Big Data Analytics su dataset “Hotel Reviews” con Apache Spark

Giuseppe Pasquale Caligiure
Mat. 280867

Contesto di lavoro e obiettivo

L’archivio “Hotel Reviews” contiene 515.738 recensioni di hotel di lusso in Europa, raccolte da Booking.com

- Ogni recensione presenta 17 campi
- Formato File: csv
- Dimensioni File: circa 238 MB
- Reperibile su: 515K Hotel Reviews Data in Europe

Obiettivo: realizzare un’app interattiva che consenta di effettuare interrogazioni sul dataset per **estrarre insight significativi dai dati**, sfruttando le potenzialità di elaborazione distribuita del framework Spark

Architettura Frontend/Backend

Backend (PySpark): la **logica di lavoro delle interrogazioni** è stata realizzata sfruttando il motore di calcolo distribuito in-memory di PySpark, cioè le API Python per Apache Spark, ed è contenuta nel file “queries.py”

- Ogni query accetta un DataFrame Spark in input e restituisce un DataFrame Spark contenente i risultati
- Altre tecnologie utilizzate: Spark SQL, Spark MLlib, Scikit-learn (in UDF Pandas)

Frontend (Streamlit): l’interfaccia utente è gestita da una **web-app** interattiva realizzata con il framework Streamlit.

- La UI consente all’utente di selezionare un’analisi da effettuare, invoca la corrispondente funzione dal backend e infine visualizza i risultati ottenuti.
- Altre tecnologie utilizzate: Altair e PyDeck per mappe geospaziali, Pandas per grafici interattivi

Analisi dei dati

Le query implementate esplorano diversi metodi di analisi dei dati, fra cui:

- **Analisi Descrittiva**: sintetizzare e riassumere caratteristiche di dati storici (metriche chiave come media e deviazione standard).

- **Analisi Diagnostica**: confrontare i dati, scoprire correlazioni e capire l'impatto di un fattore sull'altro (regressione) in modo da identificare le cause reali dei fenomeni osservati.

- **Analisi Predittiva**: prevedere risultati (clustering/trend), basandosi sull'uso di algoritmi che imparano dagli schemi presenti nei dati passati per proiettarli nel futuro.

1. Trend Recensioni (Time Series)

Obiettivo: Analizzare il trend temporale dei punteggi degli hotel, utilizzando la **Regressione Lineare Score vs Tempo**, per identificare il **trend slope** che indica quali alberghi stanno migliorando o peggiorando nel tempo.

Casi d'uso: Identificare "**stelle nascenti**" o **hotel decadenti** nonostante un alto punteggio medio storico.

Hotel_Name	The Curtain
Trend_Slope	0.00643700733781
Review_Count	30
Average_Score_Calculated	8.88333320618
Average_Score	9.10000038147
First_Review_Date	May 25, 2017
Last_Review_Date	Aug 03, 2017

Top 10 Hotel in Crescita

	Hotel_Name	Trend_Slope	Review_Count	Average_Score_Calculated	Average_Score	First_Review_Date	Last_Review_Date
1076	The Curtain	0.0064	30	8.8833	9.1	2017-05-25	2017-08-03
931	Chasse Hotel	0.0059	139	9.0144	8.9	2017-03-27	2017-08-02
125	Hotel Park Lane Paris	0.0041	154	8.6338	8.7	2016-05-23	2017-08-03
1096	Villa Lut ce Port Royal	0.004	47	6.3851	7	2015-08-15	2017-05-02
853	NYX Milan	0.0035	180	8.5956	8.8	2017-02-25	2017-08-03
552	Hotel Capitol Milano	0.0031	66	8.1242	8.3	2015-09-22	2017-07-05
1298	Lansbury Heritage Hotel	0.0031	40	9.5175	9.4	2017-04-25	2017-08-02
1126	Best Western Le 18 Paris	0.0031	86	7.4291	7.6	2016-03-22	2017-07-20
265	Boti Hotel by Elegancia	0.003	49	8.9388	9	2017-03-12	2017-08-02
90	Hilton London Euston	0.0025	470	7.1209	7.4	2015-08-04	2017-08-02

Top 10 Hotel in Calo

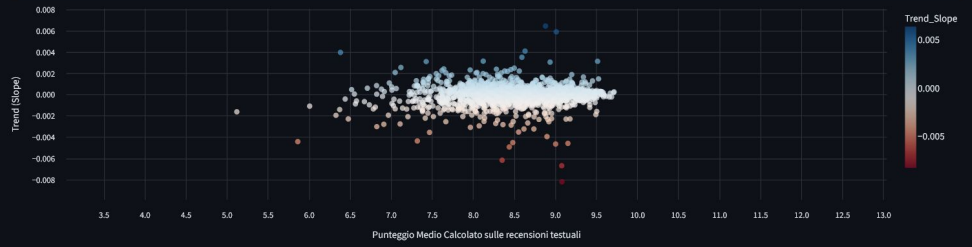
	Hotel_Name	Trend_Slope	Review_Count	Average_Score_Calculated	Average_Score	First_Review_Date	Last_Review_Date
632	Okko Hotels Paris Porte De Versailles	-0.0082	43	9.0837	9.2	2017-06-09	2017-07-31
1019	Le Tsuba Hotel	-0.0067	57	9.0807	9.3	2017-03-25	2017-08-02
380	Marlin Waterloo	-0.0062	84	8.3548	8.6	2017-05-29	2017-08-03
917	Arthotel ANA Westbahn	-0.0049	67	8.4418	8.3	2017-03-02	2017-08-03
1025	Maison Albar Hotel Paris C line	-0.0047	62	9.0065	8.9	2016-12-25	2017-08-03
1297	La Villa Haussmann	-0.0046	41	9.1561	9.2	2017-01-10	2017-08-02
1314	Majestic Hotel Spa	-0.0045	38	8.4842	8.7	2015-08-18	2017-07-29
904	Villa Eugenie	-0.0044	62	5.8645	6.8	2015-08-05	2017-08-02
830	London Suites	-0.0044	114	7.3211	7.4	2016-07-03	2017-08-02
183	Park Plaza London Park Royal	-0.004	197	8.901	8.9	2017-03-10	2017-08-03

Distribuzione Trend vs Punteggio Medio

In questo grafico è possibile osservare la distribuzione dei trend in relazione al punteggio medio degli hotel.

- **^ In alto** si trovano gli hotel con trend positivo (in crescita)
- **v In basso** gli hotel con trend negativo (in calo).
- **→ A destra** si trovano gli hotel con punteggio medio alto
- **← A sinistra** gli hotel con punteggio medio basso.
- **Gli hotel con punteggio medio alto e trend positivo sono i migliori hotel.**

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative all'hotel.




```
model = LinearRegression() # da sklearn.linear_model
model.fit(X, y)            # Calcola il modello lineare:
                           # trova la linea retta (y = mx+q) che meglio si adatta ai dati (best fit line)
                           # dove m è il coefficiente angolare e q è l'intercetta
slope = model.coef_[0]  # In questo caso ci interessa solo il valore di m (slope), che è il nostro trend temporale
# Se slope > 0: Trend Crescente
# Se slope < 0: Trend Decrescente
```

2. Analisi Influenza Tag

Obiettivo: Determinare quali fattori (es. ”Single Room”, ”No Window”) impattano positivamente o negativamente sul punteggio che i recensori assegnano agli hotel.

