Instagram Reach Analysis Report

Prepared by Arya Gowda

1. Introduction

This report presents an analysis of Instagram reach using a dataset containing various metrics such as impressions from different sources, likes, comments, shares, and more. The analysis aims to understand the distribution of impressions, explore content performance, and examine the relationships between different metrics. Additionally, a prediction model is built to forecast future impressions based on the available data.

2. Data Overview

The dataset includes the following columns:

- Impressions
- From Home
- From Hashtags
- From Explore
- From Other
- Saves
- Comments
- Shares
- Likes
- Profile Visits
- Follows
- Caption
- Hashtags

3. Data Preprocessing

The data was cleaned by removing any null values to ensure accuracy in analysis.

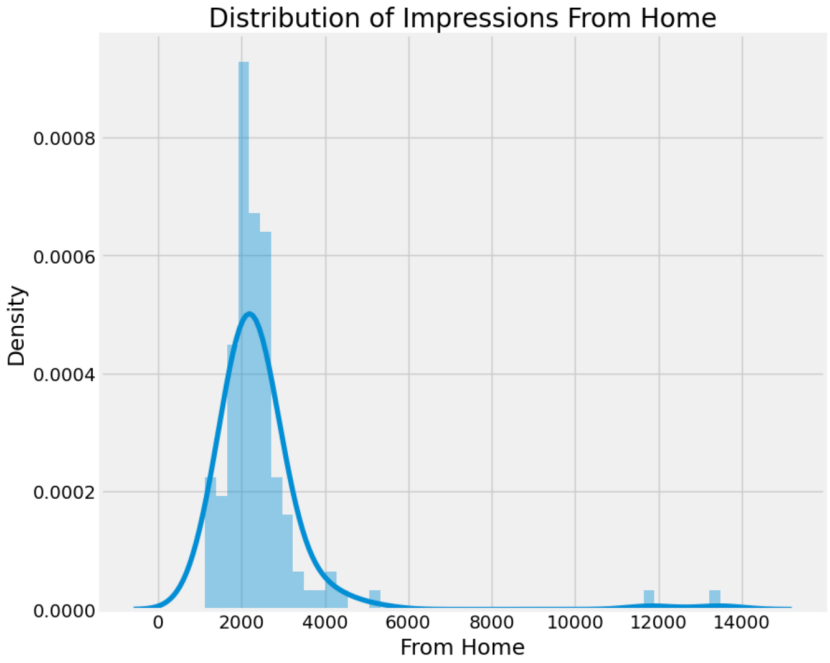
```
python
Copy code
data = pd.read_csv("Instagram.csv", encoding = 'latin1')
data.isnull().sum()
data = data.dropna()
```

4. Distribution of Impressions

We analyzed the distribution of impressions from different sources.

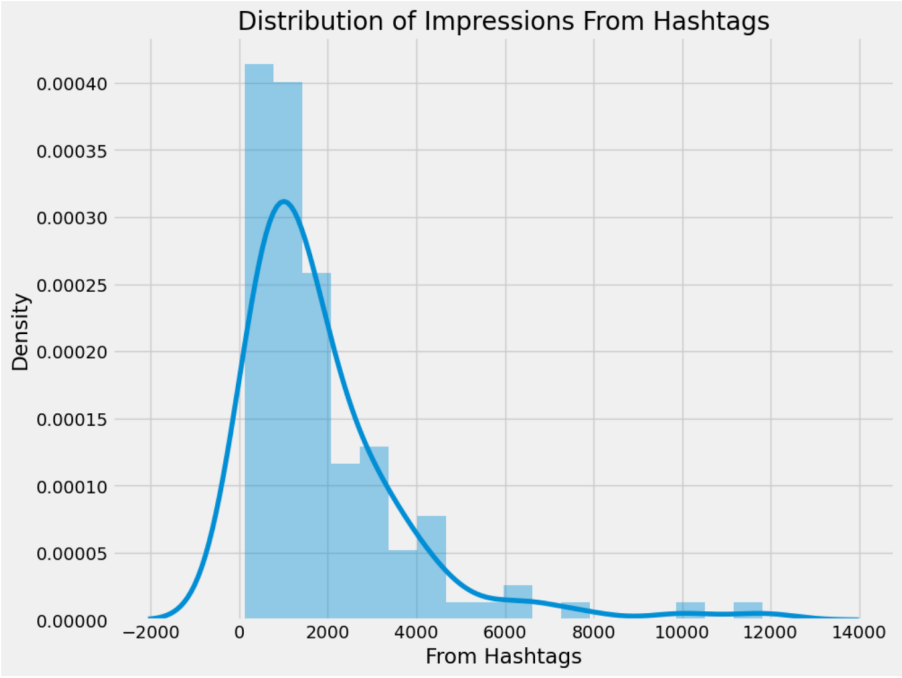
4.1. Impressions from Home

```
python
Copy code
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Home")
sns.distplot(data['From Home'])
plt.show()
```



