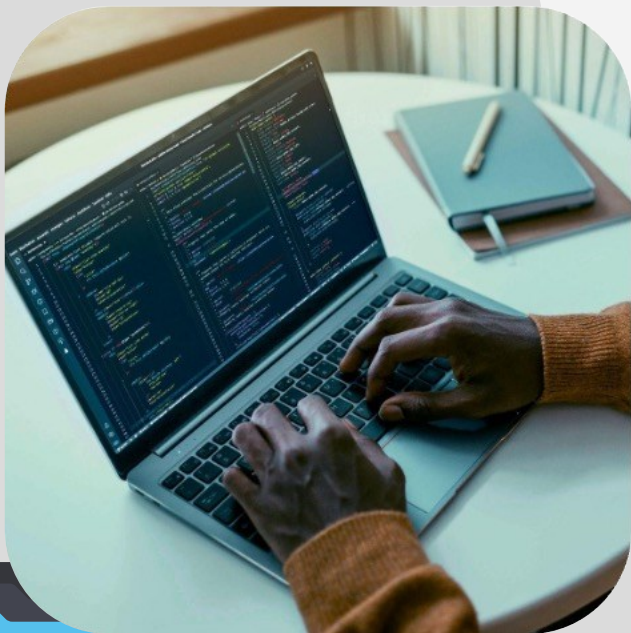


# Analisi delle Recensioni False Mediante Machine Learning

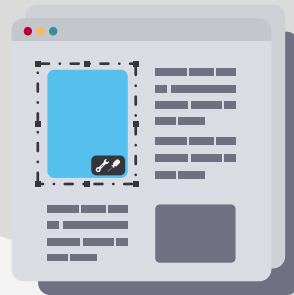
Sviluppo di un Bot per la Rilevazione Automatica





01

# Introduzione





# Contesto e Obiettivi

Le recensioni dei clienti rappresentano feedback importanti forniti da coloro che hanno utilizzato o acquistato un prodotto o un servizio.

In questo ambito però, le recensioni false sono un problema crescente che induce molto spesso in errore i consumatori, effettuando un acquisto che si rivelerà diverso da come era stato descritto.

Diventa così fondamentale sviluppare strategie innovative per proteggere i consumatori rilevando e contrastando le recensioni fasulle.



# Contesto e Obiettivi

Per affrontare questa sfida, abbiamo sviluppato FakeReviewDetector:  
Un bot in grado di distinguere e individuare con precisione le recensioni false da quelle autentiche, utilizzando tecniche di Machine Learning e di NLP.



FakeReviewDetector

# Approccio



**01**

**Studio del  
problema e  
Raccolta Dati**

**02**

**Analisi e Selezione delle  
caratteristiche**

**03**

**Pulizia e Pre-elaborazione  
dei dati**

**04**

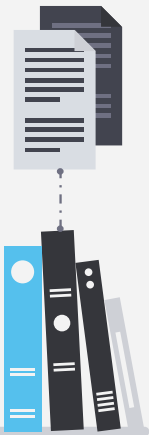
**Addestramento  
dei Modelli di  
Machine Learning**

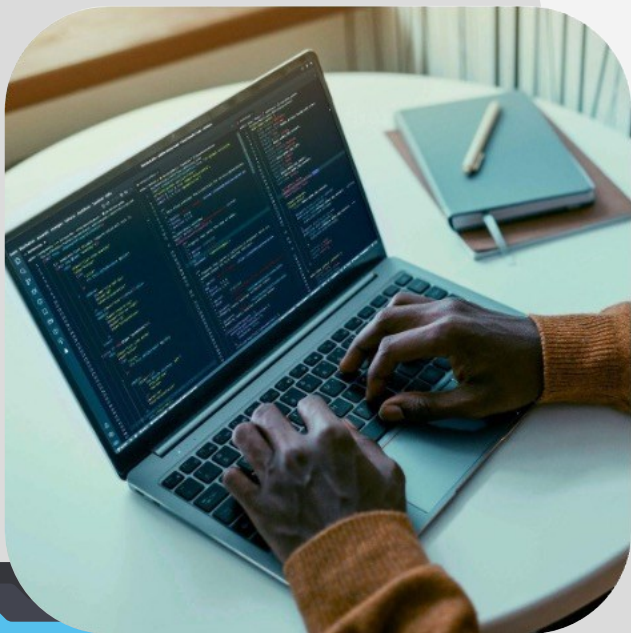
**05**

**Test e Valutazione  
delle Prestazioni dei  
modelli**

**06**

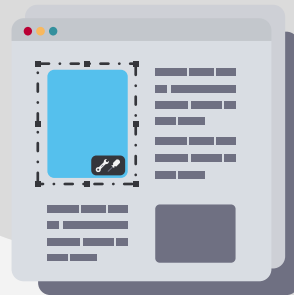
**Implementazione di  
FakeReviewDetector**





# 02

## Stato dell'Arte





# Fake Reviews

Esploriamo le tipologie di Fake Reviews nel dettaglio.

