Online News Popularity Data Analysis

Megan Steinmasel, Michael Belmear, and Rohan Kamat

Part One: Overview

Problem Statement: Analyzing Factors Influencing Online News Popularity

In the era of digital media, the popularity of online news articles plays a crucial role in determining reach and impact. To better understand the factors influencing the popularity of news articles, this project aims to analyze the Online News Popularity dataset obtained from UCI's machine learning repository. The dataset contains various features extracted from news articles published by Mashable (www.mashable.com) over a period of two years, along with their corresponding popularity metrics.

Stakeholders Involved: Beneficiaries of Insights from Online News Popularity Analysis

Stakeholders, including content creators, publishers, digital marketers, advertisers, and consumers, stand to benefit from insights gained. These insights can inform content creation, marketing strategies, ad targeting, and content recommendations, ultimately leading to more engaging and relevant online news experiences.

Summary of Dataset: Overview of the Online News Popularity Dataset

The Online News Popularity dataset, sourced from UCI's machine learning repository, comprises various features extracted from news articles published by Mashable over a two-year period. It includes a total of 61 attributes, encompassing both textual and non-textual features. These attributes cover a wide range of factors, such as the type of article content, publication time, and social media shares. The target variable in this dataset is the popularity of news articles, typically measured by the number of shares.

Methodology: Hypothesis Testing & Correlation Analysis

As a key aspect of our comprehensive data analysis, we will employ hypothesis testing to investigate the potential impact of various features on one another within the Online

News Popularity dataset. This approach will provide valuable insights into the relationships between features and their influence on article popularity.

To assess feature interactions, we will utilize bootstrapping techniques to generate multiple simulated samples and calculate sample statistics. From these resampled datasets, we will estimate confidence intervals for the relationships between pairs of features. Concurrently, we will compute p-values to evaluate the statistical significance of observed feature interactions. Lower p-values indicate stronger evidence against the null hypothesis, suggesting significant relationships between features. These methods, including bootstrapping, confidence interval estimation, and p-value calculation, will collectively inform our analysis and interpretation of feature impacts.

In addition to hypothesis testing, we will conduct correlation analysis to further investigate the relationship between news article attributes and their popularity. We will generate a correlation matrix to quantify the linear relationship between different pairs of attributes. Visualization of these correlations using heatmaps will provide intuitive insights into which attributes are most strongly correlated. By analyzing these heat maps, we aim to identify key factors driving the popularity of online news articles.

Part Two: Data Analysis

Introduction

We will be using Jupyter Lab to write Python code for data analysis. We downloaded the UCI Online News Popularity dataset and will be running experiments on the ‘OnlineNewsPopularity.csv’ file. The features and their indexes are as follows:

```
Column 0: url
Column 1: timedelta
Column 2: n_tokens_title
Column 3: n_tokens_content
Column 4: n_unique_tokens
Column 5: n_non_stop_words
Column 6: n_non_stop_unique_tokens
Column 7: num_hrefs
Column 8: num_self_hrefs
Column 9: num_imgs
Column 10: num_videos
Column 11: average_token_length
Column 12: num_keywords
Column 13: data_channel_is_lifestyle
Column 14: data_channel_is_entertainment
Column 15: data_channel_is_bus
Column 16: data_channel_is_socmed
```

Column 17: data_channel_is_tech
Column 18: data_channel_is_world
Column 19: kw_min_min
Column 20: kw_max_min
Column 21: kw_avg_min
Column 22: kw_min_max
Column 23: kw_max_max
Column 24: kw_avg_max
Column 25: kw_min_avg
Column 26: kw_max_avg
Column 27: kw_avg_avg
Column 28: self_reference_min_shares
Column 29: self_reference_max_shares
Column 30: self_reference_avg_sharess
Column 31: weekday_is_monday
Column 32: weekday_is_tuesday
Column 33: weekday_is_wednesday
Column 34: weekday_is_thursday
Column 35: weekday_is_friday
Column 36: weekday_is_saturday
Column 37: weekday_is_sunday
Column 38: is_weekend
Column 39: LDA_00
Column 40: LDA_01
Column 41: LDA_02
Column 42: LDA_03
Column 43: LDA_04
Column 44: global_subjectivity
Column 45: global_sentiment_polarity
Column 46: global_rate_positive_words
Column 47: global_rate_negative_words
Column 48: rate_positive_words
Column 49: rate_negative_words
Column 50: avg_positive_polarity
Column 51: min_positive_polarity
Column 52: max_positive_polarity
Column 53: avg_negative_polarity
Column 54: min_negative_polarity
Column 55: max_negative_polarity
Column 56: title_subjectivity
Column 57: title_sentiment_polarity
Column 58: abs_title_subjectivity
Column 59: abs_title_sentiment_polarity
Column 60: share

Basic Data Analysis

For our basic data analysis, Megan S. will focus on two key columns: Column 60, which represents the number of shares, and Column 9, which indicates the number of images in each news article. We have selected these data attributes to gain insight into our problem statement. We hypothesize that the presence of images, being powerful visual aids, may increase the popularity of online news articles. By examining the relationship between the number of shares and the number of images, we aim to shed light on this potential relationship.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Load the data
data = pd.read_csv("OnlineNewsPopularity.csv", header=0)

