Previsione della domanda di biciclette noleggiate

Addis Gaia, Marossi Clara, Tusetti Lucrezia

22 giugno 2024

Abstract

Il tema dei trasporti e della mobilità urbana è da sempre di grande interesse e oggetto di dibattito pubblico, specialmente nelle grandi città. In questo ambito, l'introduzione dei mezzi di noleggio a tempo ha sicuramente determinato una grande svolta. Questo lavoro si concentra sull'utilizzo di tecniche avanzate di machine learning (K-means, h-clust, PAM, KNN, SVM, NN, RF e XGB) per la previsione della domanda oraria di biciclette, con lo scopo di migliorare la qualità e l'efficienza del servizio offerto e di ridurre i tempi di attesa di coloro che ne usufruiscono. Il dataset scelto raccoglie dati relativi ai noleggi nella città di Seoul in un periodo di circa un anno e include una serie di variabili metereologiche.

1 Introduzione

La bicicletta è un mezzo di trasporto economico, salutare ed ecologico che contribuisce in modo significativo alla diminuzione dell'emissione di gas serra e alla decongestione del traffico stradale e incentiva le persone all'attività fisica. Sono queste le ragioni che negli ultimi decenni hanno promosso la diffusione dei servizi di bike sharing.

Il noleggio delle biciclette fu introdotto per la prima volta negli anni '60 ad Amsterdam, ma è negli ultimi anni che si è assistito ad una diffusione mondiale sempre più importante. Oggi, oltre 800 città nel mondo hanno programmi di bike sharing, integrati con altri sistemi di trasporto pubblico e influenzati dal design urbano. Molti studi dimostrano come la diffusione di questo fenomeno abbia apportato una serie di importanti benefici a livello ambientale, economico e sociale. Le città dove l'utilizzo dell'automobile è ampiamente diffuso tendono a concentrare servizi e attività commerciali in periferia, favorendo le grandi corporazioni. Al contrario, una struttura urbana che incentiva l'utilizzo della bicicletta, può stimolare le attività economiche locali e favorire le piccole imprese.

Nei paesi sviluppati, il settore dei trasporti consuma il 20-25% dell'energia totale e contribuisce significativamente (in percentuale superiore al 70%) alle emissioni di CO2. Nella città di Seoul, il traffico e l'inquinamento sono peggiorati sensibilmente dagli anni '90, inducendo il governo a promuovere il ciclismo e a implementare leggi e piani per migliorare l'infrastruttura ciclabile e ridurre le emissioni. Questo a causa

del fatto che un aumento dell'uso delle biciclette potrebbe far risparmiare al Paese oltre 450 milioni di dollari in costi energetici. Ad esempio, uno degli obiettivi prefissati è l'espansione della rete di piste ciclabili urbane fino a oltre 190 chilometri entro il 2026.

E' a causa dell'attualità e della rilevanza di questi temi che la scelta del dataset è ricaduta sul noleggio di biciclette a Seoul.

1.1 Obiettivi

L'obiettivo iniziale di questo lavoro è stato di comprendere la relazione tra la domanda oraria di biciclette noleggiate e alcune variabili atmosferiche e temporali (quali temperatura, umidità, mm di pioggia e altre che verranno descritte nel seguito), attraverso l'analisi esplorativa del dataset e ad alcuni algoritmi di **clustering**. L'obiettivo principale consiste, come accennato in precedenza, nella previsione della domanda oraria. Nella prima parte abbiamo implementato algoritmi di **regressione**, al fine di effettuare una previsione puntuale della domanda. Temendo che la previsione esatta del numero di biciclette noleggiate potesse risultare molto complessa e portare a risultati non ottimali, abbiamo approcciato il problema anche da una prospettiva più semplicistica: dopo aver diviso la domanda in fasce (Alta, Media e Bassa) abbiamo effettuato **classificazione** a 3 classi.

2 Dataset

Il totale delle osservazioni è pari a 8760 e non sono presenti missing values. Le variabili originarie fornite dal dataset sono descritte nella Tabella 1.

VARIABILE	**DESCRIZIONE**	**MODALITÀ/FORMATO**
Date	Data	Date YYYY-MM-DD
Rented Bike Count	Numero di noleggi effettuati nella giornata	Integer
Hour	Orario H in cui viene registrato il noleggio	Integer
Temperature	Temperatura registrata in Celsius	Continuous
Humidity	Umidità relativa percentuale	Continuous
Wind Speed	Velocità del vento in m/s	Continuous
Visibility	Misura della distanza a cui un oggetto è visibile distinto, in unità di 10m	Continuous
Dew point temperature	Temperatura a inizio giornata in Celsius	Continuous
Solar Radiation	Intensità della luce solare in MJ/mˆ2	Continuous
Rainfall	mm di pioggia caduti	Continuous
Snow	cm di neve caduti	Continuous
Season	Stagione dell'anno	Winter, Spring, Summer, Autumn
Holiday	Indica se il giorno considerato è giorno di vacanza	Holiday, No holiday
Functional Day	Indica se il servizio di noleggio è disponibile	Yes-No

Tabella 1: Descrizione delle variabili e formato dei dati

2.1 EDA e Pre processing

Lo studio si è concentrato soltanto sui giorni in cui il sistema di noleggio biciclette risultava funzionante, per questo motivo abbiamo eliminato dal dataset le istanze in cui la variabile Functioning Day assumeva modalità "No" (3% del totale delle osservazioni).
Data l'elevata correlazione, pari a 0.91, tra le variabili Temperature e Dew Point Temperature (punto di rugiada), si è deciso di eliminare la seconda, non apportando un guadagno di informazione rispetto alla prima (figura 1).

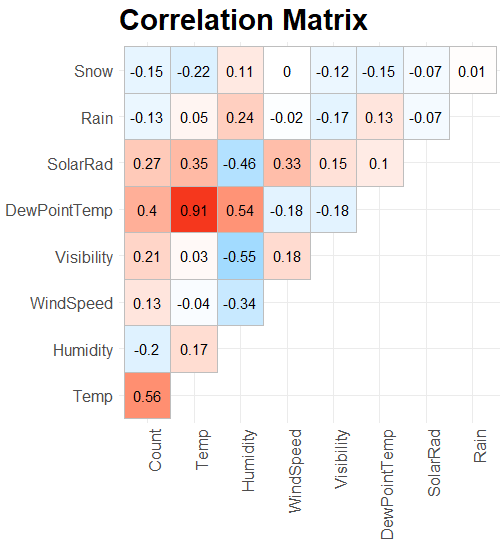

Figura 1: Matrice di Correlazione

Posto come obiettivo la previsione della domanda, ci siamo concentrate sulla variabile Count, con lo scopo di identificare quali fossero le variabili maggiormente influenti.

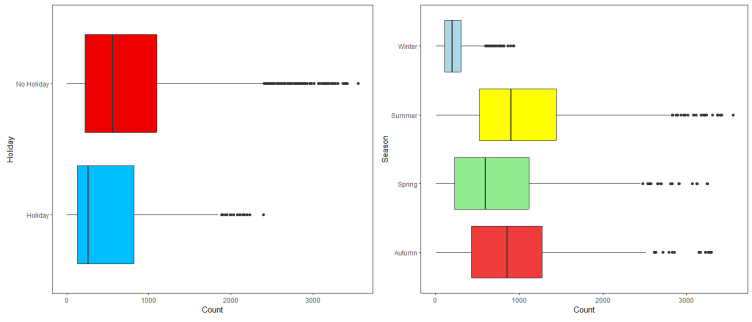

Figura 2: Boxplot Holiday e Season

Dai boxplot riportati in figura 2 si nota che la domanda di bicilette sembra essere leggermente minore quando la giornata è festiva mentre si nota un marcato calo di domanda nella stagione invernale ed uno più leggero in primavera.

