UNIVERSITÀ DEGLI STUDI DI SALERNO

CORSO DI LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI FONDAMENTI DI INTELLIGENZA ARTIFICIALE

Loan Eligibility Predicition

Autori
LIVIO VONA, VINCENZO BONAVITA

Repository Github:
https://github.com/livio-24/LoanEligibilityPrediction

Contents

1 Introduzione

I prestiti rappresentano il business principale delle banche, gran parte dei profitti deriva dagli interessi dei prestiti. Solitamente l'idoneità per un prestito è stabilità dopo un lungo processo di verifica di documenti e di un insieme di criteri di validazione, il che richiede molto tempo e risorse. Inoltre dopo questo lungo processo non si è in grado di stabilire con sicurezza se il richiedente del prestito sarà in grado di ripagare il prestito.

2 Definizione del problema

2.1 Obiettivi

Lo scopo del progetto è quello di automatizzare questo processo di verifica andando a realizzare un un agente intelligente in grado di predirre lo stato di un prestito sulla base di informazioni contenute all'interno di un form, le quali corrispondono ai criteri di validizione per il prestito.

2.2 Specifica P.E.A.S

- **Performance**: la misura di performance dell'agente è l'accuratezza con cui predice lo stato del prestito di una persona.
- **Enviroment**: l'ambiente in cui l'agente opera è composto da un insieme di criteri di validazione.
- **Actuators**: l'attuatore dell'agente corrisponde alla predizione effettuata.
- **Sensors**: I sensori dell'agente sono l'insieme dei dati forniti in input ai fini dell'apprendimento.

2.2.1 Caratteristiche dell'ambiente

- **Completamente osservabile**: l'agente ha sempre accesso a tutte le informazioni relative ai criteri.
- **Stocastico**: lo stato successivo dell'ambiente non dipende solo dallo stato corrente e dall'azione dell'agente ma anche da fattori indipendenti dell'agente.

- **Sequenziale**: la predizione è fortemente influenzata dai prestiti precedenti.
- **Dinamico**: Durante l'esecuzione dell'agente il dataset potrebbe essere aggiornato.
- **Discreto**: l'ambiente fornisce un numero limitato e ben definito di percezioni e azioni.
- **Singolo**: l'ambiente consente la presenza di un unico agente.

2.3 Analisi del problema

Una soluzione inerente al problema potrebbe essere sviluppata considerando le informazioni relative ai criteri di validizione fornite in input all'agente intelligente. Poiché l'obiettivo è quello di predirre lo stato di un prestito (approvato/non approvato), abbiamo a che fare ovviamente con un problema supervisionato (i dati sono etichettati) di classificazione binario. Sono state utilizzate tecniche di machine learning per migliorare la qualità dei dati e quindi la precisione della previsione. Inoltre dato che ci sono vari algoritmi per questo tipo di problema, ci siamo affidati al metodo empirico, costruendo vari modelli e andando a selezionare il modello con i migliori risultati.

3 Descrizione dataset

Il dataset utilizzato è stato reperito da kaggle al seguente link. Le colonne del dataset rappresentanto i criteri di validizione che devono essere considerati affinché il prestito possa essere accettato. Il dataset è composto da un totale di 13 variabili (colonne), di cui 12 indipendenti e 1 dipendente(Loan_Status, la variabile che il nostro modello dovrà predirre). Di seguito vi è una descrizione di quest'ultime:

1. **Loan_ID** (id del prestito)
2. **Gender** (maschio/femmina)
3. **Married** (sposato (SI/NO))
4. **Dependents** (numero di familiari dipendenti)
5. **Education** (laureato/non laureato)

6. **Self_Employed** (lavoro in proprio (SI/NO))
7. **ApplicantIncome** (reddito richiedente)
8. **CoapplicantIncome** (reddito co-richiedente)
9. **Loan amount** (importo prestito in migliaia)
10. **Credit_History** (storia creditizia)
11. **Property_Area** (urbana/semi-urbana/rurale)
12. **Loan_Amount_Term** (durata prestito in giorni)
13. **Loan_Status** (prestito approvato (SI/NO))

4 Exploratory Data Analysis (EDA)

La Fase di Exploratory Data Analysis è una delle fasi più importanti nel ciclo di vita di un progetto di data science. Essa si focalizza sull'analisi e sintesi degli aspetti e delle caratteristiche chiave dei dati. Di seguito sono riportati i task che sono stati eseguiti durante questa fase:

- comprensione del dataset e della sua forma (numero di righe e colonne)
- controllo del tipo di dati di ogni colonna
- controllo valori mancanti
- sintesi delle misure statistiche del dataset
- distribuzione variabile target

Risultati ottenuti:

