni albero viene addestrato su una porzione diversa dei dati e con un set di caratteristiche casuali. Questo approccio riduce il rischio di overfitting e migliora la generalizzazione, combinando le previsioni di diversi alberi.

Iperparametri migliori rilevati con la Cross Validation:

- *n_estimators* = 100 – Numero di alberi nella foresta. Un numero maggiore tende a migliorare la performance del modello ma pesa a livello computazionale.

- *max_depth* = 20 – Limita la profondità massima di ciascun albero. Questo valore aiuta a prevenire l'overfitting mantenendo gli alberi abbastanza semplici.
- *max_features* = "sqrt" – L'algoritmo seleziona un sottoinsieme casuale delle caratteristiche a ogni split. L'uso della radice quadrata aiuta a ridurre la correlazione tra gli alberi e migliora la diversità.
- *min_samples_leaf* = 4 – Ogni foglia dell'albero deve contenere almeno 4 campioni, il che aiuta a generalizzare meglio il modello.
- *Class_weight* = "balanced_subsample" – Pesa le classi in modo che il modello dia maggiore attenzione alle classi sbilanciate (in questo caso la classe delle frodi).

Curva ROC-AUC:



La curva ROC AUC per la Random Forest mostra una separazione chiara tra le classi, con un valore AUC di 0.972 ± 0.002, molto vicino a 1.0, suggerendo che il modello è molto efficace nel distinguere tra transazioni legittime e fraudolente (e nel nostro caso, il migliore).

Metriche risultato (con tuned threshold = 0.6) sul test set ottenute con i migliori iperparametri:

- *Accuracy* = 0.9461 ± 0.0115
- *Precision* = 0.8818 ± 0.0261
- *Recall* = 0.8571 ± 0.0385
- *F1-score* = 0.8693 ± 0.0294
- *ROC AUC* = 0.9753 ± 0.0087

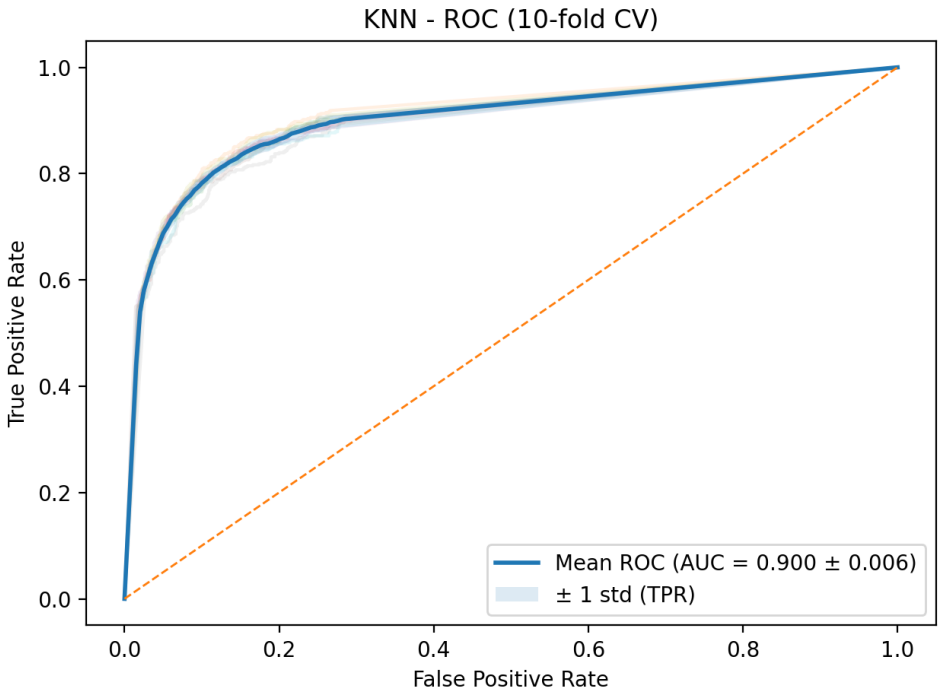
K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) è un algoritmo basato sulla similitudine. Quando viene fatta una previsione, KNN cerca i k vicini più prossimi del punto di dati e assegna la classe più comune tra questi vicini.