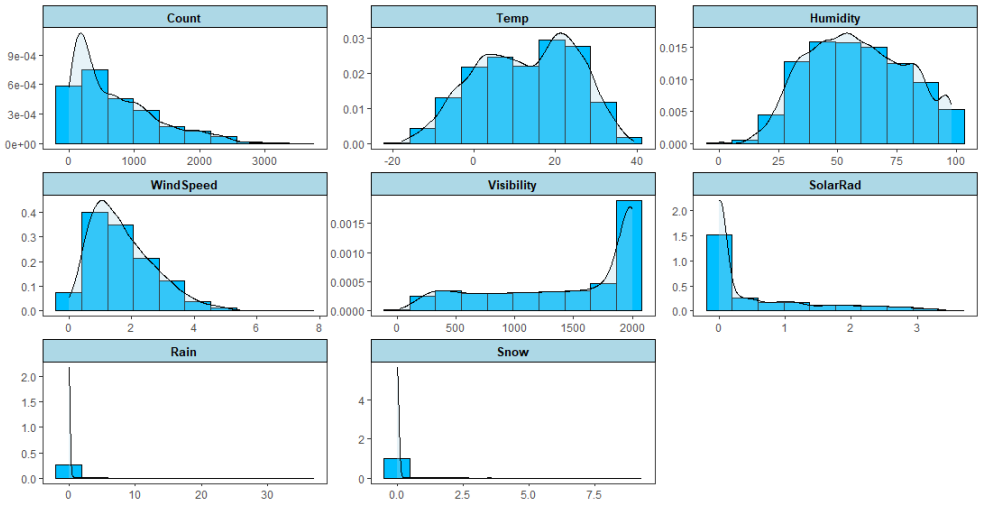

Figura 3: Istogrammi delle variabili numeriche

Gli istogrammi in figura 3 evidenziano invece il diverso campo di variazione dei dati (i.e. visibility in centinaia/migliaia vs windspeed in decine), pertanto per la fase di modeling abbiamo utilizzato i dati scalati. Inoltre, le variabili Rain e Snow hanno una distribuzione molto concentrata sullo zero (zero inflated), che corrisponde ai giorni in cui non si sono registrate precipitazioni. Per questo motivo abbiamo aggiunto due variabili dummy (presenza/assenza del fenomeno per Snow e 3 modalità di intensità per Rain).
In seguito, a partire dalla variabile Date abbiamo creato le variabili categoriche Weekday e Month.

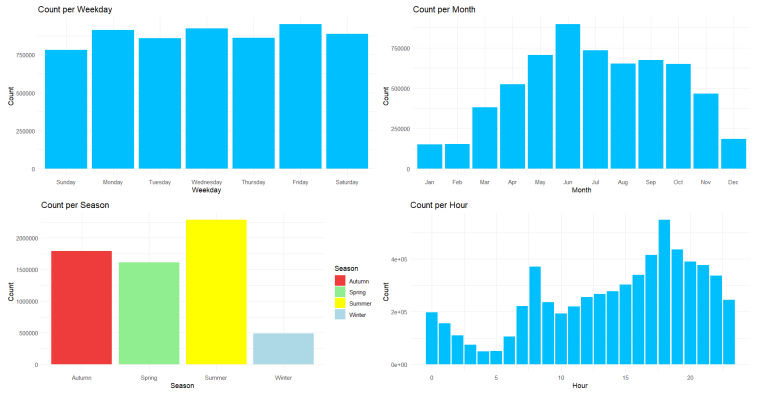

Figura 4: Diagrammi a barre della domanda per giorno della settimana, mese, stagione e orario

I diagrammi a barre (figura 4) evidenziano l'influenza dell'orario (Hour): si registra un picco significativo alle ore 8 del mattino, seguito da un aumento consistente della domanda fino alle ore 18; la domanda cala invece significativamente durante le ore notturne. Intuitivamente, la concentrazione registrata nelle fasce orarie sopracitate coincide con gli orari di inizio e di rientro dal lavoro.
Così come l'orario, anche i mesi (Month) sono significativi. Si registrano aumenti della domanda durante i mesi tra maggio e settembre, con un picco significativo durante giugno. Questo è in linea con quanto descritto dalla variabile Season: la stagione calda incentiva l'utilizzo delle biciclette, al contrario dell'inverno caratterizzato da condizioni atmosferiche tipicamente più ostili.
Il grafico 5 evidenzia invece due trend ben distinti per i giorni infrasettimanali (Mon-Fri) e per il weekend (Sat-Sun). Al fine di cogliere anche questo andamento abbiamo creato la variabile Weekend.

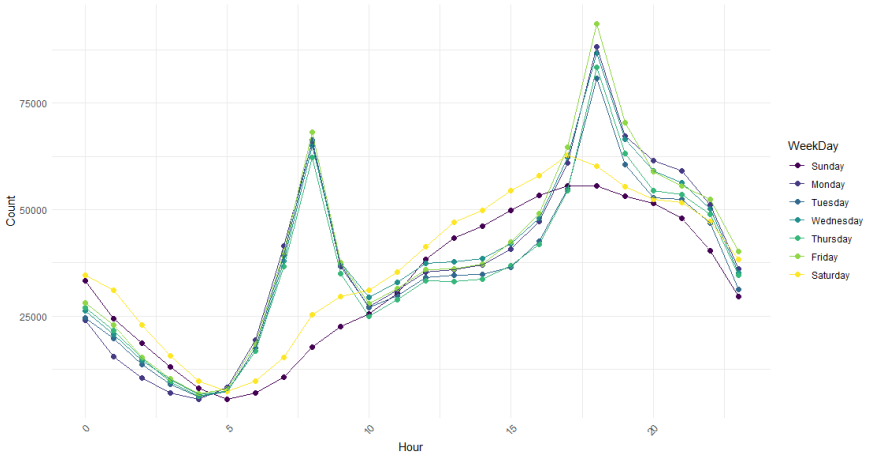

Figura 5: Andamento della domanda per orario e giorno della settimana

2.2 Individuazione punti anomali

Sfruttando il Local Outlier Factor (LOF) score abbiamo individuato i punti "candidati" a valori anomali. L'algoritmo valuta l'anomalia di un punto confrontando la densità locale del punto stesso con quella dei suoi vicini. La soglia utilizzata nell'analisi è stata pari 1.5 (come di consuetudine in letteratura).
I punti individuati rappresentano solamente il 2% del totale delle osservazioni, pertando li abbiamo analizzati per comprendere se si trattasse di isolated anomaly (dati dovuti al rumore intrinseco del fenomeno) oppure di anomalie vere e proprie (dovute a errori di misurazione).
Da questa attenta analisi è emersa la presenza di valori con % di umidità nulla, che abbiamo deciso di eliminare, essendo impossibile in condizioni naturali. Non essendoci altre problematiche altrettanto evidenti abbiamo deciso di non eliminare i restanti outlier.

2.3 Importanza delle variabili

Prima di concludere l'analisi preliminare abbiamo valutato l'influenza delle variabili nel determinare la domanda di biciclette noleggiate, sfruttando il metodo Boruta (si veda l'appendice B 1).

I risultati ci hanno permesso di concludere che tutte le variabili, seppure in percentuale diversa, siano importanti. Tra tutte, Hour è risultata decisamente la più significativa.
Occorre però tenere conto della complessa natura della variabile: potrebbe essere trattata come dummy, ma questo comporterebbe l'introduzione di un numero troppo elevato di nuove variabili; allo stesso tempo, se la si considerasse come una semplice variabile numerica, significherebbe ignorare la sua natura ciclica.
Un metodo molto utilizzato in letteratura per contrastare questo fenomeno è l'utilizzo delle sinusoidi. Abbiamo pertanto creato le due nuove variabili: $\text{sinHour} = \sin\left(\frac{2\pi \cdot \text{Hour}}{24}\right)$ e $\text{cosHour} = \cos\left(\frac{2\pi \cdot \text{Hour}}{24}\right)$.

2.4 Clustering

L'obiettivo principale del clustering è stato quello di supportare l'analisi preliminare e di valutare quali variabili determinano l'associazione di osservazioni simili.

2.4.1 K-means

Il metodo k-means viene applicato alle variabili numeriche. Per la scelta del numero dei cluster da generare si è ricorso alla Sum of Squared Quantities (SSQ) e alla Silhouette.
Il primo metodo suggerisce k=4 mentre il secondo k=2; pertanto abbiamo deciso di generare entrambi i clustering.