# Extract columns
cols = data.iloc[:, [9, -1]]

# 1. Descriptive Statistics
print("Descriptive Statistics:")
print(cols.describe())
=>
```

```
Descriptive Statistics:
          num_imgs         shares
count  39644.000000   39644.000000
mean       4.544143    3395.380184
std        8.309434   11626.950749
min        0.000000       1.000000
25%        1.000000     946.000000
50%        1.000000    1400.000000
75%        4.000000    2800.000000
max      128.000000  843300.000000
```

```
# 2. Data Visualization
plt.figure(figsize=(10, 6))
plt.subplot(1, 2, 1)
sns.histplot(cols.iloc[:, 0], kde=True)
plt.title('Number of Images Distribution')

plt.subplot(1, 2, 2)
sns.histplot(cols.iloc[:, 1], kde=True)
plt.title('Shares Distribution')
plt.show()
```

```
=>
```

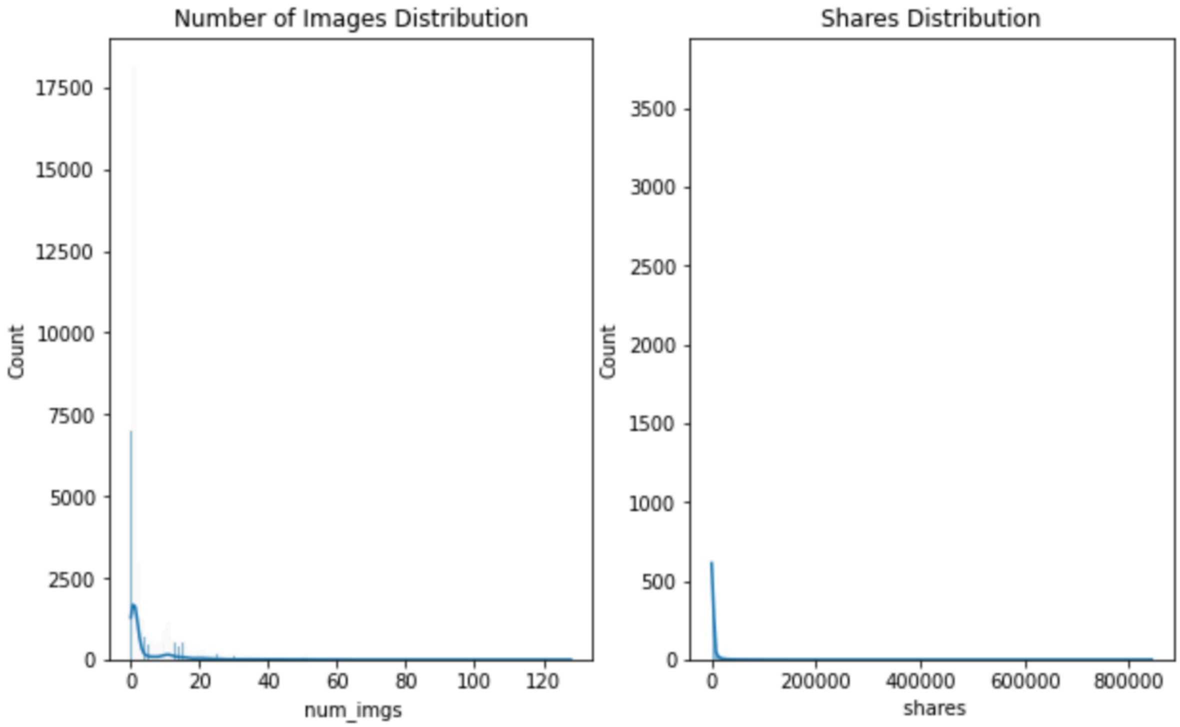


Figure 1

```
# 3. Correlation Analysis
correlation_matrix = cols.corr()
print("\nCorrelation Matrix:")
print(correlation_matrix)
=>
```

```
Correlation Matrix:
          num_imgs    shares
 num_imgs   1.000000  0.039388
 shares     0.039388  1.000000
```

```
# 4. Dimensionality Reduction (PCA)
pca = PCA(n_components=2)
pca_result = pca.fit_transform(cols)
```

```
# 5. Clustering (KMeans)
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(cols)
labels = kmeans.labels_

