



Feature Engineering

Let's engineer some features for the Tweet ranking model.

We'll cover the following



- Features for the model
 - Dense features
 - User-author features
 - User-author historical interactions
 - User-author similarity
 - Author features
 - Author's degree of influence
 - Historical trend of interactions on the author's Tweets
- User-Tweet features
- Tweet features
 - Features based on Tweet's content
 - Features based on Tweet's interaction
 - Separate features for different engagements
- Context-based features
- Sparse features

The machine learning model is required to predict user engagement on user A's Twitter feed. Let's engineer features to help the model make informed predictions.



The feature set shown here is the result of one brainstorm session.

However, feature engineering is an iterative process. As an exercise,

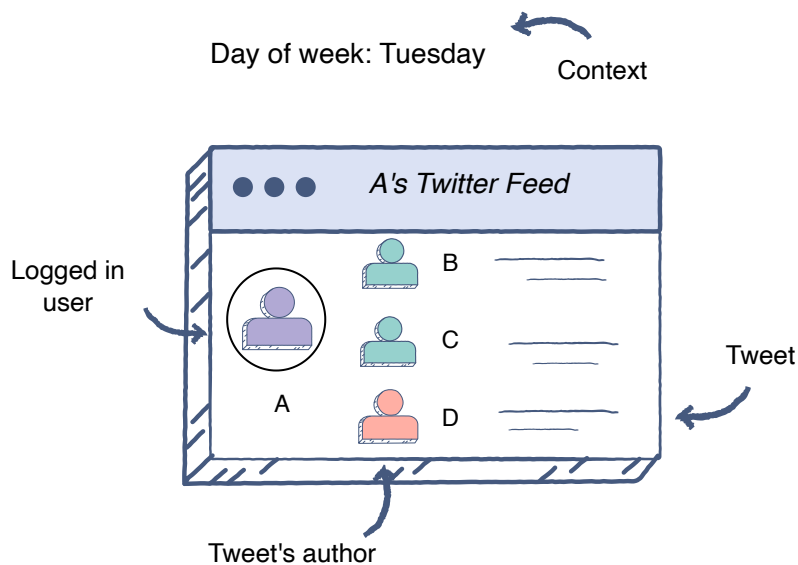


you are encouraged to think about more features.



Let's begin by identifying the four main **actors** in a twitter feed:

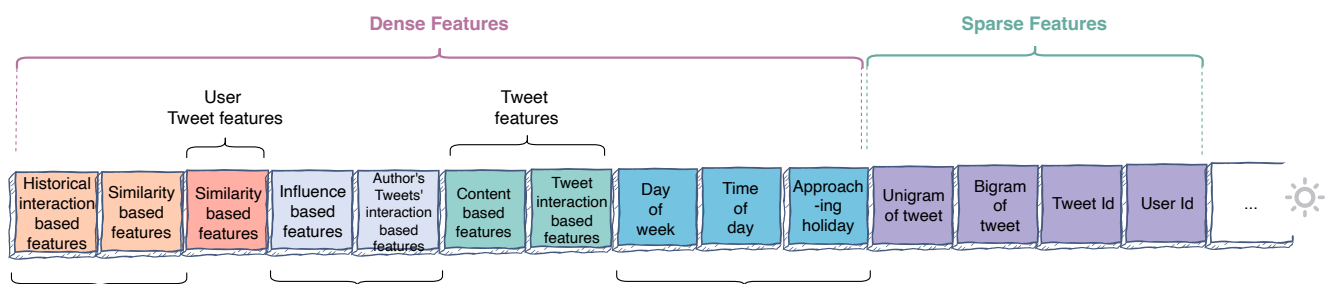
1. The logged-in user
2. The Tweet
3. Tweet's author
4. The context



Actors in a Twitter feed

Features for the model#

Now it's time to generate features based on these actors and their interactions. A subset of the features is shown below.





Features in the training data row

Let's discuss these features one by one.

Dense features#

We will start by discussing the dense features.

User-author features#

These features are based on the logged-in user and the Tweet's author. They will capture the *social relationship* between the user and the author of the Tweet, which is an extremely important factor in ranking the author's Tweets. For example, if a Tweet is authored by a close friend, family member, or someone that user is highly influenced by, there is a high chance that the user would want to interact with the Tweet.

How can you capture this *relationship* in your signals given users are not going to specify them explicitly? Following are a few features that will effectively capture this.