Weighted Impact: indice euristico che valuta l’impatto di un tag, pesandolo per la sua “affidabilità”, cioè premiando tag i cui voti hanno una **deviazione standard bassa** e una **frequenza alta**.

```
Impact = Average Score - Global Average
Reliability Index = (1 / (StdDev + 0.1)) * log(Count)
Weighted Impact = Impact * Reliability Index
```

```
# Il campo Tags è una stringa tipo "[' Leisure trip ', ' Couple ', ...]".
# Trasformazione LAZY: restituisce "Leisure trip, Couple, ..."
clean_tags = F.regexp_replace(F.col("Tags"), "[\\[\\]']", "")
# Trasformazione LAZY: restituisce ["Leisure trip", "Couple", ...]
splitted_tags = F.split(clean_tags, ",")
# explode(splitted_tags), crea una nuova riga per ogni elemento dell'array splitted_tags
# Esempio: se una riga ha Tags = "[Leisure Trip, Couple]",
# explode crea due righe: una con Single_Tag = 'Leisure Trip' e una con Single_Tag = 'Couple'
# (gli altri campi vengono duplicati dalla riga originale)
exploded_df = df.withColumn("Single_Tag", F.explode(splitted_tags))
```

Single_Tag	Standard Double Room without Window
Average_Score	7.03028052805
Count	2424
Impact	−1.36479604183
Reliability_Index	3.63222974858
Weighted_Impact	−4.95725278388
StdDev_Score	2.0455620615

Media Globale Voti (calcolata su tutte le recensioni):

8.40

👍 Top 10 Tag Positivi (per Weighted Impact)

Tag più affidabili che alzano il voto.

	Single_Tag	Average_Score	Count	Impact	Reliability_Index	Weighted_Impact
0	Camper Room	9.73	225	1.33	9.04	12.04
1	Kings Junior Suite	9.65	55	1.26	6.95	8.76
2	Luxury Double or Twin Room	9.68	68	1.28	6.78	8.67
3	Double or Twin Room Allergy Free	9.69	50	1.30	5.97	7.74
4	Double Room XL	9.67	58	1.28	5.49	7.00
5	Deluxe Double Room 1 2 Adults	9.53	305	1.13	5.73	6.48
6	Small Double Room Annex building	9.56	72	1.17	5.14	6.01
7	Double Room Annex	9.43	187	1.03	5.68	5.88
8	Delightful Queen Room	9.56	64	1.17	5.03	5.87
9	City King Room	9.30	739	0.91	6.47	5.86

👎 Top 10 Tag Negativi (per Weighted Impact)

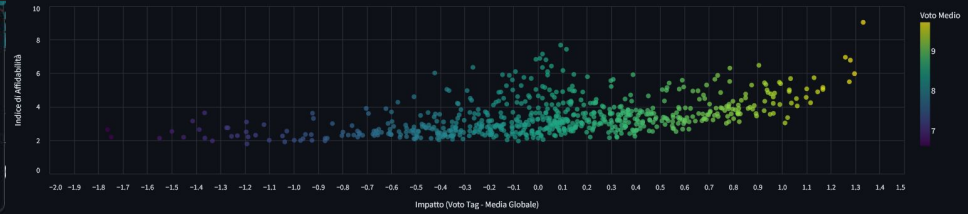
Tag più affidabili che abbassano il voto.

	Single_Tag	Average_Score	Count	Impact	Reliability_Index	Weighted_Impact
660	Standard Double Room without Window	7.03	2424	-1.36	3.63	-4.96
659	Cabin Single Room	6.63	195	-1.76	2.64	-4.65
658	Standard Double Room No Window	6.98	470	-1.41	3.14	-4.44
657	Eiffel Tower View King Room	6.65	109	-1.75	2.18	-3.82
657	Eiffel Tower View King Room	6.65	109	-1.75	2.18	-3.82
656	Single Guest Room	6.90	109	-1.50	2.53	-3.79
655	King Hilton Guest Room	7.20	646	-1.19	3.09	-3.68
654	Superior Suite with 1 Double Bed and 1 Single Bed	7.01	181	-1.39	2.64	-3.65
653	Small Single Room	7.12	312	-1.27	2.74	-3.50
652	Standard Double Room with View Terrace	7.13	142	-1.27	2.74	-3.48
651	Standard Room with 1 Double Bed	7.47	879	-0.92	3.61	-3.34

Grafico: Affidabilità vs Impatto

- **Asse X (Impact)**: Quanto il tag sposta il voto (Destra=Positivo, Sinistra=Negativo).
- **Asse Y (Reliability)**: Quanto è "solido" il dato (Alto=Molto affidabile, Basso=Incerto).
- **Obiettivo:** Cerca i tag negli angoli in alto a destra (vincenti sicuri) e in alto a sinistra (problemi certi).

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative al tag.



3. Analisi Bias Nazionalità

Obiettivo: Individuare, se esistono, le nazionalità che tendono a dare voti più alti o bassi rispetto alla media (potrebbero esistere **bias culturali** che influenzano l'esperienza dei clienti).

Score Deviation = deviazione dal voto globale medio

Sentiment Ratio = rapporto fra numero di parole positive e negative utilizzate nelle recensioni. Indica la **coerenza** dei recensori: chi è mediamente più generoso dovrebbe utilizzare più parole positive che negative, e viceversa.

Grafico **Correlazione Voto vs Positività Testo:** consente di valutare visivamente la coerenza dei recensori per ogni nazionalità.

```
global_avg = df.select(F.avg("Reviewer_Score")).first()[0] # Prendiamo la media globale per confronto
final_stats = nation_stats.filter(F.col("Count") >= min_reviews) # Filtro per rilevanza statistica (num minimo di recensioni)
# Calcolo metriche derivate (Deviazione = scostamento dalla media globale, ratio = rapporto tra positive e negative)
final_stats = final_stats.withColumn("Global_Average", F.lit(global_avg)) \
    .withColumn("Score_Deviation", F.col("Average_Score") - F.col("Global_Average")) \
    .withColumn("Total_Words_Avg", F.col("Avg_Positive_Words") + F.col("Avg_Negative_Words")) \
    .withColumn("Sentiment_Ratio",
                F.when(F.col("Total_Words_Avg") > 0, # evita divisione per zero
                       F.col("Avg_Positive_Words") / F.col("Total_Words_Avg"))
                .otherwise(0.0)
    )
```

Media Globale Voti

8.40

😤 I Più Critici (Voti Bassi)

