Data Analysis Report
14 Settembre 2023

Gruppo 3

Domenico Longobardi,Pasquale Nardiello,Riccardo Petruzziello,Simona Piergianni

Contents

1 Parte 1, R

1.1 Analisi preliminare dei dati

1.1.1 Dataset

Il dataset fornitoci consta di 70 osservazioni (n=70) e da 50 predittori (p=50). Lo split è stato fatto tenendo conto del rapporto 80/20 (80% training set, 20% test set), quindi 56 osservazioni per il training set e 16 per il test set.

1.1.2 Correlazione

la **correlazione** è l'interdipendenza che occore fra due e più variabili statisticamente quantitative. Nel nostro caso vediamo che la correlazioni tra le variabili non sia alta(figura **1**):

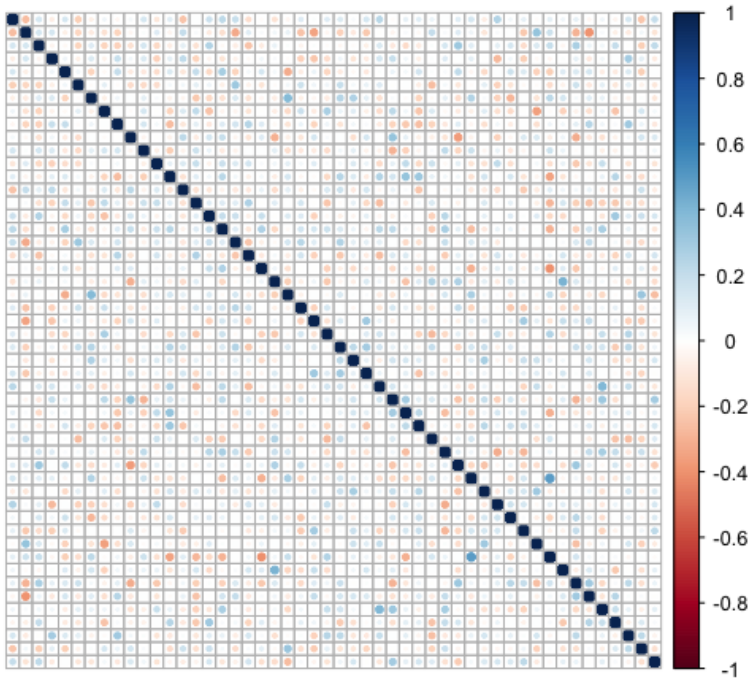

Figure 1: Matrice di Correlazione

1.1.3 Tecniche considerate

Inizialmente, abbiamo utilizzato la tecnica della **Multiple Linear Regression** dopo aver scartato , per ovvie ragioni date dal numero di predittori e dal numero di osservazioni, la Single Linear Regression. L'**MSE** della **Multiple Linear Regression** risulta comunque molto elevato, probabilmente a causa della dimensionalità dei dati e dei predittori, rendendo questa tipologia di approccio poco efficace.
L'approccio **Best Subset Selection** risulta utilizzabile per il nostro task ma, per motivi di efficienza computazionale, risulta poco auspicabile il suo effettivo utilizzo: Questo perchè il nostro dataset essendo composto da 50 predittori, comporta l'addestramento di 2^{50} modelli, rallentando in modo sensibile il processo e rendendolo inutilizzabile per un numero di predittori maggiore di 8.
Lo **Stepwise approach** si presenta come una valida alternativa al Best Subset Selection poichè, quest'ultimo, se viene utilizzato in un enorme spazio di ricerca può portare ad overfitting ed un'elevata varianza delle stime dei coefficienti. A questo si aggiunge che BSS soffre di inefficienza computazionale quando vi è un numero di predittore elevato. Gli approcci stepwise, invece, esplorano un insieme di modelli molto più ristretto. Per il nostro task abbiamo deciso di considerare sia la **Forward stepwise**

selection che la **Backward stepwise selection**. I due metodi andranno a ricercare il migliore tra un numero di modelli finito,ovvero:

- $1 + \sum_{k=0}^{p^-1}(p-k) = 1 + \frac{p*p+1}{2}$

Con p = 50, il numero di modelli considerati sarà pari a 1276 (numero ovviamente inferiore rispetto ai 2^{50} modelli considerati con BSS).
La **Ridge** regression è una tecnica per definire un modello di regressione lineare che aggiunge, nel calcolo del **Residual Sum of Squares, RSS** (e di conseguenza per il calcolo di **Ordinary Least Square, OLS**) un termine di penalità $\lambda \sum_{i=1}^{p} \beta_j^2$: Questo è un termine di normalizzazione **l2** utile a prevenire l'overfitting. Tuttavia, nel nostro caso, Ridge non svolge in alcun modo della variable selection e, associata al fatto che abbiamo un termine di penalizzazione molto basso, questo termine di penalità apporta un contributo irrisorio per ridurre la complessità del modello: Ciò ne deriva un termine di penalizzazione troppo basso, causando la perdita di informazioni potenzialmente importanti per il modello.
La **Lasso** regression è una tecnica per definire un modello di regressione lineare che aggiunge, in modo simile a Ridge, un termine di penalizzazione nel calcolo di RSS (e quindi OLS) $\lambda \sum_{i=1}^{p} |\beta_j|$: Questo è un termine di normalizzazione **l1**, il quale produce dei coefficienti di regolarizzazione pari a 0 per alcune variabili, effettuando della **Variable Selection**. Lasso risulta utile in situazioni con un alta dimensionalità, dove le variabili non sono correlate oppure hanno un minimo fattore di correlazione. Per questo motivo, la tecnica basata su Lasso è tra le migliori considerate finora.

1.2 Multiple Linear Regression

Dopo aver effettuato i calcoli sul **Multiple Linear Regression Model**, notiamo dal Summary che vi sono ben 5 predittori significativi (**X15,X16,X17,X18,X19**, figura **4**), ovvero quelli che presentano un **p-value** vicino a 0 (quindi molto minore di 1). Però, come avevamo detto in precedenza, i valori del MSE risultano molto alti e, per tali motivi, si è deciso di non considerare questa soluzione per il task.

In seguito si illustrano i grafici dove vediamo sia i valori dei coefficienti che dei loro p-value (figure **2 e 3**), andando a confermare ciò che è stato detto in precedenza con il Summary del modello.