Iperparametri migliori rilevati con la Cross Validation:

- *n_neighbors* = 7 – Il numero di vicini da considerare. Un valore maggiore rende il modello più generalista, mentre valori più bassi tendono a sovraccaricare il modello con rumore.
- *weights* = "distance" – I pesi assegnati ai vicini sono inversamente proporzionali alla loro distanza. I vicini più vicini avranno quindi un peso maggiore nella decisione.
- *metric* = "minkowski" – La metrica utilizzata per calcolare la distanza tra i punti. Minkowski è una generalizzazione della distanza Euclidea e della distanza di Manhattan.

Curva ROC-AUC:



La curva ROC-AUC per il modello KNN mostra una buona separazione. Il valore AUC 0.900 ± 0.006, sebbene buono, è inferiore rispetto a Random Forest e Logistic Regression.

Metriche risultato sul test set ottenute con i migliori iperparametri:

- *Accuracy* = 0.8591 ± 0.0183
- *Precision* = 0.6256 ± 0.0379
- *Recall* = 0.8133 ± 0.0475

- *F1-score* = 0.7072 ± 0.0348
- *ROC AUC* = 0.9040 ± 0.0192

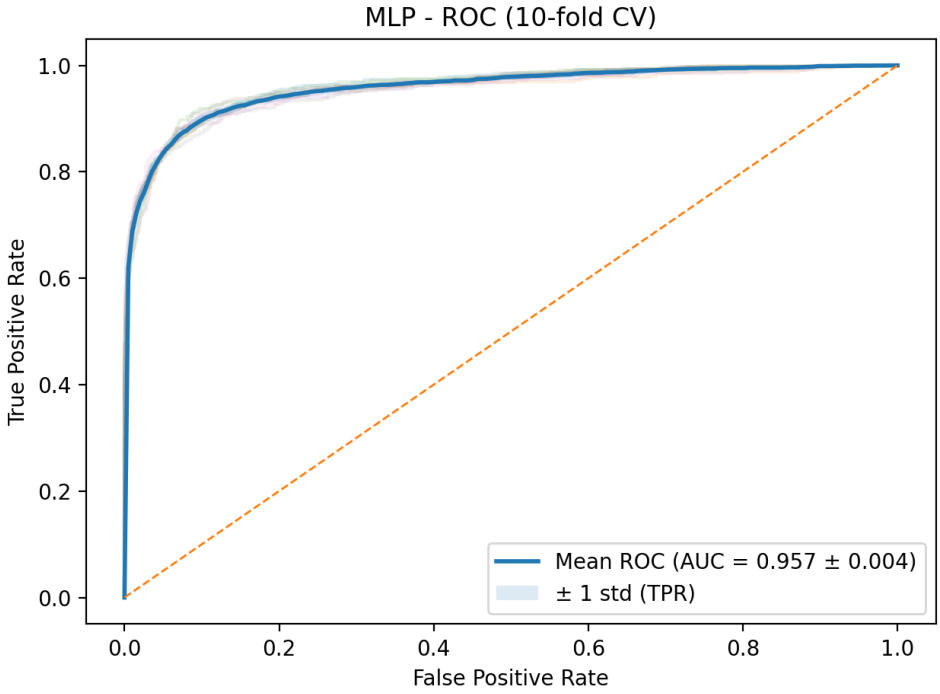
Multi-Layer Perceptron (MLP)

Il Multi-Layer Perceptron (MLP) è un modello di rete neurale feedforward in cui la presenza di uno o più strati nascosti permette di modellare relazioni non lineari tra input e output.

