For this project I will use data from Google Play reviews for the Spotify app. I will perform some exploratory data analysis, build a taxonomy and then apply it to draw insights from the dataset via manual categorisation.

The unprocessed dataset has the following columns:

- `Time_submitted` – timestamp for when review was posted
- `Review` – contains the main review text
- `Rating` – from 1–5 for Google Play store
- `Total_thumbsup` – how many people gave this review a thumbs up for being helpful
- `Reply` – contains text included in any reply from Spotify

Step 1: Data Cleaning

To start, I cleaned the data by:

- Checking for null or spam values using filters in the review column and out of bounds data in the ratings.
- Randomising the rows and reducing the file down to 250 rows of reviews to represent a sample that I can do manual work on.
- We don't need the timestamps so I removed that column.
- Set all text fields to lowercase and trimmed whitespace.
- Used REGEXREPLACE() to remove punctuation and emojis.

We are not removing stopwords as these are useful for preserving context.

Step 2: Exploratory Data Analysis

- I have created some new columns, one that holds the review word counts and another that sorts these into "buckets". These can be used to check for correlations between review length and ratings.
  - I created a pivot table to show the distribution of review lengths, finding that the vast majority of reviews are less than 50 words.

- I also wanted to see what kind of reviews people find most helpful. I expect negative reviews to have higher engagement than positive ones.
  - I checked this by creating a pivot table that holds the ratings against the sum and average number of thumbs up given to reviews with that rating.
  - I found that the opposite was true and that positive 5 star reviews had by far the most engagement, followed by 3 star reviews.

- I also created a rating distribution table that shows which ratings are most common. As expected, 1 and 5 star reviews come up the most as these represent strong feelings people are more likely to go onto the app store and share.

- For one final piece of EDA I created a new column with tokenised reviews and then created another column that recreates this with common stopwords removed.
    - I then used this to create another pivot table that shows the frequency of words appearing in reviews.
    - I filtered this to remove blanks and words that appear less than 3 times.
    - Sorting this by descending count values shows which words appear most frequently.
    - Given more time, my goal would be to map the counts of these words to the ratings that the reviews they appear within are linked to.

Step 3: Building a Taxonomy

This will have top-level themes and some sub-themes within them:

1. **App performance**
    a. Crashes
    b. Slow load times
    c. Problems opening the app
2. **Playback and audio**
    a. Sound quality
    b. Audio stopping/ skipping
    c. Connection/ offline mode issues
3. **Account and subscriptions**
    a. Billing issues
    b. Family plan problems
    c. Premium features not working
4. **UX**
    a. Hard to navigate the app
    b. Difficulties managing/ updating playlists
    c. Accessibility issues
5. **Search/ discovery pages**
    a. Irrelevant homepage
    b. Can’t find specific songs/ artists
    c. Bad recommendations
6. **User feelings**
    a. Like the app overall
    b. Requests for features
    c. Frustration/ disappointment

Using these, I manually categorised 100 reviews

Step 4: Insights

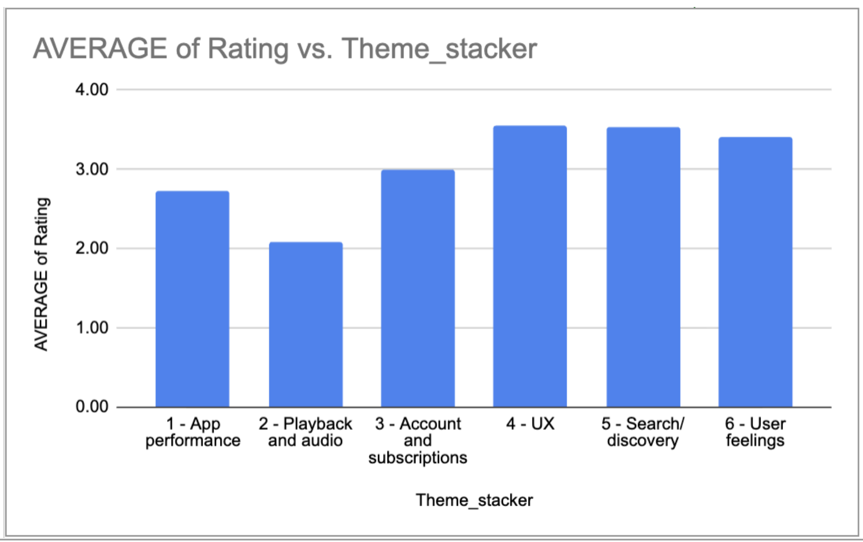
From applying this taxonomy I have been able to conclude some points from the data:

