FEATURE ENGINEERING CON RECIPE

Lorenzo Bellotti 795192, Nicola Alimonda 872207, Giovanni Tosi 873263

ABSTRACT

L'obiettivo del progetto è quello di approfondire la tematica della feature engineering utilizzando il pacchetto tidymodels, in particolare il comando Recipe. Il lavoro è stato così strutturato: una prima fase di esplorazione dei dati seguita da una fase di pre-processing, alla quale è seguita una misurazione dell'impatto in termini di RMSE per ogni step contenuto nel recipe. Successivamente è stata applicata una Principal Component Analysis per andare a sondare il contributo delle singole features che misurano le diverse superfici delle abitazioni. La finalità del progetto è quella di applicare tecniche di feature engineering, per mezzo del comando Recipe, allo scopo di rendere i regressori adatti alla previsione della variabile Sale_Price mediante un modello di regressione lineare.

INDICE

INTRODUZIONE

Al fine di approfondire la tematica della feature engineering si è deciso di utilizzare il comando Recipe del pacchetto tidymodels. Attraverso questo comando è possibile specificare, oltre che le variabili oggetto di analisi, anche alcuni step di pre-processing da applicare al dataset.

Attraverso l'utilizzo di ggplot è possibile effettuare l'esplorazione delle variabili, grazie alla quale si intuiscono le migliori trasformazioni da applicare alle varie features in maniera tale da renderle adatte all'applicazione del modello.

Successivamente si andranno ad approfondire le fasi del lavoro, in particolare, la prima riguarda l'esplorazione del dataset, seguita da una fase di pre-processing finalizzata all'ottimizzazione del modello e alla selezione delle variabili di interesse. Infine, si è provveduto alla stima dell'impatto sul RMSE dei vari passaggi riguardanti la feature engineering per mezzo di k-folds cross validation.

In questo modo è stato possibile selezionare la miglior combinazione di step relativi al pre-processing da sottoporre al modello "lm" utilizzato. Nelle conclusioni, una volta selezionato il modello migliore, saranno illustrati possibili sviluppi futuri per il progetto.

DESCRIZIONE DEI DATI

Grazie all'utilizzo del pacchetto model data si è scaricato il dataset "ames". I dati in questione presentano 2930 osservazioni, ognuna delle quali rappresenta una proprietà nella città di Ames in Iowa e 74 features che si possono raggruppare per argomento di interesse:

- Caratteristiche delle proprietà: presenza di giardino, piscina, camere da letto, garage, porticati, prezzo di vendita, ecc..
- Location: questa categoria include Latitude, Longitude, Neighborood, ecc..
- Informazioni circa la proprietà: forma, grandezza, ecc..
- Rating sulle condizioni e qualità: Bldg_tytpe, condizioni del giardino, garage, ecc..

Nella fattispecie le variabili su cui si è concentrata l'analisi sono:

- Sale_Price: Prezzo di vendita della proprietà e variabile target, espresso in logaritmo al fine ridurre l'asimmetria della distribuzione, che diversamente sarebbe orientata a sinistra dato che si hanno valori compresi tra 12 789$ e 755 000$ con il valore medio pari a 180 796$ e la mediana di 160 000$.