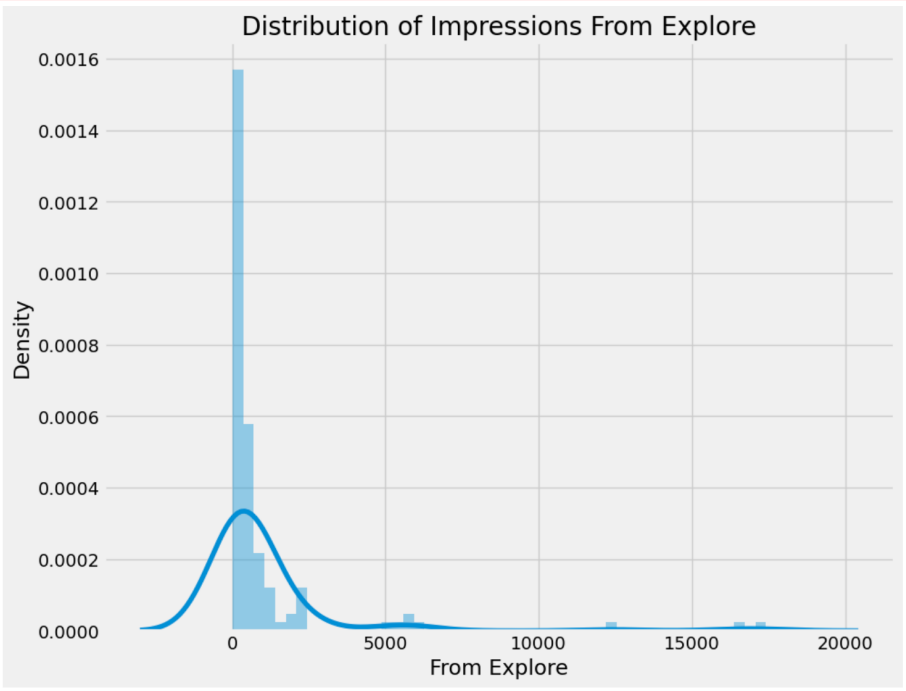
4.2. Impressions from Hashtags

```
python
Copy code
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Hashtags")
sns.distplot(data['From Hashtags'])
plt.show()
```

