Charles Deliege – Arthur Clavier – DIA2

Python Project

News Popularity Prediction

01 - Modules utilisés

Numpy - Pandas - Plotly.express
Seaborn -Matplotlib

```
import numpy as np
import pandas as pd
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
```

02 - Le Dataset

61 Colonnes
dtypes : float64(59), int64(1), object(1)
39644 Rows, 18.5 MB

	url	timedelta	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	n_non_stop_unique_tokens	num_hre
0	http://mashable.com/2013/01/07/amazon-instant-...	731.0	12.0	219.0	0.663594	1.0	0.815385	4
1	http://mashable.com/2013/01/07/ap-samsung-spon...	731.0	9.0	255.0	0.604743	1.0	0.791946	3
2	http://mashable.com/2013/01/07/apple-40-billio...	731.0	9.0	211.0	0.575130	1.0	0.663866	3
3	http://mashable.com/2013/01/07/astronaut-notre...	731.0	9.0	531.0	0.503788	1.0	0.665635	9
4	http://mashable.com/2013/01/07/att-u-verse-apps/	731.0	13.0	1072.0	0.415646	1.0	0.540890	19

5 rows × 61 columns

Data Preprocessing

Let's separate values that contains outliers from the other with only few possible values (binary columns)

```
low_number_of_value=[]
for col in num.columns:
    if np.abs(df[col].skew()) > 1 and df[col].nunique()<5:
        low_number_of_value.append(col)

low_number_of_value
```

```
['data_channel_is_lifestyle',
 'data_channel_is_entertainment',
 'data_channel_is_bus',
 'data_channel_is_socmed',
 'data_channel_is_tech',
 'data_channel_is_world',
 'weekday_is_monday',
 'weekday_is_tuesday'
```

Select columns

```
truely_continuous_value=[]
plt.figure(figsize=(20,20))
for col in num.columns:
    if np.abs(df[col].skew()) > 1 and df[col].nunique()>15
        truely_continuous_value.append(col)
truely_continuous_value
```

```
['n_tokens_content',
 'n_unique_tokens',
 'n_non_stop_words',
 'n_non_stop_unique_tokens',
 'num_hrefs',
 'num_self_hrefs',
 'num_imgs',
 'num videos',
```

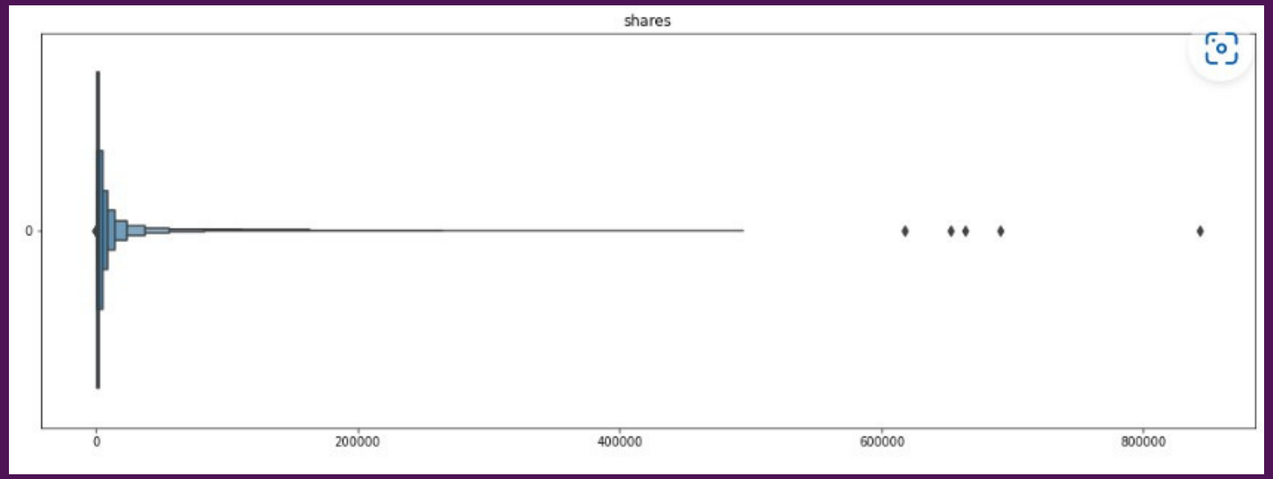
No null data

```
number_zero=df.isnull().sum().sum()
number_Nan=df.isna().sum().sum()
number_zero,number_Nan
```

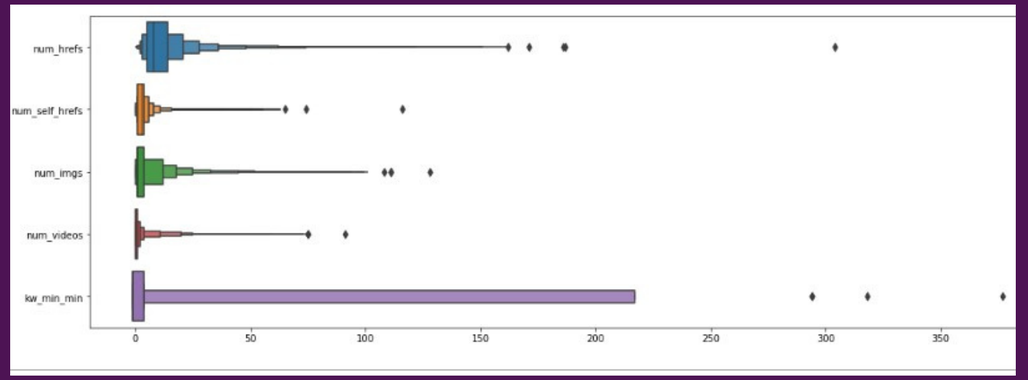
```
(0, 0)
```

Data Preprocessing

Outlier Check



We will use the shares to create our targer variable



We could easily remove these outliers

Popularity variable

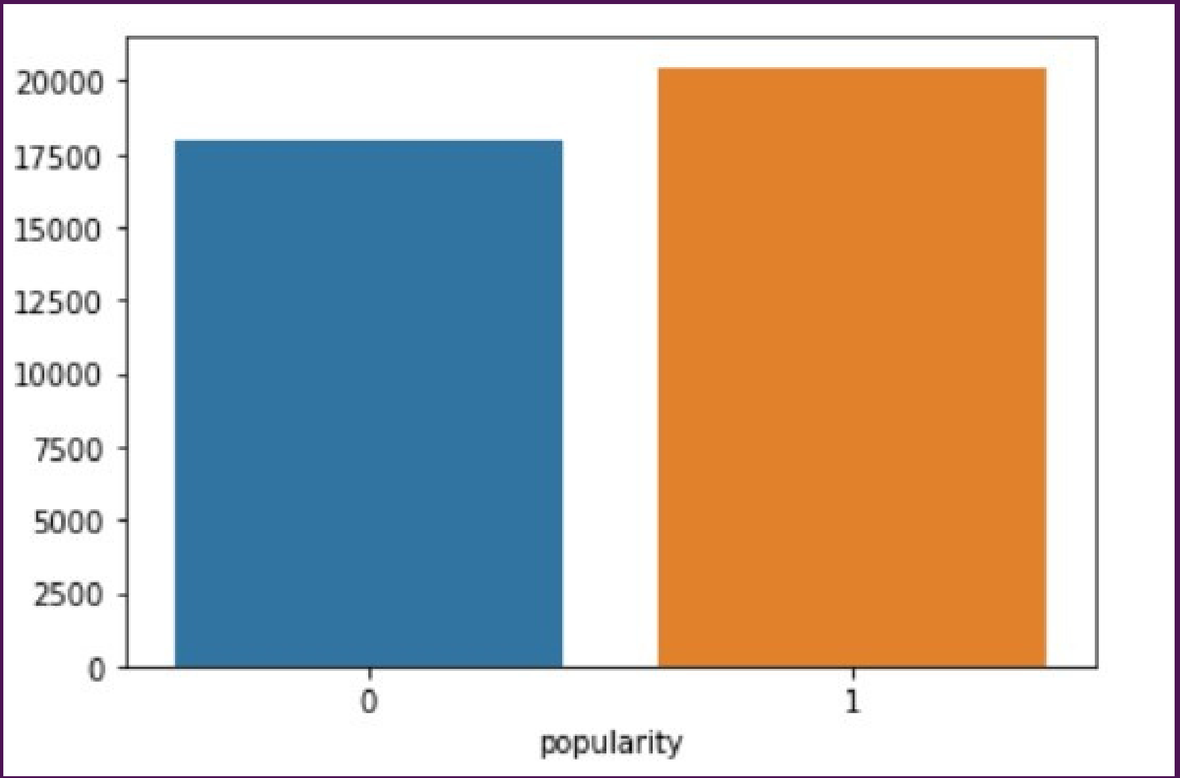
Creation of the target variable

```
df['popularity']=df['shares'].apply(lambda x: 1 if x>=1400 else 0)
df['popularity'].value_counts()
```

```
1    20464
0    17999
```

