CREDIT CARD FRAUD DETECTION - UNBALANCED DATA

1 Problema del Fraud Detection

Le truffe con carta di credito rappresentano una percentuale minima delle transazioni (circa 0.1%) ma i danni causati sono enormi.

È quindi di enorme interesse riuscire a riconoscere in modo accurato le transazioni fraudolente, in questo progetto vedremo diverse tecniche di apprendimento supervisionato per affrontare il problema.

2

Il Dataset

Class	Count	Percentage
0	284315	99.827
1	492	0.173

È stato utilizzato il dataset creditcard.csv, contenente 284807 transazioni registrate nell'arco di 2 giorni.

Le features del dataset sono Principal Components ottenute da una trasformazione PCA. L'importo della transazione e l'orario sono stati normalizzati.

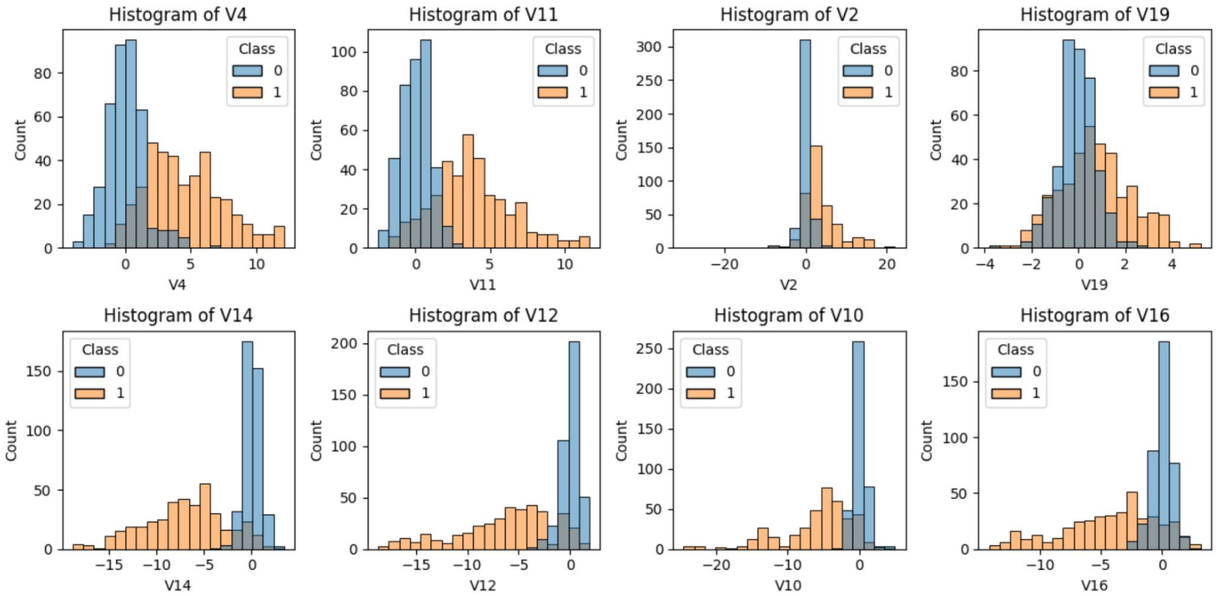
Le classi sono altamente sbilanciate con un 0.17% di transazioni fraudolente.

	Time	V1	V2	V3	V4	V5	...	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	...	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	...	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	...	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	...	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	...	-0.206010	0.502292	0.219422	0.215153	69.99	0

3 Data Pretreatment e Cross Validation

- Sono state scalate le variabili Time e Amount attraverso uno standard scaler
- Rimozione di un piccolo numero di outlier ha permesso di migliorare leggermente le prestazioni dei modelli

- È importante eseguire il resampling durante il cross validation, per non bilanciare il validation set.
  Questo per eventuale eventuale data Leakage e overfitting