	Reviewer_Nationality	Average_Score	Count	Score_Deviation	Sentiment_Ratio	Total_Words_Avg
140	Mongolia	7.25	39	-1.14	0.42	20.72
139	Syria	7.52	33	-0.87	0.24	36.85
138	Libya	7.60	72	-0.80	0.43	23.06
137	Seychelles	7.60	21	-0.80	0.43	30.14
136	Bangladesh	7.61	151	-0.79	0.46	26.85
135	Iran	7.73	1086	-0.67	0.47	26.62
134	Algeria	7.78	100	-0.62	0.44	21.55
133	Tanzania	7.81	58	-0.59	0.42	26.14
132	Jordan	7.81	757	-0.59	0.43	27.30
131	Pakistan	7.81	916	-0.59	0.49	27.81

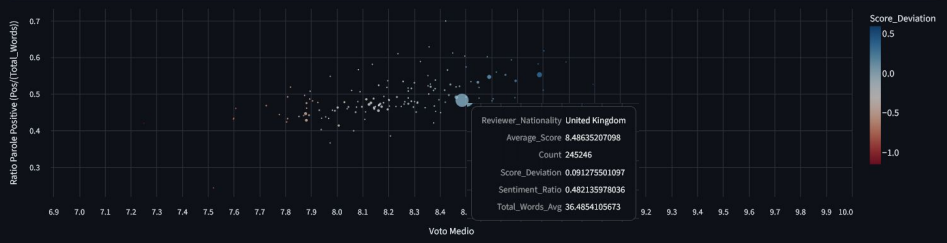
😍 I Più Generosi (Voti Alti)

	Reviewer_Nationality	Average_Score	Count	Score_Deviation	Sentiment_Ratio	Total_Words_Avg
0	Kyrgyzstan	9.00	21	0.60	0.53	37.05
1	Liechtenstein	8.89	20	0.49	0.59	31.60
2	Puerto Rico	8.80	180	0.41	0.62	29.08
3	Panama	8.80	122	0.41	0.51	25.35
4	United States of America	8.79	35437	0.39	0.55	42.02
5	Honduras	8.78	22	0.39	0.58	29.77
6	Israel	8.69	6610	0.30	0.54	33.10
7	El Salvador	8.69	24	0.30	0.49	23.08
8	Trinidad and Tobago	8.68	154	0.28	0.59	38.40
9	New Zealand	8.65	3237	0.26	0.55	35.05

Correlazione Voto vs Positività Testo (Corenza delle recensioni per nazionalità)

Chi dà voti alti scrive davvero cose positive? (In alto a destra = Sì, In basso a destra = No)

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative alla nazionalità.



4. Analisi Competitività Locale

Obiettivo: Individuare gli hotel che hanno le performance migliori rispetto ai loro concorrenti geograficamente più vicini (l'utente sceglie la dimensione dell'area di vicinato).

Score Delta = differenza fra il punteggio di un hotel e il punteggio medio del suo vicinato

Distanza fra hotel calcolata con Formula di Haversine, a partire dalle coordinate geografiche.

```
# 2. Self-Join, calcolo delle distanze, filter per distanza < radius
# Rinominiamo per distinguere Hotel A (Target) e Hotel B (Neighbor)
left = hotels.alias("a")
right = hotels.alias("b")
joined = left.crossJoin(right).filter(F.col("a.Hotel_Name") != F.col("b.Hotel_Name"))

# Formula Haversine: serve a calcolare la distanza in linea d'aria tra due punti su una sfera (la Terra)
#   1. Le coordinate lat e lng sono in gradi, ma la trigonometria funziona in radianti, quindi vanno convertite con F.radians(...)
#   2. R = 6371 km (Raggio Terra)
#   3. dLat = rad(lat2 - lat1)                                  Differenza tra latitudini in radianti
#   4. dLon = rad(lon2 - lon1)                                  Differenza tra longitudini in radianti
#   5. distance = sin^2(dLat/2) + cos(lat1) * cos(lat2) * sin^2(dLon/2)   Distanza fra due punti su una sfera (adimensionale tra 0 e 1)
#   6. angle = 2 * asin(sqrt(distance))                         Distanza angolare in radianti
#   7. distance_km = R * angle                                  Distanza in km
joined = joined.withColumn("lat_a_rad", F.radians(F.col("a.lat"))) \
               .withColumn("lon_a_rad", F.radians(F.col("a.lng"))) \
               .withColumn("lat_b_rad", F.radians(F.col("b.lat"))) \
               .withColumn("lon_b_rad", F.radians(F.col("b.lng"))) \
               .withColumn("dlat", F.col("lat_b_rad") - F.col("lat_a_rad")) \
               .withColumn("dlon", F.col("lon_b_rad") - F.col("lon_a_rad")) \
               .withColumn("distance", F.pow(F.sin(F.col("dlat") / 2), 2) + \
                           F.cos(F.col("lat_a_rad")) * F.cos(F.col("lat_b_rad")) * \
                           F.pow(F.sin(F.col("dlon") / 2), 2)) \
               .withColumn("angle", 2 * F.asin(F.sqrt(F.col("distance")))) \
               .withColumn("distance_km", F.lit(6371.0) * F.col("angle"))
# Filtriamo per distanza < radius
neighbors = joined.filter(F.col("distance_km") <= km_radius)
```

💎 Top 10 Gemme Locali (Meglio dei competitor vicini)

	Hotel_Name	Average_Score	Neighborhood_Avg_Score	Score_Delta	Competitor_Count
0	Milestone Hotel Kensington	9.50	8.23	1.27	100
1	Ritz Paris	9.80	8.59	1.21	230
2	Nu Hotel	8.90	7.82	1.08	5
3	Hotel The Peninsula Paris	9.50	8.44	1.06	129
4	Waldorf Astoria Amsterdam	9.50	8.48	1.02	65
5	Haymarket Hotel	9.60	8.59	1.01	134
6	Pillows Anna van den Vondel Amsterdam	9.40	8.39	1.01	47
7	Hotel Eiffel Blomet	9.40	8.39	1.01	46
8	Charlotte Street Hotel	9.50	8.49	1.01	136
9	H tel de La Tamise Esprit de France	9.60	8.61	0.99	228

⚠️ Top 10 Sotto la Media (Peggio dei competitor vicini)

	Hotel_Name	Average_Score	Neighborhood_Avg_Score	Score_Delta	Competitor_Count
1376	Hotel Liberty	5.20	8.42	-3.22	41
1375	Hotel Cavendish	6.40	8.53	-2.13	120
1374	The Tophams Hotel	6.60	8.71	-2.11	88
1373	Savoy Hotel Amsterdam	6.40	8.35	-1.95	41
1372	Britannia International Hotel Canary Wharf	7.10	8.96	-1.86	8
1371	Bloomsbury Palace Hotel	6.80	8.53	-1.73	119
1370	Villa Eugenie	6.80	8.47	-1.67	105
1369	Gran Hotel Barcino	7.00	8.67	-1.67	119
1368	Park Lane Mews Hotel	7.00	8.61	-1.61	137
1367	Best Western Maitrise Hotel Edgware Road	6.60	8.18	-1.58	78

Grafico: Performance Relativa

Confronto tra il voto dell'hotel (Asse X) e il voto medio della zona (Asse Y).

