PROGETTO DI
“BIG DATA & BUSINESS INTELLIGENCE”

CLASSIFICAZIONE MUTUATARI

Andrea Bertogalli – 307673 – 2021/2022

SOMMARIO

INTRODUZIONE

Il maggior guadagno delle banche deriva dall'attività che riguarda i **contratti di mutuo** (per proprietà), un contratto di mutuo è un contratto col quale un soggetto (il mutuante), in questo caso la banca, trasferisce all'altro (il mutuatario) una determinata quantità di denaro e quest'ultimo si assume l'obbligo della restituzione. Questa tipologia di prestiti è soggetta a rischi, infatti spesso il debitore potrebbe risultare inaffidabile, non restituendo il denaro. **Lo scopo del progetto presentato è quello di classificare i richiedenti il prestito come affidabili o meno** facendo uso di modelli di Machine Learning addestrati per effettuare task di **classificazione binaria** quindi **learning supervisionato**, addestrati a partire dal dataset "Loan_Default.csv" che contiene dati di vari mutuatari raccolti nel tempo.

IL DATASET

Il dataset (non pre-processato) è abbastanza vasto contiene **148670 esempi** ed ha in tutto **33 feature ed 1 target variable** quindi si presenta come una matrice (148670 x 34). Il dataset contiene molti valori mancanti (missing values). Il dataset contiene sia variabili categoriche (in quasi tutte i valori sono espressi sottoforma di stringhe) che variabili numeriche (alcune hanno anche valori negativi). Viste le caratteristiche del dataset quest'ultimo necessita di pre-processing, soprattutto occorre effettuare feature engineering e feature selection. A seguire vengono riportate le feature presenti nel dataset con alcune considerazioni iniziali:

- **ID:** rappresenta un identificativo del record, è una variabile che non ha nessun significato nel contesto del machine learning quindi si può scartare a priori senza dover fare nessuna considerazione numerica.
- **Year:** Stesso discorso fatto per l'ID questa variabile non ha nessun significato per quanto riguarda task di Machine Learning.
- **Loan_limit:**
  - Tipologia: variabile categorica di tipo nominale.
  - Valori possibili: {cf, ncf}.
  - Numero di valori mancanti: 3344.
  - Percentuale di valori mancanti: ≃2.25%.
- **Gender:**
  - Tipologia: variabile categorica di tipo nominale.
  - Valori possibili: {Male, Female, Joint, Sex not available} Nota: Sex not available non è stato classificato come missing value in quanto è un valore ben preciso.
  - Numero di valori mancanti: 0.
  - Percentuale di valori mancanti: 0%.
- **approv_in_adv:**
  - Tipologia: variabile categorica di tipo nominale
  - Valori possibili: {pre, nopre}
  - Numero di valori mancanti: 908.
  - Percentuale di valori mancanti: ≃0.6%.
- **Loan_type:**
  - Tipologia: variabile categorica di tipo nominale
  - Valori possibili: {type1, type2, type3}

    - Numero di valori mancanti: 0.
    - Percentuale di valori mancanti: 0%.
- **Loan_purpose:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {p1, p2, p3, p4}
    - Numero di valori mancanti: 134.
    - Percentuale di valori mancanti: ≃0.1%.
- **Credit_Worthiness:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {l1, l2}
    - Numero di valori mancanti: 0.
    - Percentuale di valori mancanti: 0%.
- **Open_credit:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {opc, nopc}
    - Numero di valori mancanti: 0.
    - Percentuale di valori mancanti: 0%.
- **business_or_commercial:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {b/c, nob/c}
    - Numero di valori mancanti: 0.
    - Percentuale di valori mancanti: 0%.
- **loan_amount:**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati
    - Numero di valori mancanti: 0.
    - Percentuale di valori mancanti: 0%.
- **Rate_of_interest:**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati
    - Numero di valori mancanti: 36439.
    - Percentuale di valori mancanti: ≃24.5%.
- **Interest_rate_spread**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati, sia positivi che negativi
    - Numero di valori mancanti: 36639.
    - Percentuale di valori mancanti: ≃24.65%.
- **Upfront_charges**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati
    - Numero di valori mancanti: 39642.
    - Percentuale di valori mancanti: ≃ 26.66%.
- **Term**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati
    - Numero di valori mancanti: 41.

    - Percentuale di valori mancanti: ≃ 0.03%.
- **Neg_ammortization**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {not_neg, neg_amm}
    - Numero di valori mancanti: 121
    - Percentuale di valori mancanti: ≃ 0.08%.
- **Interest_only**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {not_int, int_only}
    - Numero di valori mancanti: 0
    - Percentuale di valori mancanti: 0%.
- **lump_sum_payment:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {ipsm, not_ipsm}
    - Numero di valori mancanti: 0
    - Percentuale di valori mancanti: 0%.
- **Property_value:**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati
    - Numero di valori mancanti: 15098.
    - Percentuale di valori mancanti: ≃ 10.16%.
- **construction_type:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {sb, mh}
    - Numero di valori mancanti: 0
    - Percentuale di valori mancanti: 0%.
- **occupancy_type:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {ir, pr, sr}
    - Numero di valori mancanti: 0
    - Percentuale di valori mancanti: 0%.
- **Secured_by:**
    - Tipologia: variabile categorica di tipo nominale
    - Valori possibili: {home, land}
    - Numero di valori mancanti: 0
    - Percentuale di valori mancanti: 0%.
- **Total_units:**
    - Tipologia: variabile categorica ordinale
    - Valori possibili: {1U, 2U, 3U, 4U}
    - Numero di valori mancanti: 0
    - Percentuale di valori mancanti: 0%.
- **Income**
    - Tipologia: variabile numerica continua
    - Valori possibili: illimitati
    - Numero di valori mancanti: 9150.
    - Percentuale di valori mancanti: ≃ 6.16%.