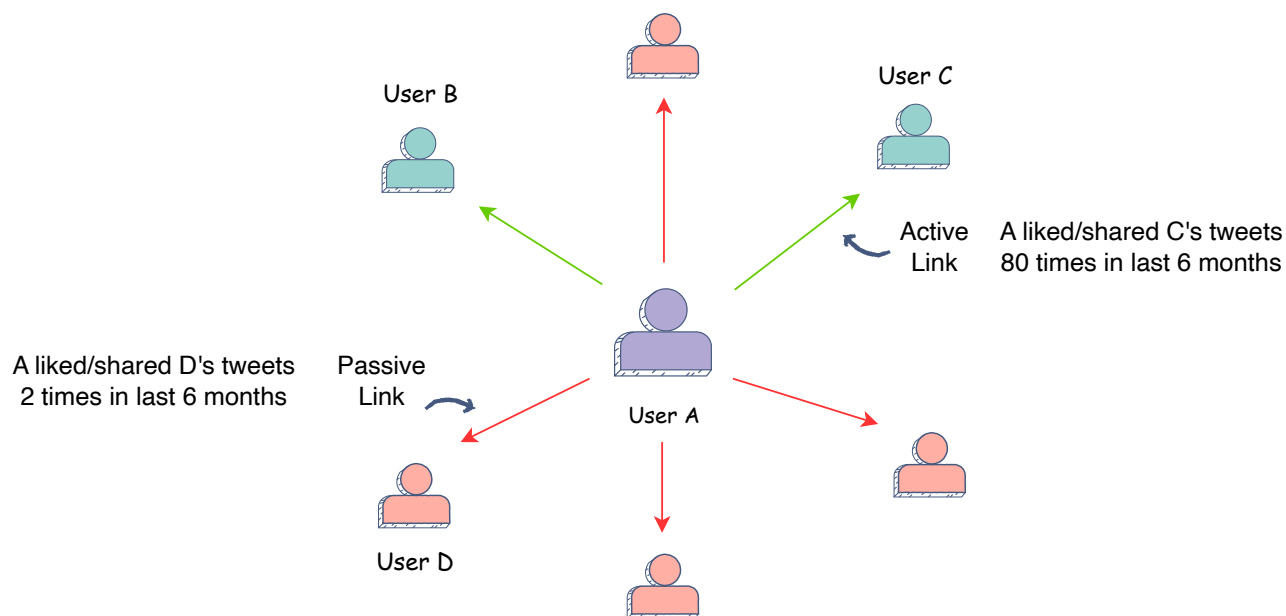
User-author historical interactions#

When judging the relevance of a Tweet for a user, the relationship between the user and the Tweet's author plays an important role. It is highly likely that if the user has actively engaged with a followee in the past, they would be more interested to see a post by that person on their feed.





A will be more interested in B's and C's tweet rather than in D's tweets



User A has interacted more with User B and User C

Few features based on the above concept can be:

- **author_liked_posts_3months**

This considers the percentage of an author's Tweets that are liked by the user in the last three months. For example, if the author created twelve posts in the last three months and the user interacted with six of these posts then the feature's value will be:

$$\frac{6}{12} = 0.5 \text{ or } 50\%$$

This feature shows a more recent trend in the relationship between the user and the author.

- **author_liked_posts_count_1year**

This considers the number of an author's Tweets that the user interacted with, in the last year. This feature shows a more long term trend in the relationship between the user and the author.



📝 Ideally, we should normalize the above features by the total number of Tweets that the user interacted with during the same periods. This enables the model to see the real picture by cancelling out the effect of a user's general interaction habits. For instance, let's say user A generally tends to interact (e.g., like or comment) more while user B does not. Now, both user A and B have a hundred interactions on user C's posts. User B's interaction is more significant since they generally interact less. On the other hand, user A's interaction is mostly a result of their tendency to interact more.