- dataset composto da 614 righe e 13 colonne
- 5 variabili numeriche e 8 variabili categoriche
- ci sono dei valori mancanti
- dalle misure statistiche si può osservare che la media è leggermente superiore rispetto alla mediana per la maggior parte delle features numeriche, quindi sono inclinate a destra (right skewed)
- leggero sbilanciamento del dataset, i prestiti approvati (422) sono superiori a quelli non approvati (192)

4.1 Data visualization

La fase di data visualization fa parte della EDA, e come quest'ultima essa ha come scopo quello di fornire una migliore compresione dei dati tramite dei grafici. Le informazioni più importanti che possiamo ottenere dalla data visualization includono: trovare la distribuzione delle features; controllare se ci sono degli outliers nel dataset; determinare la correlazione tra le varie features, ecc...

4.1.1 Analisi Univariata

Abbiamo visualizzato la distribuzione di ogni feature singolarmente tramite dei grafici, procedendo ad una suddivisione delle features in:

- **Categoriche:** queste features hanno delle categorie (Gender, Married, Self_Employed, Credit_History, Loan_Status)
- **Ordinali:** Variabli nelle featues categoriche aventi un certo ordine (Dependents, Education, Property_Area)
- **Numeriche:** features aventi valori numerici (ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term)

Nella pagine seguenti sono riportati i grafici con la distribuzione delle variabili.

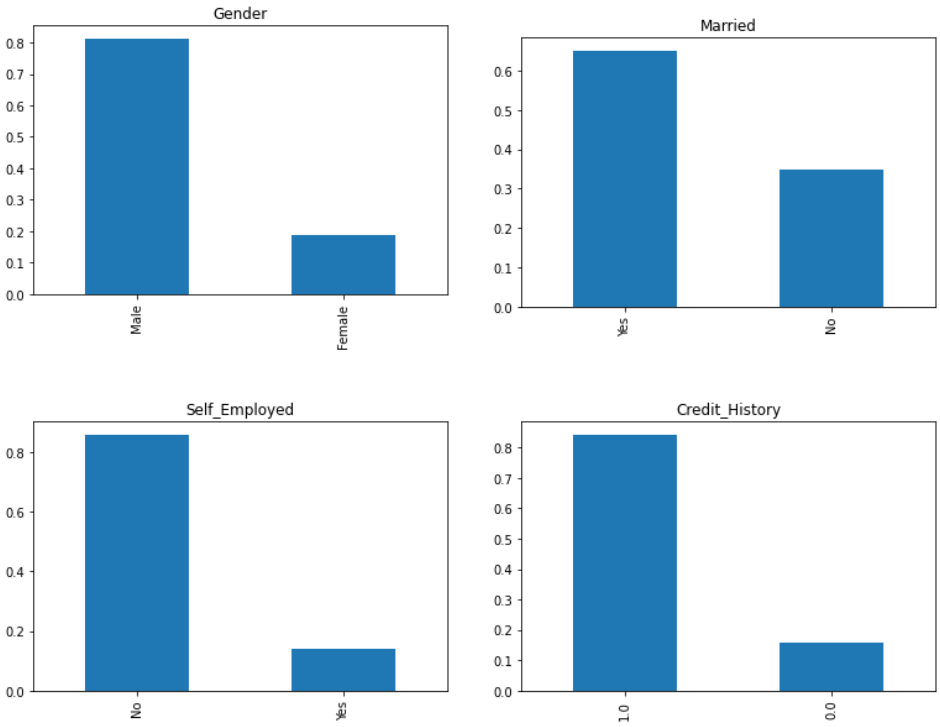


Figure 1: distribuzione variabili categoriche in percentuale

Osservazioni:

- 80% dei candidati nel dataset è maschio
- Circa il 65% dei candidati nel dataset è sposato/a
- Circa il 15% lavora in proprio
- Circa l'80% ha ripagato i propri debiti