- **credit_type**
  - Tipologia: variabile categorica di tipo nominale
  - Valori possibili: {CIB, CRIF, EQUI, EXP}
  - Numero di valori mancanti: 0
  - Percentuale di valori mancanti: 0%.
- **Credit_Score**
  - Tipologia: variabile numerica continua
  - Valori possibili: illimitati
  - Numero di valori mancanti: 0.
  - Percentuale di valori mancanti: 0%.
- **co-applicant_credit_type**
  - Tipologia: variabile categorica di tipo nominale
  - Valori possibili: {CIB, EXP}
  - Numero di valori mancanti: 0
  - Percentuale di valori mancanti: 0%.
- **Age**
  - Tipologia: Variabile categorica ordinale
  - Valori possibili: {<25, 25-34, 35-44, 45-54, 55-64, 65-74, >74}
  - Numero di valori mancanti: 200
  - Percentuale di valori mancanti: ≃0.13%.
- **submission_of_application**
  - Tipologia: variabile categorica nominale
  - Valori possibili: {not_inst, to_inst}
  - Numero di valori mancanti: 200
  - Percentuale di valori mancanti: ≃0.13%.
- **LTV**
  - Tipologia: variabile numerica continua
  - Valori possibili: illimitati
  - Numero di valori mancanti: 15098.
  - Percentuale di valori mancanti: ≃10.15%.
- **Region**
  - Tipologia: variabile categorica nominale
  - Valori possibili: {north, south, north-east, central}
  - Numero di valori mancanti: 0
  - Percentuale di valori mancanti: 0%.
- **Security_Type**
  - Tipologia: variabile categorica nominale
  - Valori possibili: {direct, indirect}
  - Numero di valori mancanti: 0
  - Percentuale di valori mancanti: 0%.
- **dtir1**
  - Tipologia: variabile numerica discreta
  - Valori possibili: illimitati
  - Numero di valori mancanti: 24121
  - Percentuale di valori mancanti: ≃16.23%.
- **Status (Target variable)**

- Tipologia: variabile categorica nominale
- Valori possibili: { 1 -> inaffidabile, 0 -> affidabile }
- Numero di valori mancanti: 0
- Percentuale di valori mancanti: 0 %.
- Percentuale mutuatari inaffidabili: ≃25%
- Percentuale mutuatari affidabili: ≃75%

CONSIDERAZIONI SUL DATASET

È necessario fare alcune considerazioni sul dataset, notiamo che il dataset è sbilanciato per natura, infatti, come è possibile immaginare sono molti di più i mutuatari affidabili che quelli non affidabili, se non fosse così infatti le banche non ne trarrebbero guadagno, di conseguenza non è possibile utilizzare l'accuratezza per valutare modelli addestrati su questo dataset. Per calcolare l'accuratezza sarebbe necessario bilanciare il test-set. Un dataset così sbilanciato comporta che una classe venga imparata meglio che l'altra.

Conviene piuttosto utilizzare misure di performance come **precision, recall e f1-score**, a proposito di queste misure se il modello addestrato per rilevare i mutuatari non affidabili andrebbe preferito un valore di recall alto in quanto in questo modo la banca identificherebbe tutti i mutuatari non affidabili e al massimo scarterà qualche mutuatario affidabile ma non perderà soldi.