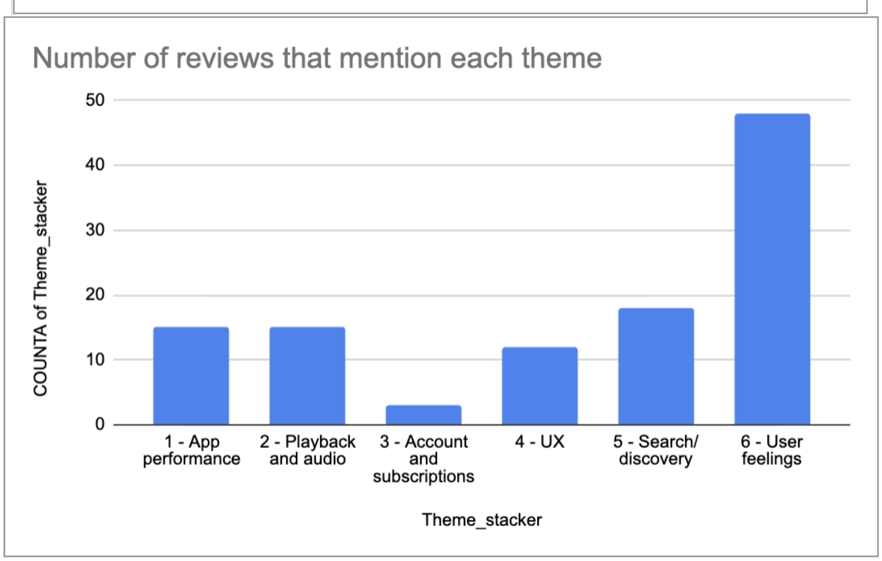
1. Reviews that mention audio stopping or cutting out have the lowest average rating.

2. Reviews that mention accessibility issues have surprisingly high average scores, this is likely due to the relatively small sample size and a strange trend I noticed in negative reviews coming with high star ratings.

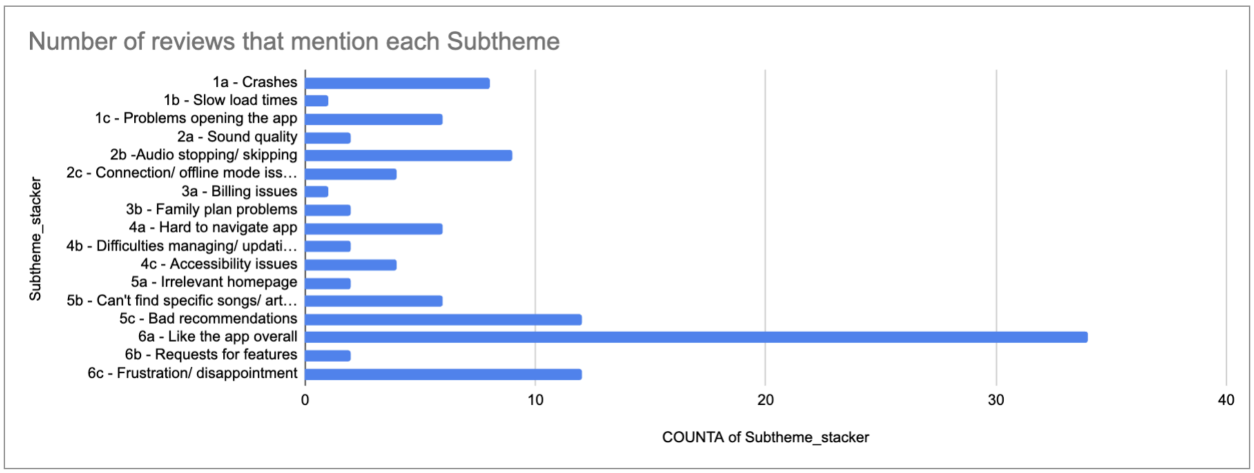
3. More than anything else, people talk about how they feel in reviews reporting both general frustration and overall satisfaction with the app very often.