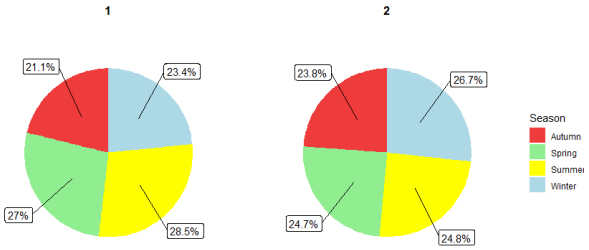

Figura 6: Grafico risultati 2-means

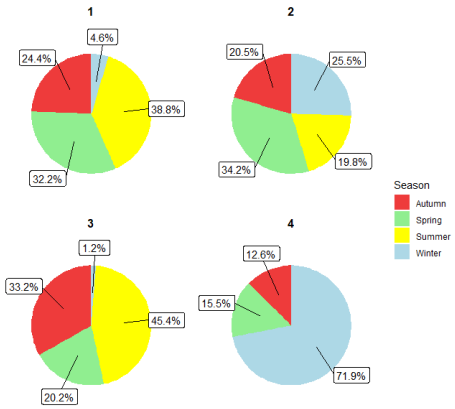

Figura 7: Grafico risultati 4-means

Cluster	Count	SolarRad	Temp	Humidity	WindSpeed	Visibility	Rain	Snow
1	995	1.42	15.9	41.4	2.48	1691.0	0.0023	0.0379
2	583	0.0979	11.0	67.6	1.31	1291.0	0.231	0.0999

Tabella 2: Centroidi per 2-means

Cluster	Count	SolarRad	Temp	Humidity	WindSpeed	Visibility	Rain	Snow
1	1132	1.98	22.1	42.5	2.34	1609.0	0.0014	0.001
2	449	0.144	10.9	78.5	1.30	621.0	0.475	0.178
3	991	0.120	19.5	65.4	1.33	1804.0	0.0714	0.0036
4	383	0.262	-1.40	41.5	2.10	1782.0	0.0013	0.118

Tabella 3: Centroidi per 4-means

Dai grafici 9 e 10 si può notare come 4-means abbia migliori capacità discriminanti per quanto riguarda la variabile Season. Concentrandoci sulla divisione in 4 gruppi notiamo che nel cluster 4 sono racchiuse le osservazioni invernali, mentre i cluster 1 e 3 isolano quelle estive. Il cluster 2 sembra essere invece quello più eterogeneo.
Si nota che l'algoritmo fa invece fatica a riconoscere le stagioni Autumn e Spring e questo può intuitivamente essere dovuto al fatto che, durante le mezze stagioni, le condizioni atmosferiche sono meno estreme.
Guardando ai centroidi è possibile comprendere cosa ha generato la divisione di Summer. I cluster 1 e 3 registrano il maggior numero di noleggi, in linea con quanto ci si aspetta durante i mesi estivi. Il primo gruppo è caratterizzato però da un punteggio della variabile Solar Radiation molto più alto rispetto a quello del terzo.
Possiamo quindi concludere che la discriminante tra i due gruppi sia proprio questa: nel cluster 3 sono contenute le giornate estive, o comunque con temperature elevate, che non sono state caratterizzate dalla presenza del sole.
La caratteristica che invece ha determinato la creazione del cluster 2 è Rain. Il secondo gruppo sembra infatti racchiudere i noleggi effettuati durante le giornate in cui si sono registrate precipitazioni significative e di conseguenza anche la variabile Visibility riporta valori piuttosto bassi.

I risultati del clustering 2-means sono invece di più difficile interpretazione. Sembra infatti che le condizioni atmosferiche non siano particolarmente influenti, dato che non osserviamo grandi differenze nei valori assunti dai centroidi, si è quindi provato a capire se l'orario potesse aver maggiormente contribuito. Dal grafico 8 è possibile notare come il primo cluster contenga i noleggi effettuati durante la giornata (in giallo), mentre il secondo raggruppi i noleggi serali o notturni (in blu scuro). Invece, gli orari dalle 8 alle 10 e dalle 18 alle 20 (in rosso), che possono essere considerato "di passaggio", accomunano entrambi i cluster.

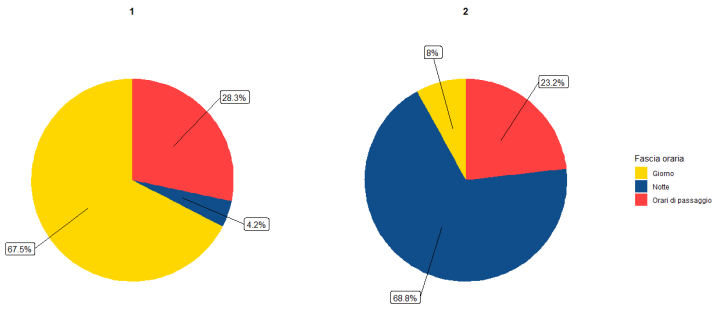

Figura 8: Grafico 2-means per fasce orarie

2.4.2 Algoritmo Gerarchico: Bottom-Up

La Silhouette suggerisce k=2, ma dato che per k=4 il valore medio è comunque elevato e il risultato potrebbe essere più rappresentativo, abbiamo proceduto con entrambi.

Come metodo, si è utilizzato quello di Ward, il quale minimizza la somma dei quadrati delle differenze all'interno dei cluster. Sono stati applicati anche altri metodi (single, complete e average linkage), ma i risultati non sono stati soddisfacenti.

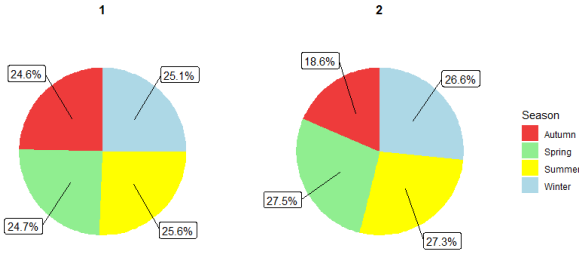

Figura 9: Grafico risultati gerarchico (k=2)

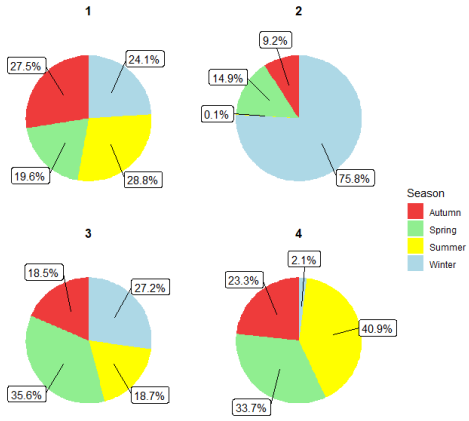

Figura 10: Grafico risultati gerarchico (k=4)

Cluster	Count	SolarRad	Temp	Humidity	WindSpeed	Visibility	Rain	Snow
1	329.0	0.262	-2.35	40.8	1.99	1822.0	0.0006	0.125
2	1031.0	0.148	20.0	64.6	1.53	1786.0	0.0805	0.0017
3	447.0	0.116	9.86	76.3	1.29	708.0	0.437	0.175
4	1100.0	1.96	21.8	43.4	2.30	1576.0	0.0013	0.0019

Tabella 4: Centroidi per gerarchico con k=4

Cluster	Count	SolarRad	Temp	Humidity	WindSpeed	Visibility	Rain	Snow
1	622.0	0.167	10.2	62.6	1.56	1392.0	0.192	0.0998
2	1100.0	1.96	21.8	43.4	2.30	1576.0	0.0013	0.0019

Tabella 5: Centroidi per gerarchico con k=2

I risultati ottenuti per k=4 sono analoghi a quelli generati dall'algoritmo k-means. Per quanto riguarda k=2, invece, in questo caso, le condizioni atmosferiche sono state maggiormente determinanti, come si può vedere dai valori assunti dal centroide per la variabile Temperature.

Possiamo quindi concludere che i metodi k-means e botton-up hanno entrambi la capacità di discriminare in modo abbastanza preciso le stagioni estate e inverno, in cui la domanda, rappresentata da Count, ha una variazione molto elevata a causa delle condizioni atmosferiche molto diverse. È inoltre in grado di raggruppare in un unico cluster le giornate in cui si registra una forte pioggia, fattore che ovviamente incide sulla domanda.

2.4.3 Partitioning Around Medoids (PAM)

Dato che i due metodi applicati precedentemente utilizzano soltanto le variabili numeriche, si è deciso di applicare la tecnica Partitioning Around Medoids, che utilizza la distanza di Gower e permette di introdurre anche le variabili categoriche. In questo caso la Silhouette ha indicato k=5.

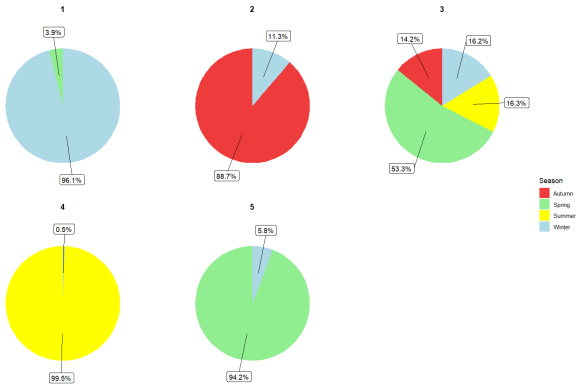

Figura 11: Grafico PAM con k=5

Cluster	Count	SolarRad	Temp	Humidity	WindSpeed	Visibility	Rain	Snow
1	244.0	0.388	-2.97	45.2	2.02	1608.0	0.018	0.227
2	890.0	0.468	12.2	57.6	1.52	1585.0	0.087	0.065
3	518.0	0.55	12.5	66.8	1.40	976.0	0.255	0.094
4	1073.0	0.761	26.7	64.8	1.65	1516.0	0.245	0.006
5	772.0	0.657	12.9	57.7	2.02	1294.0	0.167	0.011

Tabella 6: Centroidi per PAM con k=5

Si nota dal grafico 11 come la variabile Season sia la più determinante nelle partizioni.
Data questo risultato, i centroidi sintetizzano le condizioni atmosferiche tipiche di ogni stagione e la variabile Count rappresenta la media delle biciclette noleggiate, in base al periodo dell'anno. Durante l'analisi si sono anche studiati i grafici condizionati alle altre variabili categoriche, ma non si è pervenuta nessuna influenza significativa.
Il cluster 3 è, al contrario degli altri, decisamente eterogeneo e questo potrebbe essere dovuto alla variabile Visibility, la quale è significativamente più bassa e influisce negativamente su Count.

