Nachvollziehbare Schritte
Presentation and Analysis of the Dataset:
5-Minute Crafts - YouTube ClickBait Titles

Sanja Srdanovic, 10.12.2021

1. Kurze Darstellung des Problembereichs / Aufriss des Themas

1.1 Inhaltlich

Kern der Untersuchung: Data analysis of Dataset: *5-Minute Crafts - YouTube ClickBait Titles* from *Kaggel*
Grobziele der Arbeit: Analyse the variables that might affect the total views of the videos, inspect their correlation, check which videos have been trending, and what are top 100 keywords, which of these keywords have the highest total views, visualise the data and create a word cloud.

1.2 Begründung des Themas

Darstellung der Relevanz des Themas?

5-Minute Crafts is a very popular channel on YouTube and other social media platforms, such as Facebook and Instagram. They have very appealing titles, which can draw your attention and invite you to watch the videos. Sometimes you can discover some fun and useful ideas, including a lot of tips and tricks for everyday life, not just for crafts. However, as more striking video titles compel more audience, i.e., more clicks, this is why the titles nowadays are somewhat exaggerated and contain some special keywords that increase the number of clicks, therefore the name clickbait titles.
This dataset is a good example to practice data analysis and visualisation of views success/performance of a YouTube channel. Such an analysis could be easily applied to other similar channels or projects, and it is therefore also significant for companies or business owners who want to see the success of their videos (among others) and more generally, the performance of their products or projects.

Darstellung eines persönlichen Erkenntnisinteresses.

This dataset was particularly interesting to me because I have been following 5-Minute crafts for years, as the page was useful for me to get some ideas for crafts that I could do with my students when I was teaching English a couple of years ago. I kept following the page 5-Minute crafts, but I have noticed that they have more and more clickbait titles, and the tricks and hacks are sometimes a bit ridiculous or unexpected from the title itself. That is why it was interesting for me to see what causes the higher number of views, especially what keywords from the titles are the most frequent and most viewed.

2. Nachvollziehbare Schritte

2.1 Der Stand der Forschung / Auswertung der vorhandenen Literatur / Tutorials ...

Wurde das Problem früher bereits untersucht?

Welche Aspekte wurden untersucht und welche nicht?

Welche Kontroversen gab es und welche Methoden standen bis jetzt im Vordergrund?

Clickbait video titles is nowadays a popular topic, as well as the data analysis of the performance of videos on social media platforms, and it has been already investigated from different perspectives and with various methods, including ML and NLP.

Lösungswege strukturieren!

- Loading and inspecting the data, cleaning the data, analysing the data, and plotting the results.
- Finding the most common keywords, and which of them have the highest average total views, create a wordcloud.

2.2 Fragestellung

1. Which features of a video title, i.e., which variables are most important with respect to total views?
2. Which keywords are mostly used in the 5-Minut Craft Clickbait Video titles?
3. Which of these keywords have the highest average total views?

2.3 Methode

Loading the data set

```
d = path.dirname(__file__) if "__file__" in locals() else os.getcwd()
data = pd.read_csv("/Users/sanjasrdanovic/Desktop/Data Science/Python/Projekt/5-Minute Crafts.csv")
```

Inspecting the dataset

```
print(data.head())
print(data.shape)
data.shape[0]
data.shape[1]
```

There are 4978 rows and 15 columns. The video ID and title are the only non-numerical columns. There are 13 numerical variables: *active_since_days, duration_seconds, total_views, num_chars, num_words, num_punctuation, num_words_uppercase, num_words_lowercase, num_stopwords, avg_word_len, contain_digits, startswith_digits* and *title_sentiment*.

Then, I checked whether there were any null values, and there were none, and then I did descriptive statistics.

```
print(data.info())
data.describe()
```

Check the outliers with boxplots

Afterwards, I plotted boxplots to see the quartiles and outliers. Given that it would be difficult to see anything on a single plot because there are a lot of variables, and for some of them the numbers are much higher than for the others, and the range difference is huge to compare anything, I needed to create subplots, where axes is an array with each subplot. With *shape[1]* I fetched the columns of pandas dataframe (columns with numerical values), and with *sns.set()* I set the font scale, with *figsize* the size of the graph, and made it tighter with *fig.tight_layout.*

```
sns.set(font_scale=1.5)
fig, axes = plt.subplots(nrows= data.describe().shape[1], figsize = (20,25))
fig.tight_layout(pad = 3)
```

Then I specified for each plot in which subplot I wanted them with the argument *ax* (*n* is an axis counter), and saved the figure.

```
n = 0
for i in data.describe().columns:
    sns.boxplot(x = data[i], orient='h', ax = axes[n], color='skyblue')
    n += 1
plt.savefig("boxplot_data.png")
```

