



# Web conference tools

Progetto Web Data Analytics

*Ilaria Olivares, Davide Giampaolo, Ludovico Lanzo*

# Obiettivi

---

- Creazione di un dizionario dei termini più utilizzati dai professionisti del lavoro per definire le caratteristiche più vantaggiose e quelle meno ottimali, delle piattaforme di web conference.
- Sulla base dei differenti settori in cui operano i reviewer, costruire un modello che suggerisce la piattaforma più adeguata in base al settore lavorativo di provenienza.
- Individuare i termini più importanti per descrivere i vantaggi di una piattaforma classificando i risultati in base al settore lavorativo dei reviewer

# Fonti e struttura dei dati

La fonte di riferimento per gli scraper è stata il sito web Capterra.com che colleziona migliaia di recensioni effettuate dagli utenti di LinkedIn su svariate tipologie di software e strumenti di comunicazione.

<https://www.capterra.com/>

Capterra organizza le recensioni in modo da ottenere sia dati qualitativi che quantitativi. Inoltre, è stato possibile ricavare per ciascun utente, la posizione lavorativa, il settore di riferimento, il numero di dipendenti dell'organizzazione per cui lavora, il tempo di utilizzo del software e la data della recensione.




Verified Reviewer 

Information Technology and Services, 10,001+ employees

Used the software for: 2+ years

Overall Rating	★★★★★	5/5
Ease of Use	★★★★★	5/5
Customer Service	★★★★★	5/5
Features	★★★★★	5/5
Value for Money	★★★★★	5/5
Likelihood to Recommend	<div><div></div></div>	10/10

Reviewer Source 

Source: Capterra

November 20, 2020

## "Asana is my favorite Kanban Product"

**Overall:** I personally have had a dandy time with the product and would use it over Trello any day of the week. I think that the team has a great dev schedule and constantly improve the product and entire platform, clean up tech debt and make older services faster. The API docs can use some work but it is not a huge deal once you get used to them. I would tell anyone looking for a pro-featured Kanban setup to use Asana time and time again. Hands down my favorite and I have used a ton of them.

**Pros:** Asana gives you the tools to customize and even a great API to leverage data from Asana into your own tools. Nothing can beat that. Yes it costs money, but you are getting a tool set that is worth the price. If you are looking for something cheap, this is not for you because you would not use all the features anyway.

**Cons:** Asana has a learning curve but has good docs to support you. The custom data aggregation or getting data OUT of the tool is hard, but again, can be done. I had issues that were solved by customer support. Just be ready for some additional time if you plan to write a custom app that uses the API

# Scraper e ETL



Scraping delle pagine dedicate ai top 6 meeting software per numero di recensioni.

Eliminazione caratteri speciali.

Struttura data uniformata a gg/mm/aaaa.

Uniformati i valori delle valutazioni in dati numerici.

# Preprocessing

Suddivisione del dataset in due parti: il primo comprende la colonna 'Vantaggi' e la colonna 'Consigliato', il secondo 'Svantaggi' e nuovamente 'Consigliato'.

La colonna 'Consigliato' indica quanto ciascun utente è propenso a consigliare un determinato tool in base alla sua valutazione complessiva. L'indicatore è compreso in una scala di valori da 0 a 10.

Per ciascuno dei due dataset è stata eseguita la polarizzazione dei testi, stabilendo il grado di positività per i testi con valore nel campo Consigliato maggiore di 7 e di negatività per i valori compresi tra 0 e 7.

```
df_raw_vantaggi['Consigliato'].value_counts()
```

```
10.0    13242
9.0      7382
8.0      6541
7.0      3451
6.0      1467
5.0      1204
4.0       330
3.0       293
2.0       199
0.0       147
1.0        71
```

```
Name: Consigliato, dtype: int64
```

Abbiamo deciso di impostare la soglia di polarità a 7 per avere un numero di record sufficientemente consistente, considerando che le valutazioni in generale sono sbilanciate verso valori positivi.

# Preprocessing

Considerando solo il dataset dei vantaggi con record positivi e quello degli svantaggi con i record polarizzati come negativi, abbiamo ricavato la lingua di ciascun testo attraverso la libreria **langdetect**. In modo da escludere dai due dataset definitivi i testi non in lingua inglese.

```
from langdetect import detect
lista_lingua = []
for i in range(0, len(df_01)):
    try:
        Lang = detect(df_01['Vantaggi'][i])
        lista_lingua.append(Lang)
        df_01['Lang'][i] = lista_lingua[i]
    except:
        pass
```

```
df_01.Lang.value_counts()
```

```
en    34376
es         12
pt         6
fr         4
it         3
ca         1
Name: Lang, dtype: int64
```

```
df_01 = df_01[~(df_01.Lang != "en")]
```

```
len(df_01)
```

```
34376
```

```
from langdetect import detect
lista_lingua = []
for i in range(0, len(df_01)):
    try:
        Lang = detect(df_01['Inconvenienti'][i])
        lista_lingua.append(Lang)
        df_01['Lang'][i] = lista_lingua[i]
    except:
        pass
```

```
df_01.Lang.value_counts()
```

```
en    34258
es         7
pt         5
it         4
fr         4
no         3
af         1
cy         1
Name: Lang, dtype: int64
```

```
df_01 = df_01[~(df_01.Lang != "en")]
```

```
len(df_01)
```

```
34258
```