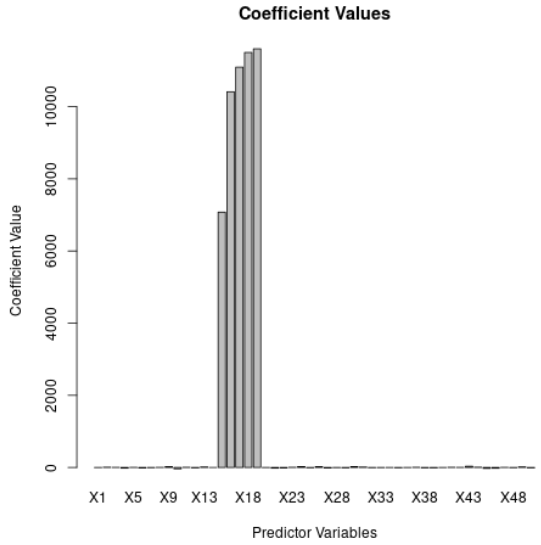

Figure 2: Coefficient Values

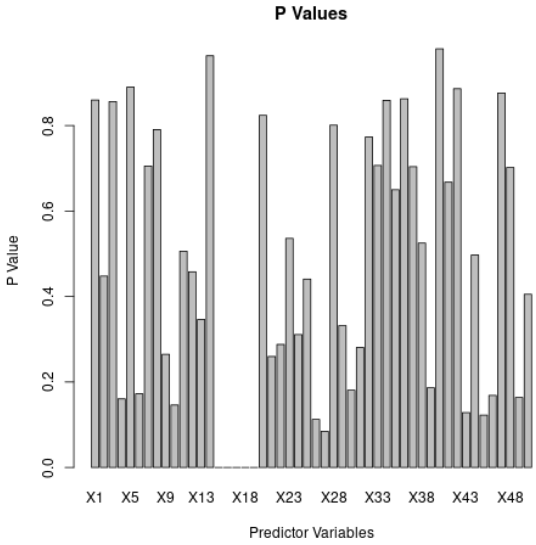

Figure 3: p-values

```
Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)    28.7824    15.5067    1.856   0.1226
X1             -1.8612    10.0098   -0.186   0.8598
X2              7.5810     9.2075    0.823   0.4478
X3              3.5594    18.6555    0.191   0.8562
X4            -20.1026    12.2081   -1.647   0.1605
X5              1.6418    11.2857    0.145   0.8900
X6            -14.4630     9.0861   -1.592   0.1723
X7             -6.7010    16.7346   -0.400   0.7054
X8              3.6350    12.9782    0.280   0.7906
X9             22.1775    17.6665    1.255   0.2648
X10           -36.7051    21.3442   -1.720   0.1461
X11             4.3318     6.0504    0.716   0.5061
X12           -10.2642    12.7640   -0.804   0.4578
X13            13.7038    13.1850    1.039   0.3463
X14             0.7068    14.6342    0.048   0.9633
X15          7076.7582    16.9710  416.990 1.51e-12 ***
X16         10413.9980    18.4982  562.974 3.36e-13 ***
X17         11096.1867     9.8946 1121.441 1.07e-14 ***
X18         11502.0471     8.6537 1329.155 4.58e-15 ***
X19         11603.5446    17.1388  677.033 1.33e-13 ***
X20            -2.3652    10.0859   -0.235   0.8239
X21           -18.7757    14.7708   -1.271   0.2596
X22           -14.9912    12.6091   -1.189   0.2879
X23             5.3405     8.0410    0.664   0.5360
X24            25.7444    22.8474    1.127   0.3110
X25            -8.2964     9.9115   -0.837   0.4407
X26            22.8709    11.8876    1.924   0.1124
X27           -15.5476     7.2385   -2.148   0.0845 .
X28            -3.4219    12.8724   -0.266   0.8010
X29            -9.5618     8.9039   -1.074   0.3319
X30            24.1284    15.5229    1.554   0.1808
X31            11.5145     9.5250    1.209   0.2808
X32            -5.0644    16.6442   -0.304   0.7732
X33            -4.1710    10.4622   -0.399   0.7066
X34            -1.4435     7.7147   -0.187   0.8589
X35            -7.0902    14.7005   -0.482   0.6500
X36            -1.5933     8.7452   -0.182   0.8626
X37             3.7825     9.3922    0.403   0.7038
X38            -9.9633    14.5906   -0.683   0.5250
X39           -11.1812     7.3044   -1.531   0.1864
X40            -0.2186     8.1075   -0.027   0.9795
X41             4.4270     9.7067    0.456   0.6675
X42             2.0976    13.9431    0.150   0.8863
X43            33.0899    18.1749    1.821   0.1283
X44             8.1956    11.1990    0.732   0.4971
X45           -25.4618    13.7150   -1.856   0.1225
X46           -22.4710    13.9672   -1.609   0.1686
X47             2.3327    14.2160    0.164   0.8761
X48            -6.4145    15.8190   -0.405   0.7019
X49            16.4430    10.0765    1.632   0.1636
X50           -10.5131    11.5720   -0.908   0.4053
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.52 on 5 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.02e+06 on 50 and 5 DF,  p-value: 3.047e-15
```

Figure 4: Risultati Multiple Linear Model

1.3 Best Subset Selection

L'approccio **Best Subset Selection** si basa sull'identificare un set di predittori, di cardinalità minore da quello di partenza, che crediamo siano quelli importanti per la risposta. per fare ciò, si definiscono una serie di modelli, creati attraverso linear regression, con tutte le combinazioni possibili di predittori decretando come miglior subset quello che presenta la stima del **test error** minore, attraverso una serie di tecniche; tra di esse abbiamo deciso di utilizzare:

- **Bayesian Information Criterion, BIC**
- **Mallow's CP, CP**
- **Adjusted R2**

Di seguito mostriamo i grafici dei modelli che utilizzano i quattro criteri sopracitati (figure **5, 6, 7**):

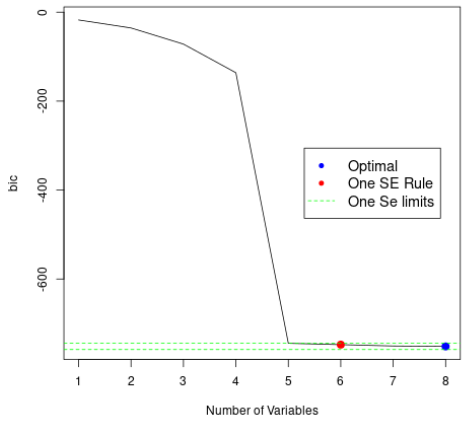

Figure 5: BIC di BSS

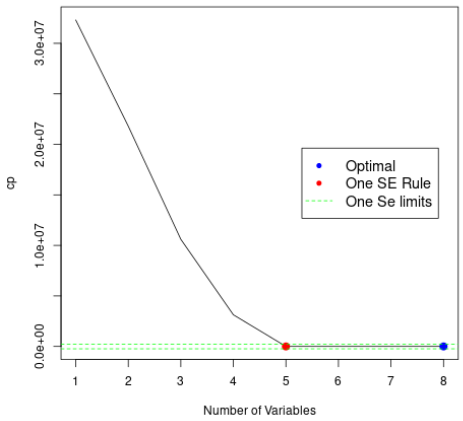

Figure 6: CP di BSS

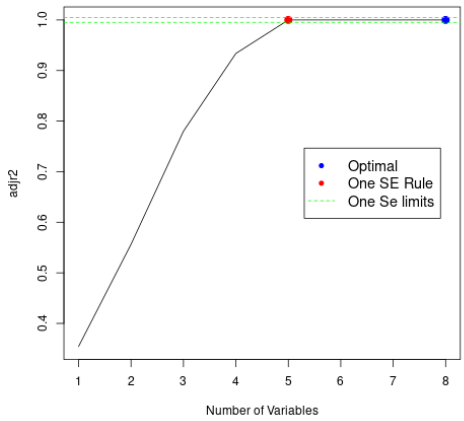

Figure 7: AdjR2 di BSS

Inoltre, abbiamo applicato la **One-Standard Error-Rule**, la quale ci permette di prendere il modello con minor numero di variabili (predittori) possibile, in accordo al fatto che il test error associato sia compreso tra il valore di one standard error ed il punto più basso della curva. Come vediamo dai grafici, il modello prevalentemente scelto è quello che presenta cinque predittori. Come possiamo vedere dai grafici **8,9,10,11**, i predittori significativi sono gli stessi della Multiple Linear Regression.