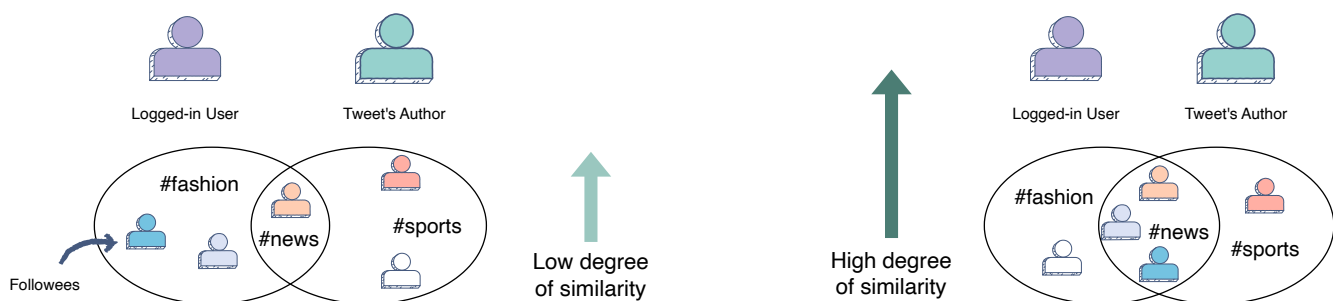


User-author similarity#

Another immensely important feature set to predict user engagement focuses on figuring out how similar the logged-in user and the Tweet's author are. A few ways to compute such features include:

- **common_followees**

This is a simple feature that can show the similarity between the user and the author. For a user-author pair, you will look at the number of users and hashtags that are followed by *both* the user and the author.



Similarity between logged-in user and Tweet's author

- **topic_similarity**

The user and author similarity can also be judged based on their interests. You can see if they interact with similar topics/hashtags. A






simple method to check this is the TF-IDF based similarity between the hashtags:

- followed by the logged-in user and author
- present in the posts that the logged-in user and author have interacted with in the past
- used by the author and logged-in user in their posts

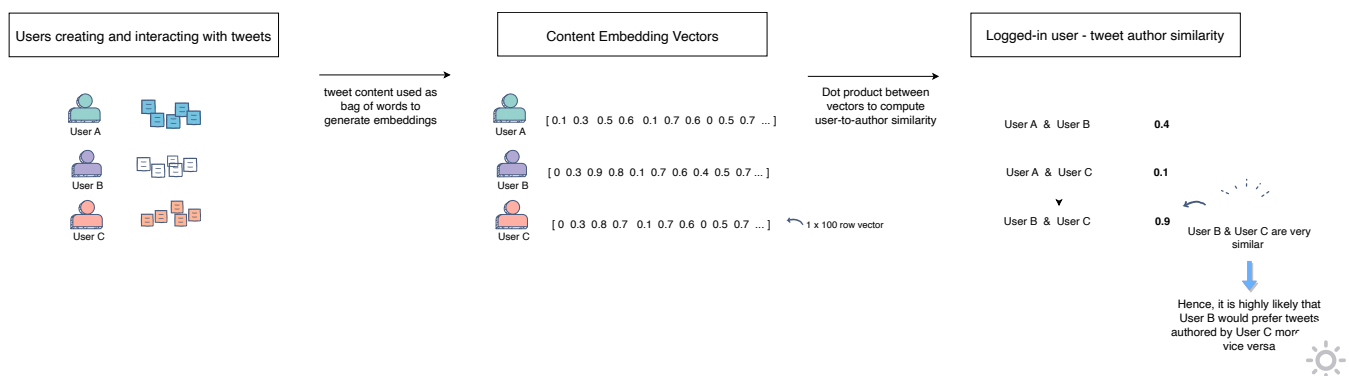
The similarity between their search histories on Twitter can also be used for the same purpose.

• **tweet_content_embedding_similarity**

A user is represented by the content that they have generated and interacted with in the past. You can utilize all of that content as a bag-of-words and build an embedding for every user. With an embedding vector for each user, the dot product between them can be used as a fairly good estimate of user-to-author similarity.

 Embedding helps to reduce the sparsity of vectors generated otherwise.

Have a look at the [lesson](#) on embeddings to get an idea on how the embedding can be learned for this scenario.



