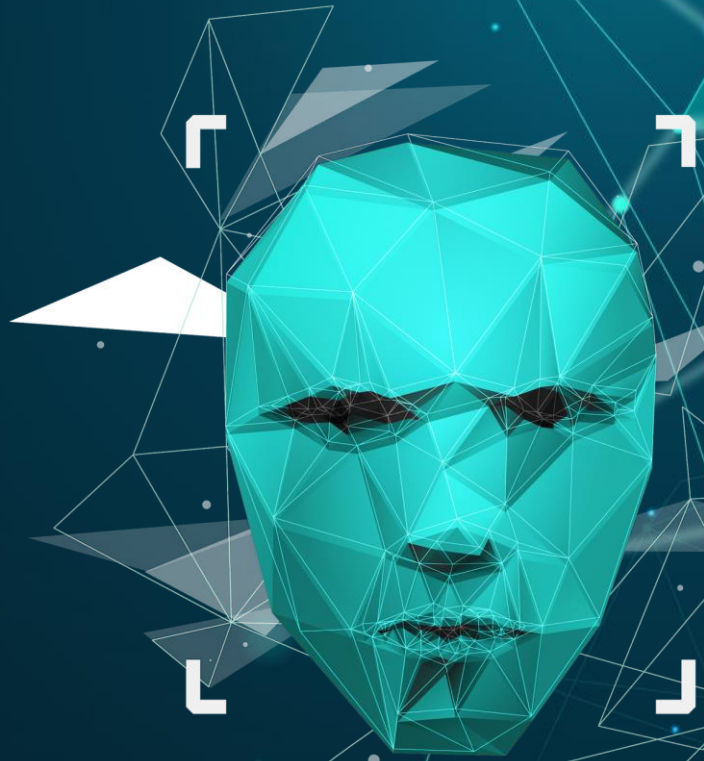
App Advisor



de Cillis Nicolò - 736575
De Tullio Roberta - 737821
Miranda Caterina - 736546

PREPROCESSING E BILANCIAMENTO

01

PREPROCESSING

Al fine di ridurre il numero di esempi e mantenere soltanto le features rilevanti, sono state effettuate le seguenti operazioni di preprocessing:

- Rimosse le colonne: “Installs”, “Minimum Installs”, “Free”, “Currency”, “Developer Email”, “Developer Website”, “Released”, “Privacy Policy”, “Scraped Time” e “Rating Count”
- Preservate solo le app con downloads > 1000 e numero di recensioni > 50
- Utilizzato il **Google Play Scraper** per ottenere le informazioni mancanti
- Raggruppato il **Content Rating** in 3 classi: "Everyone", "Teen", "Adults"
- Raggruppate alcune **categorie** del dataset originale
- Aggiunta la colonna **Success Rate** calcolata in questo modo:

$$success\_rate = \frac{normalized\_rating \ + \ normalized\_downloads}{2}$$

PREPROCESSING

Colonne del dataset prima del pre-processing

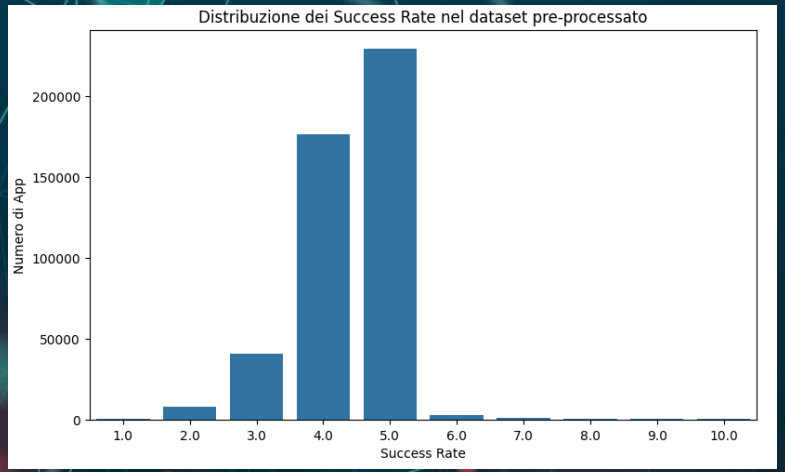
App Name	App Id	Category	Rating	Rating Count	Installs	Minimum Installs	Maximum Installs	Free	Price
Currency	Size	Minimum Android	Developer Id	Developer Website	Developer Email	Released	Last Updated		
Content Rating	Privacy Policy	Ad Supported	In App Purchases	Editors Choice	Scraped Time				

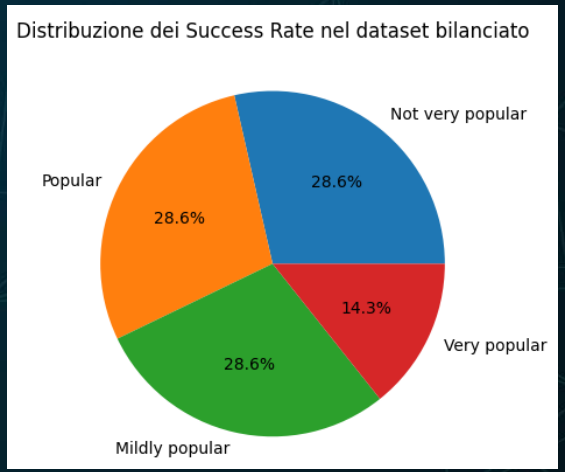
Colonne del dataset dopo il pre-processing

App Name	App Id	Category	Rating	Downloads	Price ($)	Size (MB)	Minimum Android	Developer Id
Last Updated	Content Rating	Ad Supported	In App Purchases	Editors Choice	Success Rate			

BILANCIAMENTO

- **Aggregazione** delle 10 label di **Success Rate** in 4 categorie principali:
  - Success Rate = 1: valori da 0 a 3
  - Success Rate = 2: valore pari a 4
  - Success Rate = 3: valori da 5 a 6
  - Success Rate = 4: valori da 7 a 10
- **Undersampling** ad un massimo di 9000 campioni delle label 1, 2, 3 e **oversampling** ad almeno 4500 campioni della label 4
- **Ridenominazione** delle 4 label in "Not very popular", "Mildly popular", "Popular" e "Very popular"

