

# DATA VISUALIZATION - FINAL PROJECT

Submitted by

Jithin Raj MK – 1103468

Darshan Siva Kumar – 1210870

Mallikarjun Esnapur – 1209844

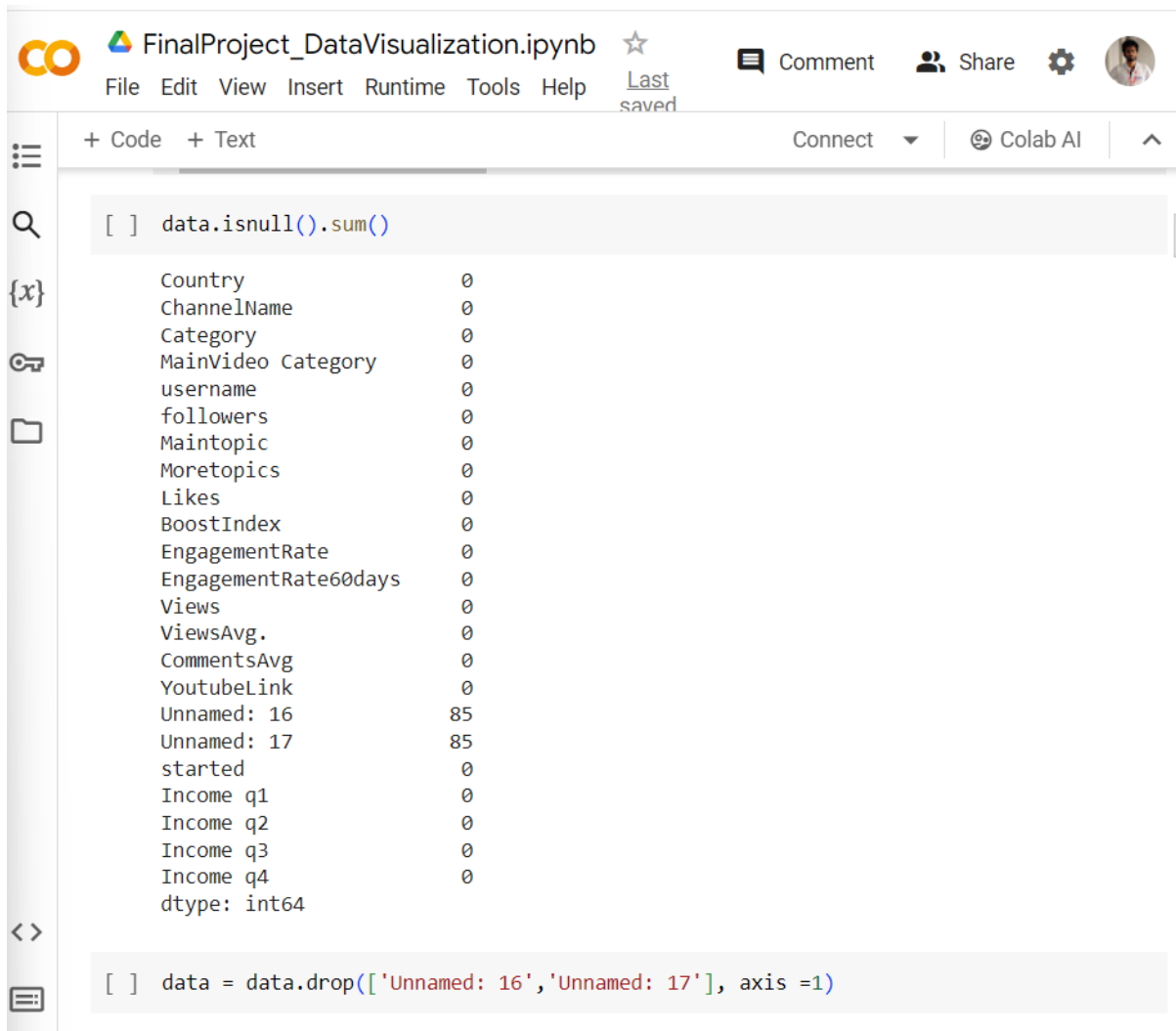
Under the guidance: Prof. Magdalene R

## **Project Requirements:**

To Perform Data Analysis on Top 100 Youtubers data which includes their subscribers, likes per videos, category of Channel, their revenues and views.