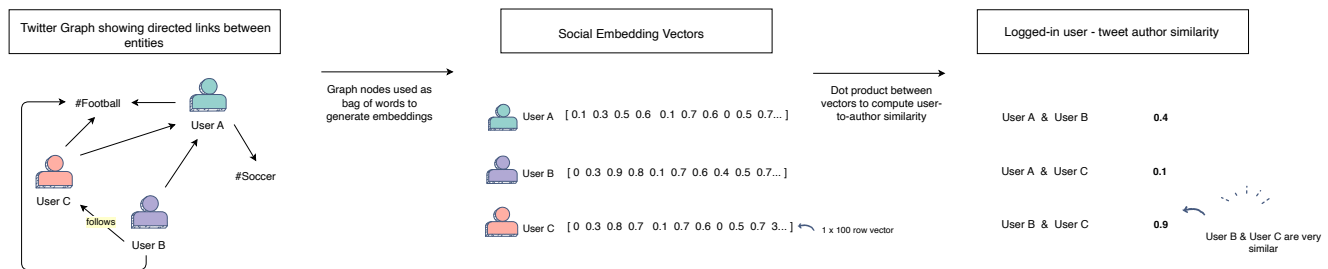
Content embedding similarity between logged-in user and Tweet's author



User B and C are very similar. Hence, it is highly likely that user B will prefer Tweets authored by user C more and vice versa.

- **social_embedding_similarity**

Another way to capture the similarity between the user and the author is to generate embeddings based on the social graph rather than based on the content of Tweets, as we discussed earlier. The basic notion is that people who follow the same or similar topics or influencers are more likely to engage with each other's content. A basic way to train this model is to represent each user with all the other users and topics (user and topic ids) that they follow in the social graph. Essentially, every user is represented by bag-of-ids (rather than bag-of-words), and you use that to train an embedding model. The user-author similarity will then be computed between their social embeddings and used as a signal.



Social similarity between user and author



Have a look at the [lesson](#) on embeddings to get an idea on how the embedding can be learned for this scenario.

Author features#

These features are based on Tweet's author.



Author’s degree of influence#

A Tweet written by a more *influential* author may be more relevant. There are several ways to measure the author’s influence. A few of these ways are shown as features for the model, below.

- **is_verified**

If an author is verified, it is highly likely that they are somebody important, and they have influence over people.




- **author_social_rank**

The idea of the author’s social_rank is similar to [Google’s page rank](#). To compute the social rank of each user, you can do a [random walk](#) (like in page rank). Each person who follows the author contributes to their rank. However, the contribution of each user is not equal. For example, a user adds more weight if they are followed by a popular celebrity or a verified user.

- **author_num_followers**

One important feature could be the number of followers the author has. Different inferences can be drawn from different follower counts, as shown below:

Follower count	Inference
150	Personal account with reasonable influence over their social circle
1 thousand	Social media influencer with a good amount of influence over fans

Follower count	Inference   
3 million	Celebrity with a great fan following

Normalised author_num_followers

You can observe a lot of variation in the follower counts of Twitter users. To bring each user's follower count in a specific range, let's say zero to ten-thousand, you can divide their follower count by the maximum observed follower count (across the platform) and then multiply it with ten-thousand.

- **follower_to_following_ratio**

When coupled with the number of followers, the follower to following ratio can provide significant insight, regarding:




1. The type of user account
2. The account's influence
2. The quality of the account's content (Tweets)

Let's see how.

Follower to Following Ratio	Follower Count	Inference
-----------------------------	----------------	-----------



Follower to Following Ratio	Follower Count	<div><div><div><div>?</div></div><div>Inference</div></div><div><div></div><div></div></div><div><div></div><div></div></div></div>
<div><div>≈ 0</div><div>0.5</div></div>	<div><div>< 500</div><div>≈ 3 thousand</div></div>	<div><div><div>The user has a personal account, mainly following brands and celebrities. The account has average content quality, and its influence is limited to the user’s friends and family.</div><div>The user is a small influencer with low-quality content. They may be using the follow/unfollow method to gain users, i.e., they follow people to receive a follow-back. Once they do, they unfollow them. People may unfollow once they figure out what is going on.</div></div><div></div></div>

Follower to Following Ratio	Follower Count	Inference   
10+	>= 15 thousand	The user is likely to be a micro-celebrity with good quality content. He/she have a wider influence, not limited to their social circle only.
370370	≈ 30 million	The user is Elon Musk!

Historical trend of interactions on the author's Tweets#

Another very important set of features is the interaction history on the author's Tweets. If historically, an author's Tweets garnered a lot of attention, then it is highly probable that this will happen in the future, too.

 A high rate of historical interaction for a user implies that the user posts high-quality content.

Some features to capture the historical trend of interactions for the author's Tweets are as follows:

- **author_engagement_rate_3months**





The *engagement rate* of the historical Tweets by the author can be a great indicator of future Tweet engagement. To compute the engagement rate in the last three months, we look at how many times different users interacted with the author's Tweets that they viewed.


Engagement rate: $\frac{\text{Tweet-interactions}}{\text{Tweet-views}}$

- **author_topic_engagement_rate_3months**

The engagement rate can be different based on the Tweet's topic. For example, if the author is a sports celebrity, the engagement rate for their family-related Tweets may be different from the engagement rate for their sports-related Tweets.

We can capture this difference by computing the engagement rate for the author's Tweets per topic. Tweet topics can be identified in the following two ways:

1. Deduce the Tweet topic by the hashtags used
2. Predict the Tweet topic based on its content

 These features don't necessarily have to be based on data from three months, but rather you should utilize different time ranges, e.g., historical engagement rates in last week, month, three months, six months, etc.

User-Tweet features#

The similarity between the user's interests and the tweet's topic is also a good indicator of the relevance of a Tweet. For instance, if user A is interested in football, a Tweet regarding football would be very relevant to them.



- **topic_similarity**



You can use the hashtags and/or the content of the Tweets that the user has either Tweeted or interacted with, in the last six months and compute the TF-IDF similarity with the Tweet itself. This indicates whether the Tweet is based on a topic that the user is interested in.

- **embedding_similarity**

Another option to find the similarity between the user's interest and the Tweet's content is to generate embeddings for the user and the Tweet. The Tweet's embedding can be made based on the content and hashtags in it. While the user's embedding can be made based on the content and hashtags in the Tweets that they have written or interacted with. A dot product between these embeddings can be calculated to measure their similarity. A high score would equate to a highly relevant Tweet for a user.

Tweet features#

These features are based on the Tweet itself.

Features based on Tweet's content#

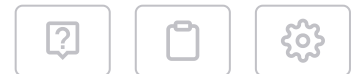
- **Tweet_length**

The length of the Tweet positively correlates with user engagement, especially likes and reshares.

It is generally observed that people have a short attention span and prefer a shorter read. Hence, a more concise Tweet generally increases the chance of getting a like by the user. Also, we know that Twitter restricts the length of the Tweet. So, if a person wants to Retweet a Tweet that has nearly used up the word limit, the person would not be able to add their thoughts on it, which might be off-putting.



- **tweet_recency**



The recency of the Tweet is an important factor in determining user engagement, as people are most interested in the latest developments.

- **is_image_video**

The presence of an image or video makes the tweet more catchy and increases the chances of user engagement.

- **is_URL**

The presence of a URL may mean that the Tweet:

1. Calls for action
2. Provides valuable information

Hence, such a Tweet might have a higher probability of user engagement.

Features based on Tweet's interaction#

You should also utilize the Tweet's interactions as features for our model. Tweets with a greater volume of interaction have a higher probability of engaging the user. For instance, a Tweet with more likes and comments is more relevant to the user, and there is a good chance that the user will like or comment on it too.

- **num_total_interactions**

The total number of interactions (likes, comments and reshares) on a Tweet can be used as a feature.


Caveat

Simply using these interactions as features might give an incomplete picture. Consider a scenario where Bill Gates Tweeted a month ago, and his Tweet received five million engagements over this period. Whereas, another Tweet posted just an hour ago has two-thousand engagements and has become a trending Tweet. Now, if your feature is based on the interaction count only Bill Gate's Tweet would be considered more