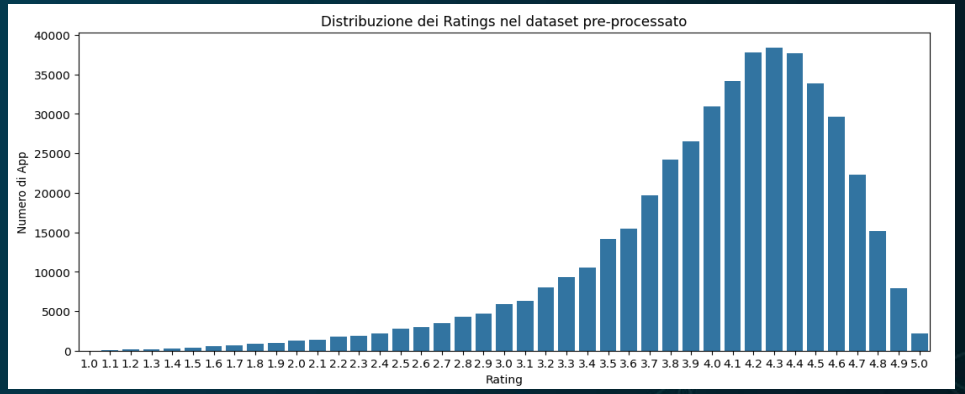


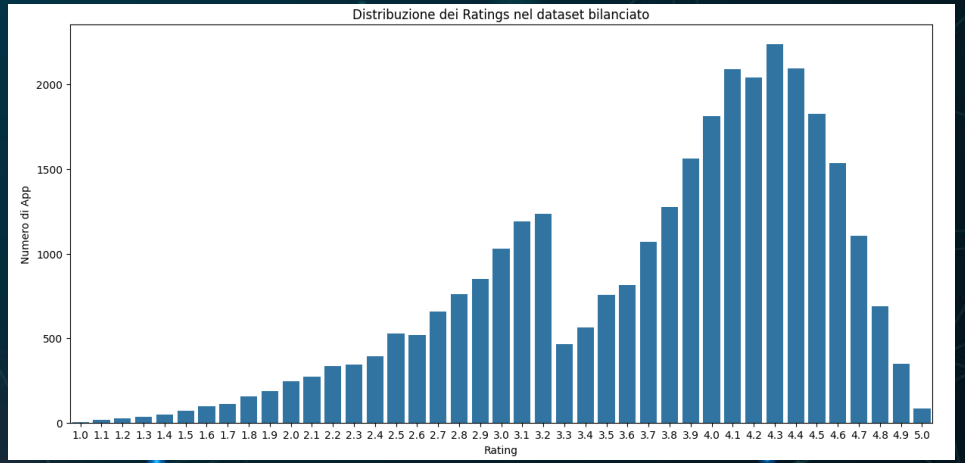


ANALISI DEL DATASET

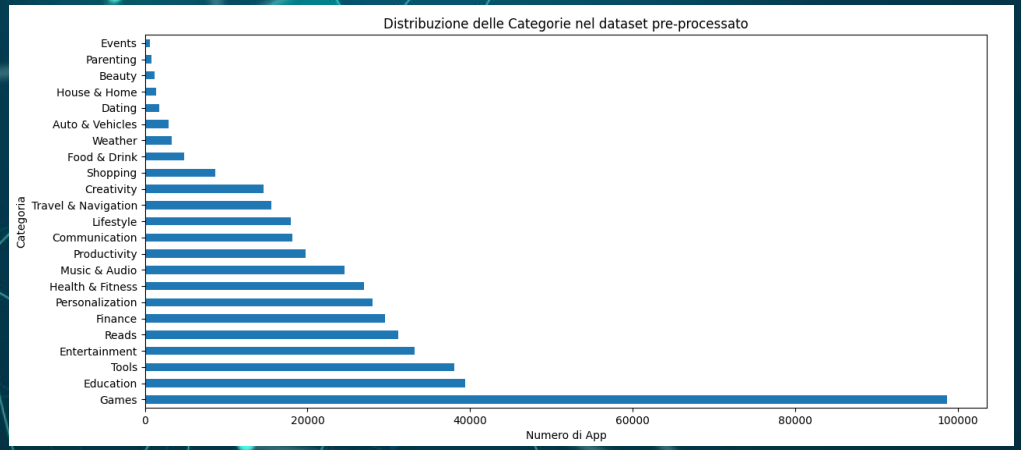
Il **primo grafico** mostra una distribuzione asimmetrica con un forte aumento del numero di app man mano che il numero di stelle aumenta da 1.0, con un picco intorno a 4.1 e 4.2. Dopo questo picco, c'è un calo graduale del numero di app man mano che il numero di stelle si avvicina a 5.0.

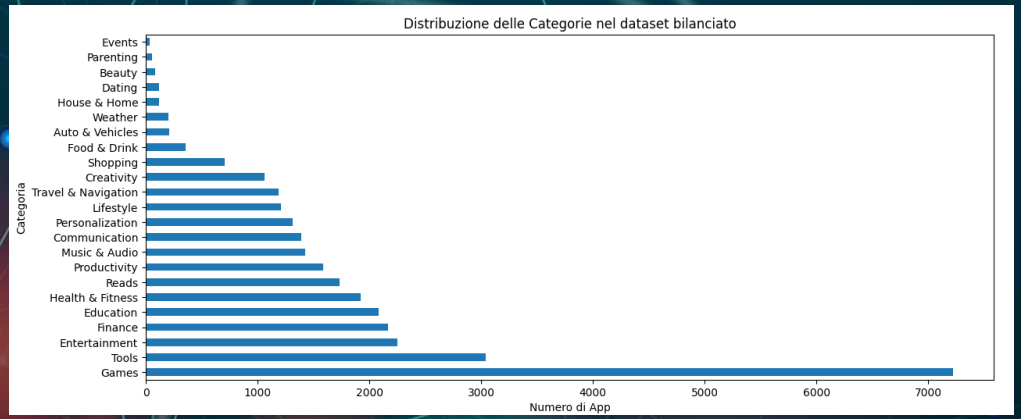
Il **secondo grafico** mostra anch'esso un picco intorno a 4.1 e 4.2; tuttavia, il numero di app con il numero di stelle intorno a 3.0 è relativamente più alto rispetto al primo grafico.