<u>Recensioni Positive Falsificate</u>	Fatte dai venditori stessi dei prodotti fingendosi dei clienti soddisfatti per aumentare la reputazione dei loro prodotti.
<u>Recensioni Negative Falsificate</u>	I concorrenti possono pubblicare recensioni negative per screditare prodotti di altri marchi.
<u>Recensioni di Bot Automatici</u>	Recensioni generiche che non forniscono dettagli sul prodotto e difficili da distinguere da quelle autentiche.
<u>Recensioni di Clienti Non Esistenti</u>	Account falsi utilizzati per simulare recensioni in modo da aumentare la popolarità del prodotto.
<u>Recensioni di Acquisti Non Verificati</u>	Recensioni effettuate senza aver acquistato effettivamente il prodotto.



# Approcci Attuali

## Apprendimento Supervisionato Tradizionale

Metodo di machine learning in cui un modello viene addestrato su un insieme di dati etichettati, cioè dati in cui l'output desiderato è noto.

## Approcci Basati su Reti Neurali

Ispirati alla struttura del cervello umano, sono in grado di simulare relazioni complesse tra input e output, consentendo loro di apprendere da enormi quantità di dati e di migliorare le loro prestazioni

## Ensemble Learning

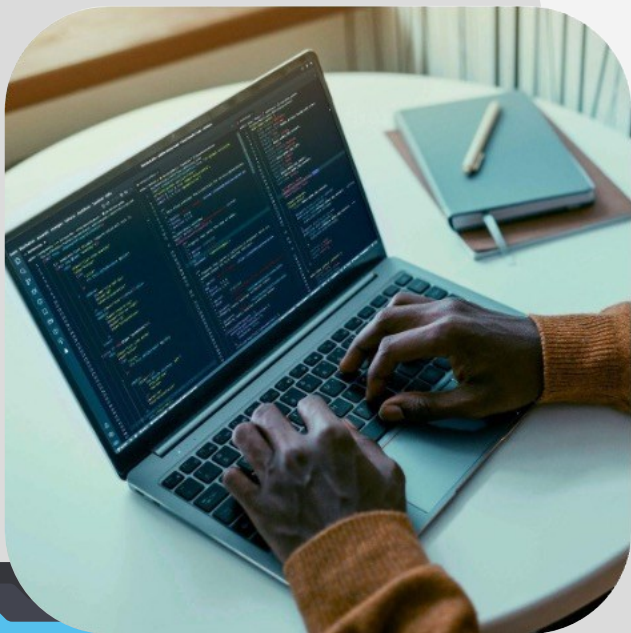
Approccio che mira a migliorare le prestazioni predittive combinando le previsioni di due o più modelli.

## Reti Neurali Ricorrenti e Generative Adversarial Networks

Due importanti sviluppi nel campo del deep learning e del machine learning che consentono di gestire dati sequenziali e di generare dati artificiali con prestazioni sempre più avanzate.

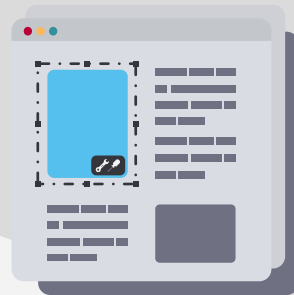






# 03

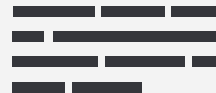
## Tools





# Tool Utilizzati

**Python (3.11.4):** Scelto per la sua versatilità e per la sua sintassi chiara e leggibile, che facilita la manutenzione e l'aggiornamento del codice. Librerie utilizzate:



- **Pandas:** Libreria utilizzata per gestire e preparare i dati prima dell'addestramento del modello, consentendo operazioni efficienti di pulizia, selezione e trasformazione dei dati.
- **Scikit-learn:** Libreria utilizzata per l'addestramento dei modelli di machine learning, principalmente perché offre una vasta gamma di algoritmi efficienti e facili da usare.
- **Matplotlib:** Impiegata per la creazione e la personalizzazione di grafici e visualizzazioni di dati.