# Plotting the clusters
plt.figure(figsize=(8, 6))
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=labels,
cmap='viridis')
plt.title('Clustering Result')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()
=>
```

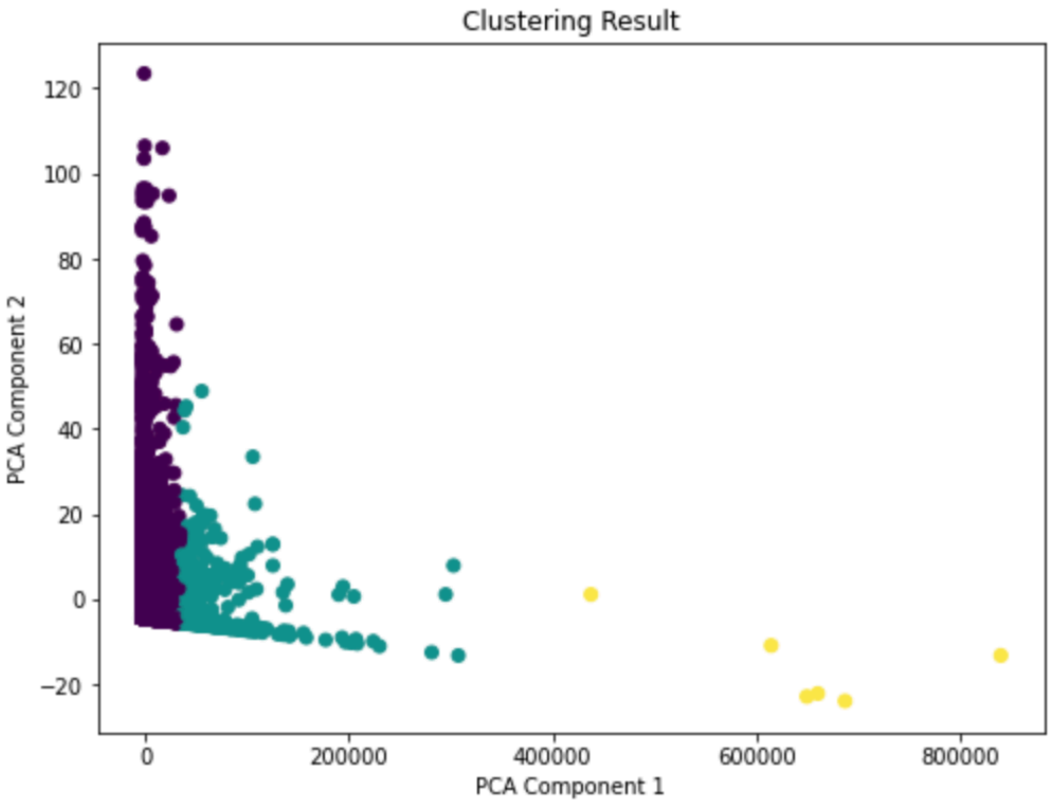


Figure 2

Hypothesis Testing

To delve deeper into the research inquiry regarding whether the inclusion of images enhances the popularity of online news articles, Megan S. will employ inference for two quantitative variables through hypothesis testing. This will entail crafting a bootstrap interval and determining the associated p-value.

- Research Question: We hypothesize that the presence of images, being powerful visual aids, may increase the popularity of online news articles.
- Explanatory: Number of images (num_imgs).
- Response: Number of shares (shares).
- Null Hypothesis: There is no true linear relationship between number of images on an article and the number of shares an article has.
- Alternative Hypothesis: There is a true positive linear relationship between images on an article and the number of shares an article has.

Below is the scatter plot illustrating the relationship between the number of images (num_imgs) and the number of shares. Additionally, the analysis includes the calculation of the slope, y-intercept, and correlation coefficient. Furthermore, the simulation p-value and confidence interval are provided to assess the significance of the observed relationship.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the data
data = pd.read_csv("OnlineNewsPopularity.csv", header=0)

# Select columns 9 and 10
cols = data.iloc[:, [9, 10]]

# Extract the columns
x = cols.iloc[:, 0]
y = cols.iloc[:, 1]

# Create scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(x, y, alpha=0.5, c='blue', label='Data')

# Add linear regression line
coefficients = np.polyfit(x, y, 1)
polynomial = np.poly1d(coefficients)
```

```
plt.plot(x, polynomial(x), 'r', label='Regression line')

plt.title("Relationship between 'num_imgs' and 'shares'")
plt.xlabel('num_imgs')
plt.ylabel('shares')
plt.legend()
plt.grid(True)
plt.show()
=>
```