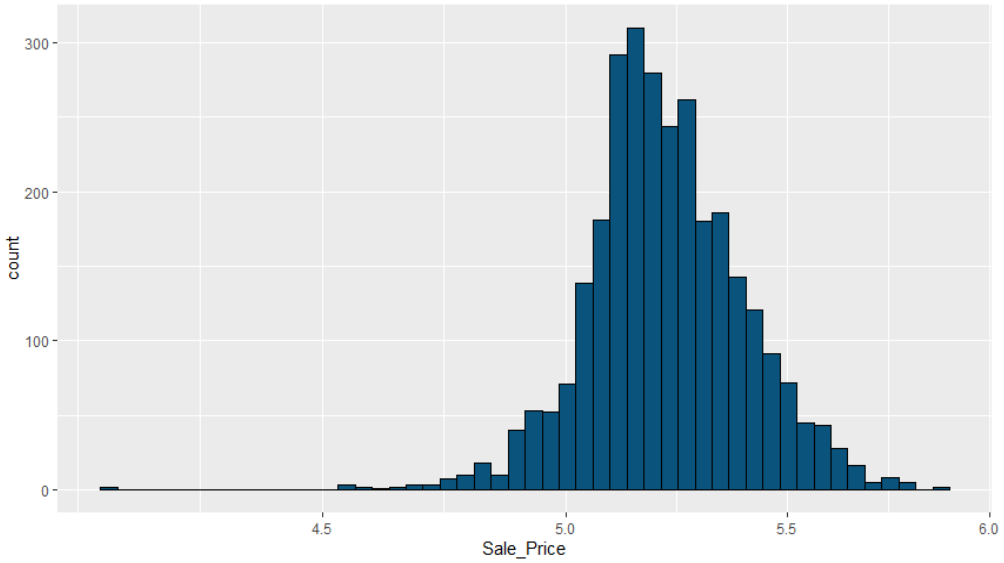

Figura 1: distribuzione di Sale_Price

- Neighborood: Quartiere di ubicazione della proprietà

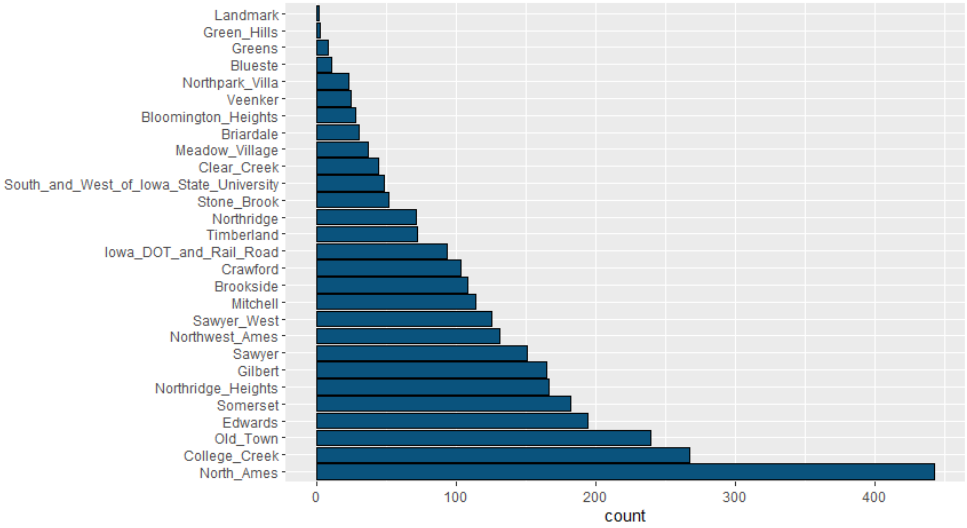

Figura 2: modalità di Neighborhood

- Gr_Liv_Area: Grandezza espressa in piedi al quadrato del lotto
- Year_Built: Anno di costruzione della struttura

- Bldg_Type: Tipologia dell'abitazione (TwoFmCon, Duplex , Twnhs, TwnhsE, OneFam)
- Latitude e Longitude: Coordinate spaziali di collocazione dell'abitazione
- Pool_QC: Presenza o meno di piscina (1/0)
- Fence: Presenza o meno di giardino (1/0)

SPLIT TRA TRAIN E TEST

Per quanto riguarda la separazione del data set iniziale in train set (80%) e test set (20%) si è scelto di applicare un campionamento stratificato sulla variabile Sale_Price. Il motivo di questa scelta risiede nella distribuzione asimmetrica di Sale_Price, come intuibile le abitazioni con un prezzo molto elevato sono inferiori rispetto a quelle con un prezzo medio, applicando il campionamento stratificato, le proprietà con un prezzo elevato non rischiano di essere escluse dal Train Set.

RECIPE

Per effettuare feature engineering si è deciso di sfruttare le potenzialità del comando Recipe contenuto nel pacchetto Tidymodels. Questo particolare comando permette di specificare la sequenza delle operazioni di pre-processing senza però applicarle immediatamente. Altri vantaggi derivanti dall'utilizzo di questo metodo sono:

- svolgere diversi task relativi al pre-processing in uno stesso oggetto applicabili a diversi data set e con diversi modelli predittivi.

- la presenza di comandi particolarmente utili per le operazioni di pre-processing.

- la sintassi compatta.

- tutte le operazioni possono essere raccolte in un unico oggetto contenuto in uno stesso script di R.

Una volta definito il Recipe, un modo per renderlo utilizzabile dal modello selezionato è quello di inserirlo all'interno di un workflow.

Nella fattispecie si è deciso di comporre il Recipe come segue.

```
ames_rec <-
  recipe(Sale_Price ~ Neighborhood + Gr_Liv_Area + Year_Built + Bldg_Type +
           Latitude + Longitude    , data = ames_train) %>%
  step_log(Gr_Liv_Area, base = 10) %>%
  step_other(Neighborhood, threshold = 0.01) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact( ~ Gr_Liv_Area:starts_with("Bldg_Type_") ) %>%
  step_ns(Latitude, Longitude, deg_free = 20)
```

Figura 3: composizione del Recipe 'ames_rec'

La variabile di risposta è Sale_Price mentre i predittori selezionati sono: Neighborhood, Gr_Liv_Area, Year_Built, Bldg_Type, Latitude e Longitude.

Il primo step del recipe è stato quello di trasformare su base logaritmica Gr_Liv_Area in quanto misura espressa in piedi quadri.

Nel secondo passaggio, grazie all'utilizzo del comando step_other, si è ridotta la numerosità delle modalità della variabile Neighborhood inserendo una soglia pari a 0.01. In questo modo la parte meno rilevante delle modalità (con una frequenza minore dell'1%) è stata accorpata all'interno di una nuova modalità chiamata "other". Grazie a questa operazione si è passati da 28 modalità a 8 modalità evitando di incorrere in un problema di dimensionalità durante il processo di creazione delle variabili dummy effettuato nello step successivo. Tale operazione è necessaria per poter utilizzare delle variabili categoriche come predittori in un modello di regressione.