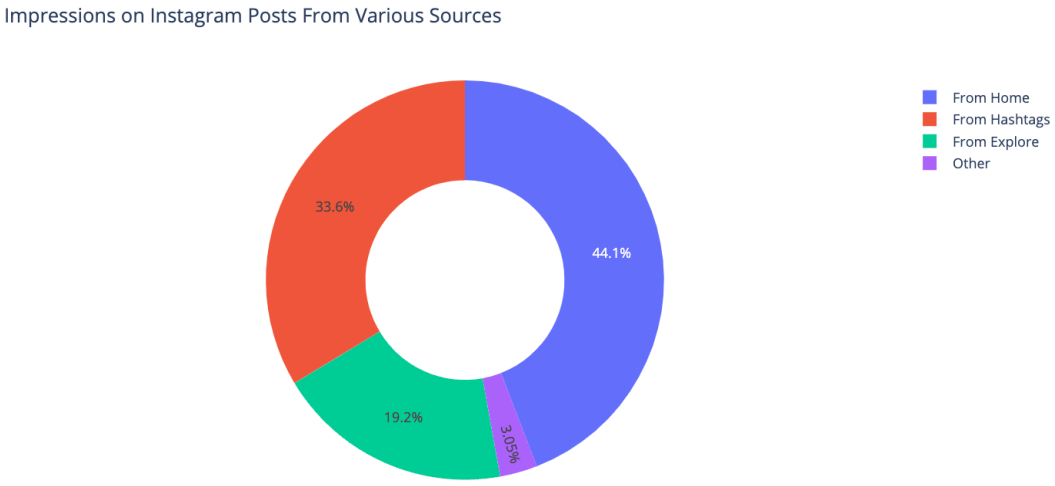


4.3. Impressions from Explore

```
python
Copy code
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Explore")
sns.distplot(data['From Explore'])
plt.show()
```



5. Total Impressions from Various Sources



A pie chart was created to visualize the total impressions from different sources.

```
python
Copy code
home = data["From Home"].sum()
hashtags = data["From Hashtags"].sum()
explore = data["From Explore"].sum()
other = data["From Other"].sum()

labels = ['From Home','From Hashtags','From Explore','Other']
values = [home, hashtags, explore, other]

fig = px.pie(data, values=values, names=labels,
             title='Impressions on Instagram Posts From Various Sources',
hole=0.5)
fig.show()
```

6. Content Analysis

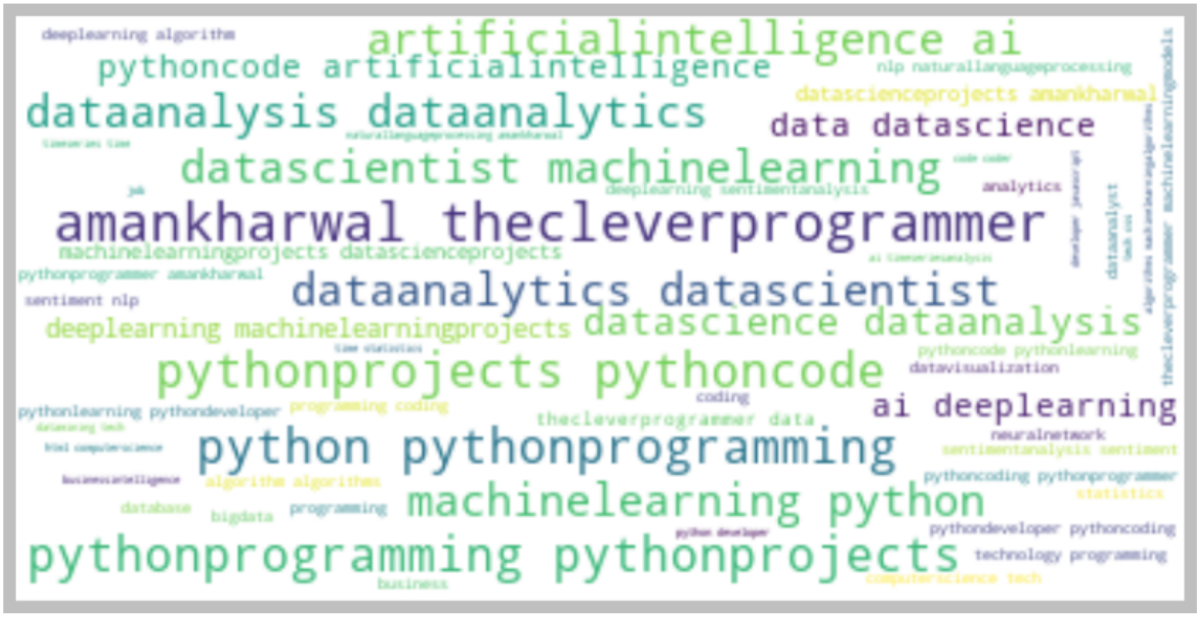
Word clouds were generated to visualize the most frequent words in captions and hashtags.