ANALISI DEL DATASET

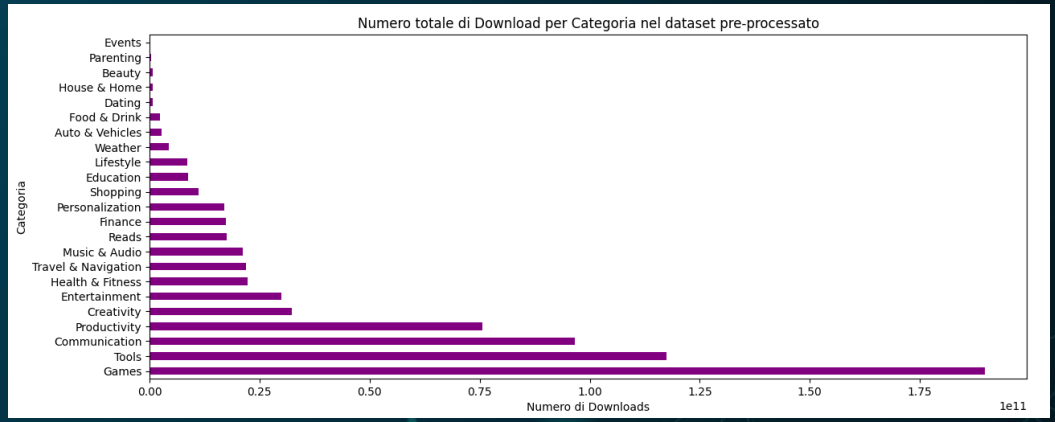


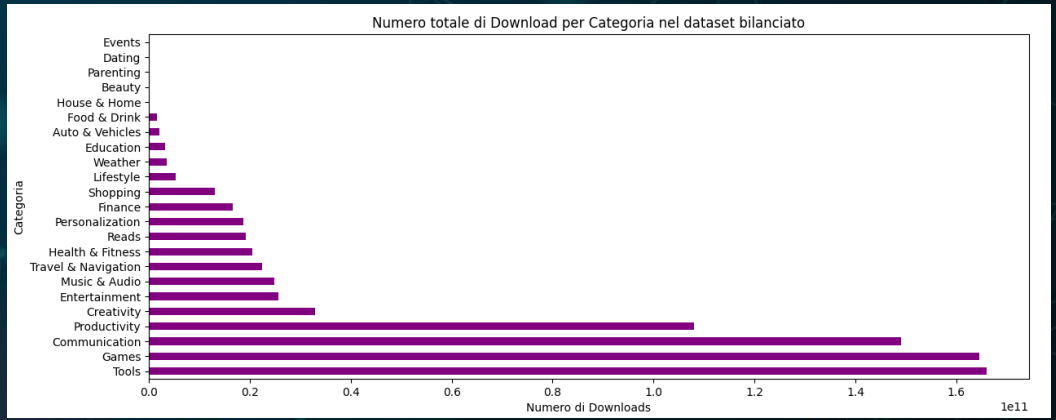


Si osserva come la distribuzione delle categorie è rimasta la stessa in entrambi i casi, con la maggior parte delle app che rientrano nella categoria “Games” mentre “Events” rimane la categoria con meno app.

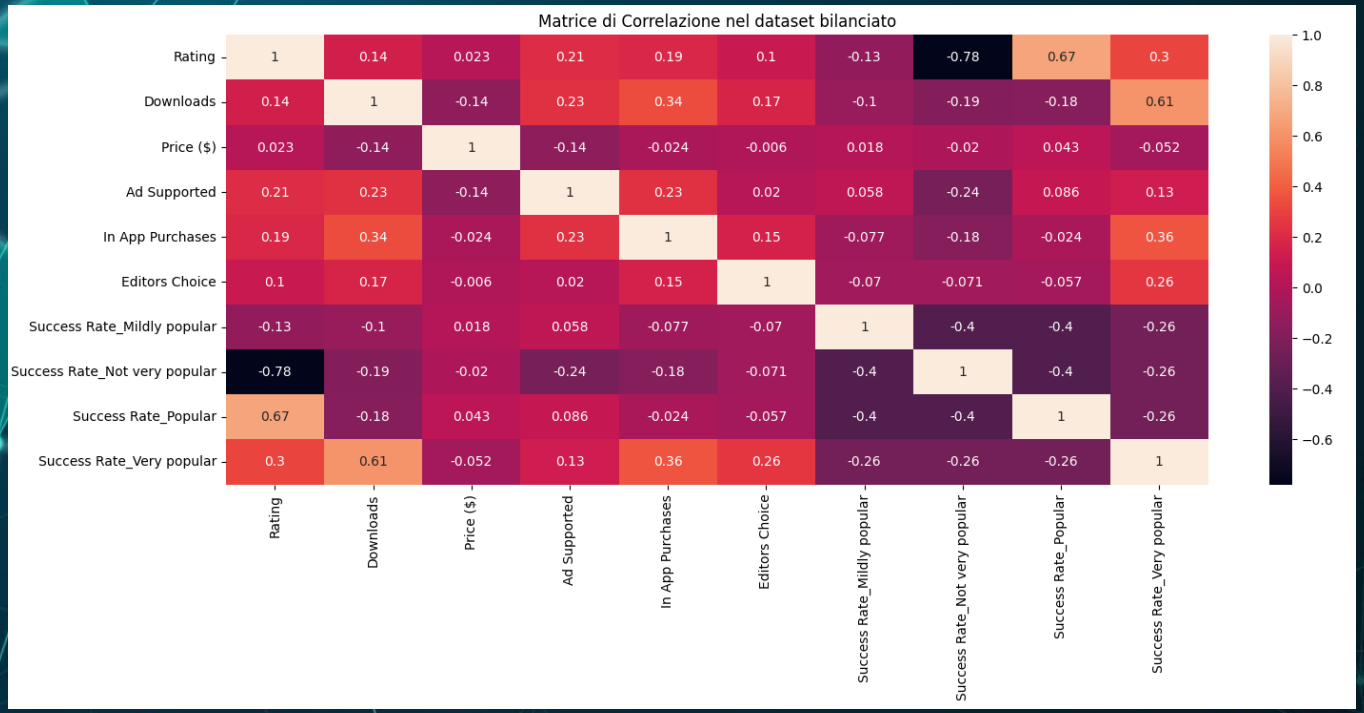
ANALISI DEL DATASET

Il confronto tra i grafici sui **download delle app** e il **numero di app per categoria** rivela che categorie come "Games" e "Tools" sono molto popolari ma saturate, mentre "Education" e "Reads" hanno molte app ma pochi download, indicando bassa popolarità. La categoria "Productivity", con poche app e molti download, sembra promettente per investimenti, mentre categorie come "Events", "Parenting" e "Beauty" presentano pochi download e poche app; può essere utile condurre uno studio per comprendere i motivi della scarsa popolarità.





ANALISI DEL DATASET



Dalla **matrice di correlazione** si osserva che gli elementi che più influenzano il success rate sono “Rating”, “Downloads”, “In App Purchases” e “Editors Choice”; in particolare il success rate è direttamente proporzionale ai primi due.