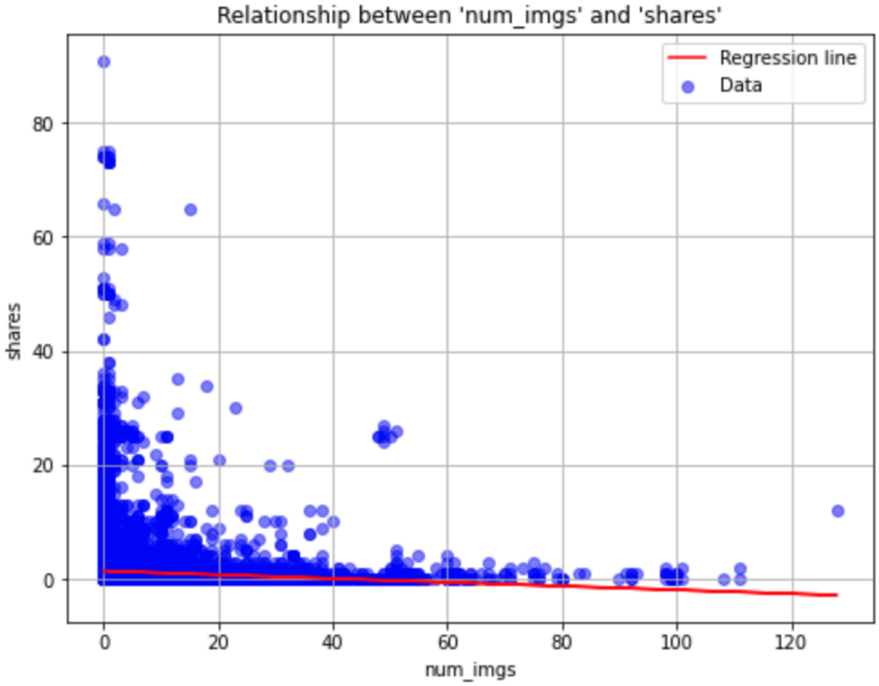


Figure 3

```
# Calculate regression statistics
slope, intercept, r_value, p_value, std_err = linregress(x, y)

# Output the results
print("Slope of the regression line:", slope)
print("Y-intercept of the regression line:", intercept)
print("Correlation coefficient:", r_value)
=>
```

```
Slope of the regression line: -0.033
Y-intercept of the regression line: 1.401
Correlation coefficient: -0.067
```

```
from scipy.stats import linregress
import numpy as np
import matplotlib.pyplot as plt

# Define the parameters
direction = 'greater'
summary_measure = 'slope'
as_extreme_as = -0.033
num_repetitions = 1000
confidence_level = 0.95

# Initialize a list to store the test statistics
test_statistics = []

# Perform the simulation test
for _ in range(num_repetitions):
    # Generate random indices for bootstrapping
    indices = np.random.choice(len(x), len(x), replace=True)
    # Perform linear regression on bootstrapped sample
    slope, intercept, r_value, p_value, std_err =
linregress(x[indices], y[indices])
    test_statistics.append(slope)

# Calculate p-value
if direction == 'greater':
    p_value = np.mean(np.array(test_statistics) >= as_extreme_as)
elif direction == 'less':
    p_value = np.mean(np.array(test_statistics) <= as_extreme_as)
else:
    p_value = np.mean(np.abs(np.array(test_statistics)) >=
np.abs(as_extreme_as))

# Calculate confidence interval
lower_bound = np.percentile(test_statistics, (1 - confidence_level) /
2 * 100)
upper_bound = np.percentile(test_statistics, (1 + confidence_level) /
2 * 100)

# Plot the distribution and confidence interval
plt.figure(figsize=(8, 6))
plt.hist(test_statistics, bins=30, density=True, alpha=0.5,
color='blue')
```

```
plt.axvline(x=as_extreme_as, color='red', linestyle='--',
label='Extreme Value')
plt.axvline(x=lower_bound, color='green', linestyle='--',
label='Lower Bound of CI')
plt.axvline(x=upper_bound, color='green', linestyle='--',
label='Upper Bound of CI')
plt.title('Distribution of Test Statistics with Confidence Interval')
plt.xlabel('Test Statistic')
plt.ylabel('Density')
plt.legend()
plt.grid(True)
plt.show()

# Output the p-value and confidence interval
print("Simulation p-value:", p_value)
print("Confidence Interval: ({:.3f}, {:.3f})".format(lower_bound,
upper_bound))
=>
```

