



**Università degli Studi di Cagliari**

## **Relazione Progetto Web Analytics e Analisi Testuale**

**Fabio Mascia - Matteo Ghisu**

**PROJECT POKEMON**

# Introduzione

L'analisi di questo progetto accademico è incentrata, in via primaria, sull'andare a capire come il parere della community *Pokemon*, attraverso un Social Network, influenzi il prezzo medio delle carte collezionabili del medesimo brand.

L'unità statistica della nostra analisi è per l'appunto il singolo Pokemon, appartenente ad ognuna delle 9 generazioni uscite dal febbraio del 1996 a novembre 2022.

Come Social Network è stato scelto *Reddit*, e il motivo dietro questa scelta è prettamente di natura operativa, ovvero *Reddit* grazie alla sua API (Application Programming Interfaces), ci ha permesso di accedere facilmente ai commenti dei singoli utenti per ogni subreddit trattato; Inoltre, la presenza di più communities *Pokemon* all'interno del social ci ha fatto optare per questo sito.

Oltre questo, in via secondaria, l'analisi si è focalizzata anche sulle altre variabili raccolte, quali *Numero di Carte*, *Generazione*, *Prezzi Reverse Holo*....

## Scraping e raccolta dati sui pokemon

Lo scraping su "[pokecardvalues.com](https://pokecardvalues.com)" è stato effettuato attraverso l'espansione Chromedriver; un driver necessario per simulare la navigazione web su un sito come se interagisse l'utente. Mentre per quanto riguarda lo script è richiesto l'utilizzo delle librerie Selenium e BeautifulSoup.

Come prima cosa, attraverso chromedriver, si accede nella pagina dove sono elencati tutti i set di carte con i relativi link, che verranno successivamente raccolti in una lista (con l'utilizzo di BeautifulSoup).

Nel sito, più precisamente nella pagina dove sono raccolti tutti i nomi dei set, è presente una finestra pop-up che abbiamo provato a rimuovere con Selenium, attraverso una funzione, ma purtroppo continua a ripresentarsi fino a quando non si preme manualmente sul pulsante 'Accept'.

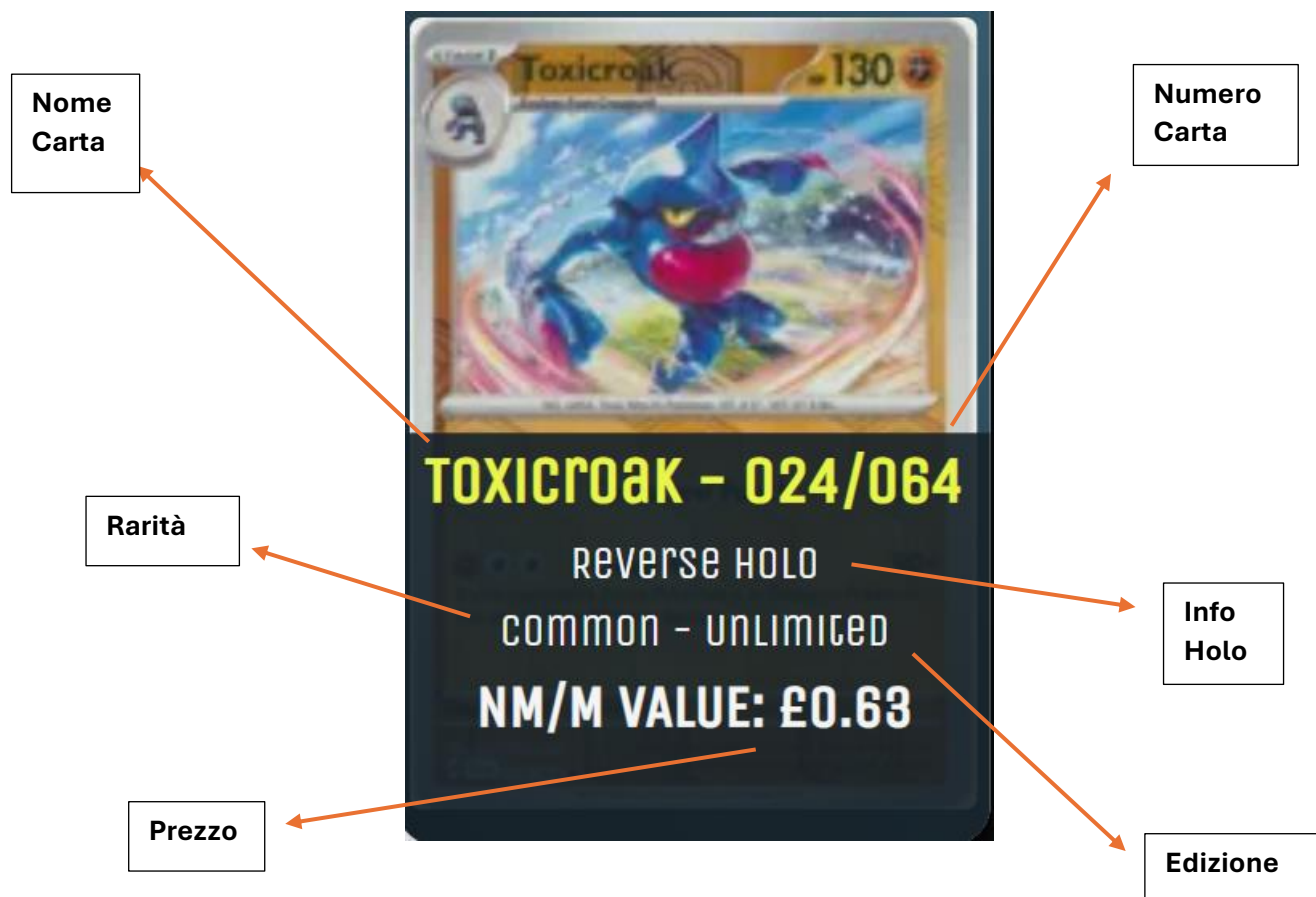
Grazie a Selenium, la pagina viene fatta scrollare fino alla fine così che venga calcolata l'altezza di quest'ultima; questo processo viene iterato fino a quando l'altezza in questione rimanga invariata così raggiungendo la fine della pagina.

Successivamente avviene la raccolta di tutti i dati, questo processo viene iterato fino all'ultima espansione.

Le variabili di dato raccolte sono le seguenti:

- ✚ Il nome della carta;
- ✚ Il numero della carta;
- ✚ Il prezzo di mercato della carta: quest'ultimo riguarda il prezzo di vendita medio di quella determinata carta in versione *Near Mint* (si intende una carta in condizioni pari al nuovo), su eBay negli ultimi 90 giorni da parte degli utenti.
- ✚ Info Holo: una carta può avere una sua versione lucente cosiddetta *Holo* oppure può presentare delle piccole decorazioni (*Reverse Holo*), infine abbiamo le versioni comuni denominate *Non-Holo*.
- ✚ Rarità
- ✚ Edizione
- ✚ Set di appartenenza

Di seguito un'immagine della carta in cui vengono riportate alcune delle info raccolte.



Questi dati vengono incasellati in un dizionario.

Tutti i dati vengono salvati poi in un csv denominato '*pokemon\_card\_data\_ultimo.csv*', in cui le osservazioni raccolte sono state all'incirca 34.000.

## Filtraggio Dataset

In questa seconda fase della nostra indagine sono state fatte delle manipolazioni sul dataset ottenuto precedentemente. In particolare, si è deciso di rinominare alcune colonne in italiano, oltre che delle ulteriori azioni di pulizia e di conversione dei dati.

Dal punto di vista operativo, in primo luogo abbiamo deciso di filtrare le carte in modo che apparissero solo quelle che raffigurassero unicamente dei Pokémon ed escludendo, perciò, tutte quelle carte quali *energia*, *capi-palestra* ....




Si è voluto procedere a questa pulizia in quanto ai fini dell'analisi, la nostra unità statistica sarà il singolo pokémon. Le osservazioni rimosse in questo caso sono state circa 5000.

Dopodiché, dal sito "[pokemondb.net](http://pokemondb.net)" si è estratto l'intera lista dei Pokémon di tutte le generazioni.

I nomi raccolti sono stati confrontati con quelli presenti nel dataset delle carte collezionabili, e solo le osservazioni che non presentavano il match sono state rimosse. Questo ulteriore filtraggio ha portato alla rimozione di circa 5000 osservazioni.

Ai fini dell'analisi esplorativa statistica, abbiamo deciso di aggiungere un'ulteriore variabile chiamata 'Gen' che indica la generazione di ciascun Pokémon.

Oltre questo si è voluto dividere la colonna prezzo medio delle carte per ogni Pokémon in tre categorie:








-  Prezzo Holo
-  Prezzo Non-Holo
-  Prezzo Reverse Holo

In questo modo si mostrano i prezzi medi per ogni versione di carta e questo potrà essere utile per estrarre ulteriori informazioni in vista della nostra indagine.

Ulteriori modifiche sono state applicate anche alle variabili 'Set', 'Nome Carta', che hanno visto una loro conversione in variabili 'Numero Set' e 'Numero carte', andando a contare rispettivamente le frequenze assolute per ogni occorrenza, così da avere un dato quantitativo.

Infine, come ultima modifica al dataset, si è voluto rimuovere la variabile 'Edizione' e 'Rarità'; questa decisione è arrivata a seguito degli innumerevoli livelli presenti per ciascuna variabile; livelli che erano privi di informazione, e che sono stati rimossi, per lasciare spazio alla variabile 'Gen'.

In conclusione, avremo un dataset, denominato 'Datasetfinale1.csv' da 1018 osservazioni in cui le nostre variabili saranno:

-  Pokemon
-  Numero Carte
-  Numero Set
-  Generazione Pokemon
-  Prezzo medio Non-Holo(€)
-  Prezzo medio Holo(€)
-  Prezzo medio Reverse Holo(€)

## Scraping dei commenti e Sentiment Analysis

Per la raccolta dei commenti dei singoli Pokemon su Reddit, ci siamo rivolti alle librerie Praw e BeatifoulSoup. Durante questa fase sono state rilevate alcune problematiche, sia in termini computazionali, sia con riferimento alla selezione del commento in base al contesto; queste problematiche saranno approfondite meglio nel paragrafo in questione.

Dal punto di vista operativo si è voluto ricercare tutti i post in cui comparisse il nome del Pokemon, e successivamente si va a raccogliere i commenti per quel post specifico. Questo processo viene iterato per ognuno dei Pokemon che sono stati raccolti dal sito <https://pokemondb.net/>. Questa prima parte viene svolta dalla funzione `get_comment_for_pokemon`.

```
def get_comments_for_pokemon(reddit, pokemon_name, subreddit='Pokemon', limit=1):
    try:
        search_results = reddit.subreddit(subreddit).search(pokemon_name, limit=limit)
        all_comments = []
        for submission in search_results:
            print(f"Post trovato: {submission.title} (ID: {submission.id})")
            submission.comments.replace_more(limit=None)
            all_comments.extend([comment.body for comment in submission.comments.list()])

        return all_comments
    except Exception as e:
        print(f"Errore durante il recupero dei commenti per {pokemon_name}: {str(e)}")
        return []
```

Come si può notare la funzione prende in ingresso un limitatore, che fissa il numero di post che devono essere ricercati per ciascun Pokemon. Questo limitatore è stato prefissato a “1” per motivi legati al calcolo computazionale; Difatti sono stati raccolti all’incirca 300 mila commenti per tutte le 1028 osservazioni, e un’ulteriore aggiunta di post sarebbe stato troppo oneroso in termini di calcolo; inoltre, a seguito di alcuni controlli, si è notato come all’aumentare del numero di post il risultato del sentiment non variava più di tanto.

Passando alla Sentiment Analysis, l’obiettivo era capire quanto il sentimento su ciascuna osservazione che in questo caso è il singolo commento, fosse Positiva e Negativa. Dopodiché l’oggetto di analisi è stato spostato sul singolo Pokemon e si è voluto prendere quello che è il “sentiment medio” per ciascuna etichetta (“Positivo”, “Negativo”, “Neutrale”).

In questo caso dal punto di vista operativo, ci siamo rivolti ad un modello roBERTa allenato su 124M di tweet; la scelta è ricaduta su un modello pre-addestrato in quanto, i commenti che sono stati raccolti non erano etichettati per ogni grado di sentiment (“Positivo”, “Negativo”, “Neutrale”). Inoltre, il modello essendo stato addestrato su commenti/tweet con una struttura sintattica molto affine a quelli raccolti su Reddit, questo si prestava bene per l’analisi che si voleva effettuare.

Di seguito si mostra la riga di codice in cui viene creato il classificatore basato sul modello roBERTa:

```
classifier = pipeline("sentiment-analysis", model='cardiffnlp/twitter-roberta-base-sentiment-latest',
                    tokenizer='cardiffnlp/twitter-roberta-base-sentiment-latest', top_k=None)
```

Il classificatore restituisce un dizionario contenente due chiavi: 'label' e 'score', per ognuna delle tre etichette possibili.

Quello che si ottiene in conclusione è un dataset, denominato *‘pokemon\_sentiment.csv’*, di cui si mostra una stampa delle prime 5 righe:

	Pokemon	Neutrale	Positivo	Negativo
1	Abomasnow	0.38348334307657245	0.2981001134447531	0.3184165469259579
2	Abra	0.2808230061580062	0.09159570357650919	0.6275812872462785
3	Abso1	0.32879604476551383	0.34470062827202097	0.32650332777145297
4	Accelgor	0.44650618538049774	0.2632735834593446	0.2902202301318721
5	Aegislash	0.44623405480136474	0.21486660702636037	0.33889934185140747

Ovviamente ai fini dell’analisi statistica si è deciso di omettere la variabile Neutrale in quanto considerata priva di informazione; la sua omissione è stata ripartita in maniera equa tra le due restanti variabili ovvero ‘Positivo’ e ‘Negativo’.