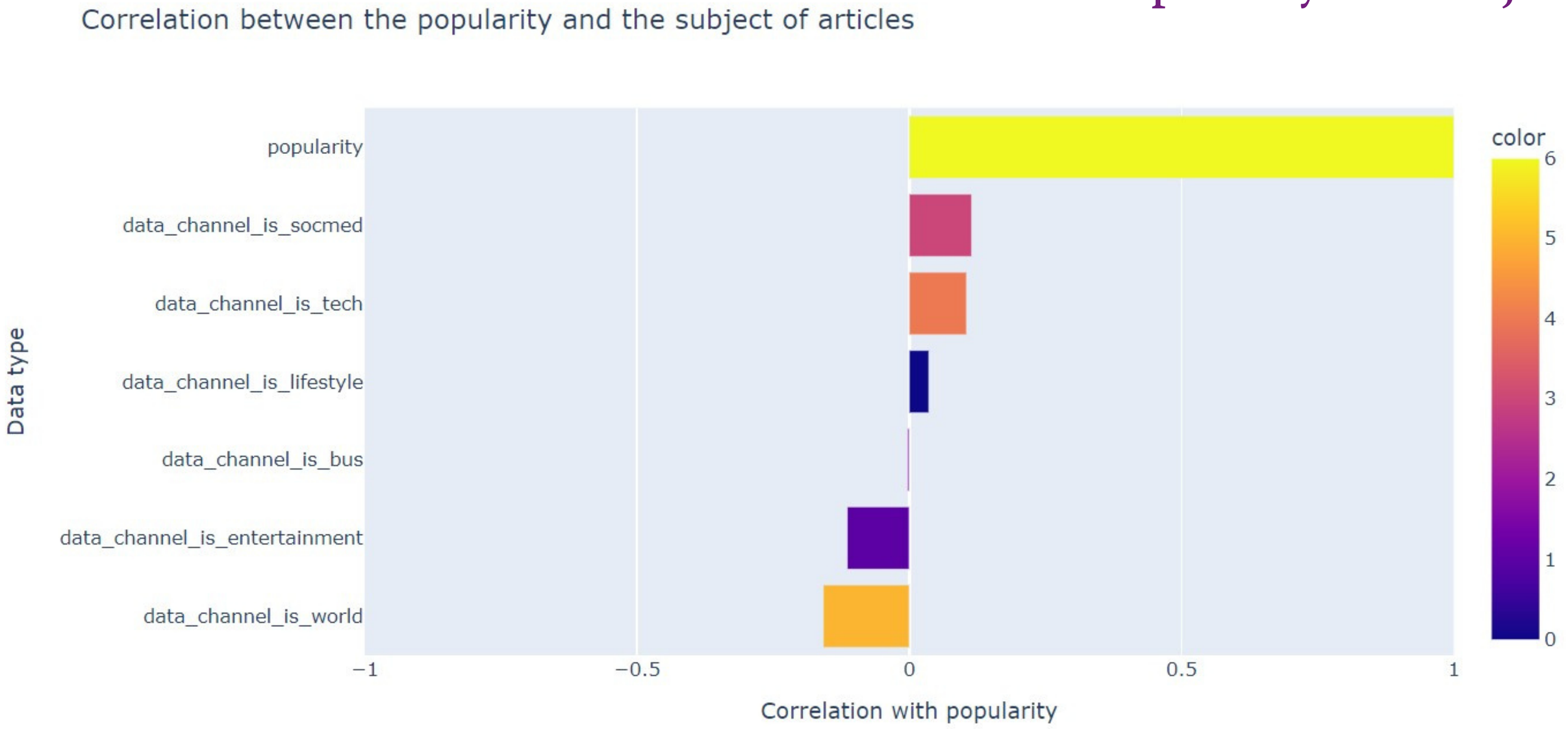
```
df['shares'].describe()
```

```
count     38463.000000
mean       3355.360398
std       11585.968776
min           1.000000
25%         945.000000
50%        1400.000000
75%        2700.000000
max      843300.000000
Name: shares, dtype: float64
```



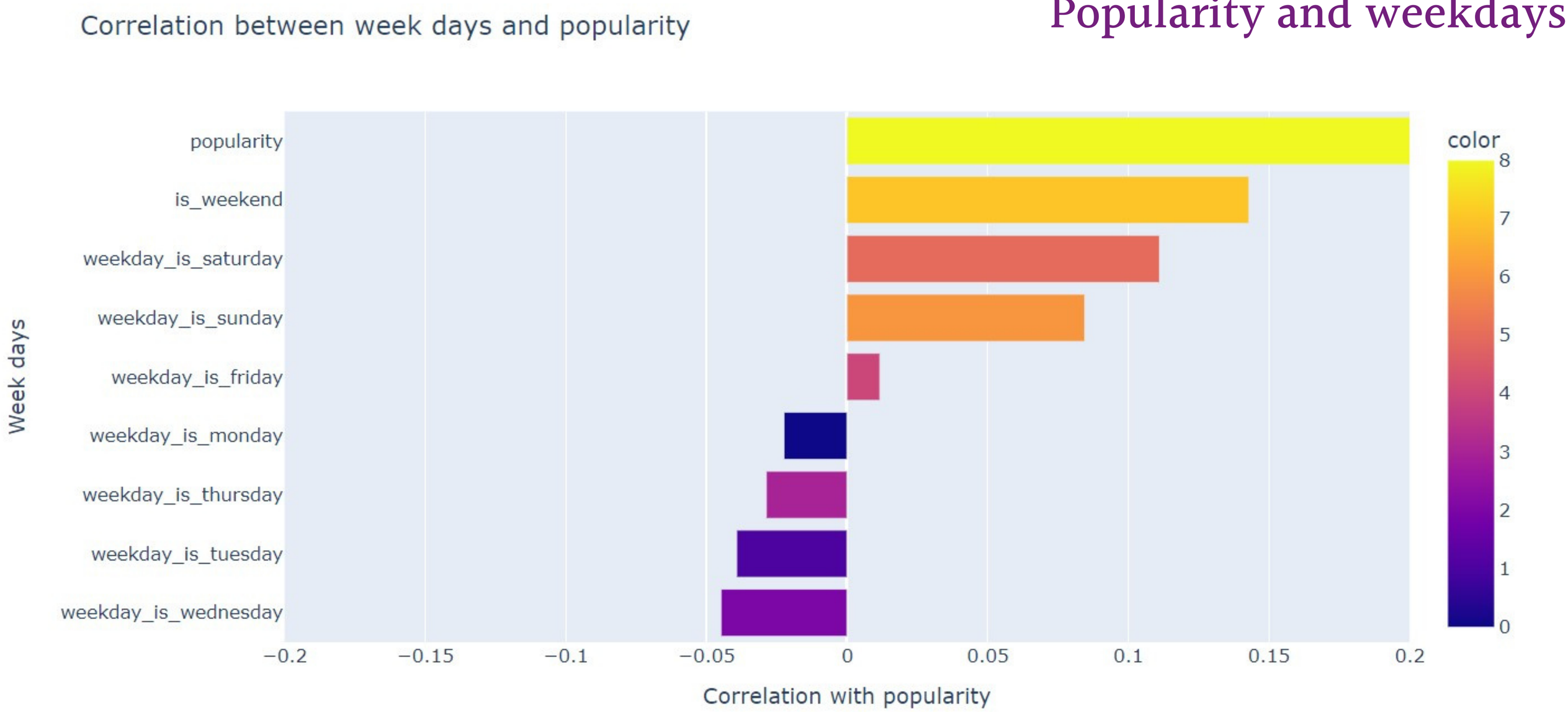
Correlations

Popularity and subject



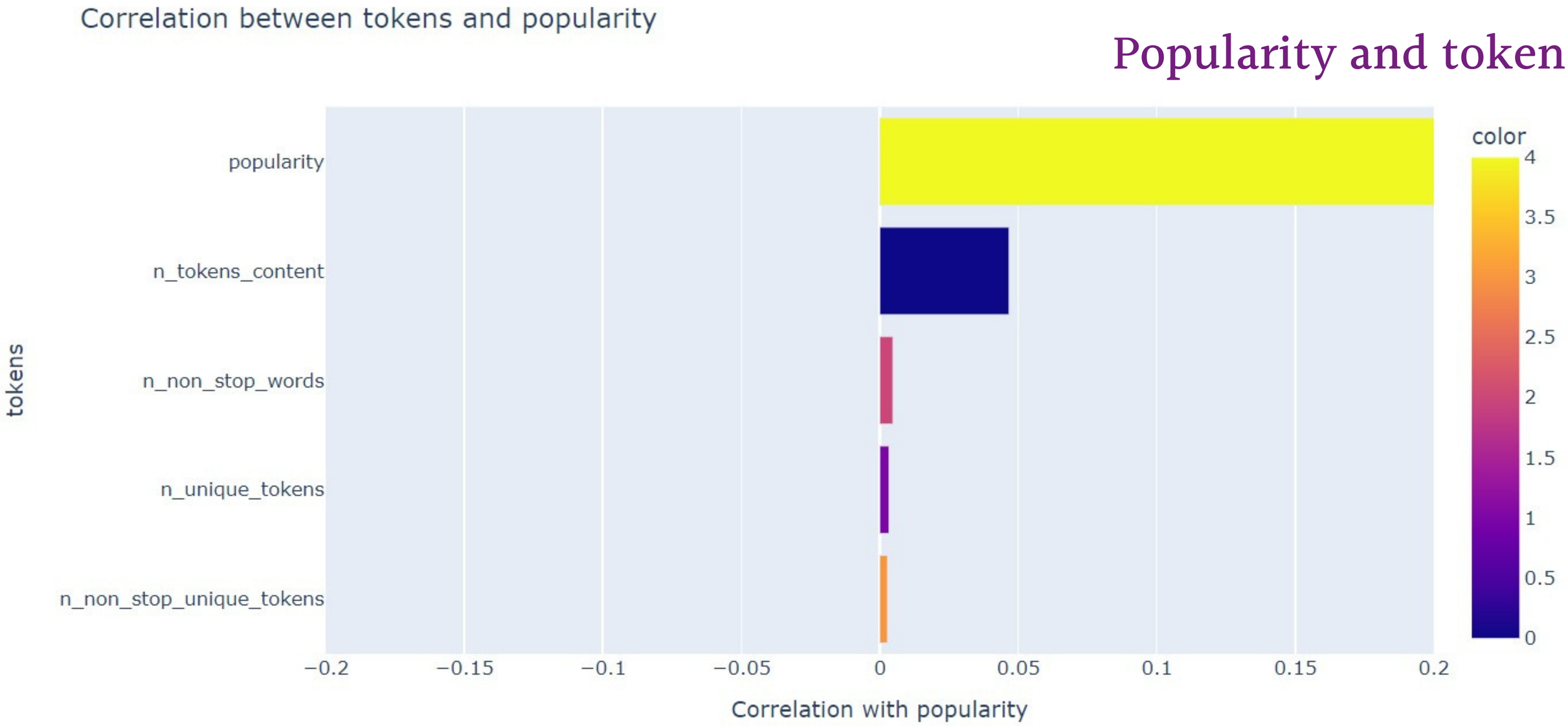
Correlations

Popularity and weekdays



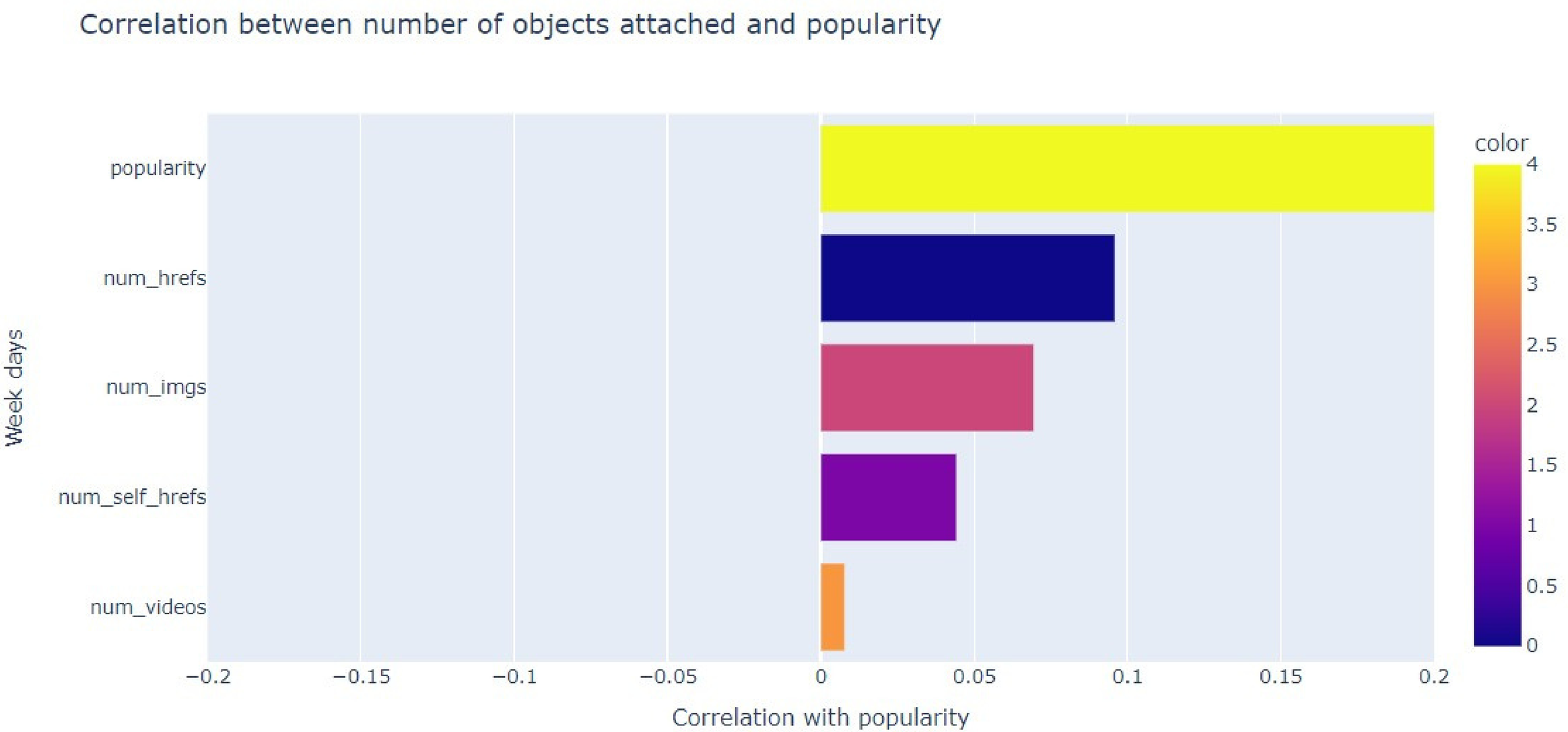
Correlations

Popularity and tokens



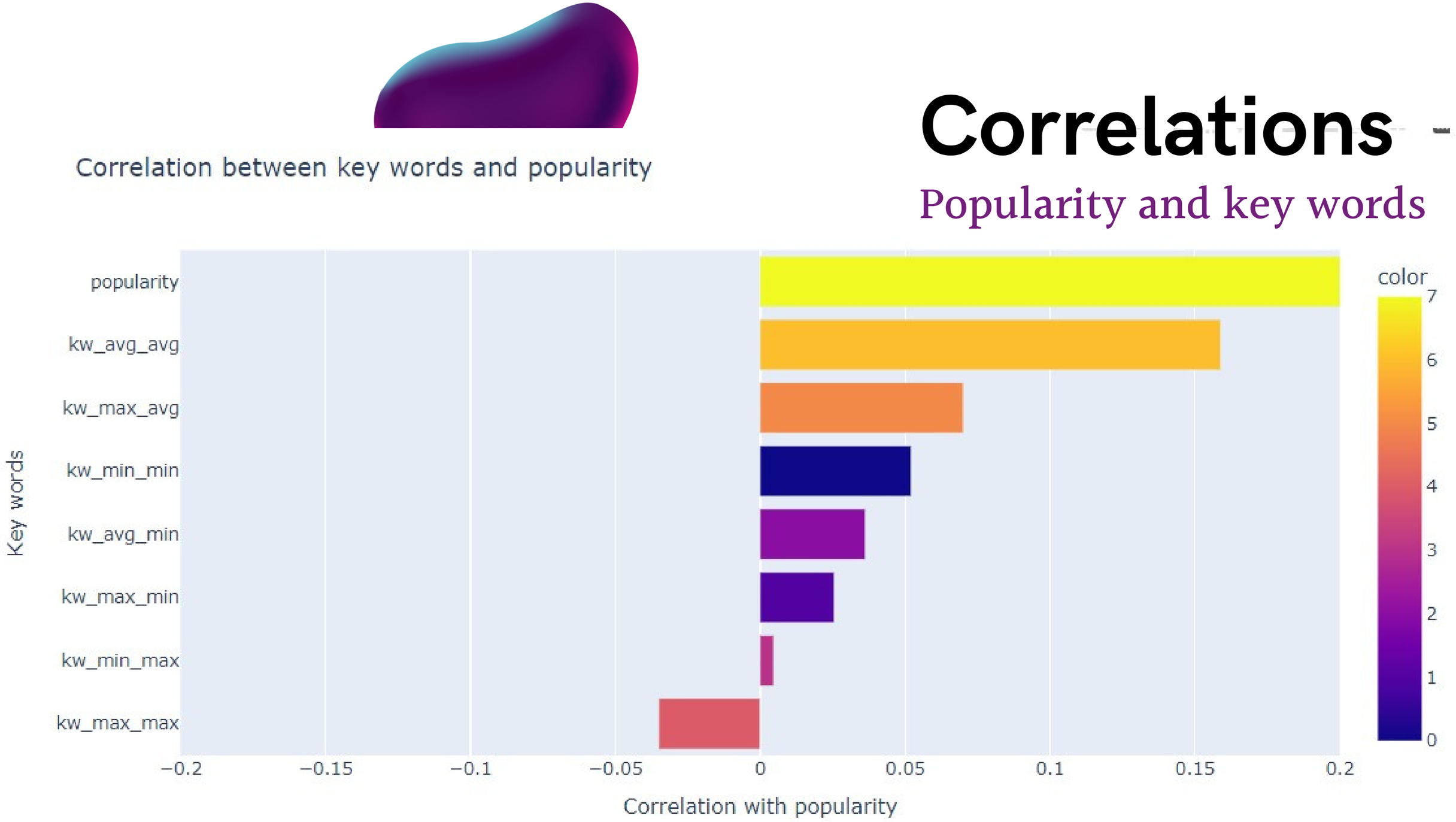
Correlations

Popularity and number of attached objects



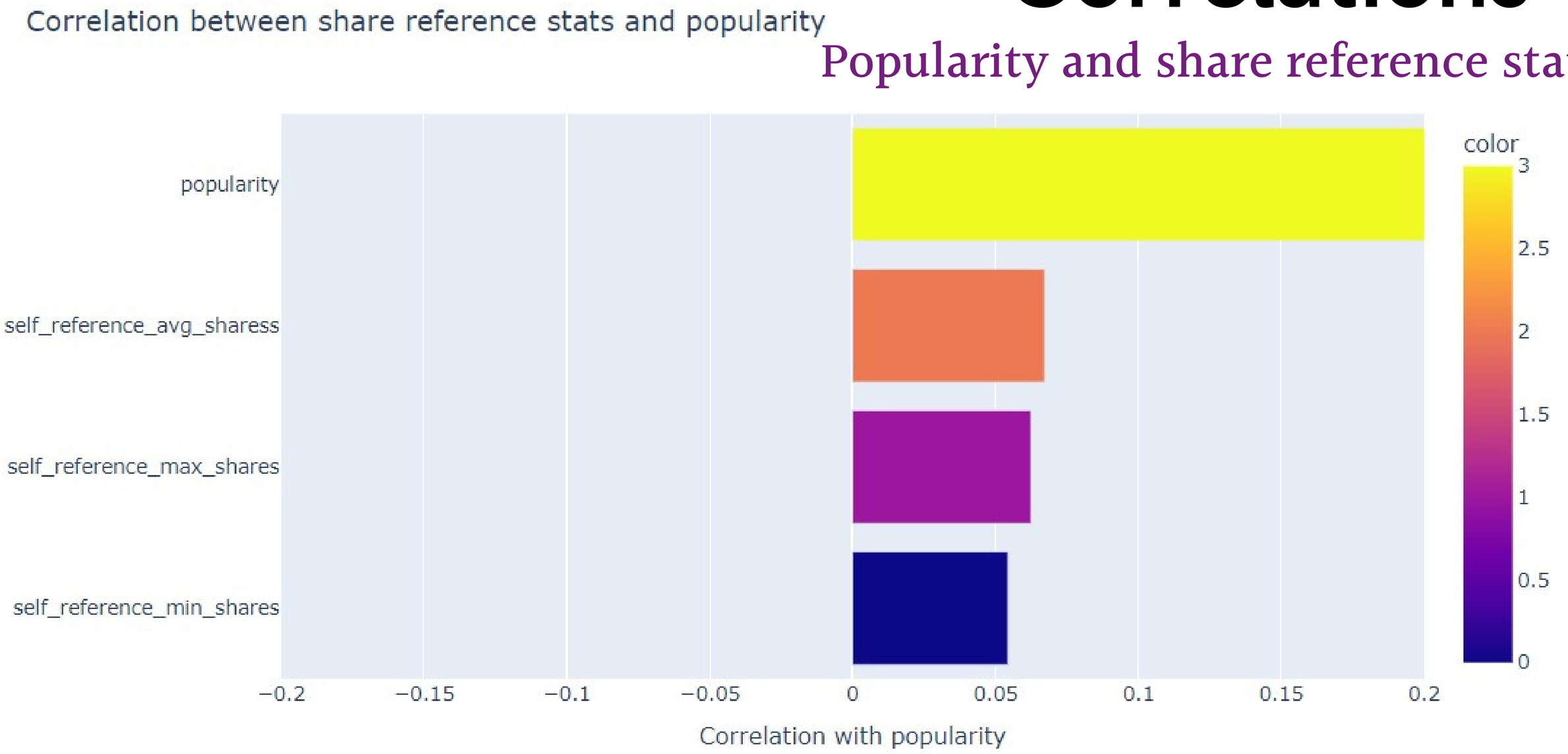
Correlations

Popularity and key words



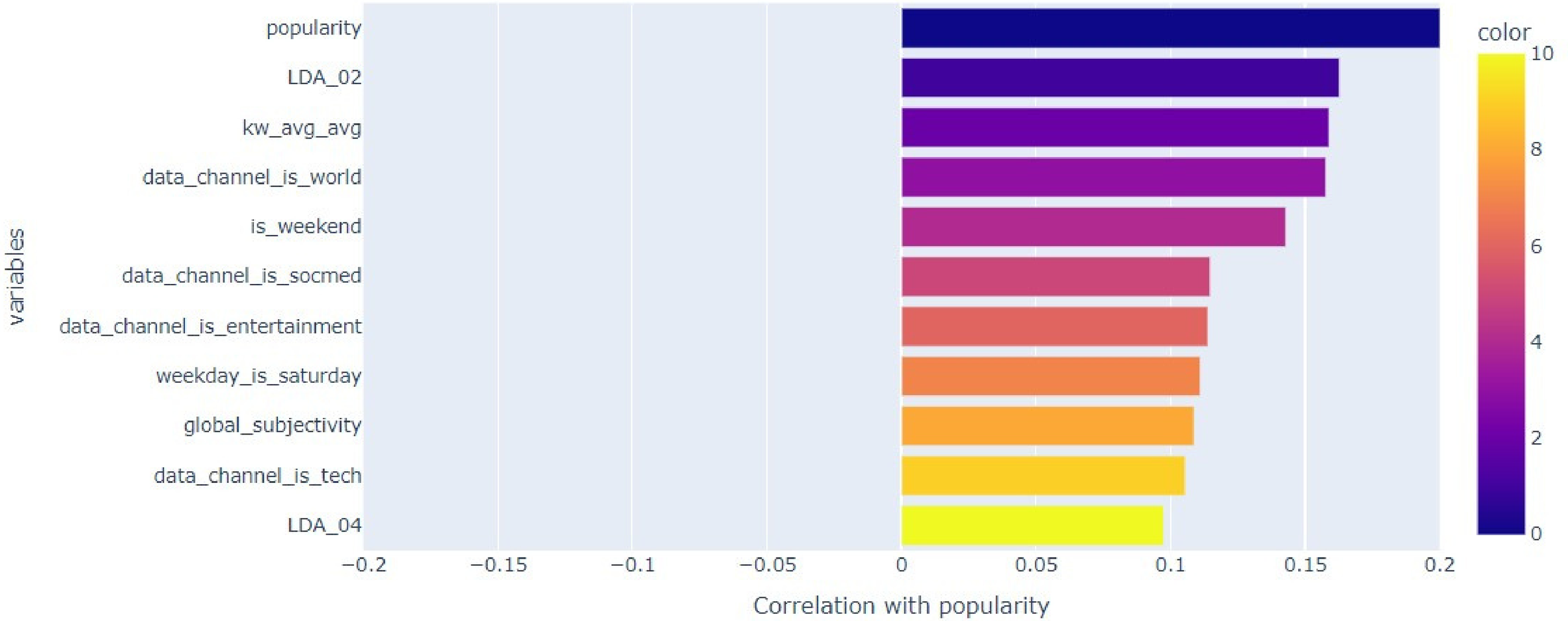
Correlations -

Popularity and share reference stats

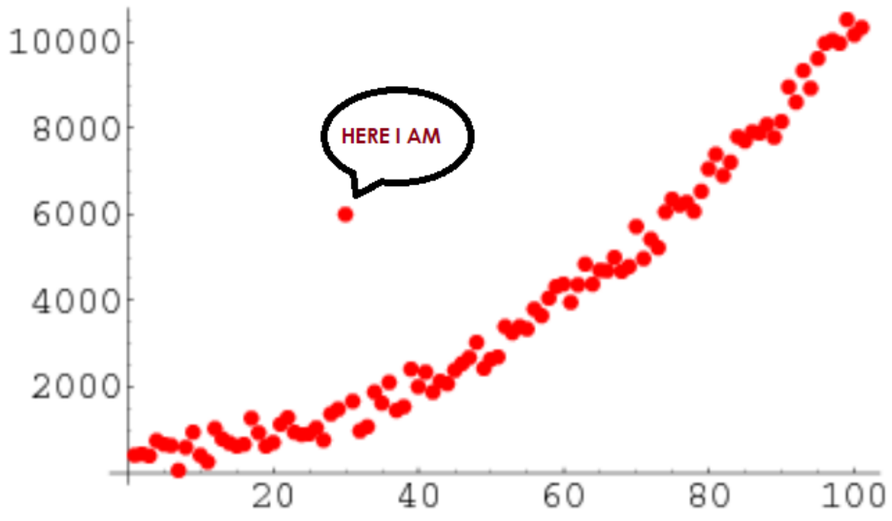


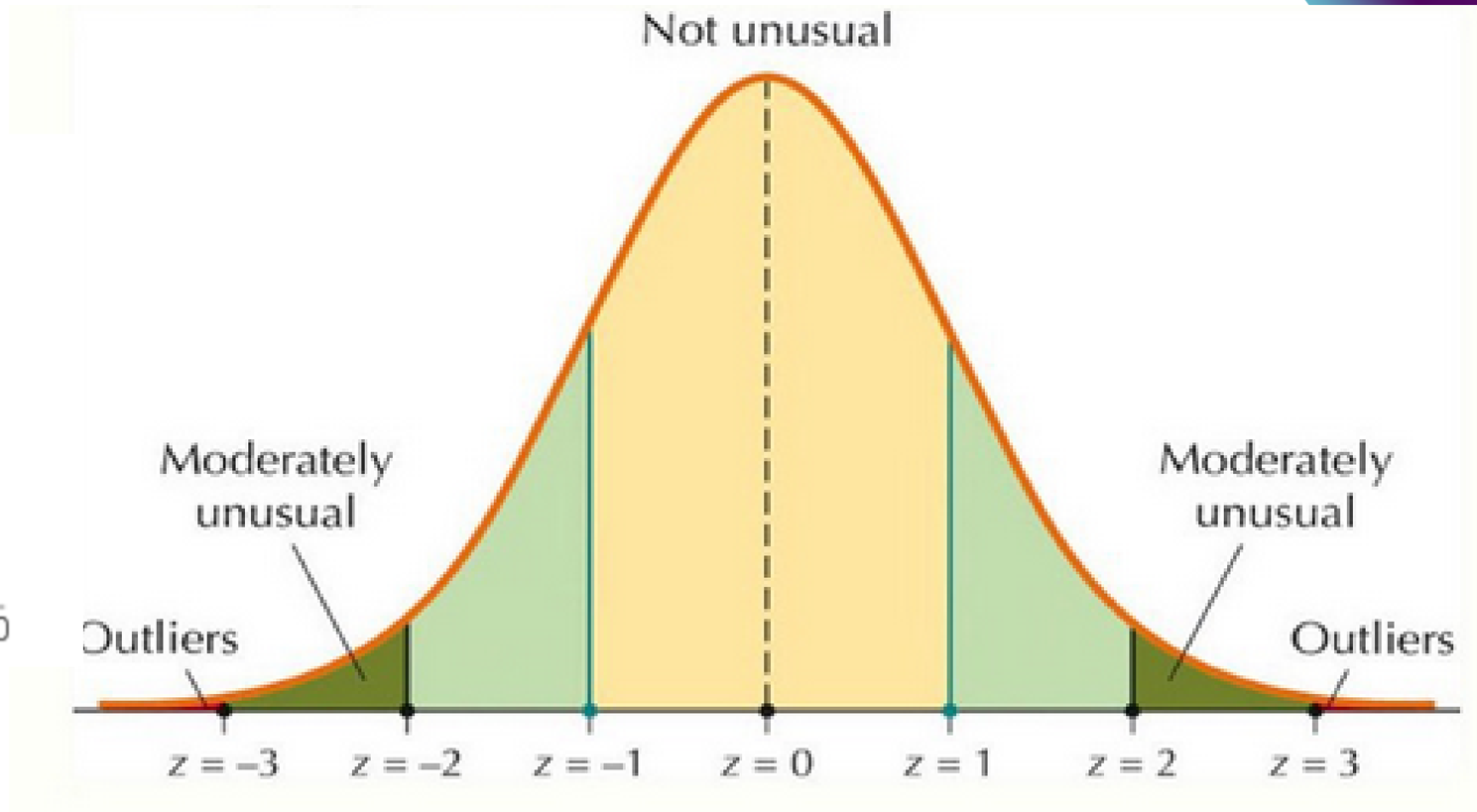
Correlations

Most correlated variables



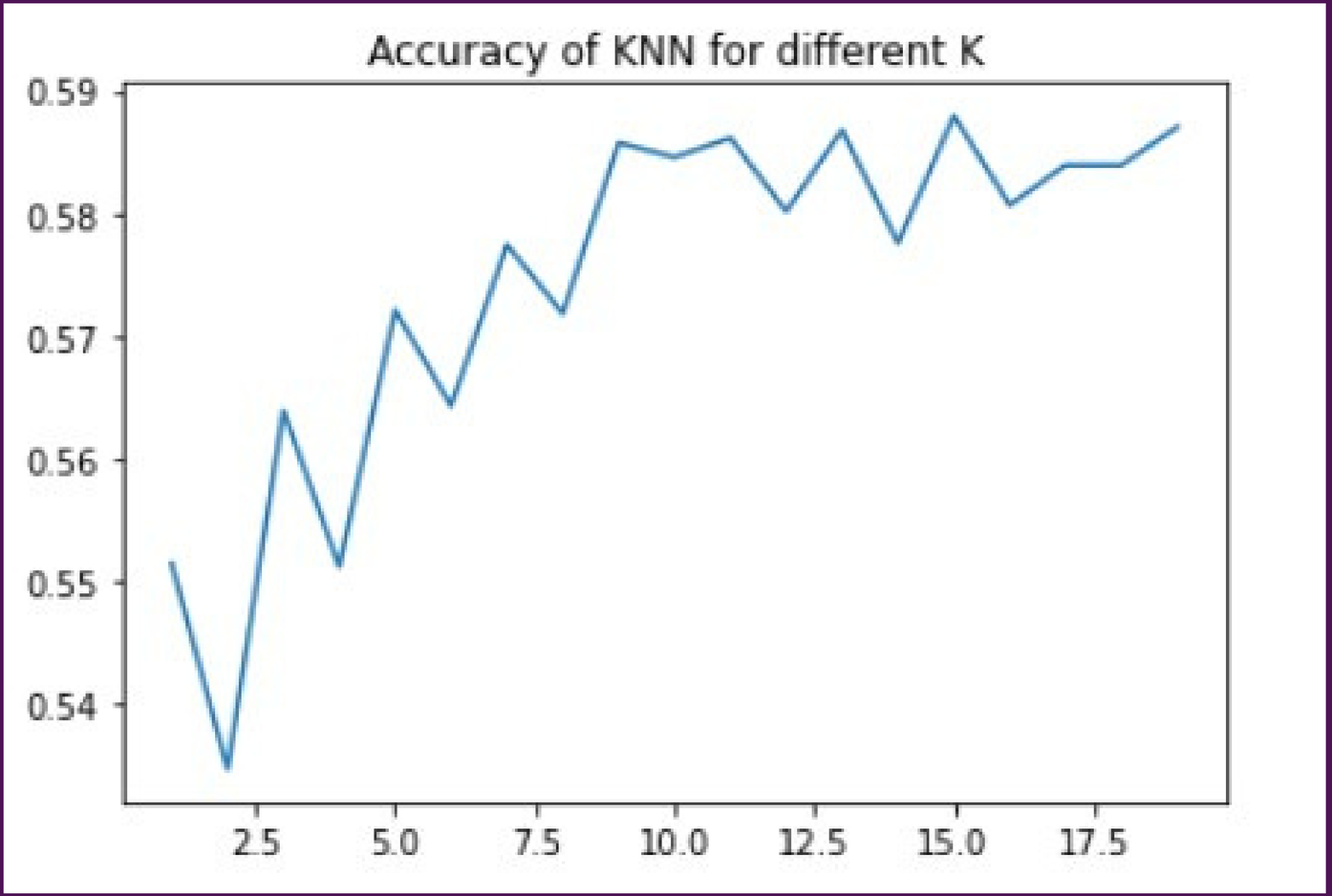
Outliers Removal and data split





Application models (All dataset without outliers)