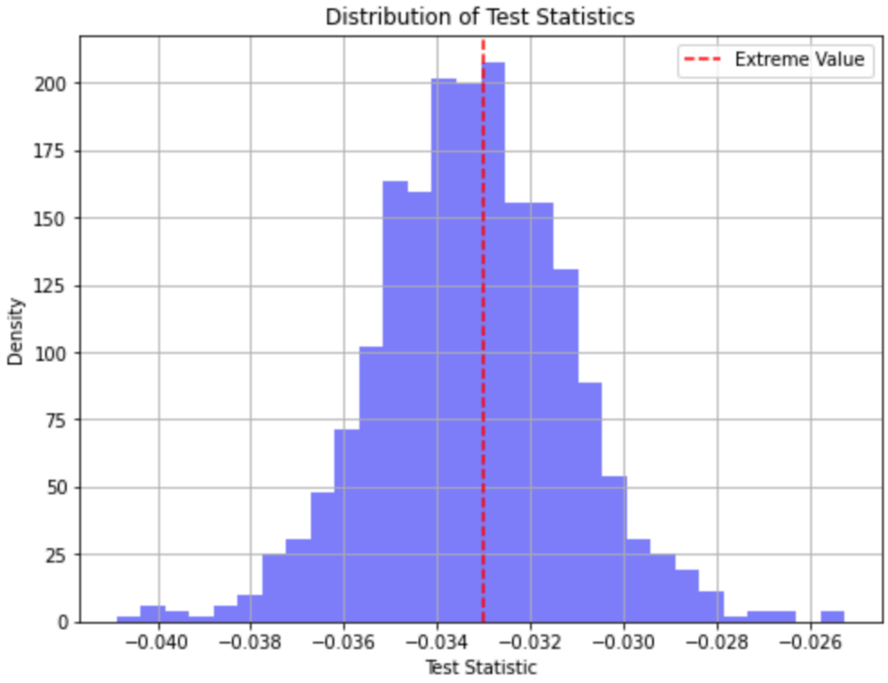


Figure 4

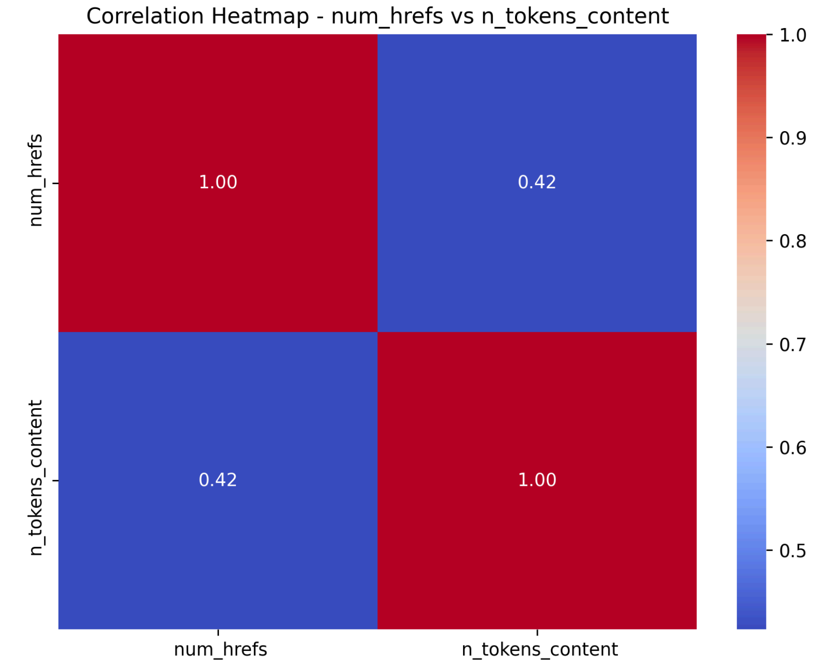
Simulation p-value: 0.456
Confidence Interval: (-0.037, -0.029)

Correlation Heat Maps

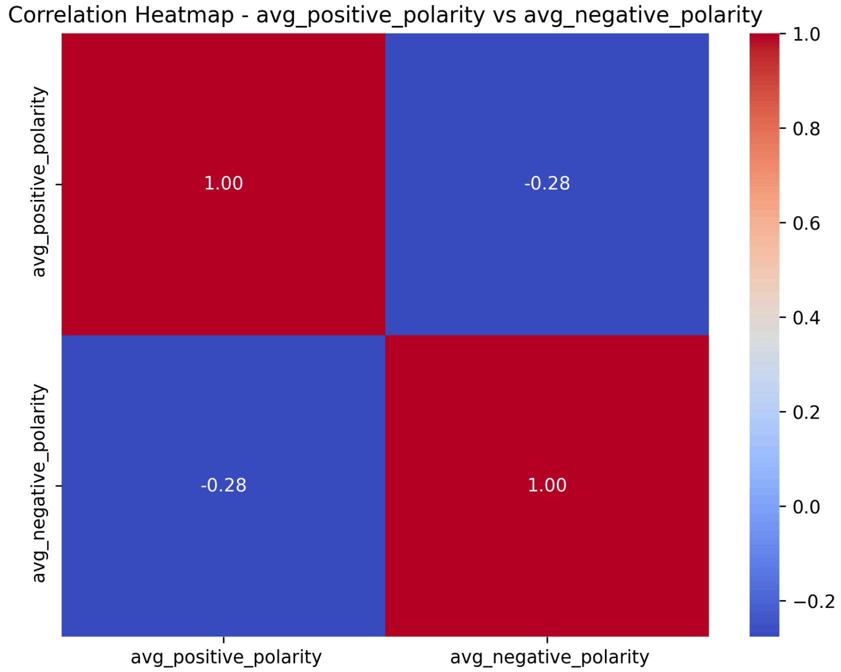
Michael B. and Rohan K. selected pairs of data attributes they believed to be related, and developed code to visualize the correlation between them using correlation heatmaps, enhancing our data analysis capabilities. Below is the code utilized by Michael B. and Rohan K. to generate correlation heatmaps, along with the corresponding output showcasing the relationships between the selected data attributes.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = pd.read_csv("C:/Users/akmik/OneDrive/Desktop/CSCI 347/Final
Project/OnlineNewsPopularity.csv", header=0)

# Assuming you have loaded your data into a DataFrame named 'data'
cols1 = data.iloc[:, [7, 3]]  # Selecting 'num_hrefs' and
'n_tokens_content'

# Compute the correlation matrix
correlation_matrix1 = cols1.corr()

# Plot the heatmap for the first pair of features
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix1, annot=True, cmap='coolwarm',
fmt=".2f")
plt.title('Correlation Heatmap - num_hrefs vs n_tokens_content')
plt.show()
# Now let's choose another pair of features, for example,
'avg_positive_polarity' and 'avg_negative_polarity'
cols2 = data.iloc[:, [50, 53]]  # Selecting 'avg_positive_polarity'
and 'avg_negative_polarity'

