



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Car Acceleration

Documentazione caso di studio

A.A. 2022-2023

Gruppo di lavoro:

- Luigi Fraccalvieri, MAT. 738953, l.fraccalvieri4@studenti.uniba.it
- Giuseppe Demauro, MAT. 738713, g.demauro6@studenti.uniba.it
- Pier Silvio Fatiguso, MAT. 738026, p.fatiguso@studenti.uniba.it

Repository associato:

https://github.com/luigifraccalvieri/Car_Acceleration

Indice

Introduzione.....	3
Elenco argomenti di interesse	3
Descrizione del dominio e dei dati.....	3
Preprocessing dei dati.....	4
Eliminazione di features.....	4
Rifinitura dei dati	4
Clustering	5
Creazione e utilizzo Knowledge Base.....	6
Individui.....	7
Proprietà.....	7
Clausole definite	7
Clausole rapporti	7
Clausole di categorizzazione.....	8
Clausole ausiliarie	9
Query	10
Regressione tempo di accelerazione	10
Introduzione.....	10
Approcci seguiti.....	11
KNN	11
Valutazione dataset.....	12
AdaBoost.....	15
Valutazione dataset.....	16
Random forest	18
Valutazione dataset.....	19
Alberi di decisione.....	23
Valutazione dataset.....	24
Valutazione conclusiva	28

Introduzione

Il caso di studio in analisi è stato sviluppato tramite il linguaggio Python, esso ha come scopo l'effettuazione di operazioni di regressione sul tempo impiegato da un'auto per accelerare da 0 a 100 km/h.

È stato utilizzato un dataset di modelli di auto prodotte dal 1945 al 2020 con relative caratteristiche, tra cui l'accelerazione precedentemente citata.

Dopo una serie di azioni di pre-processing e di clusterizzazione, sono stati integrati i dati in una base di conoscenza ingegnerizzando clausole Prolog che hanno permesso di inferire nuove informazioni attraverso il ragionamento automatico da esse prodotto. Il risultato è stato sfruttato in diverse configurazioni di dataset per valutare l'accuratezza delle predizioni dei regressori scelti.

Elenco argomenti di interesse

- **Apprendimento non supervisionato:** implementazione del k-means per l'individuazione del cluster di appartenenza di ogni occorrenza del dataset al fine di creare una nuova caratteristica indicante il cluster individuato che può potenzialmente migliorare l'accuratezza delle predizioni
- **Rappresentazione e ragionamento relazionale:** implementazione di clausole Prolog per effettuare ragionamento su una base di conoscenza ingegnerizzata a partire dai dati presenti nel dataset che permette di inferire nuove informazioni
- **Apprendimento supervisionato:** regressione effettuata tramite modelli di apprendimento quali KNN, Alberi di decisione e metodi di ensemble learning quali Random Forest e AdaBoost

Descrizione del dominio e dei dati

L'idea del caso di studio sfrutta un dataset trovato su Kaggle ovvero [Car specification dataset 1945 2020](#), contenente modelli di auto prodotti dal 1945 al 2020 con diverse caratteristiche tecniche fisiche dal significato intuitivo che influenzano l'accelerazione, maggiormente di tipo continuo, tra cui:

- length_mm (lunghezza dell'auto in mm)
- width_mm (larghezza dell'auto in mm)
- height_mm (altezza dell'auto in mm)
- maximum_torque_n_m (coppia massima del motore)

- cylinder_layout (disposizione dei cilindri del motore)
- injection_type (tipo di iniezione del motore)
- number_of_cylinders (numero di cilindri del motore)
- engine_hp (numero indicante i cavalli del motore)
- boost_type (indica, se presente il turbo, di che tipo è)
- transmission (indica il tipo di trasmissione manuale, automatica o sequenziale)

ed altre.

Preprocessing dei dati

Vista la sparsità del dataset trovato e anche la numerosità delle features (78) si è deciso di effettuare operazioni di pre-processing prima di procedere con il clustering al fine di alleggerire il dataset.

Eliminazione di features

La prima operazione che si è effettuata riguarda l'eliminazione di diverse colonne che presentavano molte occorrenze vuote o che non avevano impatto sul tempo di accelerazione:

```
[ 'id_trim', 'Trim', 'Body_type', 'load_height_mm', 'number_of_seats',
  'wheel_size_r14', 'ground_clearance_mm', 'trailer_load_with_brakes_kg',
  'payload_kg', 'back_track_width_mm', 'front_track_width_mm', 'clearance_mm',
  'full_weight_kg', 'front_rear_axle_load_kg', 'cargo_compartment_length_width_height_mm',
  'cargo_volume_m3', 'overhead_camshaft', 'compression_ratio', 'engine_placement',
  'cylinder_bore_and_stroke_cycle_mm', 'max_power_kw', 'presence_of_intercooler',
  'bore_stroke_ratio', 'turning_circle_m', 'mixed_fuel_consumption_per_100_km_l',
  'range_km', 'emission_standards', 'CO2_emissions_g/km', 'rear_brakes', 'front_brakes',
  'steering_type', 'car_class', 'country_of_origin', 'number_of_doors', 'safety_assessment',
  'rating_name', 'battery_capacity_KW_per_h', 'electric_range_km', 'charging_time_h']
```

Rifinitura dei dati

Si è proceduto con le seguenti rifiniture:

- Sostituzione dei valori vuoti nelle occorrenze della feature boost_type con la stringa 'none'
- Uniformazione dei valori 'petrol' e 'Gas (Gasoline)' in 'Gasoline' nelle occorrenze della feature engine_type

- Conversione del separatore tra unità e decimali dal simbolo virgola al simbolo punto per facilitare le procedure di acquisizione dei dati in tutte le occorrenze delle features continue
- Eliminazione delle occorrenze del dataset con almeno un campo vuoto
- Eliminazione delle occorrenze duplicate

Rifiniture applicate per eliminare problemi di sintassi presenti nell'integrazione dei dati nella base di conoscenza Prolog:

- Sostituzione degli spazi in trattini bassi e trasformazione in minuscolo di tutti i caratteri (uniformazione allo standard snake case) in tutte le occorrenze delle features categoriche
- Apposizione del prefisso 'h_' nelle occorrenze delle features generation, model, fuel_grade, series
- Rimozione delle parentesi quadre nelle occorrenze della feature generation
- Sostituzione dei punti in trattini bassi nelle occorrenze delle features generation e series
- Eliminazione dei punti nelle occorrenze della feature model
- Rimozione delle parentesi tonde, sostituzione dei trattini in trattini bassi, della virgola in '_and', del carattere '+' in '_and', del carattere '/' in '_', rimozione del carattere apostrofo e rimozione del carattere virgolette in tutte le occorrenze del dataset
- Aggiunta della caratteristica contenente un numero intero autoincrementale ID

Le precedenti azioni hanno originato il dataset **preprocessed.csv**, il primo dei quattro utilizzati per effettuare la valutazione dell'accuratezza delle predizioni dei regressori.

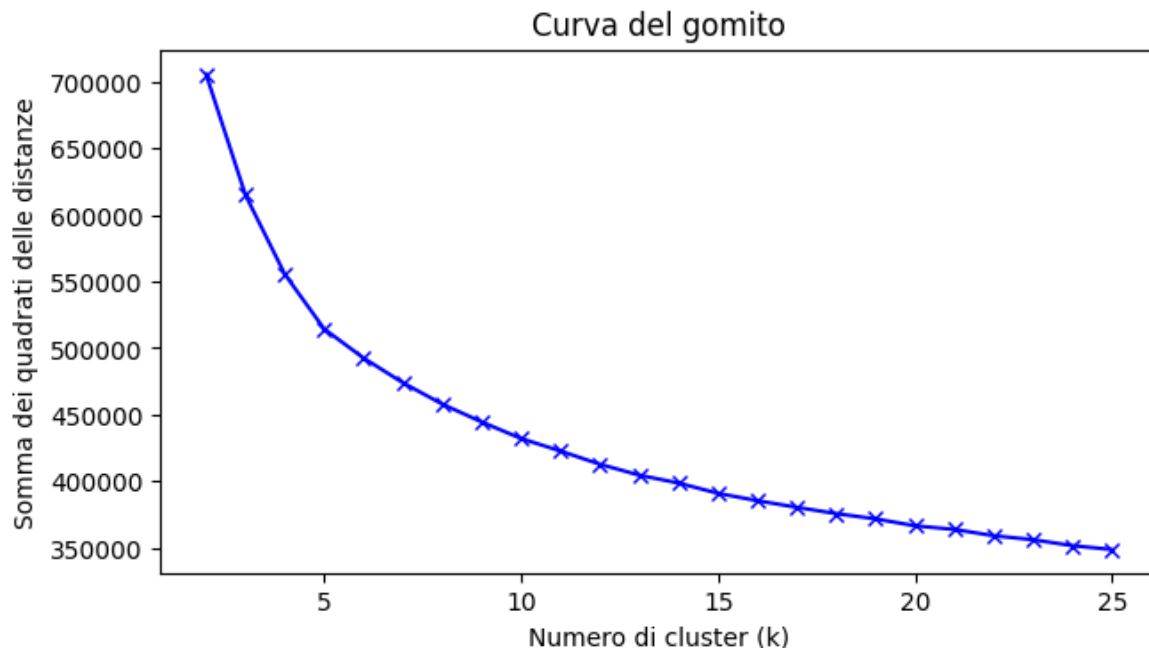
Clustering

In questa fase si procede con la creazione del secondo dataset che aggiunge al dataset prodotto nella fase precedente una nuova caratteristica indicante il cluster individuato di appartenenza per ogni occorrenza.

Tra i vari algoritmi di clustering disponibili si è scelto di utilizzare il K-Means sfruttando la distanza euclidea.

La feature ID è stata esclusa dalla clusterizzazione non essendo certamente un dato funzionale.

L'obiettivo primario è l'individuazione del numero ottimale di clusters da utilizzare; pertanto, per effettuare quest'analisi è stata implementata la [tecnica della curva del gomito](#) ed il range valutato va da 2 a 25.