Si può quindi concludere che la variabile che più determina la domanda è la stagione e di conseguenza le sue condizioni atmosferiche.

3 Regressione

Al fine di utilizzare le variabili categoriche nei successivi algoritmi, per ognuna abbiamo creato tante dummy quante sono le sue modalità, utilizzando la tecnica del One-Hot Encoding. In seguito, abbiamo suddiviso il dataset in training e test set (80-20%).

Come anticipato, la regressione ha come scopo quello di prevedere la domanda di biciclette a noleggo in una data ora a Seoul. A tal fine sono stati utilizzati diversi algoritmi di machine learning: K nearest neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), Neural Network (NN) ed Extreme Gradient Boosting (XGB).

Per tutti gli algoritmi è stata utilizzata la stessa procedura: in primo luogo è stato eseguito il tuning degli iperparametri tramite ottimizzazione bayesiana sfruttando la tecnica 3-fold CV (cross-validation) e scegliendo come metrica di validazione l'MSE, poichè più semplice da minimizzare rispetto alle altre. Una volta individuati i valori ottimali abbiamo addestrato gli algoritmi sull'intero training set e fatto previsione sulle istanze del test set. Infine, per valutare la bontà della regressione sono state utilizzate le seguenti metriche: RMSE, MAE, MAPE e R^2.

L'MSE non è stato inserito poichè di difficile interpretazione, a causa dell'elevamento alla seconda dell'unità di misura dei dati.

3.0.1 K-nearest Neighbors (KNN)

Il KNN (K-nearest Neighbors) per la regressione è uno degli algoritmi non parametrici di machine learning più semplici. Esso predice il valore target di una nuova istanza calcolando la media dei valori dei K vicini più prossimi, identificati utilizzando una specifica metrica di distanza. Di conseguenza la performance dipende principalmente dalla scelta di K e dalla metrica di distanza adottata.
Il processo di ottimizzazione bayesiana degli iperparametri del modello porta a scegliere k = 8 e la distanza di Manhattan (maggiormente discriminante in caso di elevata dimensionalità e meno sensibile agli outlier rispetto alla euclidea).

RMSE	250.982
MAE	156.4065
R^2	0.8414
MAPE	65.4161

Tabella 7: Risultati delle metriche di valutazione per KNN

La media dei valori dell'RMSE valutati nel CV è risultata pari a 266.0419, che non si discosta significativamente dall'RMSE ottenuto nel test. Questo indica che probabilmente il modello non cade in overfitting. Dai risultati nella tabella 7 possiamo trarre conclusioni contrastanti: l'R^2 ci dice che circa l'85% della variabilità dei dati è spiegata dal KNN, suggerendo che il modello è in grado di catturare bene la relazione tra le features e la target, ma il MAPE indica che le previsioni del modello si discostano in media del 70% rispetto ai

valori osservati. Questo valore è piuttosto alto e suggerisce che, nonostante un buon R^2, il modello potrebbe avere difficoltà con valori di domanda di bici molto bassi o molto alti. Valutando invece l'RMSE, in media, le previsioni del modello possono essere distanti di circa 250 bici rispetto alla domanda effettiva, mentre osservando il MAE siamo in grado di affermare che, in media, le previsioni del modello si discostano di circa 156 bici dalla domanda reale.

3.0.2 Support Vector Regression (SVR)

Il Support Vector Machine (SVM) è un algoritmo di apprendimento supervisionato che può essere utilizzato per problemi di regressione, in letteratura viene denotato anche come Support Vector Regression (SVR). SVR cerca di trovare una funzione che sia il più vicina possibile ai dati di addestramento, mantenendo un margine di tolleranza per l'errore. Inoltre, attraverso l'utilizzo del kernel trick, può gestire dati non linearmente separabili utilizzando trasformazioni dei dati in un nuovo spazio.
I principali iperparametri di SVM sono: cost, un parametro di regolarizzazione che gestisce il compromesso tra l'ampiezza del margine e l'errore di previsione, e la funzione kernel, che se scelta radiale porta ad un nuovo iperparametro gamma, che controlla la smoothness del confine decisionale.
Nel processo di ottimizzazione bayesiana degli iperparametri del modello sono stati provati i quattro kernel più noti in letteratura (linear, radial, polynomial e sigmoid), e la scelta è ricaduta su:

cost	35.0394
kernel	radiale
gamma	0.0591

Tabella 8: Risultati del tuning SVM

RMSE	337.0638
MAE	211.4919
R^2	0.7139
MAPE	71.9878

Tabella 9: Risultati delle metriche di valutazione per SVM

I valori degli iperparametri individuati sembrano discordi: il basso valore di gamma, che controlla l'influenza di un singolo punto di addestramento, indica che il modello sta cercando di evitare l'overfitting, ma al tempo stesso l'alto valore di cost indica che stiamo dando molto peso alla minimizzazione degli errori, anche a costo di un margine più stretto, rischiando di cadere in overfitting. Questo suggerisce che la separabilità e la struttura dei dati è probabilmente molto complessa. Il valore della media dell'RMSE ottenuti dal CV sembra però confermare i timori di overfitting, infatti è pari a 208.0919, significativamente più basso rispetto al valore ottenuto nel test. Valutando il modello nel test set si sono ottenuti i risultati riportati nella tabella 9, dalla quale possiamo trarre conclusioni simili a quelle fatte per KNN, con la differenza che però ora l'R^2 spiega circa il 70% della variabilità dei dati, mentre il MAPE rimane stabile su un errore del 70%. Inoltre, sia MAE che RMSE sono aumentati, infatti l'errore di previsione della domanda di biciclette si aggira 200 e 340. Considerati i risultati ottenuti, consideriamo questo modello da scartare.

3.0.3 Neural Network (NN)

Come terzo metodo di regressione abbiamo provato ad implementare le NN (Neural Network), in particolare abbiamo allenato una rete neurale composta da un unico hidden layer, in quanto è stato dimostrato che, in presenza di un numero sufficiente di neuroni, risulta in grado di approssimare una qualunque funzione

continua.
Gli iperparametri ottimizzati con il tuning bayesiano ci portano a costruire una NN con un unico hidden layer formato da 20 neuroni e con un tasso di decadimento di 0.0881. Quest'ultimo valore, essendo molto vicino allo zero, indica un lento aggiornamento dei pesi, che potrebbe portare a cadere in un ottimo locale invece che globale durante la fase di ottimizzazione.

La rete testata sul test set produce i risultati in tabella 10.

RMSE	332.9518
MAE	233.6329
R^2	0.7208
MAPE	99.2265

Tabella 10: Risultati delle metriche di valutazione per NN

Il CV RMSE risulta pari a 296.7922, leggermente inferiore a quello ottenuto sul test. Valutando le metriche riportate in tabella 10 traiamo conclusioni praticamente identiche ad SVM, eccezion fatta per il MAPE, che peggiora drasticamente, portando l'errore di previsione addirittura al 100%, rendendo queste semplici NN il peggior modello testato fin'ora.
Non abbiamo optato per reti neurali ancora più complesse, in quanto i modelli che ora seguiranno hanno portato ai risultati migliori, facendoci capire che un approccio basato su alberi decisionali risultava essere il migliore per la problematica in questione.

3.0.4 Random Forest (RF)

Il penultimo modello testato è stato RF (Random Forest), un metodo di ensemble che costruisce una foresta di alberi decisionali durante l'addestramento e restituisce la media delle previsioni degli alberi individuali.
In questo caso gli iperparametri sono due: il numero di variabile da campionare (mtry) e il numero di alberi che compongono la foresta (ntrees).
I migliori risultati sono stati ottenuti con 454 alberi e 11 variabili da campionare.