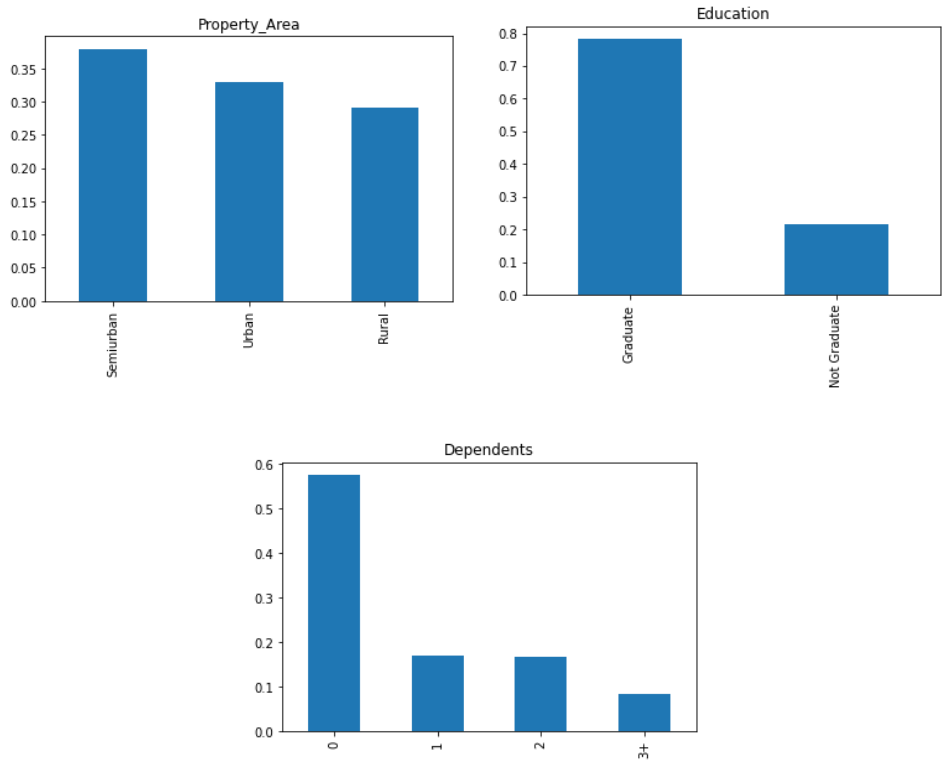


Figure 2: distribuzione variabili ordinali in percentuale

Osservazioni:

- La maggior parte dei candidati non ha familiari dipendenti
- Circa l'80% dei candidati ha una laurea
- La maggior parte dei candidati vive in un'area semiurbana

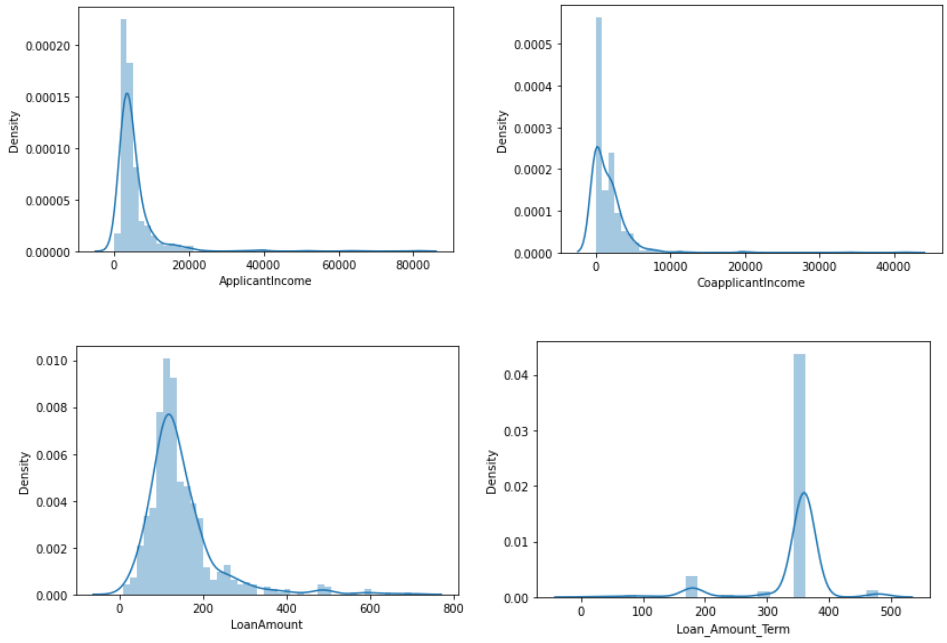


Figure 3: distribuzione variabili numeriche

Osservazioni: possiamo notare che le variabili non hanno una distribuzione normale. Questo può impattare sulle performance dei modelli, quindi successivamente saranno presi dei provvedimenti per cercare di normalizzare la distribuzione.

4.1.2 Individuazione Outliers

Per l'individuazione degli outliers abbiamo utilizzato dei boxplot, ottenendo i seguenti risultati:

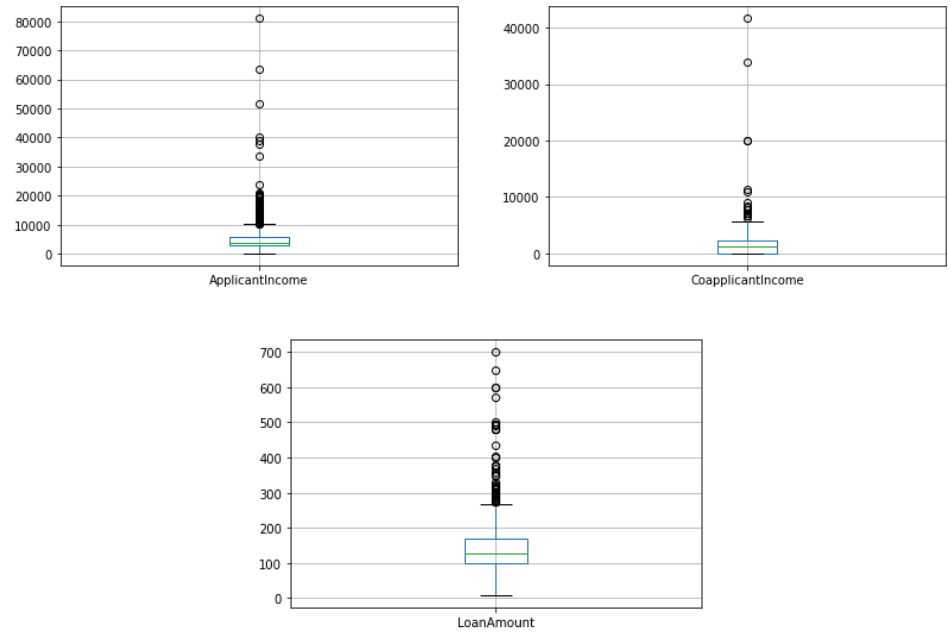


Figure 4: boxplot per individuazione outliers

Osservazioni: questi piccoli cerchi che possiamo vedere nei boxplot sono degli outliers, ossia dei valori anomali distanti dalle altre osservazioni. Quindi in queste tre features sono presenti degli outliers. Nella fase successiva vedremo come cercare di ridurre il loro impatto.

4.2 Correlazione features

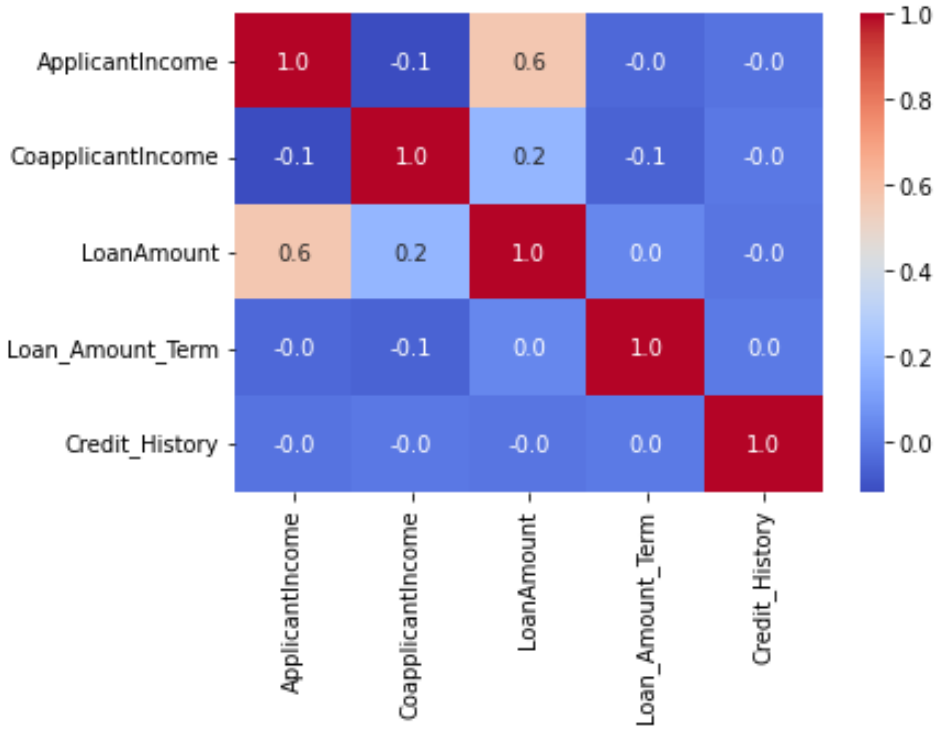


Figure 5: matrice di correlazione

Dalla matrice di correlazione otteniamo che non ci sono molte variabili correlate tra di loro, le uniche due sono ApplicantIncome - LoanAmount.