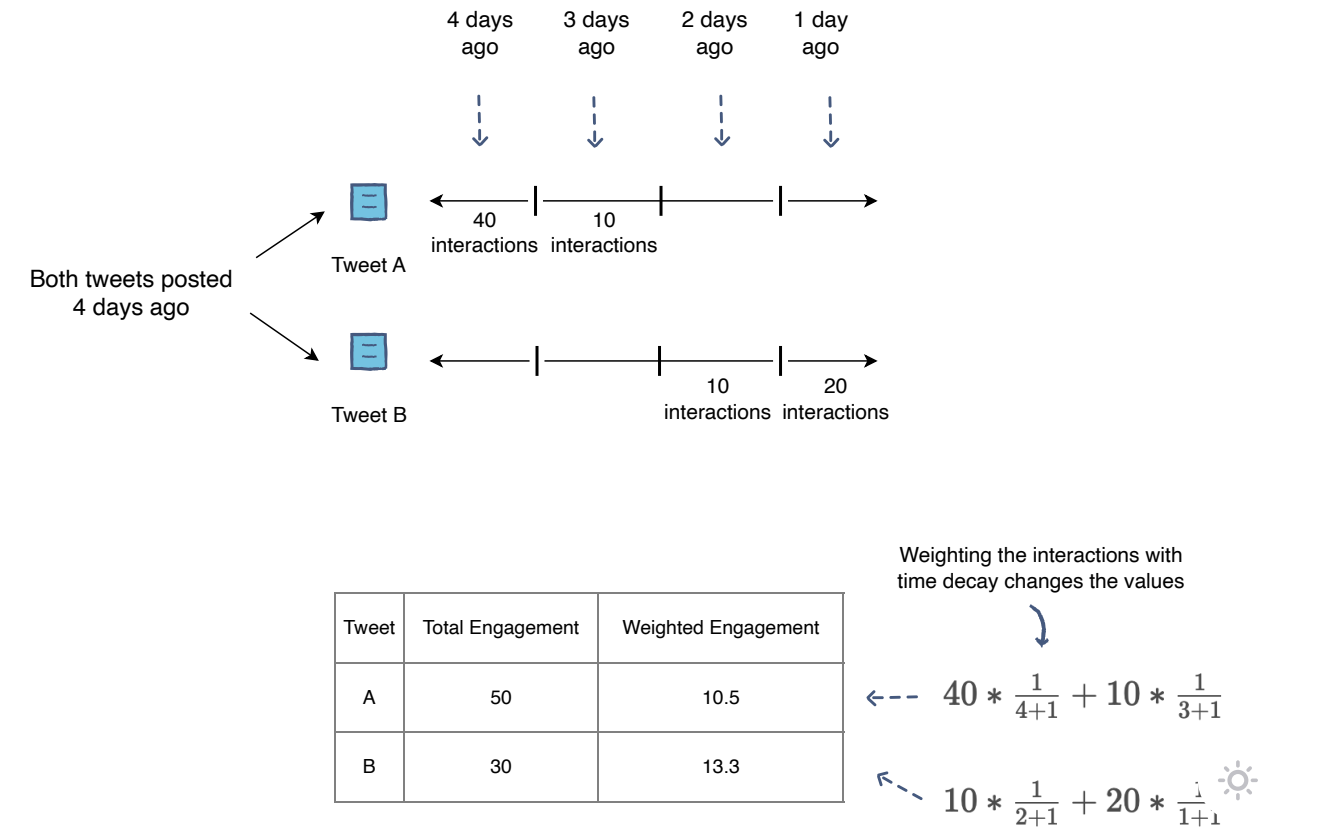
relevant, and the quick rise in the popularity of the trending Tweet would not be given its due importance.

latest interaction more than the ones that happened some time ago.

 Time decay can be used in all features where there is a decline in the value of a quantity over time.

One simple model can be to weight every interaction (*like, comment, and retweet*) by $\frac{1}{t+1}$ where *t* is the number of days from current time.

In the above scenario, you saw two Tweets, Tweeted with a lot of time difference. The same scenario can also happen for two Tweets tweeted at the same time, as shown below. Let's see how you can use time decay to see the real value of engagement on these Tweets.