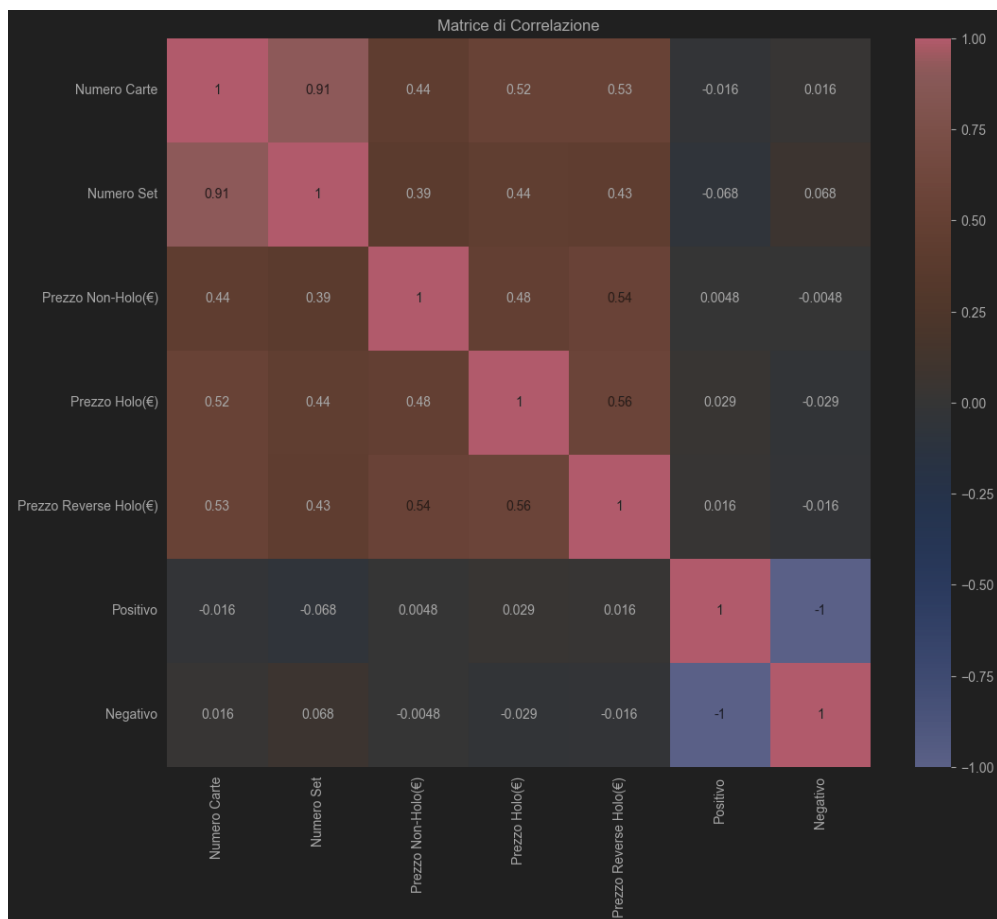
# Analisi statistica

L'analisi statistica viene effettuata su un merge tra il *Datasetfinale1.csv* e *pokemon\_sentiment.csv*.

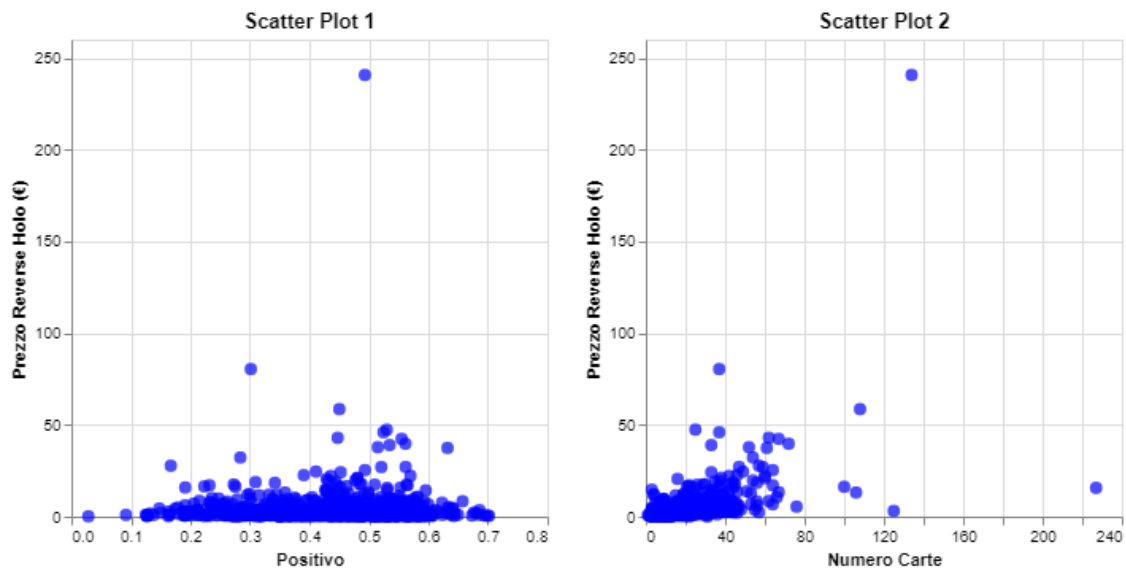
In primo luogo, si è voluto effettuare un'esplorazione del dataset, in cui andiamo ad osservare le relazioni tra le nostre variabili quantitative e qualitative.

Questa esplorazione è stata affiancata da strumenti quali il pairplot e il corplot che ci forniscono informazioni sulla dispersione delle osservazioni e sulla loro correlazione.

In particolare, la matrice di correlazione non ha mostrato valori significativi, se non tra le variabili 'Prezzo Reverse Holo' e 'Numero Carte' che hanno ottenuto un punteggio di 0.53. Mentre si può notare già da questa prima esplorazione come la variabile 'Positivo' e 'Negativo' non presenti una correlazione significativa in rapporto alla nostra variabile di risposta, con punteggi quali: 0,016.



Grazie alla libreria 'Altair' si riportano gli scatterplot, che ci forniscono la densità delle osservazioni tra le due coppie di variabili citate precedentemente.



Le osservazioni nello Scatterplot 2, risulta più esplicitivo e mostra una relazione un po' più chiara tra le due variabili rispetto allo Scatterplot 1.