6.1. Word Cloud for Captions

```
python
Copy code
text = " ".join(i for i in data.Caption)
wordcloud = WordCloud(stopwords=STOPWORDS,
background_color="white").generate(text)
plt.figure(figsize=(12, 10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

6.2. Word Cloud for Hashtags

```
python
Copy code
text = " ".join(i for i in data.Hashtags)
wordcloud = WordCloud(stopwords=STOPWORDS,
background_color="white").generate(text)
plt.figure(figsize=(12, 10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



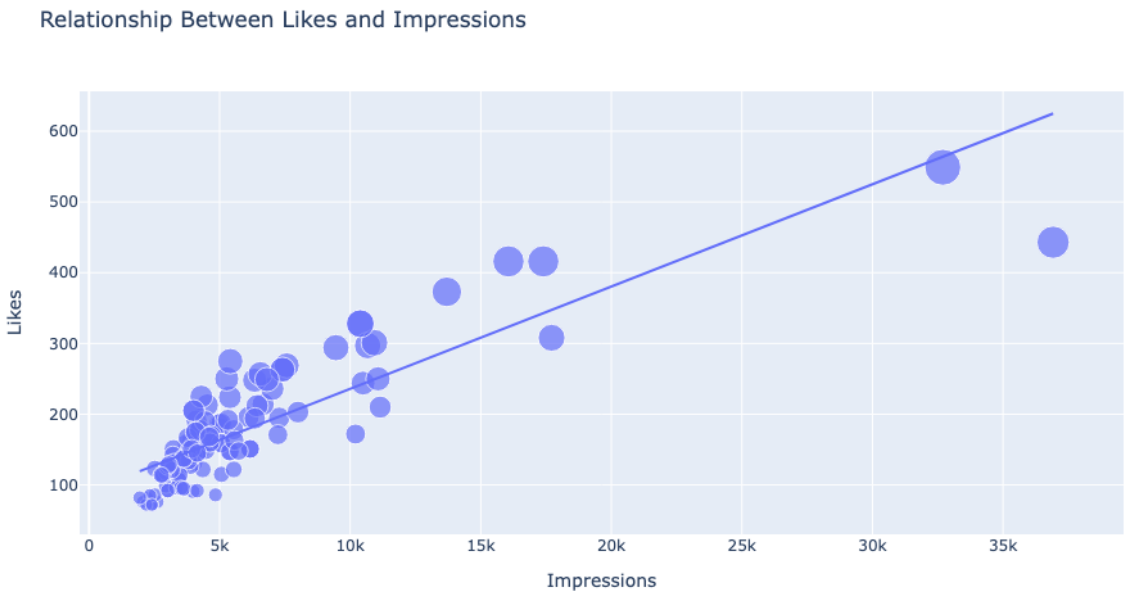
7. Analyzing Relationships

Scatter plots with trendlines were used to explore the relationships between impressions and other metrics.