RMSE	166.2253
MAE	98.8553
R^2	0.9304
MAPE	48.8015

Tabella 11: Risultati delle metriche di valutazione per RF

La media dei valori dell'RMSE nel CV è pari a 179.0784, prossimo al valore valutato sul test set. Osservando i risultati riportati nella tabella 11 notiamo che questo è il miglior modello testato fin'ora. L'R^2 ora supera il 90%, mentre il MAPE è sceso sotto il 50% ed anche MAE ed RMSE sono calati drasticamente, portando l'errore per la domanda di biciclette ad essere compreso, in media, tra le 100 e le 160 unità.

3.0.5 Extreme Gradient Boosting (XGB)

Dato il successo delle RF abbiamo deciso di implementare un nuovo algoritmo basato sugli alberi decisionali, l'XGB (Extreme Gradient Boosting), il quale è stato proposto nel 2014 e sta spopolando nel mondo del machine learning. (Per gli approfondimenti teorici si veda l'appendice B 3).

In questo caso i principali iperparametri da ottimizzare sono:

- **Learning rate**: controlla la velocità di apprendimento dell'algoritmo. Un valore più basso rende l'addestramento più lento ma può migliorare la precisione.
- **Max depth**: la profondità massima degli alberi. Un valore maggiore può portare a modelli più complessi e potenzialmente overfit.
- **Subsample**: la frazione del dataset da utilizzare per costruire ogni albero. Valori più bassi possono aiutare a prevenire l'overfitting.
- **Colsample by tree**: la frazione delle features utilizzate per costruire ogni albero. Aiuta a ridurre la correlazione tra gli alberi.
- **Gamma**: il parametro di riduzione della perdita minima per una nuova divisione. Valori più alti rendono il modello più conservativo.
- **Lambda**: la regolarizzazione L2 sui pesi delle foglie. Aiuta a prevenire l'overfitting.
- **Alpha**: la regolarizzazione L1 sui pesi delle foglie. Aiuta a selezionare un sottoinsieme delle caratteristiche più rilevanti.
- **Min child weight** : il peso minimo richiesto per creare una nuova partizione in un nodo. Valori più alti rendono il modello più conservativo.

Utilizzando l'ottimizzazione bayesiana abbiamo ottenuto i valori centenuti nella tabella 12.

eta	0.0486
max depth	9
subsample	0.6643
colsample by tree	0.9595
gamma	3.4441
lambda	1.9228
alpha	1.3264
min child weight	1.9177

Tabella 12: Risultati del tuning XGB

RMSE	149.7174
MAE	83.7227
R^2	0.9435
MAPE	36.6394

Tabella 13: Risultati delle metriche di valutazione per XGB

La media dell'RMSE sul CV è risultata pari a 162.7653, distanziandosi di poco dal valore sul test. Analizzando i risultati ottenuti sul test set, riportati in tabella 13, notiamo un leggero miglioramento rispetto alle RF, infatti il MAPE si aggira ora intorno al 38%, mentre l'R^2 rimane stabile sopra il 90%. Infine, anche MAE e RMSE migliorano, portando l'errore per la domanda di biciclette ad essere compreso, in media, tra le 85 e le 150 unità.

3.1 Risultati

In questa sezione riportiamo un breve riassunto grafico e numerico di quanto detto nelle sezioni ad hoc per ogni algoritmo.

Dalla figura 12 si nota immediatamente che la classifica di performance dei nostri algoritmi vede in ultima posizione le NN (1 hidden layer nascosto), seguite immediatamente da SVM (scartato già in precedenza per le problematiche di overfitting riscontrate) e poi da KNN; mentre le prime due posizioni sono occupate da RF e XGB.
Nella tabella 14 possiamo osservare nel dettaglio i valori per le metriche ottenuti.

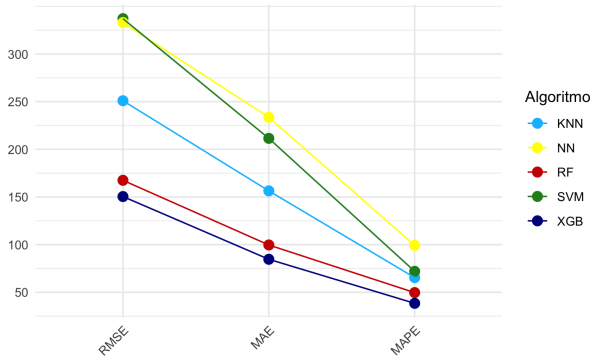

Figura 12: Grafico confronto metriche regressione

Algoritmo	RMSE	MAE	R^2	MAPE
KNN	250.9820	156.4065	0.8414	65.4161
RF	166.2253	98.8553	0.9304	48.8015
SVM	337.0638	211.4919	0.7139	71.9878
NN	332.9518	233.6329	0.7208	99.2265
XGB	149.7174	83.7227	0.9435	36.6394

Tabella 14: Risultati delle metriche di valutazione per tutti gli algoritmi

4 Classificazione

Come menzionato in precedenza, abbiamo ulteriormente sviluppato l'analisi con la classificazione della domanda oraria, dopo averla divisa in 3 fasce, risultate ben bilanciate: Low ($Count < 300$), Medium ($300 < Count < 1000$) e High ($Count > 1000$). L'attività di classificazione è stata svolta sfruttando gli stessi algoritmi di machine learning usati per la regressione, essendo tutti adattabili ad entrambi gli scopi.
Al fine di identificare il modello migliore in termini di capacità di generalizzazione abbiamo utilizzato sia metriche globali (quali accuracy e indice di Gini), che metriche calcolate per classe (precision, recall e f1-score).

La scelta dell'utilizzo dell'indice di Gini è stata dettata dalla sua capacità di cogliere non solo la classe predominante in un cluster, ma anche la distribuzione dei dati tra le classi rimanenti; limite che caratterizza invece l'accuracy. Il calcolo delle metriche per classe ha invece come obiettivo quello di valutare la presenza di una diversa capacità previsiva dei modelli tra le varie fasce di domanda.
La procedura adottata è stata la stessa utilizzata nella regressione, con l'unica ovvia differenza dell'utilizzo dell'accuracy come metrica di validazione.

Data la natura ordinale della variabile risposta, come primo step abbiamo deciso di implementare un modello di regressione logistica ordinale. L'obiettivo di questa fase non è tanto quello di fare previsione (essendo un modello troppo semplice per il problema in esame), ma quanto di fornire una prima interpretazione intuitiva della relazione tra le variabili esplicative e la risposta.
I risultati confermano quanto osservato in precedenza durante l'analisi esplorativa dei dati, ovvero che la domanda di biciclette è fortemente influenzata da due fattori principali: le condizioni atmosferiche e le variabili temporali. In presenza di condizioni atmosferiche avverse (quali pioggia, neve, temperature molto basse, scarsa visibilità, forti raffiche di vento) la richiesta di noleggio cala, mentre vi è una sorta di stagionalità oraria, giornaliera e stagionale. I timori sulla scarsa capacità di generalizzazione del modello sono stati confermati dalle misure di accuratezza calcolate sul test set.

4.1 K-Nearest Neighbors (KNN)

Il KNN per la classificazione funziona in modo analogo al caso della regressione, con l'unica ma sostanziale differenza che per prevedere la target di una nuova istanza applica un voto di maggioranza (moda) delle classi dei K vicini più prossimi. In questo caso il processo di ottimizzazione ha individuato k = 9 e distanza di Manhattan.

Accuracy sul validation	0.7966
Accuracy sul test	0.8096
Indice di Gini	0.3168

Tabella 15: Risultati KNN (1)

Classe	**Precision**	**Recall**	**F1-score**
Low	0.8605	0.8605	0.8605
Medium	0.7468	0.7480	0.7473
High	0.8277	0.8260	0.8269

Tabella 16: Risultati KNN (2)

Nonostante la semplicità del modello, le performance sono accettabili e i valori di accuracy non troppo bassi (Tabella 15). Notiamo subito una differenza tra la classe Medium e le classi più estreme: i valori di precision, recall e f1-score (riportati nella Tabella 16) sono nettamente più bassi nella classe di mezzo, ad indicare una scarsa capacità discriminatoria di quest'ultima. Questo risultato viene confermato anche dal valore dell'indice di Gini, che indica una certa rumorosità dei dati all'interno della matrice di confusione.