- **Sotto la diagonale**: L'hotel è meglio della zona (Gemma Locale)
- **Sopra la diagonale**: L'hotel è peggio della zona (Underperformer)

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative all'hotel.

Hotel_Name Hotel Cavendish
Average_Score 6.40000009537
Neighborhood_Avg_Score 8.52916670243
Competitor_Count 120
Score_Delta −2.12916660706

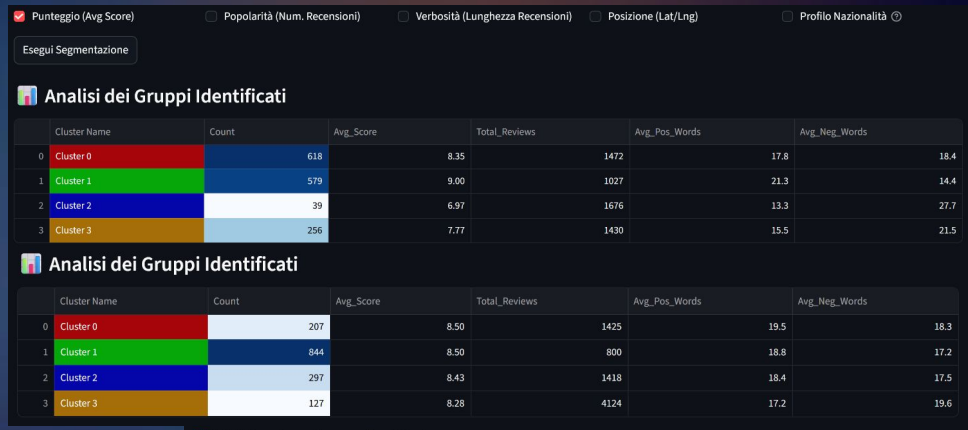
5. Segmentazione Hotel (K-Means Clustering)

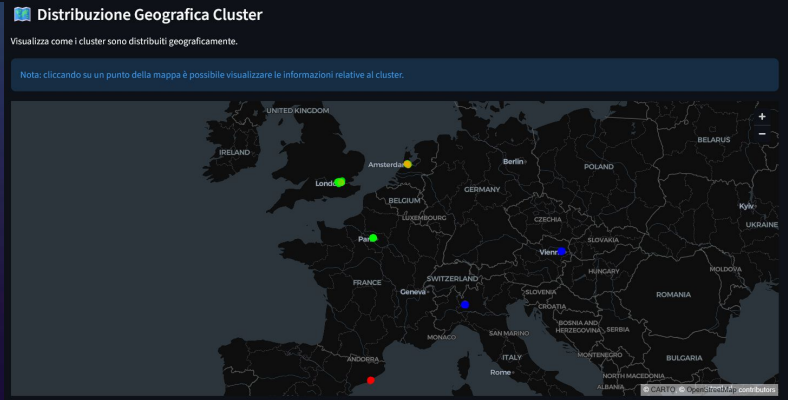
Obiettivo: Raggruppare gli hotel in cluster omogenei (utilizzando K-Means Clustering), sulla base delle seguenti feature: Punteggio Medio, Popolarità (numero recensioni), Verbosità delle recensioni, Posizione geografica, Bias di nazionalità.

```
# 3. Assemblaggio Feature Vector
input_cols = []
if use_score: input_cols.append("Avg_Score")
if use_popularity: input_cols.append("Total_Reviews")
if use_verbosity:
    input_cols.append("Avg_Pos_Words")
    input_cols.append("Avg_Neg_Words")
if use_location:
    # 3D Coordinate Transformation (Unit Sphere)
    # Convertiamo Lat/Lng (gradi) in Radianti per usare funzioni trigonometriche
    # x = cos(lat) * cos(lng)
    # y = cos(lat) * sin(lng)
    # z = sin(lat)
    hotel_features = hotel_features.withColumn("lat_rad", F.radians(F.col("Lat"))) \
                                   .withColumn("lng_rad", F.radians(F.col("Lng"))) \
                                   .withColumn("x", F.cos(F.col("lat_rad")) * F.cos(F.col("lng_rad"))) \
                                   .withColumn("y", F.cos(F.col("lat_rad")) * F.sin(F.col("lng_rad"))) \
                                   .withColumn("z", F.sin(F.col("lat_rad")))
    input_cols.append("x")
    input_cols.append("y")
    input_cols.append("z")
if use_nationality:
    # Aggiungiamo le colonne delle top nazioni generate dal pivot
    # Nota: dobbiamo recuperare i nomi delle colonne generate (che sono i nomi delle nazioni)
    # Le colonne attuali del df meno quelle base conosciute sono quelle delle nazioni
    base_cols = ["Hotel_Name", "Avg_Score", "Total_Reviews", "Avg_Pos_Words", "Avg_Neg_Words", "Lat", "Lng"]
    nat_cols = [c for c in hotel_features.columns if c not in base_cols]
    input_cols.extend(nat_cols)
# Vector Assembler: unisce le colonne in un unico vettore "features_raw"
assembler = VectorAssembler(inputCols=input_cols, outputCol="features_raw")

# Standard Scaler: normalizza le feature (media 0, dev.std 1)
# Fondamentale per K-Means perchè le distanze euclidee sono sensibili alla scala (es. Reviews=1000 vs Score=10)
scaler = StandardScaler(inputCol="features_raw", outputCol="features", withStd=True, withMean=True)

# K-Means (# seed=1 per replicabilità)
kmeans = KMeans(featuresCol="features", k=k, seed=1)

# Pipeline (serve per concatenare le operazioni)
pipeline = Pipeline(stages=[assembler, scaler, kmeans])

# Fit & Transform
model = pipeline.fit(hotel_features) # Fit: apprendimento
predictions = model.transform(hotel_features) # Transform: predizione
```

Es. 1: Clustering solo per punteggio

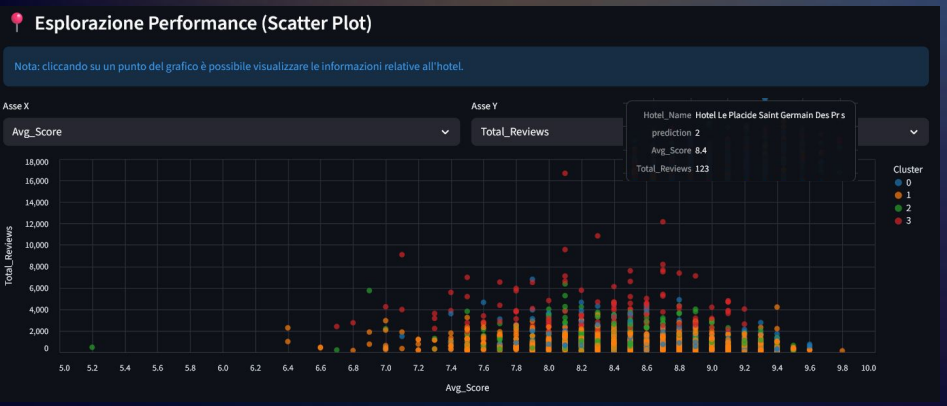
Es. 2: Clustering con tutte le feature





5. Segmentazione Hotel (K-Means Clustering)

Grafico interattivo per l'esplorazione delle performance dei cluster, con possibilità di scegliere le caratteristiche da confrontare.