Analiticamente si è evinto che il gomito della curva è in corrispondenza di un numero di cluster pari a 8.

A questo punto, individuato il numero ottimale di cluster, esso viene utilizzato per il calcolo del cluster di appartenenza per ogni occorrenza del dataset ed inserito nella caratteristica corrispondente.

Si vuole valutare se quest'informazione aggiuntiva (il cluster di appartenenza) può aiutare o meno nella regressione, è stato quindi generato il secondo dataset utilizzato nella valutazione chiamato **clustered.csv**.

Creazione e utilizzo Knowledge Base

È stata creata una Knowledge Base sfruttando il linguaggio Prolog attraverso la libreria PySwip importata nel linguaggio Python, essa ha permesso di inferire nuova conoscenza utile alla creazione degli ultimi due dataset utilizzati nella valutazione dell'apprendimento automatico sfruttando il ragionamento automatico e la struttura di individui e relazioni.

Individui

La Knowledge Base è stata popolata con fatti provenienti direttamente dal dataset preprocessato modellando il dominio, individuando la classe di individui con il simbolo di funzione associato **car(C)** che rappresenta l'auto il cui ID è C.

Proprietà

Le proprietà associate all'individuo `car(C)` corrispondono alle sole feature presenti nel dataset preprocessato che sono effettivamente utilizzate nel ragionamento automatico per inferire nuove informazioni:

- `curb_weight_kg(car(C), K)`
- `engine_hp(car(C), H)`
- `height_mm(car(C), HeightMm)`
- `width_mm(car(C), WidthMm)`
- `capacity_cm3(car(C), Cap)`
- `engine_hp(car(C), H)`
- `number_of_cylinders(car(C), N)`
- `max_speed_km_per_h(car(C), M)`
- `make(car(C), Manufacturer)`
- `max_trunk_capacity_l(car(C), Capacity)`
- `model(car(C), MaxModel)`
- `year_from(car(C), Year)`
- `cylinder_layout(car(C), Cylinder_layout)`

Clausole definite

In questa sezione sono state riportate per esteso tutte le clausole definite scritte per effettuare ragionamento con gli schemi adottati per la loro definizione.

Clausole rapporti

La seguente clausola consente di calcolare il rapporto peso/potenza di una determinata macchina

*weight_to_power_ratio(car(C), W) :- curb_weight_kg(car(C), K),
engine_hp(car(C), H), W is K / H*

La seguente clausola consente di calcolare il coefficiente aerodinamico di un'auto prendendo in considerazione caratteristiche fisiche dell'auto e fissando una densità dell'aria pari a 1.2 kg/m³ e una velocità pari a 100 km/h.

*drag_coefficient(car(C), DragCoefficient) :- height_mm(car(C), HeightMm),
width_mm(car(C), WidthMm), FrontalAreaM2 is WidthMm * HeightMm / 1000000,
AirDensityKgM3 is 1.2, VelocityKmh is 100, DragCoefficient is 2 * 450 / (AirDensityKgM3 *
FrontalAreaM2 * (VelocityKmh / 3.6)^2)*

La seguente clausola consente di calcolare la potenza specifica di un'auto (ovvero il rapporto cavalli/cilindrata)

*specific_power(car(C), S) :- capacity_cm3(car(C), Cap), engine_hp(car(C), H), S is H /
Cap*

La seguente clausola consente di calcolare il rapporto lunghezza/altezza di un'auto

*length_ratio(car(C), L) :- length_mm(car(C), LengthMm),
height_mm(car(C), HeightMm), L is LengthMm / HeightMm*

La seguente clausola consente di calcolare la capacità del singolo cilindro

*number_cylinder_capacity_ratio(car(C), Cr) :-
number_of_cylinders(car(C), N), capacity_cm3(car(C), Cap), Cr is N /
Cap*

La seguente clausola consente di calcolare il rapporto lunghezza/potenza di un'auto

*length_power_ratio(car(C), Lpr) :- length_mm(car(C), L),
engine_hp(car(C), H), Lpr is L / H*

La seguente clausola consente di calcolare il rapporto velocità massima/potenza di un'auto

*max_speed_power_ratio(car(C), Spr) :-
max_speed_km_per_h(car(C), M), engine_hp(car(C), H), Spr is M / H*

Clausole di categorizzazione

La seguente clausola consente di classificare la potenza di un'auto in "bassa", "media" o "alta" sfruttando la clausola ausiliaria "horsepower_fuzzy"

*power(car(C), Potenza) :- engine_hp(car(C), HP),
horsepower_fuzzy(HP, Potenza), !*

La seguente clausola consente di classificare la forma di un'auto in "sportiva", "coupé", "berlina" o "suv" sfruttando la clausola ausiliaria "forma"

```
shape(car(C), Forma) :- height_mm(car(C), Height),
width_mm(car(C), Width), forma(Forma, Height, Width), !
```

La seguente clausola consente di calcolare la capacità media del bagagliaio delle auto di un certo brand

```
avg_trunk_capacity(car(C), AvgCapacity) :- make(car(C),
Manufacturer), findall(Capacity, (make(car(D), Manufacturer),
max_trunk_capacity_l(car(D), Capacity)), Capacities),
sumlist(Capacities, Sum), length(Capacities, Length), Length > 0,
AvgCapacity is Sum / Length
```

La seguente clausola consente di calcolare il modello più potente di un certo brand

```
most_powerful_car(car(C), MaxModel) :- make(car(C), Manufacturer),
findall(Power, (make(car(D), Manufacturer), engine_hp(car(D),
Power))), Powers, length(Powers, NumPowers), NumPowers > 0,
max_list(Powers, MaxPower), make(car(E), Manufacturer),
engine_hp(car(E), MaxPower), model(car(E), MaxModel)
```

La seguente clausola restituisce la tipologia di layout dei cilindri motore più popolare con cadenza triennale (i trienni sono fissati in modo che il primo dei tre anni sia un multiplo di tre e gli altri due no) fissato l'anno di inizio produzione dell'automobile car(C) presa in considerazione e posto in Year

```
most_common_cylinder_layout(car(C), MostCommonCylinderLayout)
:- year_from(car(C), Year), YearGroup is floor(Year / 3),
findall(CylinderLayout, (year_from(car(D), YearD), YearGroupD is
floor(YearD / 3), YearGroup == YearGroupD, cylinder_layout(car(D),
CylinderLayout))), CylinderLayoutList,
most_frequent_element(CylinderLayoutList,
MostCommonCylinderLayout)
```

Clausole ausiliarie

Le seguenti clausole restituiscono "alta", "media" o "bassa" a seconda del valore passato come cavalli

```
horsepower_fuzzy(HP, bassa) :- HP < 150, !
horsepower_fuzzy(HP, media) :- HP >= 150, HP < 250, !
```

horsepower_fuzzy(HP, alta) :- HP >= 250, !

Le seguenti clausole restituiscono la forma dell'auto a seconda dei valori passati come altezza e larghezza

forma(sportiva, Height, Width) :- Height =< 1200, Width >= 1350, !

forma(coup , Height, Width) :- Height > 1200, Height =< 1400, Width >= 1350, !

forma(berlina, Height, Width) :- Height >= 1400, Height < 1500, Width >= 1350, !

forma(suv, Height, Width) :- Height >= 1500, Width >= 1350, !

La seguente clausola restituisce l'elemento pi  frequente all'interno di una lista

most_frequent_element(List, Element) :- findall(Count-X, ([bagof](#)(true, [member](#)(X, List), Xs), length(Xs, Count)), Pairs), [keysort](#)(Pairs, SortedPairs), [reverse](#)(SortedPairs, [Count-Element|_]))

Query

Definite le clausole in Prolog, si   proceduto effettuando query finalizzate al calcolo dei valori da inserire in ogni occorrenza delle nuove feature create, sostituendo di volta in volta la variabile C contenuta all'interno di "car(C)" presente in ogni clausola sfruttata con l'ID dell'auto.

Attraverso le suddette query si sono creati due nuovi dataset:

- **prolog_generated.csv** contenente le sole feature generate in questa fase
- **clustered_plus_prolog_generated.csv** contenente le feature presenti nel dataset "clustered.csv" unite a quelle generate in questa fase

Regressione tempo di accelerazione

Introduzione

Il target di regressione individuato   la feature "acceleration_0_100_kmh_s".

Sono stati a questo punto messe a confronto le accurat zze di vari modelli d'apprendimento su tutti i dataset generati (con opportune modifiche ai dataset laddove necessari per il corretto funzionamento del modello interessato).

Approcci seguiti

Utilizzando la libreria python **scikit-learn** è stato effettuato il training dei regressori scelti (KNN, AdaBoost, Random Forest e Alberi di decisione) per ognuno dei quattro dataset precedentemente generati.

La prima fase che si è svolta per ogni regressore è stata l'encoding delle feature categoriche (variabili a seconda del dataset utilizzato) sfruttando la classe **OrdinalEncoder**.

Dopodiché si sono definiti gli iperparametri da ottimizzare per ogni tipologia di regressore con i relativi valori e range da testare.

L'apprendimento di tutti i modelli ha seguito lo schema di 10-fold cross-validation eseguito per ogni diversa configurazione di iperparametri del modello interessato e le metriche di valutazione riportate sono tutte mediate sui 10 modelli appresi (per ciascuna combinazione di iperparametri). Laddove possibile (AdaBoost, Alberi di decisione e Random Forest) è stato impostato il parametro `random_state` pari a 0 per avere un comportamento deterministico.

Le metriche di scoring usate scelte sono le seguenti:

- `neg_mean_absolute_error`
- `neg_mean_squared_error`
- `max_error`

Infine, per avere una maggior comprensione dei fattori che influenzano la variabile target, si è calcolata la feature importance sul modello migliore (scelto sulla base del `neg_mean_squared_error`) al variare del regressore e del dataset utilizzato.

KNN

Il primo regressore analizzato attraverso i quattro dataset precedentemente ottenuti è il **K-Nearest-Neighbors**.

Con l'uso della 10-fold Cross-validation si sono misurate le prestazioni del regressore al variare di tutte le combinazioni possibili dei suoi iperparametri con i seguenti valori testati:

- **n_neighbors**: i valori testati per questo iperparametro sono 3, 5 e 7, questo rappresenta il valore di k
- **weights**: i valori testati per questo iperparametro sono uniform e distance, questo rappresenta la funzione di peso da usare nel calcolo delle distanze, con uniform viene assegnato lo stesso peso a tutti i vicini, con distance viene assegnato un peso maggiore ai vicini più vicini

- **algorithm:** il valore testato per questo iperparametro è kd-tree, con questo iperparametro si specifica il tipo di algoritmo usato per trovare i k punti più vicini
- **p:** il valore testato per questo iperparametro è 2, questo iperparametro specifica l'utilizzo della distanza euclidea per calcolare i punti k più vicini
- **leaf_size:** i valori testati per questo iperparametro sono 20, 30 e 40, questo iperparametro specifica la massima dimensione delle foglie dell'albero kd-tree

Qui di seguito si riportano per ogni dataset le migliori cinque combinazioni di iperparametri, poiché riportarle tutte e diciotto risulterebbe poco leggibile.

Valutazione dataset

preprocessed.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{algorithm: 'kd_tree', leaf_size: 20, n_neighbors: 7, p: 2, weights: 'distance'}	-1,49723788	1	-4,497272108	1	-14,03864	7
{algorithm: 'kd_tree', leaf_size: 30, n_neighbors: 7, p: 2, weights: 'distance'}	-1,49723788	1	-4,497272108	1	-14,03864	7
{algorithm: 'kd_tree', leaf_size: 40, n_neighbors: 7, p: 2, weights: 'distance'}	-1,497239122	3	-4,497273966	3	-14,03864	7
{algorithm: 'kd_tree', leaf_size: 20, n_neighbors: 7, p: 2, weights: 'uniform'}	-1,505717679	4	-4,53561283	4	-14,06429	10
{algorithm: 'kd_tree', leaf_size: 30, n_neighbors: 7, p: 2, weights: 'uniform'}	-1,505717679	4	-4,53561283	4	-14,06429	10
{algorithm: 'kd_tree', leaf_size: 40, n_neighbors: 7, p: 2, weights: 'uniform'}	-1,505719637	6	-4,535617538	6	-14,06429	10
{algorithm: 'kd_tree', leaf_size: 20, n_neighbors: 5, p: 2, weights: 'distance'}	-1,518096229	7	-4,600525656	7	-13,84469	1
{algorithm: 'kd_tree', leaf_size: 30, n_neighbors: 5, p: 2, weights: 'distance'}	-1,518096229	7	-4,600525656	7	-13,84469	1
{algorithm: 'kd_tree', leaf_size: 40, n_neighbors: 5, p: 2, weights: 'distance'}	-1,518099004	9	-4,600534088	9	-13,84469	1
{algorithm: 'kd_tree', leaf_size: 20, n_neighbors: 5, p: 2, weights: 'uniform'}	-1,524418456	10	-4,629300419	10	-13,852	4
{algorithm: 'kd_tree', leaf_size: 30, n_neighbors: 5, p: 2, weights: 'uniform'}	-1,524418456	10	-4,629300419	10	-13,852	4
{algorithm: 'kd_tree', leaf_size: 40, n_neighbors: 5, p: 2, weights: 'uniform'}	-1,524421197	12	-4,629308843	12	-13,852	4
{algorithm: 'kd_tree', leaf_size: 20, n_neighbors: 3, p: 2, weights: 'distance'}	-1,560867656	13	-4,829863094	13	-14,22068	13
{algorithm: 'kd_tree', leaf_size: 30, n_neighbors: 3, p: 2, weights: 'distance'}	-1,560867656	13	-4,829863094	13	-14,22068	13
{algorithm: 'kd_tree', leaf_size: 40, n_neighbors: 3, p: 2, weights: 'distance'}	-1,560873322	15	-4,829864407	15	-14,22068	13
{algorithm: 'kd_tree', leaf_size: 20, n_neighbors: 3, p: 2, weights: 'uniform'}	-1,564776763	16	-4,846358234	16	-14,24667	16
{algorithm: 'kd_tree', leaf_size: 30, n_neighbors: 3, p: 2, weights: 'uniform'}	-1,564776763	16	-4,846358234	16	-14,24667	16
{algorithm: 'kd_tree', leaf_size: 40, n_neighbors: 3, p: 2, weights: 'uniform'}	-1,564782854	18	-4,846359046	18	-14,24667	16

Importanza delle feature:

1. max_speed_km_per_h: 0.3741932500103024
2. engine_hp: 0.16522704958788387
3. maximum_torque_n_m: 0.09137648308043811
4. number_of_gears: 0.03611993179591277
5. rear_track_mm: 0.03345764508316736
6. number_of_cylinders: 0.027630012841870204
7. front_track_mm: 0.026811846397810448
8. capacity_cm3: 0.024573592127084192
9. curb_weight_kg: 0.02438423132235327
10. width_mm: 0.022597342651456666

clustered.csv

params	mean_test_neg _mean_absolute _error	rank_test_neg_mea n_absolute_error	mean_test_neg _mean_square _error	rank_test_neg _mean_squar ed_error	mean_test_ max_error	rank_test _max_err or
{'algorithm': 'kd_tree', 'leaf_size': 20; 'n_neighbors': 7; 'p': 2; 'weights': 'distance'}	-1,497256101	1	-4,497270491	1	-14,03863546	7
{'algorithm': 'kd_tree', 'leaf_size': 30; 'n_neighbors': 7; 'p': 2; 'weights': 'distance'}	-1,497256101	1	-4,497270491	1	-14,03863546	7
{'algorithm': 'kd_tree', 'leaf_size': 40; 'n_neighbors': 7; 'p': 2; 'weights': 'distance'}	-1,497256628	3	-4,497271132	3	-14,03863546	7
{'algorithm': 'kd_tree', 'leaf_size': 20; 'n_neighbors': 7; 'p': 2; 'weights': 'uniform'}	-1,505740521	4	-4,535626078	4	-14,06428571	10
{'algorithm': 'kd_tree', 'leaf_size': 30; 'n_neighbors': 7; 'p': 2; 'weights': 'uniform'}	-1,505740521	4	-4,535626078	4	-14,06428571	10
{'algorithm': 'kd_tree', 'leaf_size': 40; 'n_neighbors': 7; 'p': 2; 'weights': 'uniform'}	-1,505741173	6	-4,535628772	6	-14,06428571	10
{'algorithm': 'kd_tree', 'leaf_size': 20; 'n_neighbors': 5; 'p': 2; 'weights': 'distance'}	-1,518014744	7	-4,60014058	7	-13,8446867	1
{'algorithm': 'kd_tree', 'leaf_size': 30; 'n_neighbors': 5; 'p': 2; 'weights': 'distance'}	-1,518014744	7	-4,60014058	7	-13,8446867	1
{'algorithm': 'kd_tree', 'leaf_size': 40; 'n_neighbors': 5; 'p': 2; 'weights': 'distance'}	-1,518017518	9	-4,600149012	9	-13,8446867	1
{'algorithm': 'kd_tree', 'leaf_size': 20; 'n_neighbors': 5; 'p': 2; 'weights': 'uniform'}	-1,524331658	10	-4,628886988	10	-13,852	4
{'algorithm': 'kd_tree', 'leaf_size': 30; 'n_neighbors': 5; 'p': 2; 'weights': 'uniform'}	-1,524331658	10	-4,628886988	10	-13,852	4
{'algorithm': 'kd_tree', 'leaf_size': 40; 'n_neighbors': 5; 'p': 2; 'weights': 'uniform'}	-1,524334399	12	-4,628895412	12	-13,852	4
{'algorithm': 'kd_tree', 'leaf_size': 40; 'n_neighbors': 3; 'p': 2; 'weights': 'distance'}	-1,560950952	15	-4,830318739	13	-14,22067254	13
{'algorithm': 'kd_tree', 'leaf_size': 20; 'n_neighbors': 3; 'p': 2; 'weights': 'distance'}	-1,560948253	13	-4,830321741	14	-14,22067254	13
{'algorithm': 'kd_tree', 'leaf_size': 30; 'n_neighbors': 3; 'p': 2; 'weights': 'distance'}	-1,560948253	13	-4,830321741	14	-14,22067254	13
{'algorithm': 'kd_tree', 'leaf_size': 40; 'n_neighbors': 3; 'p': 2; 'weights': 'uniform'}	-1,564862037	18	-4,846792121	16	-14,24666667	16
{'algorithm': 'kd_tree', 'leaf_size': 20; 'n_neighbors': 3; 'p': 2; 'weights': 'uniform'}	-1,564858992	16	-4,846795775	17	-14,24666667	16
{'algorithm': 'kd_tree', 'leaf_size': 30; 'n_neighbors': 3; 'p': 2; 'weights': 'uniform'}	-1,564858992	16	-4,846795775	17	-14,24666667	16

Importanza delle feature:

1. max_speed_km_per_h: 0.37398403566234667
2. engine_hp: 0.16513466986312073
3. maximum_torque_n_m: 0.09132539378012168
4. number_of_gears: 0.036099736861934996
5. rear_track_mm: 0.03343893865433677
6. number_of_cylinders: 0.027614564687419102
7. front_track_mm: 0.02679685568656311
8. capacity_cm3: 0.024559852841157317
9. curb_weight_kg: 0.024370597909520814
10. width_mm: 0.022584708301937463

clustered_plus_prolog_generated.csv

params	mean_test_neg _mean_absolut e_error	rank_test t_neg_ mean_a bsolute_ error	mean_test_neg _mean_squared _error	rank_test t_neg_ mean_s quared_ error	mean_test_max _error	rank_test t_max_ error
{'algorithm': 'kd_tree', 'leaf_size': 20, 'n_neighbors': 7, 'p': 2, 'weights': 'distance'}	-1,509837733	1	-4,573551172	1	-14,41096821	1
{'algorithm': 'kd_tree', 'leaf_size': 30, 'n_neighbors': 7, 'p': 2, 'weights': 'distance'}	-1,509837733	1	-4,573551172	1	-14,41096821	1
{'algorithm': 'kd_tree', 'leaf_size': 40, 'n_neighbors': 7, 'p': 2, 'weights': 'distance'}	-1,509845431	3	-4,57356107	3	-14,41096821	1
{'algorithm': 'kd_tree', 'leaf_size': 20, 'n_neighbors': 7, 'p': 2, 'weights': 'uniform'}	-1,517816093	4	-4,609022018	4	-14,41142857	4
{'algorithm': 'kd_tree', 'leaf_size': 30, 'n_neighbors': 7, 'p': 2, 'weights': 'uniform'}	-1,517816093	4	-4,609022018	4	-14,41142857	4
{'algorithm': 'kd_tree', 'leaf_size': 40, 'n_neighbors': 7, 'p': 2, 'weights': 'uniform'}	-1,51782523	6	-4,609035443	6	-14,41142857	4
{'algorithm': 'kd_tree', 'leaf_size': 40, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}	-1,530789248	7	-4,71158243	7	-14,89490388	10
{'algorithm': 'kd_tree', 'leaf_size': 20, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}	-1,530796867	8	-4,711588986	8	-14,89490388	10
{'algorithm': 'kd_tree', 'leaf_size': 30, 'n_neighbors': 5, 'p': 2, 'weights': 'distance'}	-1,530796867	8	-4,711588986	8	-14,89490388	10
{'algorithm': 'kd_tree', 'leaf_size': 40, 'n_neighbors': 5, 'p': 2, 'weights': 'uniform'}	-1,536940429	10	-4,741945197	10	-14,894	7
{'algorithm': 'kd_tree', 'leaf_size': 20, 'n_neighbors': 5, 'p': 2, 'weights': 'uniform'}	-1,536948652	11	-4,741952342	11	-14,894	7
{'algorithm': 'kd_tree', 'leaf_size': 30, 'n_neighbors': 5, 'p': 2, 'weights': 'uniform'}	-1,536948652	11	-4,741952342	11	-14,894	7
{'algorithm': 'kd_tree', 'leaf_size': 20, 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}	-1,571209255	13	-4,943421682	13	-15,4613417	16
{'algorithm': 'kd_tree', 'leaf_size': 30, 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}	-1,571209255	13	-4,943421682	13	-15,4613417	16
{'algorithm': 'kd_tree', 'leaf_size': 40, 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}	-1,571214897	15	-4,943425549	15	-15,4613417	16
{'algorithm': 'kd_tree', 'leaf_size': 20, 'n_neighbors': 3, 'p': 2, 'weights': 'uniform'}	-1,575096239	16	-4,960702248	16	-15,45666667	13
{'algorithm': 'kd_tree', 'leaf_size': 30, 'n_neighbors': 3, 'p': 2, 'weights': 'uniform'}	-1,575096239	16	-4,960702248	16	-15,45666667	13
{'algorithm': 'kd_tree', 'leaf_size': 40, 'n_neighbors': 3, 'p': 2, 'weights': 'uniform'}	-1,57510233	18	-4,960705903	18	-15,45666667	13

Importanza delle Feature:

1. weight_to_power_ratio: 0.25851917347830666
2. max_speed_km_per_h: 0.16775473286833928
3. length_power_ratio: 0.16717654537545876
4. engine_hp: 0.07407300790561075
5. max_speed_power_ratio: 0.06380548937684422
6. maximum_torque_n_m: 0.04096502945786751
7. specific_power: 0.0396811140850274
8. number_of_gears: 0.016192941774014244
9. rear_track_mm: 0.014999410901121056
10. number_of_cylinders: 0.01238682264661136

prolog_generated.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{algorithm: 'kd_tree'; leaf_size: 40; n_neighbors: 3; p: 2; weights: 'uniform'}	-1,254687681	1	-3,587724387	1	-12,46	1
{algorithm: 'kd_tree'; leaf_size: 20; n_neighbors: 3; p: 2; weights: 'uniform'}	-1,254783615	2	-3,5887855	2	-12,46	1
{algorithm: 'kd_tree'; leaf_size: 30; n_neighbors: 3; p: 2; weights: 'uniform'}	-1,254783615	2	-3,5887855	2	-12,46	1
{algorithm: 'kd_tree'; leaf_size: 40; n_neighbors: 3; p: 2; weights: 'distance'}	-1,260862924	7	-3,600201884	4	-12,4629665	4
{algorithm: 'kd_tree'; leaf_size: 20; n_neighbors: 3; p: 2; weights: 'distance'}	-1,261016787	8	-3,601549949	5	-12,4629665	4
{algorithm: 'kd_tree'; leaf_size: 30; n_neighbors: 3; p: 2; weights: 'distance'}	-1,261016787	8	-3,601549949	5	-12,4629665	4
{algorithm: 'kd_tree'; leaf_size: 40; n_neighbors: 5; p: 2; weights: 'distance'}	-1,259902585	6	-3,663976261	7	-12,542415	7
{algorithm: 'kd_tree'; leaf_size: 20; n_neighbors: 5; p: 2; weights: 'distance'}	-1,259785178	4	-3,664655917	8	-12,542415	7
{algorithm: 'kd_tree'; leaf_size: 30; n_neighbors: 5; p: 2; weights: 'distance'}	-1,259785178	4	-3,664655917	8	-12,542415	7
{algorithm: 'kd_tree'; leaf_size: 40; n_neighbors: 5; p: 2; weights: 'uniform'}	-1,262424395	12	-3,704448602	10	-12,568	10
{algorithm: 'kd_tree'; leaf_size: 20; n_neighbors: 5; p: 2; weights: 'uniform'}	-1,262239836	10	-3,70507413	11	-12,568	10
{algorithm: 'kd_tree'; leaf_size: 30; n_neighbors: 5; p: 2; weights: 'uniform'}	-1,262239836	10	-3,70507413	11	-12,568	10
{algorithm: 'kd_tree'; leaf_size: 40; n_neighbors: 7; p: 2; weights: 'distance'}	-1,264188511	15	-3,835978414	13	-12,80292201	13
{algorithm: 'kd_tree'; leaf_size: 20; n_neighbors: 7; p: 2; weights: 'distance'}	-1,26412282	13	-3,836015266	14	-12,80292201	13
{algorithm: 'kd_tree'; leaf_size: 30; n_neighbors: 7; p: 2; weights: 'distance'}	-1,26412282	13	-3,836015266	14	-12,80292201	13
{algorithm: 'kd_tree'; leaf_size: 40; n_neighbors: 7; p: 2; weights: 'uniform'}	-1,271246884	18	-3,907258591	16	-12,84571429	16
{algorithm: 'kd_tree'; leaf_size: 20; n_neighbors: 7; p: 2; weights: 'uniform'}	-1,27119598	16	-3,907315741	17	-12,84571429	16
{algorithm: 'kd_tree'; leaf_size: 30; n_neighbors: 7; p: 2; weights: 'uniform'}	-1,27119598	16	-3,907315741	17	-12,84571429	16

Importanza delle feature:

1. weight_to_power_ratio: 0.46880846942532944
2. specific_power: 0.0719592365587329
3. length_ratio: 0.017537212455045262
4. number_cylinder_capacity_ratio: 0.015173008560909055
5. avg_trunk_capacity_brand: 0.0053020630873005575
6. shape: 0.0014408536139783607
7. most_powerful_car_brand: 0.0003252803480765833
8. drag_coefficient: 0.00021651319990577023
9. power: 0.0002067472976678387
10. most_common_cylinder_layout_yeargroup_of_three: 0.00015903730273129931

AdaBoost

Il secondo regressore analizzato attraverso i quattro dataset precedentemente ottenuti è l'**AdaBoost**.

Con l'uso della 10-fold Cross-validation si sono misurate le prestazioni del regressore al variare di tutte le combinazioni possibili dei suoi iperparametri con i seguenti valori testati:

- **n_estimators:** i valori testati per questo iperparametro sono 10, 20 e 30, questo iperparametro rappresenta il numero di stimatori deboli (weak learner) che verranno utilizzati per costruire il modello di boosting

- **learning_rate**: i valori testati per questo iperparametro sono 0.1, 0.01 e 0.001, con questo iperparametro controlla la quantità con cui viene aggiornato il peso di ogni stimatore debole durante l'addestramento del modello.

Qui di seguito si riportano per ogni dataset le migliori cinque combinazioni di iperparametri, poiché riportarle tutte e diciotto risulterebbe poco leggibile.

