Problem Statements

Q1. Imagine you have a dataset where you have different Instagram features like username , Caption , Hashtag , Followers , Time_Since_posted , and likes , now your task is to predict the number of likes and Time Since posted and the rest of the features are your input features. Now you have to build a model which can predict the number of likes and Time Since posted.

DATASET: https://www.kaggle.com/datasets/rxsraghavagrawal/instagram-reach

In [29]:
```
## Import the necessary libraries:-

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import OneHotEncoder
import re
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

In [30]:
```
## Load the dataset using pandas:
data = pd.read_csv(r"C:\Users\hrush\Downloads\archive (2)\instagram_reach.csv")
```

In [31]:
```
## Checking top 5 rows
data.head()
```

Out[31]:

	Unnamed: 0	S.No	USERNAME	Caption	Followers	Hashtags	Time since posted	Likes
0	0	1	mikequindazzi	Who are #DataScientist and what do they do? >>...	1600	#MachineLearning #AI #DataAnalytics #DataScien...	11 hours	139
1	1	2	drgorillapaints	We all know where it's going. We just have to ...	880	#deck .#mac #macintosh#sayhello #apple #steve...	2 hours	23
2	2	3	aitrading_official	Alexander Barinov: 4 years as CFO in multinati...	255	#whoiswho #aitrading #ai #aitradingteam#instat...	2 hours	25
3	3	4	opensourcedworkplace	sfad	340	#iot #cre#workplace #CDO #bigdata #technology#...	3 hours	49
4	4	5	crea.vision	Ever missed a call while your phone was chargi...	304	#instamachinelearning #instabigdata#instamarke...	3 hours	30

In [32]:
```
## Checking Rows & Columns Availabale in Dataset
data.shape
```

Out[32]:
```
(100, 8)
```

In [33]:
```
## Checking Details Information related with Dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Unnamed: 0         100 non-null    int64 
 1   S.No               100 non-null    int64 
 2   USERNAME           100 non-null    object
 3   Caption            94 non-null     object
 4   Followers          100 non-null    int64 
 5   Hashtags           100 non-null    object
 6   Time since posted  100 non-null    object
 7   Likes              100 non-null    int64 
dtypes: int64(4), object(4)
memory usage: 6.4+ KB
```

In [34]:
```
## Checking All Columns name present in dataset
data.columns
```

Out[34]:
```
Index(['Unnamed: 0', 'S.No', 'USERNAME', 'Caption', 'Followers', 'Hashtags',
       'Time since posted', 'Likes'],
      dtype='object')
```

In [35]:
```
## checking top 2 rows of dataset
data.head(2)
```

Out[35]:

	Unnamed: 0	S.No	USERNAME	Caption	Followers	Hashtags	Time since posted	Likes
0	0	1	mikequindazzi	Who are #DataScientist and what do they do? >>...	1600	#MachineLearning #AI #DataAnalytics #DataScien...	11 hours	139
1	1	2	drgorillapaints	We all know where it's going. We just have to ...	880	#deck .#mac #macintosh#sayhello #apple #steve...	2 hours	23

In [36]:
```
# Remove unnecessary columns
data= data.drop(['Unnamed: 0', 'S.No'], axis=1)
```

In [37]:
```
## Checking All Columns name present in Dataset
data.columns
```

Out[37]:
```
Index(['USERNAME', 'Caption', 'Followers', 'Hashtags', 'Time since posted',
       'Likes'],
      dtype='object')
```

In [38]:
```
## Checking top 3 rows of dataset after droping unnecessary columns.
data.head(3)
```

Out[38]:

	USERNAME	Caption	Followers	Hashtags	Time since posted	Likes
0	mikequindazzi	Who are #DataScientist and what do they do? >>...	1600	#MachineLearning #AI #DataAnalytics #DataScien...	11 hours	139
1	drgorillapaints	We all know where it's going. We just have to ...	880	#deck .#mac #macintosh#sayhello #apple #steve...	2 hours	23
2	aitrading_official	Alexander Barinov: 4 years as CFO in multinati...	255	#whoiswho #aitrading #ai #aitradingteam#instat...	2 hours	25

Doing EDA and Analyzing Instagram Reach

In [39]:
```
## Distribution of Impressions From Followers
plt.figure(figsize=(10, 8))
plt.style.use('fivethirtyeight')
plt.title("Distribution of Impressions From Followers")
sns.distplot(data['Followers'])
plt.show()
```

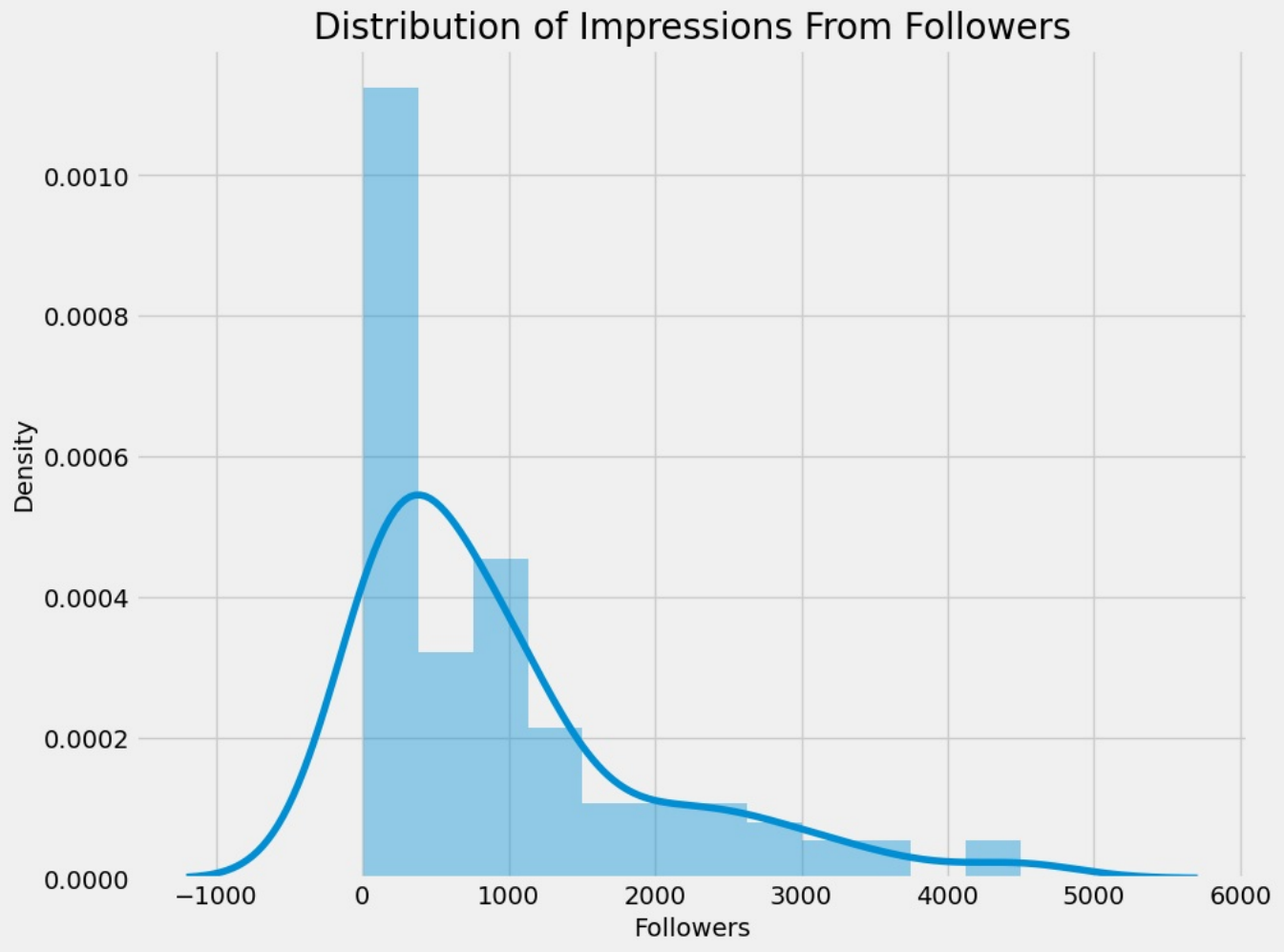
```
C:\Users\hrush\AppData\Local\Temp\ipykernel_26088\1493182440.py:5: UserWarning:



`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```



In [40]:
```
## Distribution of Impressions From Likes
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Likes")
sns.distplot(data['Likes'])
plt.show()
```

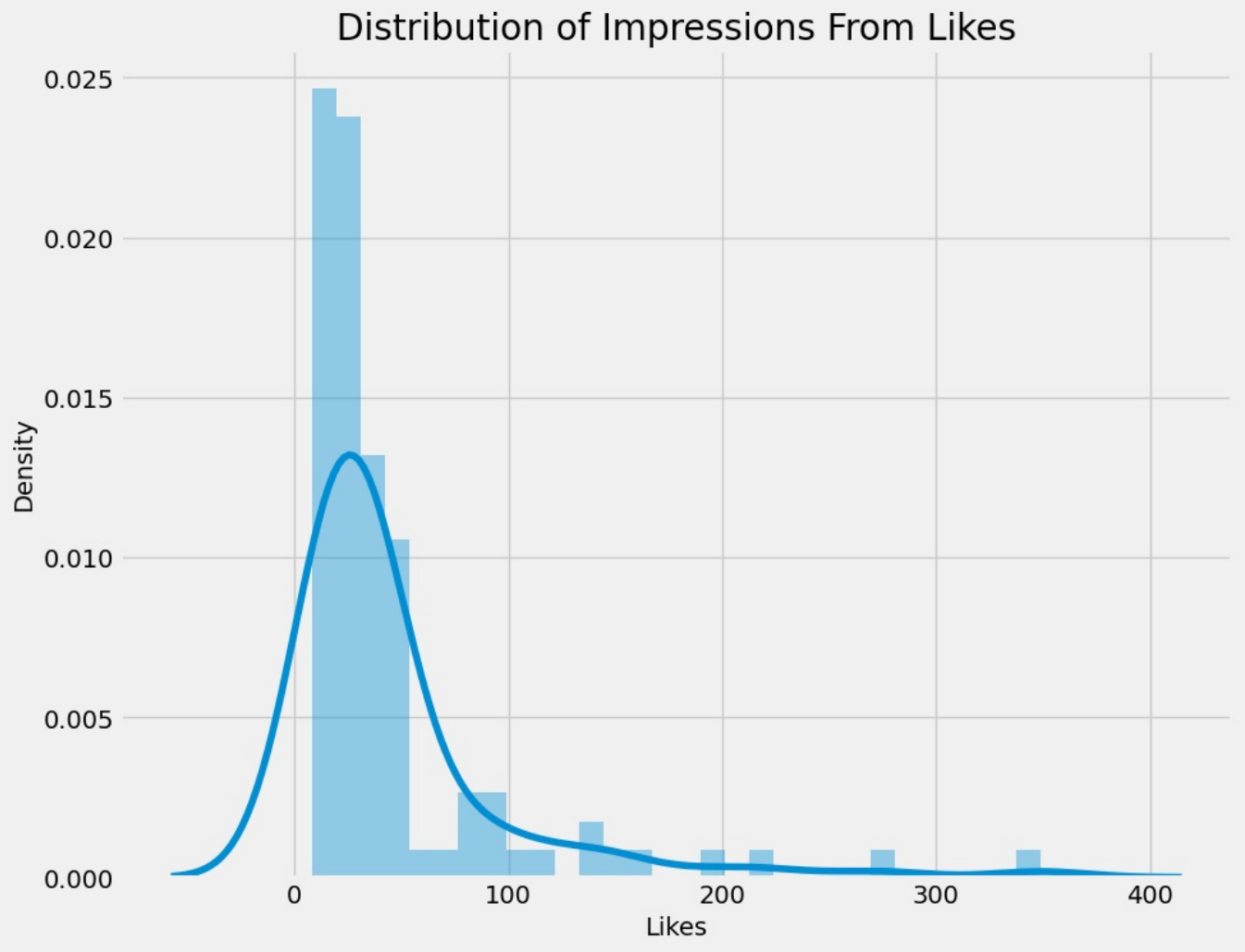
```
C:\Users\hrush\AppData\Local\Temp\ipykernel_26088\32119417.py:4: UserWarning:



`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```



In [41]:
```
## Relation between Likes and Followers

followers = data["Followers"].sum()
likes = data["Likes"].sum()

labels = ['Followers', 'Likes']
values = [followers, likes]

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

In [42]:
```
## Plotting Scatter-plot for showing Relationship Between Likes and Followers

figure = px.scatter(data_frame = data, x="Likes",
                    y="Followers", trendline="ols",
                    title = "Relationship Between Likes and Followers")
figure.show()
```

In [43]:
```
# Select the relevant features and target variables

features = ['USERNAME', 'Caption', 'Hashtags', 'Followers']
target_likes = 'Likes'
target_time_since_posted = 'Time since posted'
```

In [44]:
```
# Split the data into training and testing sets

X = data[features]
y_likes = data[target_likes]
y_time_since_posted = data[target_time_since_posted]
X_train, X_test, y_likes_train, y_likes_test, y_time_since_posted_train, y_time_since_posted_test = train_test_s
```

In [45]:
```
# Preprocess the text features using one-hot encoding
encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
X_train_encoded = encoder.fit_transform(X_train)
X_test_encoded = encoder.transform(X_test)
```

```
C:\Users\hrush\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\preprocessing\_encoders.py:828
: FutureWarning:

`sparse` was renamed to `sparse_output` in version 1.2 and will be removed in 1.4. `sparse_output` is ignored u
nless you leave `sparse` to its default value.
```

Train a model to predict the number of likes:

In [46]:
```
# Train a model to predict the number of likes
likes_model = LinearRegression()
likes_model.fit(X_train_encoded, y_likes_train)
likes_predictions = likes_model.predict(X_test_encoded)
likes_mse = mean_squared_error(y_likes_test, likes_predictions)
print("Mean Squared Error (Likes):", likes_mse)
```

```
Mean Squared Error (Likes): 1673.3748726633075
```

Train a model to predict the time since posted

In [47]:
```
# Preprocess the time since posted variable
def extract_numerical_value(time_string):
    numerical_value = re.findall(r'\d+', time_string)[0]
    return int(numerical_value)
```

In [48]:
```
y_time_since_posted_train = y_time_since_posted_train.apply(extract_numerical_value)
y_time_since_posted_test = y_time_since_posted_test.apply(extract_numerical_value)
```

In [49]:
```
# Train a model to predict the time since posted
time_since_posted_model = LinearRegression()
time_since_posted_model.fit(X_train_encoded, y_time_since_posted_train)
time_since_posted_predictions = time_since_posted_model.predict(X_test_encoded)
time_since_posted_mse = mean_squared_error(y_time_since_posted_test, time_since_posted_predictions)
print("Mean Squared Error (Time Since Posted):", time_since_posted_mse)
```

```
Mean Squared Error (Time Since Posted): 12.714517385002086
```