Iperparametri migliori rilevati con la Cross Validation:

- *learning_rate_init* = 0.001 – La velocità di apprendimento iniziale per l'algoritmo di ottimizzazione.
- *hidden_layer_sizes* = (64,) – La rete neurale ha un solo strato nascosto con 64 neuroni. Questo parametro controlla la capacità del modello.
- *alpha* = 0.001 – Un termine di regolarizzazione che penalizza le soluzioni con pesi molto grandi, prevenendo l'overfitting.

Curva ROC-AUC:



La curva ROC-AUC per l'MLP indica una buona capacità del modello di distinguere tra le classi, con un AUC di 0.957 ± 0.004, che è molto vicino a 1.0.

Metriche risultato sul test set ottenute con i migliori iperparametri:

- Accuracy = 0.9173 ± 0.0134
- Precision = 0.7639 ± 0.0321
- Recall = 0.8753 ± 0.0332
- F1-score = 0.8158 ± 0.0298
- ROC AUC = 0.9561 ± 0.0116

Confronto delle prestazioni dei modelli

Modello	*Accuracy*	*Precision*	*Recall*	*F1-score*	*ROC-AUC*
Logistic Regression	0.8845 (± 0.0124)	0.6792 (± 0.0267)	0.8496 (± 0.0317)	0.7549 (± 0.0256)	0.9403 (± 0.0148)
Decision Tree	0.9225 (± 0.0107)	0.7831 (± 0.0294)	0.8707 (± 0.0315)	0.8246 (± 0.0238)	0.9283 (± 0.0119)
Random Forest	**0.9461 (± 0.0115)**	**0.8818 (± 0.0261)**	**0.8571 (± 0.0385)**	**0.8693 (± 0.0294)**	**0.9753 (± 0.0087)**
KNN	0.8591 (± 0.0183)	0.6256 (± 0.0379)	0.8133 (± 0.0475)	0.7072 (± 0.0348)	0.9040 (± 0.0192)
MLP	0.9173 (± 0.0134)	0.7639 (± 0.0321)	0.8753 (± 0.0332)	0.8158 (± 0.0298)	0.9561 (± 0.0166)

Dall'analisi comparativa delle metriche riportate nella tabella emerge come il *Random Forest*, risulti il modello complessivamente più performante per il problema di rilevazione delle frodi e-commerce.

In particolare, il modello mostra il valore più elevato di *ROC-AUC* (0.9753), indicando un'eccellente capacità di separazione tra transazioni fraudolente e legittime. Inoltre, il *F1-score* (0.8693) evidenzia un equilibrio ottimale tra precisione e recall sulla classe di interesse, aspetto cruciale in un contesto caratterizzato da un forte sbilanciamento delle classi. Rispetto agli altri modelli, *Random Forest* combina prestazioni superiori con una deviazione standard contenuta, suggerendo una maggiore stabilità del modello rispetto alle diverse suddivisioni dei dati. Tali caratteristiche lo rendono particolarmente adatto all'integrazione con il modulo di ragionamento logico sviluppato in Prolog, dove l'output probabilistico del classificatore contribuisce alla costruzione di uno score di rischio affidabile.

Ragionamento logico

Il progetto adotta un modulo di ragionamento logico basato su Prolog, linguaggio di programmazione logica dichiarativa fondato sulla logica del primo ordine, che consente di esprimere conoscenza di dominio tramite regole e di demandare al sistema inferenziale la risoluzione delle interrogazioni [3].

Nel contesto del progetto, il ragionamento logico non viene impiegato come alternativa ai modelli di Machine Learning, bensì come componente complementare, con lo scopo di interpretare e strutturare l'informazione prodotta dalla fase di apprendimento supervisionato. In particolare, l'output probabilistico dei modelli viene integrato con ulteriori caratteristiche della transazione all'interno di un processo inferenziale che consente di individuare condizioni di rischio esplicite, difficilmente riconducibili a una singola variabile o a una semplice soglia numerica.