5 Ingegnerizzazione dei dati

5.1 Data Cleaning

Nella prima fase di ingegnerizzazione dei dati ci siamo concentrati sul data cleaning, ovvero è stata verificata la presenza di dati nulli o non validi. Da questa analisi è stato ottenuto il seguente risultato:

```
dataset.isnull().sum()
```

```
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

Abbiamo gestito la sostituzione dei valori mancanti in questo modo:

- Per le variabili categoriche abbiamo sostituito i valori mancanti con la moda
- Per la variabile LoanAmount abbiamo sostituito i valori mancanti con la mediana poiché la variabile ha degli outliers quindi non è un buon approccio utilizzare la media
- Per la variabile LoanAmountTerm abbiamo notato che il 360 è il valore che si ripete di più, quindi abbiamo utilizzato la moda

5.2 Gestione outliers

Nella fase di Data Visualization abbiamo visto che LoanAmount, ApplicantIncome e CoapplicanteIncome non hanno una distribuzione normale e presentano degli outliers. Quindi abbiamo cercato di normalizzare la distribuzione e ridurre l'impatto degli outliers andando ad applicare il logaritmo. La trasformazione logaritmica enfatizza i valori anomali e ci consente di ottenere potenzialmente una distribuzione a campana. L'idea è che prendere il logaritmo dei dati può ripristinare la simmetria dei dati.

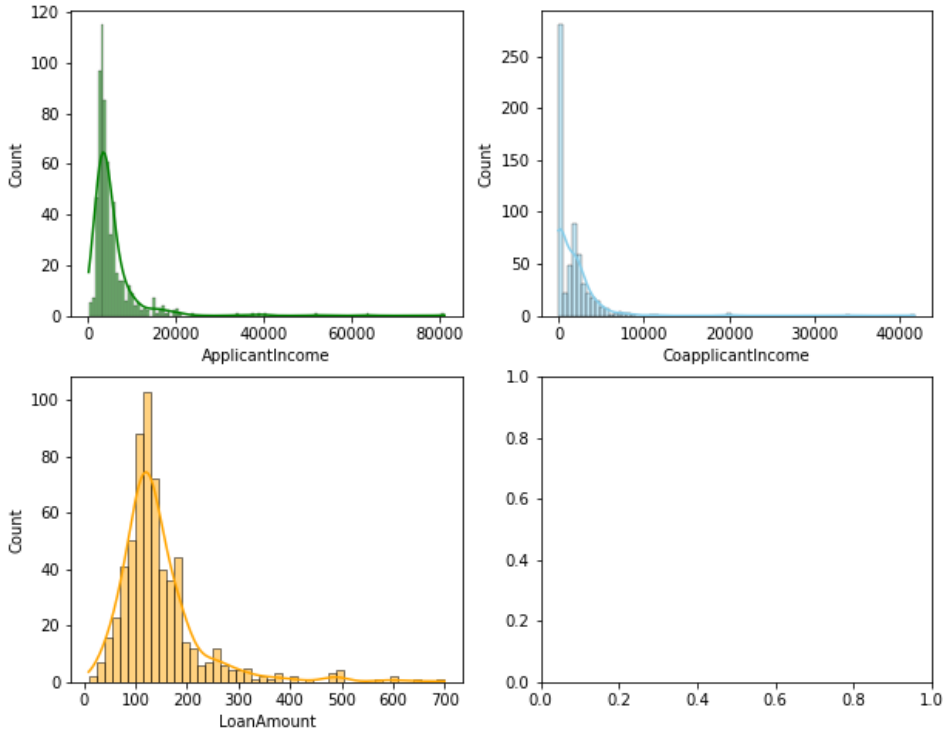


Figure 6: Distribuzioni prima di applicare trasformazione logaritmica

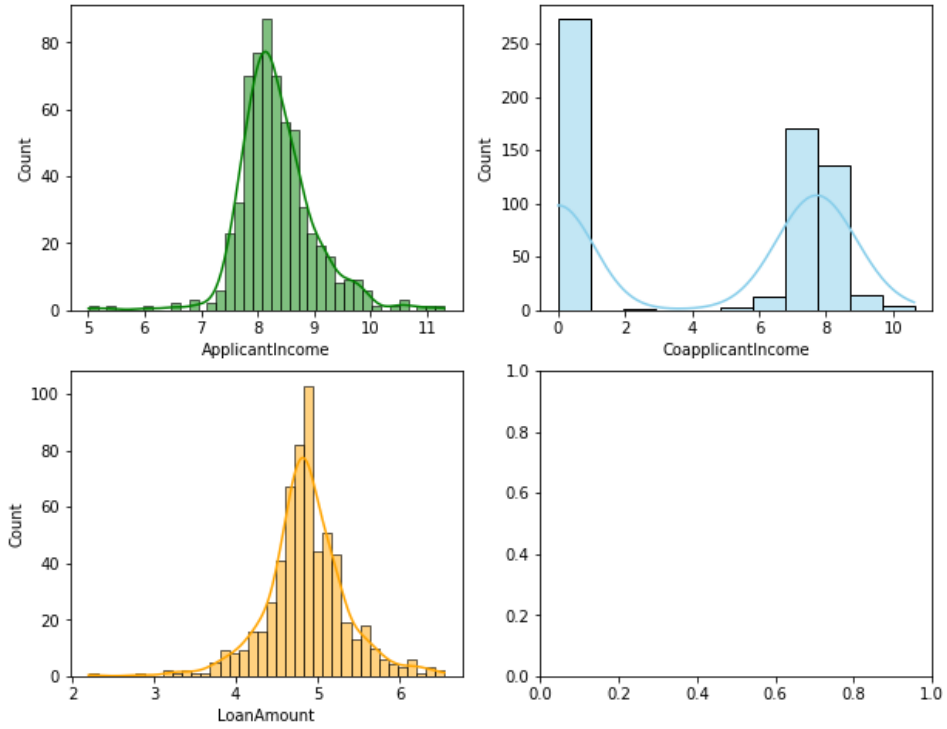


Figure 7: Distribuzioni dopo aver applicato la trasformazione logaritmica

5.3 Dataset split

Il dataset di partenza viene diviso in due insiemi:

- **Training-set**: che sarà composto delle istanze che l'algoritmo utilizzerà per l'addestramento