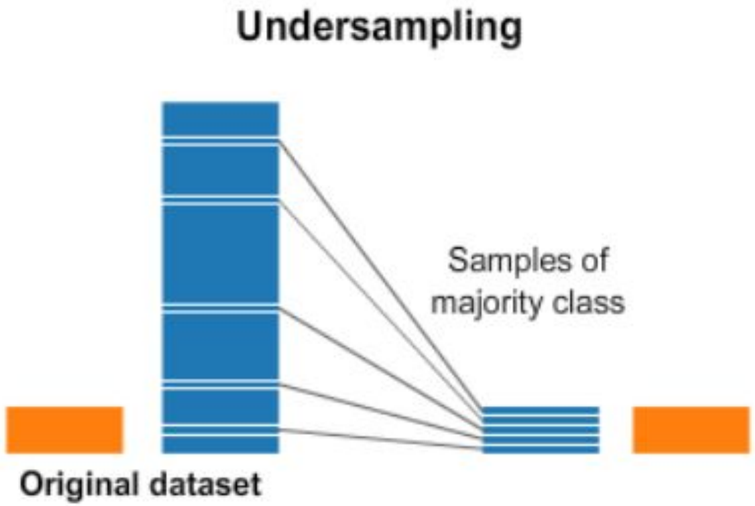


4

Random Undersample

La prima strategia vista per migliorare lo sbilanciamento delle classi è stato Random Undersample.

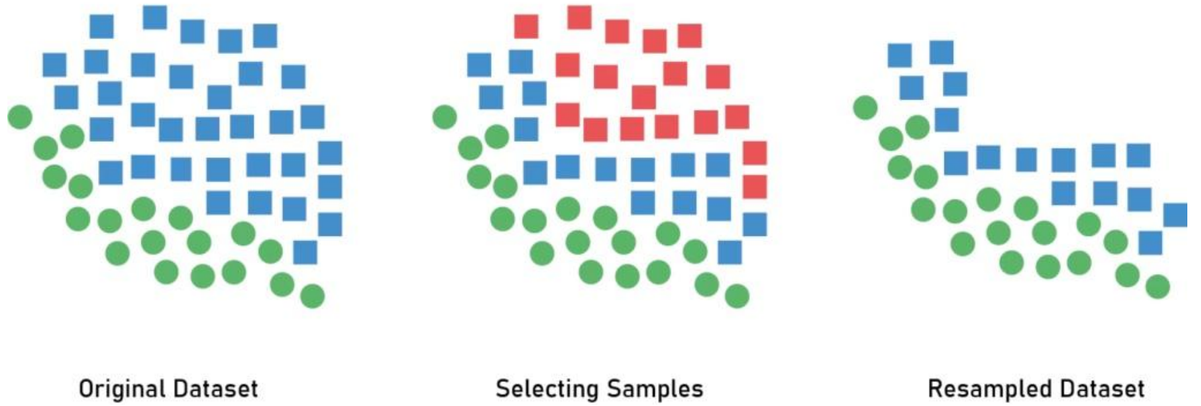
L’algoritmo consiste nel rimuovere casualmente esempi dalla classe maggioritaria, fino a raggiungere il bilanciamento.



5

Near Miss Undersampling

In NearMiss i campioni della classe maggioritaria vengono selezionati in base alla distanza dai punti della classe minoritaria, mantenendo solo quelli più vicini per ridurre lo sbilanciamento.