KNOWLEDGE BASE

02

FATTI E CLASUOLE

A partire dal dataset preprocessato, abbiamo realizzato una **base di conoscenza** scritta in **Prolog** con l'obiettivo di fornire all'utente un'interfaccia per **esplorare** i dati disponibili e delle statistiche relative al dataset attraverso delle query e **inferire** nuova conoscenza per la creazione di un nuovo dataset su cui svolgere l'apprendimento.

Fatti

- `app_name(Id, Name)`
- `app_rating_price(Name, Rating, Price)`
- `app_developer(Name, Developer)`
- `app_developer_downloads(Name, Developer, Downloads)`
- `app_rating_downloads(Name, Rating, Downloads)`
- `app_category(Name, Category)`
- `app_category_price(Name, Category, Price)`
- `app_category_edchoice(Name, Category, Editors_choice)`
- `app_category_edchoice_downloads(Name, Category, Editors_choice, Downloads)`
- `app_category_downloads(Name, Category, Downloads)`
- `app_category_rating(Name, Category, Rating)`
- `app_price_downloads(Name, Price, Downloads)`
- `app_category_developer_success(Name, Category, Developer, Success_rate)`
- `app_success_rating_downloads(Name, Success_rate, Rating, Downloads)`

Clausole

- `count_apps_by_developer(Dev, Count)`
- `top_rating_price(RatingTh, PriceTh, N, TopApps)`
- `top_downloads_by_developer(Dev, N,TopAppsWithDownloads)`
- `top_rating_low_downloads(RatingTh, N, TopApps)`
- `top_apps_by_rating(RatingTh, N, TopApps)`
- `apps_by_category_price(Category, PriceTh, N, TopApps)`
- `count_editors_choice(Category, Count)`
- `top_editors_choice(Category, N, AppList)`
- `top_downloads_by_category(Category, N, AppList)`
- `sum_downloads_by_category(Category, TotalDownloads)`
- `categories_ranked_by_downloads(TotalDownloadsList)`
- `avg_rating_by_category(Category, AvgRating)`
- `avg_downloads_by_category(Category, AvgDownloads)`
- `categories_ranked_by_rating(TotalRatingList)`
- `top_expensive_downloads(N, SortedByDownloads)`
- `top_free_downloads(N, TopApps)`
- `top_developers_by_success(Category, N, TopDevList)`

INFERENZA DI NUOVA CONOSCENZA

Per inferire nuova conoscenza sono state poste delle **query** alla KB utilizzando le **clausole** “count_editors_ choice”, “avg_downloads_by_category” e “avg_rating_by_category”. Queste ultime ci hanno permesso di inserire le seguenti colonne nel nuovo dataset finalized-playstore-apps.csv:

Num Editors Choice in Category	Average Downloads in Category	Average Rating of Category

ESEMPIO DI UTILIZZO DELLA KB

```
---- Esplorazione della Knowledge Base ----
1. Cerca app di uno specifico sviluppatore ordinate per download
2. Cerca app sotto un certo prezzo e con rating maggiore o uguale ad un certo valore
3. Cerca app poco scaricate ma con rating maggiore o uguale ad un certo valore
4. Cerca app di successo e con valutazione maggiore o uguale a un certo valore e ordinate per download
5. Cerca app sotto un certo prezzo e appartenenti ad una specifica categoria
6. Cerca app Editor's Choice appartenenti ad una specifica categoria e ordinate per download
7. Cerca app gratuite e con maggior numero di download
8. Cerca app più costose ordinate per download

---- Statistiche della Knowledge Base ----
9. Ottieni il numero di app Editor's Choice appartenenti ad una specifica categoria
10. Ottieni la valutazione media per una specifica categoria
11. Ordina tutte le categorie per valutazione media
12. Ordina tutte le categorie per numero totale di download
13. Cerca gli sviluppatori con più app di successo in una specifica categoria
X. Torna al menu principale

Scegli un'opzione:
```

```
Scegli un'opzione: 7
Inserisci il numero di app da visualizzare: 10
-----------------------------------------------
Nome App                          | Downloads
-----------------------------------------------
Google Play services              | 12057627016
YouTube                           | 9766230924
Google                            | 9154248491
Google Maps - Navigate & Explore  | 9141671889
Google Text-to-Speech             | 9034404884
Google Chrome: Fast & Secure      | 8925640788
Gmail                             | 8756574289
Android Accessibility Suite       | 7408134567
Google Drive                      | 7028265259
Facebook                          | 6782619635
-----------------------------------------------
```

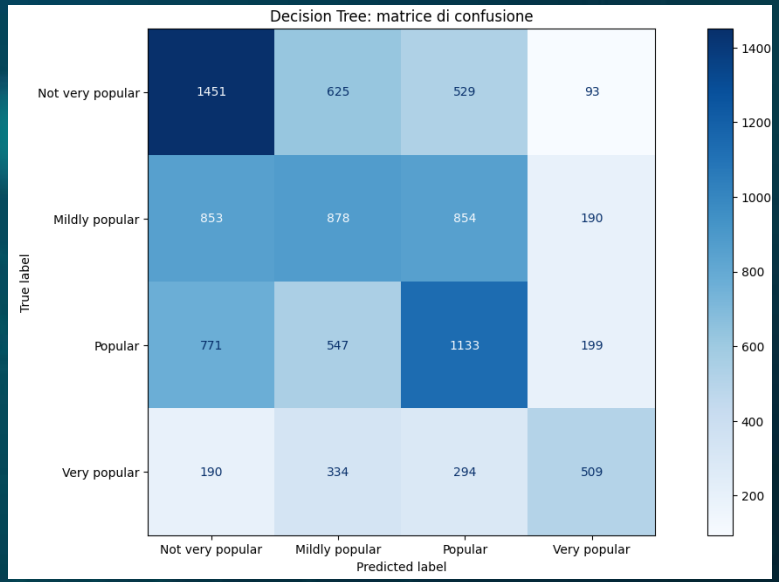
APPRENDIMENTO SUPERVISIONATO

03

DECISION TREE

La migliore combinazione di **iperparametri** è stata:

- criterion: gini
- max_depth: 20
- max_features: 0.2
- max_leaf_nodes: 30
- min_samples_leaf: 30
- min_samples_split: 20