7.1. Likes vs. Impressions

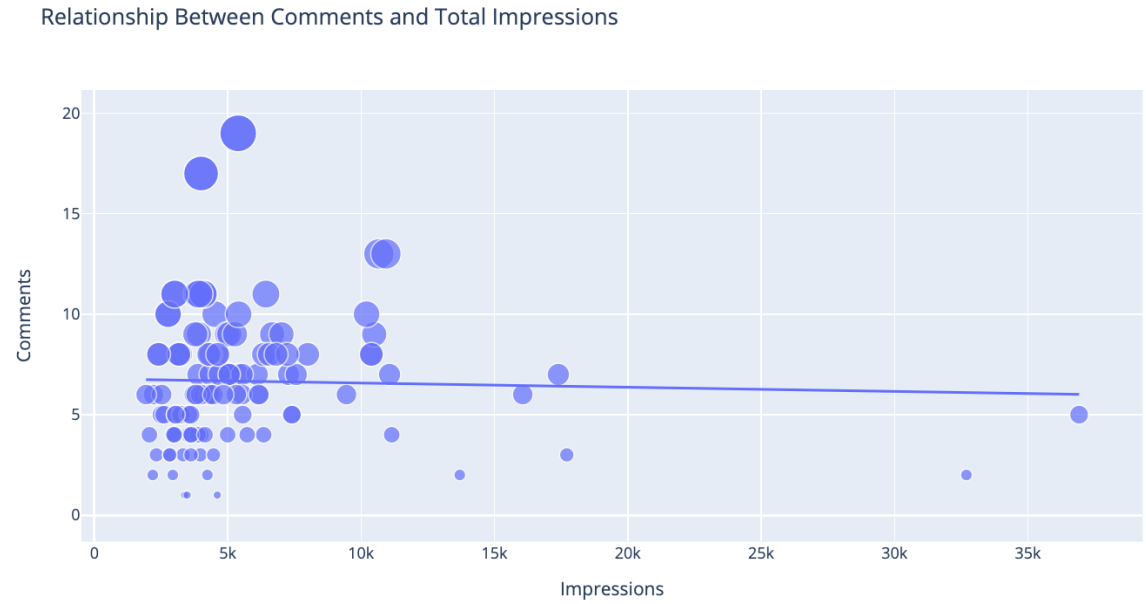
```
python
Copy code
figure = px.scatter(data_frame = data, x="Impressions", y="Likes",
size="Likes", trendline="ols", title="Relationship Between Likes and
Impressions")
```

```
figure.show()
```



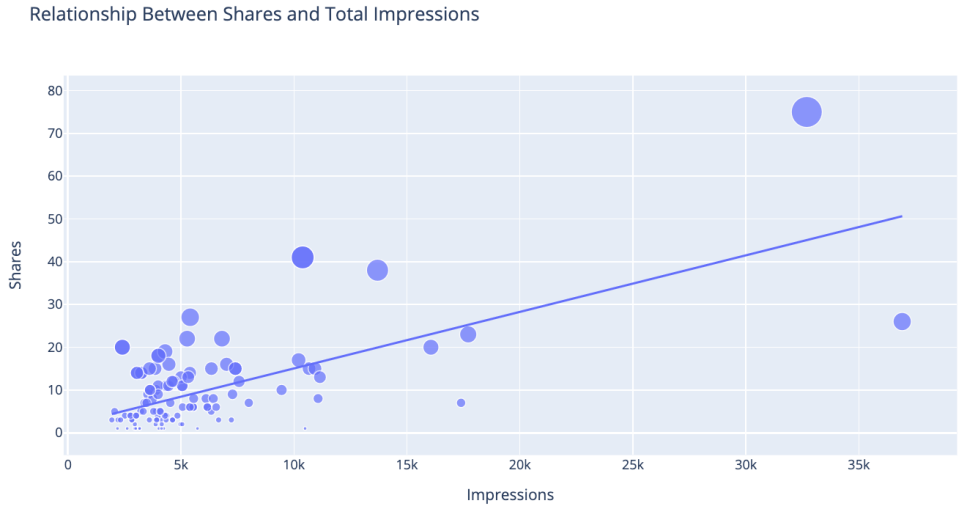
7.2. Comments vs. Impressions

```
python
Copy code
figure = px.scatter(data_frame = data, x="Impressions", y="Comments",
size="Comments", trendline="ols", title="Relationship Between Comments and
Total Impressions")
figure.show()
```