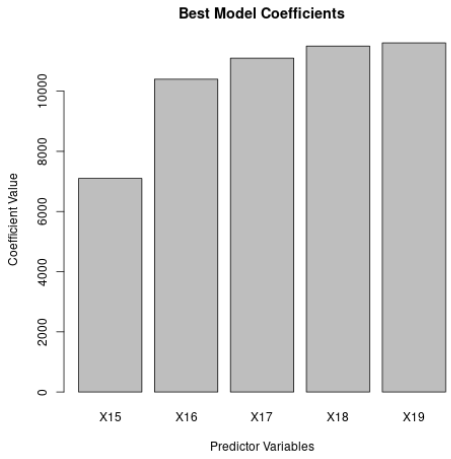

Figure 8: best coefficients

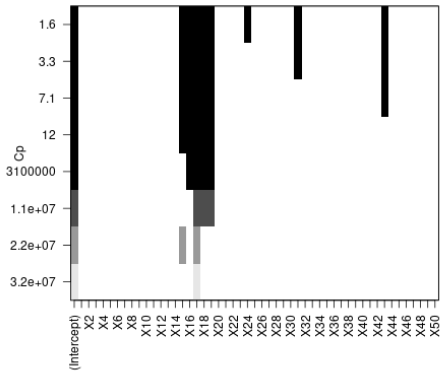

Figure 9: Coefficienti con CP

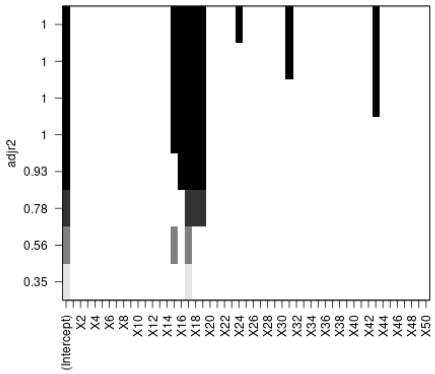

Figure 10: Coefficienti con AdjR2

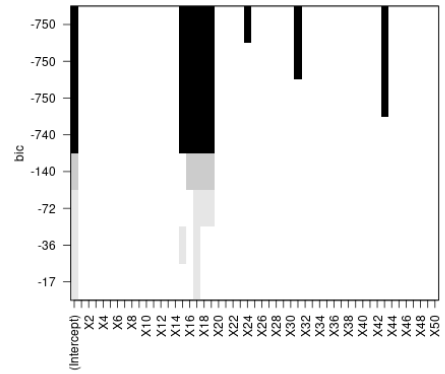

Figure 11: Coefficienti con BIC

A causa dell'inefficienza computazionale, abbiamo deciso di scartare tale approccio, pur fornendo correttamente l'indizio.

1.4 Forward stepwise selection

L'approccio Forward stepwise selection definisce un modello vuoto e va ad aggiungere un predittore alla volta,quello che massimizza un determinato criterio specificato, finchè tutti i predittori non sono nel modello. Nel nostro caso, questo approccio si è rivelato particolarmente efficace poichè, oltre ad aver dato un valore di MSE basso, è riuscito a trovare lo stesso numero di predittori significativi utili per decodificare l'indizio (figure **19, 20, 21**).

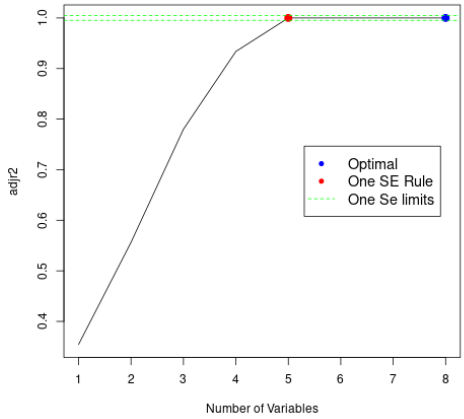

Figure 12: ADJR2 della Forward selection

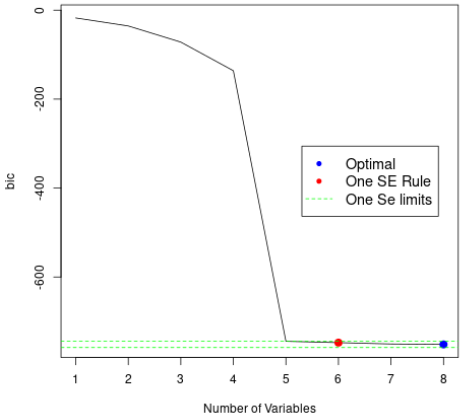

Figure 13: BIC della Forward selection

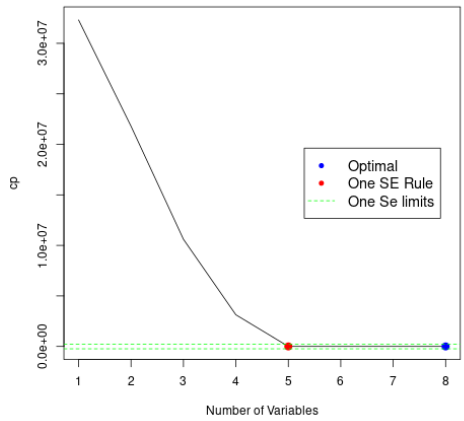

Figure 14: CP della Forward selection

Di seguito si mostra anche quali sono i predittori significativi per il task (figure **15,16,18,17**), confermando gli stessi predittori riscontrati nelle tecniche precedenti (da X15 a X19).