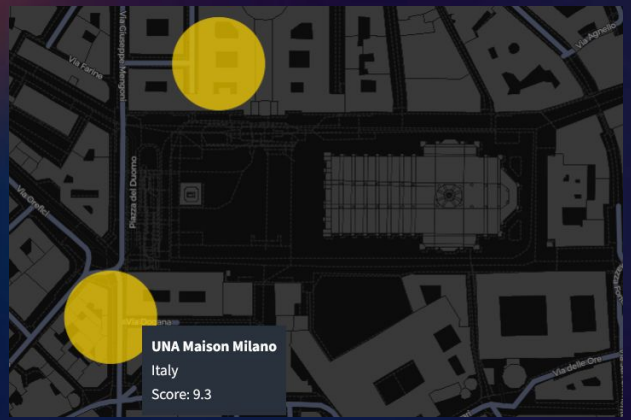


6. Migliori Hotel per Nazione

Obiettivo: Identificare le eccellenze alberghiere suddivise per paese.

Ordinamento alfabetico predefinito per nazione, modificabile dall'utente.

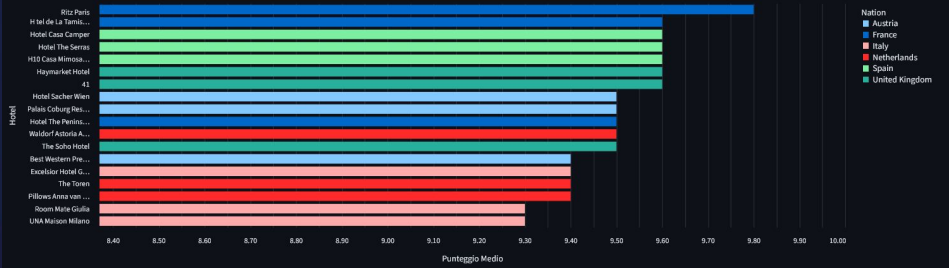
Mappa geografica dettagliata con posizioni estrapolate dalle coordinate degli hotel.



Top 3 Hotel per Nazione

	Nation	Hotel_Name	Average_Score	Total_Number_of_Reviews	Hotel_Address	lat	lng
0	Austria	Hotel Sacher Wien	9.5	632	Philharmoniker Stra e 4 01 Innere Stadt 1010 Vienna Austria	48.2039	16.3698
1	Austria	Palais Coburg Residenz	9.5	98	Coburgbastei 4 01 Innere Stadt 1010 Vienna Austria	48.2059	16.3767
2	Austria	Best Western Premier Kaiserhof Wien	9.4	1353	Frankenberggasse 10 04 Wieden 1040 Vienna Austria	48.1975	16.3686
3	France	Ritz Paris	9.8	122	15 Place Vend me 1st arr 75001 Paris France	48.8679	2.329
4	France	H tel de La Tamise Esprit de France	9.6	166	4 rue d Alger 1st arr 75001 Paris France	48.8649	2.3296
5	France	Hotel The Peninsula Paris	9.5	275	19 avenue Kleber 16th arr 75116 Paris France	48.8711	2.2934
6	Italy	Excelsior Hotel Gallia Luxury Collection Hotel	9.4	1345	Piazza Duca D Aosta 9 Central Station 20124 Milan Italy	45.4857	9.202
7	Italy	Room Mate Giulia	9.3	2011	Silvio Pellico 4 Milan City Center 20121 Milan Italy	45.4651	9.1895
8	Italy	UNA Maison Milano	9.3	1108	Via Mazzini 4 Milan City Center 20123 Milan Italy	45.4633	9.1884
9	Netherlands	Waldorf Astoria Amsterdam	9.5	443	Herengracht 542 556 Amsterdam City Center 1017 CG Amsterdam Netherlands	52.3648	4.8969

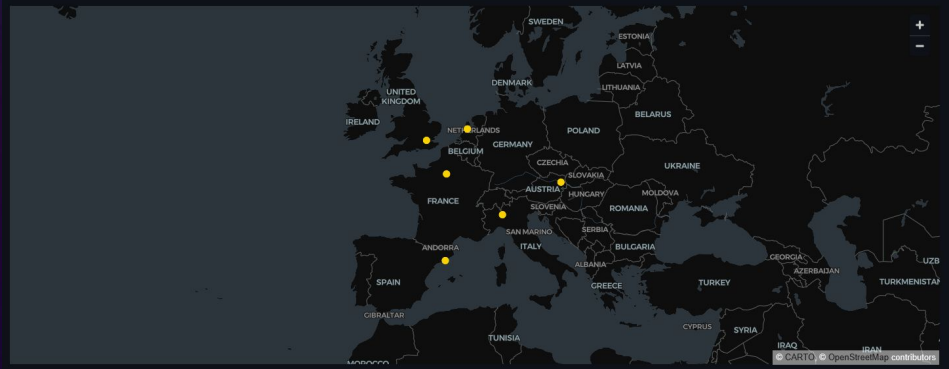
Distribuzione dei Punteggi degli Hotel



Mappa dei Migliori Hotel

Nota: cliccando su un punto della mappa è possibile visualizzare le informazioni relative all'hotel.

Visualizzazione geografica di 18 hotel (quelli con coordinate valide).



7. Locals vs Tourists

Obiettivo: Analizzare la differenza di percezione tra chi visita il proprio paese (Local) e chi viene dall'estero (Tourist).

Identificazione di "**Trappole per turisti**" o "**Preferiti dai locali**" (hotel che potrebbero offrire un'esperienza più autentica).

```python
# 3. Classificazione Review (Local vs Tourist)
# Local = Recensore della stessa nazione dell'hotel
df_tagged = df_prep.withColumn(
    "Review_Type",
    F.when(F.col("Hotel_Nation") == F.col("Reviewer_Nationality_Clean"), "Local")
     .otherwise("Tourist")
)

# 4. Aggregazione (Pivot)
# Raggruppa per Hotel e Nazione, poi fa Pivot su Review_Type
# pivot crea una colonna per ogni Review_Type, cioè una colonna per Local e una per Tourist
# a quel punto vengono calcolati i valori medi e conteggi per ogni colonna (mantenendo la divisione in gruppi dello stesso hotel)
# La tabella risultante avrà colonne:
# Hotel_Name, Hotel_Nation, Local_Avg_Score, Local_Count, Tourist_Avg_Score, Tourist_Count
stats = df_tagged.groupBy("Hotel_Name", "Hotel_Nation").pivot("Review_Type", ["Local", "Tourist"]).agg(
    F.avg("Reviewer_Score").alias("Avg_Score"),
    F.count("Reviewer_Score").alias("Count")
)
```

Grafico: Discrepanza Voti

Confronto diretto tra Voto Local (Y) e Voto Tourist (X).