- **Test-set**: che sarà composto delle istanze per cui l'algoritmo addestrato dovrà predire la classe di appartenenza.

Abbiamo deciso di addestrare i modelli sull'80% del dataset e validare le loro prestazioni sul restante 20%

5.4 Feature scaling

Nella fase di feature scaling il focus è ricaduto sulla scelta di tecniche che consentono di normalizzare o scalare l'insieme dei valori. In particolare sono state prese in considerazione le seguenti strategie:

- *Standard scaling*: che normalizza i dati in modo di ottenere la somma delle medie pari a 0 e la deviazione standard uguale a 1.
- *MinMax scaling*: normalizza i valori dei dati in valori compresi in un intervallo [a, b].

Abbiamo utilizzato MinMax scaling con intervallo [0,1].

5.5 Data balancing

Per quanto concerne la fase di Data Balancing, abbiamo ritenuto opportuno non apportare modifiche al dataset.

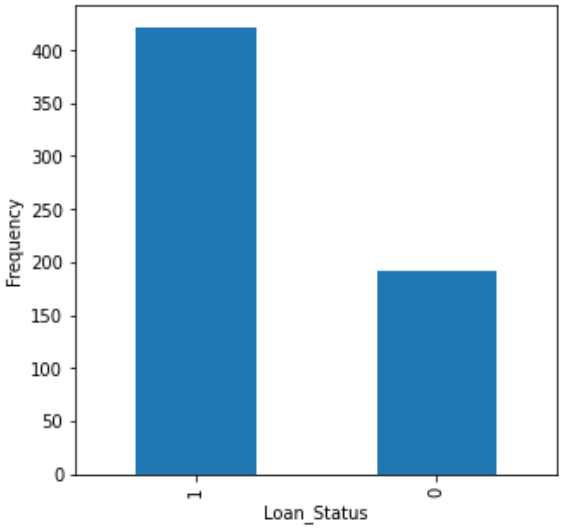


Figure 8: Distribuzione variabile target

Come mostrato dall'istogramma, la classe di minoranza è chiaramente rappresentata dai prestiti con Loan_Status uguale ad 0 (prestiti non approvati). Tuttavia, dopo un'operazione di oversampling per "bilanciare" questa differenza, abbiamo rilevato empiricamente che le prestazioni di tutti i modelli tendono a peggiorare.

5.6 Feature selection

La feature selection è il processo tramite il quale vengono selezionate le caratteristiche più correlate al problema in esame, a partire da un insieme di caratteristiche esistenti. Abbiamo deciso di approcciare la selezione delle feature utilizzando la tecnica dell'eliminazione univariata di feature attraverso l'algoritmo SelectKBest, quindi andremo a selezionare le K feature migliori correlate con la variabile dipendente. Abbiamo scelto k uguale a 5 perché ci ha dato risultati migliori.