# Jupyter Notebook

Celle eseguibili  
singolarmente:

```
[4]: def text_process(review):  
      nopunc = [char for char in review if char not in string.punctuation]  
      nopunc = ''.join(nopunc)  
      return [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
```

Combinazione di  
più linguaggi:

## Lettura Dataset

```
[2]: raw_dataset = pd.read_csv('../data/raw_dataset.csv')
```

Visualizzazione  
output :

```
[10]: print("Shape of Bag of Words Transformer for the entire reviews corpus:", bow_reviews.shape)  
      print("Amount of non zero values in the bag of words model:", bow_reviews.nnz)
```

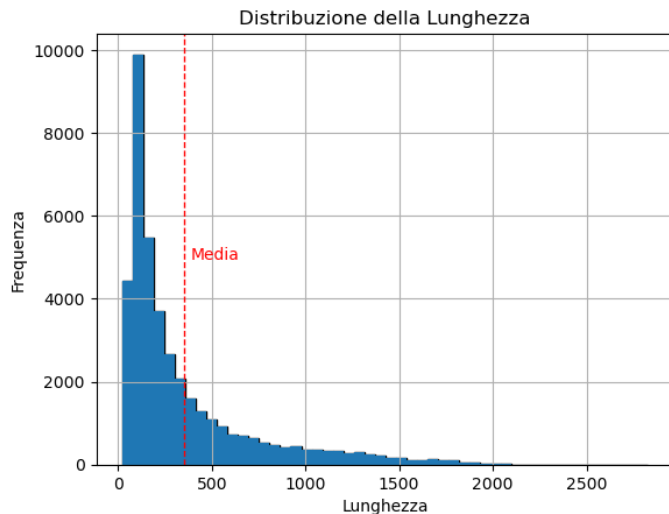
```
Shape of Bag of Words Transformer for the entire reviews corpus: (40431, 34489)  
Amount of non zero values in the bag of words model: 1001954
```

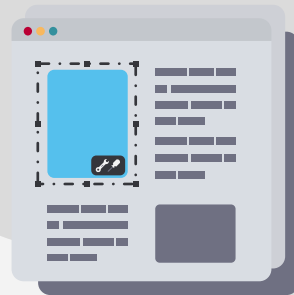
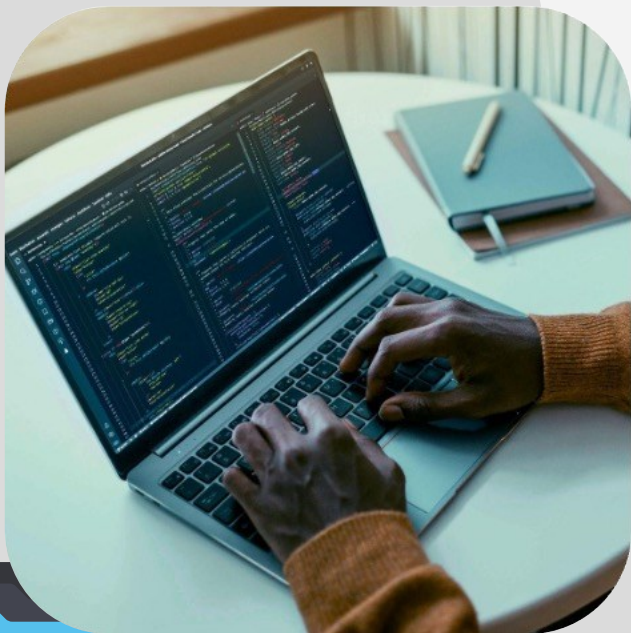


## Caratteristiche della feature "Label"

Nella prossima cella viene descritta, mediante un grafico, la distribuzione della lunghezza della feature "label"

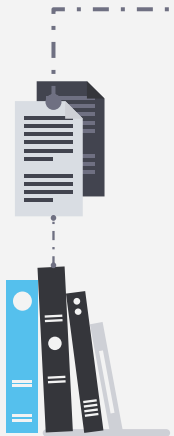
```
[16]: plt.hist(raw_dataset['length'], bins=50, color='skyblue', edgecolor='black')
plt.title('Distribuzione della Lunghezza')
plt.xlabel('Lunghezza')
plt.ylabel('Frequenza')
plt.grid(True)
plt.axvline(raw_dataset['length'].mean(), color='red', linestyle='dashed', linewidth=1)
plt.text(raw_dataset['length'].mean()*1.1, max(plt.hist(raw_dataset['length'], bins=50)[0])/2, 'Media', color = 'red')
plt.show()
```





# 04

## Presentazione dei Dati



# raw\_dataset

Dataset grezzo ancora da processare



**40 432**

Numero istanze totali

**4**

Features

# Features

## Categoria

Rappresenta la macro categoria del prodotto recensito.

## Rating

Valutazione numerica che va da 1 a 5 e riflette il giudizio dell'utente sul prodotto recensito.

## Autenticità

Valore booleano che indica se una recensione è stata generata da un computer (CG) o è originale (OR).

## Testo

Contenuto testuale della recensione.



category	rating	label	text_
Home_and_Kitchen	5.0	CG	Love this! Well made, sturdy, and very comfor...
Home_and_Kitchen	5.0	CG	love it, a great upgrade from the original. I...
Home_and_Kitchen	5.0	CG	This pillow saved my back. I love the look and...
Home_and_Kitchen	1.0	CG	Missing information on how to use it, but it i...
Home_and_Kitchen	5.0	CG	Very nice set. Good quality. We have had the s...

# «Text» come unica feature utilizzata, perché?



## Rilevanza Informativa

Il testo delle recensioni è una miniera di informazioni linguistiche e semantiche essenziali per identificare le recensioni false.

## Analisi Linguistica (NLP)

Le recensioni false tendono a presentare tratti linguistici distintivi, identificabili attraverso tecniche di Natural Language Processing (NLP).

## Semplificazione del Modello

Limitarsi al testo aiuta a prevenire l'overfitting, mantenendo il modello più semplice e interpretabile. Questo approccio riduce la complessità e facilita la manutenzione del modello.

## Accessibilità dei Dati

Il testo delle recensioni è facilmente reperibile e disponibile. Altre tipologie di dati, come informazioni utente o contesti di acquisto, possono essere limitati da restrizioni sulla privacy e normative

## Scalabilità e Riutilizzo

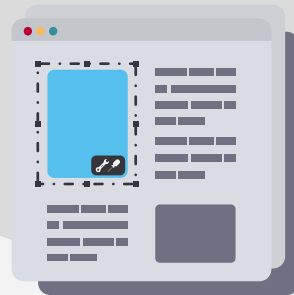
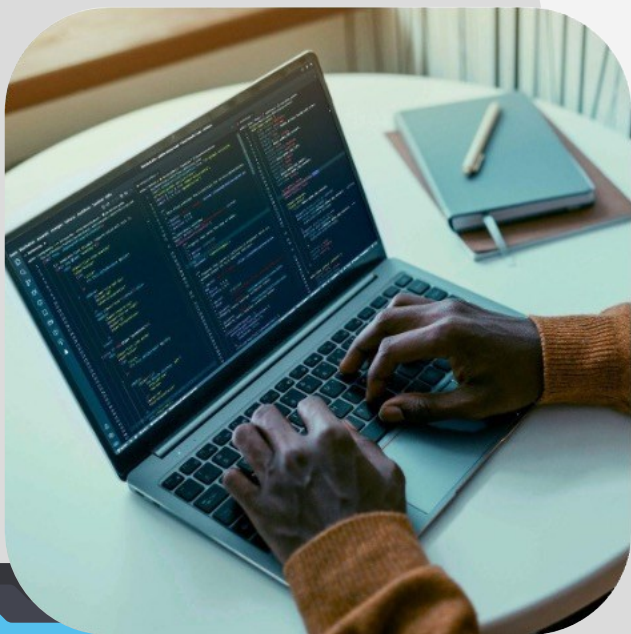
Un modello focalizzato sul testo è più facilmente adattabile a diverse piattaforme di recensioni, senza necessità di modifiche per adattarsi a set di dati non testuali.

## Considerazioni etiche

L'utilizzo di dati personali degli utenti potrebbe sollevare questioni di privacy ed etica. Pertanto, l'analisi testuale rappresenta una scelta prudente e rispettosa delle normative vigenti.







# 05

## Pre-Elaborazione dei Dati

# Pre-Elaborazione dei Dati



## Operazioni di base:

**Conversione in Stringa:** Abbiamo convertito ogni entry nella colonna «text» in stringa, un passo essenziale per gestire correttamente i dati testuali.

**Conversione in Lowercase:** Abbiamo trasformato ogni lettera nella colonna «text» in minuscolo. Cruciale per ridurre la varianza causata dalle differenze tra maiuscole e minuscole.



# Pre-Elaborazione dei Dati



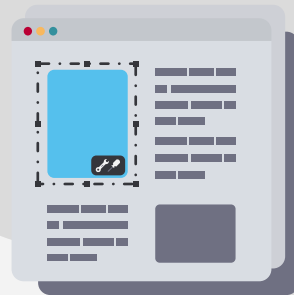
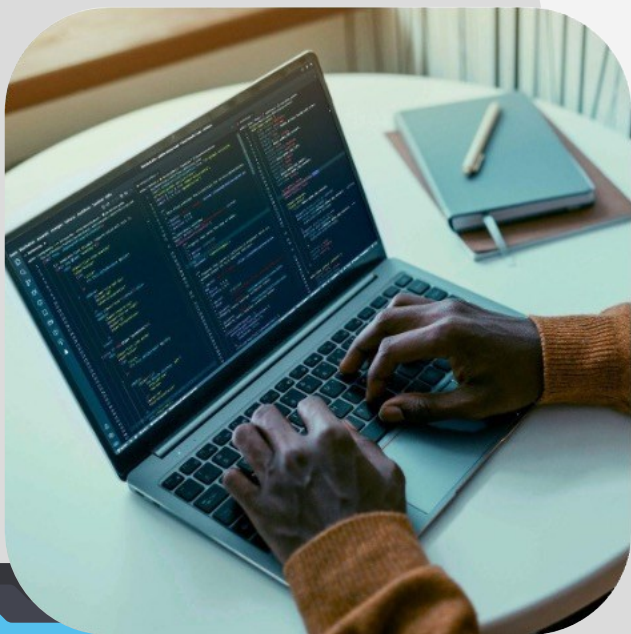
## Analisi Testuale e Pulizia:

**Tokenizzazione:** Suddivisione del testo in singole parole o token, operazione cruciale per analizzare la struttura e i contenuti delle recensioni, consentendoci di identificare schemi linguistici o usi insoliti del linguaggio che potrebbero indicare una recensione falsa.

**Rimozione delle StopWords:** Abbiamo eliminato parole comuni che aggiungevano poco significato al testo. Questo ha permesso al nostro modello di concentrarsi su parole più rilevanti, fondamentali per identificare l'autenticità o la falsità delle recensioni.

**Rimozione dei Numeri e della Punteggiatura:** Rimozione dei numeri e dei segni di punteggiatura, poiché spesso non contribuivano al rilevamento delle recensioni false.





# 06

## Addestramento

# Tecniche di NLP: BoW & TF-IDF

## Bag of Words (BoW)

- Trasforma il testo in vettori di frequenza di parole.
- Ignora l'ordine e il contesto delle parole.
- Utile per classificazione dei testi e analisi del sentiment.

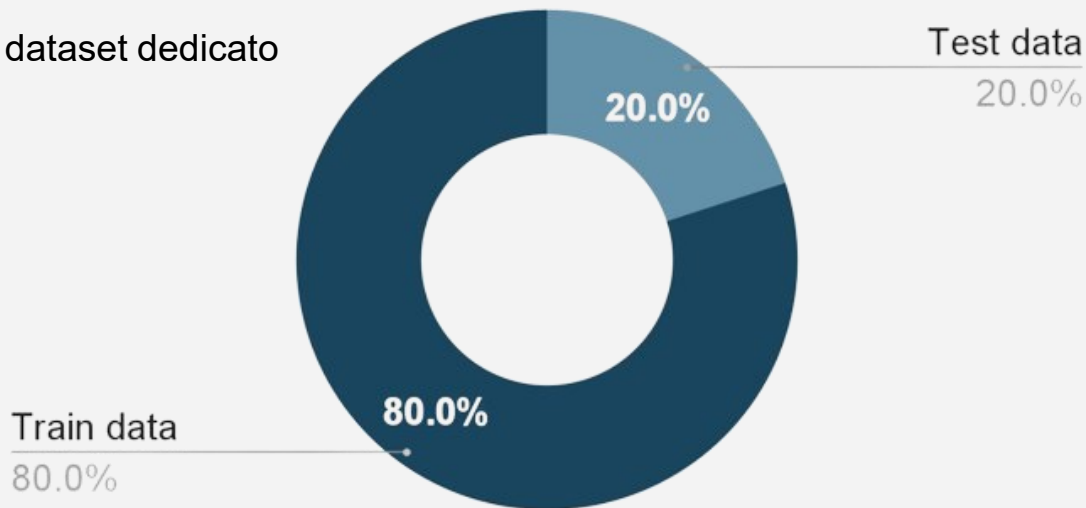
## Term Frequency-Inverse Document Frequency (TF-IDF)

- Estende il modello BoW valutando l'importanza delle parole.
- Combina la frequenza del termine (TF) con l'inverso della frequenza nei documenti (IDF).
- Evidenzia parole chiave uniche nei documenti, utile per ricerca e elaborazione di informazioni.

# Partizionamento Dataset

Abbiamo **diviso il dataset** pre-processato in due parti:

- **Train set:** 80% del dataset dedicato all'addestramento
- **Test set:** 20% del dataset dedicato al test dei modelli



# Pipeline di Scikit-learn

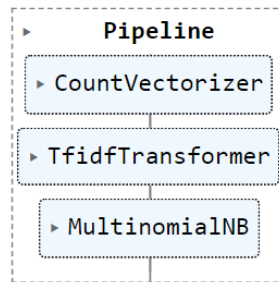
## Integrazione della Classe Pipeline

- Semplifica elaborazione del testo e addestramento del modello.
- Definisce tre operazioni principali in ogni pipeline:
  1. Conversione in Bag-of-Words.
  2. Trasformazione TF-IDF.
  3. Applicazione di una specifica funzione di classificazione.

```
[16]: pipeline = Pipeline([  
    ('bow', CountVectorizer(analyzer=text_process)),  
    ('tfidf', TfidfTransformer()),  
    ('classifier', MultinomialNB())  
])
```

```
[17]: pipeline.fit(review_train, label_train)
```

```
[17]:
```



# Algoritmi utilizzati



## Random Forest

Utilizza un insieme di alberi decisionali per migliorare la precisione della classificazione.

## KNeighbors Classifier

Basato sui vicini più prossimi, è particolarmente utile per identificare schemi non lineari.

## Multinomial Naïve Bayes

Particolarmente adatto per la classificazione di testi, e efficace nel riconoscere schemi e tendenze nelle recensioni.

## Support Vector Classifier

È rinomato per la sua efficienza nella classificazione di dati complessi. È particolarmente utile quando esistono margini sottili di differenziazione.

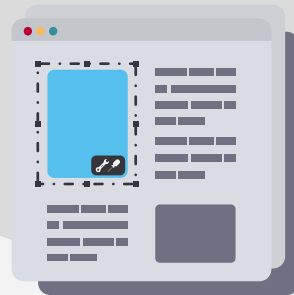
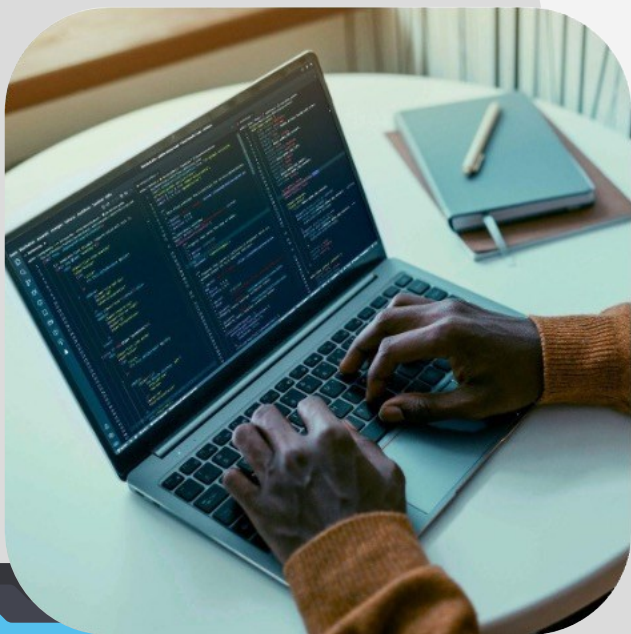
## Decision Tree Classifier

Basato sulla costruzione di alberi decisionali, questo algoritmo è utile per visualizzare e comprendere il processo decisionale.

## Logistic Regression

È particolarmente utile per la sua capacità di fornire probabilità, offrendo una misura della sicurezza con cui una recensione può essere classificata come falsa.