- **Punti Sopra la diagonale**: Meglio per i Locali.
- **Punti Sotto la diagonale**: Meglio per i Turisti.

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative all'hotel.

Discrepanza Media (Local - Tourist)

+0.19

Hotel preferiti dai Locali (Rispetto ai Turisti)

Hotel dove il voto dei locali supera di più quello dei turisti.

	Hotel_Name	Hotel_Nation	Local_Avg_Score	Tourist_Avg_Score	Preference_Delta	Top_Nationalities
0	Simply Rooms Suites	United Kingdom	7.91	6.67	1.24	Spain (2%), United Kingdom (78%)
1	London Suites	United Kingdom	7.62	6.74	0.89	Italy (2%), France (2%), Austria (3%), United Kingdom (65%), Australia (4%)
2	Crowne Plaza London Docklands	United Kingdom	8.88	8.04	0.85	United Kingdom (89%), Ireland (2%)
3	Sheraton Grand London Park Lane	United Kingdom	8.44	7.61	0.83	Turkey (2%), United Arab Emirates (6%), Kuwait (5%), Saudi Arabia (7%), Australia (5%), United
4	DoubleTree by Hilton London West End	United Kingdom	8.43	7.63	0.80	United Kingdom (80%)
5	The Westbury A Luxury Collection Hotel Mayfair London	United Kingdom	8.49	7.72	0.77	United Arab Emirates (2%), United States of America (2%), South Africa (2%), Kuwait (4%), Saud
6	DoubleTree by Hilton Hotel London Marble Arch	United Kingdom	8.43	7.67	0.76	Saudi Arabia (4%), Kuwait (5%), Israel (4%), Ireland (2%), Australia (2%), United States of Americ
7	The Dorchester Dorchester Collection	United Kingdom	9.52	8.77	0.75	Saudi Arabia (4%), Kuwait (4%), United Arab Emirates (5%), United States of America (6%), Unite
8	Holiday Inn London Oxford Circus	United Kingdom	7.70	6.96	0.75	Kuwait (3%), United Arab Emirates (2%), United Kingdom (57%), India (2%), Ireland (2%), Israel
9	Novotel London Paddington	United Kingdom	8.80	8.06	0.73	United States of America (5%), Ireland (2%), United Kingdom (51%), Saudi Arabia (3%), New Zea

Hotel preferiti dai Turisti (Rispetto ai Locali)

Hotel dove il voto dei turisti supera di più quello dei locali.

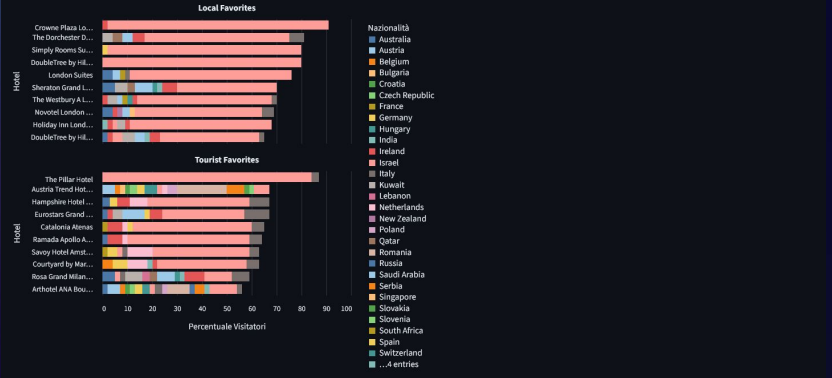
	Hotel_Name	Hotel_Nation	Local_Avg_Score	Tourist_Avg_Score	Preference_Delta	Top_Nationalities
458	Austria Trend Hotel Bosei Wien	Austria	6.76	8.18	-1.41	Poland (4%), Croatia (2%), United Kingdom (6%), Bulgaria (2%), Israel (2%), Belgium (2%), Czech
457	Rosa Grand Milano Starhotels Collezione	Italy	7.39	8.46	-1.07	United Arab Emirates (8%), Israel (2%), United Kingdom (11%), Italy (2%), Kuwait (7%), Qatar (3%
456	Hampshire Hotel The Manor Amsterdam	Netherlands	7.49	8.54	-1.05	Ireland (5%), Australia (2%), Netherlands (7%), United States of America (8%), United Kingdom (4
455	Arthotel ANA Boutique Six	Austria	7.11	8.10	-0.99	Russia (2%), Croatia (2%), Hungary (3%), Israel (2%), Italy (3%), Poland (2%), United Kingdom (11
454	Savoy Hotel Amsterdam	Netherlands	5.33	6.09	-0.76	France (2%), United States of America (4%), Germany (4%), Netherlands (10%), Israel (2%), Unitec
453	The Pillar Hotel	United Kingdom	8.15	8.88	-0.73	United Kingdom (66%), Israel (18%), United States of America (3%)
452	Eurostars Grand Marina Hotel GL	Spain	7.74	8.47	-0.73	Saudi Arabia (9%), United Kingdom (33%), Kuwait (4%), Australia (2%), Ireland (2%), Spain (2%),
451	Catalonia Atenas	Spain	7.43	8.09	-0.66	Ireland (6%), United Kingdom (48%), Spain (2%), United States of America (5%), Netherlands (2%
450	Courtyard by Marriott Amsterdam Arena Atlas	Netherlands	7.95	8.57	-0.62	United Arab Emirates (2%), United States of America (5%), Turkey (2%), Germany (6%), Netherlan
449	Ramada Apollo Amsterdam Centre	Netherlands	7.71	8.31	-0.60	Netherlands (2%), United States of America (5%), Australia (2%), United Kingdom (49%), Ireland (

Distribuzione Nazionalità

Visualizza la composizione percentuale delle nazionalità dei visitatori degli Hotel per ciascuna categoria.

Sono escluse le nazionalità che rappresentano meno del 2% dei visitatoridi un hotel. La non totalità delle percentuali per un hotel può dipendere sia da questa esclusione arbitraria, sia dall'eventuale mancanza di dati sulla nazionalità dei visitatori.

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative ad hotel e distribuzione nazionalità.



8. Analisi Stagionale & Target

Obiettivo: Analizzare come varia il gradimento e individuare gli hotel migliori in base alla stagione e al tipo di viaggio (Leisure/Business/Solo/Family/Group/Couple).

Top Hotel: Summer & Couple, Leisure

I migliori hotel per la combinazione selezionata.