To begin with, for any dataset chosen/given we need to perform Data Cleaning, or data preprocessing before we start our analysis.

For Data Preprocessing we have considered checking for null values, and we observed two whole columns have empty records, which are unnamed. So, we chose to drop these columns as it has no significance to the data.



```
[ ] data.isnull().sum()

Country          0
ChannelName      0
Category         0
MainVideo Category 0
username         0
followers        0
Maintopic        0
Moretopics       0
Likes            0
BoostIndex       0
EngagementRate   0
EngagementRate60days 0
Views            0
ViewsAvg         0
CommentsAvg      0
YoutubeLink      0
Unnamed: 16      85
Unnamed: 17      85
started          0
Income q1        0
Income q2        0
Income q3        0
Income q4        0
dtype: int64

[ ] data = data.drop(['Unnamed: 16', 'Unnamed: 17'], axis =1)
```

We wanted to see if there are any outliers in the dataset for the columns Likes, Subscribers as we need to build a relationship for Predictive Analysis. We used IQR – Inter Quartile range approach to address the outliers.

```
# Calculate IQR
Q1 = data['followers'].quantile(0.25)
Q3 = data['followers'].quantile(0.75)
IQR = Q3 - Q1

# Set a threshold for outlier detection (e.g., values outside 1.5*IQR)
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
# outliers = data[(data['Likes'] < lower_bound) | (data['Likes'] > upper_bound)]
# data = data[(data['Likes'] >= lower_bound) & (data['Likes'] <= upper_bound)]
data['followers'] = data['followers'].clip(lower=lower_bound, upper=upper_bound)
```

For Predictive Analysis Model, we need the columns to be encoded, for which we used Label Encoder (). We retained the existing columns and concatenated the new columns.

We have retained these columns to establish and build charts for the relationships asked.

```
from sklearn.preprocessing import LabelEncoder

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Create new columns for encoded values
for column in ['Country', 'Category', 'MainVideo Category', 'Maintopic', 'Moretopic']:
    encoded_column = column + '_encoded'
    data[encoded_column] = label_encoder.fit_transform(data[column])

# Display the updated DataFrame
print(data.head())
```

Feature Scaling is another preprocessing step we performed as to maintain the Normalization of features, maintain equal weights over different features, to have consistency in the data.

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

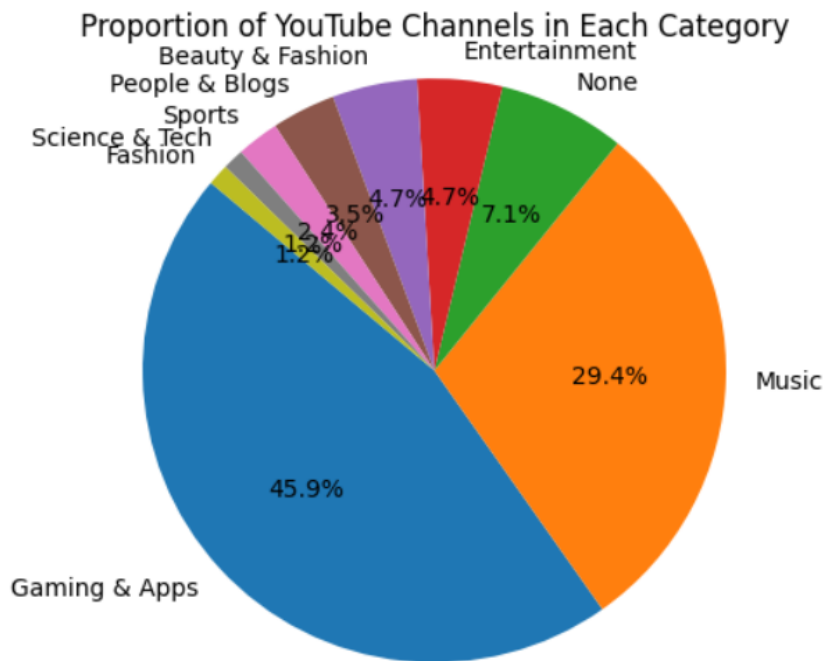
### Chart 1:

**Analyze the top 100 YouTube channels to identify their categories.**

**Visualize the proportion of channels in each category using a pie chart**

From the pie chart, we could infer that the Gaming and Apps industry is huge on YouTube from the given dataset, Next comes Music Industry. In simple words, we have more YouTube channels in Gaming and Apps industry and Music, next.

```
streamlit run /usr/local/lib/python3.10/dist-packages/colab_kernel_launcher.py [
```

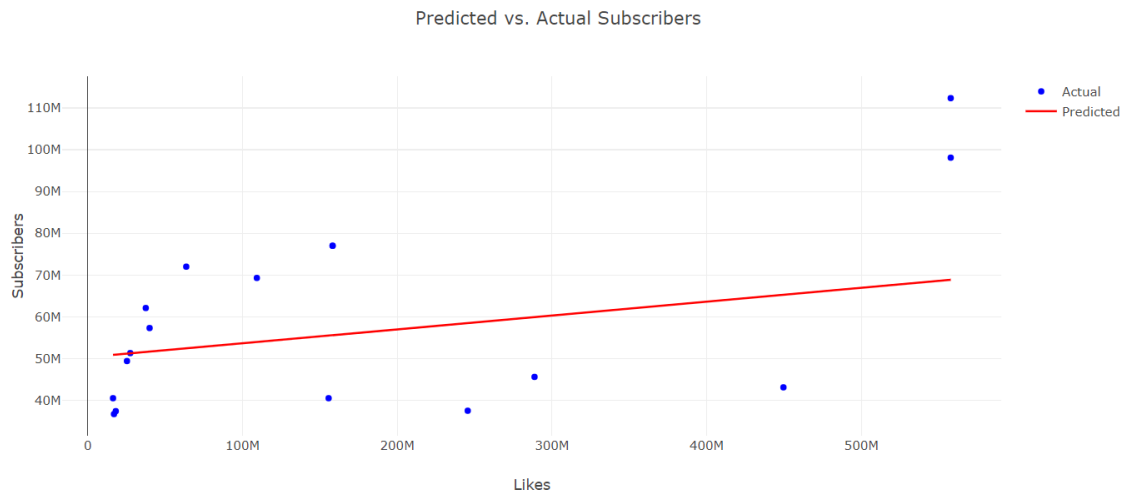


## Chart 2:

### Relationship between the 'likes' and 'subscribers'

- **Predictive Analysis of Likes and Subscribers Relationship – develop a predictive model to understand the relationship between 'likes' and 'subscribers'**
- **Predict the number of subscribers based on the number of 'likes' a channel receives**
- **Create a visualization to show its accuracy and the relationship identified**

We have implemented Linear Regression for Predicting the number of subscribers based on the number of likes for a channel receives. As we have 1 independent and 1 dependent variable, we have used Linear Regression to have the predictions.



As we could see the data is Non-Linear as the correlational value is very low.

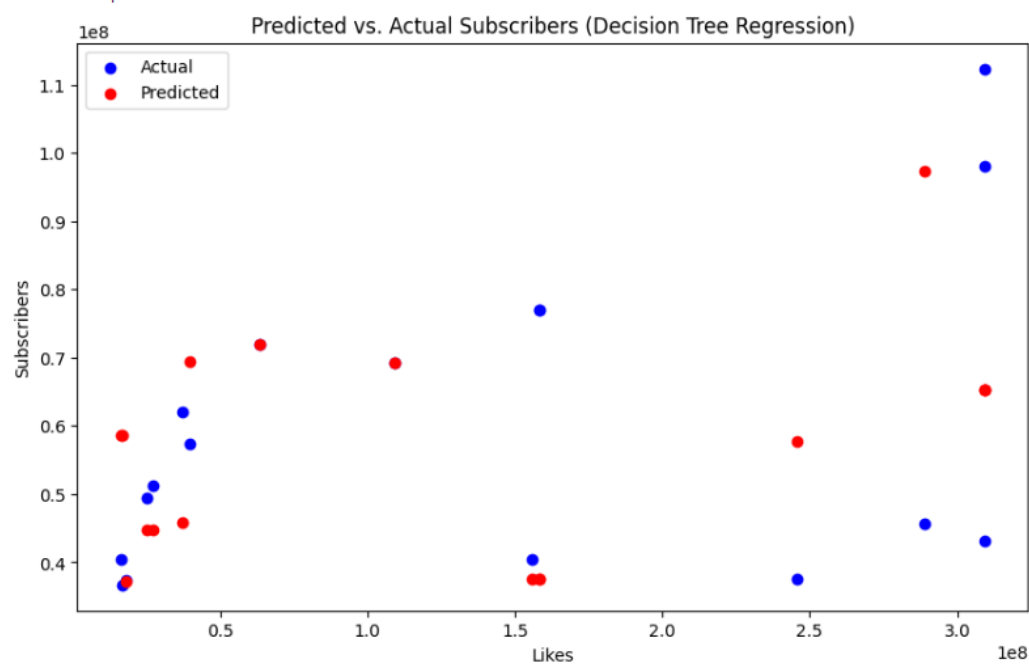
```
y = data[ 'followers' ].values
print(data[ 'Likes' ].corr(data[ 'followers' ]))
```

0.3739980130045567

It is better to use a Non Linear ML Model for this problem.

We have implemented Decision Tree which has better Predicted and Actual Values.

Mean Absolute Error: 19763333.333333332  
Mean Squared Error: 663423723529411.8  
Root Mean Squared Error: 25757013.09409559



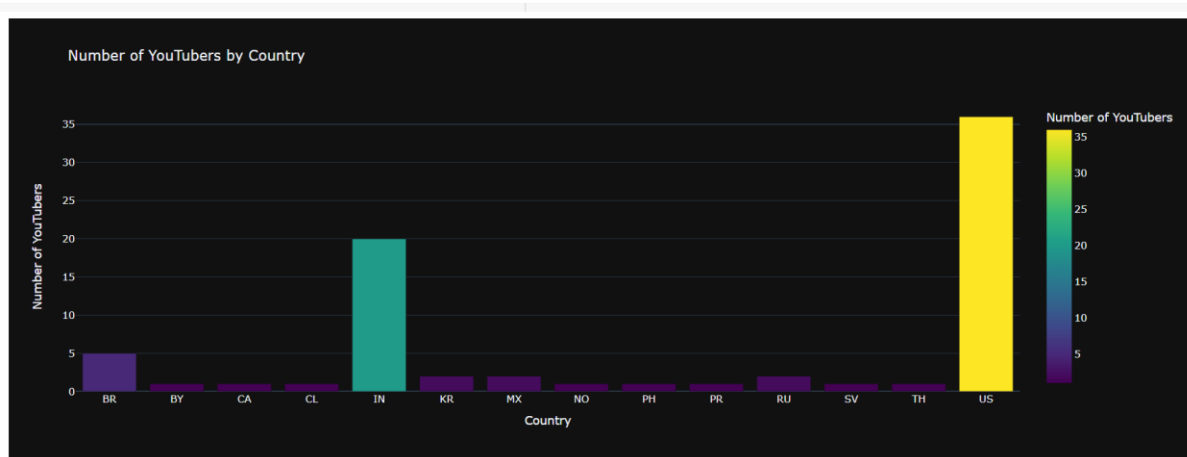
### Chart 3:

#### Global distribution of YouTubers

- Aggregate the number of YouTubers by country.
- Visualize each country's number of YouTubers using a map or bar chart.

We grouped the data by Country and Channel names and created a bar graph.

From the plot, we infer that USA has most of the Youtubers and India has next most Youtubers.



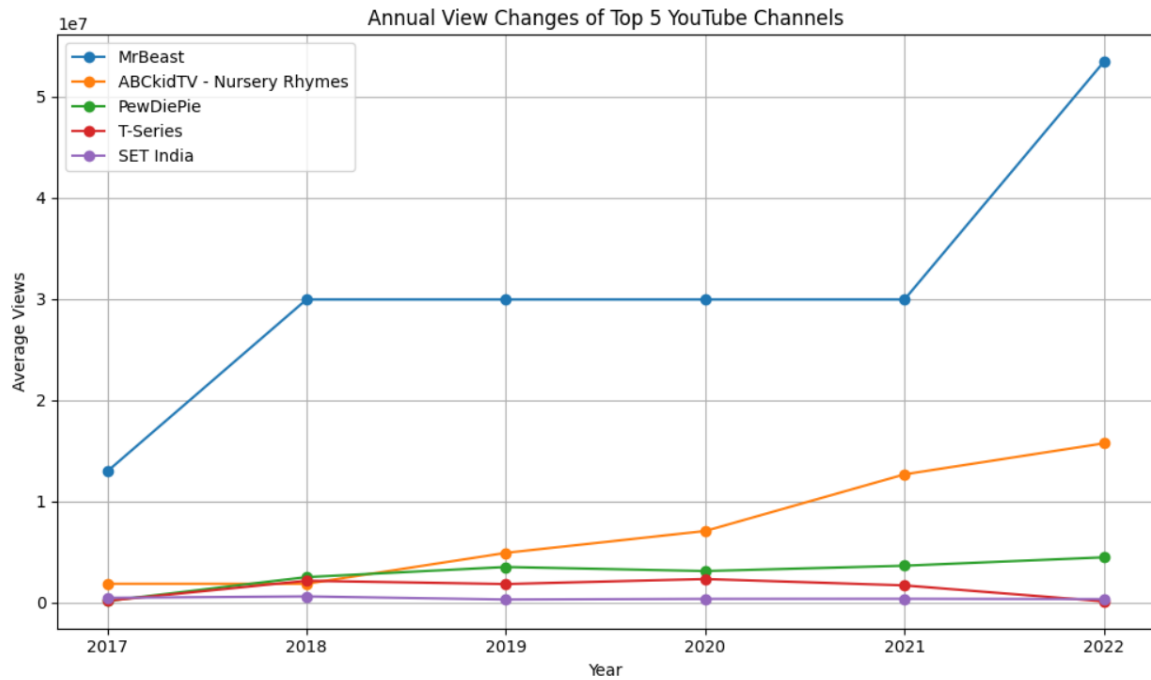
### Chart 4:

#### Annual View changes of top 5 YouTube channels

- Track the average view changes of the top 5 YouTube channels over the years.
- Use a line chart to display the trends in view changes annually.

From the top 5 Youtube channels, we could see that the trend increased in the recent years.

Out of the top 5, we could see Mr. Beast channels has grown rapidly in year 2022 which has been consistent till 2021. We could also see ABCKid TV has a consistent growth from 2019, it could be due to various reasons like Covid, much adaptation of Digital Devices in households.

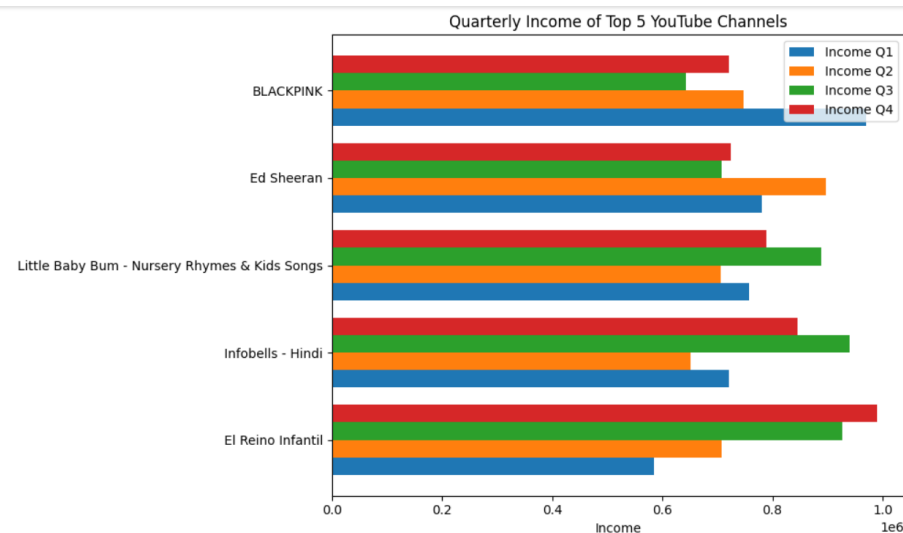


**Chart 5:**

#### Quarterly income of top 5 YouTube channels

- Calculate the average quarterly income of the top 5 YouTube channels.
- Visualize this data using a group bar horizontal chart.

