movie_analysis

March 8, 2024

#Classifcation Problem: English difficulty of movies based on subtitles (Apiary Project)

0.1 Introduction

Hello and welcome to my take on this project launched by the Apiary Team at Practicum by Yandex! This project is part of my learning path as a Data Scientist at Practicum USA Bootcamp.

English is one of the most spoken languages worldwide, that's not a secret to anybody. As present as it is on the internet, learning it can still be a challenging task for natives of different languages. A fun way to improve listening skills and familiarize with a language could be watching movies and TV series with subtitles, but how much English is too much English?

Being a linguist enthusiast myself and an intermediate Chinese speaker, I know that choosing the right show to watch in a second language can be challenging! It would be great to know how do vocabulary and grammar used in a show compare to my current skills, otherwise things cat get frustrating when each new sentence needs to be paused to look up the meaning of a sentence.

In this project we seek to develop a classification model to decide on movie difficulty based on subtitles. It works by checking English words in a subtitle file and trying to predict how would a professional linguist classify it based on its difficulty. The levels range A2 to B2, based on the English CEFR (Common European Framework of Reference for Languages).

Enjoy!

0.2 Importing modules

0.3 Reading target data

First we will check what data is available from the linguists, which is a table about the movies and respective CEFR levels!

	Movie	Level	Subtitles	Kinopoisk
0	10 Cloverfield Lane	B1	Yes	NaN
1	10 things I hate about you	B1	Yes	No subs
2	A knight's tale	B2	Yes	Everything
3	A star is born	B2	Yes	Nope
4	Aladdin	A2/A2+	Yes	Everything

Data from 88 movies is available. This data will be split between training and testing to be used as learning material for our prediction model!

0.4 Reading wordlists

A good starting point to classify movie difficulties based on their subtitles is to analyze vocabulary. For that, we will use Oxford and Cambridge wordlists previously scraped from the internet to create a word reference table!

```
          word   pos level
2542  academic  NOUN    B2
1733  academic   ADJ    B1
2545   account  VERB    B2
1736   account  NOUN    B1
3257      acid  NOUN    B2
…          …    …    …
5242      well  NOUN    C1
1726     while  CONJ    A2
2522     while  NOUN    B1
2523     whole  NOUN    B1
1727     whole   ADJ    A2

[698 rows x 3 columns]
```

```
6020
```

```
       word        pos level
1818  about  SCONJ ADV    A1
732   about        ADV    B2
141   above        ADV    C2
1819  above  SCONJ ADV    A1
4932  abuse  NOUN VERB    C1
…     …          …    …
3172  while       CONJ    A2
3798  while       NOUN    B1
1093  while        NaN    B2
3173  whole        ADJ    A2
3799  whole       NOUN    B1

[1237 rows x 3 columns]
```

```
Creating lookup table to define word levels in movies…
```

```
The following words are repeated for same POS in the wordlists: ['bank', 'any',
'about', 'above', 'across', 'answer', 'around', 'back', 'back', 'behind',
'below', 'black', 'blue', 'blue', 'break', 'brown', 'brown', 'call', 'capital',
'change', 'clean', 'clean', 'cold', 'complete', 'cost', 'cost', 'dance',
'design', 'design', 'dress', 'dress', 'drink', 'east', 'email', 'email', 'end',
'exercise', 'have', 'have', 'ice', 'need', 'after', 'all', 'alone', 'along',
'anywhere', 'arrangement', 'assistant', 'attack', 'attention', 'average',
'average', 'before', 'best', 'blank', 'bottom', 'brush', 'camp', 'camp', 'care',
'care', 'cause', 'cause', 'chat', 'circle', 'circle', 'control', 'control',
```

```
'copy', 'cross', 'cross', 'cycle', 'cycle', 'download', 'dream', 'expert',
'light', 'rest', 'ring', 'rock', 'access', 'access', 'aim', 'arrest', 'arrest',
'balance', 'ban', 'base', 'bend', 'bite', 'bite', 'block', 'bomb', 'brand',
'calm', 'campaign', 'charge', 'charge', 'cheat', 'chemical', 'claim', 'claim',
'click', 'click', 'commercial', 'contact', 'contact', 'contrast', 'contrast',
'damage', 'direct', 'dislike', 'doubt', 'doubt', 'escape', 'escape', 'exchange',
'exchange', 'export', 'extra', 'lie', 'race', 'used', 'advance', 'aid',
'appeal', 'appeal', 'approach', 'attempt', 'attempt', 'bet', 'bet', 'beyond',
'blame', 'blame', 'broadcast', 'capture', 'capture', 'cast', 'cast',
'characteristic', 'characteristic', 'chief', 'classic', 'collapse', 'collapse',
'comfort', 'command', 'concern', 'concern', 'conduct', 'conflict', 'contest',
'contract', 'contract', 'core', 'crash', 'cure', 'cure', 'curve', 'debate',
'decline', 'decrease', 'defeat', 'defeat', 'delay', 'delay', 'delight',
'demand', 'desire', 'desire', 'display', 'display', 'dozen', 'draft', 'draft',
'encounter', 'estimate', 'evil', 'excuse', 'excuse', 'executive', 'tear',
'besides', 'bid', 'boost', 'chase', 'cheer', 'cheer', 'comic', 'concrete',
'crack', 'crack', 'cruise', 'cruise', 'dive', 'dive', 'divorce', 'divorce',
'equivalent', 'exhibit', 'abuse', 'advocate', 'alert', 'alert', 'alike',
'alike', 'amateur', 'assault', 'attribute', 'blend', 'breed', 'civilian',
'compromise', 'compromise', 'consent', 'consent', 'dispute', 'distress', 'ease',
'echo', 'explosive']
```