# Compute the correlation matrix for the second pair of features
correlation_matrix2 = cols2.corr()

# Plot the heatmap for the second pair of features
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix2, annot=True, cmap='coolwarm',
fmt=".2f")
```

```
plt.title('Correlation Heatmap - avg_positive_polarity vs
avg_negative_polarity')
plt.show()
=>
```

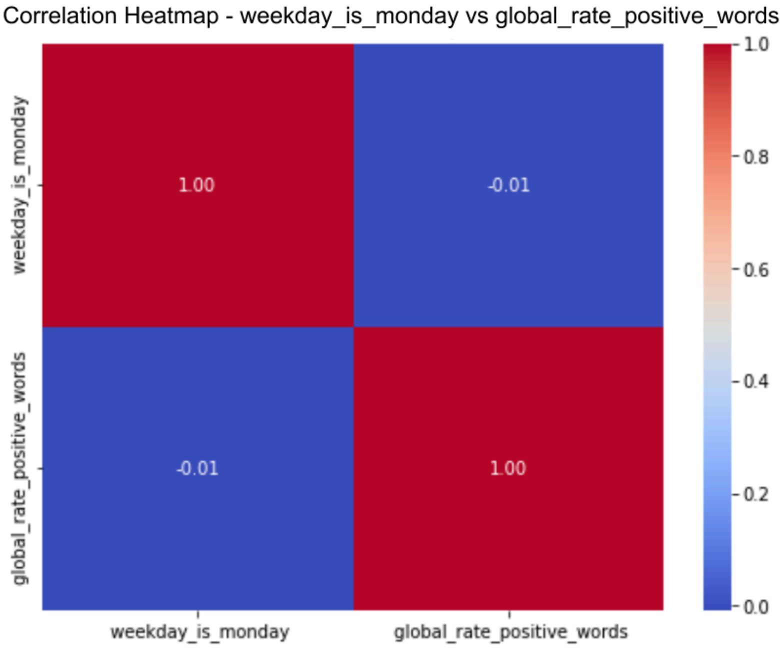


Heatmap 1 (Michael)

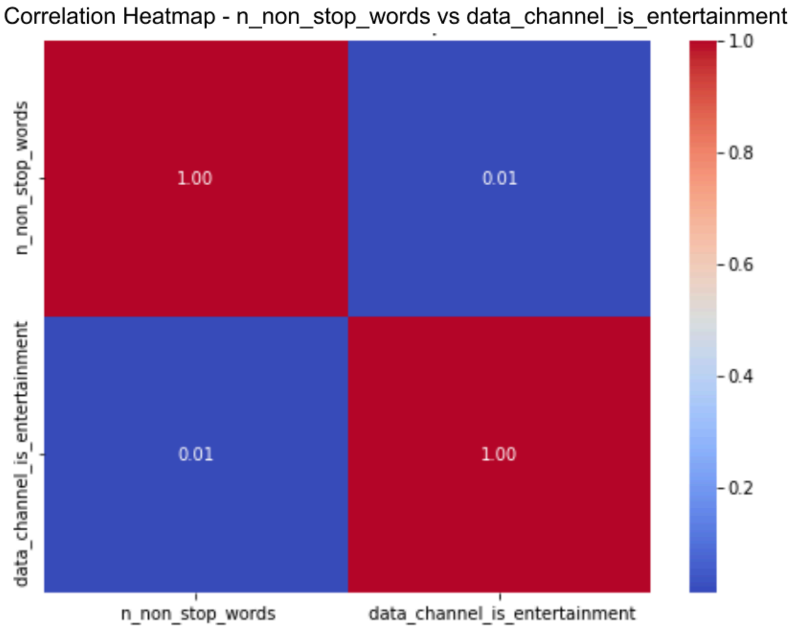


Heatmap 2 (Michael)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = pd.read_csv("OnlineNewsPopularity.csv", header=0)

# Assuming you have loaded your data into a DataFrame named 'data'
cols1 = data.iloc[:, [31,46]]

# Compute the correlation matrix
correlation_matrix = cols1.corr()

# Plot the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()

# Assuming you have loaded your data into a DataFrame named 'data'
cols1 = data.iloc[:, [5,14]]

# Compute the correlation matrix
correlation_matrix = cols1.corr()

# Plot the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
=>
```



Heatmap 3 (Rohan)