For the top 5 YouTube channels, we calculated the Average Income from different quarters and plotted the chart. As we performed the Feature scaling, we could say that the average quarterly income by top 5 Youtubers is at least 0.6 every quarter.



**Chart 6:**

#### YouTube channels by category

- Implement an ML model to cluster the YouTube channels based on their characteristics such as category, likes, views, subscriber count
- Create a visualization to present the clusters using different colours/markers

We have implemented K means clustering, for understanding how the dataset can be segregated on basis of like and views and the categories.

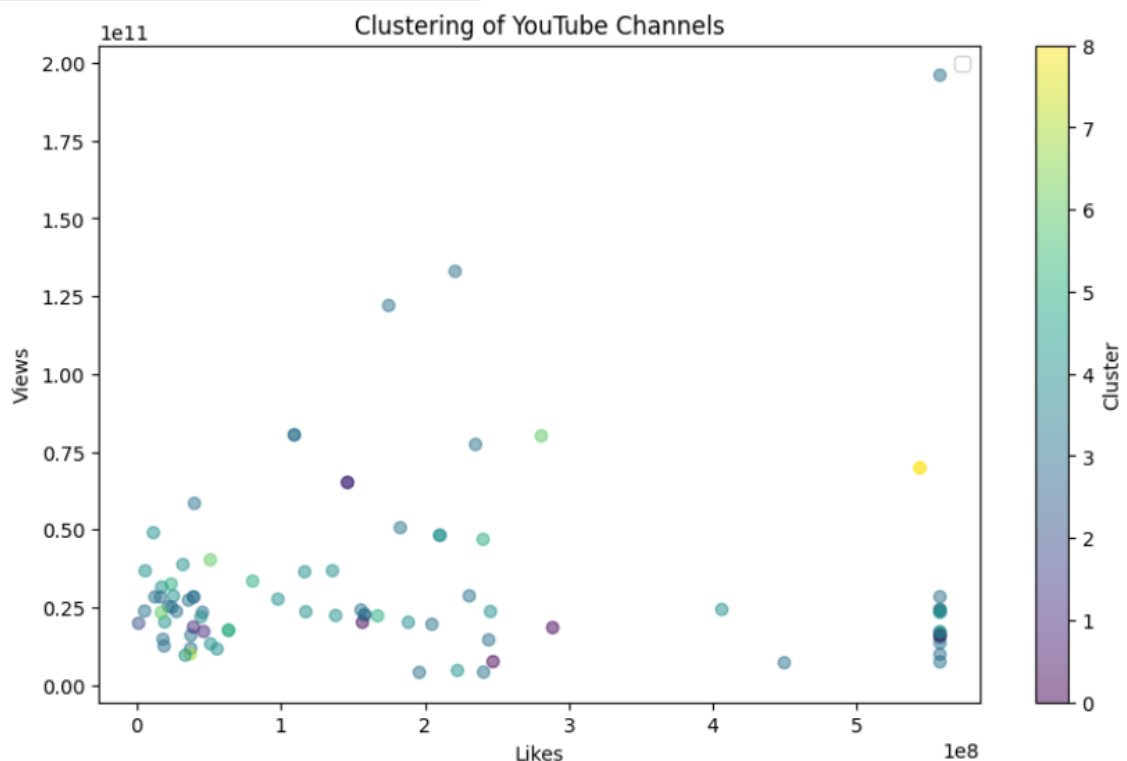
We used Silhouette Score to evaluate the clustering, we could observe that

Silhouette Score: 0.7524827286277835.

Silhouette Score, gives the confidence with which a datapoint is associating with its cluster.