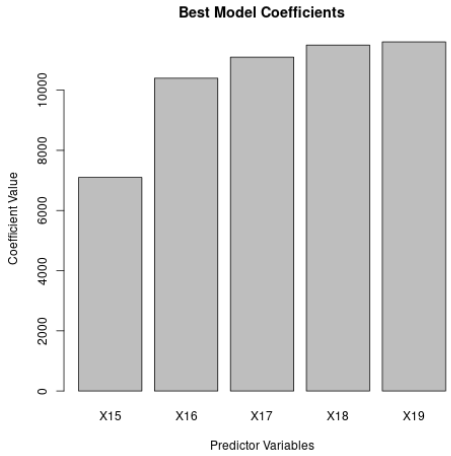

Figure 15: Best coefficients

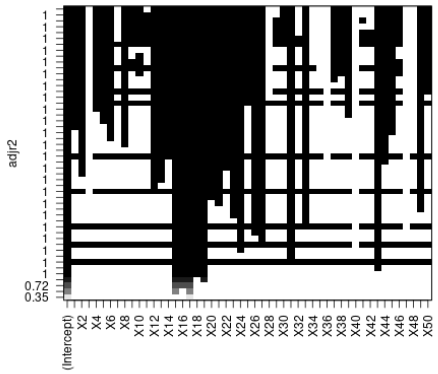

Figure 16: Coefficienti con ADJR2

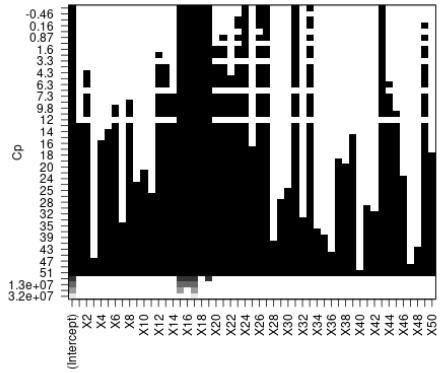

Figure 17: Coefficienti con CP

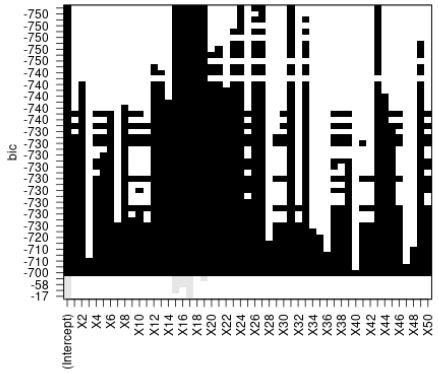

Figure 18: Coefficienti con RSS

1.5 Backward stepwise selection

L'approccio Backward stepwise selection iniziulizza un modello con tutti i predittori e va ad eliminare i predittori meno utili, uno per step, finchè non viene raggiunta una certa condizione: Nel nostro caso, notiamo come la Backward stepwise selection si comporti similmente a ciò che abbiamo già visto nel Forward stepwise selection. Vengono riportati i grafici relativi ad ADJR2, BIC, CP e RSS (**19, 20, 21**)

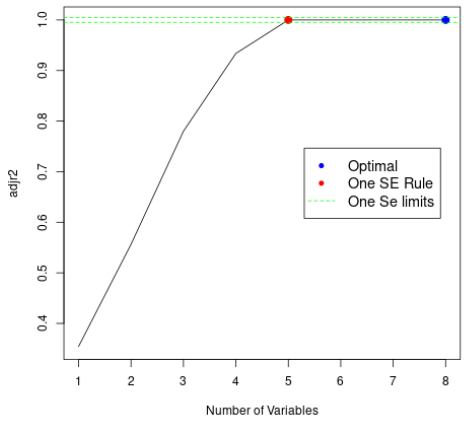

Figure 19: ADJR2 della Backward Selection

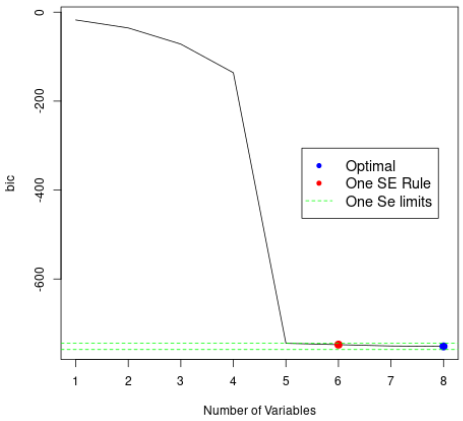

Figure 20: BIC della Backward Selection

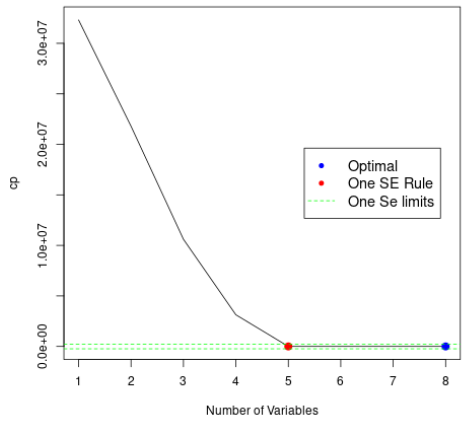

Figure 21: CP della Backward Selection

Anche in questo caso, la scelta del numero di regressori cade su 5; questo ci porta a considerare che i due approcci, con questa conformazione dei dati e grandezza del dataset, porta a dei risultati equivalenti (o almeno simili), pur utilizzando metodi di soluzione differenti. Di seguito si illustrano i grafici dove vediamo quali sono i predittori scelti (figure **22, 23, 24, 25**)