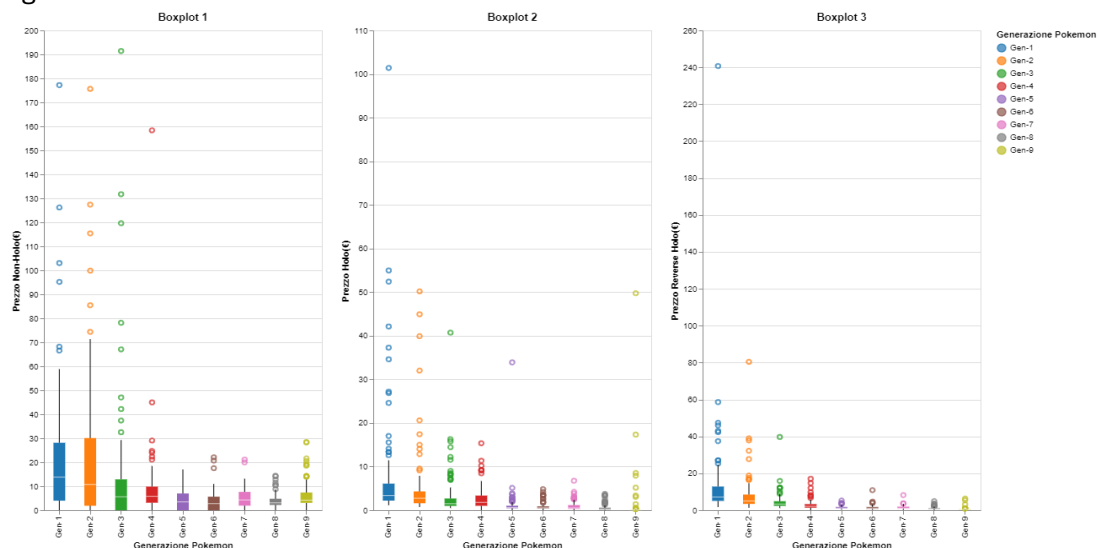
Infatti, in quest'ultimo grafico le osservazioni sono distribuite in modo uniforme lungo l'asse delle ascisse non mostrando una relazione interessante.

Per quanto riguarda la variabile categorica 'Generazione Pokemon' abbiamo deciso di utilizzare dei boxplot che mostrassero quali generazioni avessero il prezzo medio delle carte più alto in base alle varie categorie di carte.

Abbiamo notato che per tutte e tre le categorie di prezzo le prime 2-3 generazioni hanno un prezzo medio leggermente maggiore rispetto ai Pokémon delle ultime generazioni. Questo può significare che le carte Pokémon delle prime generazioni siano più apprezzate dalla community e le relative carte abbiano assunto un valore maggiore.

L'innalzamento del prezzo medio è dato anche dalla presenza di numerosi outlier che rappresentano Pokémon iconici delle prime generazioni che sono sicuramente tra quelli preferiti del brand. Un esempio è il Pokémon denominato '**Charizard**' che ha un prezzo medio di 177.18(€).

Notiamo inoltre che c'è un po' più di variabilità nei prezzi delle carte non-holo rispetto che alle altre due categorie di carte.



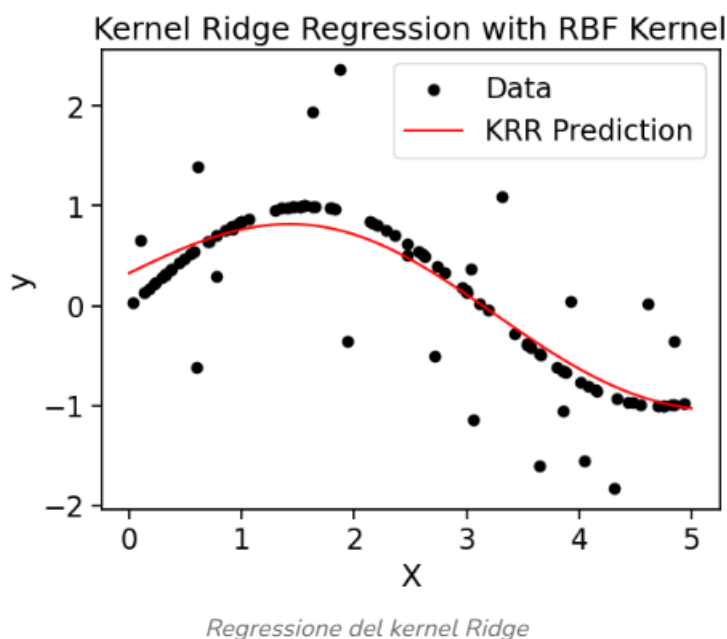
Dal punto di vista dell'analisi inferenziale abbiamo deciso di confrontare due modelli statistici caricati con la libreria 'sklearn'. Le variabili prese nel modello sono quelle che hanno presentato delle relazioni significative in sede di analisi esplorativa.

Per semplicità abbiamo preso solo una categoria di prezzo, dato che le relazioni risultavano abbastanza simili per tutte e tre le categorie di prezzo.

Abbiamo deciso di utilizzare la KRR (Kernel Ridge Regression) una tecnica di Ridge Regression che implementa il **kernel**. Il Kernel, genericamente impiegata nei SVM (Support Vector Machines) ci consente di capire le relazioni ad alta dimensionalità su un piano lineare grazie al *Kernel Trick*.