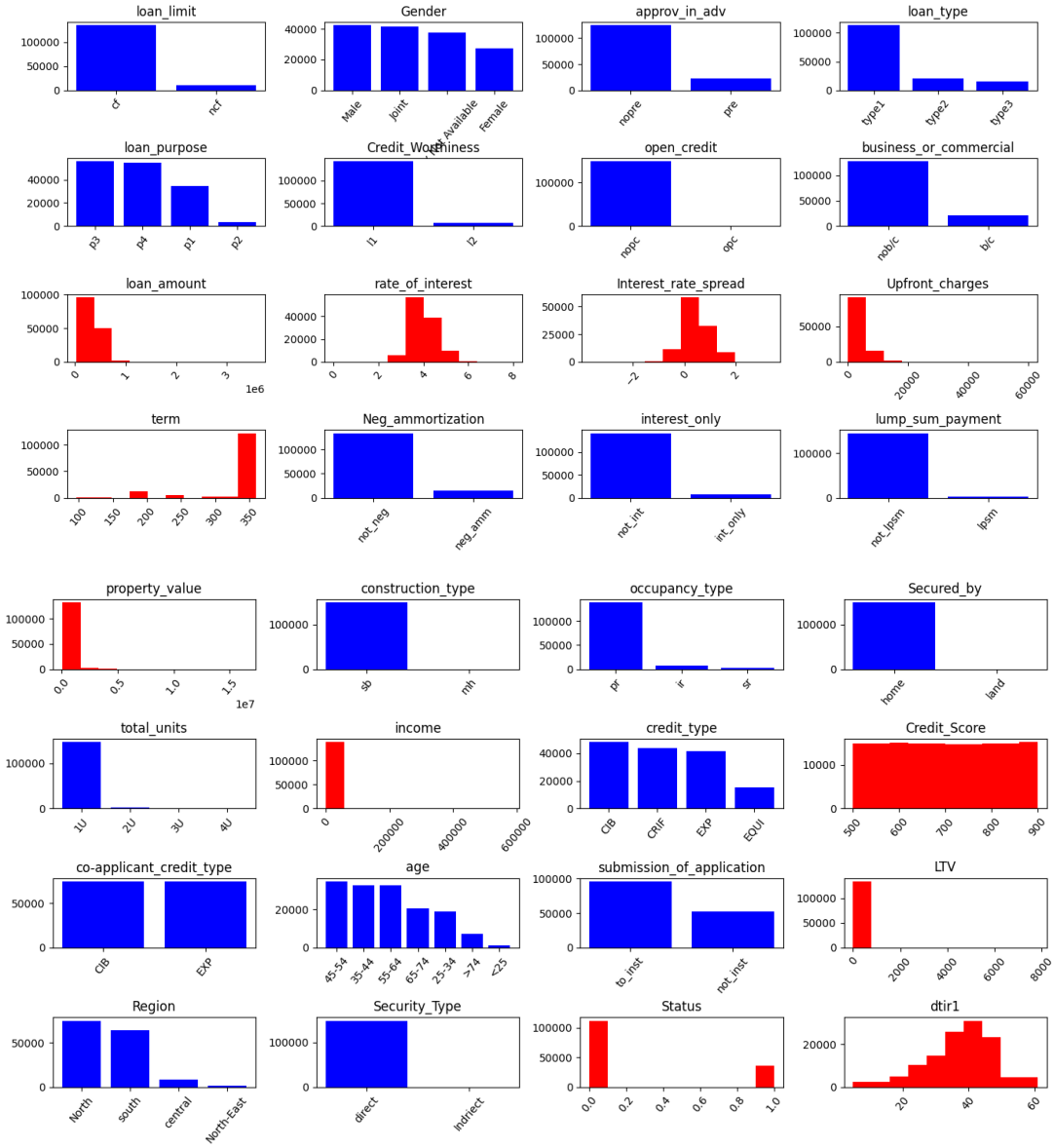
VISUALIZZAZIONE DEI DATI

Qui sotto vengono riportate alcune misure effettuate sul dataset non pre-processato, in particolare per ogni feature (numerica) si considerano la deviazione standard, la varianza, la media e la mediana mentre la moda è riportata sia per le variabili categoriche che per le numeriche.

NOTA: Normalmente sarebbe possibile calcolare la mediana anche su variabili categoriche ordinali, ma in questo caso sono sottoforma di stringhe e non è stata ancora effettuata alcuna codifica, di conseguenza la mediana non è riportata in quei casi.

Feature	D_type	Missing_count	Percent_missing	Standard_deviation	Variance	Average	Median	Mode
Credit_Score	int64	0	0.0	115.87585660446858	13427.214143819365	699.7891033833322	699.0	763
Credit_Worthiness	object	0	0.0					l1
Gender	object	0	0.0					Male
Interest_rate_spread	float64	36639	24.64451469697989	0.5130427357847288	0.26321284874147904	0.4416556604868295	0.3904	-0.028
LTV	float64	15098	10.15537768211475	39.96760298303225	1597.4092882092882	72.74645733387138	75.13586957	81.25
Neg_ammortization	object	121	0.08138830967915518					not_neg
Region	object	0	0.0					North
Secured_by	object	0	0.0					home
Security_Type	object	0	0.0					direct
Status	int64	0	0.0	0.4309422068621314	0.18571118565520406	0.24644514696979888	0.0	0
Upfront_charges	float64	39642	26.66442456447165	3251.1215097103263	10569791.070901152	3224.996126591334	2596.45	0.0
age	object	200	0.13452613170108293					45-54
approv_in_adv	object	908	0.6107486379229166					nopre
business_or_commercial	object	0	0.0					nob/c
co-applicant_credit_type	object	0	0.0					CIB
construction_type	object	0	0.0					sb
credit_type	object	0	0.0					CIB
dtir1	float64	24121	16.224524113809107	10.545434929873682	111.20619786019994	37.73293242017198	39.0	37.0
income	float64	9150	6.154570525324544	6496.586382220254	42205634.621649645	6957.338876146789	5760.0	0.0
interest_only	object	0	0.0					not_int
loan_amount	int64	0	0.0	183909.3101270855	33822634351.420513	331117.7439967714	296500.0	206500
loan_limit	object	3344	2.2492769220421067					cf
loan_purpose	object	134	0.09013250823972557					p3
loan_type	object	0	0.0					type1
lump_sum_payment	object	0	0.0					not_lpsm
occupancy_type	object	0	0.0					pr
open_credit	object	0	0.0					nopc
property_value	float64	15098	10.15537768211475	359935.31556194386	129553431388.6761	497893.46569640347	418000.0	308000.0
rate_of_interest	float64	36439	24.509988565278807	0.5613911934670424	0.3151600721023502	4.045475804367777	3.99	3.99
submission_of_application	object	200	0.13452613170108293					to_inst
term	float64	41	0.027577856998722002	58.409083508964336	3411.6210363571695	335.1365816899797	360.0	360.0
total_units	object	0	0.0					1U