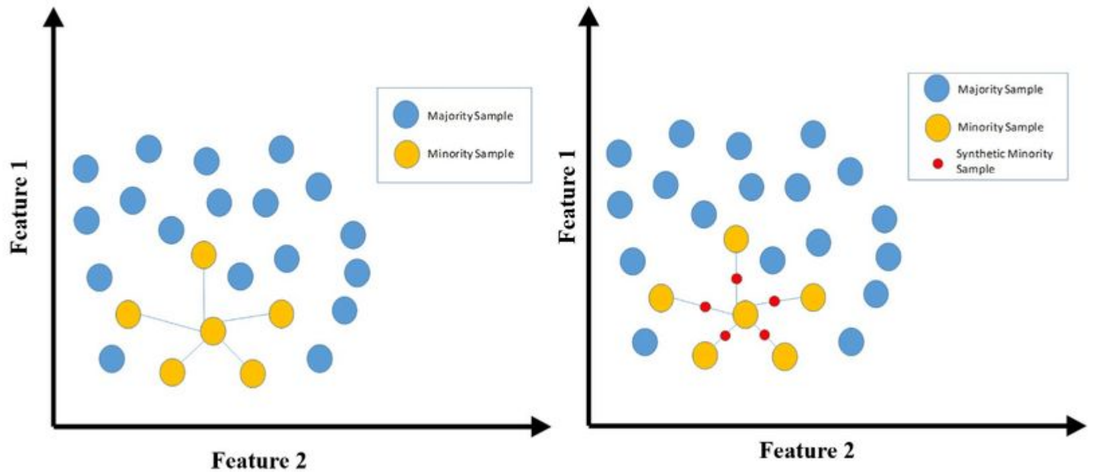


6

SMOTe Oversampling

Synthetic Minority Oversample Technique aumenta il numero di campioni nella classe “Fraud” inserendo dei dati sintetici

I nuovi dati vengono generati lungo i segmenti che collegano i nearest neighbour della classe minoritaria.



7

Metriche

A causa del forte sbilanciamento del dataset, la sola accuracy non è sufficiente per valutare le performance del modello, questa tenderà sempre ad avere un valore alto anche quando il modello non prevede correttamente nessuna osservazione della classe minoritaria.

Le seguenti metriche sono state utilizzate per la valutazione dei modelli

- F1 score
- ROC-AUC
- Precision & Recall

8

K Fold Cross Validation

Per la scelta dei parametri è stata utilizzato Stratified K Fold CV sul dataset originale, applicando l'undersample solamente ai dati di training di ogni fold.

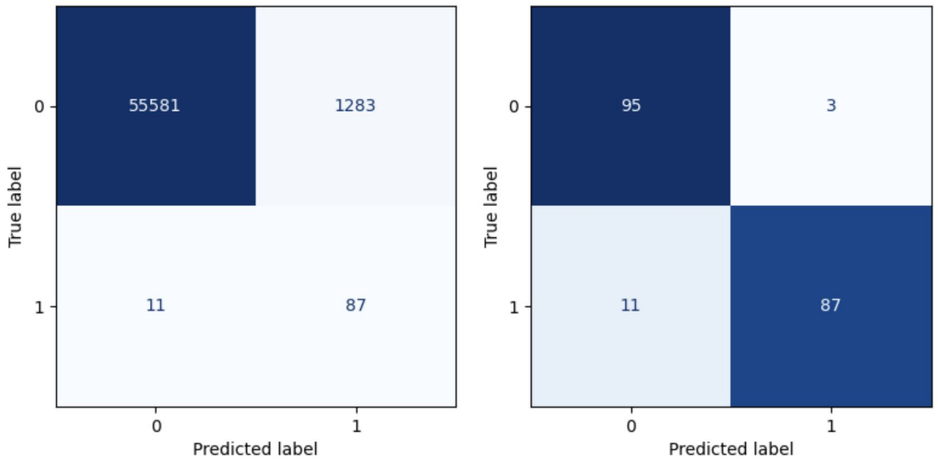
Questo per mantenere nei set di test e validazione sempre dati con una proporzione reale delle classi, ridurre l'overfitting e il data leakage.