4.2 Support Vector Machine (SVM)

SVM per la classificazione si pone come obiettivo quello di individuare il miglior piano di separazione tra i dati, ovvero quello che massimizza il margine. Come già visto in precedenza per la regressione, a causa della complessità del problema in esame, abbiamo utilizzato il kernel radiale, mappando l'input space in un feature space in cui i dati fossero linearmente separabili.
I valori degli iperparametri ottimali individuati dalla ricerca sequenziale sono cost pari a 43.2451 e gamma pari a 0.0408. Per l'interpretazione di questi iperparametri valgono le stesse considerazioni fatte nel caso

della regressione.
Il numero di support vectors (719, 1259 e 649) sembra, a primo impatto, elevato (indice di overfitting), ma dato il numero di osservazioni nel dataset può essere considerato accettabile. Essi, infatti, rappresentano una percentuale compresa tra l'8.2% e il 16.1% del numero totale di istanze del training (data la forte dipendenza dai dati, dal numero di features e dal kernel utilizzato, non esiste una soglia univocamente accettata ma in letteratura si tende a considerare come moderata una percentuale compresa tra il 10% e il 30%).

Accuracy sul validation	0.8426
Accuracy sul test	0.8344
Indice di Gini	0.2827

Tabella 17: Risultati SVM (1)

Classe	**Precision**	**Recall**	**F1-score**
Low	0.8773	0.8744	0.8758
Medium	0.7790	0.7802	0.7796
High	0.8529	0.8547	0.8538

Tabella 18: Risultati SVM (2)

Rispetto all'algoritmo precedente, notiamo un miglioramento delle metriche di prestazione riportate nella Tabella 17. La percentuale di valori correttamente previsti è aumentata, mentre l'indice di Gini indica una minore "confusione" nei dati. La Tabella 18 evidenzia invece una similarità con il KNN: anche in questo caso sembra che il modello fatichi maggiormanete a prevedere la classe Medium rispetto alle due più estreme.

4.3 Neural Network (NN)

Come terzo metodo abbiamo sfruttato le Reti Neurali a singolo hidden layer, per le stesse motivazioni viste in precedenza. Il numero di neuroni ottimale individuato è 11, mentre il tasso di decadimento è 0.0991. Notiamo che il learning rate è piuttosto vicino allo zero, ad indicare che vi è la probabilità che la rete cada in un ottimo locale (invece che in un ottimo globale). Riteniamo però quest'ipotesi poco probabile, viste le performance piuttosto buone del modello, riassunte di seguito.

Accuracy sul validation	0.8674
Accuracy sul test	0.8681
Indice di Gini	0.2332

Tabella 19: Risultati NN (1)

Classe	**Precision**	**Recall**	**F1-score**
Low	0.9025	0.8935	0.8980
Medium	0.8177	0.8298	0.8237
High	0.8908	0.8852	0.8880

Tabella 20: Risultati NN (2)

Le prestazioni del modello migliorano nuovamente: i valori di accuratezza sono leggermente più alti e di confusione più bassi.

4.4 Random Forest (RF)

Siamo quindi passate ai Random Forest, dove i valori degli iperparametri da ottimizzare sono il numero di alberi (500) e il numero di variabili da considerare in ogni nodo nella costruzione di ogni albero (10). Come nel caso del KNN, la previsione viene prodotta attraverso un voto di maggioranza dei valori previsti dai singoli alberi decisionali allenati parallelamente.

Accuracy sul validation	0.8829
Accuracy sul test	0.8746
Indice di Gini	0.2227

Tabella 21: Risultati RF (1)

Classe	Precision	Recall	F1-score
Low	0.9193	0.8823	0.9004
Medium	0.8129	0.8442	0.8283
High	0.8991	0.9029	0.9011

Tabella 22: Risultati RF (2)

Le Random Forest portano ad un ulteriore aumento della capacità previsiva del modello, con un leggero miglioramento di tutte le metriche. Ancora una volta, la classe Medium si conferma essere la più svantaggiata tra le tre.

4.5 Extreme Gradient Boosting (XGB)

L'ultimo algoritmo testato è stato XGB anche in questo caso. Utilizzando l'ottimizzazione bayesiana abbiamo ottenuto i seguenti valori per gli iperparametri (tabella 23):

eta	0.0886
max depth	9
subsample	0.9029
colsample by tree	0.7359
gamma	0.7294
lambda	1.0340
alpha	0.1571
min child weight	0.8114

Tabella 23: Risultati del tuning XGB

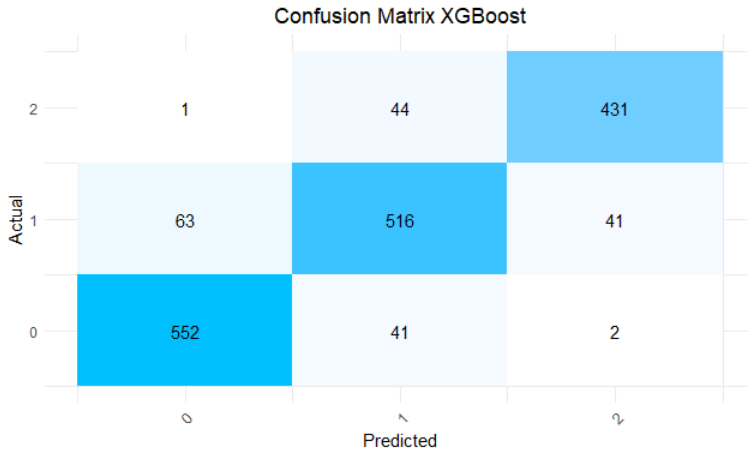

Figura 13: Confusion Matrix Extreme Gradient Boosting

Accuracy sul validation	0.8855
Accuracy sul test	0.8864
Indice di Gini	0.2043

Tabella 24: Risultati XGB (1)

Classe	Precision	Recall	F1-score
Low	0.9277	0.8961	0.9116
Medium	0.8323	0.8586	0.8452
High	0.9055	0.9093	0.9074

Tabella 25: Risultati XGB (2)

I risultati riportati nelle Tabelle 24 e 25 indicano ottime capacità previsive. Abbiamo riportato in questo caso anche la matrice di confusione, date le eccellenti performance.

4.6 Risultati

Terminata l'analisi sui singoli modelli abbiamo confrontato tutti i risultati ottenuti tramite le metriche riassunte nella tabella 26, al fine di individuare, se presente, l'algoritmo migliore per il problema in esame.

Algoritmo	Accuracy sul validation	Accuracy sul test	Indice di Gini
KNN	0.7966	0.8096	0.3168
SVM	0.8426	0.8344	0.2827
NN	0.8674	0.8681	0.2332
RF	0.8829	0.8746	0.2228
XGB	0.8856	0.8865	0.2043

Tabella 26: Risultati delle metriche di valutazione

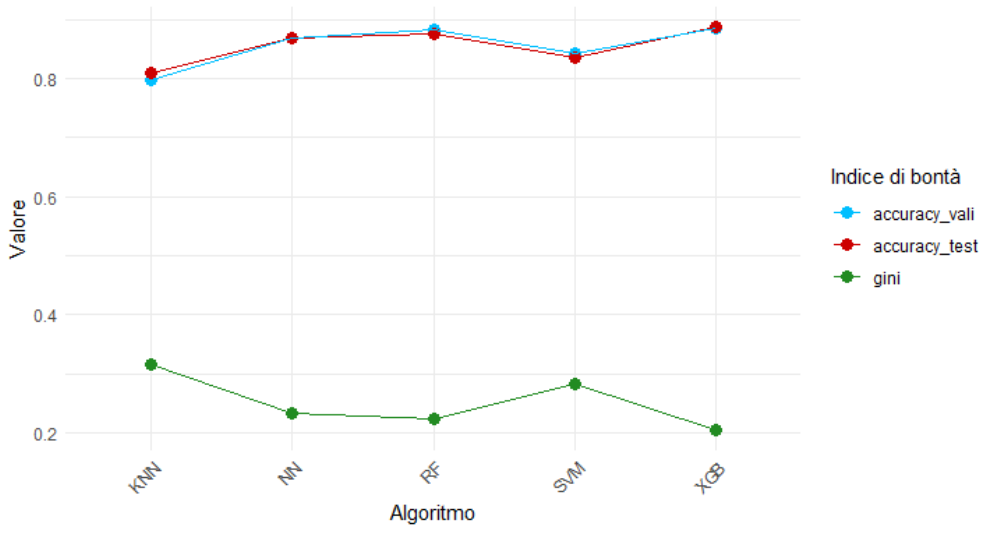

Figura 14: Risultati delle metriche di valutazione

Le metriche evidenziano che XGB è sicuramente l'algoritmo che performa meglio: è caratterizzato dall'accuratezza più elevata e dall'indice di Gini minore. Questo è seguito da RF e NN, che hanno performance

assimilabili (lievemente migliori per RF), SVM e all'ultimo posto troviamo KNN (che si discosta di quasi 9 punti percentuali per l'accuracy e circa 11 per Gini dal modello XGB). Abbiamo dunque deciso di "scartare" quest'ultimo, essendo quello che performa peggio.

Per una maggiore comprensione del problema e al fine di confrontare anche visivamente le performance dei modelli abbiamo scelto le curve ROC, disegnate a partire dalle probabilità di appartenenza alle categorie (Low, Medium e High) di ogni punto e rappresentate nella figura 15.

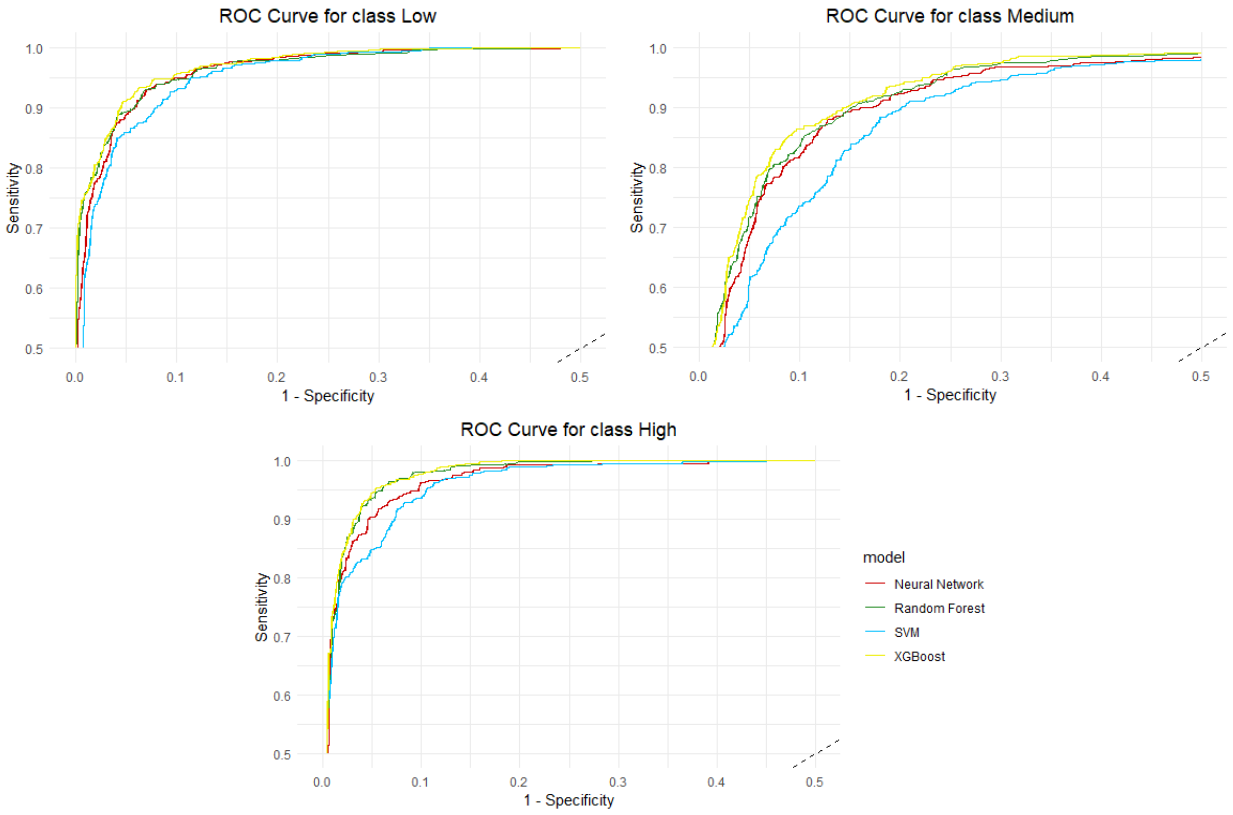

Figura 15: Curve ROC

Le curve avvalorano quanto osservato in precedenza: tra i 4 algoritmi SVM è quello con performance peggiori, trovandosi al di sotto delle altre per tutte e tre le classi. I restanti algoritmi hanno un comportamento molto simile nel discriminare la classe Medium e Low, mentre NN ha performance peggiori nella classe High.

Infine, osserviamo che l'AUC (Area Under The Roc Curve) per la classe Medium e' inferiore rispetto alle altre due classi, per tutti gli algoritmi considerati. Questo denota una minore capacità discriminatoria (seppur comunque abbastanza buona) della classe Medium rispetto alle due classi più estreme. Questo risultato era in realtà già stato evidenziato in precedenza dalle tabelle contenenti precision, recall e f1-score: in tutti

gli algoritmi allenati i valori relativi alla seconda classe risultano sempre inferiori rispetto agli altri.

Abbiamo poi deciso di soffermarci sull'algoritmo di boosting per cercare di visualizzare il problema di classificazione. Per la rappresentazione è stato ovviamente necessario considerare solo due variabili alla volta, selezionate tra quelle che influenzano maggiormente la creazione delle classi.

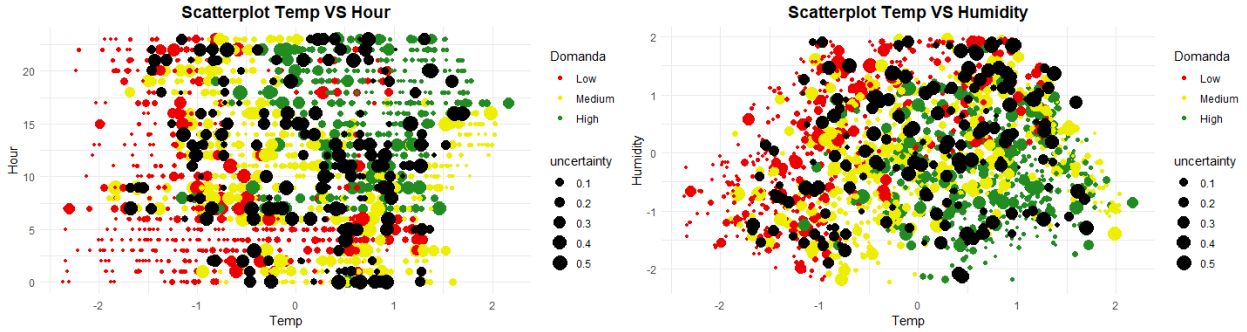

Figura 16: Scatterplot dei punti

È evidente che le rappresentazioni siano molto riduttive e non permettano di cogliere la complessità del problema in esame, dato che consideriamo solo due variabili per volta rispetto alla totalità di quelle in gioco.

Abbiamo infine provato a rappresentare un grafico tridimensionale al fine di cogliere meglio la differenziazione tra le classi. Ovviamente, anche in questo caso la rappresentazione non riesce a cogliere la totalità del problema.

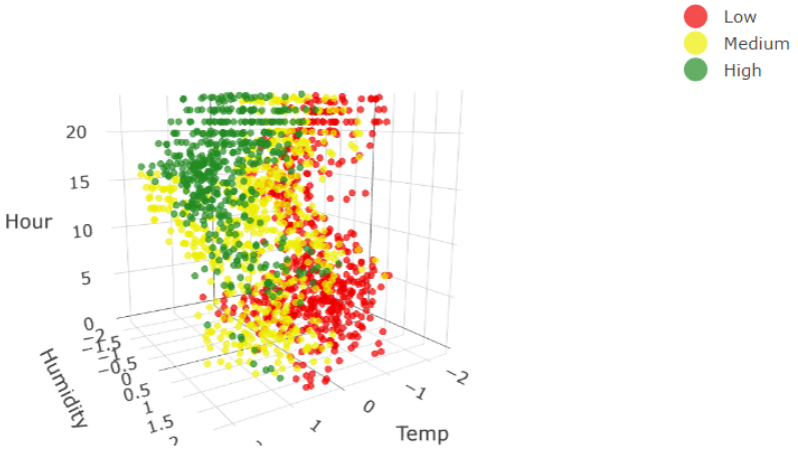

Figura 17: Rappresentazione tridimensionale

4.7 Combinazione dei risultati

In quest'ultima fase abbiamo deciso di "combinare" gli esiti dei 4 algoritmi allenati in precedenza. L'obiettivo è quello di aumentare la robustezza delle stime sfruttando i diversi approcci e sistemi di ipotesi dei vari algoritmi, al fine di attenuare errori sistematici che potrebbero essere prodotti da uno specifico algoritmo. L'idea sottostante è dunque assimilabile ad un "ensamble learning". Il costo computazionale che richiederebbe questa operazione per una ipotetica fase di previsione per nuove istanze è ovviamente molto elevato: è infatti necessario l'allenamento di 4 diversi algoritmi di machine learning, tutti molto complessi. Nonostante ciò, abbiamo deciso di analizzare i risultati di questa operazione, ritenendo possano essere interessanti per la nostra analisi.