La KRR utilizza la perdita dell'errore quadratico e la regolarizzazione L2; quindi cerca di minimizzare la differenza quadratica tra le previsioni e i dati reali, con una penalizzazione sui pesi del modello.

Di seguito un'immagine di esempio della KRR con l'utilizzo di kernel radiale:

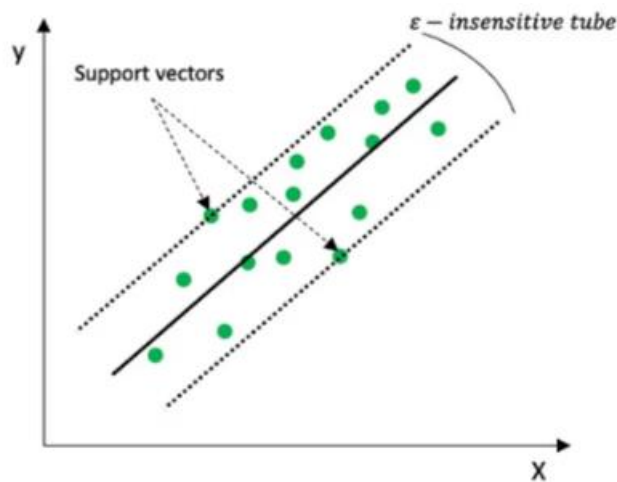


Il secondo modello che abbiamo utilizzato è stato l'SVR (Support Vector Regression) che è una tecnica di regressione che si basa anch'essa sui principi delle SVM.

La SVR cerca di trovare un *iperpiano* (o linea, nel caso di regressione con una sola variabile) che abbia il massimo margine possibile. In altre parole, cerca una linea o superficie che si avvicini il più possibile ai dati, ma con una tolleranza definita dal margine epsilon.

Di seguito un'immagine di esempio di SVR:





Regression problem using SVR

Le principali differenze tra le due tecniche riguardano quindi l'utilizzo di due diverse funzioni di perdita. Il modello KRR minimizza l'errore quadratico medio con una regolarizzazione L2, mentre il modello SVR utilizza la perdita epsilon-insensitive.

Questa differenza va a riflettersi anche nella velocità computazionale del modello; infatti, KRR utilizza una soluzione in forma chiusa per ottimizzare il suo modello: questo porta a calcolare i coefficienti ottimali con una formula matematica diretta. In particolare, la soluzione ottimale si ottiene con l'inversione della matrice di kernel, che riduce notevolmente i tempi di addestramento.

La SVR invece non ha una soluzione in forma chiusa quindi gli algoritmi devono eseguire diversi cicli di ottimizzazione iterativa per trovare i support vectors, e questo rallenta notevolmente l'addestramento.

Tuttavia, la SVR genera un modello sparso in cui utilizza i support vectors per effettuare le predizioni e questo lo rende un modello più efficiente rispetto a KRR in termini di tempo durante la fase di predizione, poiché il modello finale è spesso meno complesso.

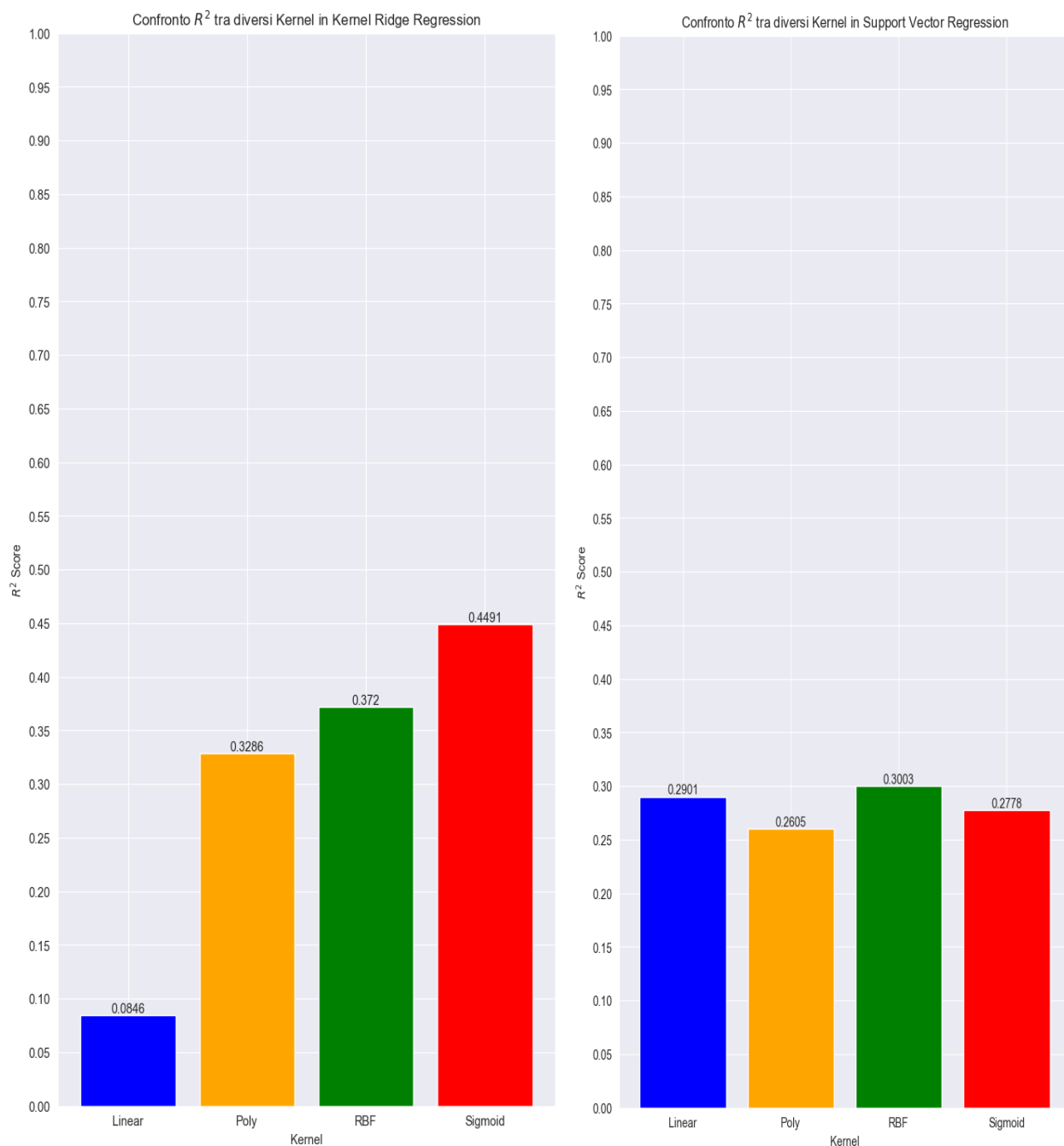
Inoltre, la SVR permette di ignorare errori minori, concentrandosi sui trend globali e riducendo la sensibilità agli outlier, il che può essere adatto per l'analisi del nostro fenomeno.