Plot the distribution for numerical variables

I plotted histograms for all variables to see their distribution. Again, I used a *for*-loop so that I did not have to do it manually 1 by 1, and this way all plots were produced at once, and I also saved them at once.

```
fig, axes = plt.subplots(nrows= data.describe().shape[1], figsize = (40,40))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
 n = 0
for i in data.describe().columns:
    sns.displot(x=data[i], color='skyblue')
    plt.ticklabel_format(style='plain', axis='x',useOffset=False)
    n += 1
    plt.savefig(i)
```

Check the correlation of numerical values

I plotted a heatmap to see the correlations and which of the variables have the most correlation with total views. I set the font, figure size, changed the palette, added linecolor and annotation so that it is more comprehensive.

```
sns.set(font_scale=1.8)
plt.figure(figsize=(20,20))
sns.heatmap(data.corr(),annot = True, cmap = "magma", linecolor = "white", linewidth = 1.5)
plt.savefig("correlation_plot_data_2.png")
```

Pre-process the textual data to get clean data in order to find most common keywords

Cleaning data: convert to lowercase, remove punctuation and special characters, stopwords, numbers

```
text = " ".join(data['title'])
text = text.lower()
text = "".join(t for t in text if t not in punctuation)
text = [t for t in text.split() if t not in stopwords] # removing the stopwords
text = [t for t in text if not t.isdigit()]
```

Find the most common keywords

Plot the frequency of 10 most popular keywords

```
x = [count for word, count in Counter(text).most_common(10)]
y = [word for word, count in Counter(text).most_common(10)]
plt.figure(figsize=(15,10));
axes = sns.barplot(x=x, y=y, palette="pastel")
plt.title("10 Most Frequent Keywords used in 5-Minute Craft Titles");
plt.xlabel("Frequency", fontsize=18);
plt.yticks(fontsize=16);
plt.xticks(fontsize=16);
plt.ylabel("Keywords", fontsize=18);
plt.savefig("most_freq_keywords.png")
```

Plot most common keywords according to the average total views

None of the variable was significant predictor for the number of total views, so it might be something else from the title that drew attention of the audience, after all, these titles are clickbait titles and the words chosen are extremely important. So, I examined the average views of the top 100 keywords (with *Counter*) that I later filtered out to see which 15 keywords from the titles had most views.

```
top100 = Counter(text).most_common(100)
for word in top100:
    word =word[0]
    data[word] = data['title'].apply(lambda x : 1 if word.lower() in x.lower() else 0)
```

I created an empty dictionary *keyword_views* where I put keys(*k*) keywords from top100 and values(*v*)

```
keyword_views = {}
for word in top100:
    word = word[0]
```

I needed to return grouped lists in a column as a dictionary and to aggregate total views and mean and to add this grouped value as a column to dict. I decided to use the mean rather than the sum of total views values, and to show an average total view per keyword.

```
  dg = data.groupby(word).agg({"total_views" : "mean"}).to_dict()['total_views']
  if 1 in dg:
    keyword_views[word] = dg[1]
keyword_views = {k: v for k, v in sorted(keyword_views.items(), key=lambda item: item[1])}
```

Then I plotted the bar graph, I first specified the axes: *x* axis are values (average views), and *y* axis are keys(keywords).

```
x = [_ for _ in (keyword_views.values())][::-1]
y = [_ for _ in (keyword_views.keys())][::-1]
plt.figure(figsize=(20,10));
axes = sns.barplot(x=x[:15], y=y[:15], palette = "pastel")
plt.title("Keywords with the highest number of views");
plt.xlabel("Average Views", fontsize=18);
plt.yticks(fontsize=16);
plt.xticks(fontsize=16);
plt.ticklabel_format(style='plain', axis='x',useOffset=False)
plt.savefig("keywords_x_avg_views.png")
```

Make a wordcloud

First, I specified the style, and from which columns were words taken and added them to a list. Afterwards, I specified that the *stopwords* are from *nltk.corpus.stopwords.words("english").*

```
def grey_color_func(word, font_size, position, orientation, random_state=None, **kwargs):
  return "hsl(0, 0%%, %d%%)" % random.randint(60, 100)
saved_column = data["title"]
# print(saved_column)
lst =[]
for i in saved_column:
  lst.append(i)
# print(type(lst))
keywords = " ".join(lst)
print(type(keywords))
print(keywords)
print(len(keywords))
print(len(set(keywords)))
stopwords = nltk.corpus.stopwords.words("english")
```

Then I set up the *mask*. I used a lightbulb picture because that is in the logo of 5-Minute Crafts.

```
mask = Image.open("/Users/sanjasrdanovic/Desktop/Data Science/Python/Projekt/lightbulb.jpg")
print(mask)
```

Setting up the *wordcloud*, I specified the *stopwords*, converted the *mask* to be a *numpy array*, and set the background color as blue (as in the logo of 5-Minute Crafts) and I decided to use 500 words max, and I specified the font of words and size of the figure.

```
wordcloud =
WordCloud(font_path="/Users/sanjasrdanovic/Desktop/DataScience/Python/Projekt/XeroxSansSerif
WideBoldOblique.ttf",
                    stopwords = stopwords,
                    mask = np.array(mask),
                    background_color="blue",
                    max_words=500,
                    max_font_size=200,
                    width=1500,
                    height=2500,
                    ).generate(keywords)
plt.axis("off")
plt.imshow(wordcloud.recolor(color_func=grey_color_func, random_state=3),interpolation='bilinear')
plt.savefig("wordcloud_5min_crafts.png", dpi=300)
plt.show()
```

Check the top 10 trending videos last week

Finally, I wanted to observe what videos have been trending last week, so I made a top 10 list.

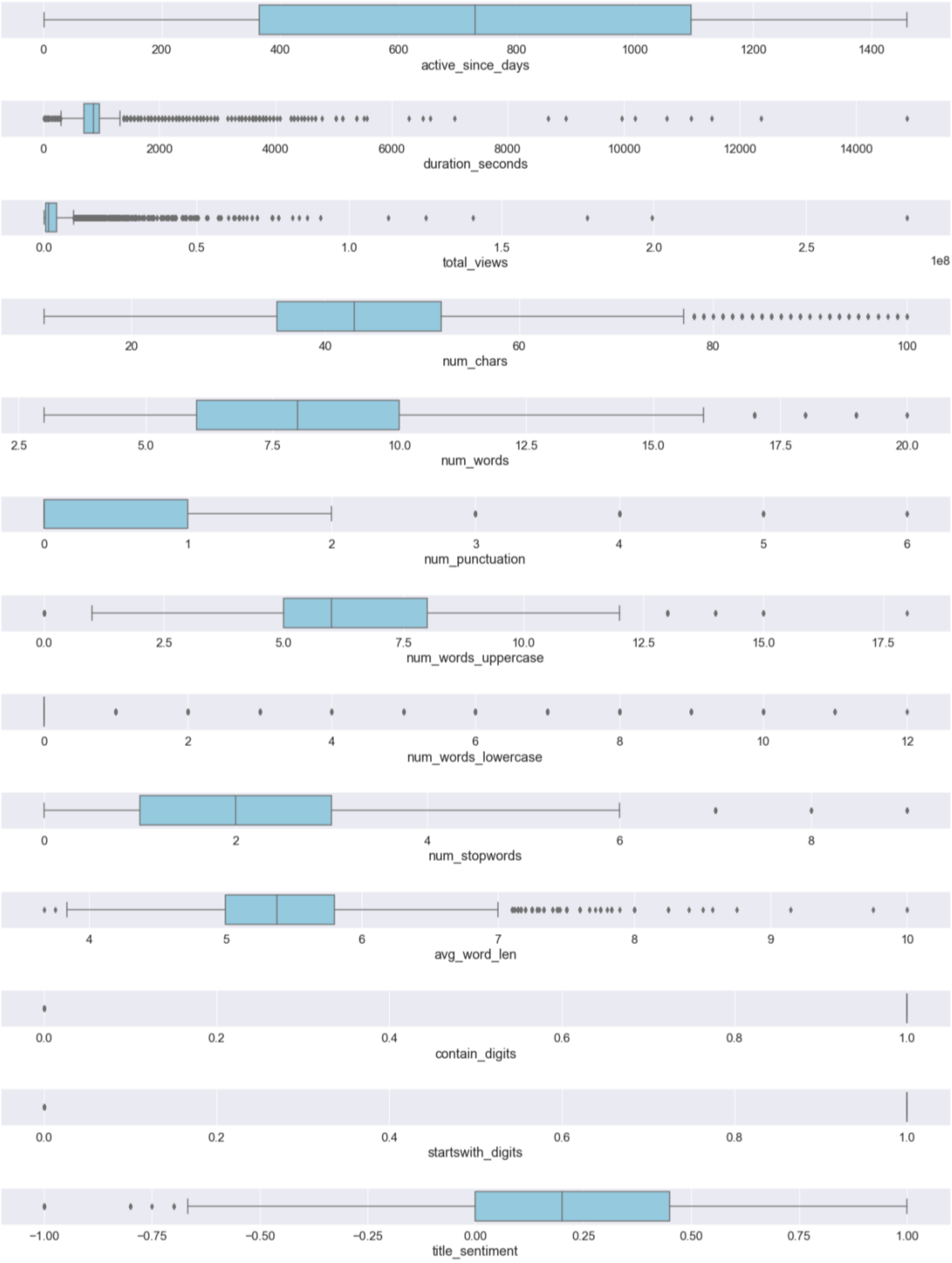
```
trending = data[data['active_since_days'] <=7]
trending = trending.sort_values(by =['total_views'] , ascending = False).head(10)
trending = trending.reset_index()
print(trending)
trending.drop(trending.columns.difference(['title','total_views']), 1, inplace = True)
position = range(1,11)
trending["position"] = position
trending = trending.set_index('position')
print(trending)
```

I saved the new trending videos dataframe in Excel.

```
top10 = pd.ExcelWriter('trending_last_week.xlsx')
trending.to_excel(top10)
top10.save()
print('DataFrame is written successfully to Excel File.')
```

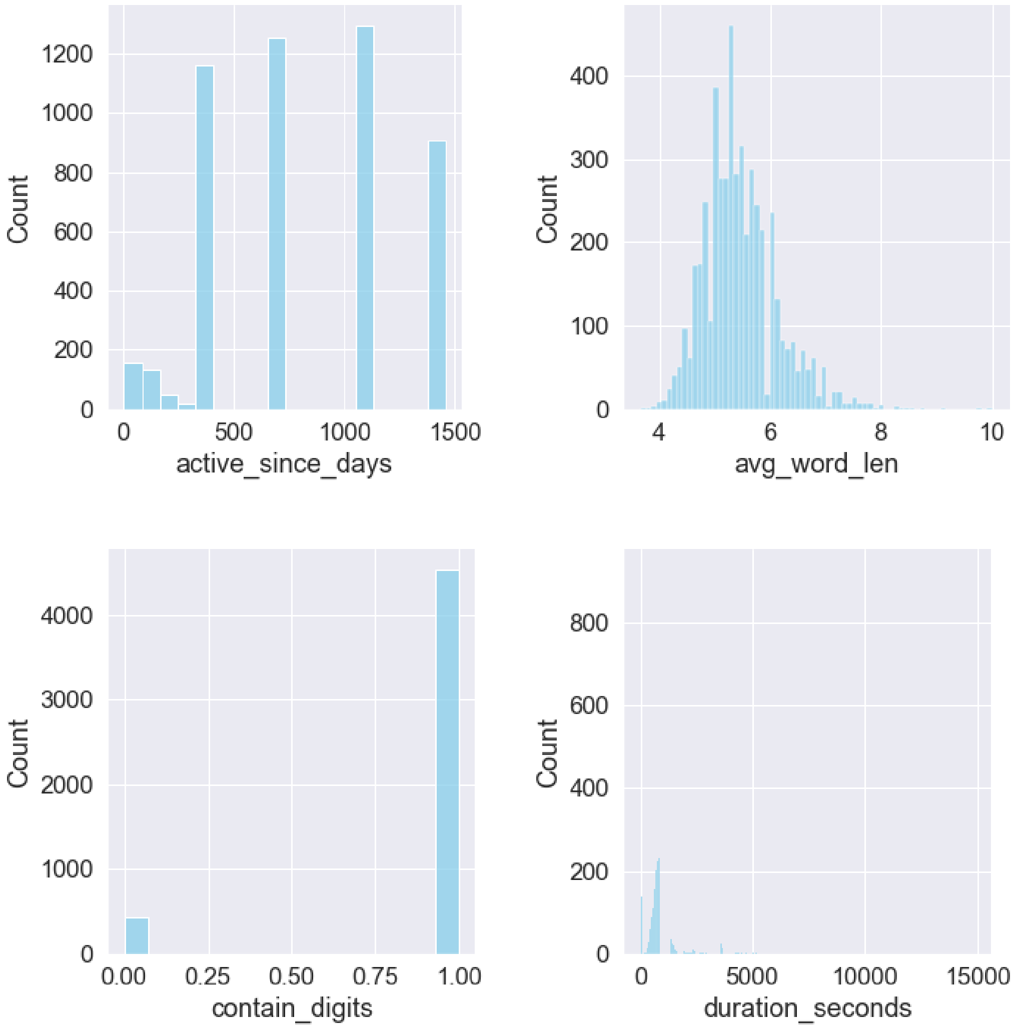
2.6 Ergebnisse

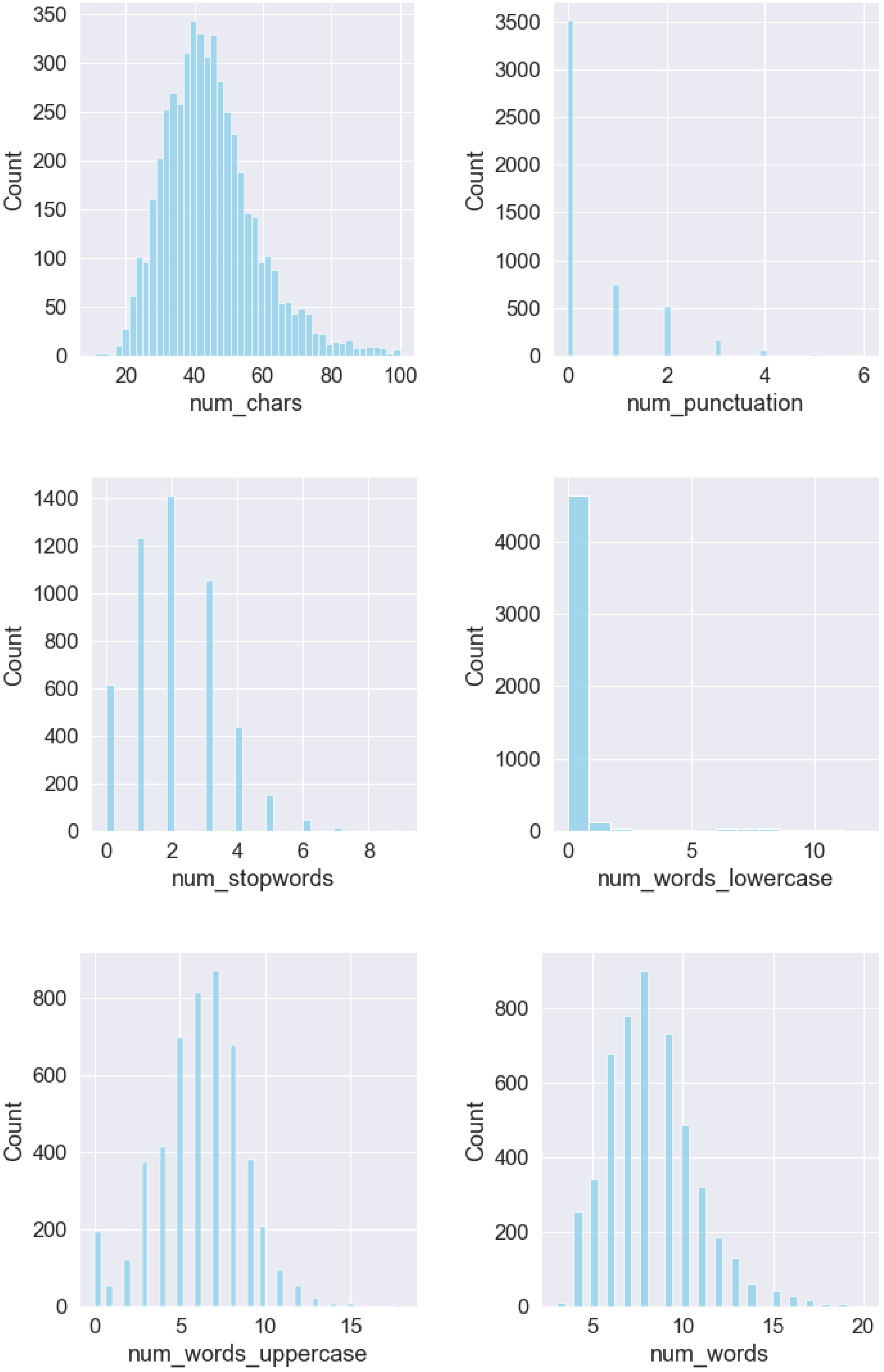
Boxplots

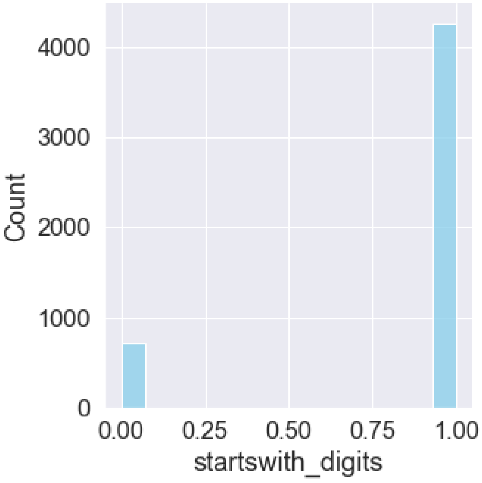


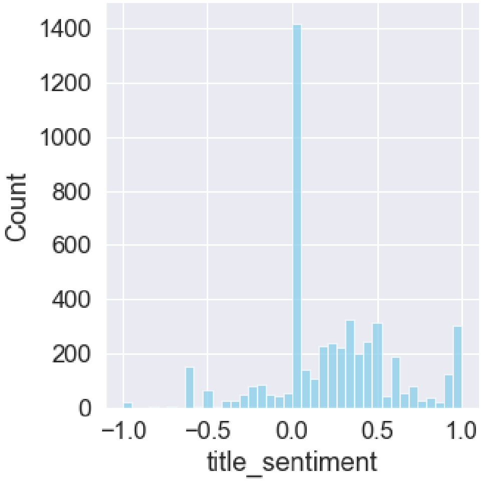
Here are a few observations from boxplots for all the respective variables. For the period of activity, it is mostly from 400 to 1000 days. For the duration of the videos, we can observe that there are many outliers, and in general, most videos are relatively short. When it comes to the total number of views: most videos only have less than millions of views (and the majority is much under half million), but there are also a few with extreme values. Number of characters is on average around 40-50, there are some outliers with more than 80 characters. Number of words is pretty short, average from 6 to 10, with only a few outliers. Number of punctuations is very low, usually either none or only until 2 punctuations in a title, with a few exceptions. There are some uppercase words, but for lowercase just several outliers. Number of stopwords is very low, mostly 1-3. The average word length is around 5,6, and there are some instances above as well. Contain digits and starts with digits are categorial values, so just 0 or 1. Finally, the sentiment of the titles is mostly positive (above 0), negative sentiments (under 0) are mostly outliers. Next, we can see the distribution of all the numerical variables mentioned.

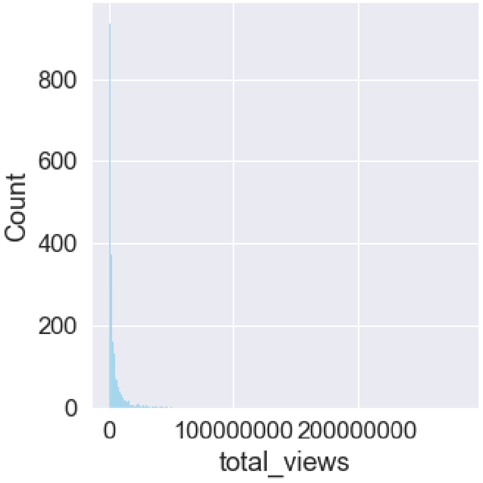
Distribution of all numerical variables




Count
350
300
250
200
150
100
50
0
20
40
60
80
100
num_chars
Count
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
num_punctuation
Count
1400
1200
1000
800
600
400
200
0
0
2
4
6
8
num_stopwords
Count
4000
3000
2000
1000
0
0
5
10
num_words_lowercase
Count
800
600
400
200
0
0
5
10
15
num_words_uppercase
Count
800
600
400
200
0
5
10
15
20
num_words



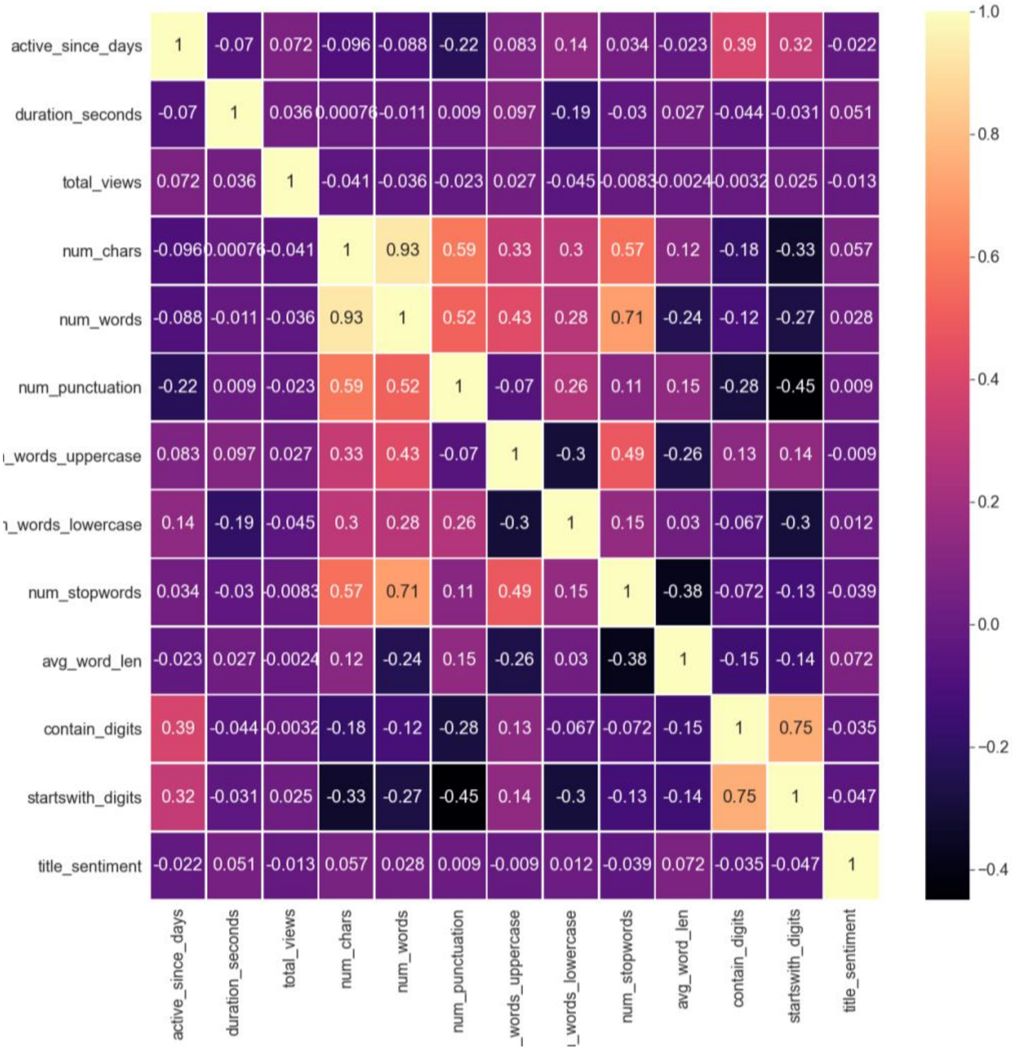




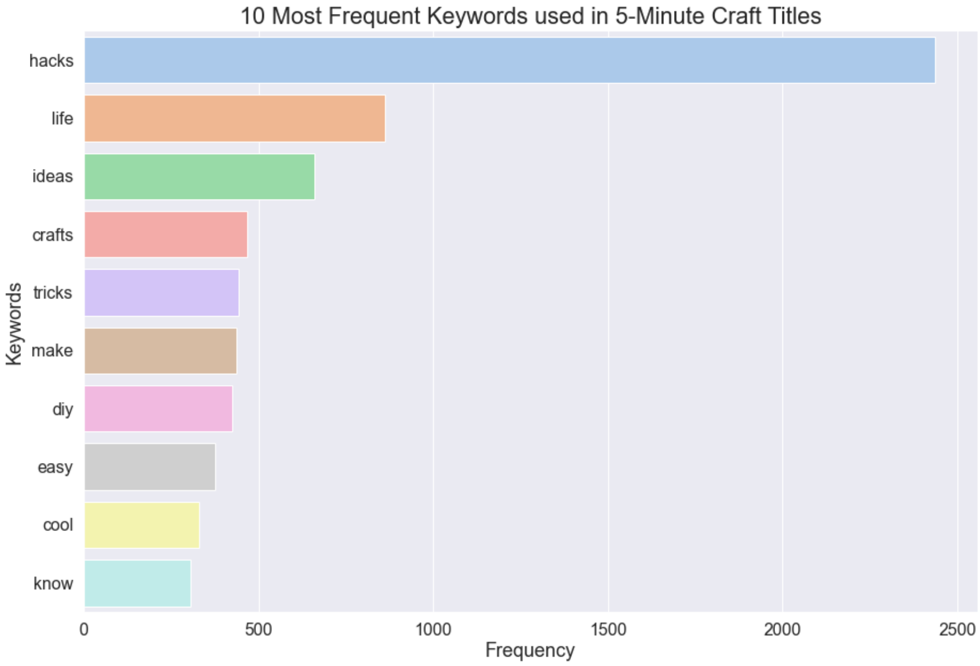
Correlation of numerical variables

Contrary to the expectations, we can observe from the correlation matrix that none of the features strongly correlate with the total number of views. *Num_words* and *num_char* have a strong correlation (0.93) which of course makes sense; the variable *num_words* has a pretty high correlation with *num_stopwords* (0.71); and also *contains_digits* and *starts_with_digits* (0.75) but their correlation with each other is not so important for the whole analysis, because we want to see what affects the number of *total_views*, and none of the variables has a high correlation with it.