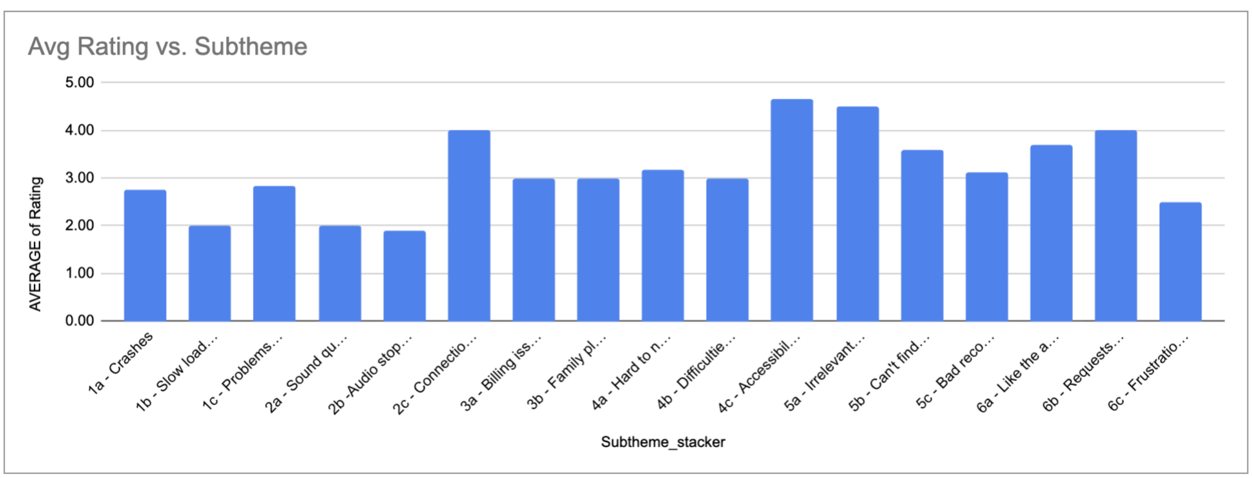
Theme_stacker	COUNTA of Theme_stacker	AVERAGE of Rating
1 - App performance	15	2.73
2 - Playback and audio	15	2.09
3 - Account and subscriptions	3	3.00
4 - UX	12	3.55
5 - Search/ discovery	18	3.53
6 - User feelings	48	3.40





Subtheme_stacker	COUNTA of Subthem	AVERAGE of Rating
1a - Crashes	8	2.75
1b - Slow load times	1	2.00
1c - Problems opening the app	6	2.83
2a - Sound quality	2	2.00
2b -Audio stopping/ skipping	9	1.89
2c - Connection/ offline mode issues	4	4.00
3a - Billing issues	1	3.00
3b - Family plan problems	2	3.00
4a - Hard to navigate app	6	3.17
4b - Difficulties managing/ updating playlists	2	3.00
4c - Accessibility issues	4	4.67
5a - Irrelevant homepage	2	4.50
5b - Can't find specific songs/ artists	6	3.60
5c - Bad recommendations	12	3.11
6a - Like the app overall	34	3.69
6b - Requests for features	2	4.00
6c - Frustration/ disappointment	12	2.50





Step 5: Lessons learned OR what I'd do differently next time…

- Too many of the sub-themes had a negative tone meaning that positive reviews often lacked specific tags and just had the "Like the app overall" tag.
- In future taxonomies, I would add a step to carefully read reviews and build up my themes as I go rather than having them prepared beforehand.
    - It became clear once I started applying my themes that many were either irrelevant or too broad to be useful - this is a symptom of creating the themes without first reviewing them
- It would be great to be able to use ML models to properly draw insights from the whole dataset, however even doing it manually I could see how useful this analysis would be for businesses who want to see what people are really saying about their product.