Il quarto step riguarda l'utilizzo dell'interaction term, nel caso oggetto di studio si è deciso di applicare tale tecnica alle variabili Gr_Liv_Area e alle colonne estratte da Bldg_Type.

Tale termine di interazione si applica quando l'effetto di un regressore sulla variabile di risposta è contingente ad uno o più predittori.

In ultimo, si è applicato il comando step_ns alle variabili Latitude e Longitude, poiché entrambe queste variabili sono caratterizzate da una relazione non lineare con la variabile target Sale_Price. In questo modo si aggiunge una componente non lineare all'interno di una regressione lineare.

INTERACTION TERM

Per comprendere al meglio l'effetto dell'interaction term supponiamo di considerare solamente due predittori x_1 e x_2, l'interazione tra queste variabili può essere rappresentata come segue:

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_1x_2+error$$

Nell'equazione β_0 rappresenta l'intercetta, β_1 e β_2 rappresentano rispettivamente i coefficienti angolari relativi alle variabili x_1 e x_2, β_3 rappresenta il coefficiente angolare relativo all'interazione dei due predittori, infine, error rappresenta l'errore stocastico. Nel caso β_3 risulti non significativamente diverso da 0 la relazione tra x_1 e x_2 viene detta additiva e la componente di interazione non viene presa in considerazione, viceversa quando il coefficiente β_3 risulta significativo l'interazione di x_1 e x_2 aggiunge informazione per spiegare la varianza della variabile di risposta.

Il concetto fondamentale sul quale si basa l'interaction term è che in alcune situazioni l'andamento della variabile di risposta è condizionato dall'effetto combinato di due o più predittori. Come precedentemente detto, nel nostro caso, i regressori che costituiscono il termine di interazione sono Gr_Liv_Area e Bldg_Type, come intuibile l'effetto di Gr_Liv_Area su Sale Price è fortemente condizionato dal tipo di abitazione che si considera (Bldg_Type). Dalla figura che segue si può evincere quanto appena descritto, con la linea rossa che va ad evidenziare una tendenza lineare, con diversi coefficienti angolari, nell'andamento dei prezzi in relazione alla superficie per ciascun tipo di abitazione.

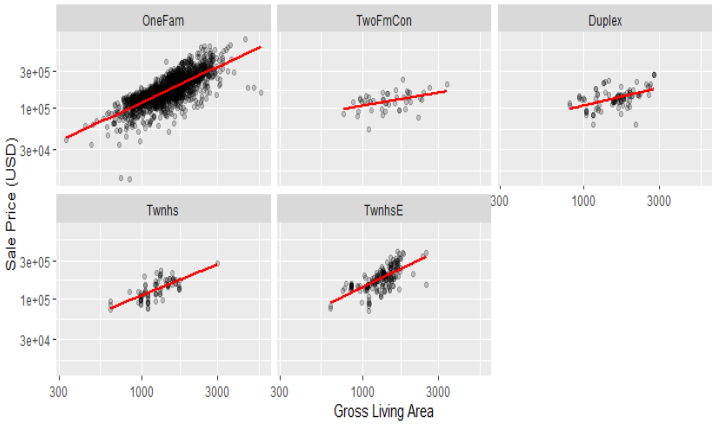

Figura 4: Gr_Liv_Area vs Bldg_type per ciascun tipo di abitazione

SPLINE

Nel caso in cui uno o più predittori abbiano una relazione non lineare con l'outcome è possibile utilizzare una funzione polinomiale o una funzione spline per approssimare l'andamento della variabile target. Tuttavia, l'utilizzo di una polinomiale potrebbe risultare problematico nei casi che presentano un numero elevato di punti critici e nei casi con elevato rumore e varianza. Questo è il motivo che spinge l'autore a selezionare la funzione spline per quanto concerne il caso in esame.

Attraverso l'utilizzo del comando step_ns si crea una colonna aggiuntiva che rappresenta la basis expansion della variabile su cui viene applicata una natural cubic splines. Nella fattispecie, il parametro sul quale si può attuare del tuning è rappresentato dal df (degrees of freedom). La relazione tra df e il numero di nodi nel caso di una natural cubic spline, che differisce dal caso dalla cubic spline poiché non considera i nodi agli estremi, è quella che segue:

$$\text{number of knots} = \text{degrees of freedom} - 1$$

come si vede di seguito l'andamento del valore di df influenza il trade off tra distorsione e varianza. Minori gradi di libertà comportano una maggior distorsione e una minor varianza, viceversa, aumentando i gradi di libertà la varianza aumenta a discapito della distorsione.

La scelta della posizione dei nodi viene presa in maniera automatica dalla funzione smoothing spline che seleziona la funzione g, tra tutte quelle che hanno derivata seconda continua, in grado di risolvere il problema seguente:

$$\text{minimize} \underbrace{\sum_{i=1}^{n}(y_i - g(x_i))^2}_{\text{RSS}} + \underbrace{\lambda \int g''(t)^2 dt}_{\text{Roughness penalty}}$$

Nell'equazione sovrastante ricopre una particolare importanza il parametro **λ** il cui cambiamento influisce sul trade off distorsione varianza:

- λ = 0 non impone restrizioni e la funzione g interpola i dati, questo caso è caratterizzato da una alta varianza e da una bassa distorsione.
- λ = ∞ rende la funzione g lineare, in questa situazione, la distorsione è alta e la varianza è bassa.

Si è deciso di applicare la spline sia a Latitude che a Longitude, di seguito viene mostrato l'andamento della funzione al mutare dei gradi di libertà nel caso in cui si utilizzi come variabile Latitude.

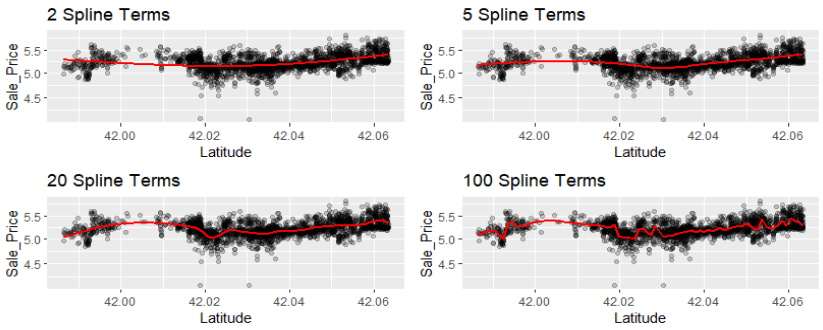

Figura 5: effetto 'Spline Terms' sull'andamento della funzione spline

Dall'analisi dei grafici si evince come il miglior compromesso tra distorsione e varianza si ottiene fissando uno spline term pari a 20.

RMSE

Al fine di valutare l'impatto delle singole operazioni di pre-processing in termini di miglioramento di affidabilità del modello si è scelto di creare 3 diversi recipe ciascuno dei quali presenta un'operazione aggiuntiva rispetto al precedente:

```
basic_rec <-
  recipe(Sale_Price ~ Neighborhood + Gr_Liv_Area + Year_Built + Bldg_Type +
           Latitude + Longitude , data = ames_train) %>%
  step_log(Gr_Liv_Area, base = 10) %>%
  step_other(Neighborhood, threshold = 0.01) %>%
  step_dummy(all_nominal_predictors())

interaction_rec <-
  basic_rec %>%
  step_interact( ~ Gr_Liv_Area:starts_with("Bldg_Type_") )

spline_rec <-
  interaction_rec %>%
  step_ns(Latitude, Longitude, deg_free = 20)

preproc <-
  list(basic = basic_rec,
       interact = interaction_rec,
       splines = spline_rec
  )

lm_models <- workflow_set(preproc, list(lm = lm_model), cross = FALSE)
lm_models
```

Figura 6: 'basic_rec', 'interaction_rec' e 'spline_rec'

All'interno del primo step del recipe (basic_rec) si svolgono le operazioni di pre-processing riguardanti la trasformazione logaritmica della feature Gr_Liv_Area, la riduzione delle modalità di neighborhood e il trattamento delle variabili nominali.

Nel secondo step, nominato interaction_rec, oltre alle operazioni precedentemente svolte viene aggiunto il comando step_interact grazie al quale è possibile inserire un termine di interazione riguardante Gr_Liv_Area e le features che descrivono la tipologia di abitazione.

Il terzo e ultimo step (spline_rec), aggiunge al recipe precedentemente creato un ulteriore passaggio grazie al quale è possibile implementare la funzione spline riferita alle variabili Latitude e Longitude.

La misura di performance scelta per valutare la bontà del modello è l'RMSE. Esso rappresenta la deviazione standard dei residui che a loro volta mostrano di quanto si discosta la previsione del modello dal valore reale.

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Per ognuno dei vari step elencati precedentemente è stata effettuata una K-folds cross validation per ottenere una misura dell'RMSE più accurata e consistente grazie a partizionamenti differenti del training test. Infatti, la KF-CV consiste nell'ulteriore divisione (ripetuta K volte) del training set in training e test dataset, ogni volta utilizzando training set differenti. La parte rimanente di dati sarà il validation dataset che ha il compito di simulare l'entrata di nuovi dati fin tanto che essi siano identicamente e indipendentemente distribuiti. La Cross-Validation inoltre è applicabile ad ogni algoritmo a differenza di altre procedure come il Cp di Mallow. L'errore sarà dato dalla seguente formula che rappresenta l'errore medio calcolato sui vari validation dataset:

$$\widehat{\mathrm{Err}} = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{\#V_k}\sum_{i\in V_k}(y_i - \hat{f}^{-V_k}(x_i))^2\right]$$

Ovviamente quest'approccio limita ulteriormente i dati che si usano per addestrare il modello, il che può non essere un problema quando abbiamo un campione con n sufficientemente grande.

I risultati ottenuti vengono riportati nel grafico sottostante, come prevedibile, il modello lm da risultati diversi per i differenti recipe utilizzati, in particolare, il modello lm applicato a basis_rec (basic_lm) è quello con un RMSE maggiore (0,0794). Lo stesso modello applicato all'interaction_rec (interact_lm) presenta un leggero miglioramento in termini di RMSE (0,0789), ciò non sorprende poiché come precedentemente scritto, in questo recipe viene aggiunto un ulteriore passaggio di pre-processing. Infine, il modello applicato con spline_rec (splines_lm) è quello con il miglior valore di RMSE (0,0775). Risulta quindi evidente che l'RMSE decresce grazie all'aggiunta di opportuni passaggi di pre-processing rendendo il modello maggiormente efficace.

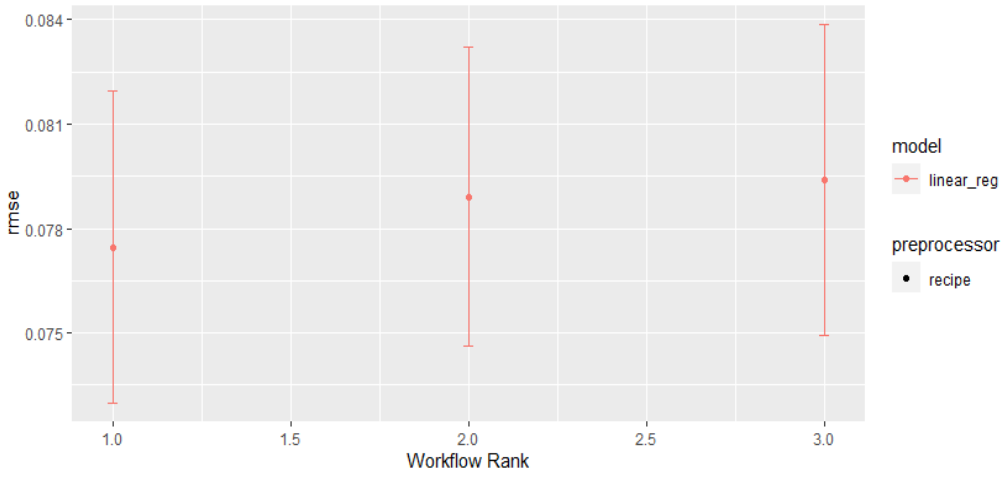

Figura 7: RMSE per i vari step del recipe

Nonostante un costante incremento delle performance del modello, sostanzialmente, esso non subisce grosse variazioni in termini di errore di previsione anche se si aggiungono delle nuove operazioni di pre-processing. Resta quindi da capire se valga la pena inserire step al recipe a fronte di un miglioramento nelle previsioni che risulta essere limitato.

MODELLO CON POOL E FENCE

Una volta determinato che il valore migliore di RMSE, per quanto riguarda il modello, è garantito utilizzando il recipe completo si è deciso di validare il modello selezionato sui dati del test set, il risultato ottenuto è un RMSE pari a 0.0754. Questi risultati sono in linea con i risultati della cross validation effettuata sul train set. Ciò conferma la robustezza del modello sviluppato. Infine, si è deciso di applicare il medesimo modello aggiungendo però due variabili nuove tra i predittori (Fence e Pool_QC), lo scopo di tale "esperimento" è quello di verificare quanto l'inserimento di queste variabili incida in termini di riduzione dell'errore nella previsione. Il risultato dell'RMSE del nuovo modello ottenuto sul test set è pari a 0.0754. L'interpretazione che ne consegue è che l'inserimento delle variabili Fence e Pool_QC non impatta in alcun modo sull'errore di previsione nei confronti di Sale_Price.

PCA

Una delle problematiche più frequenti nell'ambito dell'analisi di modelli che trattano dati ad alta dimensionalità è proprio la numerosità dei predittori presenti nel dataset. Infatti, tali predittori potrebbero essere correlati tra loro e quindi, se impiegati nel modello, l'informazione che contengono potrebbe essere ridondante e causare problemi di multicollinearità e overfitting.

Una delle tecniche più utilizzate di feature extraction per la riduzione della dimensionalità, evitando la perdita di informazioni, è la Principal Component Analysis (PCA). In particolare, la PCA è un metodo non supervisionato che mira a creare nuove features operando una combinazione lineare tra le features originali. Così facendo si riduce la dimensionalità dello spazio identificato dai predittori. Nel dettaglio Date p variabili X_1, X_2,…X_p (vettore casuale multivariato) l'analisi delle componenti principali consente di individuare k<p variabili Y_1, Y_2,…Y_k, aventi varianza massima, ognuna combinazione lineare delle p variabili di partenza.
Le Y_i sono delle variabili capaci di evidenziare e sintetizzare l'informazione insita nella matrice iniziale X così fatta:

$$X = (X_1, \quad X_2, \ldots, X_p) = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{pmatrix}$$

dove ogni colonna rappresenta le n osservazioni effettuate per una delle p variabili considerate per il fenomeno in analisi. Quindi, il generico elemento x_{ij} rappresenta la determinazione della j-esima variabile quantitativa osservata sull'i-esima unità statistica (i=1,…,n; j=1,…,p)
Da questa matrice se ne estrae un'altra $\tilde{X}$ chiamata matrice dei dati centrata, ossia:

$$\widetilde{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} - \begin{pmatrix} \mu_1 & \mu_2 & \dots & \mu_p \\ \mu_1 & \mu_2 & \dots & \mu_p \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1 & \mu_2 & \dots & \mu_p \end{pmatrix}$$

dove μ_i è il valore medio dell'i-esima variabile X_i.

La determinazione della prima componente Y_1 richiede l'individuazione del vettore p-dimensionale V_1 tale che:

$$Y_1 = V_1 \widetilde{X}$$

Per trovare V_1 bisogna risolvere un problema di massimo: lo scopo infatti è quello di massimizzare la varianza della prima componente principale in modo che quest'ultima spieghi la massima quantità possibile della variabilità totale. Tale problema si riconduce alla ricerca degli autovalori λ_i e degli autovettori V_i della matrice di covarianza C ricavata partendo da $\tilde{X}$ (una matrice n×p il cui generico elemento è C_{hk}=COV(X_h,X_k)).

$$C = \begin{pmatrix} COV(\widetilde{X_1}, \widetilde{X_1}) & COV(\widetilde{X_1}, \widetilde{X_2}) & \dots & COV(\widetilde{X_1}, \widetilde{X_p}) \\ COV(\widetilde{X_2}, \widetilde{X_1}) & COV(\widetilde{X_2}, \widetilde{X_2}) & \dots & COV(\widetilde{X_2}, \widetilde{X_p}) \\ \vdots & \vdots & \ddots & \vdots \\ COV(\widetilde{X_n}, \widetilde{X_1}) & COV(\widetilde{X_n}, \widetilde{X_2}) & \dots & COV(\widetilde{X_n}, \widetilde{X_p}) \end{pmatrix}$$

Gli autovalori di C si ricavano risolvendo la seguente equazione nell'incognita λ:

$$det(C - \lambda I) = 0$$

dalla quale si avranno al massimo p soluzioni (p autovalori λ_i). Dove I è la matrice identità p×p.
Ordinando le soluzioni in senso decrescente, si avrà:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$$

Dove λ_1 coincide con la varianza della prima componente principale Y_1, da cui si può ricavare il corrispettivo autovettore V_1 risolvendo l'equazione:

$$(C - \lambda_1 I)V_1 = 0$$

Analogamente si ottengono le altre componenti ed in particolare il numero di CP_k da estrarre, è dato dal numero di autovalori maggiori di 1, i quali corrispondo a quelle componenti con maggiore variabilità. Le componenti estratte sono ortogonali tra loro e quindi incorrelate.

Il punteggio (o score) della prima componente principale per l'i-esima unità statistica è:

$$y_{i1} = v_{11}\widetilde{x_{i1}} + v_{12}\widetilde{x_{i2}} + \cdots + v_{1p}\widetilde{x_{ip}}$$

dove v_{1j} è il coefficiente della prima componente Y_1 e della j-esima variabile X_j.
Esso fornisce il peso assegnato alla j-esima variabile nella definizione della prima componente.

In generale, considerando le prime k CP, la matrice dei punteggi sarà:

$$Y = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ v_{p1} & v_{p2} & \cdots & v_{pp} \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

Ovvero Y=V·X.

All'interno dei dati Ames ci sono diversi predittori, 7 per la precisione, che esprimono misure relative alla grandezza delle abitazioni come ad esempio: Gr_Liv_Area, Total_Bsmt_SF, First_Flr_SF e Second_Flr_SF.

Quindi risulta interessante sfruttare la PCA in modo da ridurre la dimensionalità dei dati cercando di rappresentare le variabili ridondanti mediante un set ridotto di predittori.

Prima di procedere con la PCA si è scelto di cercare se vi fosse correlazione fra queste variabili ed in particolare dal correlogramma sottostante si nota come le variabili First_Flr_SF e Total_Bsmt_SF siano correlate positivamente, in quanto è probabile che le misure del seminterrato e del primo piano di una casa siano molto simili. Inoltre, si evince che la variabile Gr_Liv_Area è la variabile che presenta una correlazione con quasi tutte le altre variabili.

Gr_Liv_Area è la variabile che possiede media e varianza più elevate, questo aspetto è di fondamentale importanza nel momento in cui si effettua la PCA e verrà approfondito in seguito.