# N\_Grams

---

● Per la fase di topic modeling abbiamo utilizzato il Phraser di gensim, creando in seguito i bigrammi e gli ngrammi dei termini. Il processo è stato ripetuto sia per i Vantaggi che per gli Svantaggi e ha previsto la sostituzione dei caratteri maiuscoli in minuscoli, l'eliminazione dei segni di interpunzione, punteggiatura e una serie di stopwords.

```
: from gensim.models.phrases import Phrases
stop_words = pd.read_csv('stop_list.txt', sep=" ", header=None)
stop = stop_words[0].tolist()
bigram = Phrases(text, min_count=5, threshold=0.5, common_terms=stop)
bigrams = [bigram[item] for item in text]
ngrams = [bigram[item] for item in bigrams]
nngrams = [bigram[item] for item in ngrams]
print(ngrams[0])
train_sentences = []
for row in ngrams:
    train_sentences.append(' '.join([item for item in row if item not in stop]))
df_01['Vantaggi'] = train_sentences
```

# Lemmatizzazione

---

- La Lemmatizzazione è stata eseguita mediante WordNetLemmatizer, dal corpus sono state rimosse le stopwords e i caratteri numerici.

```
corpus = df_01['Vantaggi']
corpus = corpus.str.split()
corpus = corpus.apply(lambda x: [lemmatizer.lemmatize(item) for item in x if item not in stop])
corpus = corpus.apply(lambda x: [item for item in x if not item.isnumeric()])
corpus
```

```
0      [tried, solution, solution, access, conference...
1      [integration_with_sharepoint, office_365, awes...
2      [setup, view, point, admin, setting, end, invi...
3      [implement, onboard, new, interface, attractiv...
4      [review, title, confirm, conference, support, ...
...
32600      [operate, right, call, right_away, clarity]
32601      [least, google_hangout, sometimes, receive, no...
32602      [communication, piece, find, communicate, whet...
32603      [pretty, remote_access, another, help, youre_t...
32604      [customer, talk, via, prior, experience, intui...
Name: Vantaggi, Length: 32605, dtype: object
```