Di seguito inoltre sono riportati gli istogrammi relativi alle variabili numeriche e i grafici a barre relativi alle variabili categoriche (alcuni sono riportati in notazione esponenziale):



ELABORAZIONE DEI VALORI MANCANTI

Alcune features del dataset contengono valori mancanti, le percentuali sono riportate sopra, per questo motivo bisogna procedere con il riempimento/rimozione dei valori mancanti. Le features che contengono missing values sono: loan_limit, approv_in_adv, loan_purpose, rate_of_interest, Interest_rate_spread, Upfront_charges, Neg_ammortization, property_value, income, submission_of_application, LTV, dtir1. Per decidere in che modo andare a riempire questi valori nelle **variabili categoriche** si è fatta un'analisi in base al **numero di occorrenze per ogni valore** e per le variabili loan_limit, submission_of_application, approv_in_adv, Neg_ammortization per le quali vi era un valore che occorre in netta maggioranza i missing values sono stati riempiti con questo valore (la moda), mentre per le features age, loan_purpose gli esempi sono stati rimossi. Per features numeriche si sono riempiti con la media solo i valori mancanti delle features che presentavano una percentuale di valori mancanti superiore al 10%, per quelle con percentuale inferiore si sono rimossi gli esempi, questa decisione è stata presa per limitare la perdita di dati, infatti gli esempi contenenti missing values sulle features numeriche costituiscono circa il **33.8% del dataset**. **Seguendo l'approccio proposto invece perdiamo circa il 6.3% dei dati**. Sotto è riportata una tabella che mostra come sono cambiati i dati dopo il processo di pulizia

Feature	D_type	Missing_count	Percent_missing	Standard_deviation	Variance	Average	Median	Mode
Credit_Score	int64	0	0.0	115.82974920931238	13416.530801892202	699.8683917820211	699.0	867
Credit_Worthiness	object	0	0.0					l1
Gender	object	0	0.0					Male
Interest_rate_spread	float64	0	0.0	0.4197752811563192	0.17621128666986682	0.46439697628562704	0.4415880489940321	0.4415880489940321
LTV	float64	0	0.0	17.871284715702426	319.3828173896992	71.52719036295568	71.64804469	71.5271903629557
Neg_ammortization	object	0	0.0					not_neg
Region	object	0	0.0					North
Secured_by	object	0	0.0					home
Security_Type	object	0	0.0					direct
Status	int64	0	0.0	0.4351463158709794	0.1893523162160862	0.2537297367118038	0.0	0
Upfront_charges	float64	0	0.0	2792.2621132358277	7796727.70901221	3340.2257247306197	3227.9958727488342	3227.9958727488342
age	object	0	0.0					45-54
approv_in_adv	object	0	0.0					nopre
business_or_commercial	object	0	0.0					nob/c
co-applicant_credit_type	object	0	0.0					CIB
construction_type	object	0	0.0					sb
credit_type	object	0	0.0					CIB
dtir1	float64	0	0.0	9.959539147275077	99.19242002610477	37.73576238324713	37.73576238324712	37.73576238324712
income	float64	0	0.0	6497.505787963757	42217581.46462252	6956.694581386838	5760.0	0.0
interest_only	object	0	0.0					not_int
loan_amount	int64	0	0.0	184495.2035240341	34038480123.374763	328986.849942233	296500.0	206500
loan_limit	object	0	0.0					cf
loan_purpose	object	0	0.0					p3
loan_type	object	0	0.0					type1
lump_sum_payment	object	0	0.0					not_lpsm
occupancy_type	object	0	0.0					pr
open_credit	object	0	0.0					nopc
property_value	float64	0	0.0	347289.58597224334	120610056524.7722	502332.824887869	458000.0	498328.2945794617
rate_of_interest	float64	0	0.0	0.4711018475660172	0.22193695078011488	4.079042050702213	4.044861612407534	4.044861612407534
submission_of_application	object	0	0.0					to_inst
term	float64	0	0.0	59.80112544377408	3576.1746043420044	333.75633822020336	360.0	360.0
total_units	object	0	0.0					1U