By the end of this section, we have python dictionaries with words and their respective parts of speech. This is important because different parts of speech may define different levels for the same word! An additional dictionary has been created for cases where the part of speech is not correctly recognized.

0.5 Reading subtitle files

Now subtitle files will be read and organized into tables by lines. This data will be used for later analysis for each movie.

```
                                name  year  \
0                          mamma_mia  2008
1                           die_hard  1988
2                     the_blind_side  2009
3           the_theory_of_everything  2014
4   the_secret_life_of_walter_mitty  2013
..                                 …    …
81                      pleasantville  1998
82                  the_invisible_man  2020
83                 back_to_the_future  1985
84                       notting_hill  1999
85                     a_star_is_born  2018

                                     filename
0                         Mamma_Mia(2008).srt
1                          Die_hard(1988).srt
2                    The_blind_side(2009).srt
```

```
3          The_theory_of_everything(2014).srt
4   The_secret_life_of_Walter_Mitty(2013).srt
..                                        …
81                     Pleasantville(1998).srt
82                 The_invisible_man(2020).srt
83                Back_to_the_future(1985).srt
84                      Notting_Hill(1999).srt
85                     A_star_is_born(2018).srt

[86 rows x 3 columns]
```

Tables have been created, one for each movie. Each of these tables contain all words, their respective parts of speech, what line they are in and what time of the movie they appear.

0.6 Merging ref tables

After reading subtitle files available, we will join information of tables to make sure data is available for each movie analyzed. During this section, we will also standardize levels for classification.

	Movie	Level
0	10 Cloverfield Lane	B1
1	10 things I hate about you	B1
2	A knight's tale	B2
3	A star is born	B2
4	Aladdin	A2/A2+

	Movie	Level
0	10_cloverfield_lane	B1
1	10_things_i_hate_about_you	B1
2	a_knights_tale	B2
3	a_star_is_born	B2
4	aladdin	A2

We know that 86 movies have their respective subtitles, the table above shows the first five entries, and connects movies with their subtitle files.

0.7 Analyzing movies

Movies will now be analyzed and useful features will be drawn to feed into the classifcation model. Words in each movie will be classified according to their difficulty levels, and we will count these words, as well as other useful information, such as how many words per minute appear for each movie. Words not contained in our wordlist reference will be marked as 'Unk' (unknown)!