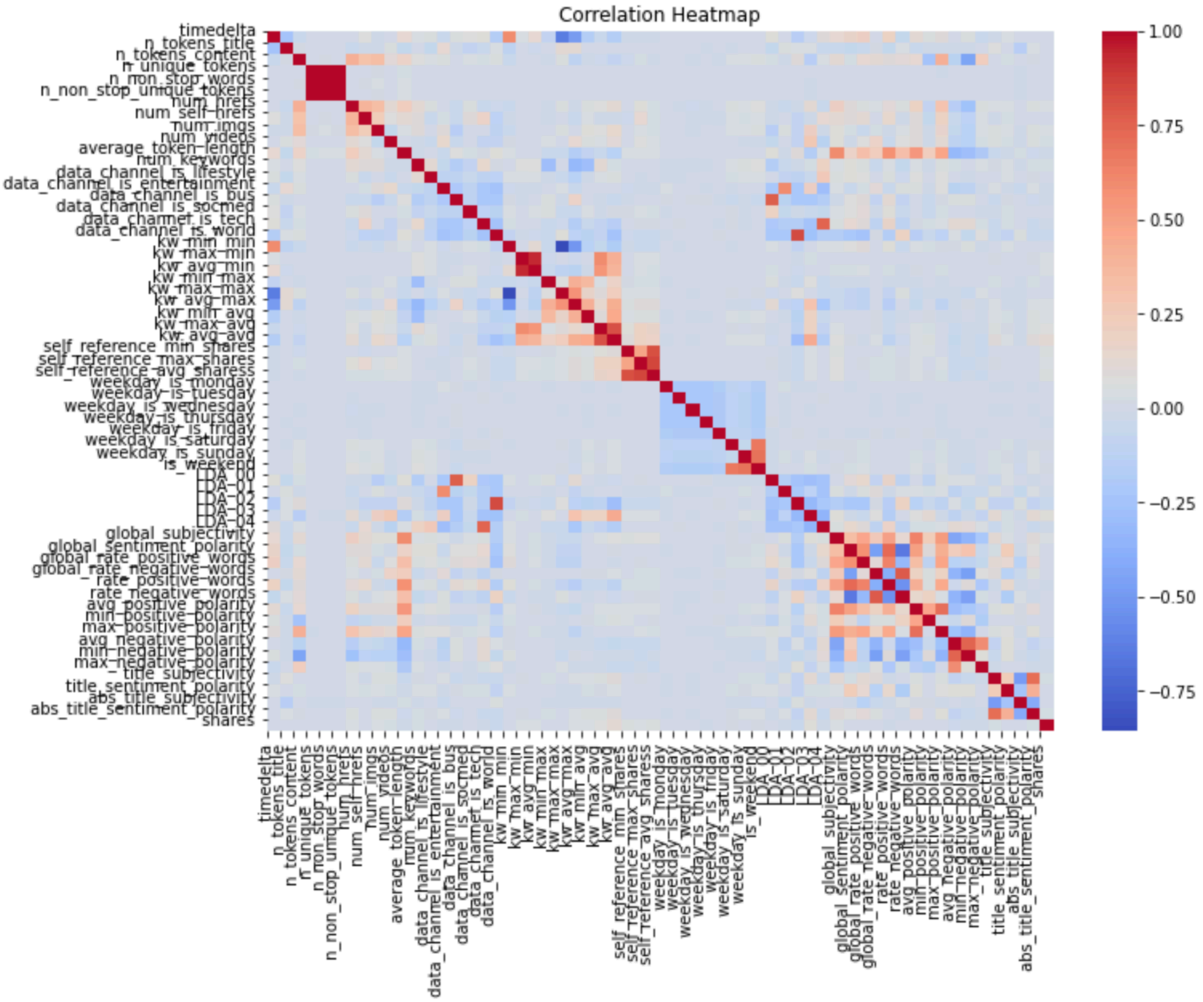


Heatmap 4 (Rohan)

Total Correlation

To deepen the data analysis, Megan S. devised a total correlation heatmap to visualize all attributes and their correlations. Below, you'll find Megan's code alongside the resulting visualization, offering a comprehensive understanding of the dataset's interconnections.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the data
data =
pd.read_csv("/Users/megansteinmasel/Desktop/OnlineNewsPopularity/Onli
neNewsPopularity.csv", header=0)