CODIFICA DELLE VARIABILI CATEGORICHE

Molte delle feature del dataset non hanno valori in formato numerico (non sono codificate), di conseguenza bisogna effettuare un altro passaggio di pre-processing e andare a codificare le feature che nella tabella riportata sopra vengono indicate come “object”. Qui sotto è riportata la “mappatura” effettuata:

```
CATEGORICAL FEATURES ENCODING:
Feature loan_limit mapping: ['cf', 'ncf'] -> [0, 1]
Feature Gender mapping: ['Sex Not Available', 'Male', 'Joint', 'Female'] -> [0, 1, 2, 3]
Feature approv_in_adv mapping: ['nopre', 'pre'] -> [0, 1]
Feature loan_type mapping: ['type1', 'type2', 'type3'] -> [0, 1, 2]
Feature loan_purpose mapping: ['p1', 'p4', 'p3', 'p2'] -> [0, 1, 2, 3]
Feature Credit_Worthiness mapping: ['l1', 'l2'] -> [0, 1]
Feature open_credit mapping: ['nopc', 'opc'] -> [0, 1]
Feature business_or_commercial mapping: ['nob/c', 'b/c'] -> [0, 1]
Feature Neg_ammortization mapping: ['not_neg', 'neg_amm'] -> [0, 1]
Feature interest_only mapping: ['not_int', 'int_only'] -> [0, 1]
Feature lump_sum_payment mapping: ['not_lpsm', 'lpsm'] -> [0, 1]
Feature construction_type mapping: ['sb', 'mh'] -> [0, 1]
Feature occupancy_type mapping: ['pr', 'sr', 'ir'] -> [0, 1, 2]
Feature Secured_by mapping: ['home', 'land'] -> [0, 1]
Feature total_units mapping: ['1U', '2U', '3U', '4U'] -> [0, 1, 2, 3]
Feature credit_type mapping: ['EXP', 'EQUI', 'CRIF', 'CIB'] -> [0, 1, 2, 3]
Feature co-applicant_credit_type mapping: ['CIB', 'EXP'] -> [0, 1]
Feature age mapping: ['25-34', '55-64', '35-44', '45-54', '65-74', '>74', '<25'] -> [0, 1, 2, 3, 4, 5, 6]
Feature submission_of_application mapping: ['to_inst', 'not_inst'] -> [0, 1]
Feature Region mapping: ['south', 'North', 'central', 'North-East'] -> [0, 1, 2, 3]
Feature Security_Type mapping: ['direct', 'Indriect'] -> [0, 1]
```

FEATURES SELECTION

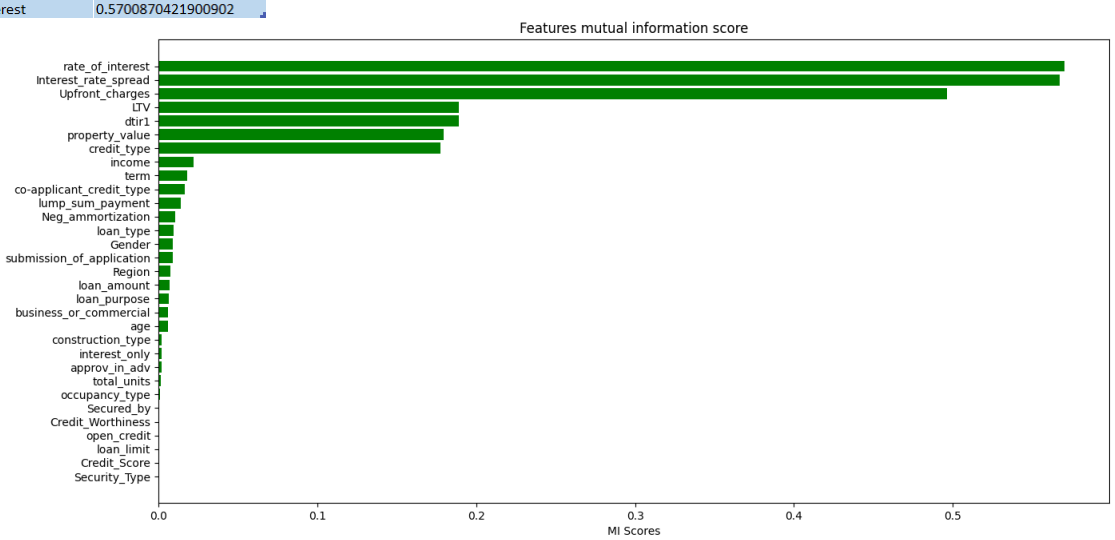
Il dataset contiene valori negativi e inoltre potrebbe contenere dipendenze non lineari tra le variabili, di conseguenza le features vengono selezionate basandosi sulla mutua informazione, di seguito si riportano gli scores per ogni feature ottenuti basandosi sulla mutua informazione più in particolare utilizzando il metodo “*SelectKBest*”:

Feature	MI_Score
Secured_by	0.0
total_units	0.0
Credit_Worthiness	0.0
open_credit	0.0
Credit_Score	0.0
construction_type	0.0003175023921877518
interest_only	0.0003322250323414266 4
Security_Type	0.0007902430412309869
occupancy_type	0.0020972352501313907
approv_in_adv	0.0029782513556140255
loan_limit	0.003717829645564885
age	0.0038358468039334515
business_or_commercial	0.005840766158124344
loan_amount	0.006326253613643473
Region	0.006416850324378887
loan_type	0.007699687930271981
loan_purpose	0.008330445794572316
submission_of_application	0.008938688011234985
Gender	0.009275790172420528
Neg_ammortization	0.009628931397589469
lump_sum_payment	0.01450878265020239
co-applicant_credit_type	0.017931576752413525
term	0.01824580365009787
income	0.020959376887993875
credit_type	0.17663720366618518
property_value	0.17847187735518721
dtir1	0.1886121034047752
LTV	0.1888353145186128
Upfront_charges	0.49647103078093235
Interest_rate_spread	0.5670774272743275
rate_of_interest	0.5700870421900902

Si sono selezionate le 7 features con punteggio più alto, di conseguenza si selezionano le ultime 7 della tabella riportata a sinistra in quanto sono le features più significative.