Abbiamo combinato i risultati e, attraverso un voto di maggioranza, imputato l'etichetta maggiormente assegnata (quando possibile) e il valore "Unknown" nel caso di massima incertezza, ovvero quando due algoritmi riportavano risultati diversi dagli altri due.

In seguito abbiamo osservato l'assenza di unità classificate erroneamente ma nella stessa classe da tutti gli algoritmi, mentre sono presenti 75 osservazioni per cui vi è massima incertezza (valori "Unknown"). Abbiamo dunque deciso di analizzarli e studiarne le caratteristiche, per comprendere quali possano essere i motivi che inducono gli algoritmi al disaccordo.

Una buona parte sono valori che si trovano sulle "soglie" che separano le classi (o tra Low e Medium oppure tra Medium e High): non siamo dunque stupiti che gli algoritmi non siano concordi nella loro classificazione. Notiamo inoltre che i valori appartenenti alla classe "Medium" sono anch'essi molto numerosi: questo conferma quando visto in precedenza sulla maggiore incertezza degli algoritmi nell'identificazione degli elementi di questa classe.

È inoltre interessante analizzare alcuni casi specifici, per esempio:

- Il 09/07/2018 sono state noleggiate solamente 59 biciclette alle ore 15: essendo un giorno infrasettimanale ed essendoci una temperatura piuttosto mite, non ci stupisce che i modelli possano prevedere una domanda più elevata;
- Il 13/06/2018 sono state noleggiate ben 1274 biciclette a mezzanotte: essendo un orario notturno, nuovamente non siamo stupite che i modelli prevedano una domanda più bassa;
- Il 05/12/2017, nonostante ci fossero quasi -8 gradi, è stata osservata una domanda pari a 462 bici. Una ricerca più attenta ci ha portato a scoprire che quel giorno vi è stato uno sciopero dei mezzi di trasporto pubblico a Seoul, dunque la domanda elevata potrebbe essere dovuta a questo fattore, che i modelli ovviamente non riescono a cogliere.

5 Conclusioni

In sintesi, il clustering (supportato dall'EDA) ha evidenziato che la domanda di biciclette è influenzata significativamente da variabili metereologiche, come la temperatura, la visibilità e le precipitazioni.
La stagionalità e l'orario del giorno permettono anch'essi di determinare i picchi di domanda.
Nonostante la complessità riscontrata nel prevedere esattamente il numero di biciclette noleggiate, i modelli

di regressione RF e XGB hanno dimostrato una buona capacità predittiva, sbagliando in eccesso o in difetto di circa 150 biciclette, rispetto ai loro concorrenti KNN e NN, che sforano del doppio rispetto ai precedenti, e a SVM, che cade in overfitting.
La suddivisione della domanda in classi ha permesso una migliore gestione della previsione. Il modello XGB è risultato il migliore (con un'accuracy quasi del 90%), senza però segnare un distacco netto da NN e RF, mentre SVM e KNN risultano essere i fanalini di coda. Inoltre, tutti gli algoritmi di classificazione riescono ad etichettare molto bene le fasce di domande estreme, mentre per la classe medium hanno una minore capacita' discriminatoria. Questa classe è infatti caratterizzata da valori nella media, che rendono più difficile per gli algoritmi la distinzione.
In conclusione, questo progetto ha dimostrato che l'uso combinato di analisi esplorativa, tecniche di preprocessing e modelli avanzati di machine learning può fornire insight preziosi e previsioni accurate sulla domanda di biciclette. Questi risultati possono essere utilizzati per ottimizzare la gestione del servizio di noleggio biciclette, migliorando l'esperienza degli utenti e l'efficienza operativa.

Riferimenti bibliografici

[1] Ten methods to assess Variable Importance: https://f0nzie.github.io/machine_learning_compilation/ten-methods-to-assess-variable-importance.htm

[2] https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.848169/full

[3] https://development.asia/case-study/how-seoul-eased-traffic-congestion-and-reduced-pollution-through

[4] https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A689876&dswid=3572

[5] XGBoost: A Scalable Tree Boosting System: https://arxiv.org/abs/1603.02754

[6] XGBoost in R: https://xgboost.readthedocs.io/en/latest/R-package/xgboostPresentation.html

[7] Kursa, Miron B., and Witold R. Rudnicki. "Feature selection with the Boruta package." Journal of Statistical Software

[8] Gower, J. C. (1971) A general coefficient of similarity and some of its properties, Biometrics

[9] https://english.seoul.go.kr/subway-line-9-strike-seoul-city-implements-emergency-transport-measures/

A Appendice: Metodi

1. **Boruta**
   L'algoritmo Boruta serve all'identificazione delle variabili più significative all'interno di un dataset ed è basato sull'algoritmo di classificazione Random Forest.
   Il metodo consiste nella creazione di nuove variabili fittizie dette shadow, che sono dei duplicati delle originali ma dove i dati sono mescolati in modo casuale. Viene poi addestrato un modello Random Forest, utilizzando sia le variabiabili originali che quelle shadow.
   La funzione di importanza, propria dell'algoritmo Random Forest, assegna un grado di importanza ad ogni variabile. Se una variabile originale ho un'importanza significativamente maggiore rispetto alla migliore delle shadow, allora questa è considerata influente. Se al contrario non si distingue dalle shadow, le quali sono ovviamente non significative, allora è considerata non importante.

2. **Partitioning Around Medoids e distanza di Gower**
   Il PAM è un metodo di clustering che si basa sulla ricerca dei medoidi, ossia i punti che minimizzano la distanza intra cluster e che sono essi stessi osservazioni del dataset.
   Dato che l'algoritmo è stato implementato su variabili numeriche e categoriche, si è ricorso alla misura di distanza di Gower. Tale indice si presenta come una sorta di media pesate di misure di diverso tipo, in base alla tipologia di variabili che si stanno considerando, dove i pesi sono proporzionali alla contribuzione di ciascuna variabile.
   Per le variabili numeriche utilizza la distanza euclidea e per quelle ordinali, la distanza dipende dalla differenza nelle posizioni ordinali.
   Per le categoriche, la distanza tra due osservazioni è 0 se una data variabile assume stessa modalità e 1 altrimenti. Quindi, limitandosi a considerare solo le variabili categoriche, la distanza tra due osservazioni è la proporzione di variabili in cui differiscono per la modalità assunta.

3. **XGBoost**
   XGBoost (Extreme Gradient Boosting) è un algoritmo di machine learning introdotto da Tianqi Chen nel 2014 che ha rapidamente guadagnato popolarità per la sua efficienza computazionale e le sue ottime prestazioni in numerosi compiti di regressione e classificazione.
   E' basato sul gradient boosting, una tecnica di ensemble che combina molti modelli deboli (i.e. alberi decisionali poco profondi) in un modello forte al fine di migliorare le previsioni.
   Dunque, a differenza delle RF, l'XGB costruisce modelli in sequenza, dove ogni nuovo modello cerca di correggere gli errori del modello precedente. Questo processo continua fino a quando gli errori non sono minimizzati. L'idea è dunque di concentrarsi sui dati che i modelli precedenti hanno trattato male, migliorando iterativamente le prestazioni complessive.
   Sia per la regressione che per la classificazione, XGBoost ottimizza una funzione obiettivo che combina una funzione di perdita e alcuni termini di regolarizzazione.
   La funzione di perdita misura quanto le previsioni si discostano dai valori/etichette reali, mentre i due termini di regolarizzazione usati sono L1 (lasso), che penalizza la complessità del modello, e L2 (ridge) che previene l'overfitting.

4. **Curve ROC**
   Le curve ROC vengono utilizzate per valutare la capacità discriminatoria di un test, tracciando 1-specificità (sull'asse delle x) e sensitività (sull'asse delle y) per una serie di punti di "cut-off" o valori soglia.
   La sensitività (percentuale di osservazioni correttamente classificate come la classe di interesse) e la

specificità (percentuale di osservazioni correttamente classificate come l'altra classe) sono le due misure principali che vengono utilizzate per valutare la capacià di un test di individuare gli individui provvisti di un carattere e quelli che ne sono privi.
Più l'area sotto la curva (AUC, Area Under The Roc Curve) si avvicina ad 1, migliore è il modello.
Nonostante nascano per problemi di classificazione a 2 classi possono essere estese anche ai casi multiclasse, utilizzando due approcci diversi: One vs One oppure One vs Rest.
Il primo approccio richiede di creare un classificatore binario per ogni coppia di classi. Il secondo invece, adottato in questo report, confronta ogni classe con tutte le altre, producendo dunque un numero di curve pari al numero di classi.