# Exclude the first column (URL column)
data_no_url = data.iloc[:, 1:]

# Compute the correlation matrix
correlation_matrix = data_no_url.corr()

# Plot the heatmap with all attributes as axis labels
plt.figure(figsize=(11, 8))
sns.heatmap(correlation_matrix, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.xticks(range(len(data_no_url.columns)), data_no_url.columns,
rotation=90)
plt.yticks(range(len(data_no_url.columns)), data_no_url.columns)
plt.show()
=>
```



Heatmap 5

Part Three: Findings

Basic Data Analysis (Megan S.)

In our basic data analysis, we concentrated on two data attributes: the number of shares (shares) and the number of images in each news article (num_imgs). Our selection of these attributes aimed to provide insights into our problem statement, hypothesizing that the inclusion of images, as potent visual aids, might enhance the popularity of online news articles. Descriptive statistics reveal that the mean number of images is approximately 4.54, while the mean number of shares is around 3395.38. Visualizing the distributions (Figure 1) of shares and images showcased their spread across the dataset. The correlation matrix we found indicates a minimal correlation between the number of images and shares. Employing dimensionality reduction via PCA and clustering using the KMeans algorithm allowed us to discern potential groupings within the data, as illustrated in the clustering result plot (Figure 2). These analyses collectively contribute to our understanding of the relationships between image count, shares, and potential clusters within the dataset.

Hypothesis Testing Results (Megan S.)

Our research question put forward that the presence of images, serving as potent visual aids, might enhance an article's shareability. To explore this relationship more, we examined the number of images (num_imgs) as the explanatory variable and the number of shares (shares) as the response variable. Our null hypothesis suggested that there is no true linear relationship between the number of images on an article and its shares, while the alternative hypothesis proposed a positive linear relationship.

We developed Python code to examine the relationship between the explanatory and response variables, and the resulting scatter plot displayed a weak relationship (Figure 3). Additionally, we created code to output the slope of the regression line and the y-intercept. The slope is -0.033, and the y-intercept is 1.401. This implies that for every one-unit increase in images, there is a 0.033 decrease in total shares for Mashable articles. Based on our data analysis, we derived the following equation for estimated shares: Estimated Shares = 1.401 - 0.033 * Number of Images on Article

We also explored creating a simulation p-value and confidence interval (Figure 4). The 95% confidence interval is (-0.037, -0.029). This confidence interval means that we are 95% confident that every increase of 1 image on a Mashable article is associated with between a predicted 0.037 to 0.029 decrease in shares. The p-value is 0.456, indicating that there is little to no evidence that the true slope of the regression line between number of images and number of shares for Mashable articles is greater than 0. In conclusion, we did not find evidence that images on Mashable articles correlate with more shares.

Correlation Results (Michael B.)

After considering the relationship between the number of images on an article and shares, we moved on to test other data attributes. In considering the relationship between num_hrefs and n_tokens_content, a hypothesis arises: articles containing more content may incorporate a greater number of references or external links to bolster their arguments or provide supplementary details. Consequently, one might anticipate a positive correlation between the quantity of links (num_hrefs) and the word count within the content (n_tokens_content). This hypothesis suggests that as the length of an article increases, so too does the likelihood of including additional references or links to support its content.

In our analysis of num_hrefs and n_tokens_content (Heatmap 1), the correlation coefficient of 0.42 suggests a moderate positive correlation between the quantity of links (num_hrefs) and the word count within the content (n_tokens_content). This finding aligns with the initial hypothesis, indicating that articles with greater content are inclined to feature a higher number of references or external links. Thus, the observed correlation provides empirical support for the notion that articles with more extensive content tend to incorporate a larger number of supplementary resources or citations.

In considering the relationship between avg_positive_polarity and avg_negative_polarity, a hypothesis emerges: articles exhibiting a higher average positive polarity within their text may correspondingly display a lower average negative polarity, and conversely. This supposition stems from the notion that articles conveying positive sentiments often emphasize optimistic aspects, potentially resulting in fewer occurrences of negative language, and conversely for articles with predominantly negative sentiments. Thus, we anticipate a negative correlation between these two features, reflecting an inverse relationship wherein an increase in one variable corresponds to a decrease in the other.

In our analysis of avg_positive_polarity and avg_negative_polarity (Heatmap 2), the correlation coefficient of -0.28 suggests a moderate negative correlation between the average positive polarity and the average negative polarity. This finding aligns with our initial hypothesis, indicating that articles with a greater average positive polarity are inclined to exhibit a lower average negative polarity, and vice versa. Thus, the observed correlation provides support for the notion that articles with predominantly positive sentiments tend to contain fewer instances of negative language, and conversely for articles with predominantly negative sentiments.

Based on these correlation coefficients, our hypotheses appear to be valid for both pairs of features. Articles with more content indeed tend to have more links, and articles with a higher average positive polarity tend to have a lower average negative polarity, and vice versa.

Correlation Results (Rohan K.)

For the correlation between weekday_is_monday and global_rate_positive_words, we suspect a weak negative correlation might exist. This suspicion arises because

Mondays are often associated with the start of the work or school week, which could potentially lead to lower levels of positivity in online content as people adjust to the demands of the week. However, this relationship might not be strong as other factors could influence the positivity of content irrespective of the day of the week.

The correlation found between weekday_is_monday and global_rate_positive_words was -0.01 (Heatmap 3). This result indicates an extremely weak negative correlation, aligning with the hypothesis of a minor association between Monday and decreased positivity in online content, although this correlation is nearly negligible.

As for the correlation between n_non_stop_words and data_channel_is_entertainment, We hypothesize a weak positive correlation. This assumption stems from the notion that entertainment-focused content might involve more words in general, given the descriptive nature of entertainment reporting or discussions. Therefore, articles belonging to the entertainment data channel might tend to have a higher count of non-stop words compared to articles in other channels.

For the correlation between n_non_stop_words and data_channel_is_entertainment, a correlation of 0.01 was found (Heatmap 4). Again, this represents an extremely weak positive correlation, supporting the hypothesis of a slight increase in the count of non-stop words in articles related to the entertainment data channel, although the correlation is practically insignificant.

In both cases, while the hypotheses hinted at a potential relationship, the correlations found were so close to zero that they hold little practical significance. Therefore, the initial suspicions were largely insignificant, and it seems other factors may play more significant roles in determining these variables.

Total Correlation (Megan S.)

Megan S. crafted a comprehensive correlation heatmap (Heatmap 5), showing the relationships between each data attribute and providing invaluable evidence of their interconnections within the dataset.