Progetto ICON 21/22

Francesco Bottalico

2022

Introduzione

Obiettivo

Migliorare modelli supervisionati e non attraverso l'estrazione di conoscenza aggiuntiva dal web semantico.

In particolare:

- Interrogazione di un'ontologia (DBPedia) per estrarre feature utili per migliorare le qualità delle predizioni
- Allenamento di modelli di apprendimento supervisionato visti a lezione (con model selection per la scelta degli iperparametri)
- Combinare i precedenti modelli con dei modelli di clustering + background knowledge di DBPedia per migliorare ulteriormente le performance

Features

Le seguenti sono le feature originali del dataset:

- **date**
- **price**
- **bedrooms**
- **bathrooms**
- **sqft_living**
- **sqft_lot**
- **floors**
- **waterfront**
- **view**
- **condition**
- **sqft_above**
- **sqft_basement**
- **yr_built**
- **yr_renovated**
- **street**
- **city**
- **statezip**
- **country**

Feature estratte dal web semantico

- **lat**
- **long**
- **density**

Correlazione tra posizione e prezzo delle case

La posizione della casa effettivamente influisce sul prezzo, mentre la densità sembra influire meno, come si può vedere nella figura.

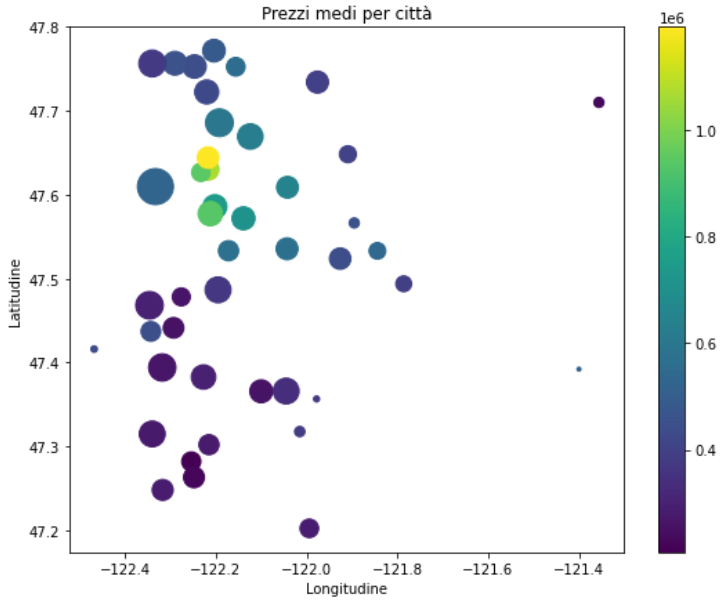
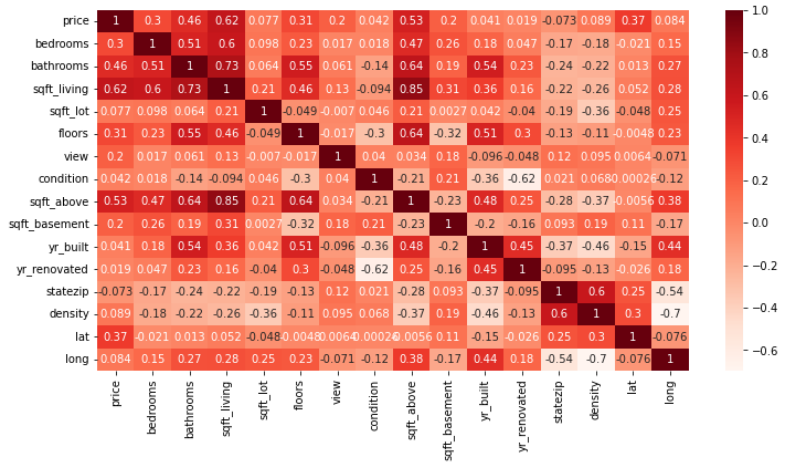