	Hotel_Name	Season	Traveler_Type	Avg_Score	Review_Count	Nation
6	Hotel Le Six	Summer	Couple	9.85	11	France
11	Pillows Anna van den Vondel Amsterdam	Summer	Leisure	9.84	10	Netherlands
41	Hotel Casa Camper	Summer	Couple	9.79	56	Spain
47	Hotel Sacher Wien	Summer	Couple	9.78	22	Austria
87	Covent Garden Hotel	Summer	Leisure	9.74	14	United Kingdom
92	Hotel Casa Camper	Summer	Leisure	9.74	88	Spain
105	Hotel Eiffel Blomet	Summer	Leisure	9.72	12	France
114	Lansbury Heritage Hotel	Summer	Couple	9.72	10	United Kingdom
115	The Beaumont Hotel	Summer	Leisure	9.72	20	United Kingdom
116	Waldorf Astoria Amsterdam	Summer	Leisure	9.72	36	Netherlands

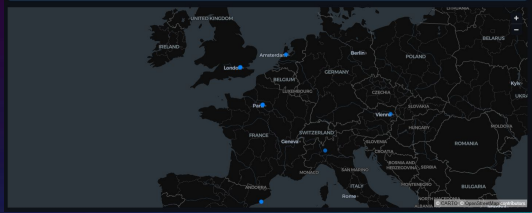
Top Hotel: Winter, Spring, Summer, Autumn & Family, Couple, Business, Solo, Group, Leisure

I migliori hotel per la combinazione selezionata.

	Hotel_Name	Season	Traveler_Type	Avg_Score	Review_Count	Nation
2	Hotel The Peninsula Paris	Autumn	Couple	9.88	13	France
3	Banks Mansion All Inclusive Hotel	Winter	Group	9.87	12	Netherlands
4	Le Narcisse Blanc Spa	Winter	Couple	9.87	18	France
6	Hotel Le Six	Summer	Couple	9.85	11	France
8	Best Western Premier Kaiserhof Wien	Autumn	Family	9.85	11	Austria
9	H tel de La Tamise Esprit de France	Winter	Couple	9.85	13	France
10	Hotel Spadari Al Duomo	Winter	Family	9.84	10	Italy
11	Pillows Anna van den Vondel Amsterdam	Summer	Leisure	9.84	10	Netherlands
12	H10 Casa Mimosa 4 Sup	Winter	Solo	9.84	10	Spain
14	The Wellesley Knightsbridge a Luxury Collection Hotel Lor	Winter	Leisure	9.83	12	United Kingdom

Mappa dei migliori risultati (Winter, Spring, Summer, Autumn - Family, Couple, Business, Solo, Group, Leisure)

Nota: cliccando su un punto della mappa è possibile visualizzare le informazioni relative all'hotel.



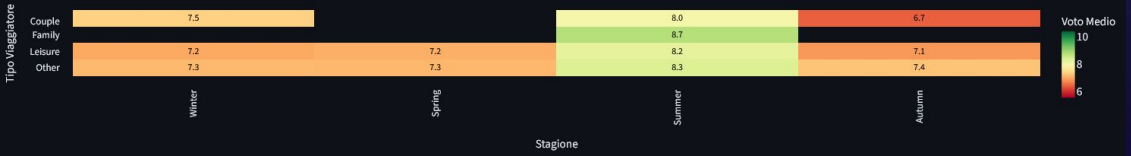
Heatmap Stagionale Hotel

Seleziona un hotel per vedere come performa in tutte le stagioni e con tutti i tipi di viaggiatori.

Scegli Hotel da analizzare

Best Western Premier Kaiserhof Wien

Meli Milano




```
# Creiamo una colonna array con tutti i tipi trovati
# Nota: definiamo una lista di regole (Type, Keyword)
# Se la keyword è presente nei tags, aggiungiamo il Type all'array
types_expr = F.array(
    F.when(F.col("Tags_Lower").contains("business"), "Business").otherwise(None),
    F.when(F.col("Tags_Lower").contains("family"), "Family").otherwise(None),
    F.when(F.col("Tags_Lower").contains("couple"), "Couple").otherwise(None),
    F.when(F.col("Tags_Lower").contains("solo"), "Solo").otherwise(None),
    F.when(F.col("Tags_Lower").contains("group"), "Group").otherwise(None),
    F.when(F.col("Tags_Lower").contains("leisure"), "Leisure").otherwise(None),
    F.lit("Other")
)

# explode(types_expr) -> crea una riga per ogni elemento dell'array (duplicando gli altri campi
# filter rimuove le righe con null
df_exploded = df_enriched.withColumn("Traveler_Type_Raw", F.explode(types_expr)) \
                         .filter(F.col("Traveler_Type_Raw").isNotNull()) \
                         .withColumnRenamed("Traveler_Type_Raw", "Traveler_Type")
# Se una recensione non ha nessun tag riconosciuto, apparirà solo con il tag "Other"
```

9. Analisi Durata Soggiorno

Obiettivo: Analizzare la correlazione tra la durata del soggiorno (numero di notti) e il livello di soddisfazione dei clienti.

Score Delta: indicatore globale delle performance (indica quanto il punteggio medio di ogni categoria di recensioni si discosta dalla media globale)

Performance Globale (Scostamento dalla Media)

Punteggio medio di riferimento (globale, calcolato su tutte le categorie)

8.27

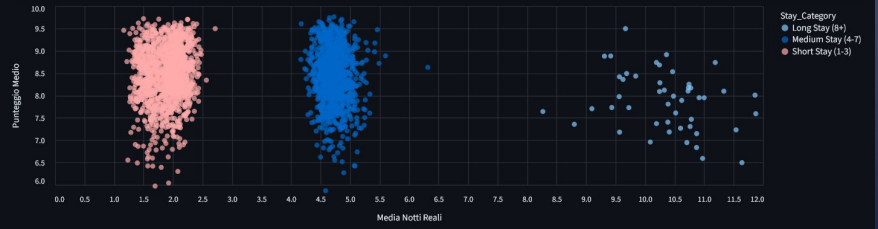
Il delta indica di quanto il punteggio medio di una categoria si discosta dal punteggio medio globale.

Se una categoria ha un delta positivo, significa che i clienti che hanno fatto soggiorni di quella durata sono più soddisfatti rispetto alla media generale. Viceversa, un delta negativo indica una soddisfazione inferiore.

Dettagli Performance Hotel

Ogni punto è un hotel in una specifica categoria di durata.