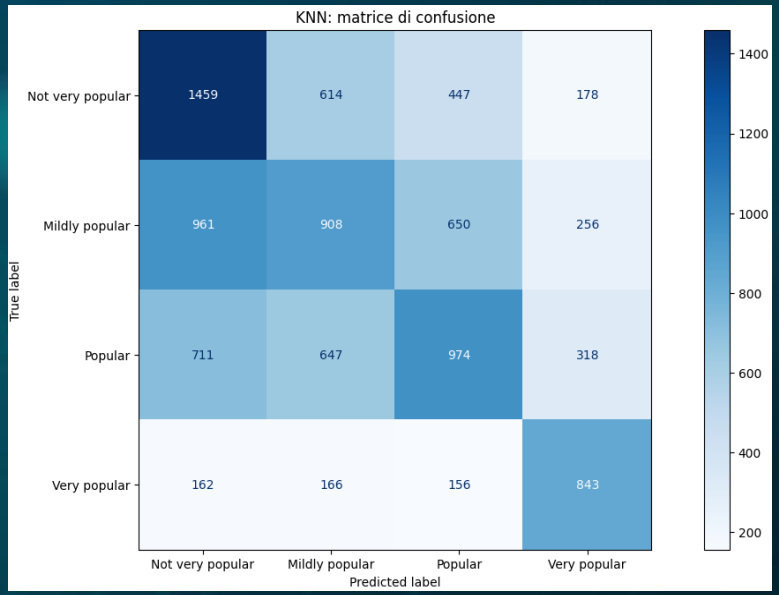
	Precision	Recall	F1-Score
1	0.44	0.54	0.49
2	0.37	0.32	0.34
3	0.40	0.43	0.42
4	0.51	0.38	0.44
Macro avg	0.43	0.42	0.42
Weighted avg	0.42	0.42	0.42

Accuracy	0.42

K-NEAREST NEIGHBORS

La migliore combinazione di **iperparametri** è stata:

- metric: manhattan
- n_neighbors: 15
- weights: uniform



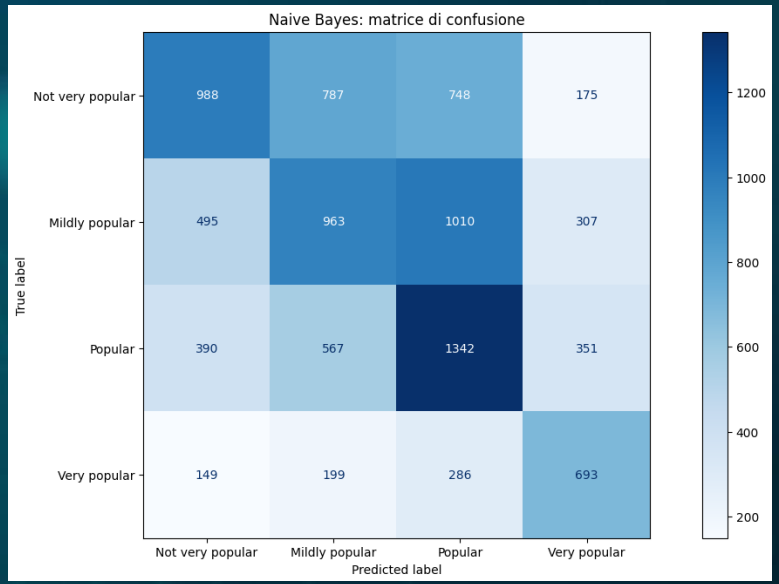
	Precision	Recall	F1-Score
1	0.44	0.54	0.49
2	0.39	0.33	0.36
3	0.44	0.37	0.40
4	0.53	0.64	0.58
Macro avg	0.45	0.47	0.45
Weighted avg	0.44	0.44	0.44

Accuracy	0.44

GAUSSIAN NAIVE BAYES

Il miglior **iperparametro** trovato è il seguente:

- var_smoothing: 0.43470131581250243



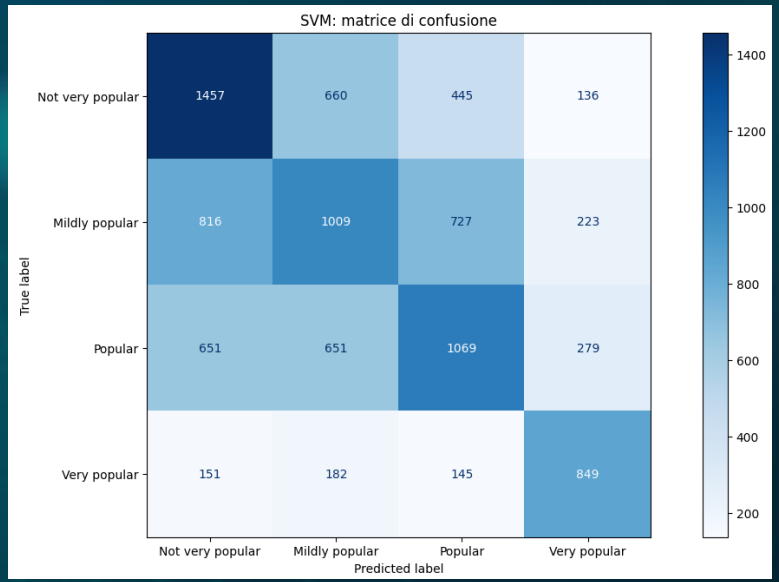
	Precision	Recall	F1-Score
1	0.49	0.37	0.42
2	0.38	0.35	0.36
3	0.40	0.51	0.44
4	0.45	0.52	0.49
Macro avg	0.43	0.44	0.43
Weighted avg	0.43	0.42	0.42

Accuracy	0.42

SUPPORT VECTOR MACHINE

La migliore combinazione di **iperparametri** è stata:

- C: 10
- gamma: scale
- kernel: rbf



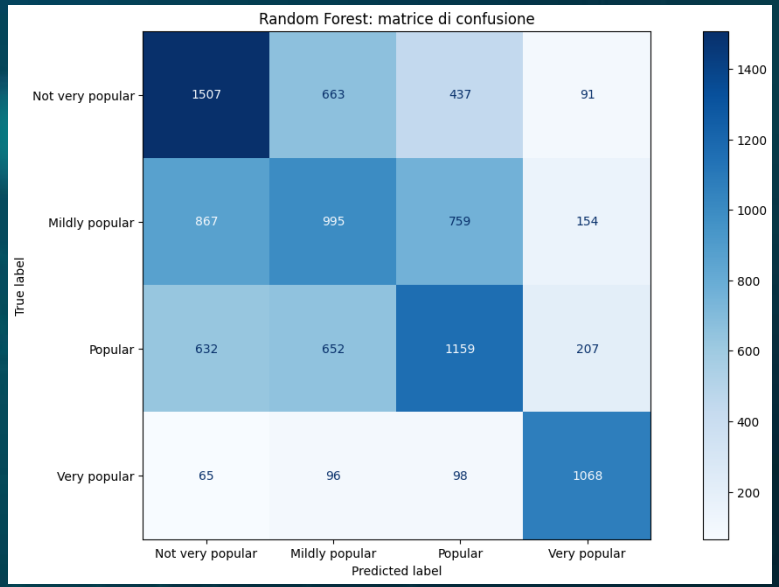
	Precision	Recall	F1-Score
1	0.47	0.54	0.50
2	0.40	0.36	0.38
3	0.45	0.40	0.42
4	0.57	0.64	0.60
Macro avg	0.47	0.49	0.48
Weighted avg	0.46	0.46	0.46