Valutazione dataset

preprocessed.csv

params	mean_test_neg_ g_mean_absolute_error	rank_test_neg_ eg_mean_absolute_error	mean_test_neg_ mean_squared_error	rank_test_neg_ g_mean_squared_error	mean_test_pos_ ax_error	rank_test_pos_ x_error
{'learning_rate': 0.1; 'n_estimators': 30; 'random_state': 0}	-0,974411521	1	-1,829204881	1	-10,18337401	1
{'learning_rate': 0.1; 'n_estimators': 20; 'random_state': 0}	-0,993474636	2	-1,914128396	2	-10,32836986	2
{'learning_rate': 0.01; 'n_estimators': 30; 'random_state': 0}	-1,031791594	3	-2,066518906	3	-10,78274288	5
{'learning_rate': 0.1; 'n_estimators': 10; 'random_state': 0}	-1,032633493	4	-2,070877921	4	-10,76262809	3
{'learning_rate': 0.001; 'n_estimators': 10; 'random_state': 0}	-1,036651703	5	-2,085634356	5	-10,78953603	6
{'learning_rate': 0.01; 'n_estimators': 10; 'random_state': 0}	-1,037408135	6	-2,094651475	8	-10,91251937	9
{'learning_rate': 0.001; 'n_estimators': 30; 'random_state': 0}	-1,038025357	7	-2,093148253	6	-10,80331479	8
{'learning_rate': 0.001; 'n_estimators': 20; 'random_state': 0}	-1,039265925	8	-2,094854559	9	-10,79841506	7
{'learning_rate': 0.01; 'n_estimators': 20; 'random_state': 0}	-1,040284054	9	-2,094156803	7	-10,77585663	4

Importanza delle feature:

1. max_speed_km_per_h : 0.8744482465561646
2. engine_hp : 0.09349169039464952
3. year_to : 0.009093427132181319
4. front_suspension : 0.005687001199809287
5. engine_type : 0.005474325793626104
6. engine_hp_rpm : 0.004570557130535968
7. highway_fuel_per_100km_l : 0.003118253894453129
8. turnover_of_maximum_torque_rpm : 0.001559357038759476
9. width_mm : 0.001311522097050225
10. city_fuel_per_100km_l : 0.0011757142712788733

clustered.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'learning_rate': 0.1; 'n_estimators': 30; 'random_state': 0}	-0,974895271	1	-1,833749467	1	-10,264989	1
{'learning_rate': 0.1; 'n_estimators': 20; 'random_state': 0}	-0,996480868	2	-1,925412548	2	-10,35037913	2
{'learning_rate': 0.01; 'n_estimators': 30; 'random_state': 0}	-1,031791594	3	-2,066518906	3	-10,78274288	5
{'learning_rate': 0.1; 'n_estimators': 10; 'random_state': 0}	-1,032633493	4	-2,070877921	4	-10,76262809	3
{'learning_rate': 0.001; 'n_estimators': 10; 'random_state': 0}	-1,036651703	5	-2,085634356	5	-10,78953603	6
{'learning_rate': 0.001; 'n_estimators': 30; 'random_state': 0}	-1,038025357	7	-2,093148253	6	-10,80331479	8
{'learning_rate': 0.01; 'n_estimators': 20; 'random_state': 0}	-1,040284054	9	-2,094156803	7	-10,77585663	4
{'learning_rate': 0.01; 'n_estimators': 10; 'random_state': 0}	-1,037408135	6	-2,094651475	8	-10,91251937	9
{'learning_rate': 0.001; 'n_estimators': 20; 'random_state': 0}	-1,039265925	8	-2,094854559	9	-10,79841506	7

Importanza delle feature:

1. max_speed_km_per_h : 0.8774168082621858
2. engine_hp : 0.09375896419498145
3. year_to : 0.008361046684023828
4. engine_hp_rpm : 0.004561586149857855
5. engine_type : 0.0033764526604931275
6. highway_fuel_per_100km_l : 0.003112133459977012
7. cluster : 0.0031007669573710804
8. front_suspension : 0.0020302013382527025
9. turnover_of_maximum_torque_rpm : 0.001799685812832596
10. width_mm : 0.0013089478727150979

clustered_plus_prolog_generated.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'learning_rate': 0.1; 'n_estimators': 30; 'random_state': 0}	-0,838975499	1	-1,269836942	1	-8,309312815	1
{'learning_rate': 0.1; 'n_estimators': 20; 'random_state': 0}	-0,876192829	2	-1,383094078	2	-8,858427036	2
{'learning_rate': 0.1; 'n_estimators': 10; 'random_state': 0}	-0,953923089	3	-1,685189104	3	-9,355250806	3
{'learning_rate': 0.01; 'n_estimators': 30; 'random_state': 0}	-0,987230801	4	-1,820429884	4	-9,870559512	4
{'learning_rate': 0.01; 'n_estimators': 20; 'random_state': 0}	-0,996202094	5	-1,861797542	5	-10,13156082	6
{'learning_rate': 0.001; 'n_estimators': 20; 'random_state': 0}	-1,011733992	6	-1,906801125	7	-10,18661934	7
{'learning_rate': 0.01; 'n_estimators': 10; 'random_state': 0}	-1,012235323	7	-1,899187402	6	-10,12666858	5
{'learning_rate': 0.001; 'n_estimators': 10; 'random_state': 0}	-1,015305938	8	-1,908399283	8	-10,27103472	9
{'learning_rate': 0.001; 'n_estimators': 30; 'random_state': 0}	-1,015731721	9	-1,925671093	9	-10,20302672	8

Importanza delle feature:

1. weight_to_power_ratio : 0.6781008871957722
2. max_speed_km_per_h : 0.2968071226548724
3. highway_fuel_per_100km_l : 0.0073871800229591365

- length_power_ratio : 0.007346195210231009
- specific_power : 0.00577643821504934
- city_fuel_per_100km_l : 0.0031860404335125945
- number_of_gears : 0.0013961362676032914
- front_suspension : 0.0
- drive_wheels : 0.0
- transmission : 0.0

prolog_generated.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'learning_rate': 0.1; 'n_estimators': 30; 'random_state': 0}	-1,006232624	1	-1,78357893	1	-9,327000593	2
{'learning_rate': 0.1; 'n_estimators': 20; 'random_state': 0}	-1,01017841	2	-1,796627605	2	-9,192376857	1
{'learning_rate': 0.1; 'n_estimators': 10; 'random_state': 0}	-1,038787593	3	-1,910406068	3	-9,731402126	3
{'learning_rate': 0.01; 'n_estimators': 30; 'random_state': 0}	-1,066120629	4	-2,010476355	4	-9,785516867	5
{'learning_rate': 0.01; 'n_estimators': 20; 'random_state': 0}	-1,068217711	5	-2,019157318	5	-9,784549257	4
{'learning_rate': 0.001; 'n_estimators': 30; 'random_state': 0}	-1,071427993	7	-2,033145921	6	-9,951793967	6
{'learning_rate': 0.01; 'n_estimators': 10; 'random_state': 0}	-1,071408857	6	-2,036902229	7	-9,965322001	7
{'learning_rate': 0.001; 'n_estimators': 20; 'random_state': 0}	-1,072274214	8	-2,039216172	8	-9,982694743	8
{'learning_rate': 0.001; 'n_estimators': 10; 'random_state': 0}	-1,073976759	9	-2,04862404	9	-10,02892119	9

Importanza delle feature:

- weight_to_power_ratio : 0.9355954128638821
- length_power_ratio : 0.042844487620891415
- specific_power : 0.00741892362046784
- most_common_cylinder_layout_yeargroup_of_three : 0.006286019269592744
- length_ratio : 0.004174712115397652
- avg_trunk_capacity_brand : 0.0036804445097683705
- drag_coefficient : 0.0
- number_cylinder_capacity_ratio : 0.0
- max_speed_power_ratio : 0.0
- power : 0.0

Random forest

Il terzo regressore analizzato attraverso i quattro dataset precedentemente ottenuti è il **Random forest**.

Con l'uso della 10-fold Cross-validation si sono misurate le prestazioni del regressore al variare di tutte le combinazioni possibili dei suoi iperparametri con i seguenti valori testati:

- n_estimators:** i valori testati per questo iperparametro sono 10, 20 e 30, questo iperparametro rappresenta il numero di alberi decisionali (decision tree) che verranno creati durante l'addestramento del modello

- **max_depth:** i valori testati per questo iperparametro sono 4, 6 e 8, questo iperparametro rappresenta la massima profondità (depth) degli alberi decisionali (decision tree) che vengono utilizzati per creare il modello
- **min_samples_split:** : i valori testati per questo iperparametro sono 2, 4 e 6, questo iperparametro rappresenta il numero minimo di campioni (samples) richiesti per suddividere un nodo interno (non foglia) dell'albero decisionale in ulteriori sotto-alberi
- **min_samples_leaf:** i valori testati per questo iperparametro sono 1, 2 e 4, questo iperparametro rappresenta il numero minimo di campioni (samples) richiesti in una foglia dell'albero decisionale
- **max_features:** i valori testati per questo iperparametro sono sqrt e log2, questo rappresenta il numero massimo di feature che vengono utilizzate per creare ogni albero decisionale (decision tree) del modello. Sqrt seleziona un numero di feature pari alla radice quadrata del numero totale di feature, mentre log2 seleziona un numero di feature pari al logaritmo in base 2 del numero totale di feature.

Qui di seguito si riportano per ogni dataset le migliori cinque combinazioni di iperparametri, poiché riportarle tutte e diciotto risulterebbe poco leggibile.

Valutazione dataset

preprocessed.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,79557709	1	-1,310666021	1	-9,471651297	5
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,802928434	6	-1,32134675	2	-9,220226452	1
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,801024065	2	-1,325061692	3	-9,690489259	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,801024065	2	-1,325061692	3	-9,690489259	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,801024065	2	-1,325061692	3	-9,690489259	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,802253215	5	-1,327229113	6	-9,760175066	33
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,80397669	7	-1,332350391	7	-9,581252253	9
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,80397669	7	-1,332350391	7	-9,581252253	9
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,808263919	10	-1,346024674	9	-9,453563638	4
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,808333376	11	-1,347544509	10	-9,620906226	11
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,8069088	9	-1,353893627	11	-9,670740916	19
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,810950378	12	-1,363479303	12	-9,770933816	34
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,810950378	12	-1,363479303	12	-9,770933816	34
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,814116078	14	-1,375734519	14	-9,860321435	38
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,817851904	18	-1,383835018	15	-9,732036871	28
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,815336878	15	-1,383981447	16	-10,05351738	45
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,815336878	15	-1,383981447	16	-10,05351738	45
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,815336878	15	-1,383981447	16	-10,05351738	45
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,823300562	19	-1,39819148	19	-9,671670296	20
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,827733704	20	-1,415914999	20	-9,95809426	39
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,828016544	21	-1,428104199	21	-9,843980635	37
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,82975321	22	-1,429112223	22	-10,00291489	43
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,831804135	23	-1,434248196	23	-9,664208608	17
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,831804135	23	-1,434248196	23	-9,664208608	17
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 10, 'random_state': 0}	-0,835093119	28	-1,438741566	25	-9,660204045	16
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,834012791	25	-1,440877078	26	-9,641073519	13
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,834012791	25	-1,440877078	26	-9,641073519	13
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,834012791	25	-1,440877078	26	-9,641073519	13
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,835694525	30	-1,44236989	29	-9,712559501	27
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,837570353	31	-1,443395728	30	-9,787018783	36