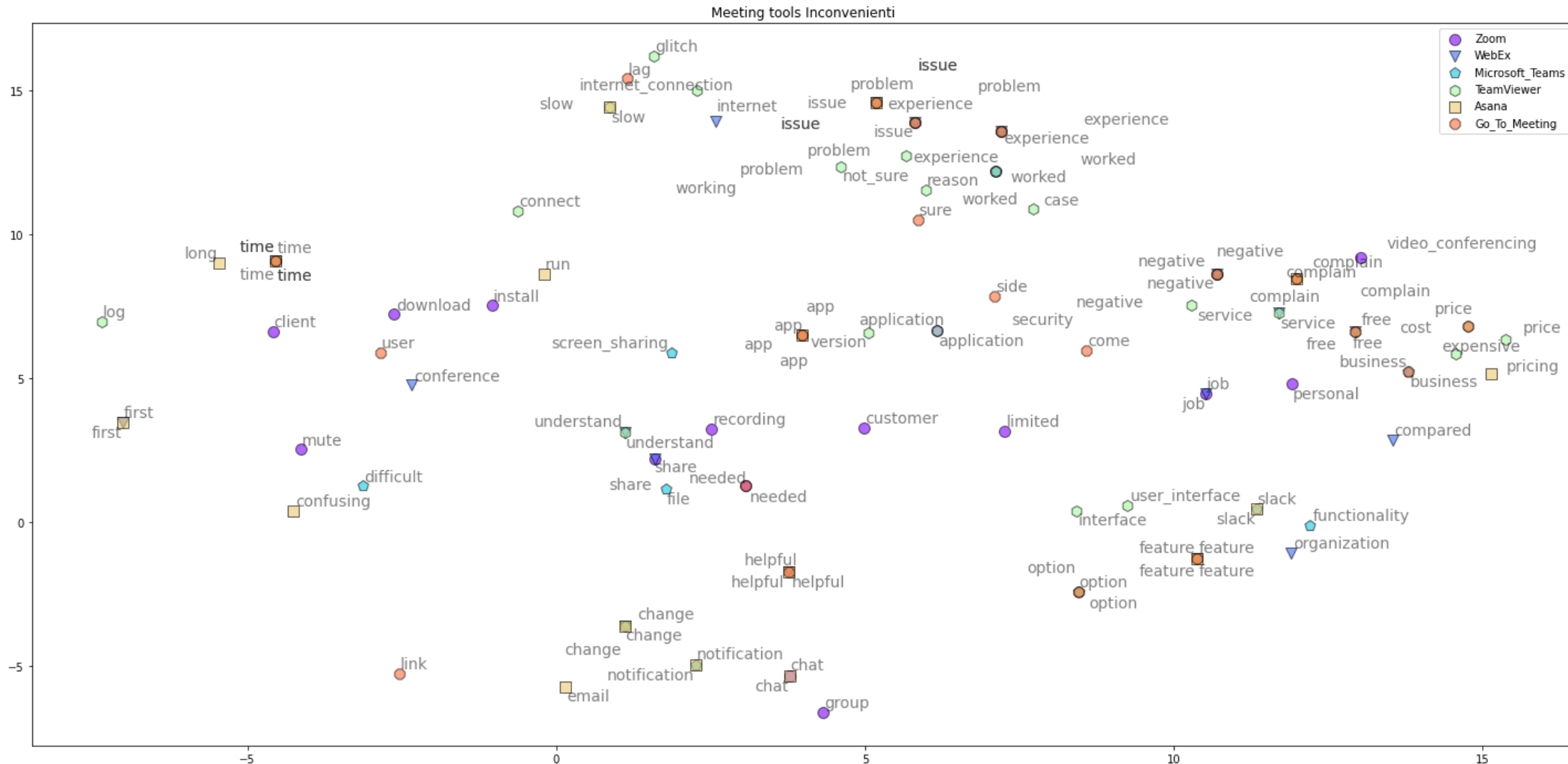
# Word Embedding

● Per la parte di Word Embedding abbiamo identificato il modello migliore attraverso alcuni tentativi di utilizzo del Word2Vec e del Fast Text, testati entrambi sui modelli Cbow e Skipgram. Sulla base dei risultati ottenuti, è stato individuato il modello migliore.

```
from gensim.models import FastText
%time ft_model = FastText(sg=0, sentences=corpus, size=100, window=5, min_count=10, iter=20, min_n=3, max_n=10)
%time ft_model_2 = FastText(sg=0, sentences=corpus, size=100, window=5, min_count=20, iter=20, min_n=3, max_n=9)
%time ft_model_3 = FastText(sg=1, sentences=corpus, size=100, window=5, min_count=10, iter=20, min_n=3, max_n=9)
%time ft_model_4 = FastText(sg=1, sentences=corpus, size=100, window=5, min_count=20, iter=20, min_n=3, max_n=10)
%time ft_model_5 = FastText(sg=0, sentences=corpus, size=150, window=5, min_count=10, iter=50, min_n=3, max_n=10)
%time ft_model_6 = FastText(sg=0, sentences=corpus, size=150, window=5, min_count=10, iter=50, min_n=3, max_n=8)
%time ft_model_7 = FastText(sg=1, sentences=corpus, size=200, window=5, min_count=10, iter=50, min_n=3, max_n=7)
%time ft_model_8 = FastText(sg=1, sentences=corpus, size=200, window=5, min_count=20, iter=50, min_n=3, max_n=8)
%time ft_model_9 = FastText(sg=1, sentences=corpus, size=100, window=5, min_count=10, iter=50, min_n=3, max_n=8)
%time ft_model_10 = FastText(sg=1, sentences=corpus, size=200, window=5, min_count=20, iter=50, min_n=3, max_n=9)
%time ft_model_11 = FastText(sg=1, sentences=corpus, size=100, window=5, min_count=10, iter=50, min_n=3, max_n=10)
%time ft_model_12 = FastText(sg=1, sentences=corpus, size=100, window=5, min_count=15, iter=50, min_n=3, max_n=9)
%time ft_model_13 = FastText(sg=0, sentences=corpus, size=100, window=5, min_count=15, iter=50, min_n=3, max_n=9, word_ngrams=2)
%time ft_model_14 = FastText(sg=0, sentences=corpus, size=100, window=5, min_count=20, iter=50, min_n=3, max_n=9, word_ngrams=3)
%time ft_model_15 = FastText(sg=0, sentences=corpus, size=100, window=5, min_count=20, iter=50, min_n=3, max_n=9, word_ngrams=4)
%time ft_model_16 = FastText(sg=0, sentences=corpus, size=100, window=5, min_count=20, iter=50, min_n=3, max_n=10, word_ngrams=4)
%time ft_model_17 = FastText(sg=0, sentences=corpus, size=150, window=5, min_count=20, iter=50, min_n=3, max_n=10, word_ngrams=4)
```

# Umap top words 'Piattaforme – Svantaggi'

- Plot dei principali termini degli ‘Svantaggi’ che più si associano alle diverse piattaforme in base alla loro cosin similarity.