Accuracy	0.46

RANDOM FOREST

La migliore combinazione di **iperparametri** è stata:

- n_estimators: 300
- max_depth: None
- min_samples_leaf: 1
- min_samples_split: 4
- max_features: 0.3



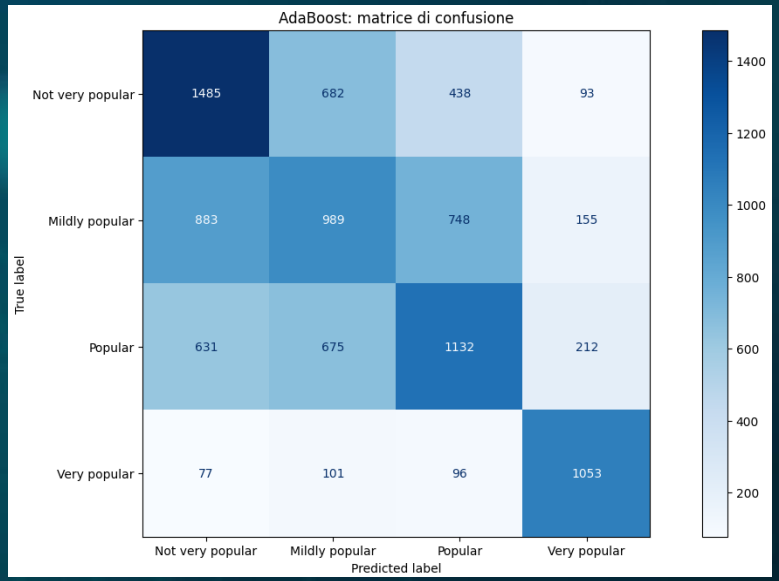
	Precision	Recall	F1-Score
1	0.49	0.56	0.52
2	0.41	0.36	0.38
3	0.47	0.44	0.45
4	0.70	0.80	0.75
Macro avg	0.52	0.54	0.53
Weighted avg	0.49	0.50	0.49

Accuracy	0.50

ADA BOOST

La migliore combinazione di **iperparametri** è stata:

- n_estimators: 100
- learning_rate: 0.1
- algorithm: SAMME
- estimators: RandomForestClassifier



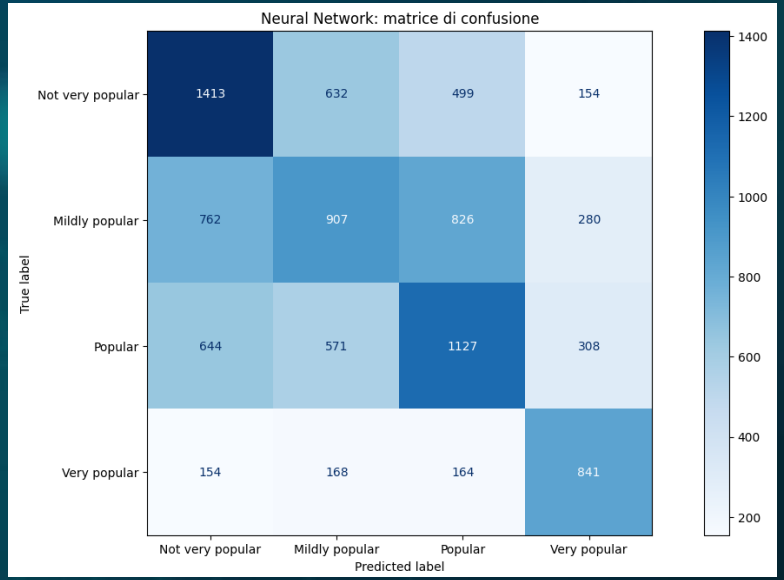
	Precision	Recall	F1-Score
1	0.48	0.55	0.51
2	0.40	0.36	0.38
3	0.47	0.43	0.45
4	0.70	0.79	0.74
Macro avg	0.51	0.53	0.52
Weighted avg	0.49	0.49	0.49

Accuracy	0.49

NEURAL NETWORK

La migliore combinazione di **iperparametri** è stata:

- hidden_layer_sizes: (50,)
- activation: relu
- solver: adam
- alpha: 0.05
- learning_rate: constant
- max_iter: 1500



	Precision	Recall	F1-Score
1	0.48	0.52	0.50
2	0.40	0.33	0.36
3	0.43	0.43	0.43
4	0.53	0.63	0.58
Macro avg	0.46	0.48	0.47
Weighted avg	0.45	0.45	0.45

Accuracy	0.45

CONFRONTO TRA I MODELLI

Il modello con le prestazioni migliori risulta essere **Random Forest**, seguito da **Ada Boost**.
Dal confronto delle prestazioni mostrate nelle slide precedenti si osserva come tutti i modelli, tranne il Gaussian Naive Bayes, classificano quasi sempre correttamente la classe **Not very popular**, probabilmente perché è ben rappresentata o distintiva.
Tuttavia, precision e recall sono più basse per la classe **Mildly popular**, suggerendo una sovrapposizione delle caratteristiche con altre classi, rendendola difficile da distinguere.
La classe **Very popular**, nonostante i pochi esempi, è invece classificata con alta precision e recall: questo può essere dovuto probabilmente ad **overfitting** sui pochi campioni disponibili.