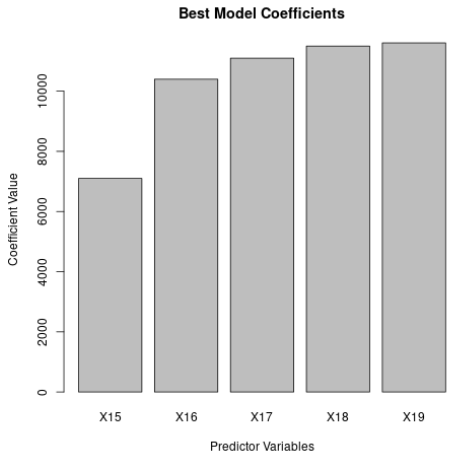

Figure 22: Best coefficients

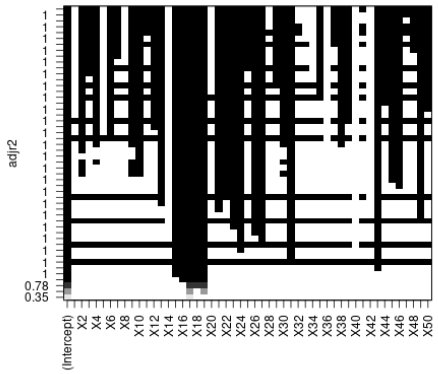

Figure 23: Coefficienti con ADJR2

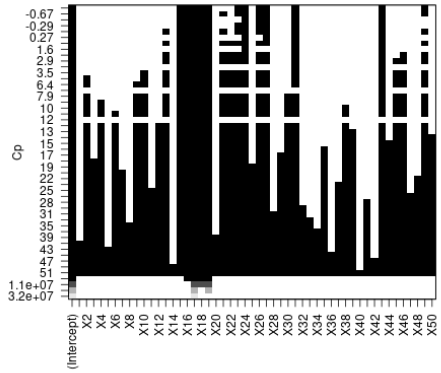

Figure 24: Coefficienti con CP

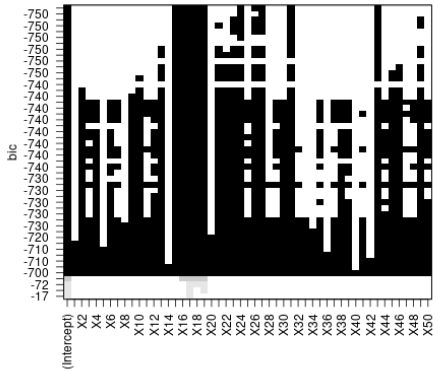

Figure 25: Coefficienti con RSS

1.6 Ridge

Ricordando che Ridge utilizza la penalizzazione l2, possiamo notare come i risultati siano i peggiori rispetto a tutte le tecniche utilizzate finora, avendo un MSE decisamente alto: Una delle cause è che il valore ottimo di lambda, scelto mediante la tecnica di cross validation, risulta molto basso e di conseguenza il termine di penalizzazione raggiunge valori prossimi allo 0 mantenendo quei predittori che non possono dare un apporto significativo al modello ma al contario potrebbero aumentare le probabilità d'errore. Quindi il modello avra un comportamento simile a quello lineare, dove i regressori significativi saranno solo i cinque di nostro interesse, lasciando i restanti a dei valori molto bassi, vicini allo 0. In definitiva, per tutte le problematiche discusse in precedenza, utilizzare Ridge è fortemente sconsigliato, pocihè andremo ad utilizzare un modello più complesso per avere lo stesso risultato di un modello più semplice ed inoltre può portare ad una lettura più difficile dei risultati (figure **26, 27**).

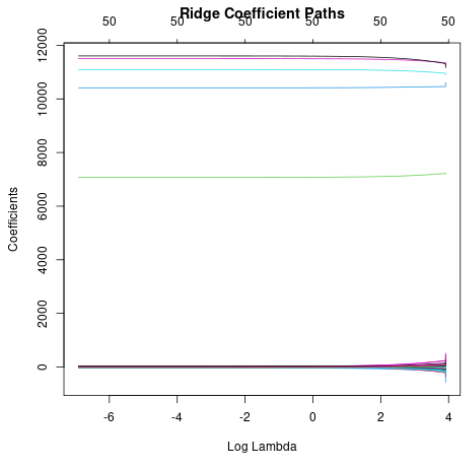

Figure 26: Coefficienti con Ridge

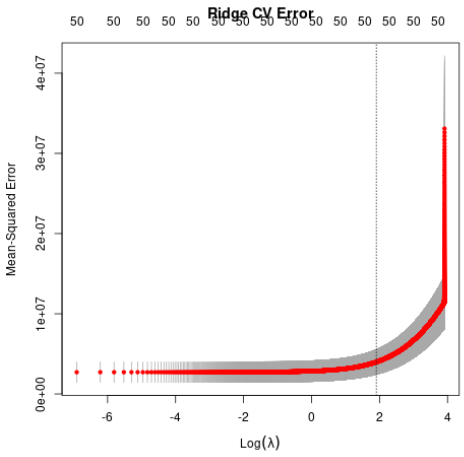

Figure 27: Cross Validation error con Ridge

1.7 Lasso

Ricordando che Lasso, a differenza di Ridge, utilizza un termine di penalizzazione l1, il quale comporta la variable selection (pone a 0 i valori dei regressori non significativi), notiamo come i risultati sono stati ben più positivi rispetto a Ridge: si hanno la totalità dei regressori non significativi uguali a 0, mentre i 5 regressori utili per trovare l'indizio hanno valori significativamente grandi, portando quindi a un MSE molto più basso rispetto all'utilizzo di Ridge (**28, 29**).

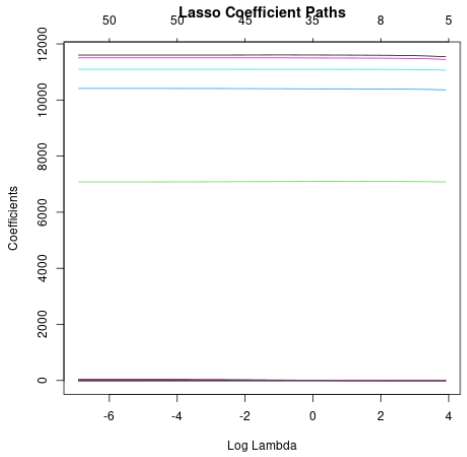

Figure 28: Coefficienti con Lasso

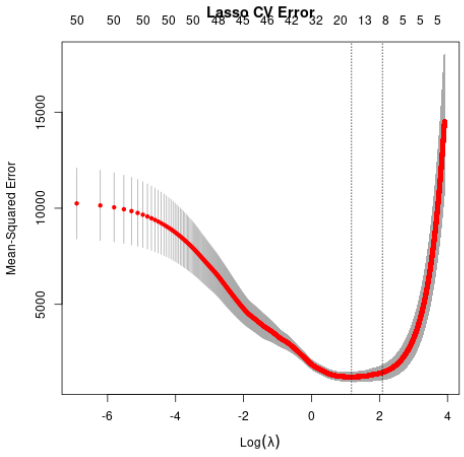

Figure 29: Cross Validation error con Lasso

1.8 Conclusioni e confronto

Alla fine, abbiamo deciso di confrontare il valore di MSE sui dati di test di tutti gli approcci considerati per decretare quale sia il migliore: Da come si nota nella figura **30**, Lasso è la tecnica con il minor MSE; inoltre possiamo considerare delle alternative gli apporcci stepwise. Invece, nel caso di BSS ciò non è possibile a causa della bassa efficcienza computazionale dell'approccio.

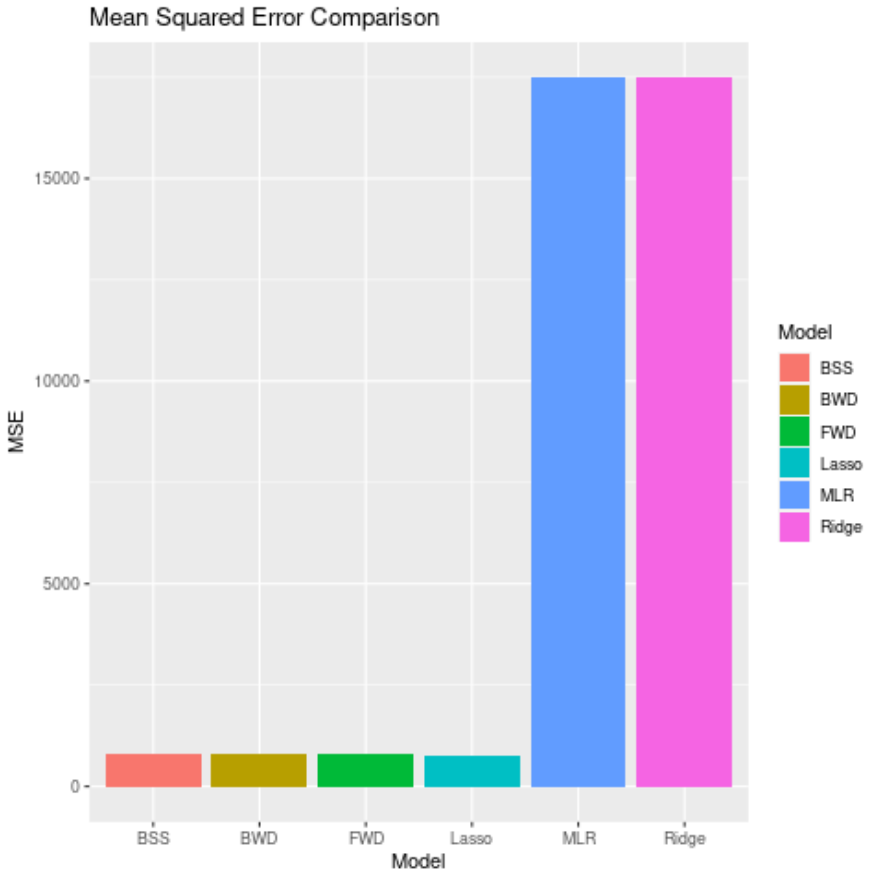

Figure 30: Confronto MSE

2 Parte 2, Python

Il nostro obiettivo è quello di realizzare il sistema (figura **31**) attraverso il linguaggio Python, per trovare il secondo indizio dell'enigma:

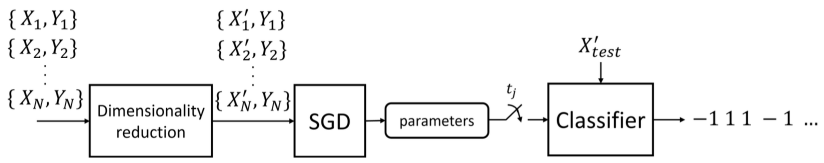

Figure 31: Diagramma a blocchi del sistema

Nel primo blocco, prendiamo in ingresso un **Training set** costituito dalle coppie feature-label (X_1, Y_1),...,(X_N, Y_N) e si svolge una riduzione dimensionale dei dati attraverso **PCA**); nel blocco successivo andiamo ad addestrare un classificatore logistico attraverso l'algoritmo **Stochastic Gradient Descent (SGD)**. Dopodichè, nell'ultimo blocco andiamo ad effettuare la classificazione degli elementi, prendendo le feature del test set $X_{test}(1), ..., X_{test}(K)$, etichettandoli con i seguenti valori:

- **1** se $X'_{test}(j)\hat{\beta}(t_j) > 0$;
- **-1** altrimenti;

Al termine, convertiremo in ASCI la stringa di 1 e -1(convertiti in 0) per ottenere l'ultimo indizio che completa la frase originale.

2.1 Analisi preliminare

2.1.1 Dataset

Il nostro dataset è composto da due file, "train.mat", "test.mat", che rappresentano rispettivamente il training set ed il test set:

- Il training set è composto da 21.600 righe e 21 colonne, dove le prime venti colonne rappresentano le 20 feature considerate per ogni elemento, mentre l'ultima colonna rappresenta la label associata;
- Il test set è composto da 72 righe e 21 colonne, dove le prima venti colonne rappresentano le 20 feature considerate per ogni elemento, mentre l'ultima colonna rappresenta l'istante di tempo associato;

Dopo aver ridotto le features tramite l'analisi della correlazione e prima di utilizzare PCA, abbiamo standardizzato l'intero dataset (sottraendo la media e dividendo per la varianza), ottenendo la seguente distribuzione dei dati X_{st} $N(0, 1)$; questo è un passaggio obbligatorio, altrimenti la PCA non verrebbe calcolata in modo corretto.

2.1.2 Correlazione

Per verificare la correlazione tra i dati, andiamo a visualizzare la matrice di correlazione (figura **32**): Si vede come le feature abbiano una bassa correlazione tra di loro, tranne nel caso delle features **0** e **1**. Per questo abbiamo eliminato una delle feature (nel nostro caso la feature 1) ed abbiamo ottenuto la seguente matrice (figura **33**):

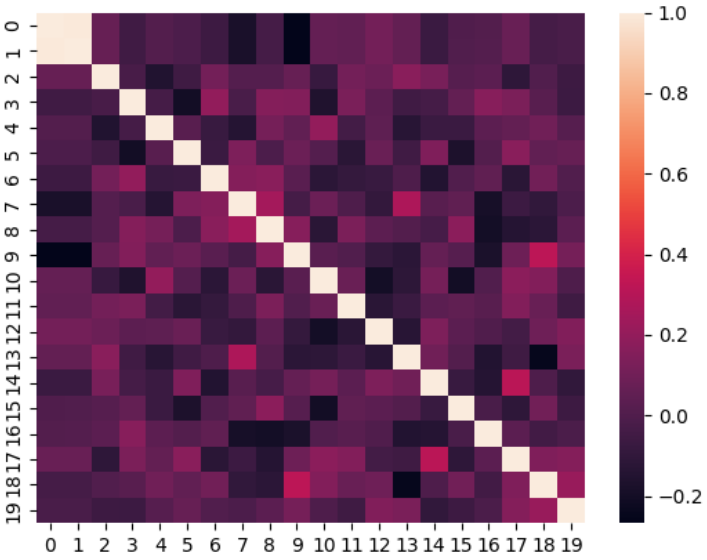

Figure 32: Matrice di correlazione

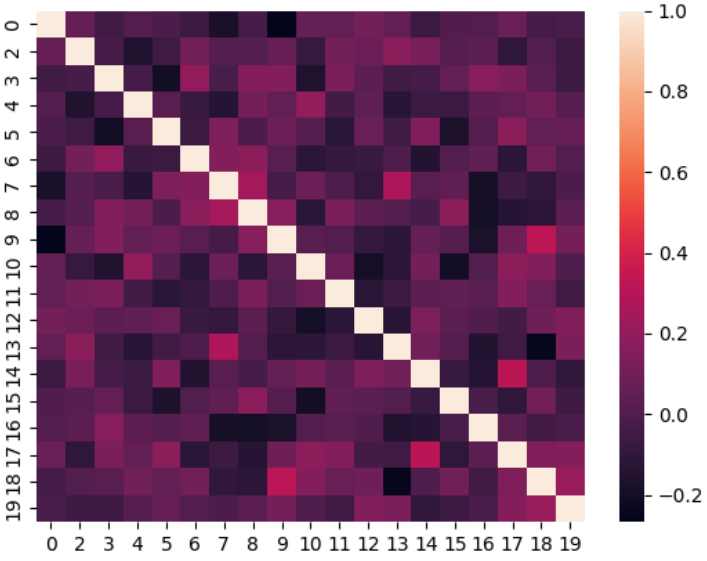

Figure 33: Matrice di correlazione dopo la riduzione

2.2 Dimensionality reduction, PCA

Dopo aver effettuato una prima riduzione di dimensionalità dei dati tramite l'analisi della correlazione, abbiamo proseguito nell'intento applicando la PCA, dopo aver standardizzato i dati per eliminare possibile scale differenti usate sulle varie features. In seguito abbiamo selezionato le K componenti che portassero ad un contributo minimo totale della varianza pari al 95 per cento della varianza totale.(figura **34**):