Knowledge Base

La Knowledge Base rappresenta il cuore del sistema rule-based ed è progettata come un Knowledge-Based System (KBS), nel quale la conoscenza di dominio è codificata sotto forma di regole logiche. La KB non è utilizzata come archivio di dati, ma come meccanismo inferenziale che consente di applicare tale conoscenza alle informazioni fornite in input dal sistema.

La rappresentazione della conoscenza si basa sulle feature originali della transazione, mantenute intenzionalmente in forma non preprocessata per preservarne il significato descrittivo. Tali feature descrivono aspetti concreti del contesto transazionale e costituiscono una base semantica adeguata alla definizione di condizioni logiche espressive e interpretabili.

Dal punto di vista computazionale, la complessità del ragionamento dipende dal numero di regole applicate a ciascuna transazione. Poiché il sistema opera su singole istanze e su un insieme limitato di regole, il costo computazionale rimane contenuto e compatibile con scenari applicativi reali.

Regole

Le regole costituiscono l'elemento centrale del modulo di ragionamento logico e rappresentano il mezzo attraverso cui la conoscenza di dominio viene applicata alle transazioni e-commerce per la rilevazione delle frodi. Esse formalizzano condizioni e scenari tipici di comportamento potenzialmente fraudolento, consentendo al sistema di combinare in modo strutturato le informazioni disponibili su una singola transazione.

Un aspetto rilevante del sistema di regole è la capacità di associare alla valutazione finale delle transazioni un insieme di motivazioni esplicite (*reasons*), che rappresentano le cause specifiche alla base della classificazione assegnata. Questo approccio rende trasparente il percorso decisionale seguito dal sistema e facilita la comprensione e la verifica delle decisioni assunte, migliorando l'interpretabilità complessiva del processo di rilevazione delle frodi.

Tipologie di regole adottate

Le regole possono essere suddivise in tre categorie principali: regole basate su singole condizioni, regole composite e regole che integrano l'output dei modelli di Machine Learning.

Ai fini della documentazione vengono discusse solo alcune delle regole implementate nella Knowledge Base, scelte per illustrare le principali logiche decisionali e i pattern di rischio modellati.

Regole basate su condizioni elementari

Una prima categoria comprende regole che modellano condizioni di rischio elementari, considerate individualmente indicative di comportamento anomalo.

Tali regole operano su singoli attributi della transazione e attivano una motivazione di rischio quando una determinata condizione risulta vera:

1 – Discrepanza tra paese della transazione e paese della carta *(country_mismatch)*

La regola *country_mismatch* viene attivata quando il paese in cui viene effettuata la transazione non coincide con il paese di emissione della carta di pagamento. Tale condizione può rappresentare un indicatore di rischio, in quanto l'utilizzo di una carta in un paese differente da quello di origine è frequentemente associato a tentativi di frode o utilizzo non autorizzato.

```
trigger(country_mismatch, Country, BinCountry, _, _, _, _, _, _, _, _) :-
    Country \= BinCountry.
```

2 – Fallimento del controllo CVV *(cvv_fail)*

La regola *cvv_fail* viene attivata quando il codice di sicurezza CVV associato alla carta non risulta valido (CVV =:= 0). Il CVV rappresenta uno dei principali meccanismi di verifica dell'autenticità della transazione; il suo fallimento costituisce pertanto un segnale di rischio significativo.

```
trigger(cvv_fail, _, _, _, _, _, _, CVV, _, _, _) :-
    CVV =:= 0.
```

3 – Distanza spedizione elevata (*far_shipping*)

La regola *far_shipping* identifica le transazioni caratterizzate da una distanza di spedizione elevata. La motivazione di rischio viene attivata quando la distanza di spedizione supera una soglia prefissata (1000 km), modellando una condizione potenzialmente anomala nel contesto delle transazioni e-commerce.

```
trigger(far_shipping, _, _, _, ShippingDist, _, _, _, _, _, _) :-
    far_shipping(ShippingDist).
```

Regole composite e pattern di rischio

Accanto alle regole elementari, il sistema include regole composite che modellano combinazioni di condizioni particolarmente critiche. Tali regole permettono di rappresentare pattern di rischio più complessi, difficilmente riconducibili a una singola variabile o a una soglia fissa:

4 – Importo elevato e assenza di 3D Secure (*no_3ds_high_amount*)

La regola composita attiva la motivazione di rischio *no_3ds_high_amount* quando una transazione classificata come di importo elevato viene eseguita senza autenticazione 3D Secure. Tale combinazione rappresenta uno scenario

particolarmente sensibile, poiché l'assenza di un meccanismo di strong customer authentication su importi elevati aumenta la probabilità di utilizzo non autorizzato.

```
trigger(no_3ds_high_amount, _, _, _, _, _, _, _, ThreeDS, HighAmount, _) :-
    HighAmount =:= 1,
    ThreeDS =:= 0.
```

5 – Distanza di spedizione elevata e controllo CVV fallito (*far_shipping_cvv_fail*)

La regola composita *far_shipping_cvv_fail* identifica transazioni potenzialmente fraudolente combinando una distanza di spedizione elevata con il fallimento del controllo CVV. Sebbene ciascuna condizione, se considerata singolarmente, possa non essere sufficiente a indicare una frode, la loro co-occorrenza rappresenta uno scenario di rischio più significativo e rafforza l'ipotesi di utilizzo non autorizzato della carta.

```
trigger(far_shipping_cvv_fail, _, _, _, ShippingDist, _, _, CVV, _, _, _) :-
    far_shipping(ShippingDist),
    CVV =:= 0.
```

Integrazione dell'output dei modelli di Machine Learning

Il sistema di ragionamento logico include specifiche regole che utilizzano direttamente l'output probabilistico prodotto dai modelli di Machine Learning. Tali regole consentono di integrare l'informazione statistica fornita dai modelli supervisionati all'interno della Knowledge Base, trattando la probabilità di frode come un ulteriore segnale di rischio.

In particolare, sono state definite due regole distinte, *ml_high* e *ml_very_high*, che attivano motivazioni di rischio differenti in base al livello della probabilità stimata:

6 – Probabilità di frode elevata stimata dal modello (*ml_high*)

La regola *ml_high* attiva una motivazione di rischio quando la probabilità di frode stimata dal modello di Machine Learning supera la soglia del 55%, indicativa di un livello di rischio significativo ma non estremo. Tale regola segnala che il modello ha individuato pattern sospetti nella transazione, pur senza raggiungere un grado di confidenza estremamente elevato. La motivazione ml_high contribuisce alla valutazione complessiva del rischio come segnale di allerta intermedio, da interpretare in combinazione con le altre evidenze disponibili.

```
trigger(ml_high, _, _, _, _, _, _, _, _, _, Proba) :-
    ml_high(Proba).
```

7 – Probabilità di frode molto elevata stimata dal modello (*ml_very_high*)

La regola *ml_very_high* viene attivata quando la probabilità di frode stimata dal modello di Machine Learning supera la soglia dell'80%, corrispondente a un livello di rischio molto elevato. Questo riflette un'elevata confidenza del modello e contribuisce in modo significativo all'aumento del punteggio di rischio della transazione.

```
trigger(ml_very_high, _, _, _, _, _, _, _, _, _, Proba) :-
    ml_very_high(Proba).
```

Input al ragionamento logico

Il modulo di ragionamento logico opera su un insieme di informazioni fornite dinamicamente per ciascuna transazione e-commerce analizzata. Tali informazioni descrivono lo stato della singola transazione in un determinato istante e includono sia le caratteristiche osservabili del contesto transazionale sia l'output dei modelli di Machine Learning, in particolare la probabilità di frode stimata.

Questi input non rappresentano conoscenza persistente, ma costituiscono dati istantanei che vengono utilizzati esclusivamente per l'analisi dell'evento corrente, riflettendo la natura event-driven del dominio applicativo. La scelta di trattare tali informazioni come input dinamici, anziché come fatti memorizzati stabilmente nella Knowledge Base, consente di mantenere il sistema flessibile e scalabile, evitando dipendenze tra eventi distinti e l'accumulo di informazioni storiche all'interno del modulo logico.