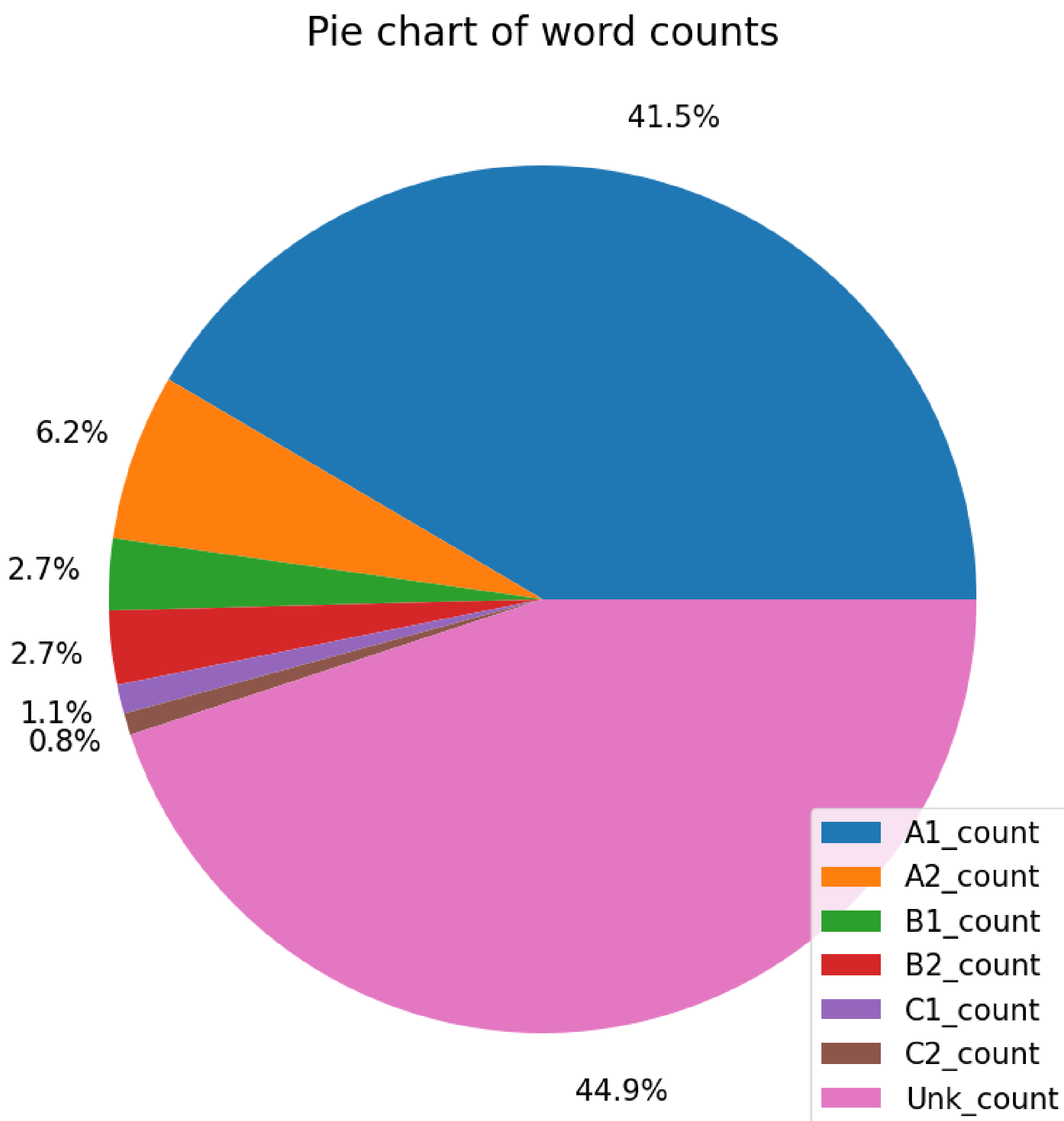
0.8 EDA

Now we have our dataset with counts of words by their levels, and this is the dataset we will use to draw meaningful information about how to determine the difficulty level of a movie.