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative all'hotel.




```
clean_tags = F.regexp_replace(F.col("Tags"), "[\\[\\]']", "") # Pulisce la stringa di tags eliminando parentesi quadre [ ] e apici '
splitted_tags = F.split(clean_tags, ",") # Divide la stringa di tags in singole stringhe ottenendo una colonna di tag singoli
# explode(splitted_tags) -> crea una riga per ogni tag (duplicando gli altri campi dalla riga originale corrispondente nel DataFrame)
exploded = df.withColumn("Single_Tag", F.explode(splitted_tags)) \
             .withColumn("Single_Tag", F.trim(F.col("Single_Tag"))) # trim rimuove spazi vuoti a inizio e fine di ogni tag

# Filtra le righe che contengono tag come "Stayed 1 night", "Stayed 10 nights", etc. usando la Regex: 'Stayed % night%'
# Il % finale serve per catturare sia night che nights
duration_tags = exploded.filter(F.col("Single_Tag").like("Stayed % night%"))

# Estrazione del numero con F.regexp_extract(column_name, pattern, groupIdx)
duration_df = duration_tags.withColumn(
    "Nights",
    F.regexp_extract(F.col("Single_Tag"), r"Stayed (\d+) night", 1).cast("int") # in questo caso accetta sia night che nights
)
```

Top Short Stays

	Hotel_Name	Avg_Score	Review_Count
2	Ritz Paris	9.72	26
3	41	9.71	87
5	Hotel Casa Camper	9.70	205
6	H tel de La Tamise Esprit de France	9.70	45
12	H10 Casa Mimosa 4 Sup	9.63	66

Top Medium Stays

	Hotel_Name	Avg_Score	Review_Count
0	Hotel Casa Camper	9.76	95
1	Hollmann Beletage Design Boutique	9.74	20
4	H10 Casa Mimosa 4 Sup	9.70	50
7	Mercer Hotel Barcelona	9.68	42
8	H tel D Aubusson	9.67	70

Top Long Stays

	Hotel_Name	Avg_Score	Review_Count
59	Staybridge Suites London Stratford	9.50	21
635	Dorsett Shepherds Bush	8.91	36
676	Park Grand London Lancaster Gate	8.88	24
684	St James Court A Taj Hotel London	8.88	58
888	Hyatt Regency London The Churchill	8.74	32

10. Analisi Esperienza Recensore

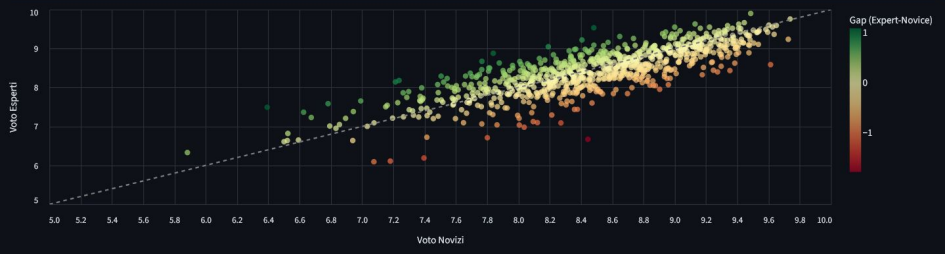
Obiettivo: Valutare se il punteggio delle recensioni degli hotel cambia in base all'esperienza dei recensori.

Experience Gap: differenza fra il voto medio assegnato dagli esperti e il voto medio assegnato dai novizi. Costituisce la metrica di riferimento per valutare quanto in media i recensori esperti sono più critici dei novizi, o viceversa.

Grafico: Novizi vs Esperti

Confronto voti: Novizi (X) vs Esperti (Y).

Nota: cliccando su un punto del grafico è possibile visualizzare le informazioni relative all'hotel.



Gap Medio (Esperti - Novizi): se negativo, gli esperti sono mediamente più severi dei novizi

-0.05

I più criticati dagli Esperti (Gap Negativo)

	Hotel_Name	Novice_Avg_Score	Intermediate_Avg_Score	Expert_Avg_Score	Experience_Gap	Total_Analyzed_Reviews	Novice_Count	Intermediate_Count	Expert_Count
0	Hotel Les Bulles De Paris	8.44	8.24	6.67	-1.78	159	67	80	12
1	Best Western Blue Tower Hotel	7.40	7.30	6.19	-1.21	845	582	236	27
2	Mercure Paris Montmartre Sacr Coeur	8.27	8.46	7.09	-1.18	269	174	85	10
3	Hilton Diagonal Mar Barcelona	7.80	7.70	6.71	-1.09	140	80	50	10
4	Hilton London Euston	7.18	7.05	6.11	-1.07	470	337	122	11
5	Holiday Inn London West	8.04	7.89	6.98	-1.06	701	526	161	14
6	Hotel Sans Souci Wien	9.61	9.53	8.59	-1.03	235	120	98	17
7	Le Dokhan s a Tribute Portfolio Hotel	8.12	8.44	7.10	-1.02	68	37	20	11
8	Ilunion Almirante	7.08	7.06	6.09	-0.98	264	127	121	16
9	Hotel Serhs Rivoli Rambla	8.01	8.09	7.04	-0.97	357	207	133	17

I più apprezzati dagli Esperti (Gap Positivo)

	Hotel_Name	Novice_Avg_Score	Intermediate_Avg_Score	Expert_Avg_Score	Experience_Gap	Total_Analyzed_Reviews	Novice_Count	Intermediate_Count	Expert_Count
838	Hotel Cavendish	6.39	6.47	7.49	1.10	920	627	272	21
837	Banke H tel	8.48	8.20	9.53	1.05	159	86	61	12
836	Hotel Alimara	7.84	7.70	8.88	1.04	195	114	71	10
835	Radisson Blu Champs Elys es Paris	7.24	7.47	8.18	0.94	152	87	55	10
834	Hilton London Green Park	7.21	7.38	8.14	0.93	484	346	126	12
833	Hotel Tiziano Park Vita Parcour Gruppo MiniHotels	7.77	8.17	8.67	0.90	61	27	24	10
832	Sloane Square Hotel	8.19	8.25	9.05	0.86	462	269	170	23
831	Grosvenor House Suites by Jumeirah Living	8.41	8.51	9.22	0.81	100	30	60	10
830	BEST WESTERN Maitrise Hotel Maida Vale	6.78	6.99	7.58	0.80	816	501	284	31
829	FourSide Hotel Vienna City Center	7.51	7.99	8.29	0.79	173	52	97	24

```
# 1. Definizione Categorie Esperienza
# Usiamo il campo Total_Number_of_Reviews_Reviewer_Has_Given
df_exp = df.withColumn(
    "Experience_Level",
    F.when(F.col("Total_Number_of_Reviews_Reviewer_Has_Given") < 5, "Novice")
     .when((F.col("Total_Number_of_Reviews_Reviewer_Has_Given") >= 5) &
           (F.col("Total_Number_of_Reviews_Reviewer_Has_Given") <= 25), "Intermediate")
     .otherwise("Expert")
)
```

Possibili Sviluppi Futuri

- **Analisi del Testo Avanzata:** implementare modelli NLP per Sentiment Analysis sulle recensioni testuali.

- **Streaming**: integrare Spark Streaming per elaborare recensioni in tempo reale.

- **Recommendation System**: sviluppare un sistema di raccomandazione basato sulla similarità utente-utente


Grazie per l’attenzione
REVIEWS
200
717
Rain
Cloud Data
REVIEWS
HOTEL