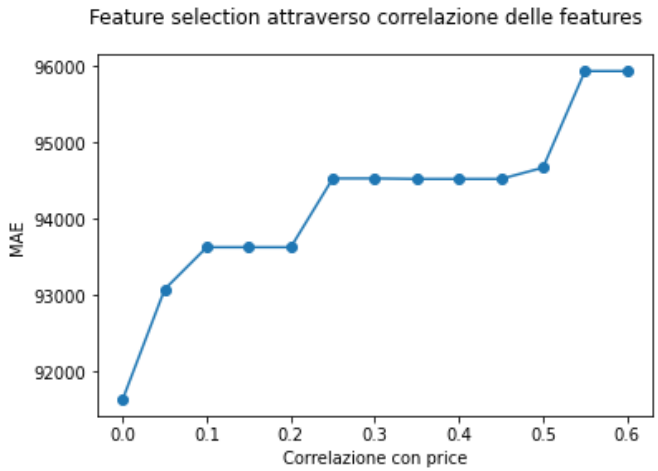
Figure: Case raggruppate per città, il colore rappresenta il prezzo medio delle case, la grandezza del punto mostra la densità di popolazione. Si nota un gruppo di città limitrofe per il quale il prezzo medio delle case è maggiore.

Preprocessing sul dataset

Preprocessing

1. Eliminazione degli elementi con prezzo nullo
2. Feature selection: vengono eliminate le feature date, street, country, waterfront
3. yr_renovated = yr_built se la casa non è mai stata ristrutturata
4. Rimozione degli outliers ($zscore < |2.5|$)
5. Standardizzazione
6. One hot encoding





Splitting del dataset

Il dataset viene diviso in training e testing set in due modi:

- Splitting degli elementi randomico
- Splitting in modo tale da avere elementi senza città in comune nel training e testing set. In questo modo è possibile simulare la situazione nella quale viene effettuata una predizione di un elemento in una città mai incontrata nel training. Utile per valutare l'utilita delle feature estratte lat e long.

Apprendimento supervisionato

Vengono allenati i seguenti modelli di apprendimento supervisionato, la model selection viene effettuata con una k-fold cross validation.

Rete neurale Per la rete neurale viene effettuata la model selection su diversi parametri quali: algoritmo di ottimizzazione, numero di neuroni, strati e parametro per la regolarizzazione L2.

Regressione lineare Gli iperparametri che vengono stimati dalla cross validation sono il learning rate, il suo metodo di aggiornamento ed il parametro di regolarizzazione α. Viene provata sia la regolarizzazione L1 che la L2.

Decision tree Per il decision tree la cross validation viene utilizzata per stimare parametri quali la massima profondità dell'albero ed il numero minimo di sample che una foglia deve contenere.

Random forest Per la random forest vengono allenati 100 decision trees, gli iperparametri stimati attraverso la cross validation sono la profondità massima degli alberi ed il numero massimo di feature prese in considerazione durante il training.

Support vector machine Nel caso della SVM vengono provate diverse funzioni kernel (per il kernel polinomiale si provano diversi gradi) e alcuni valori per il parametro per la regolarizzazione C (che è inversamente proporzionale alla forza della regolarizzazione).

Model selection rete neurale

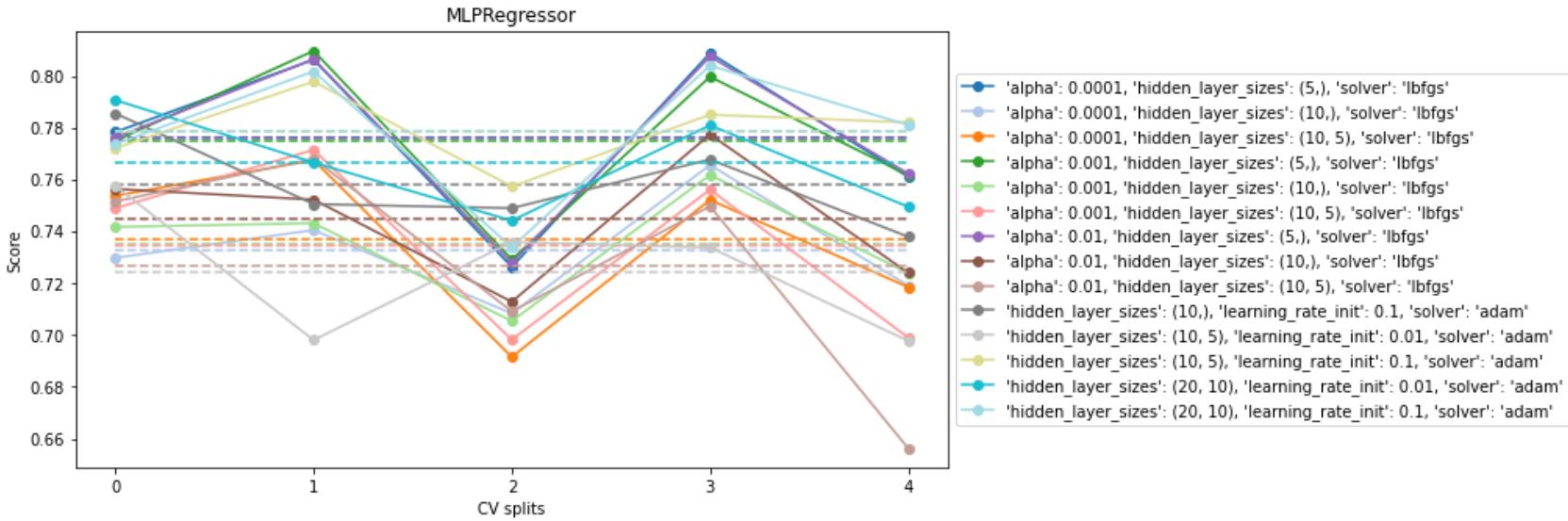


Figure: Andamento dello score sui vari fold della cross validation dati diversi valori per gli iperparametri. le rette tratteggiate indicano il valore medio.

Migliori parametri

solver = 'adam'
hidden_layer_sizes=(20,10)
learning_rate_init=0.1

Model selection regressione lineare L1 e L2

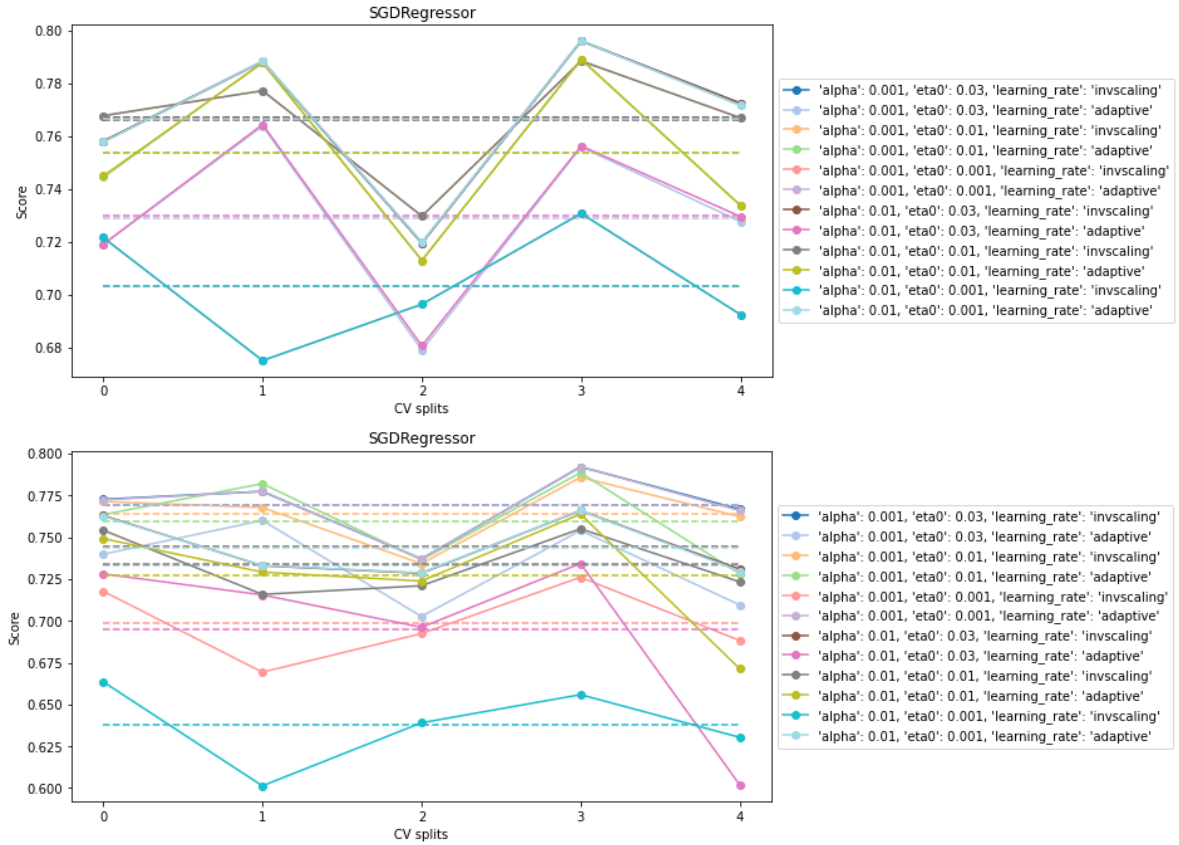


Figure: Score delle cross validation per la model selection dei modelli di regressione lineare con regolarizzazione L1 (sopra) ed L2 (sotto).

Migliori parametri L1

alpha = 0.001
eta0 = 0.03
learning_rate = 'invscaling'

Migliori parametri L2

alpha = 0.001
eta0 = 0.03
learning_rate = 'invscaling'

Model selection decision tree

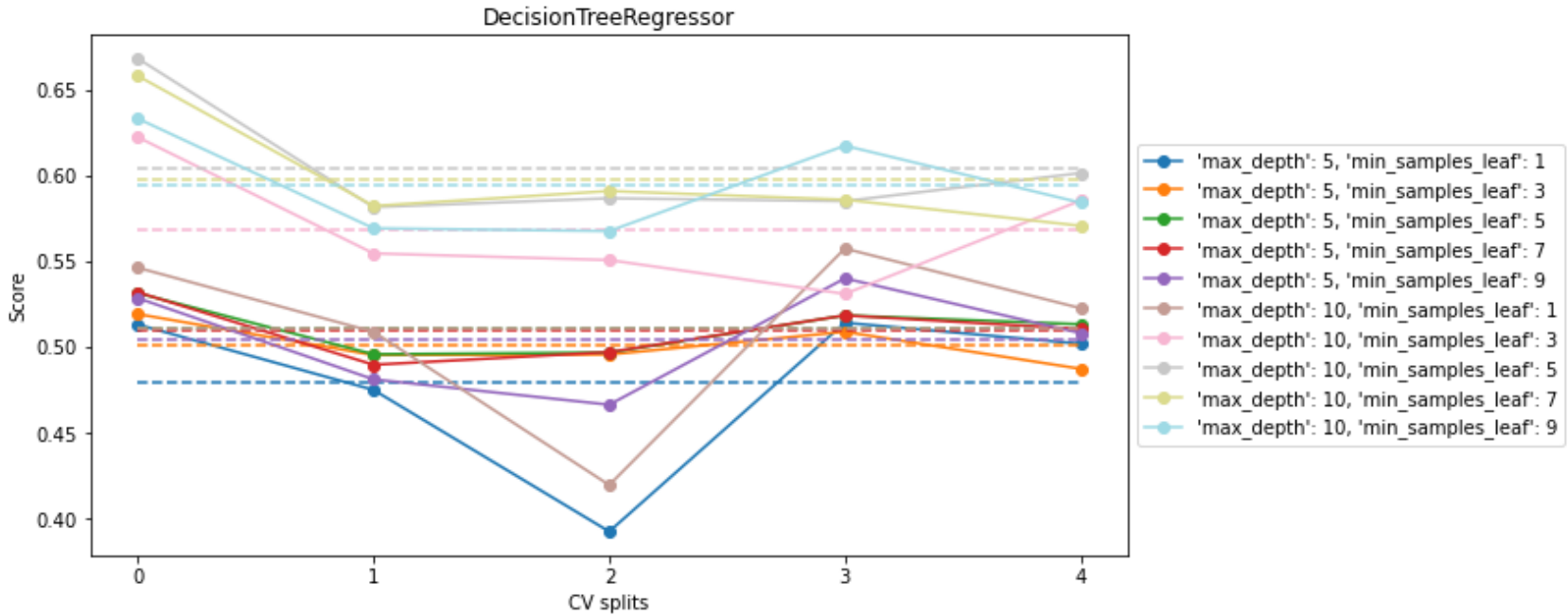


Figure: Score delle cross validation per la model selection del decision tree.

Migliori parametri

max_depth $= 10$

min_samples_leaf $= 5$

Model selection random forest

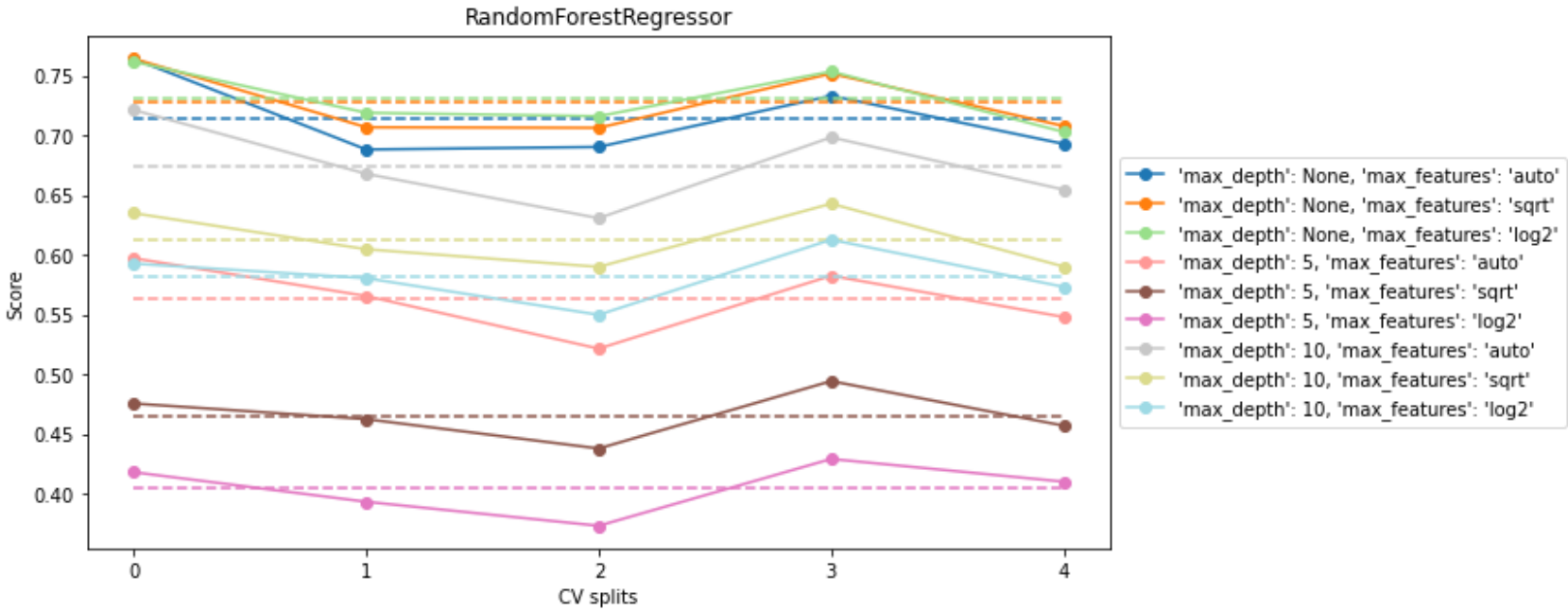


Figure: Score delle cross validation per la model selection del decision tree.

Migliori parametri

max_depth = None

max_features = 'log2'

Model selection SVM

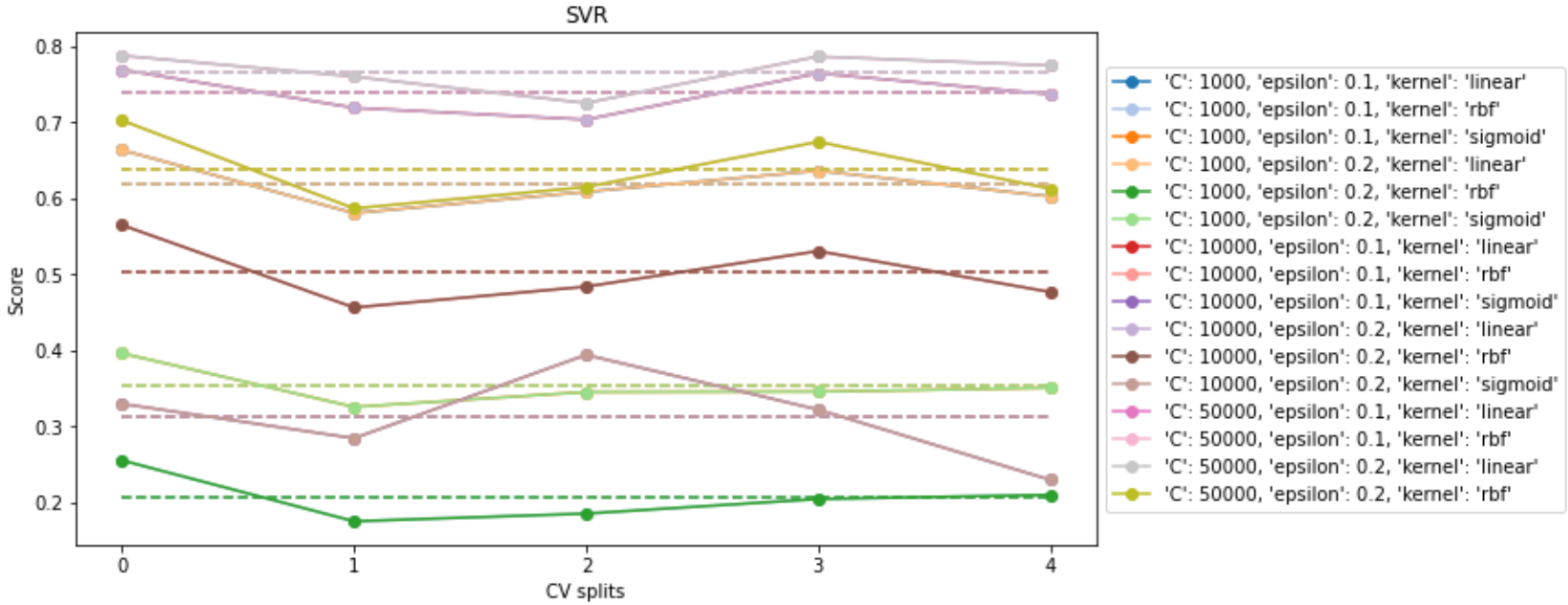


Figure: Score delle cross validation per la model selection della support vector machine.

Migliori parametri

C = 50000

kernel = 'linear'

Apprendimento non supervisionato: K-means

Le predizioni dei modelli possono essere migliorate con l'utilizzo del clustering:

- Dividere il training set in cluster
- Allenare indipendentemente un modello supervisionato su ogni cluster
- Per la predizione: assegnare il dato al cluster ed effettuare la predizione con il rispettivo modello

Apprendimento non supervisionato: K-means

Il clustering può essere effettuato in due modi:

- K-means su tutte le features del dato $\rightarrow$ dati non divisibili per $k > 2$
- K-means solo sulle features lat e long tenendo conto della relazione con il prezzo vista in precedenza $\rightarrow$ valore ottimale $k = 3$ con l'elbow method