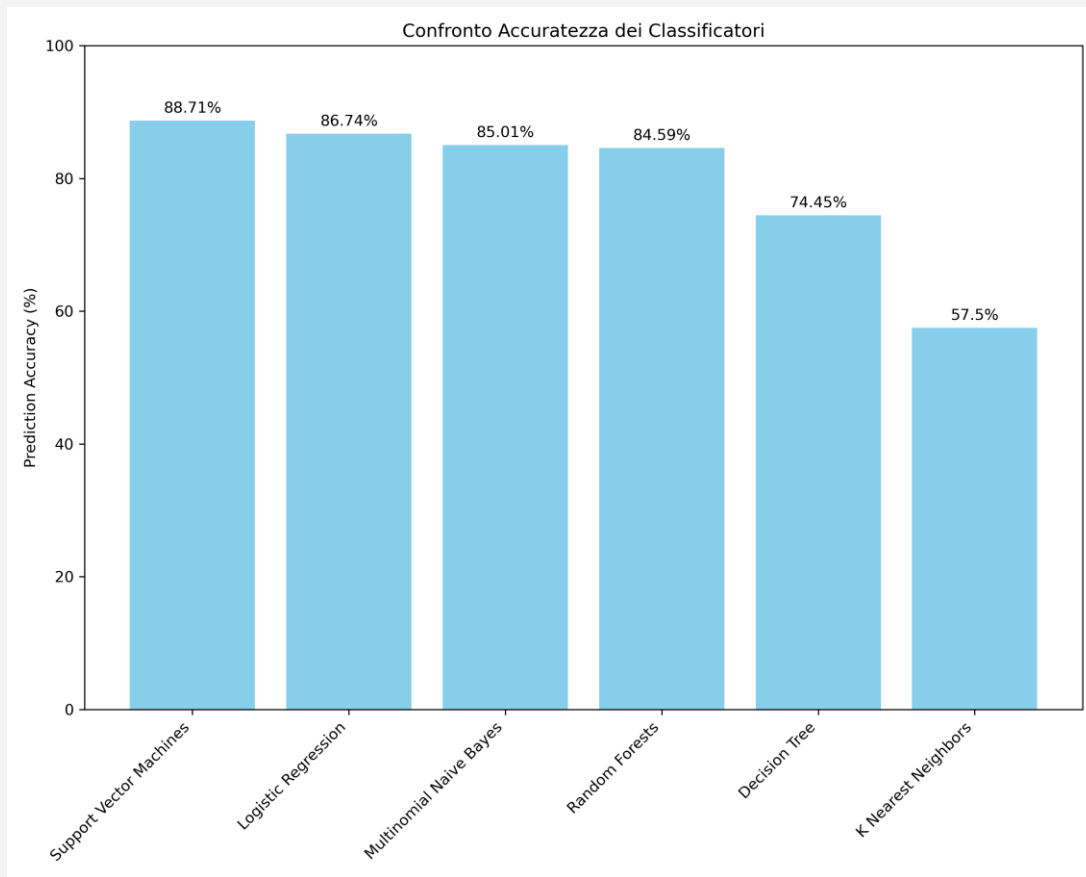
# 07

## Test e Valutazione

# Valutazione dei Modelli

- **Obiettivo:** Testare l'accuratezza dei modelli su dati non visti durante l'addestramento (20% del dataset).
- Metrica utilizzata per il confronto: **Accuratezza...** Perché?
  - Fornisce una misura immediata e comprensibile delle prestazioni.
  - Affidabile in contesti con classi bilanciate (recensioni autentiche vs false).
- **Metodologia:** Impiego del metodo «predict» su ogni istanza del Test set e comparazione dei modelli basata sull'accuratezza media (percentuale di predizioni corrette).

# Confronto Accuratezza Modelli



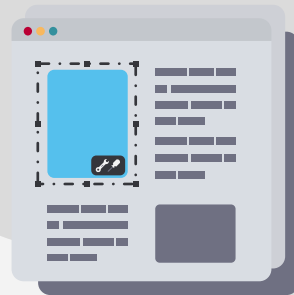
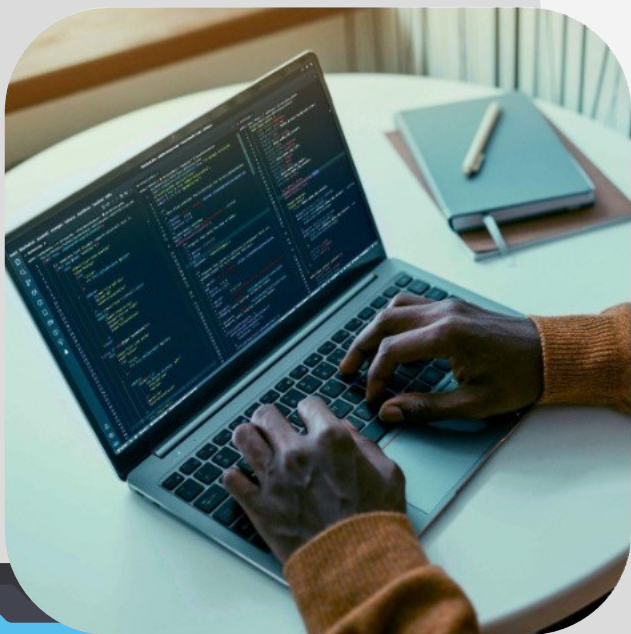
# Modello Migliore

1. SUPPORT VECTOR CLASSIFIER

2. LOGISTIC REGRESSION

3. MULTINOMIAL NAÏVE BAYES





# 08

## Implementazione bot

# Implementazione



## Sviluppo in Telegram

Piattaforma di messaggistica che offre la possibilità di creare un proprio bot attraverso un altro bot «BotFather» che permette di creare, gestire e personalizzare i propri bot.

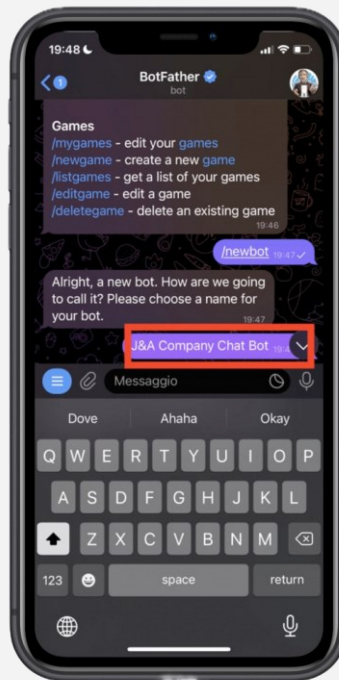
## TeleBot

Libreria Python che facilita lo sviluppo di bot per Telegram. Con essa, è possibile gestire le richieste degli utenti, inviare messaggi, rispondere a comandi e interagire con gli utenti attraverso il proprio bot.