A causa del forte sbilanciamento è stato utilizzato lo Stratified K Fold con k=5 per garantire che ogni sottoinsieme avesse punti delle classe "Fraud"

9

Regressione Logistica

(smote)



Parametri di regolarizzazione testati

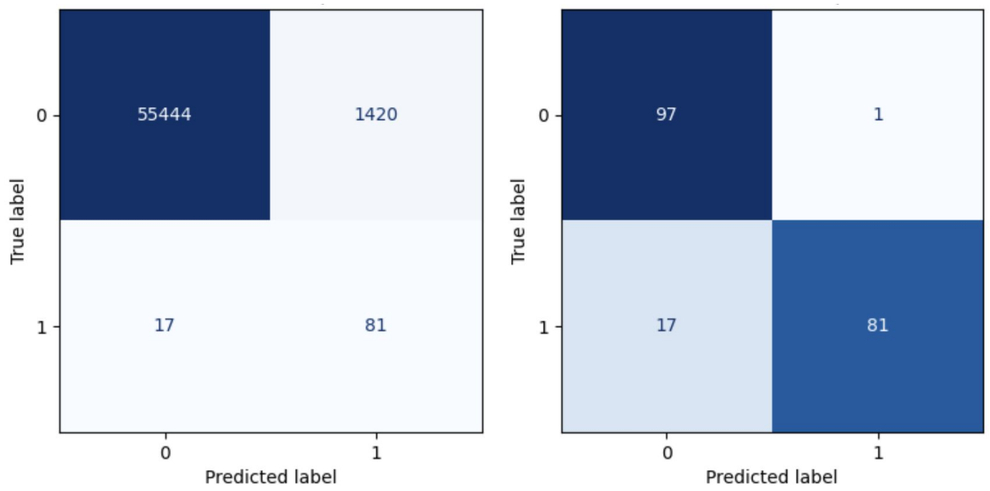
- C=0.01, 0.001
- Penalty=L2, L2

	Accuracy	Precision	Recall	f1 score	ROC-AUC score
Random Undersample	92.6	96.7	88.8	92.6	92.6
Near Miss	74.0	69.1	86.7	76.9	76.6
SMOTE Oversample	92.9	96.7	88.8	98.7	93.2

10

Decision Tree

rus

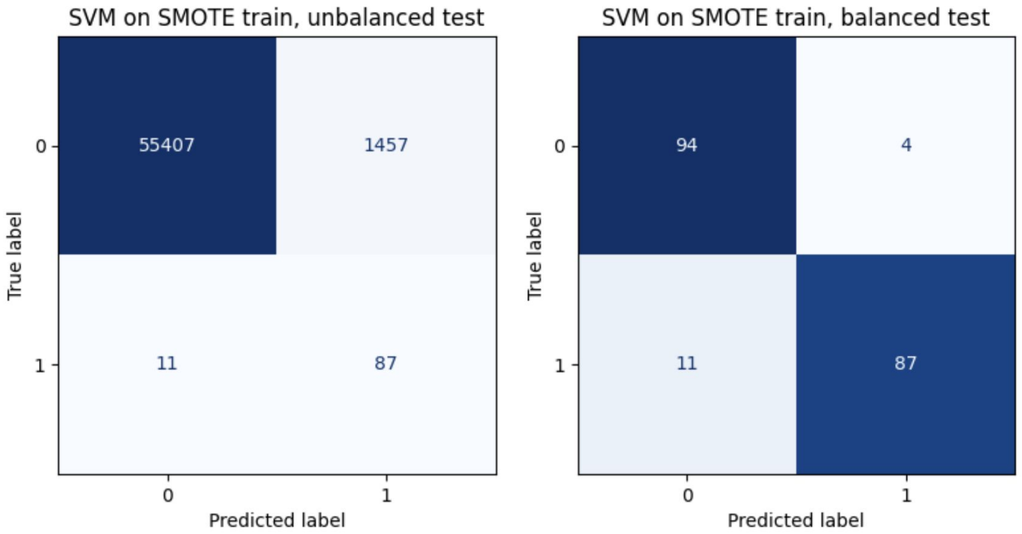


Parametri testati

- criterio = gini
- maxdepth = 2
- minsample leaf = 5

	Accuracy	Precision	Recall	f1 score	ROC-AUC score
Random Undersample	90.8	98.8	82.7	90.0	90.1
Near Miss	87.2	86.1	88.8	92.3	87.4
SMOTE Oversample	90.8	98.8	82.7	90.0	90.0