Per l'impostazione di entrambi i modelli, abbiamo impostato una grid-search con una 5 fold cross validation per la ricerca della migliore combinazione dei parametri da utilizzare, dopo aver effettuato le normali procedure di divisione in training e test set (70-30%) e dopo aver normalizzato i dati.

Su entrambi i modelli abbiamo deciso di confrontare 4 kernel:

- ✚ Lineare
- ✚ Polinomiale
- ✚ Radiale
- ✚ Sigmoidale

I grafici sui modelli e i risultati ottenuti sono mostrati qui di seguito per le relative coppie di variabili analizzate:



In questo caso andiamo a visualizzare la relazione tra 'Numero Carte' e 'Prezzo Reverse Holo' e notiamo che il kernel sigmoide ottiene risultati migliori nel KRR rispetto agli altri kernel, questo mostra che è necessario utilizzare un kernel non lineare, rispetto al lineare che evidenzia un risultato di 0.0846.

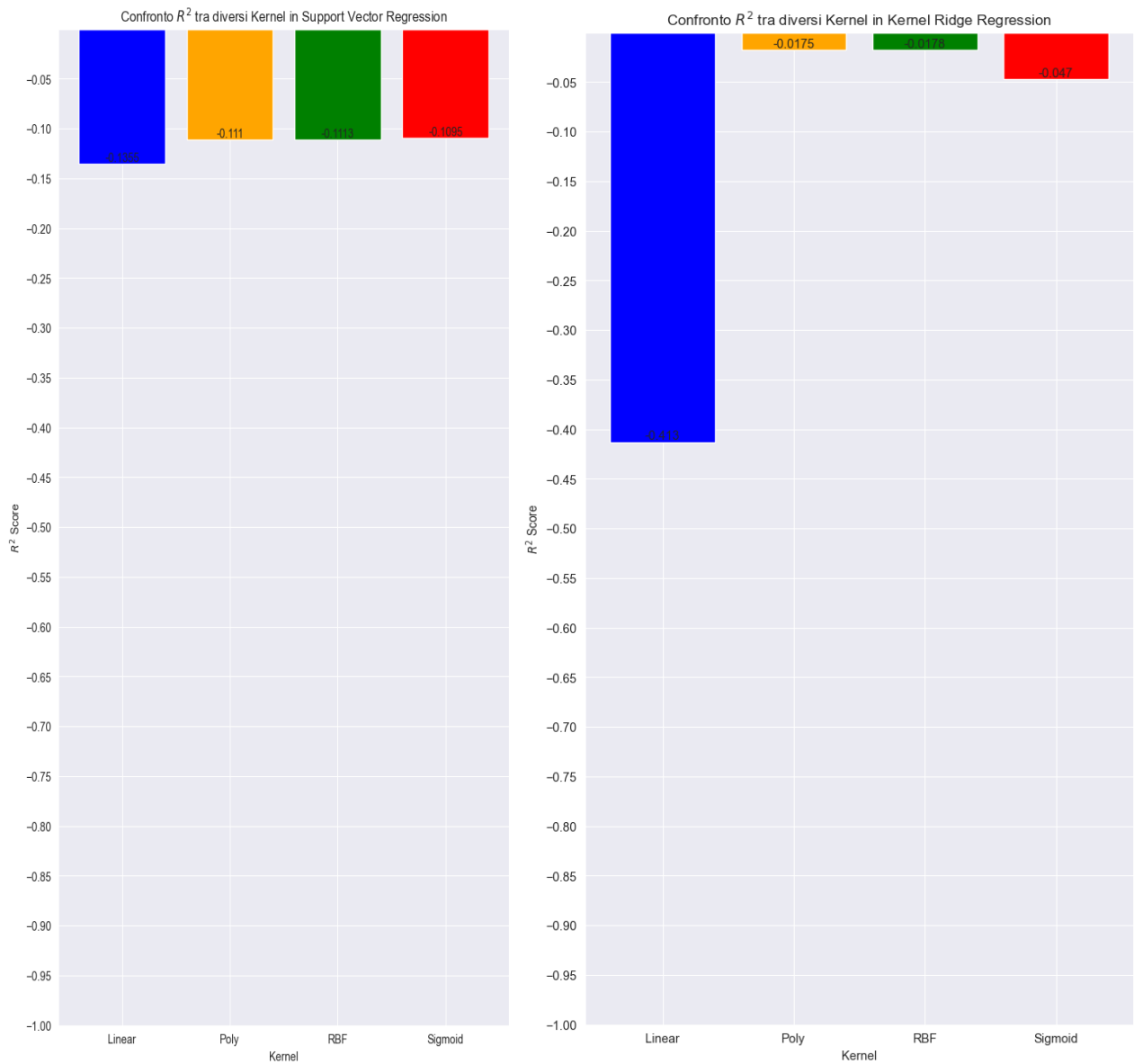
Nell'SVR i risultati dei kernel sono più equilibrati ma comunque abbastanza bassi.