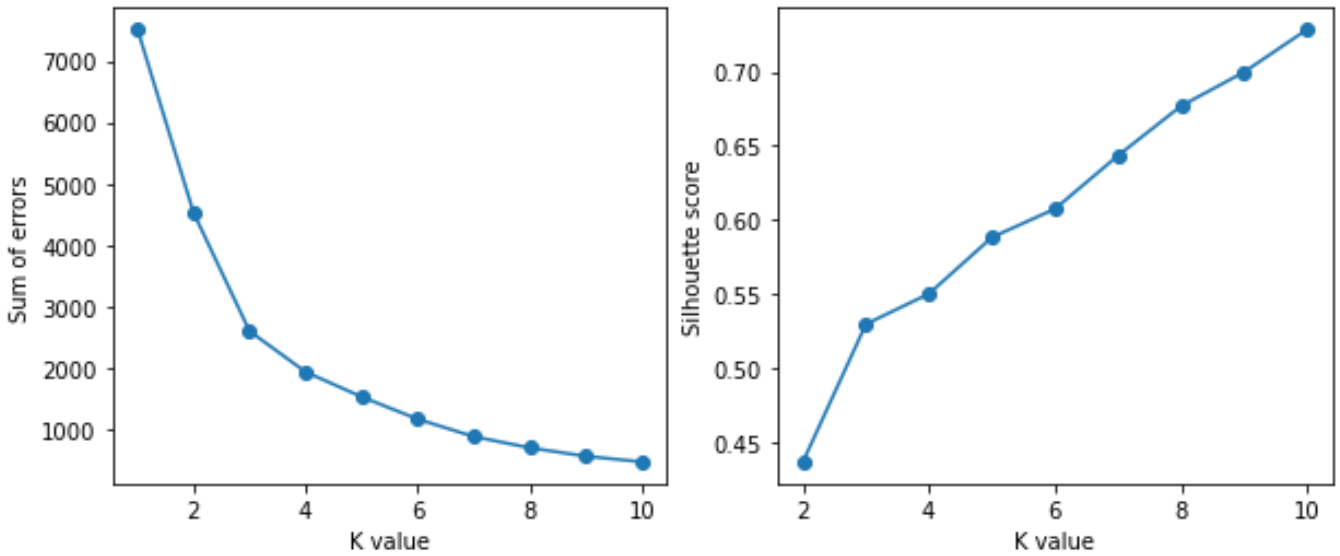


Figure: Andamento dell'errore all'aumentare di k nel secondo caso

Confronto modelli - equal splitting

	Original data			Extended data			
	R2	RMSE	MAE	R2	RMSE	MAE	decr. MAE
Dummy	-0.003	253592.36	191642.73	-0.003	253592.36	191642.73	**0%**
Rete neurale	0.835	102802.54	67716.71	0.843	100257.86	66375.14	**2.021%**
Lasso	0.82	107530.43	71892.61	0.818	107884.67	72514.84	**-0.858%**
Ridge	0.811	109967.49	73629.67	0.812	109665.55	72766.01	**1.187%**
Decision tree	0.688	141346.1	100310.09	0.711	136167.39	91516.67	**9.609%**
Random forest	0.772	120975.81	77201.7	0.785	117388.42	74131.08	**4.142%**
SVM	0.8	113336.14	70691.88	0.796	114286.55	71563.63	**-1.218%**

Table: Risultati predizioni usando la funzione di equal splitting.

	Cluster 1 (304/751)		**Cluster 2** (206/751)		**Cluster 3** (241/751)		
	RMSE	MAE	RMSE	MAE	RMSE	MAE	Tot MAE
Rete neurale	120621.8	84109.2	54167.9	36626.5	107014.3	70194.7	**66619.4**
Lasso	122742.4	85282.6	53754.3	37012.4	110568.8	72466.9	**67929.4**
Ridge	126733.2	86639.7	54087.4	36824.5	111961.8	73232.2	**68672.8**
Decision tree	175090.9	125769.1	71349.5	48866.8	139656.1	86026.7	**91921.2**
Random forest	133296.1	92910.0	57140.5	40125.2	130805.0	76968.6	**73315.4**
SVM	120161.8	82193.3	55079.3	36726.1	117846.7	75034.6	**67424.3**

Table: Risultati delle predizioni combinando apprendimento supervisionato e K-means. Lasso, SVM e Ridge hanno miglioramenti.

Confronto modelli - different cities splitting

	Original data			Extended data			
	R2	RMSE	MAE	R2	RMSE	MAE	decr. MAE
Dummy	0	241153.04	189087.16	0	241153.04	189087.16	**0%**
Rete neurale	0.117	226651.86	182441.63	0.392	188103.04	141451.75	**28.978%**
Lasso	0.477	174465.26	137111.43	0.6	152593.23	109693.36	**24.995%**
Ridge	0.487	172658.18	134693.9	0.6	152514.45	109608.88	**22.886%**
Decision tree	-0.211	265360.75	197765.79	0.542	163191.47	108231.19	**82.725%**
Random forest	0.368	191684.83	149244.86	0.63	146678.88	104214.31	**43.21%**
SVM	0.439	180650.8	136976.35	0.599	152661.49	106088.53	**29.115%**

Table: Risultati predizioni con nuove città durante il testing, il miglioramento sui dati "estesi" è notevole.

	Cluster 1 (95/671)		**Cluster 2** (354/671)		**Cluster 3** (222/671)		
	RMSE	MAE	RMSE	MAE	RMSE	MAE	Tot MAE
Rete neurale	329103.3	265676.8	138318.8	106167.8	64009.5	45005.8	**108515.6**
Lasso	246069.3	209740.8	147408.7	100309.3	65457.8	46995.7	**98163.8**
Ridge	241350.0	205236.3	147258.7	100224.0	65334.7	46834.0	**97427.6**
Decision tree	279222.8	195405.9	177526.5	125350.3	83347.6	57663.3	**112874.5**
Random forest	231919.2	176273.3	156945.9	107479.8	68201.1	44938.3	**96527.7**
SVM	162850.3	123706.7	141990.1	99907.8	64456.9	42933.3	**84427.3**

Table: Risultati delle predizioni combinando apprendimento supervisionato e K-means, Lasso ha un incremento notevole delle performance.