SVM Classifier



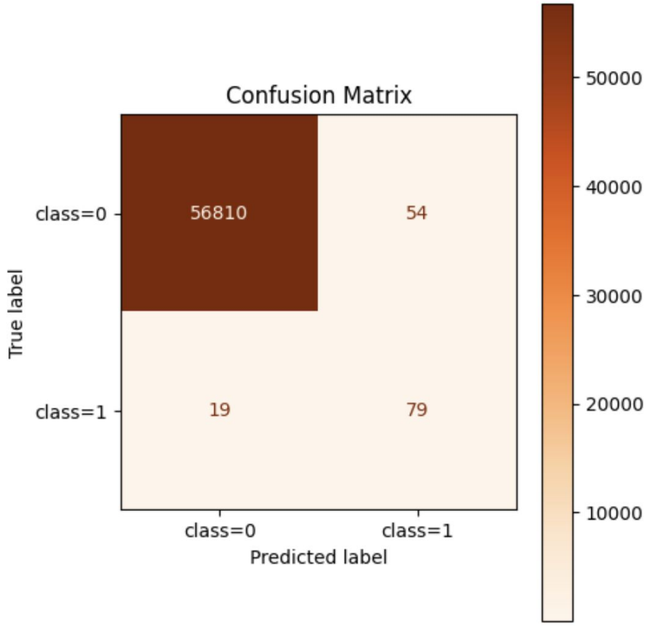
Parametri di testati

- C = 0.5
- Kernel = Liner, Poly

	Accuracy	Precision	Recall	f1 score	ROC-AUC score
Random Undersample	92.9	96.7	88.8	92.6	92.6
Near Miss	79.6	99.9	59.2	74.4	79.5
SMOTE Oversample	92.3	95.6	88.8	92.1	93.1

12

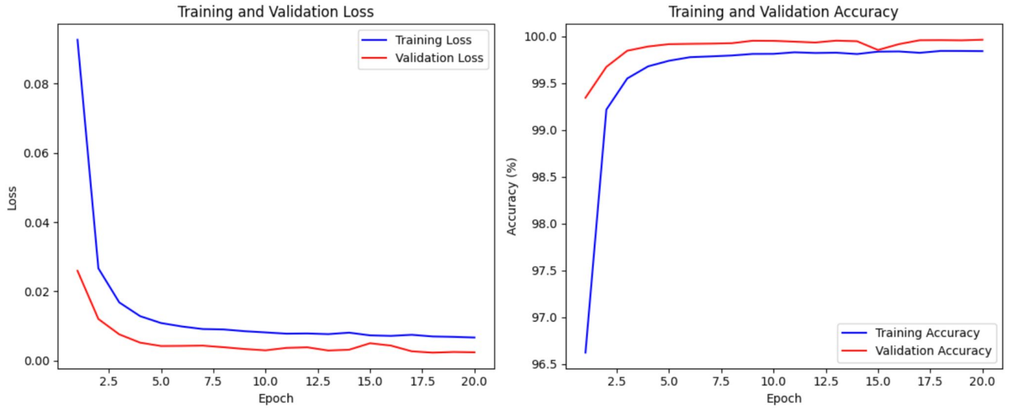
Neural Network



↓Input Layer (30 neuroni)
↓ ReLU + Dropout(0.3)
↓ Hidden Layer (32 neuroni)
↓ ReLU + Dropout(0.3)
↓ Output Layer (2 neuroni)
↓ Softmax

- Loss: Cross Entropy
- Learning rate: 0.001
- Optimizer: Adam
- Batch size: 300
- Num epochs: 20