C'è da sottolineare una rilevante differenza di tempi computazionali nei due modelli: SVR è risultato estremamente più lento quindi per dataset di queste dimensioni e viste le performance ottenute è preferibile utilizzare il KRR.

In generale i risultati ottenuti per questa coppia di variabili è abbastanza basso, e questo sta a significare che il prezzo non è spiegato bene dal numero delle carte per quel suddetto Pokemon.

Successivamente abbiamo analizzato la relazione tra la variabile ‘Positivo’ (commenti che hanno ottenuto un punteggio positivo su Reddit) e ‘Prezzo Reverse Holo’.

I risultati purtroppo sono stati disastrosi con l'utilizzo di entrambi i modelli:



Questo risultato sottolinea che probabilmente il prezzo delle carte è influenzato da altri fattori più specifici riguardanti il mercato collezionistico, come la tiratura della carta stessa, oppure l'illustrazione. Diversamente i commenti raccolti non erano correlati direttamente al valore di mercato della carta stessa.

# Limiti e Problematiche della Ricerca

Durante la ricerca si sono presentate alcune problematiche e alcune limitazioni a cui abbiamo cercato di porre rimedio, ma che purtroppo non hanno trovato soluzione, e questo per ovvi motivi ha inficiato su quello che sarà l'esito dell'analisi.

Uno dei primi limiti riscontrato è proprio in sede di raccolta dei commenti; difatti sebbene il nostro intento iniziale fosse di trovare post coerenti con il nostro obiettivo di ricerca, era abbastanza difficile trovare contenuti contestualizzati singolarmente a quanto un particolare Pokémon, piaccia o meno. Inoltre, ancora di più, è difficile trovarne per quei Pokémon che non risultano essere troppo 'chiacchierati'. Per cui il sentiment che è stato raccolto risente di un cosiddetto *bias di selezione* che introdurrà nella nostra analisi una certa soglia di variabilità che avrà un peso importante anche in sede di creazione del modello.

Per cercare di ovviare a questo abbiamo cercato di aumentare il numero di post raccolti per ciascun Pokémon, questo come accennato precedentemente non ha portato ad un miglioramento in termini di risultati.



**FIGURA 2 - GENGAR**

Un esempio di questa problematica è stato il pokémon chiamato "**Charizard**" (*Figura 1*), uno dei personaggi principali del brand sia nel gioco di carte collezionabili che nella serie animata. Quest'ultimo secondo le nostre aspettative doveva presentare un



**FIGURA 1 - CHARIZARD**

riscontro quasi in totalità positivo, eppure si trova in 165° posizione con riferimento al sentiment Positivo. Questo perché avviene? I commenti che sono stati selezionati non sono riferiti direttamente ad un post che parla direttamente di quanto Charizard piaccia o meno; sebbene qualche

commento possa essere in linea con quanto ricercato dall'analisi, altri purtroppo risentono di questo rumore.

Tuttavia, ci sono state anche alcune osservazioni in linea con quello che ci si aspettava, questo è il caso del Pokémon chiamato "**Gengar**" (*Figura 2*) che si presenta in 25° posizione.

Oltre questo limite, sono stati riscontrati problemi computazionali causati dalla quantità di dati raccolti, sia in sede di scraping delle carte collezionabili, sia in sede di raccolta dei commenti da parte della community.

## Conclusioni

In conclusione, si può affermare che il prezzo delle carte collezionabili Pokemon andrebbe analizzato in relazione ad altri fattori, magari legati all'utilità del gioco competitivo, alla tiratura della carta, all'illustratore della carta, alla qualità dell'artwork ecc.