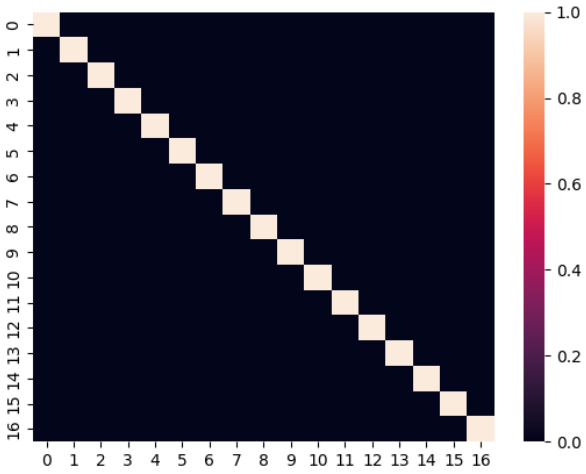

Figure 34: Matrice di correlazione dopo PCA

Di seguito, andiamo a vedere il grafico che mostra le varianze delle componenti della PCA (figura **35**):

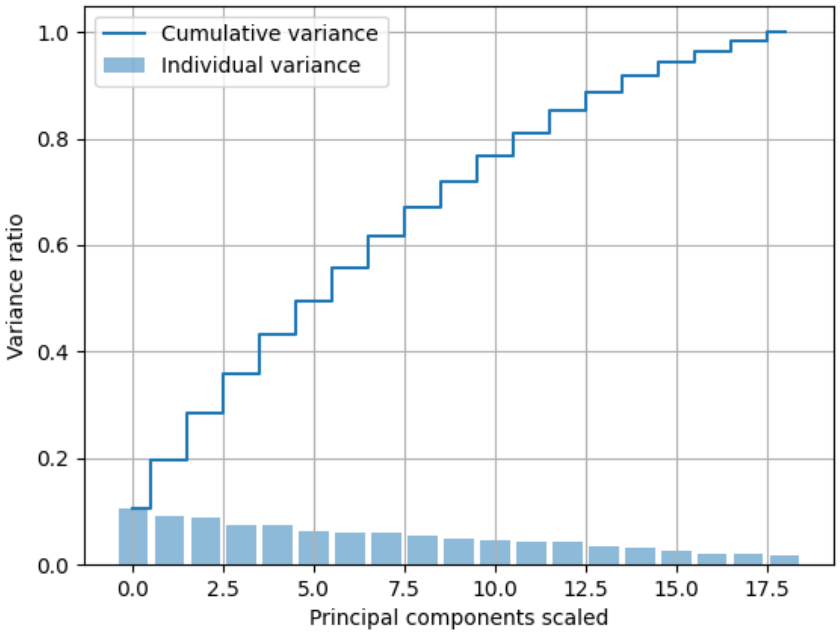

Figure 35: Varianza delle feature con PCA

Secondo il criterio precedentemente descritto sono state eliminate tutte le componenti successive al raggiungimento della varianza minima totale scelta, in particolare sono state eliminate le ultime due componenti, portando il totale di quelle mantenute a 17.

2.3 Stochastic Gradient Descent, SGD

L'algoritmo SGD è un metodo iterativo per l'ottimizzazione di funzioni differenziabili ed è ampiamente usato per il training di modelli come, nel nostro caso, quello di un classificatore logistico; il classificatore è stato allenato utilizzando la **logistic loss** e i parametri di addestramento scelti sono:

- 20 epoche di training
- vettore di step size $[0.00001, 0.0001, 0.001, 0.01, 0.1, 1]$

Attraverso la funzione **evaluate_step_size** andiamo a definire sei classificatori, ognuno con un step size diverso, ed andiamo a confrontare i valori del gradiente per epoca tra i sei classificatori. Lo scopo è scegliere quale tra gli step-size, si rivela il migliore per il nostro task, ovvero quello che ci fornisce il più alto valore di **Logistic Classifier Accuracy**. Di seguito vediamo i grafici di confronto (figure **36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47**):

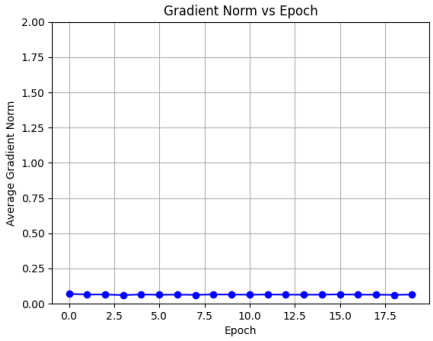

Figure 36: Gradiente con step size 1

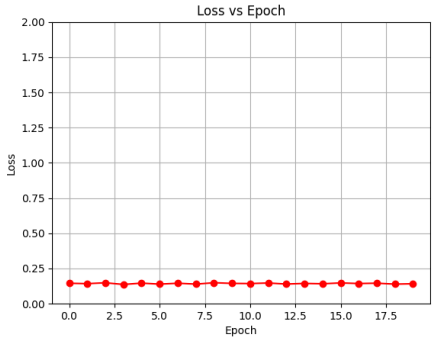

Figure 37: Loss con step size 1

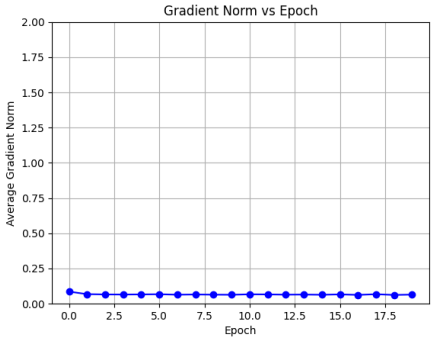

Figure 38: Gradiente con step size 0.1

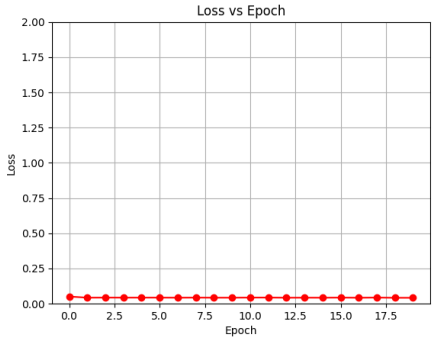

Figure 39: Loss con step size 0.1

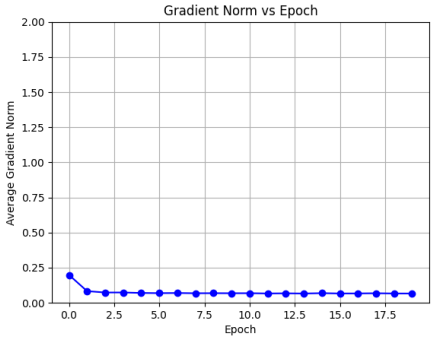

Figure 40: Gradiente con step size 0.01

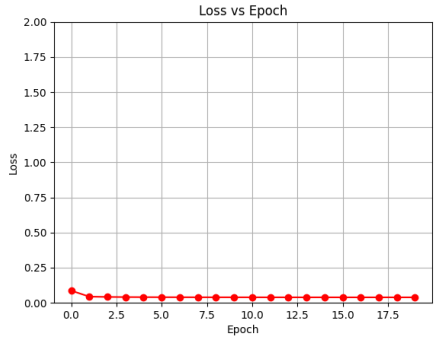

Figure 41: Loss con step size 0.01

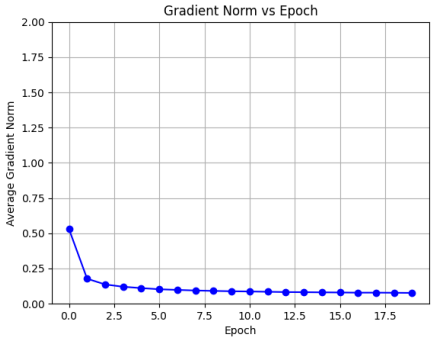

Figure 42: Gradiente con step size 0.001

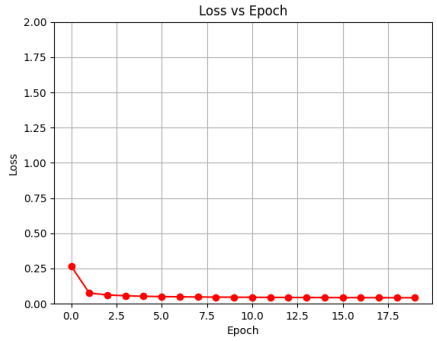

Figure 43: Loss con step size 0.001

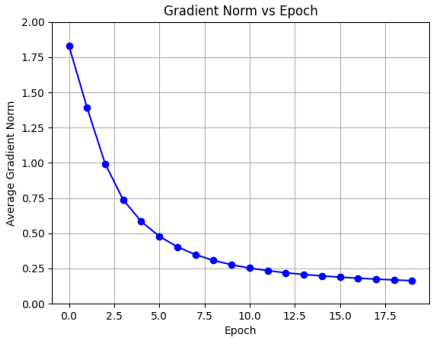

Figure 44: Gradiente con step size 0.0001

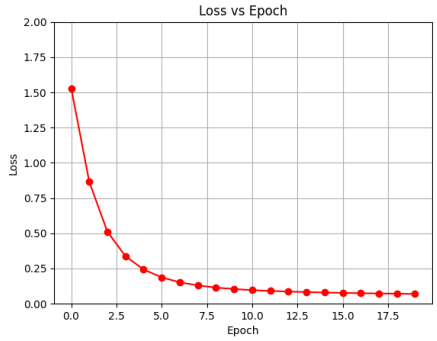

Figure 45: Loss con step size 0.0001

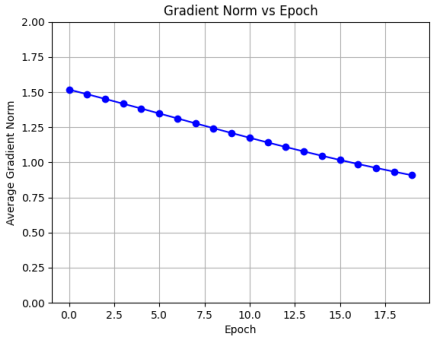

Figure 46: Gradiente con step size 1e-05

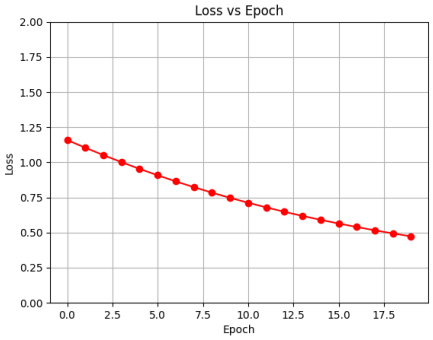

Figure 47: Loss con step size 1e-05

Al termine dell'esecuzione, il miglior step size sarà $0,0001$ con un accuracy pari a $0,991$ (figura **48**).

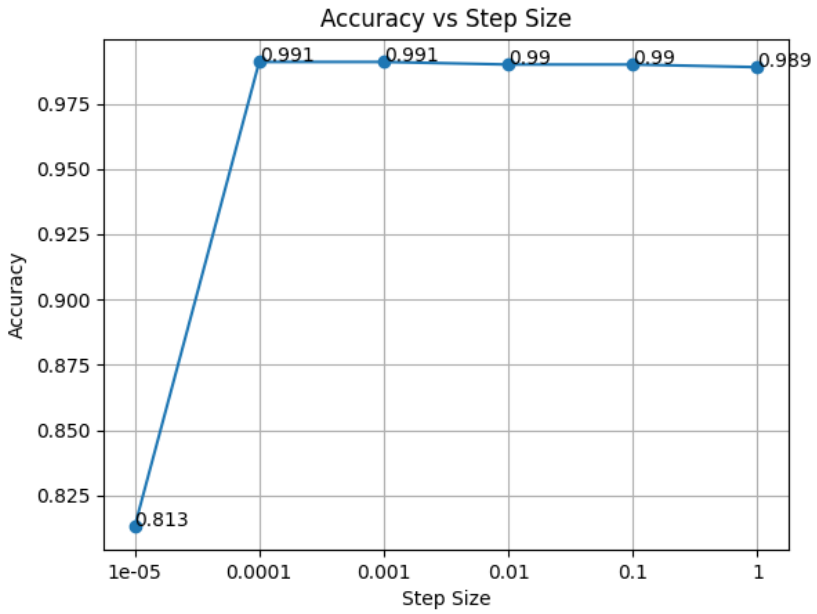

Figure 48: Confronto delle accuracy

2.4 Classificatore

Il classificatore esegue la seguente operazione:

- 1 se $X^{'}_{test}(j)\hat{\beta}(t_j) > 0$;
- -1 altrimenti;

Dove i t_j sono gli specifici istanti di tempo considerati $(t_1, ..., t_K)$, $X^{'}_{test}(j)$ è la feature di test nell'istante j-esimo e $\hat{\beta}(t_j)$ è il predittore stimato dall'algoritmo SGD al tempo t_j.
Attraverso la funzione **logistic_inference** andiamo ad applicare tale regola: Applichiamo il prodotto puntuale tra i due vettori **x_value** e **best_beta** e se il loro valore è maggiore di 0, si aggiunge il valore 1 per quel j-esimo confronto, altrimenti -1.

2.5 Conversione in stringa

Per la conversione della stringa da binaria a caratteri ASCII abbiamo utilizzato due funzioni:

- Con **decision_rule** abbiamo preso in ingresso le predictions e abbiamo convertito tutti i valori "-1" in "0" ottenendo una stringa binaria;
- Successivamente, con **to_ascii** abbiamo diviso la stringa binaria in chunk da 8 bit e poi, ognuno di essi, è stato convertito in un carattere ascii;

3 Risoluzione enigma e conclusione

Gli indizi trovati sono i seguenti:

- Indizio 1: Ghost;
- Indizio 2: AmoSa+Nnt

Unendo gli indizi, siamo risaliti alla celebre frase: **"Ti amo si dice troppo spesso, non sa più di niente"**, del film cult anni 90 **Ghost**. La Scena (dal minuto 1:46 in poi)