Importanza delle feature:

- 1. max_speed_km_per_h : 0.31742819700748054
- 2. engine_hp : 0.22022606950313028
- 3. maximum_torque_n_m : 0.07938422066037451
- 4. number_of_gears : 0.04550556363265879
- 5. capacity_cm3 : 0.041299912628863664
- 6. rear_track_mm : 0.026119991468353206
- 7. number_of_cylinders : 0.02312388376945096
- 8. engine_hp_rpm : 0.022807691287891684
- 9. valves_per_cylinder : 0.02238880846326793
- 10. turnover_of_maximum_torque_rpm : 0.01835643188440403

clustered.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,807094202	3	-1,337108526	1	-9,721584483	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,807094202	3	-1,337108526	1	-9,721584483	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,807094202	3	-1,337108526	1	-9,721584483	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,804985951	1	-1,347008142	4	-9,575731233	13
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,812621456	11	-1,348635787	5	-9,646877894	15
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,812621456	11	-1,348635787	5	-9,646877894	15
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,812621456	11	-1,348635787	5	-9,646877894	15
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,807584576	6	-1,34884595	8	-9,468476026	6
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,812969557	14	-1,351809858	9	-9,54774069	10
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,806686342	2	-1,354155	10	-10,0086142	43
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,811078186	8	-1,356273791	11	-9,727527368	24
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,808422488	7	-1,360340663	12	-9,554265794	12
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,814255098	15	-1,377848252	13	-9,987636369	42
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,812551161	9	-1,383358867	14	-9,85335645	40
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,812551161	9	-1,383358867	14	-9,85335645	40
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,820037474	16	-1,401853147	16	-9,853222697	39
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,823839521	17	-1,420864102	17	-10,03234684	44
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,823839521	17	-1,420864102	17	-10,03234684	44
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 10, 'random_state': 0}	-0,838137153	25	-1,427101857	19	-10,16146715	49
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,828488952	19	-1,437024023	20	-9,305270785	3
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 10, 'random_state': 0}	-0,837394476	24	-1,438074437	21	-10,27913203	50
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,836795518	21	-1,446979347	22	-9,819000764	33
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,836795518	21	-1,446979347	22	-9,819000764	33
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,836795518	21	-1,446979347	22	-9,819000764	33
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,835488308	20	-1,449415924	25	-9,515332013	8
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,838603395	26	-1,462405936	26	-9,652225019	18
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 10, 'random_state': 0}	-0,848804025	37	-1,467626404	27	-10,03331578	46
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 10, 'random_state': 0}	-0,848804025	37	-1,467626404	27	-10,03331578	46
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 10, 'random_state': 0}	-0,848804025	37	-1,467626404	27	-10,03331578	46
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,842935884	29	-1,469839304	30	-9,268503139	1

Importanza delle feature:

- 1. engine_hp : 0.21476941412368059
- 2. max_speed_km_per_h : 0.21116428286323438
- 3. maximum_torque_n_m : 0.11693927945431674
- 4. capacity_cm3 : 0.07319886661035055
- 5. number_of_gears : 0.04500234676977929
- 6. number_of_cylinders : 0.031352393252131384
- 7. front_track_mm : 0.02604581362931944
- 8. engine_hp_rpm : 0.025825319772735986
- 9. rear_track_mm : 0.025690778369047203
- 10. turnover_of_maximum_torque_rpm : 0.024607738107654018

clustered_plus_prolog_generated.csv

params	mean_test_neg _mean_absolute _error	rank_test _neg_m ean_abs olute_err or	mean_test_neg _mean_squared _error	rank_test _neg_m ean_s quared_ error	mean_test_ma x_error	rank_test _max_err or
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,679273798	1	-0,947401508	1	-8,218893267	5
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,681427283	2	-0,953990067	2	-8,387206882	11
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,683847586	5	-0,958475428	3	-8,202579789	3
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,685917982	6	-0,964757573	4	-8,284489314	8
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,685917982	6	-0,964757573	4	-8,284489314	8
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,682960829	3	-0,964896354	6	-8,541248547	17
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,682960829	3	-0,964896354	6	-8,541248547	17
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,686996372	8	-0,96726299	8	-8,43370956	13
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,687126327	9	-0,96996629	9	-8,512569945	14
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,687126327	9	-0,96996629	9	-8,512569945	14
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,687126327	9	-0,96996629	9	-8,512569945	14
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,687556625	12	-0,978852806	12	-8,729152074	26
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,688635947	13	-0,9799879	13	-8,39183332	12
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,690577783	14	-0,983109707	14	-8,623790684	22
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,690577783	14	-0,983109707	14	-8,623790684	22
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,690577783	14	-0,983109707	14	-8,623790684	22
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,692983603	17	-0,987744484	17	-8,710081305	25
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,694281602	18	-0,996067892	18	-8,341882735	10
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 10, 'random_state': 0}	-0,698883406	19	-1,002896565	19	-8,588618186	19
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 10, 'random_state': 0}	-0,698883406	19	-1,002896565	19	-8,588618186	19
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 10, 'random_state': 0}	-0,698883406	19	-1,002896565	19	-8,588618186	19
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 10, 'random_state': 0}	-0,707509038	24	-1,005874363	22	-8,233712632	6
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 10, 'random_state': 0}	-0,706246767	23	-1,008424352	23	-8,28204112	7
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 10, 'random_state': 0}	-0,7011764	22	-1,017524313	24	-8,204503066	4
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 10, 'random_state': 0}	-0,707776042	25	-1,019537874	25	-8,067048834	1
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 10, 'random_state': 0}	-0,707776042	25	-1,019537874	25	-8,067048834	1
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 10, 'random_state': 0}	-0,711538244	27	-1,04264184	27	-8,857855635	47
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,719539582	28	-1,0598772	28	-8,766271917	32
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,719539582	28	-1,0598772	28	-8,766271917	32
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,719539582	28	-1,0598772	28	-8,766271917	32

Importanza delle feature:

- weight_to_power_ratio : 0.20694746531998626
- max_speed_km_per_h : 0.1908592232343832
- length_power_ratio : 0.10047876373850186
- engine_hp : 0.09503718197765383
- specific_power : 0.06089165888498991
- max_speed_power_ratio : 0.05783812826461269
- power : 0.05634429714029278
- maximum_torque_n_m : 0.04744581824428263
- number_of_gears : 0.02510722041390823
- capacity_cm3 : 0.019525120369678038

prolog_generated.csv

params	mean_test_ neg_mean_ absolute_err or	rank_test_ neg_mean_ absolute_ error	mean_test_neg mean_squared_ error	rank_test_ neg_mean_ squared_ error	mean_test_max _error	rank_test_ max_error
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,7844229	1	-1,188952204	1	-8,413910367	5
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,7844229	1	-1,188952204	1	-8,413910367	5
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,7883736	11	-1,200404494	3	-8,40987564	3
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,7883736	11	-1,200404494	3	-8,40987564	3
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,7883038	9	-1,209745571	5	-8,612465214	23
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,7883038	9	-1,209745571	5	-8,612465214	23
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,7878499	3	-1,213267255	7	-8,676593572	37
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,7878499	3	-1,213267255	7	-8,676593572	37
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,7878499	3	-1,213267255	7	-8,676593572	37
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,7878499	3	-1,213267255	7	-8,676593572	37
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,788257	7	-1,21555913	11	-8,808395007	51
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,788257	7	-1,21555913	11	-8,808395007	51
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,7927076	29	-1,216293873	13	-8,545540308	13
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,7909821	13	-1,217920294	19	-8,615729397	25
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,7909821	13	-1,217920294	19	-8,615729397	25
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,7909821	13	-1,217920294	19	-8,615729397	25
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,7909821	13	-1,217920294	19	-8,615729397	25
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 30, 'random_state': 0}	-0,7909821	13	-1,217920294	19	-8,615729397	25
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 6, 'n_estimators': 30, 'random_state': 0}	-0,7909821	13	-1,217920294	19	-8,615729397	25
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,7925715	21	-1,218798051	25	-8,585730517	21
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 30, 'random_state': 0}	-0,7925715	21	-1,218798051	25	-8,585730517	21
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,7919613	19	-1,222337384	27	-8,720742206	41
{'max_depth': 8, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 20, 'random_state': 0}	-0,7919613	19	-1,222337384	27	-8,720742206	41
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 20, 'random_state': 0}	-0,7925842	23	-1,223134688	29	-8,660093095	33
{'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 20, 'random_state': 0}	-0,7925842	23	-1,223134688	29	-8,660093095	33

Importanza delle feature:

1. weight_to_power_ratio : 0.3809143309977849
2. length_power_ratio : 0.25293531808823366
3. specific_power : 0.1141632045427596
4. max_speed_power_ratio : 0.09930883651295189
5. power : 0.07453327516994976
6. length_ratio : 0.02786713854405837
7. drag_coefficient : 0.016591058682757943
8. number_cylinder_capacity_ratio : 0.014481169269079781
9. avg_trunk_capacity_brand : 0.007280322673268886
10. most_powerful_car_brand : 0.005225411252857861

Alberi di decisione

Il quarto regressore analizzato attraverso i quattro dataset precedentemente ottenuti è il **Decision Tree**.

Con l’uso della 10-fold Cross-validation si sono misurate le prestazioni del regressore al variare di tutte le combinazioni possibili dei suoi iperparametri con i seguenti valori testati:

- **max_depth:** i valori testati per questo iperparametro sono None e range(3, 8), questo iperparametro rappresenta la massima profondità (depth) dell'albero decisionale che verrà creato durante il processo di apprendimento
- **min_samples_split:** : i valori testati per questo iperparametro sono 2, 5 e 10, questo iperparametro rappresenta il numero minimo di campioni (samples) richiesti per suddividere un nodo interno (non foglia) dell'albero decisionale in ulteriori sotto-alberi
- **min_samples_leaf:** i valori testati per questo iperparametro sono 1, 2 e 4, questo iperparametro rappresenta il numero minimo di campioni (samples) richiesti per una foglia dell'albero decisionale
- **max_features:** i valori testati per questo iperparametro sono 0.5, sqrt e log2, questo iperparametro rappresenta il numero massimo di feature che vengono considerate per la ricerca della miglior suddivisione dei dati in ogni nodo dell'albero decisionale, con 0.5 il modello considererà un sottoinsieme casuale di circa il 50% delle feature disponibili nel dataset per la ricerca della miglior suddivisione dei dati, sqrt seleziona un numero di feature pari alla radice quadrata del numero totale di feature, mentre log2 seleziona un numero di feature pari al logaritmo in base 2 del numero totale di feature.

Qui di seguito si riportano per ogni dataset le migliori cinque combinazioni di iperparametri, poiché riportarle tutte e diciotto risulterebbe poco leggibile.

Valutazione dataset

preprocessed.csv

params	mean_test_ neg_mean_ absolute_err or	rank_test_ t_neg_m ean_abs olute_err	mean_test_n eg_mean_sq uared_error	rank_test_ neg_mean_ _squared_ error	mean_test_ _max_erro r	rank_t est_ma x_error
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 10; 'random_state': 0}	-0,8740571	1	-1,58912948	1	-10,23793	7
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 10; 'random_state': 0}	-0,8775918	2	-1,59620157	2	-11,12405	37
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 2; 'random_state': 0}	-0,8849438	4	-1,61693662	3	-10,59121	24
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 10; 'random_state': 0}	-0,8923566	6	-1,62059797	4	-10,57894	23
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 10; 'random_state': 0}	-0,8801823	3	-1,64319788	5	-10,55254	22
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 2; 'random_state': 0}	-0,9005229	10	-1,64935592	6	-10,70506	25
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 5; 'random_state': 0}	-0,9005229	10	-1,64935592	6	-10,70506	25
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 10; 'random_state': 0}	-0,8934021	7	-1,65035998	8	-11,11243	36
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 2; 'random_state': 0}	-0,9086098	18	-1,65069527	9	-9,628555	1
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 5; 'random_state': 0}	-0,8936057	8	-1,66097786	10	-9,660945	2
{'max_depth': None; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 10; 'random_state': 0}	-0,8878537	5	-1,66568083	11	-10,15687	5
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 2; 'random_state': 0}	-0,9046497	13	-1,67099986	12	-10,22961	6
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 10; 'random_state': 0}	-0,9286492	23	-1,67806467	13	-10,38404	12
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 10; 'random_state': 0}	-0,9296234	24	-1,68841952	14	-10,38404	12
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 2; 'random_state': 0}	-0,9298819	25	-1,68873789	15	-10,03256	4
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 10; 'random_state': 0}	-0,8969341	9	-1,68971398	16	-11,22279	39
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 2; 'random_state': 0}	-0,9302142	26	-1,69136411	17	-10,38404	12
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 5; 'random_state': 0}	-0,9302142	26	-1,69136411	17	-10,38404	12
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 10; 'random_state': 0}	-0,9302275	28	-1,69139198	19	-10,38404	12
{'max_depth': None; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 2; 'random_state': 0}	-0,907153	15	-1,71058178	20	-10,3491	9
{'max_depth': None; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 5; 'random_state': 0}	-0,907153	15	-1,71058178	20	-10,3491	9
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 2; 'random_state': 0}	-0,9260386	22	-1,71494856	22	-9,752809	3
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 2; 'random_state': 0}	-0,9387771	29	-1,72459272	23	-10,47969	17
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 5; 'random_state': 0}	-0,9387771	29	-1,72459272	23	-10,47969	17
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 5; 'random_state': 0}	-0,9075344	17	-1,73041875	25	-11,56478	64
{'max_depth': 7; 'max_features': 0.5; 'min_samples_leaf': 2; 'min_samples_split': 5; 'random_state': 0}	-0,905948	14	-1,73385172	26	-11,34089	51
{'max_depth': 6; 'max_features': 0.5; 'min_samples_leaf': 1; 'min_samples_split': 5; 'random_state': 0}	-0,9397514	31	-1,73494757	27	-10,47969	17
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 2; 'random_state': 0}	-0,9192744	19	-1,75627381	28	-10,92589	31
{'max_depth': 8; 'max_features': 0.5; 'min_samples_leaf': 4; 'min_samples_split': 5; 'random_state': 0}	-0,9192744	19	-1,75627381	28	-10,92589	31

Importanza delle feature:

1. engine_hp : 0.6752494094661372
2. max_speed_kmh : 0.18134542318496272
3. maximum_torque_nm : 0.0611072672643531
4. curb_weight_kg : 0.024203466110284726
5. height_mm : 0.010355424117589688
6. fuel_tank_capacity_l : 0.008516387241955468
7. capacity_cm3 : 0.007322705516901904
8. wheelbase_mm : 0.0053377605067307056
9. transmission : 0.00433305569914654
10. cylinder_bore_mm : 0.0033350573446861392

clustered.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.891082357	1	-1.617797409	1	-9.408650794	5
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0.908174983	5	-1.679510342	2	-10.35579413	23
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0.904125777	4	-1.685235258	3	-9.198353258	3
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0.898723469	2	-1.690019991	4	-10.98649566	58
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0.903316883	3	-1.691557408	5	-10.91471843	53
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.911703753	7	-1.701150273	6	-8.817203061	1
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.911703753	7	-1.701150273	6	-8.817203061	1
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0.9169054	10	-1.723182087	8	-10.8875113	51
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.910015694	6	-1.733013738	9	-9.805748477	10
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0.941933566	21	-1.736777906	10	-9.567326526	7
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0.941895675	20	-1.745242048	11	-10.53137429	30
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.945359867	23	-1.752996028	12	-9.297040812	4
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.918812291	11	-1.76962359	13	-10.38865464	26
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0.950529384	25	-1.784763862	14	-10.70147598	35
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.914733026	9	-1.788425235	15	-10.77642857	47
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.954809298	28	-1.792450101	16	-10.06460791	21
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.954809298	28	-1.792450101	16	-10.06460791	21
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.927487137	14	-1.792747132	18	-10.70986403	39
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0.932954448	16	-1.793051642	19	-10.62582745	32
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.929562848	15	-1.80808999	20	-9.715229936	9
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.964569591	30	-1.808292204	21	-9.822872735	12
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0.965506599	31	-1.812860297	22	-9.69999416	8
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.921477093	12	-1.814420706	23	-10.92880952	55
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.921477093	12	-1.814420706	23	-10.92880952	55
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.966019965	33	-1.816247303	25	-9.563614128	6
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.939382268	17	-1.827922728	26	-10.7898372	48
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.939382268	17	-1.827922728	26	-10.7898372	48
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.94373742	22	-1.836141529	28	-10.52548722	29
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.945667428	24	-1.860523902	29	-9.96	13
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.954639201	27	-1.870003794	30	-10.77313346	46

Importanza delle feature:

1. maximum_torque_n_m : 0.4057056769240332
2. max_speed_km_per_h : 0.24060628112339227
3. engine_hp : 0.12770158430943374
4. engine_hp_rpm : 0.0856171485844455
5. height_mm : 0.03631426703760649
6. curb_weight_kg : 0.013833936334644813
7. fuel_tank_capacity_l : 0.011137942610822247
8. length_mm : 0.009756448066976233
9. wheelbase_mm : 0.008056304328011927
10. transmission : 0.0064906618836714025

clustered_plus_prolog_generated.csv

params	mean_test_neg _mean_absolut e_error	rank_test _neg_me an_absol ute_error	mean_test_neg _mean_squ are_d_error	rank_test _neg_m ean_squ ared_err or	mean_test_m ax_error	rank_test _max_er ror
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0,759868089	1	-1,165481168	1	-8,386889291	11
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0,769201279	2	-1,190736001	2	-8,436551081	12
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0,776148123	4	-1,197800266	3	-9,21170569	31
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0,776148123	4	-1,197800266	3	-9,21170569	31
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0,780110643	6	-1,232948734	5	-8,5855043	13
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0,785142829	11	-1,233012746	6	-9,526790422	38
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0,7910355	13	-1,2387146	7	-9,638977573	44
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0,7910355	13	-1,2387146	7	-9,638977573	44
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0,770536321	3	-1,246151333	9	-10,07948687	57
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0,783679713	8	-1,251150089	10	-8,782738663	17
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0,792211479	15	-1,262065132	11	-9,673887196	47
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0,793515102	16	-1,268370218	12	-9,673887196	47
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0,78381969	9	-1,275358418	13	-8,724395375	14
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0,783820716	10	-1,276300292	14	-9,040420699	26
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0,794869749	17	-1,285348053	15	-9,898792773	53
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0,788166149	12	-1,301808506	16	-9,988965838	55
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0,797264959	18	-1,327029427	17	-10,38563893	88
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0,78235368	7	-1,330129639	18	-9,211760087	33
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0,834680697	25	-1,360751183	19	-8,792461215	18
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0,834680697	25	-1,360751183	19	-8,792461215	18
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0,834707931	27	-1,361072324	21	-8,815008834	21
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0,834502908	22	-1,362501695	22	-9,057499677	27
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0,834502908	22	-1,362501695	22	-9,057499677	27
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0,836004248	29	-1,365163044	24	-8,792461215	18
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0,836031483	30	-1,365484184	25	-8,815008834	21
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0,835880853	28	-1,369691418	26	-8,815008834	21
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0,836364101	31	-1,369923122	27	-9,057499677	27
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0,824596532	19	-1,422873577	28	-8,7655	15
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0,824596532	19	-1,422873577	28	-8,7655	15
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0,833931988	21	-1,444277691	30	-8,962059524	25

Importanza delle feature:

1. max_speed_power_ratio : 0.37225895544233556
2. max_speed_km_per_h : 0.35326620508255163
3. length_power_ratio : 0.121337831145888
4. weight_to_power_ratio : 0.08352694657434492
5. engine_type : 0.013485041408930464
6. engine_hp : 0.009539852234960549
7. maximum_torque_n_m : 0.009375671043219572
8. number_of_gears : 0.005348460656675702
9. specific_power : 0.005070671101521841
10. transmission : 0.00425849478674309

prolog_generated.csv

params	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error	mean_test_neg_mean_squared_error	rank_test_neg_mean_squared_error	mean_test_max_error	rank_test_max_error
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0.880311992	1	-1.490665379	1	-7.89867505	1
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.893540009	13	-1.517677978	2	-9.011196812	26
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.888170912	3	-1.526225052	3	-8.740747191	20
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.889826609	8	-1.5320739	4	-8.987031798	24
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.889826609	8	-1.5320739	4	-8.987031798	24
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.888505086	6	-1.534758252	6	-9.158372762	29
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.886775895	2	-1.537051403	7	-8.889419779	23
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0.888356897	4	-1.538283217	8	-8.637771461	10
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0.890942623	12	-1.546760647	9	-9.792970704	68
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.888581054	7	-1.548516271	10	-8.746386085	21
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.89086007	10	-1.550227202	11	-9.197738358	31
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.89086007	10	-1.550227202	11	-9.197738358	31
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.897108033	16	-1.582109864	13	-8.280465991	3
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0.896400283	15	-1.582338163	14	-8.221463434	2
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0.896079635	14	-1.586263486	15	-8.63161704	9
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0.898055731	17	-1.614015301	16	-9.385896944	37
{'max_depth': 8, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0.888488614	5	-1.629369943	17	-10.53249964	89
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.923737504	19	-1.637645204	18	-9.182115079	30
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.924240982	20	-1.656158402	19	-8.396166667	4
{'max_depth': None, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.924240982	20	-1.656158402	19	-8.396166667	4
{'max_depth': 7, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0.901437746	18	-1.665928871	21	-10.31168678	82
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'random_state': 0}	-0.940286658	22	-1.676901879	22	-8.55561573	7
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 5, 'random_state': 0}	-0.940286658	22	-1.676901879	22	-8.55561573	7
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 10, 'random_state': 0}	-0.943273037	24	-1.692824983	24	-8.885856433	22
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 0}	-0.943285385	25	-1.695792403	25	-9.328107793	33
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 5, 'random_state': 0}	-0.943285385	25	-1.695792403	25	-9.328107793	33
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'random_state': 0}	-0.945505993	29	-1.700930119	27	-9.12863421	27
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 0}	-0.944302358	27	-1.702369054	28	-9.328107793	33
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 5, 'random_state': 0}	-0.944302358	27	-1.702369054	28	-9.328107793	33
{'max_depth': 6, 'max_features': 0.5, 'min_samples_leaf': 2, 'min_samples_split': 10, 'random_state': 0}	-0.946785948	32	-1.708993289	30	-9.12863421	27

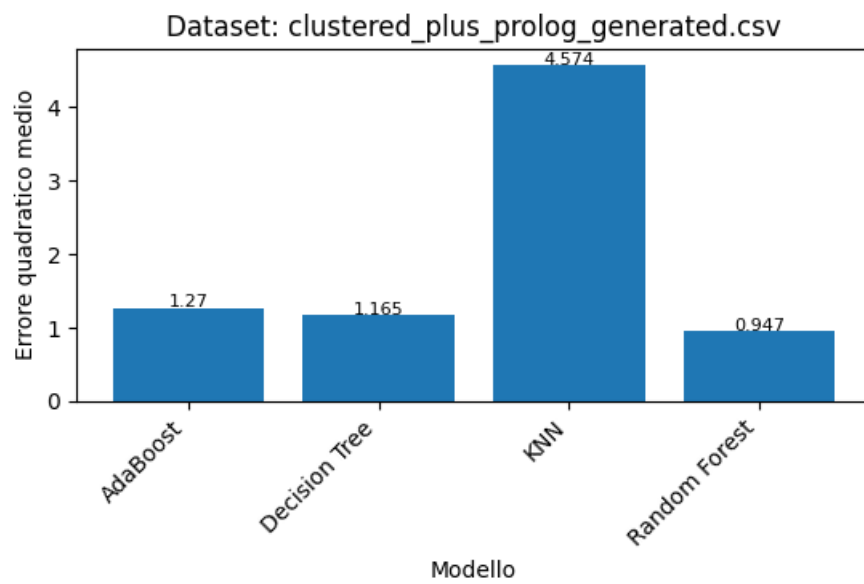
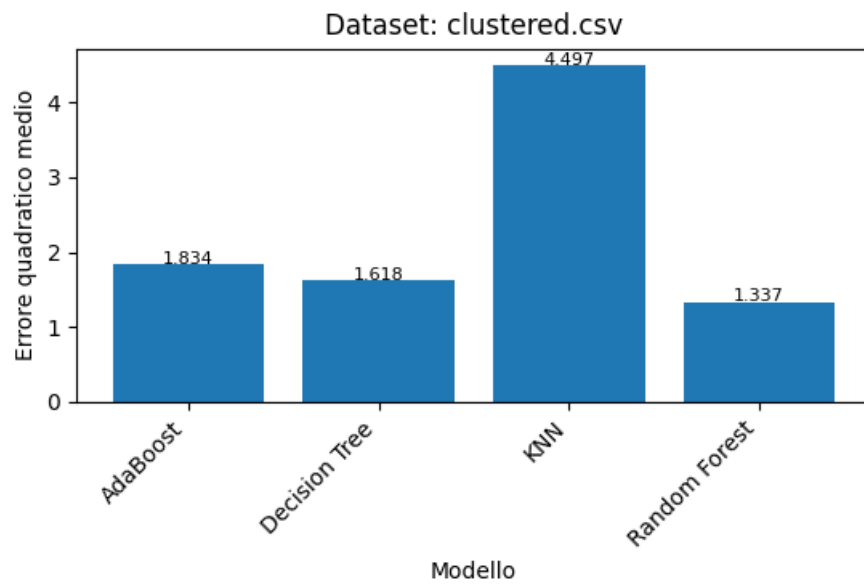
Importanza delle feature:

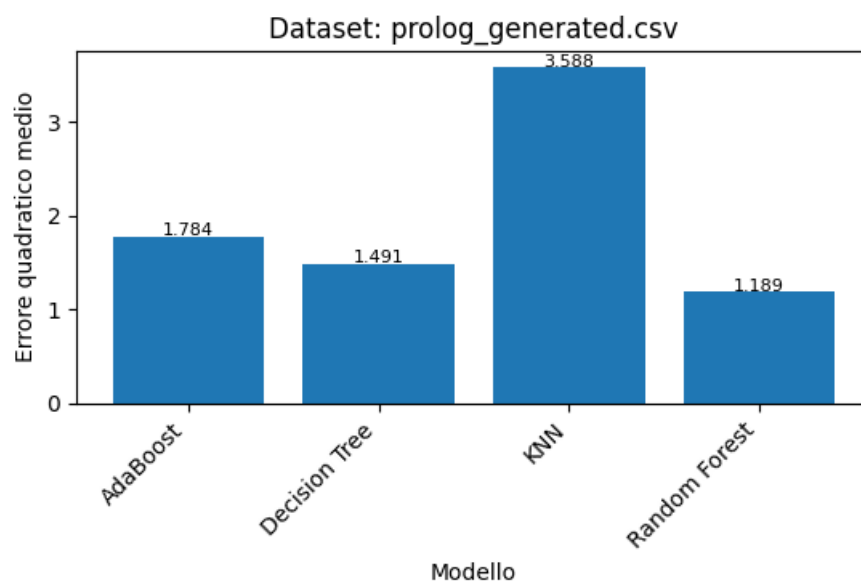
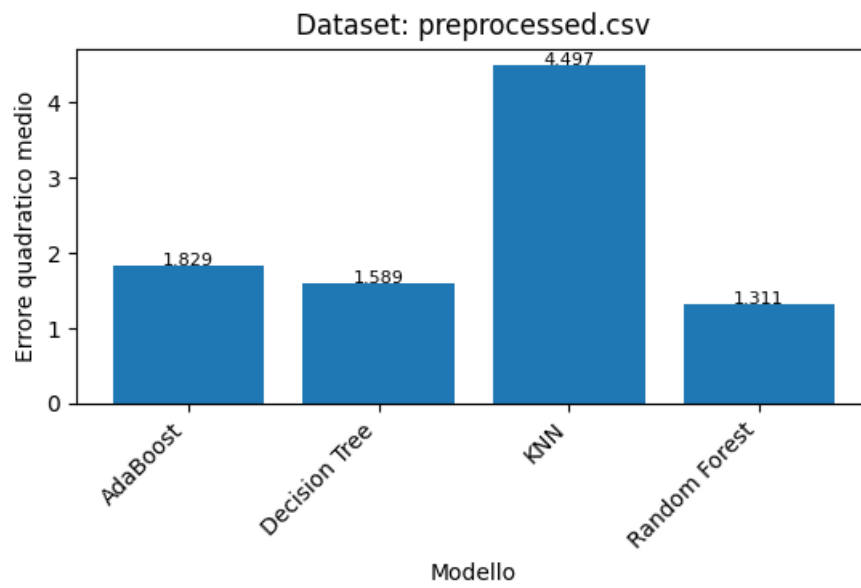
1. max_speed_power_ratio : 0.4309311541108177
2. weight_to_power_ratio : 0.32357746964689554
3. specific_power : 0.12583564816265733
4. length_power_ratio : 0.09208031123657731
5. drag_coefficient : 0.0071950489921374025
6. length_ratio : 0.006640492750786971
7. most_powerful_car_brand : 0.003532717730352478
8. avg_trunk_capacity_brand : 0.0028161241323684874
9. most_common_cylinder_layout_yeargroup_of_three : 0.0026305725777830877
10. shape : 0.0021684105359491427

Valutazione conclusiva

Dall’analisi dei dati raccolti è stato possibile mettere a confronto i vari migliori modelli (in termini di neg_mean_squared_error) individuando il miglior regressore in generale.

Qui di s e la specifica del dataset utilizzato.





Il miglior regressore è risultato essere "Random Forest" addestrato sul dataset "clustered_plus_prolog_generated".