Computer Hardware Data Set
Report per l'Esame di Fondameti di Machine Learning

Paolo Castagnetti

matr. n. 143098
Ingegneria Informatica
267731@studenti.unimore.it

Abstract

In questo report si vogliono evidenziare i risultati ottenuti durante l'applicazione di algoritmi di regressione lineare su un dataset contenente dettagli tecnici di alcune CPU per validarne le performance e predire le performance di eventuali processori futuri.

1 Introduction

Il dataset scelto per il progetto è reperibile al link [3]. Esso contiene dati relativi alle prestazioni delle CPU, descritti in termini di tempi di ciclo, dimensioni delle memorie, ecc.
Sono presenti 209 samples e 9 features, di cui 2 con valore nominale.
Non ci sono valori mancanti all'interno del dataset.
Una veloce panoramica degli attributi per ogni CPU:

1. Vendor name
2. Model Name
3. MYCT: Machine Cycle Time in nanosecondi (int)
4. MMIN: Minimum Main mMmory in kB (int)
5. MMAX: Maximum Main Memory in kB (int)
6. CACH: Cache Memory in kB (int)
7. CHMIN: Minimum Channels (int)
8. CHMAX: maximum channels (int)
9. PRP: Published Relative Performance (int)
10. ERP: Estimated Relative Performance (int)

ERP è riferito ad una valutazione fatta in precedenza in un altro progetto. Per questo viene scartata all'inizio dell'elaborazione dei dati.
PRP invece è la variabile target che si vuole raggiungere.
Ho estratto un grafico di correlazione tra le feature ed uno di correlazione di ogni feature con la feature target (PRP) per evidenziarli singolarmente.

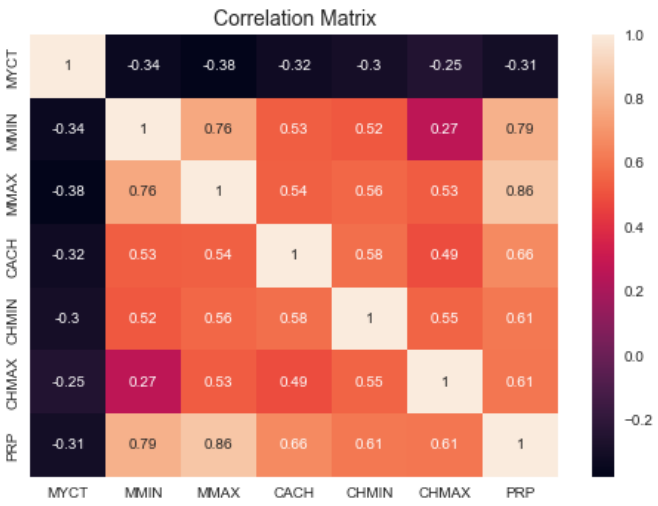

Figure 1: Correlation matrix

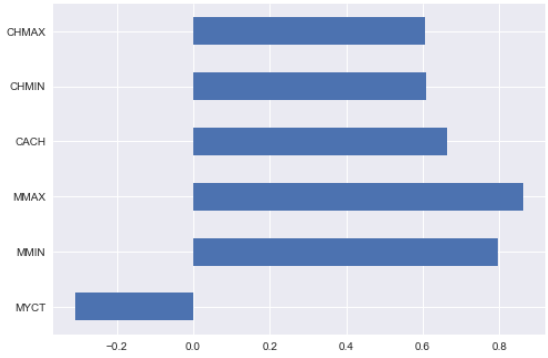

Figure 2: Correlation with target Column

2 Problem Definition and Algorithm

2.1 Task Definition

L'obiettivo è di ottenere un risultato più simile possibile al valore PRP di ogni CPU attraverso l'utilizzo di task di regressione lineare.
Per questo progetto ho scelto di utilizzare 3 modelli: Linear regression, Lasso regression e Ridge regression. E valutare quale modello si adattasse meglio ai dati forniti in input tramite 3 metriche principali: R2-score, MSE (Mean Square Error) e MAE (Mean Absolute Error).
Come prima cosa ho proceduto ad eliminare le colonne "Model name" e "ERP" essendo superflue. Successivamente ho diviso il dataset in training, validation e testing dopo aver mescolato i samples. Tenendo 90% e 10% per training e testing e il 20% della parte di training per la validation.

2.2 Algorithm Definition

LinearRegression:
"LinearRegression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation." [1].
LassoRegression:
"Linear Model trained with L1 prior as regularizer (aka the Lasso). Technically the Lasso model

is optimizing the same objective function as the Elastic Net with “l1_ratio=1.0“ (no L2 penalty).” [1].
RidgeRegression:
”Linear least squares with l2 regularization. This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression” [1].