The score generally has a variation from Negative 1 to Postive 1. (-1 to +1)

Silhouette Score: 0.7524827286277835



## Chart 7:

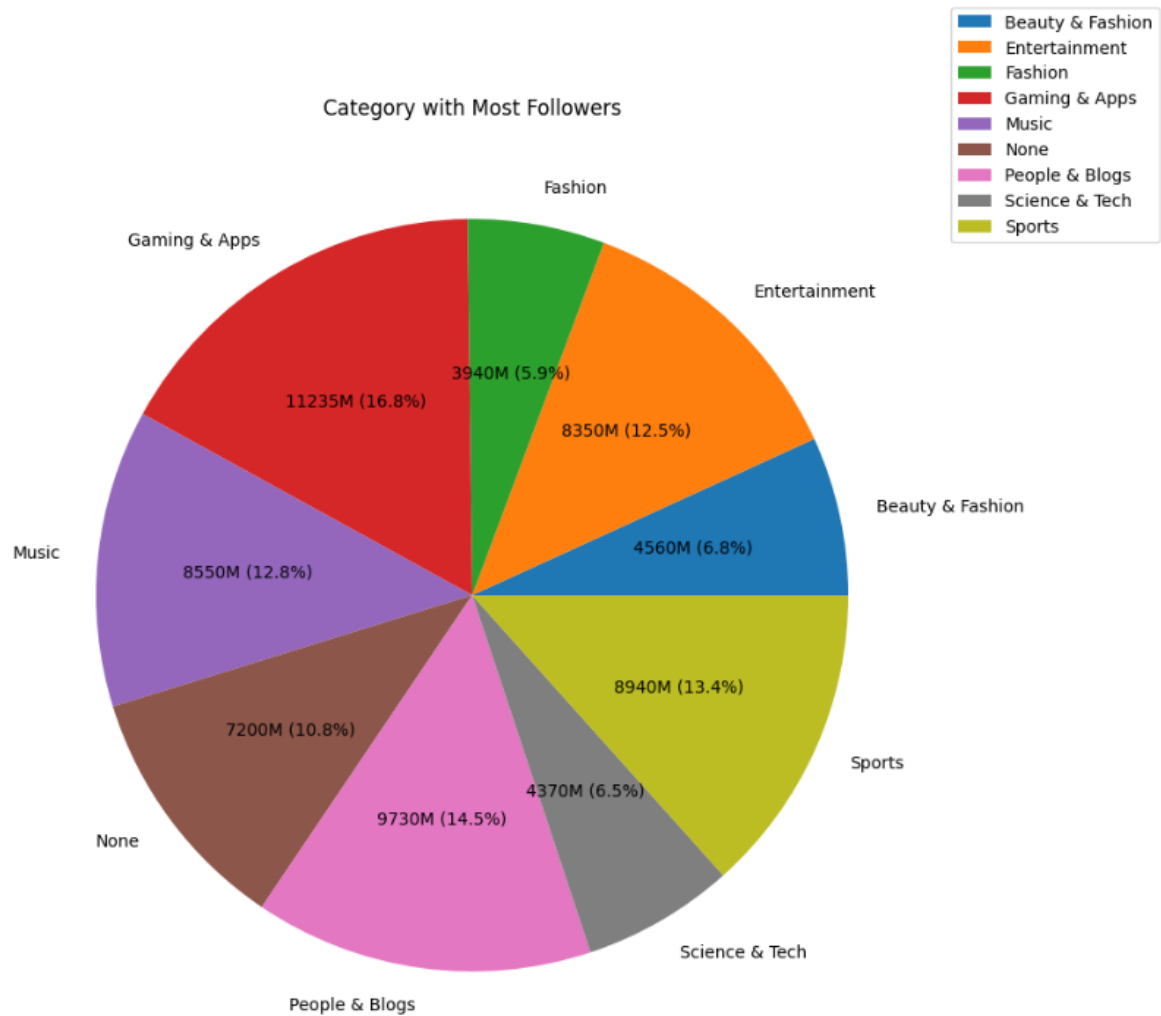
**Category with most followers • Identify the category with the most followers under each category • Create a visualization to highlight the category with the most followers.**

We have grouped by Category and Followers and found out the max followers and converted that to a list and created a pie chart.

From the Pie Chart, we can infer that Gaming and Apps Category has most followers, followed by People and Blogs.

If we recall about the YouTube Channels in each category, we have Gaming and apps, and Music has most Youtubers, but the followers count is more for Gaming and apps, and People and Blogs, and followed by Music and 4<sup>th</sup> is Entertainment.





## Chart 8:

### Channel with Most Subscribers

- Identify the channel with the most subscribers.
- Display the name of the channel and its subscriber count in a prominent, visually engaging manner.

We can infer that T series has the highest number of subscribers with count of 11235000, closely followed by ABCKidTV and SET India.