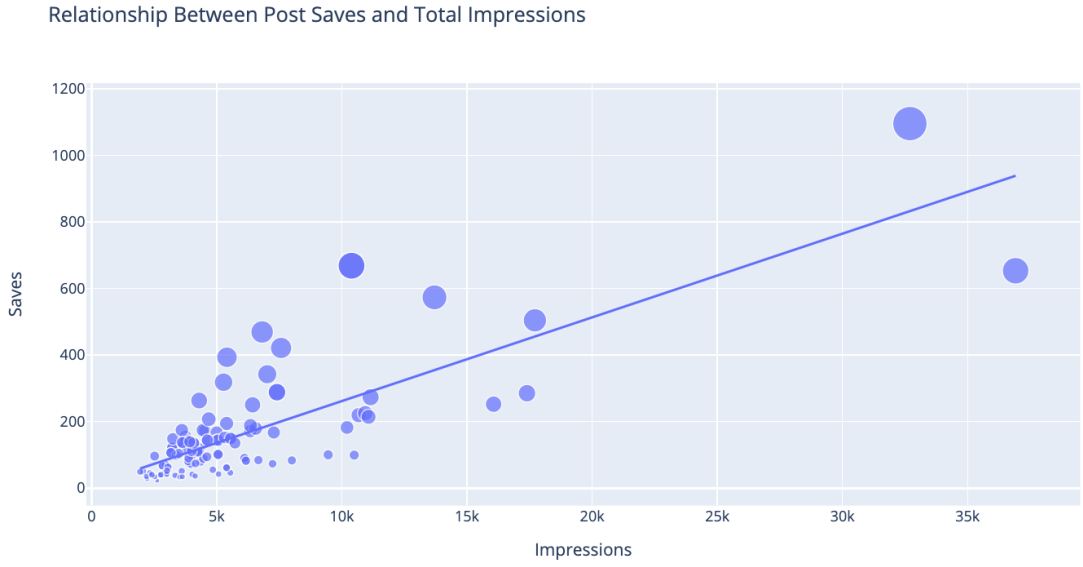
7.3. Shares vs. Impressions

```
python
Copy code
figure = px.scatter(data_frame = data, x="Impressions", y="Shares",
size="Shares", trendline="ols", title="Relationship Between Shares and Total
Impressions")
figure.show()
```



7.4. Saves vs. Impressions

```
python
Copy code
figure = px.scatter(data_frame = data, x="Impressions", y="Saves",
size="Saves", trendline="ols", title="Relationship Between Post Saves and
Total Impressions")
figure.show()
```



8. Correlation Analysis

The correlation between impressions and other metrics was calculated.

```
python
Copy code
correlation = data.corr()
print(correlation["Impressions"].sort_values(ascending=False))
```

Key Findings:

- Strongest correlation with `From Explore` (0.89)
- High correlation with `Follows` (0.88) and `Likes` (0.85)

9. Conversion Rate Analysis

The conversion rate was calculated to determine the effectiveness of profile visits in generating new followers.

```
python
Copy code
conversion_rate = (data["Follows"].sum() / data["Profile Visits"].sum()) *
100
print(conversion_rate)
```

Conversion Rate: 41.00%

10. Prediction Model

A prediction model was built using Passive Aggressive Regressor to forecast future impressions.

```
python
Copy code
x = np.array(data[['Likes', 'Saves', 'Comments', 'Shares', 'Profile Visits',
'Follows']])
y = np.array(data["Impressions"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2,
random_state=42)

model = PassiveAggressiveRegressor()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))

features = np.array([[282.0, 233.0, 4.0, 9.0, 165.0, 54.0]])
print(model.predict(features))
```

Model Score: 0.928

Predicted Impressions: 11650

11. Conclusion

The analysis provides valuable insights into the factors influencing Instagram reach. The prediction model demonstrates high accuracy, making it a useful tool for forecasting future performance based on current metrics.