Queste 7 features quindi sono:

- **credit_type**: con un puntegio di 0.176637
- **property_value**: con un puntegio di 0.178472
- **dtir1**: con un puntegio di 0.188612
- **LTV**: con un puntegio di 0.188835
- **Upfront_charges**: con un puntegio di 0.496471
- **Interest_rate_spread**: con un puntegio di 0.567077
- **rate_of_interest**: con un puntegio di 0.570087



DATASET SPLITTING

Il dataset è stato suddiviso in **30% test** e **70% training**, successivamente dal training-set è stato estratto un **10%** di esempi che vanno a comporre il **validation-set**. Per valutare gli algoritmi non viene usato un approccio di cross validation, come per esempio k-fold in quanto essendo un dataset ampio non vi è il bisogno. Alla fine dello split si ottiene che il training-set è composto da **87792** esempi, il test-set da **41806** e infine il validation-set da **9755**.

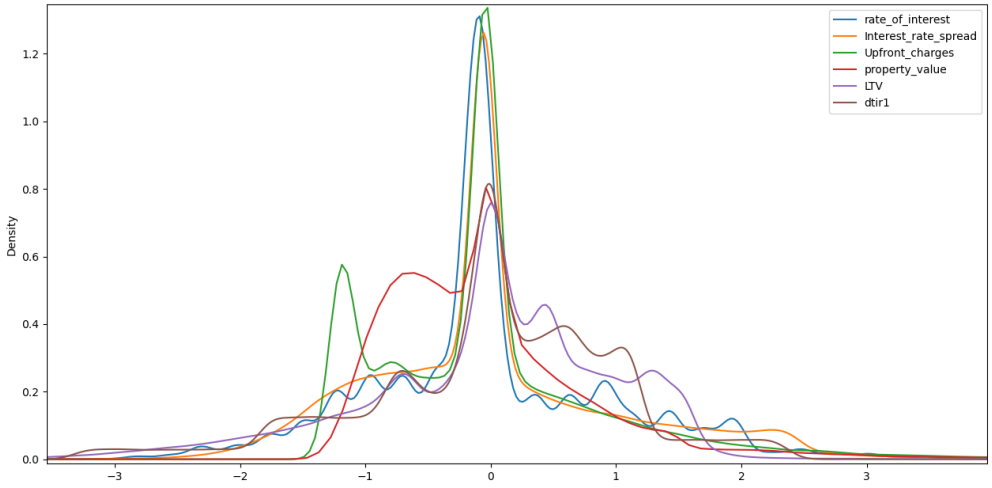
FEATURES SCALING

6 delle 7 features selezionate sono feature numeriche che hanno range molto diversi, per questo su queste features conviene fare features scaling, dato che non si conosce il massimo e il minimo di queste features, quest'ultime sono scalate mediante **z-score**, lo z-score permette di scalare le feature in modo che abbiano **media nulla e varianza unitaria**, lo z-score è definito come:

$$x' = \frac{x - \bar{x}}{\sigma} \; con \; \bar{x} = avg(x)$$

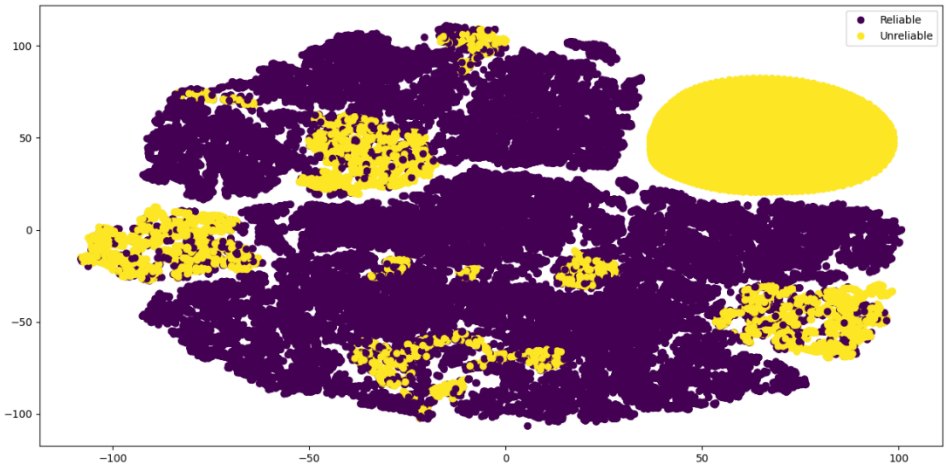
Per effettuare lo scaling si è utilizzato lo Standard scaler di sklearn, il **fitting dello scaler viene effettuato sui valori delle features numeriche del training-set** infatti i parametri devono essere determinati sul training-set e non sul test-set, e successivamente vengono trasformati sia le 6

features del training-set che quelle test-set e anche quelle del validation-set. Questa operazione di scaling consente di rendere più veloce gli addestramenti di modelli basati sull'algoritmo di discesa del gradiente. Di seguito viene riportato un density plot delle 6 feature in questione dopo lo scaling con z-score (NOTA: non è riportato il density plot prima dello scaling in quanto i range estremamente diversi dei valori lo rendevano incomprensibile).



DATA VISUALIZATION CON T-SNE

T-SNE (T-Distributed Stochastic Neighbor Embedding) permette di andare a ridurre le dimensioni dello spazio delle features con lo scopo di mantenere le distanze relative tra i due spazi, quello con più dimensioni (7 in questo caso) e il nuovo spazio che è uno spazio virtuale, l'unico scopo di questo nuovo spazio è quello di visualizzare i dati in quanto le features perdono di significato. Di seguito è riportato il plot dello spazio in 2 dimensioni generato con T-SNE a partire dallo spazio a 7 dimensioni dove ogni dimensione è rappresentata da una delle 7 features selezionate basandosi sulla mutua informazione e successivamente scalate utilizzando z-score. Sono rappresentati gli esempi contenuti solo nel training-test per motivi computazionali.