```
fs = SelectKBest(score_func=chi2,k=5)
fs.fit_transform(X_train, y_train)

X_new_train = fs.transform(X_train)
X_new_test = fs.transform(X_test)
print(X_new_train.shape)

x.columns[fs.get_support(indices=True)]
print("features selezionate: ", x.columns[fs.get_support(indices=True)].tolist())
```

```
(491, 5)
features selezionate:  ['Married', 'Education', 'CoapplicantIncome', 'Credit_History', 'Property_Area']
```

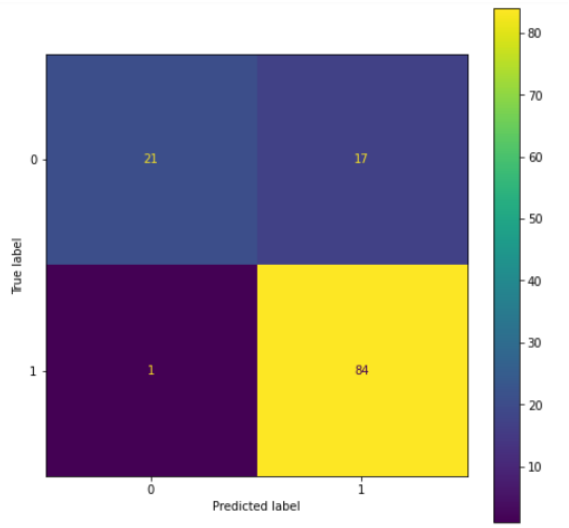
Figure 9: Feature Selection

6 Data Modeling

In seguito alla fase di ingegnerizzazione dei dati, occorre selezionare l'algoritmo da utilizzare. Avendo a che fare con un problema di classificazione abbiamo utilizzato diversi algoritmi di classificazione andando a confrontarne i risultati. Gli algoritmi selezionati sono i seguenti 5: Naïve Bayes; Logistic Regression ;Decision Tree; Random Forest; XGBoost

6.1 Logistic Regression

La Logistic Regression è utilizzata comunemente per stimare la probabilità di appartenenza a particolari classi. Di seguito i risultati ottenuti:



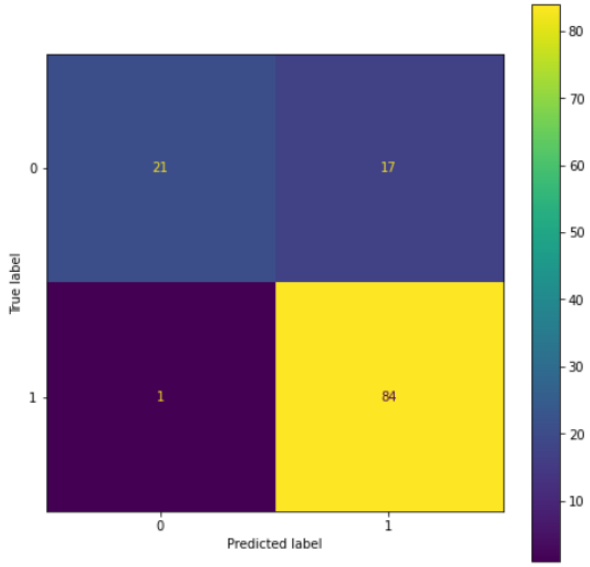
	precision	recall	f1-score	support
0	0.95	0.55	0.70	38
1	0.83	0.99	0.90	85
accuracy			0.85	123
macro avg	0.89	0.77	0.80	123
weighted avg	0.87	0.85	0.84	123

Accuracy on training data: 0.7983706720977597
Accuracy on test data: 0.8536585365853658

Figure 10: Risultati Logistic Regression

6.2 Naive Bayes

L'algoritmo considera le caratteristiche della nuova istanza da classificare e calcola la probabilità che queste facciano parte di una classe tramite l'applicazione del teorema di Bayes. Di seguito i risultati ottenuti:



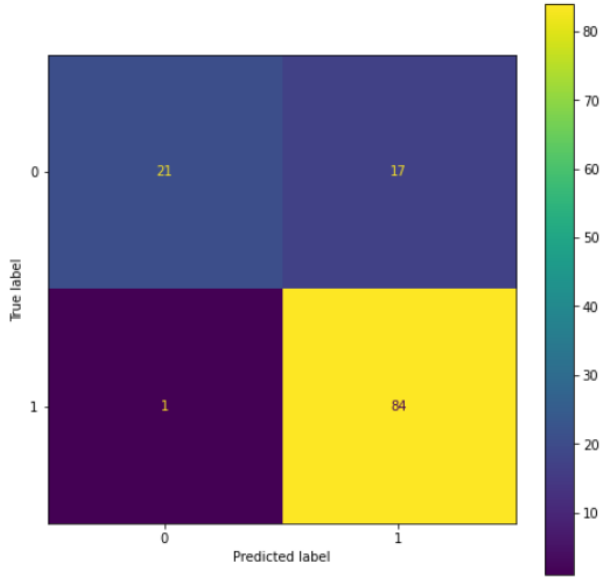
	precision	recall	f1-score	support
0	0.95	0.55	0.70	38
1	0.83	0.99	0.90	85
accuracy			0.85	123
macro avg	0.89	0.77	0.80	123
weighted avg	0.87	0.85	0.84	123