3 Experimental Evaluation

3.1 Methodology

Inizialmente, in tutti i file, ho effettuato una standardizzazione dai dati attraverso lo standard scaler di sklearn.
Ho deciso di effettuare 3 prove differenti per la selezione delle feature. Inizialmente ho utilizzato tutte le feature numeriche presenti [’MYCT’, ’MMIN’, ’MMAX’, ’CACH’, ’CHMIN’, ’CHMAX’] per addestrare i vari modelli ed effettuare una cross-validation (file Wo_Vendor).
Successivamente, pensando che la feture Vendor potesse essere utile, ho deciso di aggiungerla nuovamente al dataset non come stringa, ma aggiungendo alla matrice contenente le feature 30 colonne (numero totale dei vendor) nominate rispettivamente col nome del vendor e contenente come dato un 1 se la CPU è di quel vendor altrimenti 0. Questo attraverso il metodo di pandas get_dummies che converte variabili categoriche in indicatori (file W_Vendor).
In terzo luogo, osservando la matrice di correlazione delle feature ho provato a rimuovere la feature MYCT (file Wo_MYCT).
Dopo la selezione delle feature, in tutti e 3 i file ho proceduto ad addestrare i modelli registrando anche il tempo di esecuzione di ogni modello.
Per la Lasso regression e la Ridge regression ho cercato il parametro alpha migliore tra un array di alpha dati, questo mi ha permesso di ottenere un fit migliore sui dati in ingresso valutando i risultati sul set di validation.

3.2 Results

Dopo il fit dei modelli predittivi ho effettuato dei test sul set di validation per avere delle metriche.

-------Model || Without Vendors Data || Result:-------

	Model	R2 Score	MSE	MAE	Fitting time
0	Linear Regression	0.8582	0.09181	0.21547	0.97036
1	Lasso Regression	0.8698	0.08433	0.19340	6.98018
2	Ridge Regression	0.8609	0.09008	0.20555	6.98137

(a) Without Vendor Data
Lasso alpha: 0.1
Ridge alpha: 10.0

-------Model || With Vendors Data || Result:-------

	Model	R2 Score	MSE	MAE	Fitting time
0	Linear Regression	0.8758	0.08046	0.20596	1.00827
1	Lasso Regression	0.8781	0.07891	0.20248	10.95891
2	Ridge Regression	0.8764	0.08004	0.20517	8.97574

(b) With Vendor Data
Lasso alpha: 0.0001
Ridge alpha: 0.01

-------Model || Without MYCT Data || Result:-------

	Model	R2 Score	MSE	MAE	Fitting time
0	Linear Regression	0.8743	0.08142	0.21405	0.99707
1	Lasso Regression	0.8767	0.07983	0.21081	10.97059
2	Ridge Regression	0.8750	0.08098	0.21330	9.97329

(c) Without MYCT Data
Lasso alpha: 0.0001
Ridge alpha: 0.01

Da questi risultati si può notare che le performance migliori si ottengono dall'utilizzo di tutte le feature numeriche più i vendor (caso b).
Possiamo poi notare che i 3 modelli predittivi hanno risultati molto simili tra loro, quello che offre performance leggermente migliori è la Lasso regression (alpha=0.0001) con R2 score di 0.8781, MSE di 0.07891 e MAE di 0.20248, a discapito della velocità di training dei dati (che è relativa, essendo effettuata anche per la ricerca del miglior iperparametro alpha).

3.3 Discussion

Avendo ora definito il modello su cui poter fare considerazioni possiamo procedere con l'utilizzo della Lasso regression e addestrarla su una percentuale maggiore di dati per poi testarla sul testing set (file Final).

4 Conclusion

Dopo aver effettuato il trainig, il risultato finale si può visualizzare con queste performance sul testing set:

```
---------------Prediction with VENDOR---------------

---------------Lasso Regression:----------------

Fitting time:  3.988981246948242  ms
R^2 score for lasso_reg testing set: 0.8952
Mean Square Error for testing set: 0.32442
Mean Absolute Error for testing set: 0.37998
```

Figure 4: Final results

Le due metriche MSE e MAE risultano più elevate rispetto che alla fase di validation probabilmente per il numero ridotto di istanze presenti nel dataset. Notiamo comunque che l'R-2 score mantiene le aspettative.

5 Related Work

Si possono trovare Report che citano questo dataset quali:

1. Scalable and Practical Probability Density Estimators for Scientific Anomaly Detection. [2]
2. A New Approach to Fitting Linear Models in High Dimensional Spaces. [4]

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] Dan Pelleg. Scalable and practical probability density estimators for scientific anomaly detection, 2004.

[3] UCI. Computer hardware data set. `https://archive.ics.uci.edu/ml/datasets/Computer+Hardware`, October 1987.

[4] Alastair Scott Yongge Wang. A new approach to fitting linear models in high dimensional spaces.