SELEZIONE DEL MODELLO OTTIMALE

Come già specificato nel paragrafo riguardante lo splitting del dataset non vi è la necessità di utilizzare un approccio cross-validation, i modelli verranno valutati sullo stesso validation-set. I modelli che sono stati confronatati sono: Logistic regression classifier, random forest classifier, ada boost, gradient boosting, extreme gradient boosting, decision tree e anche una rete neurale. Di seguito è riportata la tabella dei risultati ottenuti con i vari modelli:

Model	Precision	Recall	F1-Score
logistic_regression	0.3333333333333333	0.0008233841086867024	0.0016427104722792608
random_forest	1.0	1.0	1.0
ada_boost	1.0	1.0	1.0
gradient_boosting	1.0	1.0	1.0
extreme_gradient_boosting	1.0	1.0	1.0
decision_tree	1.0	1.0	1.0
neural_network	0.9975359439849854	1.0	0.9987664522270878

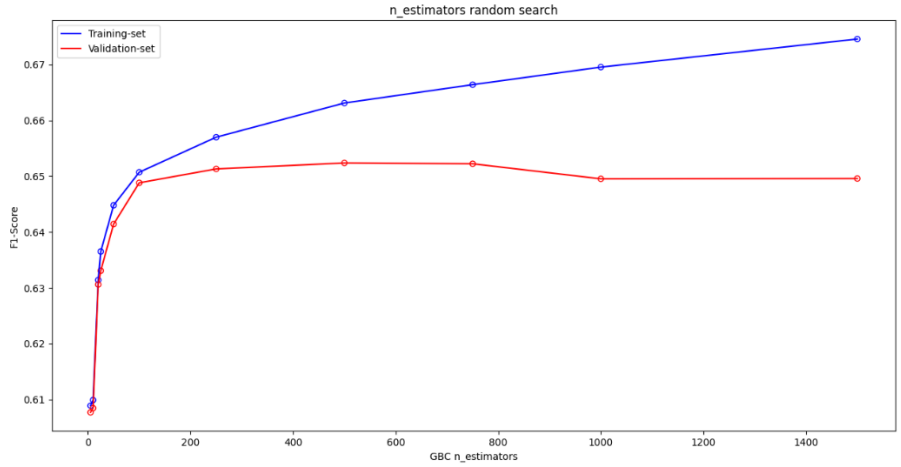
Come era già stato precisato dato il forte sbilanciamento del dataset non si è calcolata l'accuratezza. Dai risultati riportati emerge che la logistic regression non riesce a fittare i dati e risulta essere pessima. I valori per gli altri modelli sono elevatissimi, questi valori fanno pensare che qualcosa sia andato storto o che i dati non siano di qualità. Facendo ulteriori analisi in realtà è emerso che la variabile target è estremamente correlata con le 3 features **Interest_rate_spread, rate_of_interest** e appena meno ma comunque in modo estremamente significativo con **Upfront_charges.** Quindi in conclusione basta che nel dataset sia presente 1 di queste 3 feature che si riesce a predire in maniera ottima la variabile target, infatti (come riporta la tabella di risultati sottostante), rimuovendo queste 3 features otteniamo performance abbastanza scarse.

Model	Precision	Recall	F1-Score
logistic_regression	0.0	0.0	0.0
random_forest	0.7445008460236887	0.5420944558521561	0.6273764258555132
ada_boost	0.9400499583680266	0.4636550308008214	0.621012101210121
gradient_boosting	0.9418515684774292	0.5055441478439425	0.65793693212186
extreme_gradient_boosting	0.9201183431952663	0.5108829568788501	0.6569844203855294
decision_tree	0.6045794804051079	0.5638603696098563	0.583510412239694
neural_network	0.9682819247245789	0.45133471488952637	0.6156862835124907

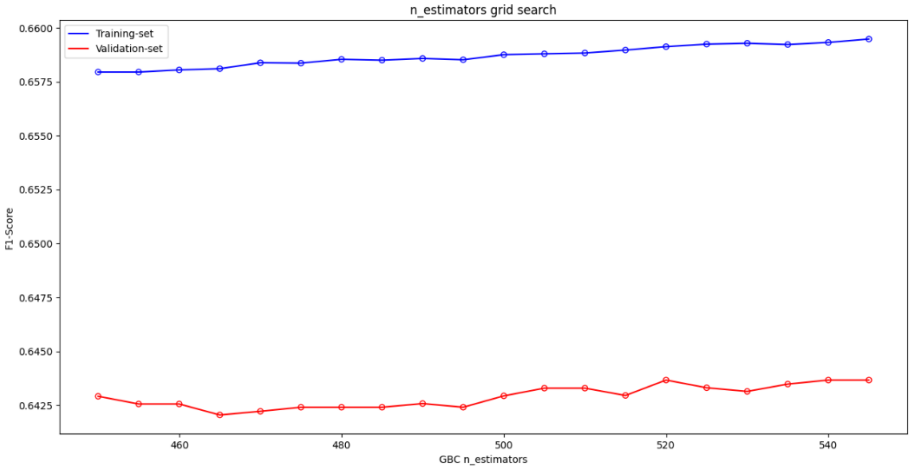
Come modello ottimale si è considerato il gradient boosting in quanto anche rimuovendo le 3 feature in questione da le performance migliori. Ovviamente **Interest_rate_spread, rate_of_interest** e **Upfront_charges** non vengono rimosse, le performance di riferimento sono quelle della prima tabella, la seconda tabella serve solo a supporto dell'ipotesi fatta.