Una volta forniti al sistema, gli input vengono interpretati alla luce delle regole di dominio definite nella Knowledge Base. Il ragionamento logico utilizza tali informazioni per verificare la presenza di condizioni di rischio transazionale e per combinarle in modo strutturato.

Valutazione del rischio

La valutazione del rischio rappresenta l'esito finale del processo decisionale e deriva dall'aggregazione delle evidenze individuate attraverso il ragionamento logico. L'obiettivo è fornire una stima del rischio associato a ciascuna transazione e-commerce che non sia limitata a una classificazione binaria, ma che tenga conto della gradazione e della natura delle condizioni di rischio individuate.

Il sistema calcola inizialmente un punteggio di rischio in forma grezza, che viene successivamente normalizzato e tradotto in un livello di rischio discreto, espresso tramite classi qualitative. Tale rappresentazione consente di distinguere tra

transazioni caratterizzate da diversi gradi di criticità e di supportare decisioni operative differenziate (LEGITIMATE, REVIEW, BLOCKED).

Questo approccio garantisce coerenza tra dati osservati, conoscenza di dominio e risultato prodotto, rendendo il sistema interpretabile e controllabile.

Sviluppi futuri

L'attuale sistema implementa già una policy decisionale che associa il livello di rischio stimato dal modulo di ragionamento logico a un'azione operativa, distinguendo tra transazioni considerate legittime, transazioni soggette a revisione e transazioni da bloccare. A partire da questa impostazione, un possibile sviluppo futuro riguarda l'integrazione di tali decisioni all'interno di flussi antifrode operativi più articolati.

In particolare, il livello di rischio e le motivazioni simboliche prodotte dal sistema potrebbero essere utilizzati per attivare in modo selettivo meccanismi di verifica aggiuntiva nei casi caratterizzati da rischio intermedio. Tra questi rientrano, ad esempio, procedure di autenticazione forte come 3D Secure o sistemi di One-Time Password, nonché processi di revisione manuale supportati dalle spiegazioni fornite dal modulo logico.

Analogamente, le transazioni classificate come ad alto rischio potrebbero essere gestite attraverso politiche di blocco automatico integrate con sistemi di monitoraggio o di notifica, consentendo una risposta tempestiva agli eventi potenzialmente fraudolenti.

Un ulteriore sviluppo futuro consiste nell'introduzione di dipendenze temporali tra eventi transazionali, al fine di modellare pattern comportamentali nel tempo. In particolare, il sistema potrebbe essere esteso per considerare sequenze di transazioni riconducibili allo stesso utente o strumento di pagamento, consentendo di definire regole basate su finestre temporali, frequenza degli eventi o ripetizione di condizioni sospette (*velocity rules*). Tale estensione richiederebbe la gestione esplicita di informazioni storiche o l'integrazione con un modulo di persistenza esterno, mantenendo separata la Knowledge Base dalla gestione dello stato e preservando la chiarezza e la controllabilità del sistema.

In questo scenario, il sistema sviluppato non si limiterebbe a fornire una valutazione del rischio per singola transazione, ma costituirebbe il nucleo decisionale di una pipeline antifrode più ampia, capace di integrare valutazioni istantanee e comportamentali, mantenendo al contempo la trasparenza e la spiegabilità delle decisioni assunte.

Bibliografia

[1] Menardi, G., Torelli, N. Training and assessing classification rules with imbalanced data. *Data Min Knowl Disc* **28**, 92–122 (2014). https://doi.org/10.1007/s10618-012-0295-5

[2] IBM, “What is supervised learning?”, https://www.ibm.com/topics/supervised-learning.

[3] D. Poole and A. Mackworth, Artificial Intelligence: Foundations of Computational Agents, 3rd ed. Cambridge, UK: Cambridge University Press, [Ch.15].