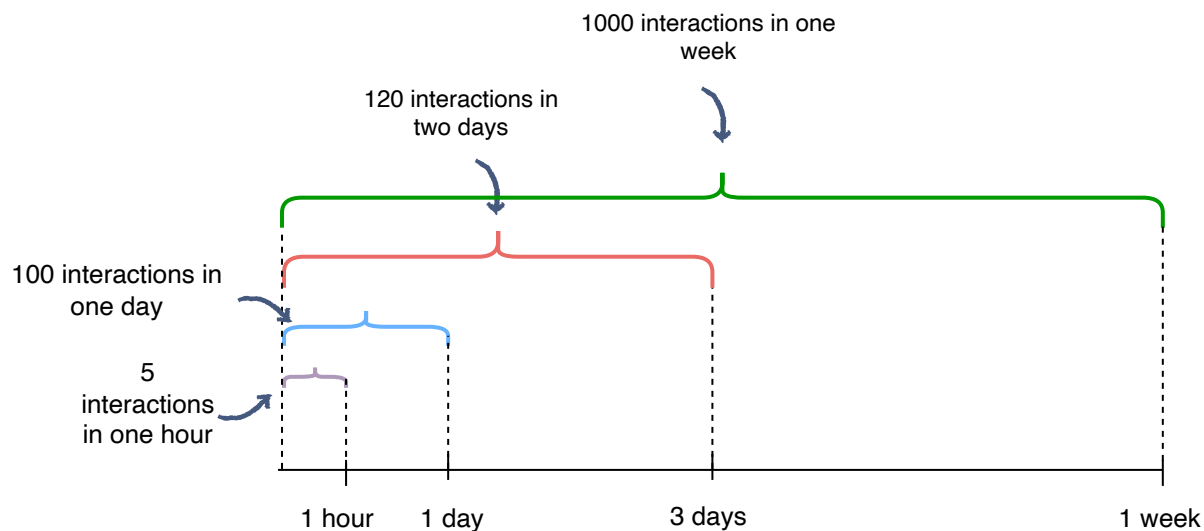




Tweet A had a total of fifty interactions, while Tweet B had a total of thirty. However, the interactions on Tweet B were more recent. So, by using time decay, the weighted number of likes on Tweet B became greater than those for Tweet A.

Another remedy is to use different **time windows** to capture the recency of interactions while looking at their numbers. The interaction in each window can be used as a feature:

- **interactions_in_last_1_hour**
- **interactions_in_last_1_day**
- **interactions_in_last_3_days**
- **interactions_in_last_week**



Different overlapping time windows to capture interactions on tweets

Separate features for different engagements#

Previously, we discussed combining all interactions. You can also keep them as separate features, given you can predict different events, e.g., the probability of *likes*, *Retweets* and *comments*. Some potential features can be



- **likes_in_last_3_days**

- **comments_in_last_1_day**



- **reshares_in_last_2_hours**

The above three features are looking at their respective forms of interactions from all twitter users. Another set of features can be generated by looking at the interactions on the Tweet made only by user A's network. The intuition behind doing this is that there is a high probability that if a Tweet has more interactions from A's network, then A, having similar tastes, will also interact with that Tweet.

The set of features based on user's network's interactions would then be:

- **likes_in_last_3_days_user's_network_only**
- **comments_in_last_1_day_user's_network_only**
- **reshares_in_last_2_hours_user's_network_only**


Context-based features#

These features are based on the context.

- **day_of_week**

The day of the week can affect the type of Tweet content a user would like to see on their feed.

- **time_of_day**

Noting the time of the day (coupled with the day of the week) can provide useful information. For example, if a user logs-in on a Monday morning, it is likely that they are at their place of work. Now, it would make sense to show shorter Tweets that they can quickly read. In contrast, if they login in the evening, they are probably at their home and would have more time at their disposal. Therefore, you can show them longer Tweets with video content as well, since the sound from t⁷  video would not be bothersome in the home environment.

- **current_user_location**



The current user location can help us show relevant content. For example, a person may go to San Francisco where a festival is happening (it is the talk of the town). Based on what is popular in a certain area where the user is located, you can show relevant Tweets.

- **season**

User's viewing preference for different Tweet topics may be patterned according to the four seasons of the year.

- **latest_k_tag_interactions**

You can see the latest “k” tags included in the Tweets a user has interacted with to gain valuable insights. For example, assume that the last $k = 5$ Tweets a user interacted with contain the tag “Solar eclipse”. From this, we can discern that the user is most likely interested in seeing Tweets centered around the solar eclipse.

- **approaching_holiday**

Recording the approaching holiday will allow the model to start showing the user more content centred around that holiday. For instance, if Independence Day is approaching, Tweets with more patriotic content would be displayed.

Until now, all the features that we discussed have been dense features.

Sparse features#

For some ML models, sparse features can be useful. Following are a few of the sparse features that would prove to be helpful when predicting engagement.

- *unigrams/bigrams of a Tweet*

The presence of certain unigrams or bigrams may increase the probability of engagement for a tweet. For example, during data



probability of engagement for a tweet. For example, during data visualisation, you may observe that Tweets with the big



now” have higher user engagement. The reason behind this might be that such Tweets are useful and informative for users as they redirect them towards websites where they can purchase things according to their needs.

- *user_id*

This is a very simple feature used to identify users with very high engagement rates, such as celebrities and influencers.

- *tweets_id*

This feature is used to identify Tweets that have high engagement and hence have a higher probability of engaging users.



Have a look at the embeddings [lesson](#) to find out how some of these sparse features can be utilized in our models, via embeddings.

[← Back](#)[Next →](#)

Tweet Selection

Training Data Generation



Mark as Completed

[Report an Issue](#)