FINE TUNING DEL MODELLO OTTIMALE

Premettendo che visti i valori di performance ottimi ottenuti dal Gradient boosting non è del tutto necessario andare a fare un fine tuning del modello, si è deciso di fare fine tuning senza considerare le 3 feature **Interest_rate_spread, rate_of_interest** e **Upfront_charges** , in modo tale da ottenere le performance migliori del modello senza queste feature e poi utilizzare i parametri trovati per addestrare il modello finale mantenedo le 3 features. Vista la dimensione del dataset anche per il fine tuning non verrà adottata la cross-validation, ma piuttosto vengono combinate una **random search con una grid search**. Il fine tuning viene effettuato esclusivamente sul parametro n_estimators ovvero il numero di predittori utilizzati dal modello. Il grafico riportato qui sotto rappresenta come varia l'F1-Score del modello al variare di n_estimators sia sul training che sul validation test:



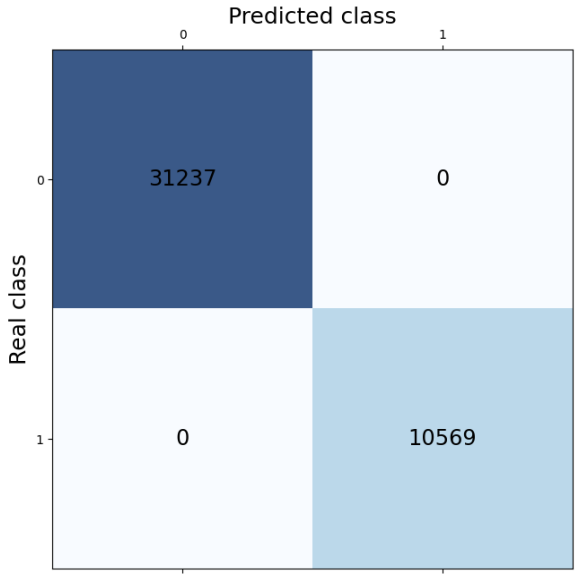
Come possiamo vedere dal grafico all'inizio il modello cade in underfitting mentre dai 750 stimatori in poi va in overfitting. Successivamente è stata effettuata una grid search in modo più preciso intorno al n_estimators ottimale trovato (che è emerso essere 500, si può notare dal grafico sopra) con la random search in modo da trovare il parametro definitivo (che è risultato essere 520). Sotto è riportato un grafico che rappresenta sempre l'F1-Score al variare del numero di predittori intorno al valore trovato con la random search.



VALUTAZIONE FINALE

Analizzato e pre-processato il datataset, trovato il modello migliore e trovati i parametri ottimali ora è possibile valutare il modello finale su test-set che ricordiamo essere il 30% del dataset. Come specificato in precedenza la valutazione sul test set è stata fatta mantenendo le features **Interest_rate_spread, rate_of_interest** e **Upfront_charges**, e utilizzando n_estimators trovato tramite il fine tuning. Vengono utilizzate sempre le 3 metriche di precision, recall, F1-Score e inoltre viene riportata in forma grafica la confusion matrix.

model	Precision	Recall	F1-Score
gradient boosting fine tuned	1	1	1



INTEGRAZIONE CON MONGODB

Come aggiunta al progetto è stata integrata la possibilità di caricare il dataset da un database MongoDB, è disponibile infatti uno script aggiuntivo "MongoDB_Load.py" che permette di caricare il dataset su un database e successivamente attingere ai dati da quest'ultimo. Nello script principale "Loans.py" viene verificata la presenza del db se il db viene trovato i dati sono caricati da esso altrimenti i dati vengono caricati dal file csv menzionato all'inizio della relazione.

INFORMAZIONI CONCLUSIVE SUL PROGETTO

Il progetto contiene 4 script python:

- "Loans.py" costituisce lo script principale
- "Constants.py" contiene tutte le configurazioni del progetto
- "MongoDB_Load.py" permette come spiegato in precedenza di caricare su un db il dataset
- "Plot_utils.py" contiene tutte le funzioni per la visualizzazione di grafici e tabelle (riportati anche nella relazione)

Nel file Constants.py tra le configurazioni è possibile specificare l'host per MongoDB e è possibile specificare il parametro VERBOSE_PLOT_ENABLED se quest'ultimo è False allora il programma verrà eseguito senza visualizzazioni grafiche dei dati, Inoltre durante l'esecuzione vengono generati dei files che contengono dati e informazioni riguardo le fasi intermedie dell'esecuzione.

NOTA: l'esecuzione del programma è piuttosto lunga in quanto esegue tutte le operazioni descritte nella relazione dall'inizio alla fine.

REFERENCES

Appunti personali del corso

scikit-learn.org - documentazione per la libreria sklearn

pandas.pydata.org - documentazione libreria pandas

matplotlib.org - documentazione libreria matplotlib