# BotFather



- Si digita il comando /newbot
- Si inserisce il nome che si desidera del bot
- Se non è già utilizzato, verrà creato il bot e restituito un token, fondamentale per l'autenticazione delle richieste API al bot



# Sviluppo in Dettaglio



## Comando Start

```
#Function that handles the response to the start command
@bot.message_handler(commands=["start"])
def start(message):
    bot.reply_to(message, "Hi! Welcome to FakeReviewDetector, send me a review to check it")
```

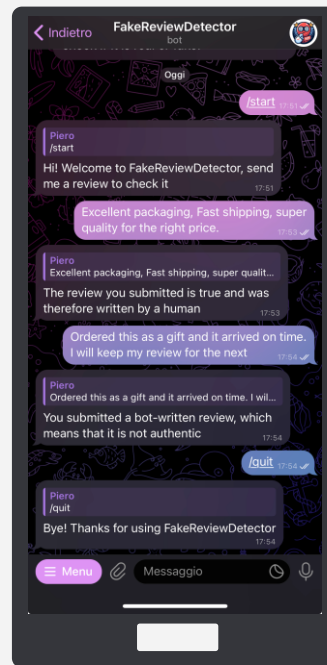
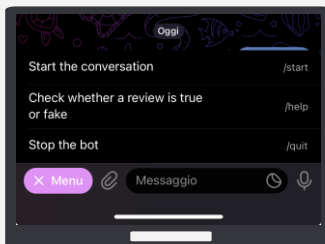
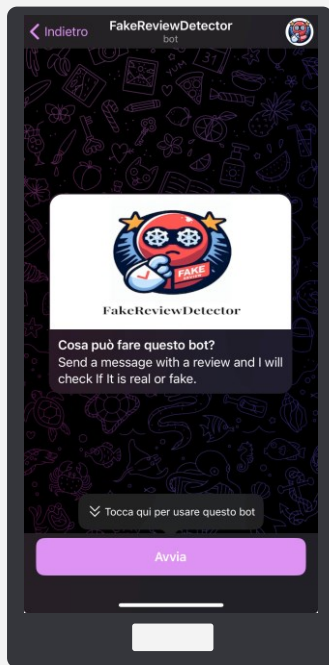
## Funzione per gestire la risposta alla recensione

```
#Function that handles the review sent in the form of a message
@bot.message_handler()
def message(message):
    recensione = message.text
    prediction = isReviewReal(recensione)
    if prediction == True:
        bot.reply_to(message, "The review you submitted is true and was therefore written by a human")
    else:
        bot.reply_to(message, "You submitted a bot-written review, which means that it is not authentic")
```





# FakeReviewDetector in Dettaglio



# Conclusioni 🧡



- Il modello da noi implementato, grazie all'impiego di tecniche di NLP e Machine Learning si è dimostrato abile nel distinguere accuratamente recensioni autentiche da quelle create artificialmente, esso si propone quindi, come un mezzo per filtrare le recensioni e garantire una valutazione basata esclusivamente su esperienze autentiche.
- Il modello può essere integrato come funzionalità all'interno di un e-commerce.
- Avendolo implementato in Telegram, il bot può anche essere utilizzato da chiunque e su qualsiasi dispositivo.

# Sviluppi Futuri

## **Addestramento su ulteriori Features del Dataset**

Introdurre la possibilità di addestrare il modello su più caratteristiche del dataset, come il rating e la categoria del prodotto.

## **Integrazione del modello all'interno di un e-commerce**

Il modello può essere integrato come funzionalità all'interno di un e-commerce. Questo garantirà l'identificazione della veridicità di una recensione, rendendo le recensioni un mezzo affidabile per scegliere se acquistare un prodotto o meno.

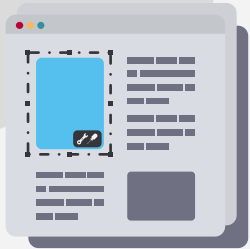
## **Affinamento delle Tecniche NLP**

Continuare a raffinare i modelli di NLP per migliorare la capacità di comprendere il linguaggio umano in modo più profondo.

## **Ospitare il Bot su una piattaforma di hosting**

Questo approccio garantirà la continuità operativa assicurando l'esecuzione continua del servizio e consentendo agli utenti di accedervi in qualsiasi momento.





**Grazie per  
l'attenzione!**