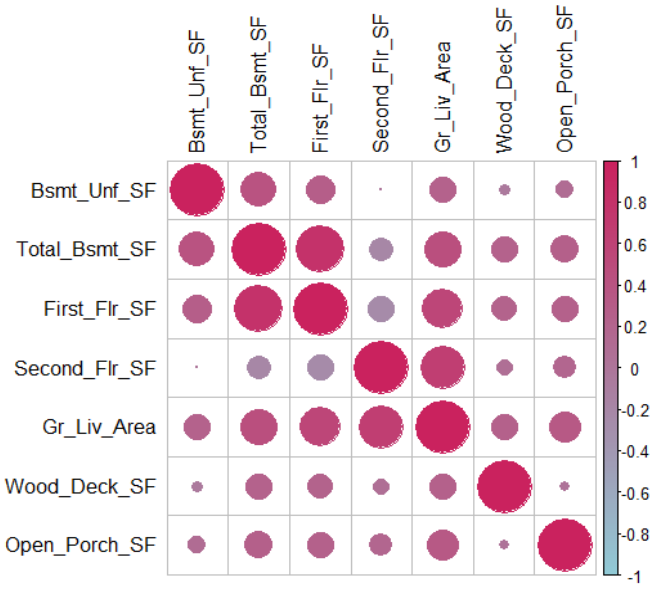

Figura 8: correlogramma con le variabili scelte per la PCA

Per poter effettuare la PCA sfruttando le potenzialità del pacchetto tidymodels è stato creato un recipe ad hoc, strutturato come segue:

I dati utilizzati nel recipe sono quelli relativi al training set da cui vengono estratte solo le 7 variabili d'interesse.

Successivamente, mediante il comando step_center, si "centra" la matrice X formata dai predittori selezionati. Questo passaggio è necessario prima di calcolare la PCA in quanto se non venisse eseguito le componenti estratte sarebbero fortemente influenzate dalla variabile che presenta la media più elevata, che come già evidenziato è Gr_Liv_Area.

Un altro step che solitamente si applica è quello relativo alla normalizzazione della matrice dei predittori che consiste nella centratura e nella standardizzazione della matrice X. Questa pratica è necessaria quando si effettua la PCA su variabili che hanno diverse unità di misura e ne influenza fortemente l'algebra in quanto se si lavora con la sola matrice centrata la PCA verrà calcolata partendo dalla matrice di varianze-covarianze, mentre se si opera una normalizzazione delle variabili la PCA viene calcolata partendo dalla matrice di correlazione tra i diversi predittori.

Nel caso in esame le variabili sono tutte espresse in piedi quadri quindi, come anticipato, si è scelto di operare solo la centratura della matrice X.

Per valutare la varianza spiegata da ognuna delle componenti estratte si è deciso di utilizzare un istogramma, dal quale si evince che selezionando le prime due componenti si riesce a catturare circa l'80% della varianza dei dati.

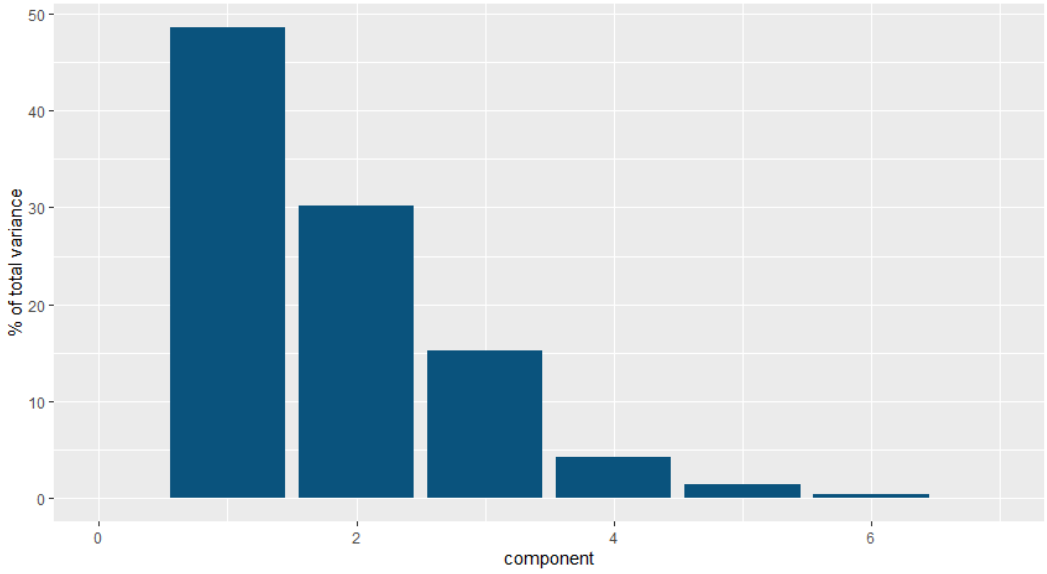

Figura 9: percentuale totale della varianza spiegata dalle varie componenti

Per poter esplorare quali siano le variabili che maggiormente influenzano le prime due componenti estratte si è deciso di ricorrere al seguente grafico:

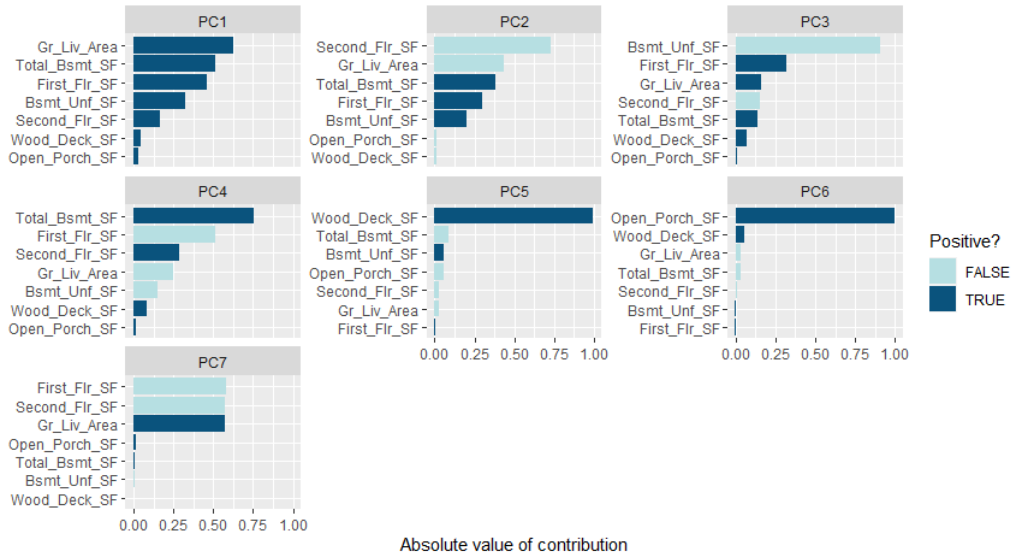

Figura 10: contributi delle singole variabili alle componenti

Si nota che la prima componente è caratterizzata dal contributo positivo di tutte le variabili in gioco ed inoltre le più influenti sono Gr_Liv_Area, Total_Bsmt_SF e First_Flr_Sf questo perché al crescere della dimensione del seminterrato e del primo piano è intuitivo che anche i piedi quadri totali della proprietà (Gr_Liv_Area) aumentino. Molto interessante è l'analisi della seconda componente in quanto essa è fortemente legata a Second_Flr_Sf che influenza Gr_Liv_Area. Questo potrebbe essere imputabile al fatto che non tutte le case hanno il secondo piano e che questa variabile incida particolarmente sul conto dei piedi quadri totali dell'abitazione.

Dalle analisi appena condotte risulta quindi evidente che Gr_Liv_Area sia la variabile che più caratterizza la varianza della dimensione delle abitazioni, per questo viene utilizzata nel modello come unica variabile legata a questo aspetto, ma potrebbe essere interessante sfruttare PC1 e PC2 come nuovi regressori nel modello lineare per la previsione del prezzo delle case.

CONCLUSIONI E SVILUPPI FUTURI

Il lavoro così strutturato è consistito nell'approfondimento del capitolo 8 (Feature Engeneering) del libro "Tidy Modeling with R", con particolare focus sulla scelta delle features e della loro trasformazione ai fini di renderle maggiormente adatte alla stima del prezzo delle case in funzione del modello selezionato. Al netto dei vari step di pre-processing e feature engineering non sono stati riscontrati miglioramenti significativi dell'RMSE, anche dopo l'aggiunta delle due variabili Fence e Pool_QC non si è ottenuto alcun miglioramento in termini di RMSE in quanto la porzione maggiore di varianza viene già spiegata da variabili come Gr_Liv_Area, Neighborood, Bldg_Type*Gr_Liv_Area, ecc..

La Principal Component Analysis inoltre suggerisce che all'interno della prima componente principale, che spiega la maggior porzione di varianza, Gr_Liv_Area è la feature che contribuisce maggiormente. Un ulteriore sviluppo del progetto potrebbe essere quello di utilizzare le componenti estratte dalla PCA per fittare un modello di regressione che tenga conto delle componenti ricavate dal processo.

Pertanto, il modello migliore selezionato, tenendo conto del criterio della parsimonia, è quello privo delle feature Fence e Pool_QC, la cui aggiunta non comporta differenze particolarmente significative a livello di performance.

FONTI

Kuhn, Johnson (2019). Feature Engineering and Selection. Chapman and Hall/CRC.

https://www.rdocumentation.org/packages/recipes/versions/0.1.17/topics/step_ns

https://stats.stackexchange.com/questions/517375/splines-relationship-of-knots-degree-and-degrees-of-freedom

tmwr.org

https://www.math.ntnu.no/emner/TMA4215/2008h/cubicsplines.pdf

https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-019-0666-3

https://recipes.tidymodels.org/reference/step_pca.html

https://juliasilge.com/blog/cocktail-recipes-umap/

https://www.statforbiology.com/pca/

http://www.jkarreth.net/files/RPOS517_Day11_Interact.html