Instagram Reach Analysis

objective-

- to analyse the trend and pattern
- to find which engagement feature leads to highest impressions
- to create a predictive model

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

Data exploration

In [ ]:

```
data=pd.read_csv('datasets/Instagram_data.csv',encoding='latin1')
```

In [ ]:

```
data.head(5)
```

In [ ]:

```
data.info()
```

there are no null values and the data types do not require changing

In [ ]:

```
data.describe()
```

number of unique hashtags

In [ ]:

```
data['Hashtags'].nunique()
```

number of unique captions

In [ ]:

```
data['Caption'].nunique()
```

Correlation Analysis

In [ ]:

```
corr_mat=data.iloc[:,0:11].corr()
corr_mat
```

In [ ]:

```
sns.heatmap(corr_mat,annot=True)
```

Feautre engineering

calculating engagement rate

shows percentage of people who have been shown the post and have interacted with it.

In [ ]:

```
data['Engagement Rate']=(data['Likes']+data['Comments']+data['Shares'])/data['
```

In [ ]:

```
data.head(4)
```

calculating conversion rate

conversion rate means how many followers are you getting from the number of profile visits from a post.

In [ ]:

```
conversion_rate=(data['Follows'].sum()/data['Profile Visits'].sum())*100
print(conversion_rate)
```

Analyzing trends and patterns

reach over time

In [ ]:

```
sns.set(palette='deep',rc={"figure.figsize": (6, 4)},style='whitegrid')
sns.lineplot(x=data.index,y=data['Impressions'])
plt.title('reach over time')
plt.xlabel('post no.')
plt.ylabel('impressions')
```

sources of impressions

In [ ]:

```
values=[data['From Home'].sum(),data['From Hashtags'].sum(),data['From Explore
labels=['From Home','From Hashtags','From Explore','Other']
fig, ax = plt.subplots()
ax.pie(values, labels=labels, autopct='%2.1f%%')
plt.title('impressions on instagram posts from various sources')
```

most used hashtags

In [ ]:

```
!pip install wordcloud  -q
from wordcloud import WordCloud
```

In [ ]:

```
hashtags=' '.join(data['Hashtags'].tolist())
wordcloud=WordCloud(width=900,height=500,background_color='white').generate(ha

plt.figure(figsize=(10,8))
plt.imshow(wordcloud,interpolation='bilinear')
plt.title('most used hashtags')
plt.axis('off')
```

realtionship between impressions and number of comments

In [ ]:

```
sns.regplot(x='Comments',y='Impressions',data=data)
```

comments do not affect the reach

realtionship between impressions and likes

In [ ]:

```
sns.regplot(x='Likes',y='Impressions',data=data)
```

positive relationship

realtionship between impressions and saves

In [ ]:

```
sns.regplot(x='Saves',y='Impressions',data=data)
```

positive relationship

relationship between impressions and shares

In [ ]:

```
sns.regplot(x='Shares',y='Impressions',data=data)
```

positive relationship

Predictive Modeling

In [ ]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [ ]:

```
x=data.drop(['Impressions','Hashtags','Caption'],axis=1)
y=data['Impressions']
```

splitting the data

In [ ]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
```

trainig machine learning model

In [ ]:

```
model=LinearRegression()
model.fit(x_train,y_train)
```

make predictions

In [ ]:

```
y_pred=model.predict(x_test)
```

evaluate the model

In [ ]:

```
r2=r2_score(y_test,y_pred)
print('r-squared score ',r2)
```

- 99.99% of the variance in the actual values is explained by the model's predictions.
- model's predictions are very close to the actual values.

finding most important feature

which form of engagement results in higher impressions

In [ ]:

```
x1=data.iloc[:,5:11]
y1=data['Impressions']
x1_train,x1_test,y1_train,y1_test=train_test_split(x1,y1,test_size=0.2,random_
model1=LinearRegression()
model1.fit(x1_train,y1_train)
coefficients = model1.coef_
feature_importance = pd.Series(coefficients, index=x1.columns).sort_values(asc
print(feature_importance)
```

the impressions increases most when number of folowers increase

when 1 person follows the instagram page the impressions incease by approximately 32

In [ ]: