

Esempio Modulo Machine Learning

May 4, 2022

1 Esempi di Utilizzo del MODULO MACHINE LEARNING della libreria IntelligenzaArtificiale

1.1 Installare la libreria

```
[ ]: #Per installare la libreria sul tuo computer puoi usare il comando :  
pip3 install intelligenzaartificiale  
  
#se invece utilizzi google colab puoi usare:  
!pip install intelligenzaartificiale
```

1.2 Importare la libreria

```
[1]: from intelligenzaartificiale import dataset as dt  
from intelligenzaartificiale import machinelearning as ml  
  
il_mio_dataset = dt.leggi_csv("exams.csv")
```

Tempo impiegato per leggere il file: 0.0034322738647460938

1.3 Vedere se il dataset è Pulito

```
[2]: from intelligenzaartificiale import statistica as st  
  
print("\n ----- Numero Valori nulli per colonna")  
print(st.valori_nan(il_mio_dataset))  
print("\n\n ----- Dataset \n")  
print(il_mio_dataset)
```

```
----- Numero Valori nulli per colonna  
gender                0  
race/ethnicity        8  
parental level of education  7  
lunch                 1  
test preparation course  5  
math score            3  
reading score         8
```

```
writing score          2
dtype: int64
```

----- Dataset

	gender	race/ethnicity	parental level of education	lunch	\
0	male	group E	bachelor's degree	standard	
1	female	group D	some college	free/reduced	
2	male	group E	high school	free/reduced	
3	male	group C	master's degree	free/reduced	
4	male	NaN	master's degree	free/reduced	
..	
995	female	group B	high school	standard	
996	female	group C	some college	standard	
997	male	group C	some high school	free/reduced	
998	female	group D	master's degree	standard	
999	male	group A	high school	standard	

	test preparation course	math score	reading score	writing score
0	none	78.0	59.0	64.0
1	none	47.0	52.0	50.0
2	none	62.0	47.0	46.0
3	completed	55.0	65.0	68.0
4	none	61.0	54.0	55.0
..
995	none	33.0	36.0	33.0
996	none	52.0	59.0	64.0
997	none	66.0	64.0	62.0
998	completed	99.0	100.0	100.0
999	none	46.0	33.0	30.0

[1000 rows x 8 columns]

1.4 PreProcessiamo il nostro Set di Dati

```
[3]: from intelligenzaartificiale import preprocessing as pp

#leviamo i valori nulli
il_mio_dataset = pp.rimuovi_nan(il_mio_dataset)

#effettuare il labelencoding su più colonne
il_mio_dataset = pp.label_encoding_multiplo(il_mio_dataset, ["test preparation_
↳course", "gender", "race/ethnicity", "parental level of education" ,
↳"lunch"])

print(il_mio_dataset)
```

	gender	race/ethnicity	parental level of education	lunch	\
0	1	4		1	1
1	0	3		4	0
2	1	4		2	0
3	1	2		3	0
5	0	3		1	1
..
995	0	1		2	1
996	0	2		4	1
997	1	2		5	0
998	0	3		3	1
999	1	0		2	1

	test preparation course	math score	reading score	writing score
0	1	78.0	59.0	64.0
1	1	47.0	52.0	50.0
2	1	62.0	47.0	46.0
3	0	55.0	65.0	68.0
5	1	50.0	58.0	61.0
..
995	1	33.0	36.0	33.0
996	1	52.0	59.0	64.0
997	1	66.0	64.0	62.0
998	0	99.0	100.0	100.0
999	1	46.0	33.0	30.0

[966 rows x 8 columns]

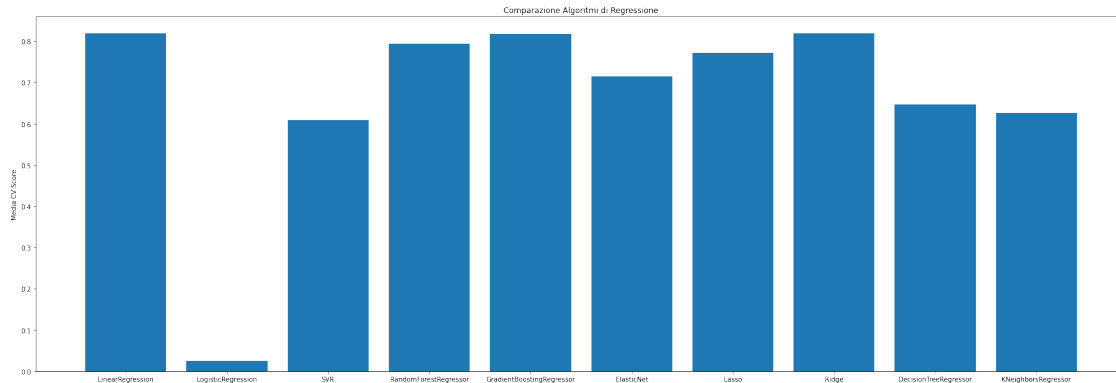
1.5 Scoprire l'algoritmo più performante

```
[4]: colonne_x = ["test preparation course", "gender", "race/ethnicity", "parental_
↳level of education" , "lunch", "reading score", "writing score"]
target = "math score"

# scoprire algoritmo di regressione più performante
modello = ml.performance_modelli_regressione(il_mio_dataset, colonne_x, target)

# scoprire algoritmo di classificazione più performante
#modello = ml.performance_modelli_classificazione(il_mio_dataset, colonne_x,
↳target)

print("il modello migliore è : " + str(modello))
```



il modello migliore è : LinearRegression()

1.6 Creare il modelli di regressione [SEMPLICE]

```
[5]: # dopo aver scoperto l'algoritmo più performante lo potrai implementare in una
      ↪ sola riga
      il_mio_modello = ml.regressione_gradient_boosting(il_mio_dataset, colonne_x,
      ↪ target)

      print("Il modello è stato allenato " + str(il_mio_modello))
```

Il modello è stato allenato GradientBoostingRegressor()

1.6.1 Lista di modelli Disponibili :

Regressori :

- `modello1 = ml.regressione_lineare(il_mio_dataset, colonne_x, target)`
- `modello2 = ml.regressione_logistica(il_mio_dataset, colonne_x, target)`
- `modello3 = ml.regressione_SVR(il_mio_dataset, colonne_x, target)`
- `modello4 = ml.regressione_SVC(il_mio_dataset, colonne_x, target)`
- `modello5 = ml.regressione_random_forest(il_mio_dataset, colonne_x, target)`
- `modello6 = ml.regressione_gradient_boosting(il_mio_dataset, colonne_x, target)`
- `modello7 = ml.regressione_decision_tree(il_mio_dataset, colonne_x, target)`
- `modello8 = ml.regressione_knn(il_mio_dataset, colonne_x, target)`
- `modello9 = ml.modello_elastic_net(il_mio_dataset, colonne_x, target)`
- `modello10 = ml.modello_lasso(il_mio_dataset, colonne_x, target)`
- `modello11 = ml.modello_ridge(il_mio_dataset, colonne_x, target)`

Classificatori :

- `modello1 = ml.classificatore_random_forest(il_mio_dataset, colonne_x, target)`
- `modello2 = ml.classificatore_gradient_boosting(il_mio_dataset, colonne_x, target)`
- `modello3 = ml.classificatore_decision_tree(il_mio_dataset, colonne_x, target)`
- `modello4 = ml.classificatore_knn(il_mio_dataset, colonne_x, target)`

- `modello5 = ml.classificatore_logistico(il_mio_dataset, colonne_x, target)`
- `modello6 = ml.classificatore_naivebayes(il_mio_dataset, colonne_x, target)`
- `modello7 = ml.classificatore_svm(il_mio_dataset, colonne_x, target)`

1.7 Valutare, Prevedere, Salvare e Caricare un modello

```
[6]: # Valutare un modello
print(ml.valutazione_modello(il_mio_modello,il_mio_dataset, colonne_x, target))

# Spiegare un modello
ml.spiega_modello(il_mio_modello,il_mio_dataset, colonne_x, target)

# Previsione con un modello
nuovo_dataset = dt.leggi_csv("dataset_test.csv")
#leviamo i valori nulli
nuovo_dataset = pp.rimuovi_nan(nuovo_dataset)
#effettuare il labelencoding su più colonne
nuovo_dataset = pp.label_encoding_multiplo(nuovo_dataset,["test preparation_
↪course", "gender", "race/ethnicity", "parental level of education" ,
↪"lunch"])

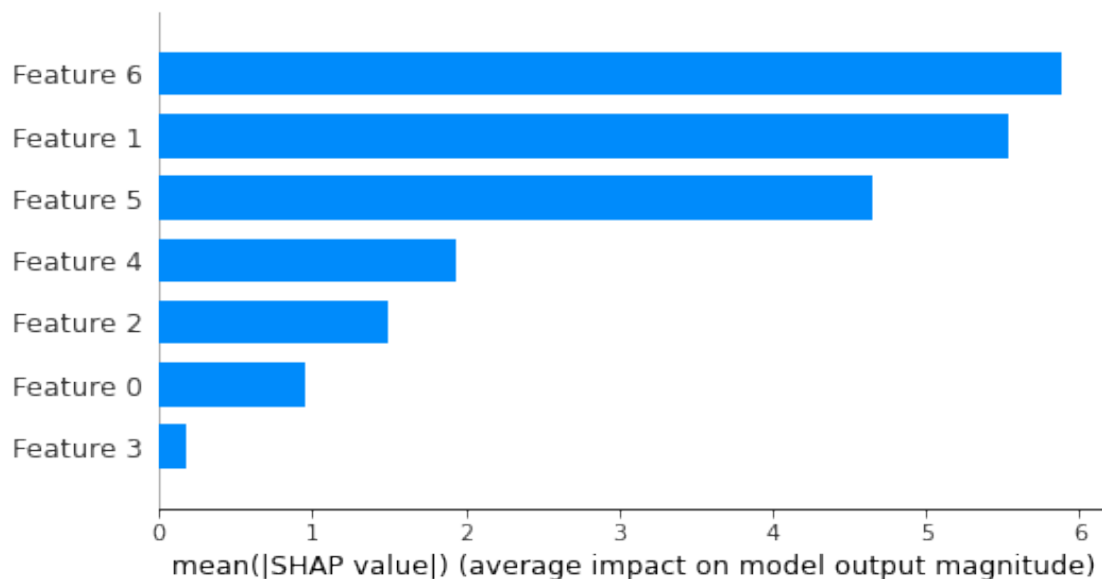
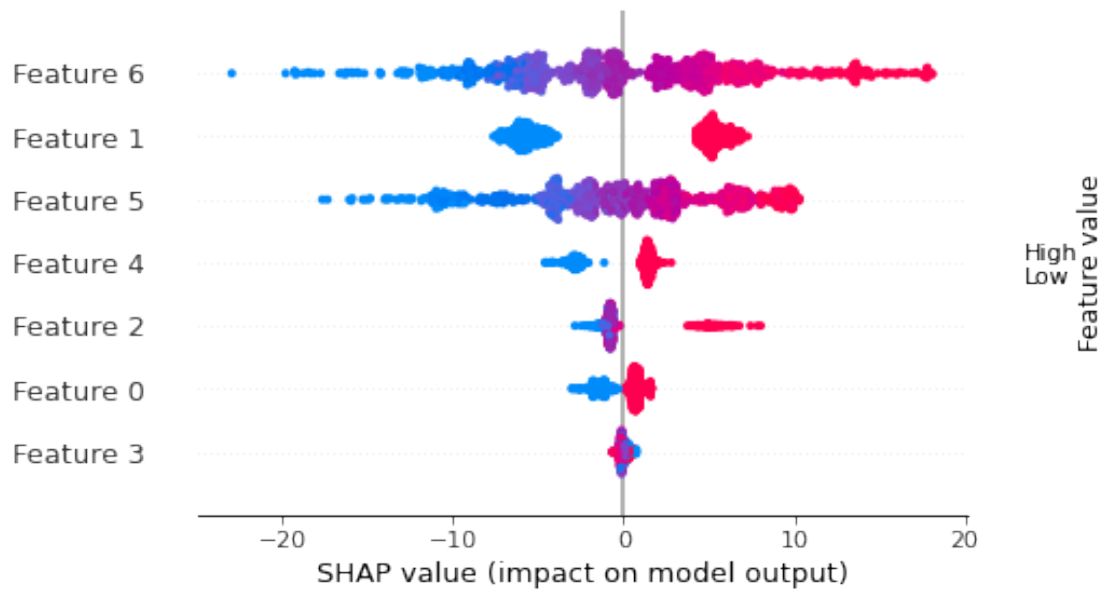
previsioni = ml.predizione_y(il_mio_modello,nuovo_dataset)
print(previsioni)

# Salvare e Caricare un modello
ml.salva_modello(il_mio_modello,"nome_modello")

vecchio_modello = ml.carica_modello("nome_modello")
```

0.8720584862249142

```
Feature 0 : test preparation course
Feature 1 : gender
Feature 2 : race/ethnicity
Feature 3 : parental level of education
Feature 4 : lunch
Feature 5 : reading score
Feature 6 : writing score
```



Tempo impiegato per leggere il file: 0.0020928382873535156

66.84271213	61.83662268	52.31382548	66.11906333	59.42655177	62.77534272
68.74875798	86.42373963	71.39188096	73.9828311	88.72600086	83.36908134
54.01600518	68.60257738	49.95320093	93.82308641	61.41654404	74.8149285
61.80346554	69.1610261	77.57524143	83.19602997	72.16056839	54.30939007
50.07943696	75.93225747	81.64408563	86.12767091	49.12534691	86.31633209
85.07435461	59.7625009	72.59924757	59.51509962	91.08612808	67.57389125
95.45736007	67.73130997	50.33766591	64.91773883	79.09882821	52.75166538

90.99868585 86.15748795 59.91171349 53.04888707 69.52448189 81.11994678
90.52348981 69.69634439 67.81845839 70.13043753 99.11857026 81.37451939
51.15747059 69.12878077 64.9219627 68.42990118 93.81985533 90.15230664
86.28322134 80.67966904 79.03667897 84.26331199 81.64408563 80.24193654
56.81337801 66.47213791 58.06671441 67.70444244 90.18778551 74.79730823
58.7771907 78.6735824 75.80291878 81.37510074 51.25464091 73.24810364
72.4639118 50.93059133]

1.8 Altre risorse

- [Documentazione Ufficiale](#)
- [Blog Ufficiale](#)
- [Corsi Gratis](#)
- [Ebook Gratis](#)
- [Progetti Python Open Source](#)
- [Dataset Pubblici](#)
- [Editor Python Online per il M.L.](#)

2 Per favore citaci se usi la Libreria.