# Umap top words 'Piattaforme – Vantaggi'



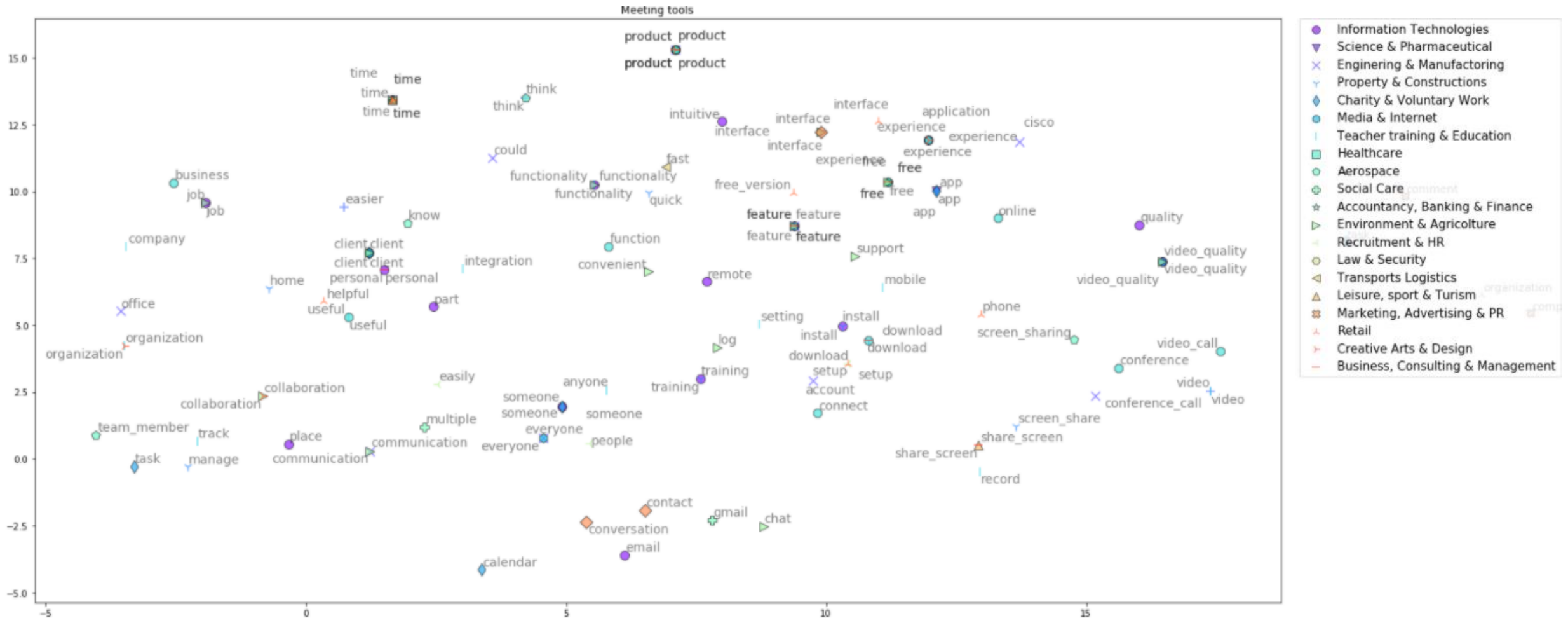
# Labeling dei settori lavorativi

- Una delle informazioni ricavate dagli scraper riguarda il settore lavorativo di provenienza del reviewer. Partendo da questi dati abbiamo eseguito una classificazione dei 294 settori e una traduzione di quelli rimasti in italiano.
- Le categorie iniziali sono quindi state ridotte a 25 macro categorie



# Umap top words 'Settore – Vantaggi'

- In questo caso abbiamo utilizzato le categorie ricavate dal settore lavorativo per individuare i termini più importanti utilizzati dai diversi reviewer all'interno dei 'Vantaggi'.



# Sviluppi Futuri

---

- Analizzare come sono cambiati i termini per descrivere le piattaforme in un arco temporale a cavallo del periodo covid, per individuare eventuali miglioramenti sulle piattaforme.
- Valutare per profili lavorativi i termini che meglio descrivono i vantaggi e gli svantaggi di ciascuna piattaforma. In modo da poter replicare il modello agli altri software utilizzati nelle aziende.
- Creare un sistema di raccomandazione delle piattaforme aggiornato costantemente sulla base delle recensioni.

**Grazie**

---