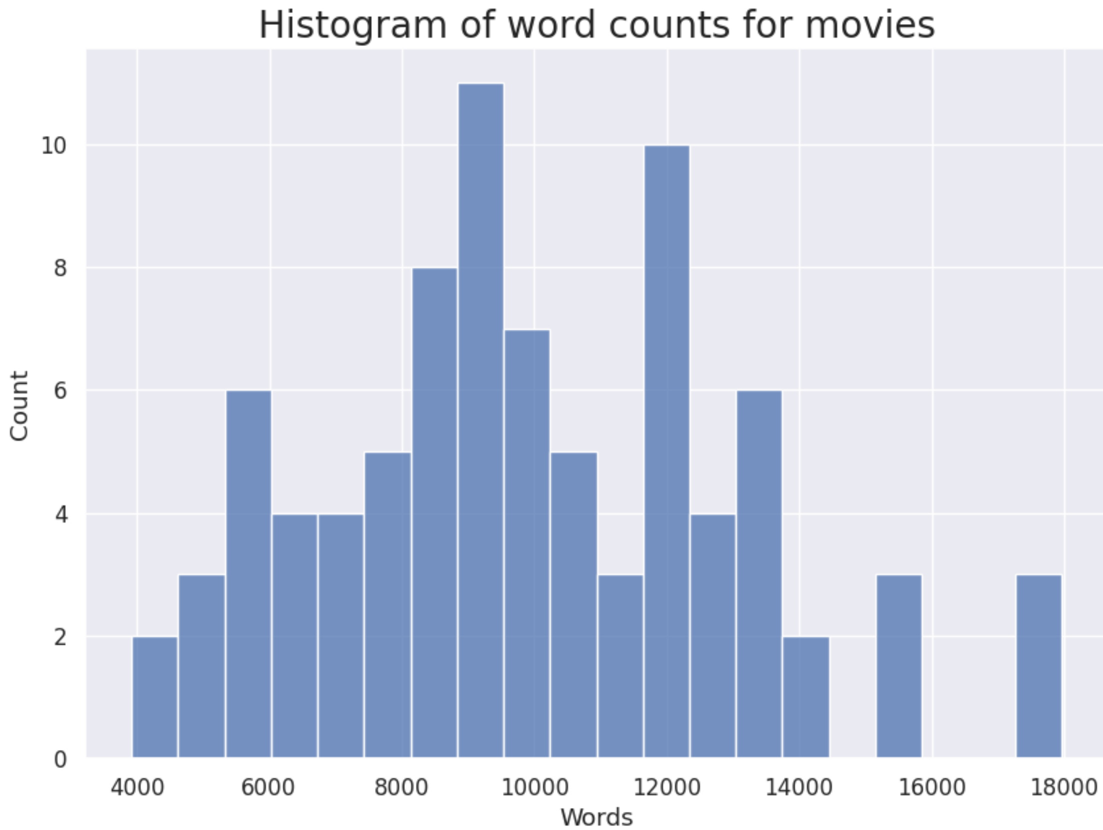
```
                                               movie level  year  \
name
10_cloverfield_lane                  10_cloverfield_lane    B1  2016
10_things_i_hate_about_you  10_things_i_hate_about_you    B1  1999
a_knights_tale                            a_knights_tale    B2  2001
a_star_is_born                            a_star_is_born    B2  2018
aladdin                                          aladdin    A2  1992
…                                                    …   …     …
twilight                                        twilight    A2  2008
up                                                    up    A2  2009
venom                                              venom    B2  2018
warm_bodies                                  warm_bodies    B1  2013
we_are_the_millers                      were_the_millers    B1  2013

                                                     filename A1_count  \
name
10_cloverfield_lane                10_Cloverfield_lane(2016).srt     2366
10_things_i_hate_about_you  10_things_I_hate_about_you(1999).srt     3726
a_knights_tale                          A_knights_tale(2001).srt     2831
a_star_is_born                          A_star_is_born(2018).srt     6248
aladdin                                        Aladdin(1992).srt     3322
…                                                              …        …
twilight                                      Twilight(2008).srt     4127
up                                                  Up(2009).srt     2440
venom                                            Venom(2018).srt     3574
warm_bodies                                Warm_bodies(2013).srt     2159
we_are_the_millers                  We_are_the_Millers(2013).srt     6676

                            A2_count B1_count B2_count C1_count C2_count  …  \
name                                                                      …
10_cloverfield_lane              336      183      160       72       65  …
10_things_i_hate_about_you       482      236      198       95       59  …
a_knights_tale                   457      227      241      110       67  …
a_star_is_born                   835      234      265       78      102  …
aladdin                          559      294      256      104       55  …
…                                  …        …        …        …        …  …
twilight                         592      222      250       61       77  …
up                               362      160      179       42       39  …
venom                            543      219      260      102       68  …
warm_bodies                      307      112      117       51       44  …
we_are_the_millers              1017      364      409      167       90  …

                              C2_pct    Unk_pct 1_verb_lines 2_verb_lines  \
name
10_cloverfield_lane         1.173921  42.532057          485          207
10_things_i_hate_about_you  0.679332  44.778353          478          295
a_knights_tale              0.895124   47.45491          521          222
```

```
a_star_is_born              0.722175  45.043897         913          539
aladdin                     0.618534  48.380567         628          295
…                                …          …         …           …
twilight                     0.80806  44.075979         794          362
up                          0.643777   46.81413         372          224
venom                       0.793558  44.380908         561          285
warm_bodies                 0.902009  42.804428         380          199
we_are_the_millers          0.574896  44.279783         715          514

                           3_verb_lines 4+_verb_lines    duration         wpm  \
name
10_cloverfield_lane                  55            14   96.433333   57.417905
10_things_i_hate_about_you          127            29   95.366667   91.069556
a_knights_tale                       66            12  131.866667   56.761881
a_star_is_born                      134            30       135.2  104.467456
aladdin                              67            25       87.55   101.56482
…                                  …             …          …          …
twilight                             71             4       121.1   78.687036
up                                   59            13   94.966667   63.790804
venom                                99            20   94.716667   90.469822
warm_bodies                          41             4   90.833333   53.702752
we_are_the_millers                  210            70      118.35  132.277144

                           sconj_count word_count
name
10_cloverfield_lane                654       5537
10_things_i_hate_about_you        1362       8685
a_knights_tale                     798       7485
a_star_is_born                    1914      14124
aladdin                            834       8892
…                                 …         …
twilight                          1284       9529
up                                 618       6058
venom                             1152       8569
warm_bodies                        570       4878
we_are_the_millers                1578      15655

[86 rows x 26 columns]
```

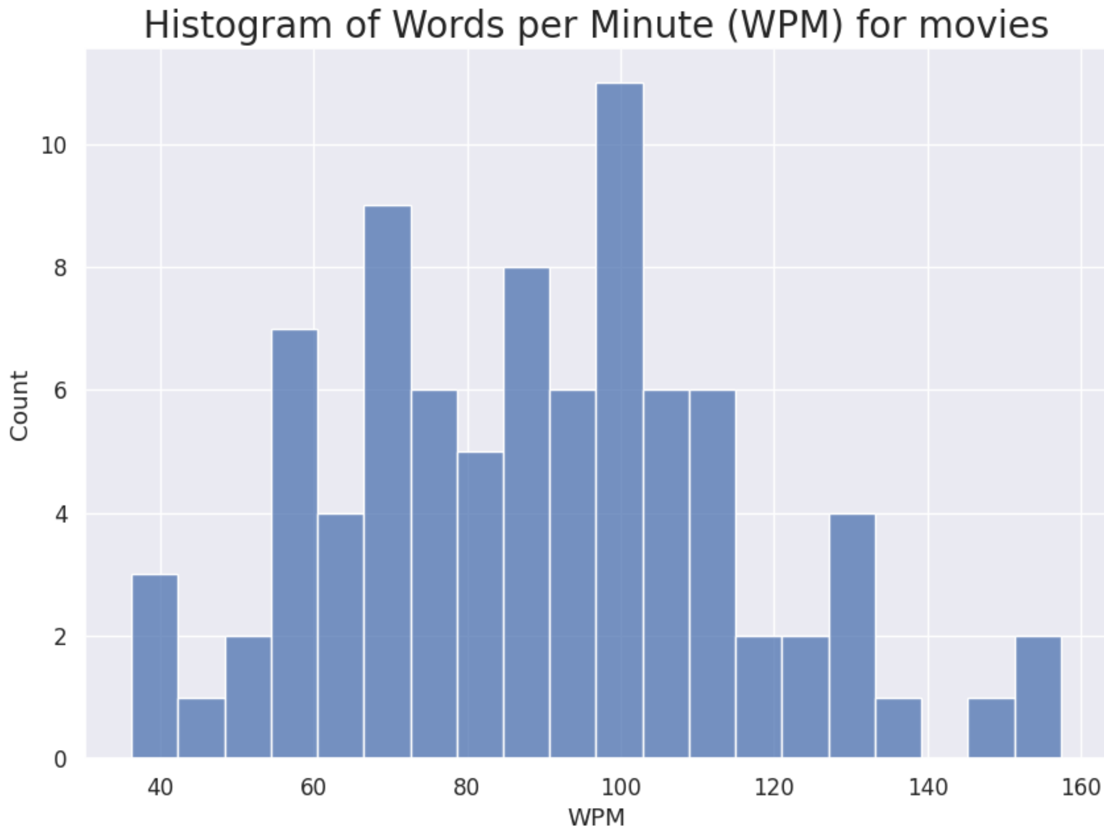
Firstly, we can see that about 40% of the words that appear in movies are not present in our dataset. Although this related to proper nouns in movies, this is an indication that data quality is very low. That is expected, since we are using unofficial *.srt files.



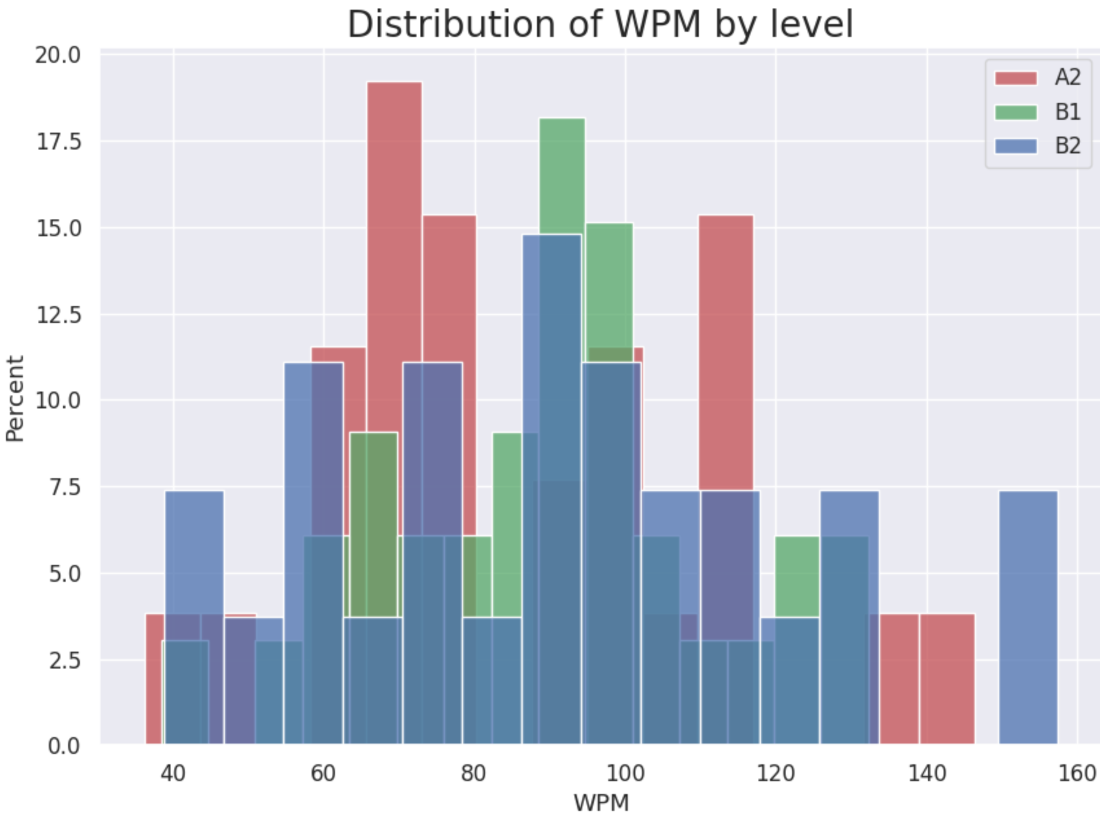
Next, here is an histogram of word counts for movies. Most movies have about 9000 to 12000 words.



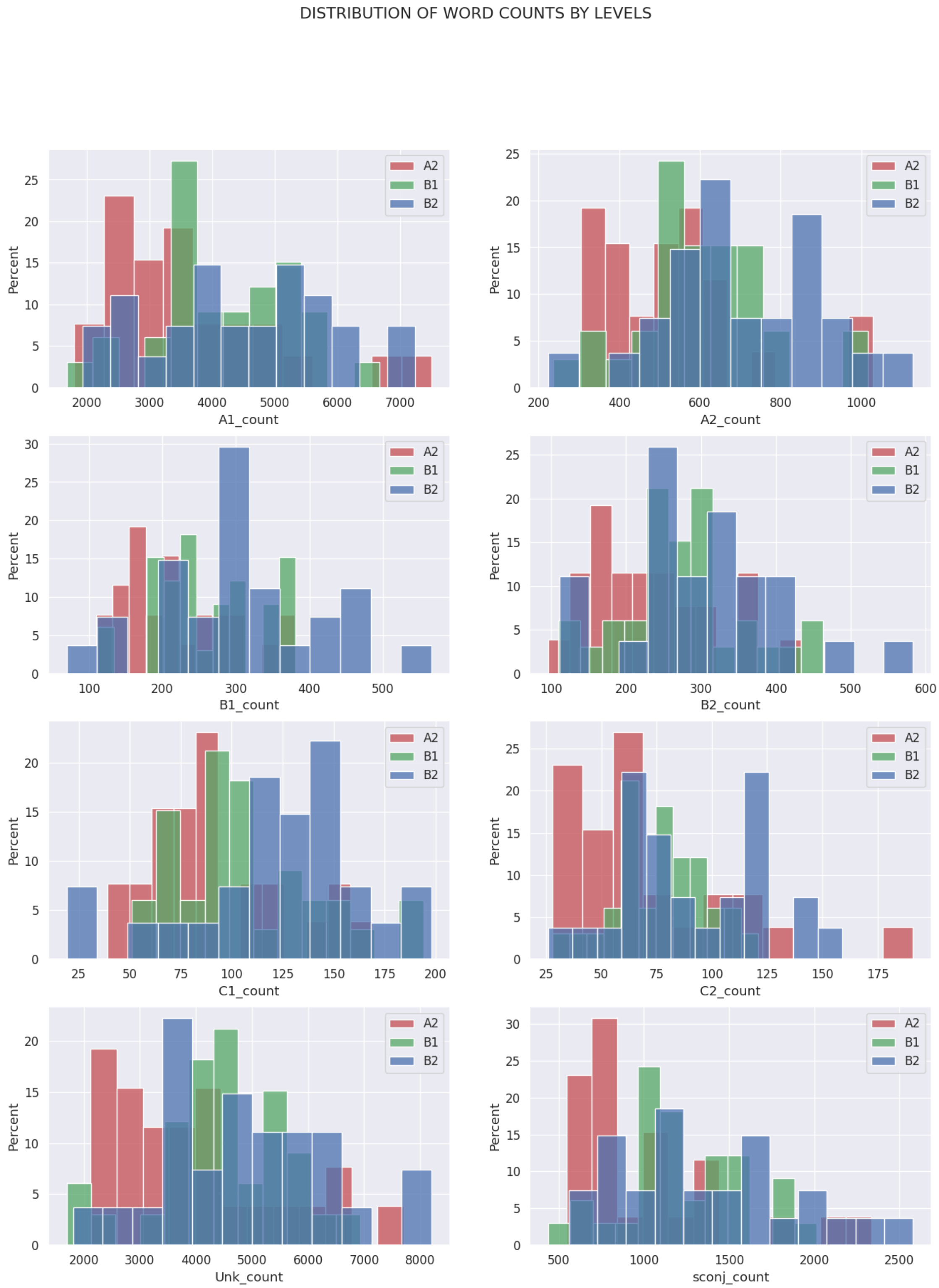
Next is a histogram of words per minute for movies. This is mostly a curiosity feature. Most movies have about 70 to 120 words per minute on average!



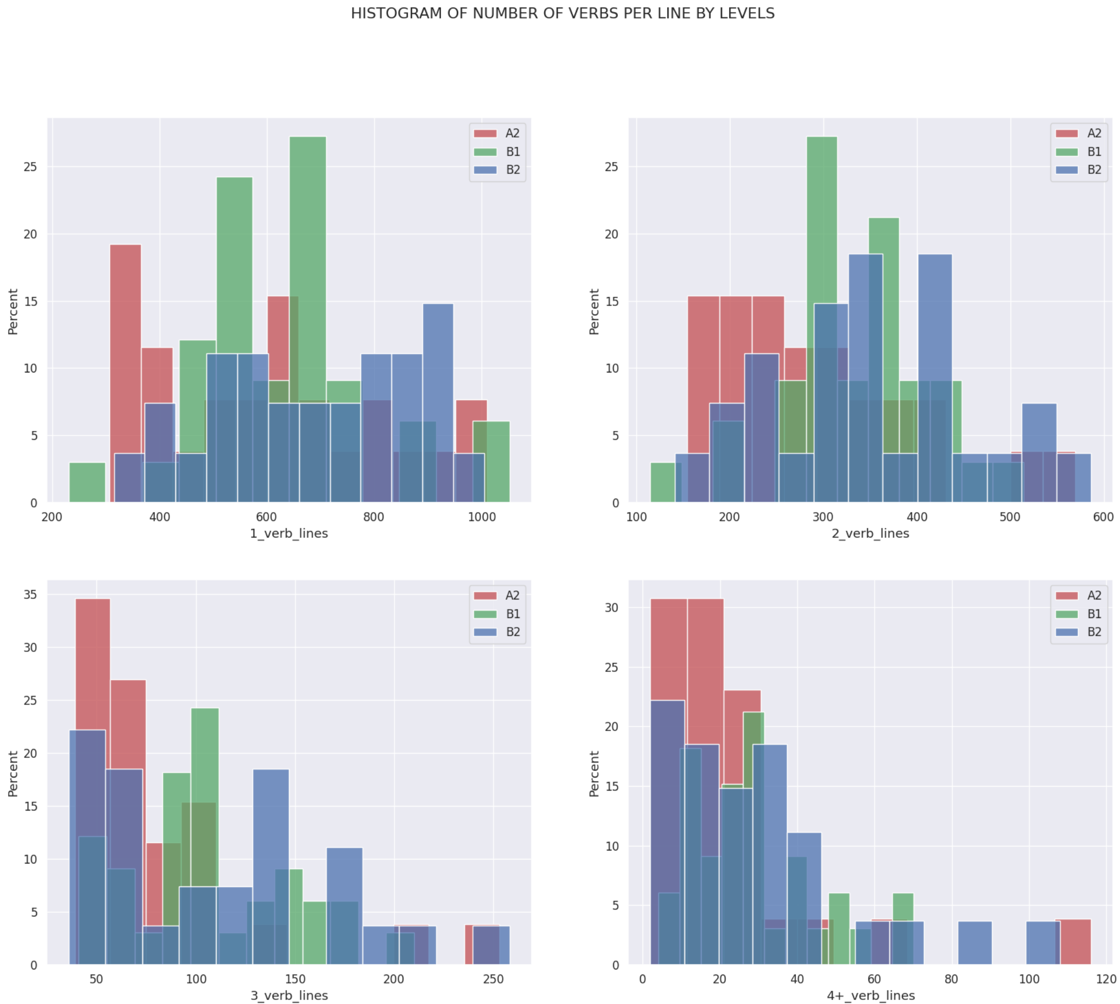
This is the same histogram, but now it gets more interesting. We can see that movies with about 70 words per minute generally belong to the lower level.



Next we can see histograms of word presence by their difficulty. Easier movies tend to use easier words! This is an important factor when deciding the difficulty of a movie.



Number of verbs is an interesting feature as well. Sentences with more verbs tend to be harder because they convey too many actions together, so easier movies will have a lower number of verbs per sentence. The graph below can confirm that.