KNN



With default parameters

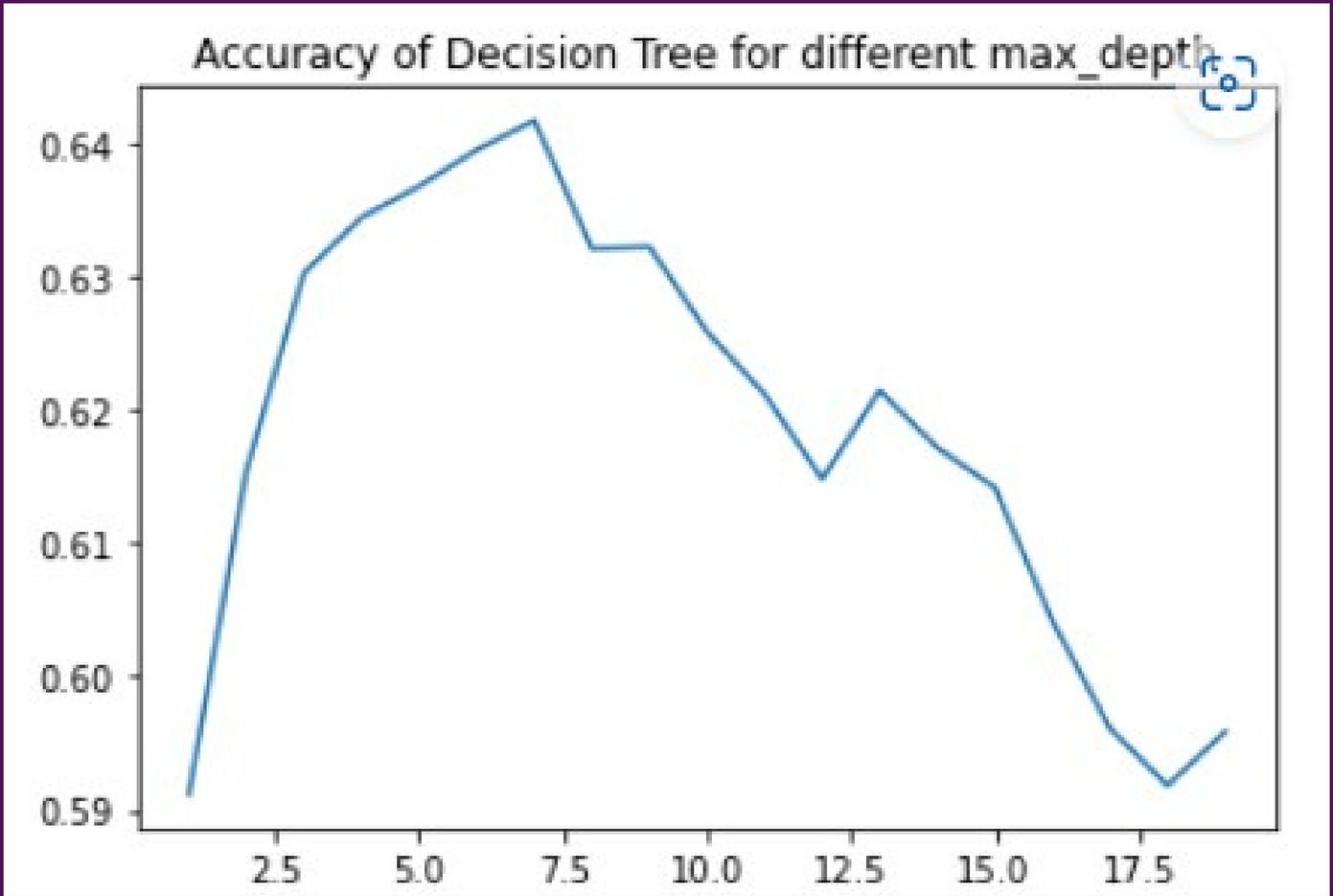
Accuracy = 0.561

With best K :

Accuracy = 0.575

Application models (All dataset without outliers)

Decision Tree



With default parameters

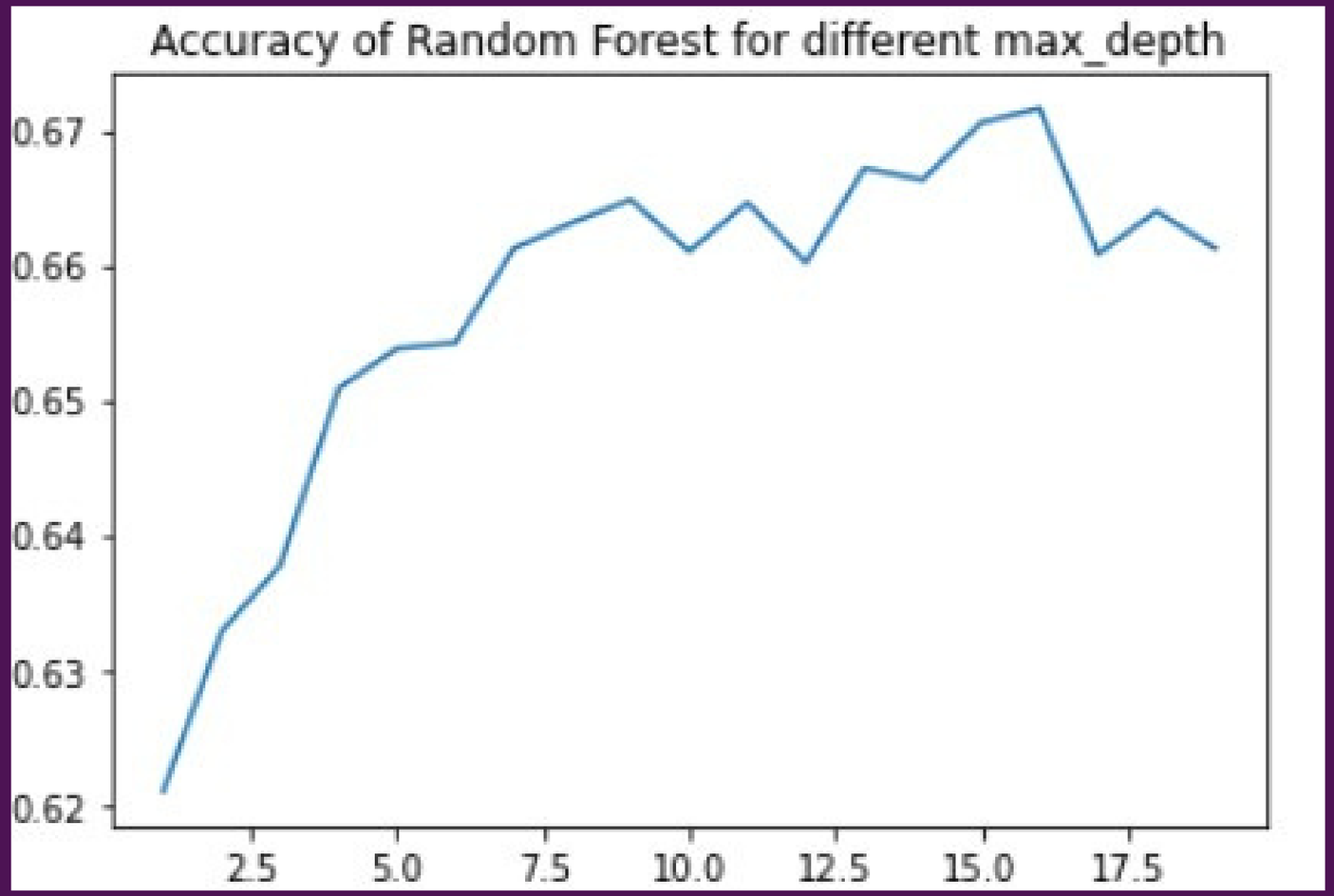
Accuracy = 0.583

With best max_depth :

Accuracy = 0.643

Application models (All dataset without outliers)

Random Forest



With default parameters

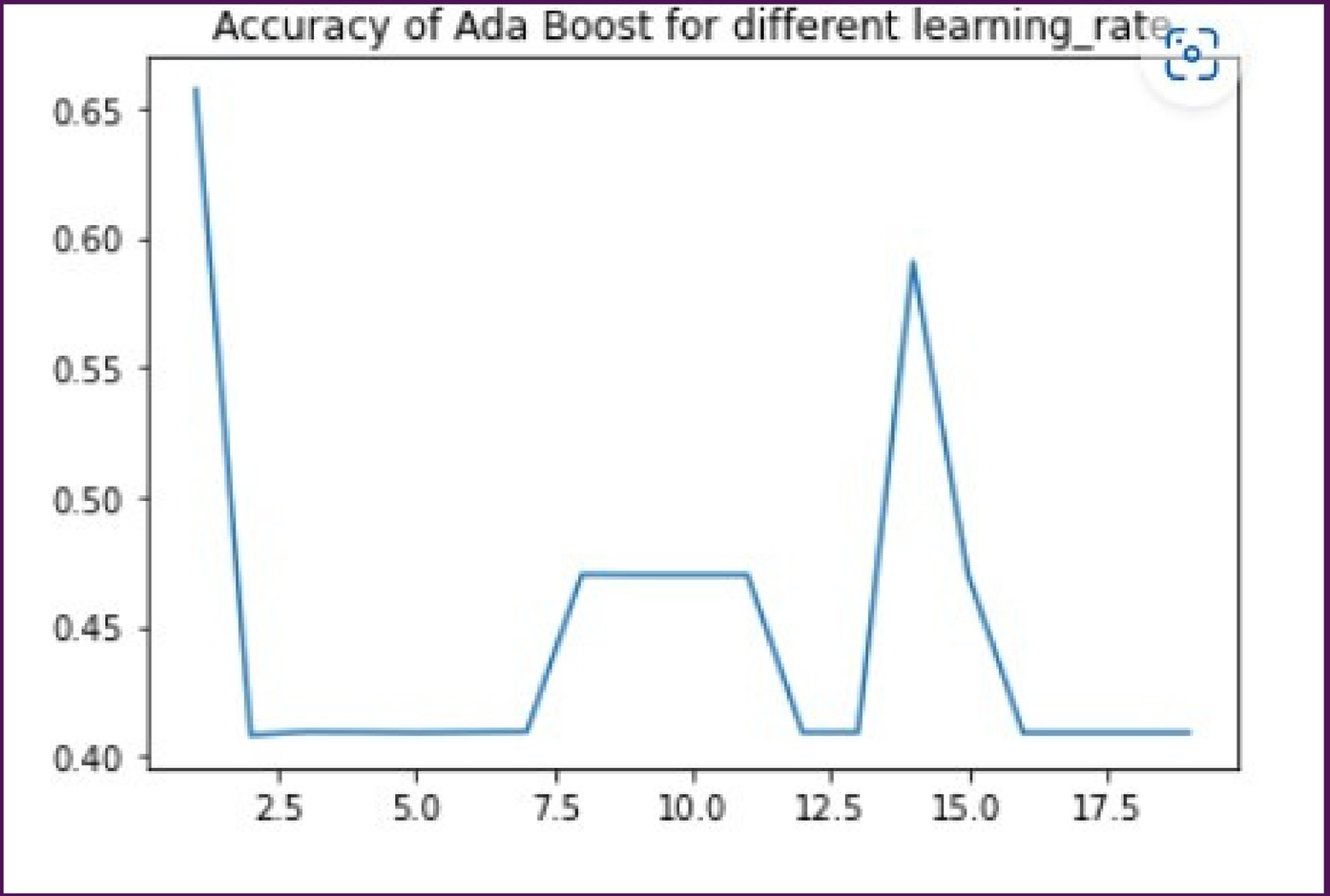
Accuracy = 0.632

With best max_depth :

Accuracy = 0.671

Application models (All dataset without outliers)

Ada Boost



With default parameters

65Accuracy = 0.665

With best learning rate :

Accuracy = 0.665

Application models Summary

(All dataset without outliers)

Model	Default Accuracy	Best Accuracy
Random forest	63%	67%
Ada boost	66%	66%
Desicion tree	58%	64%
Knn	56%	57%

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest (RF)	**0.67**	0.67	**0.71**	**0.69**	**0.73**
Adaptive Boosting (AdaBoost)	0.66	0.68	0.67	0.67	0.72
Support Vector Machine (SVM)	0.66	0.67	0.68	0.68	0.71
K-Nearest Neighbors (KNN)	0.62	0.66	0.55	0.60	0.67
Naïve Bayes (NB)	0.62	**0.68**	0.49	0.57	0.65

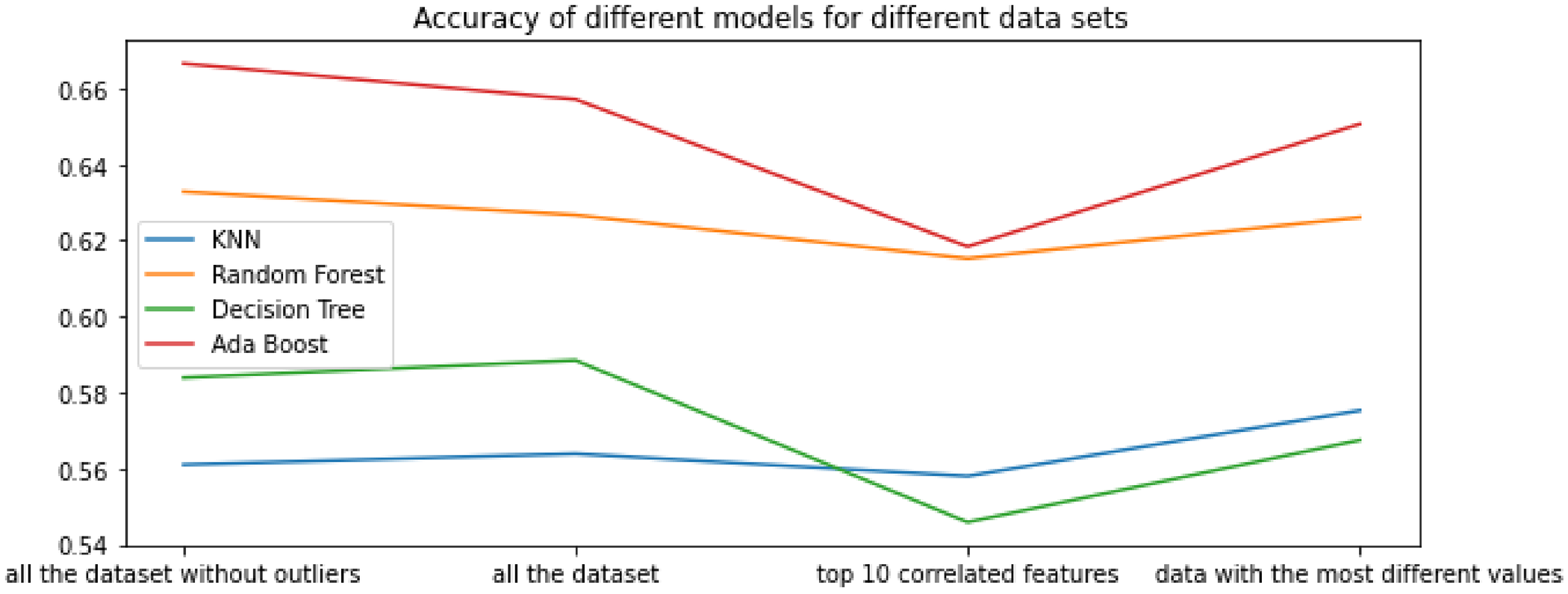
Application models Summary

(Changes of dataset, default parameters)

Model	All without outler	All with outliers	High Correlation Variables	Most different values
Random forest	63.2%	62.6%	60.6%	62.3%
Ada boost	66.6%	65.7%	64%	64.7%
Desicion tree	58.3%	58.8%	56.1%	56.2%
Knn	56.1%	56.3%	56.5%	57.4%

Application models Summary

(Changes of dataset, default parameters)



End