Modello	Accuracy
Decision Tree	0.42
KNN	0.44
Gaussian Naive Bayes	0.42
SVM	0.46
Random Forest	0.50
Ada Boost	0.49
Neural Network	0.45

APPRENDIMENTO NON SUPERVISIONATO

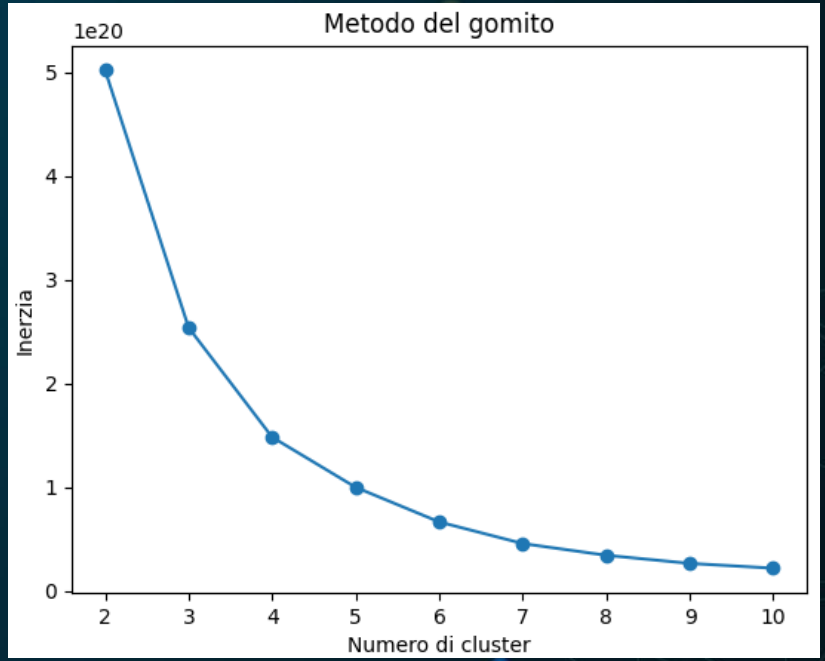
04

CLUSTERING

Il **clustering** è un metodo di apprendimento non supervisionato che raggruppa esempi simili in cluster. Nel nostro progetto è stato utilizzato per realizzare un **recommender system**.

Per individuare il numero ottimale di cluster, abbiamo utilizzato il **metodo del gomito**.

Dal grafico si osserva come i valori migliori di **k** su cui applicare il **K-Means** siano 3, 4 e 5: abbiamo pertanto scelto di dividere il dataset in **4 cluster**.



RECOMMENDER SYSTEM

Il **recommender system** realizzato utilizza la similarità del coseno per confrontare app simili. Il dataset viene prima convertito in formato numerico, compresa la feature **App Name** tramite **tokenizzazione** ed **embedding** con **Word2Vec**.
Successivamente, l'algoritmo di clustering K-Means suddivide il dataset in 4 cluster. Quando l'utente fornisce in input le informazioni dell'app che sta cercando, il sistema calcola la **similarità del coseno** tra questa e le app dello stesso cluster, ordinandole in maniera decrescente in base alla **somiglianza** per fornire le raccomandazioni.

```
Benvenuto!
Questo sistema è progettato per aiutare gli utenti e gli sviluppatori a fare scelte informate nel
mercato delle app, offrendo suggerimenti personalizzati e previsioni basate su analisi dei dati.

Scegli una delle seguenti opzioni:
1 - Raccomandazione di app simili
2 - Predizione del tasso di successo di un app non ancora sul mercato
3 - Calcolo della probabilità di successo di un app non ancora sul mercato tramite belief network
4 - Esplorazione della base di conoscenza
X - Esci
1
Le categorie disponibili sono:
-  Auto & Vehicles
-  Beauty
-  Communication
-  Creativity
-  Dating
-  Education
-  Entertainment
-  Events
-  Finance
-  Food & Drink
-  Games
-  Health & Fitness
-  House & Home
-  Lifestyle
-  Music & Audio
-  Parenting
-  Personalization
```

ESEMPIO DI UTILIZZO

```
-  Productivity
-  Reads
-  Shopping
-  Tools
-  Travel & Navigation
-  Weather
Quale categoria di app stai cercando?
Communication
Cerchi un'app gratuita o a pagamento?
gratis
Suggeriscimi il nome di un'app di tuo gradimento appartenente a questa categoria:
Whatsapp
Le app si classificano sulla base dei contenuti in:
-  Everyone
-  Teen
-  Adults
Quale dovrebbe essere la classificazione dei contenuti dell'app?
Everyone
Un'app "Editor's Choice" è un'app scelta dalla redazione come una delle app più innovative, creative e degne di nota presenti nello store.
Stai cercando un'app Editor's Choice?
si
Inserisci il numero di download che l'app dovrebbe avere:
5000000000
Qual è il numero minimo di stelle (tra 1 e 5) che l'app deve possedere?
4
Ricerca delle app simili...
```

ESEMPIO DI UTILIZZO

```
Ecco le applicazioni suggerite in base alle tue richieste:
```

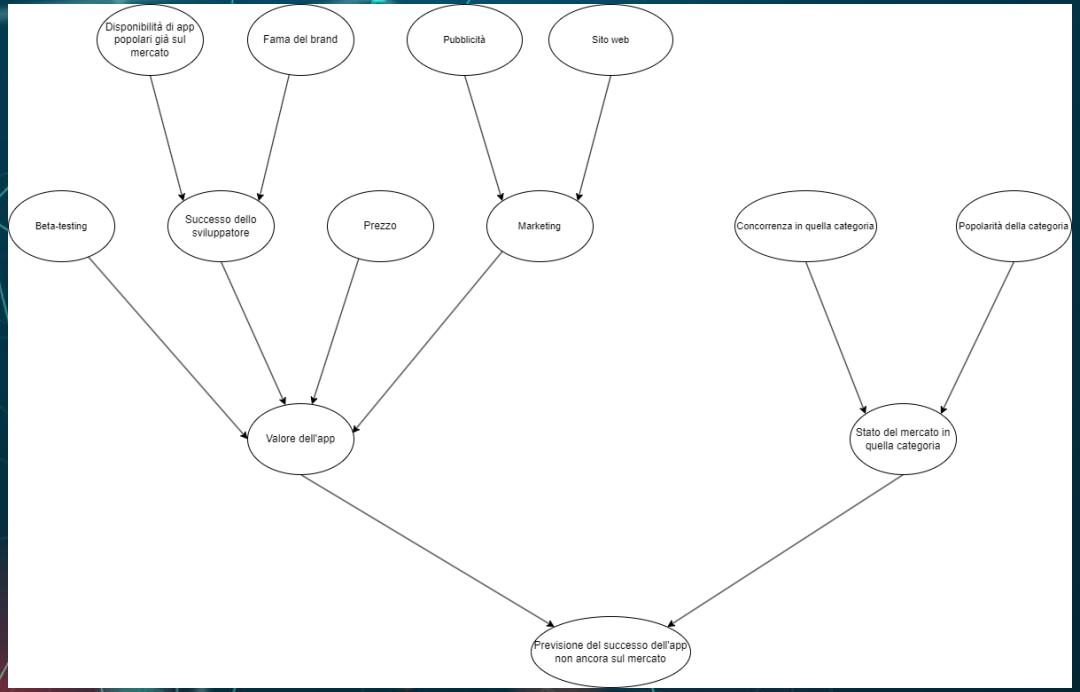
App Name	Rating	Downloads	Price ($)	Editors Choice	Success Rate
Contacts	4.3	885927111	0.0	False	Very popular
Opera Mini - fast web browser	4.3	551153058	0.0	False	Very popular
ANT Radio Service	4.0	1494252350	0.0	False	Very popular
Hangouts	4.0	5019518222	0.0	False	Very popular
Facebook	2.3	6782619635	0.0	False	Very popular
Messenger Lite: Free Calls & Messages	3.9	810116851	0.0	True	Very popular
TikTok	4.4	1645811582	0.0	False	Very popular
Gmail	4.2	8756574289	0.0	False	Very popular
Skype - free IM & video calls	4.3	1769991234	0.0	True	Very popular
WhatsApp Messenger	4.0	6265637751	0.0	True	Very popular
Samsung Push Service	4.2	4186667750	0.0	False	Very popular
Google Duo - High Quality Video Calls	4.5	4022259636	0.0	False	Very popular
LINE: Free Calls & Messages	4.1	861495092	0.0	False	Very popular
Carrier Services	4.3	1793502218	0.0	False	Very popular
Google Chrome: Fast & Secure	4.1	8925640788	0.0	False	Very popular
Messages	4.4	1931517750	0.0	False	Very popular
Snapchat	4.3	1621265491	0.0	True	Very popular
Instagram	3.8	3559871277	0.0	True	Very popular
Facebook Lite	3.1	2072296494	0.0	False	Very popular
imo free video calls and chat	4.1	926726001	0.0	False	Very popular

```

Vuoi vedere altri risultati? (si/no):
```