0.9 Model training

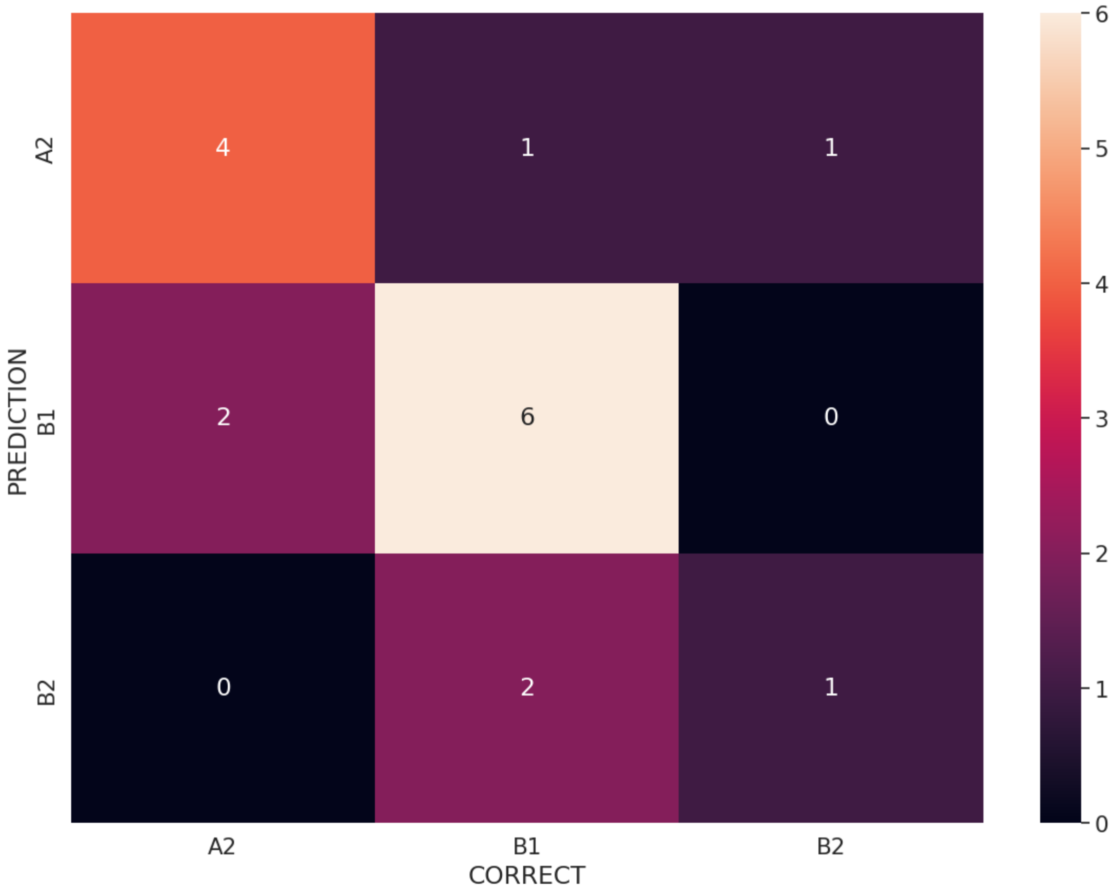
After checking a few of the most notable factors that could help in deciding the difficulty of a movie, we will try to develop a machine learning model to automatically determine difficulty.

```
Splitting dataframes in proportion: 3, 1
Train dataframe size: 51
Test dataframe size: 17
```

After testing quite a few models, we decided on the CatBoostClassifier. The accuracy achieved was:

```
0.6470588235294118
```

Below we can see the confusion matrix of our results.



Out of our validation sample of 17 movies, 11 were guessed correctly! This is a reasonable result for an automatic movie difficulty guesser, considering the low data quality available. This result could definitely be improved using cleaner subtitles or transformer models to analyze context of the movies.

0.10 Conclusion

The main insights gathered during this experimentation were:

- A model for automatic guessing of movie difficulty was created.
- Accuracy achieved was 64%.
- About 40% of the words used in movies are A1 level words.
- More difficult words hold a small share of total word percentage.
- Easier movies indeed tend to have less words per minute.
- They also tend to have less verbs per sentence.
- Data quality gathered had a significantly low quality.

This approach required significant preprocessing and most of the algorithm was done with a general approach. Although no hypothesis testing has been done, the simple results we could acquire considering how much time had to be spent on preprocessing were valuable. Lastly, context analysis could prove useful in this task! This approach only used word counts without any real natural language processing. Still, it proved itself a valid approach for the task with the data available.

I hope you enjoyed this quick analysis of movies based on subtitles.

Thanks for putting up with me until here.

See you next time!