```
Accuracy on training data: 0.7983706720977597
Accuracy on test data: 0.8536585365853658
```

Figure 11: Risultati Naive Bayes

6.3 Decision Tree

L'algoritmo mira a creare un albero i cui nodi rappresentano un sotto-insieme di caratteristiche del problema e i cui archi rappresentano delle decisioni. Di seguito i risultati ottenuti:



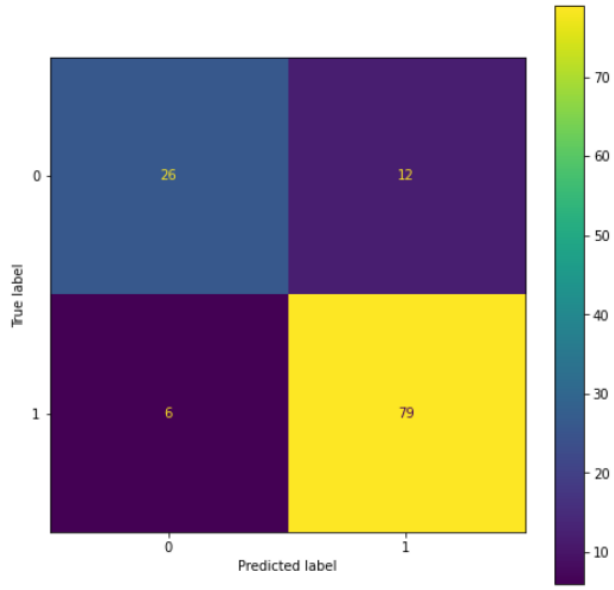
	precision	recall	f1-score	support
0	0.71	0.63	0.67	38
1	0.84	0.88	0.86	85
accuracy			0.80	123
macro avg	0.77	0.76	0.76	123
weighted avg	0.80	0.80	0.80	123

```
Accuracy on training data: 0.8859470468431772
Accuracy on test data: 0.8048780487804879
```

Figure 12: Risultati Decision Tree

6.4 Random Forest

Il RandomForest è un algoritmo di apprendimento supervisionato formato da un insieme di diversi alberi decisionali. Una foresta casuale stabilisce il risultato in base alle previsioni degli alberi decisionali prendendo l'output più frequente. Di seguito i risultati ottenuti:



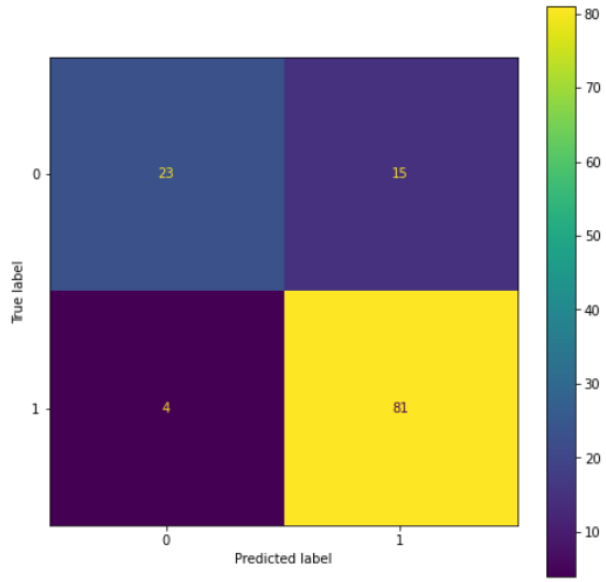
	precision	recall	f1-score	support
0	0.81	0.68	0.74	38
1	0.87	0.93	0.90	85
accuracy			0.85	123
macro avg	0.84	0.81	0.82	123
weighted avg	0.85	0.85	0.85	123

```
Accuracy on training data: 0.8859470468431772
Accuracy on test data: 0.8536585365853658
```

Figure 13: Risultati Random Forest

6.5 XGBoost

XGBoost è un algoritmo di apprendimento automatico basato sull'albero decisionale che utilizza un framework di potenziamento del gradiente. Di seguito i risultati ottenuti:



	precision	recall	f1-score	support
0	0.85	0.61	0.71	38
1	0.84	0.95	0.90	85
accuracy			0.85	123
macro avg	0.85	0.78	0.80	123
weighted avg	0.85	0.85	0.84	123

Accuracy on training data: 0.8207739307535642
Accuracy on test data: 0.8455284552845529

Figure 14: Risultati XGBoost

7 Valutazioni finali

Model	Accuracy	Precision	Recall	f1_score
Logistic Regression	0.853659	0.831683	0.988235	0.903226
Decision Tree	0.804878	0.842697	0.882353	0.862069
Naive Bayes	0.853659	0.831683	0.988235	0.903226
Random Forest	0.853659	0.868132	0.929412	0.897727
XGBoost	0.845528	0.843750	0.952941	0.895028

Come possiamo vedere i 5 modelli hanno valori buoni e molto simili per tutte le metriche, in particolare i due migliori sono Logistic Regression e Naive Bayes. In conclusione possiamo quindi ritenerci soddisfatti dai valori ottenuti dalle metriche di valutazione.