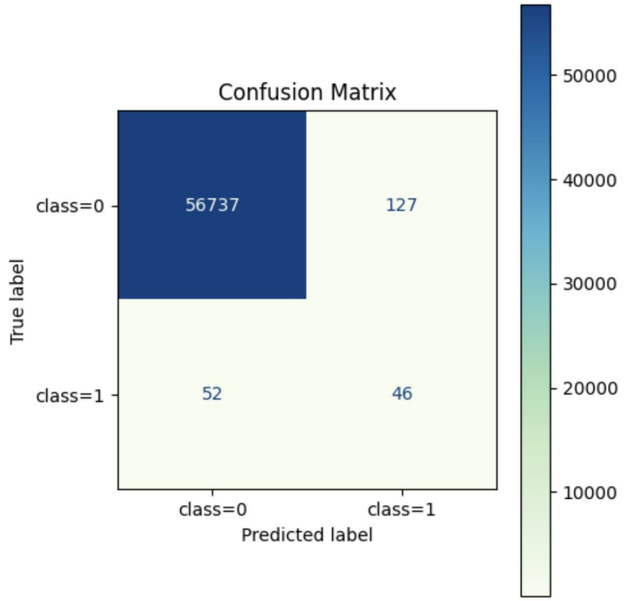
	Accuracy	Precision	Recall	f1 score
Near Miss	99.7	26.6	46.9	33.9
SMOTE	99.9	65.79	82.7	73.3



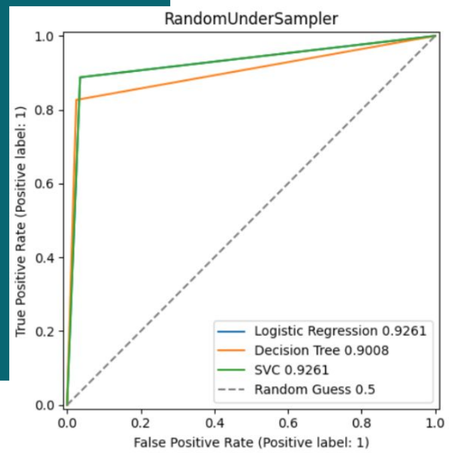
13 Confronto tra modelli e conclusioni

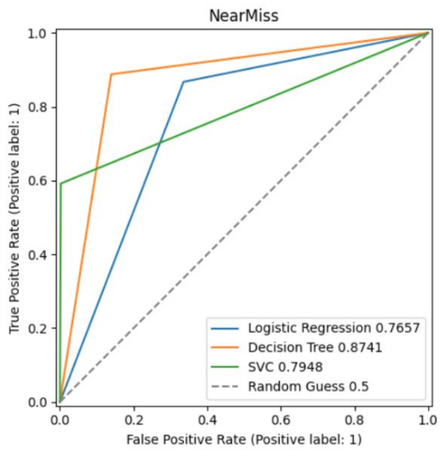
Si nota che in generale a parità di modello l'algoritmo Near Miss ha delle performance meno consistenti tra i modelli testati.

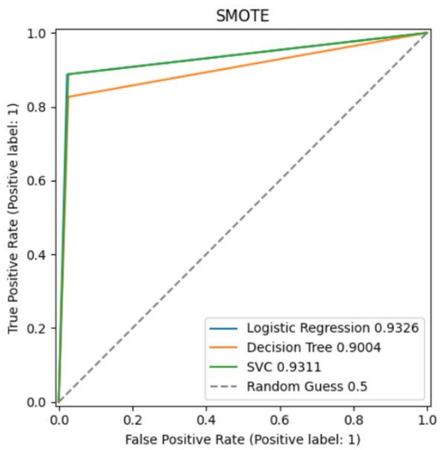
La rete neurale con il dataset Oversampled ha ottenuto le prestazioni migliori con un accuracy di 99% e uno score F1 di 73%



Near Miss NN

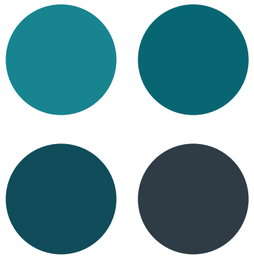






14 Conclusioni

- Metriche come l’accuracy non riescono a esprimere al meglio la bontà del modello e quanto riesca a generalizzare
- L’Oversampling è preferibile all’undersampling per questo tipo di problema per il forte sbilanciamento
- Limitare l’overfitting valutare la generalizzazione del modello



Grazie per l’attenzione