BELIEF NETWORK 05

STRUTTURA DELLA RETE BAYESIANA



Nel nostro progetto, la **rete bayesiana** è stata utilizzata al fine di predire la probabilità con cui una nuova app non ancora sul mercato possa riscuotere successo una volta rilasciata.

Come primo step, abbiamo costruito il **grafo aciclico orientato** mostrando le variabili da cui riteniamo possa dipendere il successo di una nuova applicazione e le loro relazioni.

PROBABILITÀ

Di seguito riportiamo alcuni esempi di **probabilità a priori** e **probabilità condizionate** da noi stimate:

- P(sviluppatoreSulMercato = si) = 0,3
- P(sviluppatoreSulMercato = no) = 0,7
- P(affiliazioneBrand = si) = 0,19
- P(affiliazioneBrand = no) = 0,81
- P(successoSviluppatore = alto | sviluppatoreSulMercato = si ∧ affiliazioneBrand = si) = 0,95
- P(successoSviluppatore = basso | sviluppatoreSulMercato = si ∧ affiliazioneBrand = si) = 0,05
- P(pubblicità = si) = 0,2
- P(pubblicità = no) = 0,8
- P(sitoWeb = si) = 0,1
- P(sitoWeb = no) = 0,9
- P(marketing = ottimo | pubblicità = si ∧ sitoWeb = si) = 0,97
- P(marketing = scarso | pubblicità = si ∧ sitoWeb = si) = 0,03
- P(betaTesting = si) = 0,26
- P(betaTesting = no) = 0,74
- P(prezzo = gratis) = 0,88
- P(prezzo = aPagamento) = 0,12
- P(valoreApp = alto | betaTesting = si ∧ successoSviluppatore = alto ∧ prezzo = gratis ∧ marketing = ottimo) = 0,98
- P(valoreApp = basso | betaTesting = si ∧ successoSviluppatore = alto ∧ prezzo = gratis ∧ marketing = ottimo) = 0,02

ESEMPIO DI UTILIZZO

```
Benvenuto!
Questo sistema è progettato per aiutare gli utenti e gli sviluppatori a fare scelte informate nel
mercato delle app, offrendo suggerimenti personalizzati e previsioni basate su analisi dei dati.

Scegli una delle seguenti opzioni:
1 - Raccomandazione di app simili
2 - Predizione del tasso di successo di un app non ancora sul mercato
3 - Calcolo della probabilità di successo di un app non ancora sul mercato tramite belief network
4 - Esplorazione della base di conoscenza
X - Esci
3
Per stimare la probabilità di successo della tua app, rispondi alle seguenti domande:
Hai già app di successo sul mercato?
Risposte possibili: si / no
no
Sei affiliato ad un brand famoso?
Risposte possibili: si / no
no
La tua app ha previsto una fase di beta testing?
Risposte possibili: si / no
si
La tua app è gratis o a pagamento?
Risposte possibili: gratis / a pagamento
gratis
```

Abbiamo poi codificato la struttura della rete bayesiana utilizzando la libreria **pybbn**. Le **probabilità condizionate** vengono aggiornate dinamicamente quando l'utente fornisce nuove informazioni come **evidenze**, permettendo una previsione più accurata del successo dell'applicazione.

```
La tua app ha un sito web associato?
Risposte possibili: si / no
si
Hai investito in pubblicità per la tua app?
Risposte possibili: si / no
si
La concorrenza nella categoria associata alla tua app è alta o bassa?
Risposte possibili: alta / bassa
bassa
Quanto è popolare la categoria associata alla tua app?
Risposte possibili: molto / poco
molto
La probabilità di successo dell'app è pari a 80.0%
Probabilità di successo alta.
Ottimo lavoro! Le probabilità di successo della tua app sono eccellenti.
Mantieni questo ritmo e continua a innovare; il successo è dietro l'angolo e il tuo impegno sarà sicuramente ricompensato.
```

CONCLUSIONI

06

SVILUPPI FUTURI

Possibli miglioramenti:

1. migliorare l'**accuracy** dei modelli (attualmente non superiore al 50%) cercando features più distintive, ad esempio delle recensioni testuali per un'analisi più accurata del **sentiment** verso le varie app;
2. **ampliare** il dataset con i dati del **Google Play Store** aggiornati al 2024;
3. migliorare il **calcolo del Success Rate** affinché tenga conto di altre features (es. Rating Count e Category), **pesate** in base alla loro influenza sul tasso di successo;
4. aggiungere al **recommender system** una funzionalità che consenta agli utenti di personalizzare un **profilo di interessi**, affinato e aggiornato automaticamente in base alle preferenze passate. Questo migliorerebbe la precisione delle raccomandazioni nel tempo;
5. realizzare un'**interfaccia grafica** per rendere la navigazione più semplice e intuitiva.

GRAZIE
PER
L'ATTENZIONE