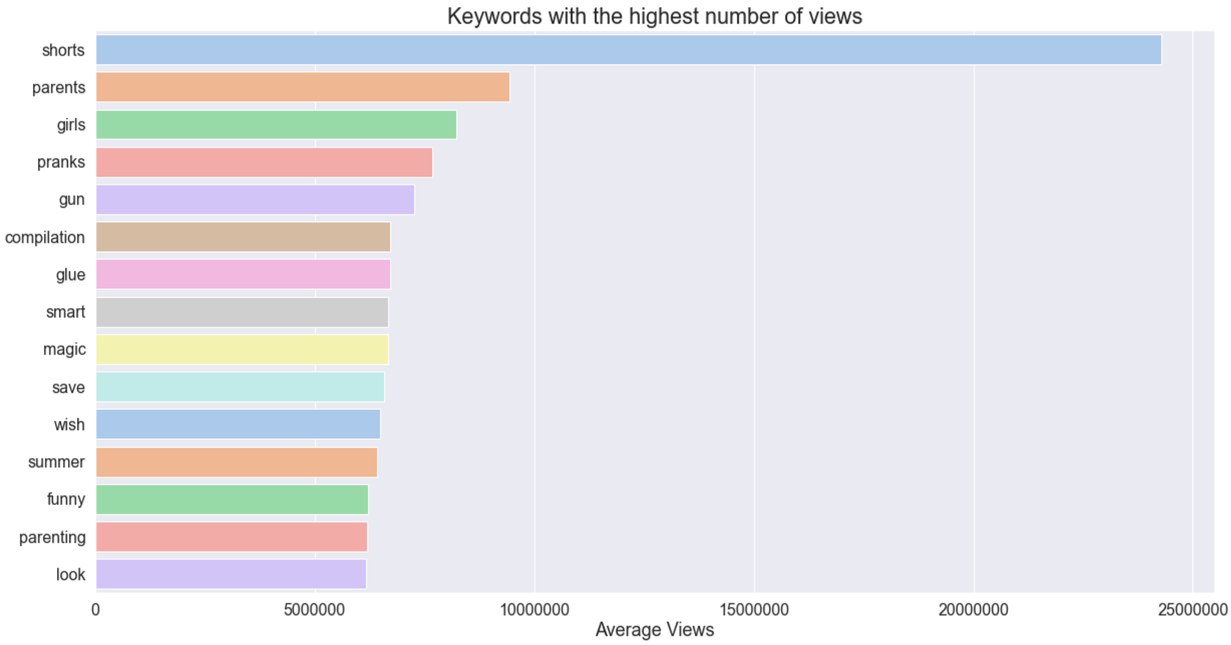
Given that none of the expected variables such as title sentiment, number of words, or duration have significant impact on the number of total views, it must be something else from the title that draws attention to videos and their views. Since clickbait titles are all about being more appealing and attention-seeking, they are directly tied to the *choice of words.* Therefore, I decided to have a look what keywords are mostly used in the titles.



Top 10 Keywords

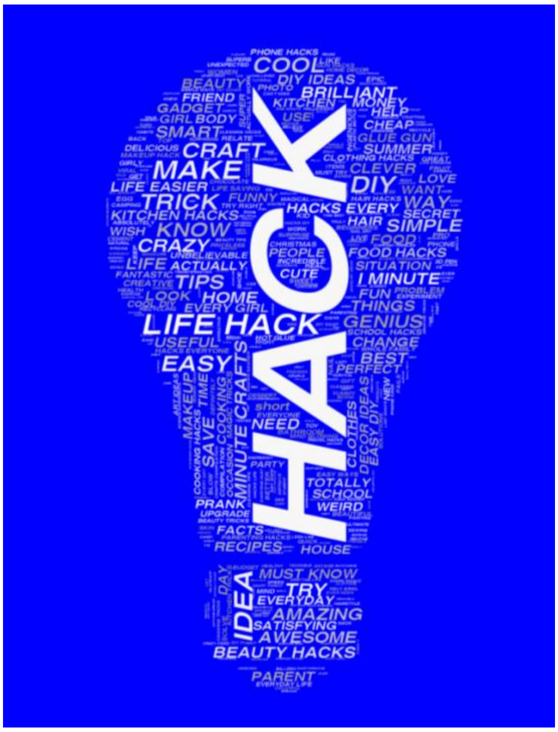


Keywords with the highest average total views



Wordcloud

Since the logo of 5-Minute crafts is a lightbulb with blue background, this is how I decided to create my wordcloud:



Trending last week: Top 10 Videos

position	title	total_views
1	MEGA PARENTING COMPILATION || Best DIYs & Hacks For Parents	1856595
2	GENIUS SCHOOL HACKS and TIPS FOR PARENTS	1639826
3	KID'S ROOM MAKEOVER IDEAS || Awesome Home Decorating Hacks	582492
4	SMART GADGETS TO MAKE YOUR LIFE EASIER	495602
5	Brilliant Clothing Hacks For Adults and Their Kids	471701
6	LOW-BUDGET ROOM MAKEOVER IDEAS || Smart Repair Hacks & Tricks	441260
7	These Awesome Cleaning Hacks Will Blow Your Mind 🤯 #shorts	425344
8	Amazing Beauty Ideas & Gadgets You Need To See	393097
9	SOAP BUBBLES CHALLENGE with 5-MINUTE CRAFTS #shorts	335063
10	CLEVER WAYS TO HIDE YOUR TREASURES || Useful Home Hacks & Tips	179509

Coming back to the research questions, we observed that none of the given variables correlate with the number of total views. Therefore, I decided to inspect what are the most common keywords, and which of them had the highest average total views, i.e., what words compel the audience to watch the videos. The 10 most frequent words are the following: *hacks, life, ideas, crafts, tricks, make, diy, easy, cool, know*. And the 15 keywords from the titles that drew most attention to the videos and which have the highest total views are: *shorts, parents, girls, pranks, gun, compilation, glue, smart, magic, save, wish, summer, funny, parenting* and *look*.

2.7 Ausblick

What could be done in further projects is creating and fitting models to get clearer picture on what could influence the total views, and training models to predict the pattern of total views of videos.

The procedure described here could be applied to other datasets with many variables, in particular for the analysis of the performance of other channels or social media contents, campaigns or projects.