Deliverable 2

Team Members:

- Pavan Kalyan Reddy Madatala
- Veda Samhitha Dyawanapally
- Meghana Chikyala
- Poornima Pulakandam
- Amarnath Reddy Chinthapalli

Dataset:

The dataset we have chosen is from Kaggle, where it gives data related to the startup project. We can use this data to derive some descriptive visualizations like how many projects were successful, which country has invested more in these projects, whether the number of successful projects is increasing year by year, etc.

Kaggle Dataset Link:

https://www.kaggle.com/datasets/ulrikthygepedersen/kickstarter-projects

Domain Knowledge:

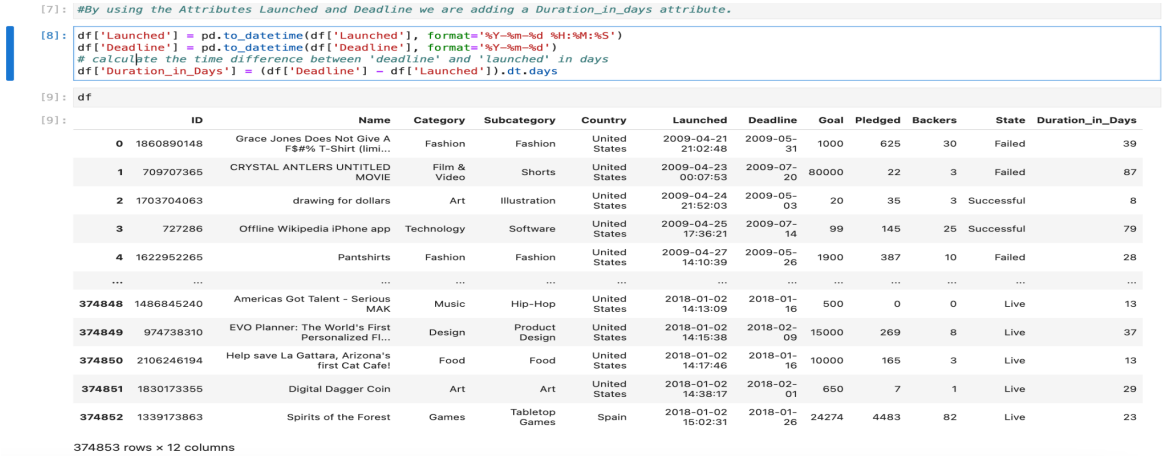
- AWS S3
- AWS Sage Maker for Analytics and the ML Process
- AWS Quick Sight for the Creation of Dashboard Visualizations

Data Preparation:

The dataset was checked for any missing values and those were removed to ensure accurate analysis. To enhance the analysis, a new attribute called 'Duration_in_days' was created using the 'Launched' and 'Deadline' attributes. Additionally, null values were checked and examined for each column, as they can hinder the machine learning algorithm's ability to learn. Afterwards, each column's data was observed thoroughly, and statistical measures such as standard deviation, mean, maximum, and other distributions were analyzed to determine the evenness of the data distribution.

Using the attributes Launched and Deadline, we have added a new column, i.e., the 'Duration_in_days' attribute, for our analysis.

Screenshot:



```
[7]: #By using the Attributes Launched and Deadline we are adding a Duration_in_days attribute.

[8]: df['Launched'] = pd.to_datetime(df['Launched'], format='%Y-%m-%d %H:%M:%S')
     df['Deadline'] = pd.to_datetime(df['Deadline'], format='%Y-%m-%d')
     # calculate the time difference between 'deadline' and 'launched' in days
     df['Duration_in_Days'] = (df['Deadline'] - df['Launched']).dt.days

[9]: df
```

[9]:

	ID	Name	Category	Subcategory	Country	Launched	Deadline	Goal	Pledged	Backers	State	Duration_in_Days
0	1860890148	Grace Jones Does Not Give A F$#% T-Shirt (limi...	Fashion	Fashion	United States	2009-04-21 21:02:48	2009-05-31	1000	625	30	Failed	39
1	709707365	CRYSTAL ANTLERS UNTITLED MOVIE	Film & Video	Shorts	United States	2009-04-23 00:07:53	2009-07-20	80000	22	3	Failed	87
2	1703704063	drawing for dollars	Art	Illustration	United States	2009-04-24 21:52:03	2009-05-03	20	35	3	Successful	8
3	727286	Offline Wikipedia iPhone app	Technology	Software	United States	2009-04-25 17:36:21	2009-07-14	99	145	25	Successful	79
4	1622952265	Pantshirts	Fashion	Fashion	United States	2009-04-27 14:10:39	2009-05-26	1900	387	10	Failed	28
...	...	...	...	...	...	...	...	...	...	...	...	...
374848	1486845240	Americas Got Talent - Serious MAK	Music	Hip-Hop	United States	2018-01-02 14:13:09	2018-01-16	500	0	0	Live	13
374849	974738310	EVO Planner: The World's First Personalized Fl...	Design	Product Design	United States	2018-01-02 14:15:38	2018-02-09	15000	269	8	Live	37
374850	2106246194	Help save La Gattara, Arizona's first Cat Cafe!	Food	Food	United States	2018-01-02 14:17:46	2018-01-16	10000	165	3	Live	13
374851	1830173355	Digital Dagger Coin	Art	Art	United States	2018-01-02 14:38:17	2018-02-01	650	7	1	Live	29
374852	1339173863	Spirits of the Forest	Games	Tabletop Games	Spain	2018-01-02 15:02:31	2018-01-26	24274	4483	82	Live	23

374853 rows × 12 columns

2.Checked if there are any missing values in the dataset to eliminate them.

Screenshot:

```
[17]: # Check the number of missing values in each column
      print(df.isnull().sum())
```

```
ID                0
Name              0
Category          0
Subcategory       0
Country           0
Launched          0
Deadline          0
Goal              0
Pledged           0
Backers           0
State             0
Duration_in_Days  0
dtype: int64
```

```
[18]: #The dataset do not have any missing values so it doesn't lead to biased or inaccurate analysis and modeling.
```

Data Exploration:

Exploratory data analysis was performed using AWS SageMaker. EDA helps to uncover insights and patterns within the data, validate assumptions, and determine the best approach to analyze the data. The hyperlink to the corresponding file is provided below.
https://github.com/PavanKalyanReddyM/BDA_Project/blob/main/Delivarable2_BDA.ipynb

1.Correlation between Goal, Pledged and Backers

```
[22]: import matplotlib.pyplot as plt
      import seaborn as sns
```

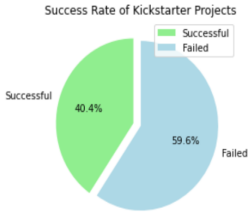
```
[23]: #Correlation between Goal, Pledged and Backers
      df2 = df1.drop("ID", axis=1)
      corr = df2.corr()
      #corr = df.corr(numeric_only=True)

      # Create a heatmap to visualize the correlation matrix
      sns.heatmap(corr, cmap="Blues", annot=True)
      plt.show()
```

2.How many projects were successful and how many failed?What is the overall success rate of Kickstarter projects?

[24]:
```
#How many projects were successful and how many failed?What is the overall success rate of Kickstarter projects?
```

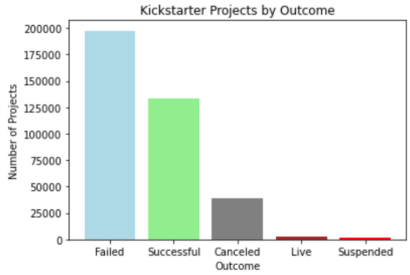
[25]:
```
#Success Rate
```

[26]:
```
success_rate = df["State"].value_counts(normalize=True)["Successful"]
failure_rate = df["State"].value_counts(normalize=True)["Failed"]

# Create a pie chart to visualize project success rates
labels = ["Successful", "Failed"]
sizes = [success_rate, failure_rate]
colors = ["lightgreen", "lightblue"]
explode = (0.1, 0)  # explode the "Successful" slice

plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct="%1.1f%%", startangle=90)

# Add title and legend
plt.title("Success Rate of Kickstarter Projects")
plt.legend(labels, loc="best")

# Show the plot
plt.show()
```



3. Count the number of projects by state.

[27]:
```
# Count the number of projects by state

data = df["State"].value_counts()
# Create a bar chart
plt.bar(data.index, data.values, color=["lightblue", "lightgreen", "gray","brown","red"])

# Add title and labels
plt.title("Kickstarter Projects by Outcome")
plt.xlabel("Outcome")
plt.ylabel("Number of Projects")

# Show the plot
plt.show()
```



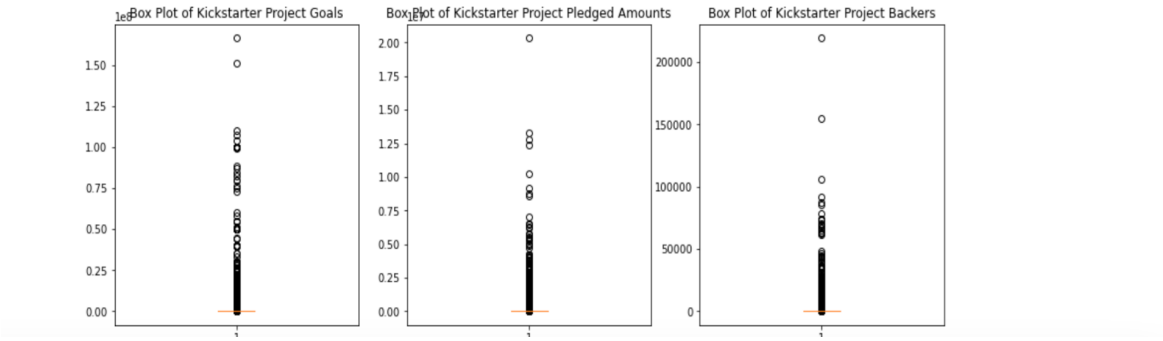
4. Count of projects by category by WordCloud Visualization

[29]:
```
#Count of number of projects by category by WordCloud Visualization
from wordcloud import WordCloud
```

[30]:
```
data = df["Category"].value_counts()

wordcloud = WordCloud(width=300, height=100, background_color="lightblue", colormap="tab10").generate_from_frequencies(data)

plt.figure(figsize=(7, 7), facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```

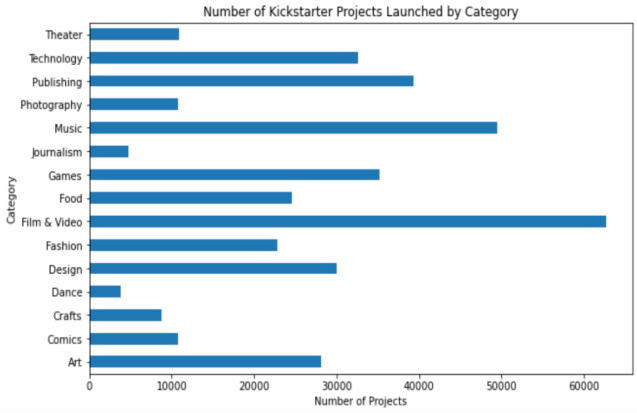
5. Outliers for Goal, Pledged,and Backers Fields

[31]:
```
# Outliers for Goal, Pledged and Backers Fields
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Create a horizontal box plot for the "Goal" attribute
axes[0].boxplot(df["Goal"], vert=True)
axes[0].set_title("Box Plot of Kickstarter Project Goals")
axes[0].set_xlabel("Goal Amount (USD)")

# Create a horizontal box plot for the "Pledged" attribute
axes[1].boxplot(df["Pledged"], vert=True)
axes[1].set_title("Box Plot of Kickstarter Project Pledged Amounts")
axes[1].set_xlabel("Pledged Amount (USD)")

# Create a horizontal box plot for the "Backers" attribute
axes[2].boxplot(df["Backers"], vert=True)
axes[2].set_title("Box Plot of Kickstarter Project Backers")
axes[2].set_xlabel("Number of Backers")

plt.show()
```



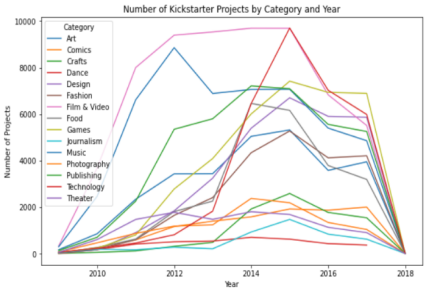
6. How many projects were launched for each category?

[32]:
```
## How many projects were launched for each category
import matplotlib.pyplot as plt

# Create a horizontal bar chart of the category counts
fig, ax = plt.subplots(figsize=(10, 6))
category_counts = df.groupby("Category")["ID"].count()
category_counts.plot(kind="barh", ax=ax)
#category_counts.plot(kind="barh", ax=ax, color="lightblue")


# Set the chart title and axis labels
ax.set_title("Number of Kickstarter Projects Launched by Category")
ax.set_xlabel("Number of Projects")
ax.set_ylabel("Category")
plt.show()
```



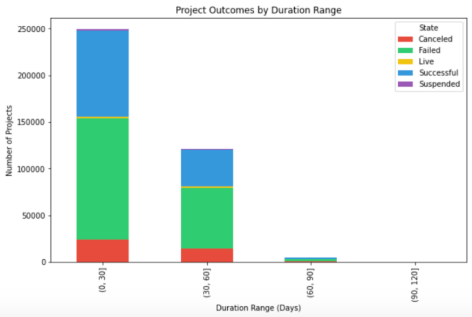
7. What is the overall trend of crowdfunding projects over time, and how does this vary across different categories?

[34]:
```
#What is the overall trend of crowdfunding projects over time,and how does this vary across different categories?
import matplotlib.pyplot as plt

# Convert the Launched and Deadline columns to datetime
df["Launched"] = pd.to_datetime(df["Launched"])
df["Deadline"] = pd.to_datetime(df["Deadline"])

# Extract the year from the Launched column and create a new column
df["Year"] = df["Launched"].dt.year

# Group the DataFrame by category and year and count the number of projects in each group
category_year_counts = df.groupby(["Category", "Year"])["ID"].count()

# Unstack the multi-level index to create a pivot table
pivot_table = category_year_counts.unstack(level=0)

# Create a line chart of the number of projects by category and year
fig, ax = plt.subplots(figsize=(10, 6))
pivot_table.plot(kind="line", ax=ax)

# Set the chart title and axis labels
ax.set_title("Number of Kickstarter Projects by Category and Year")
ax.set_xlabel("Year")
ax.set_ylabel("Number of Projects")

plt.show()
```



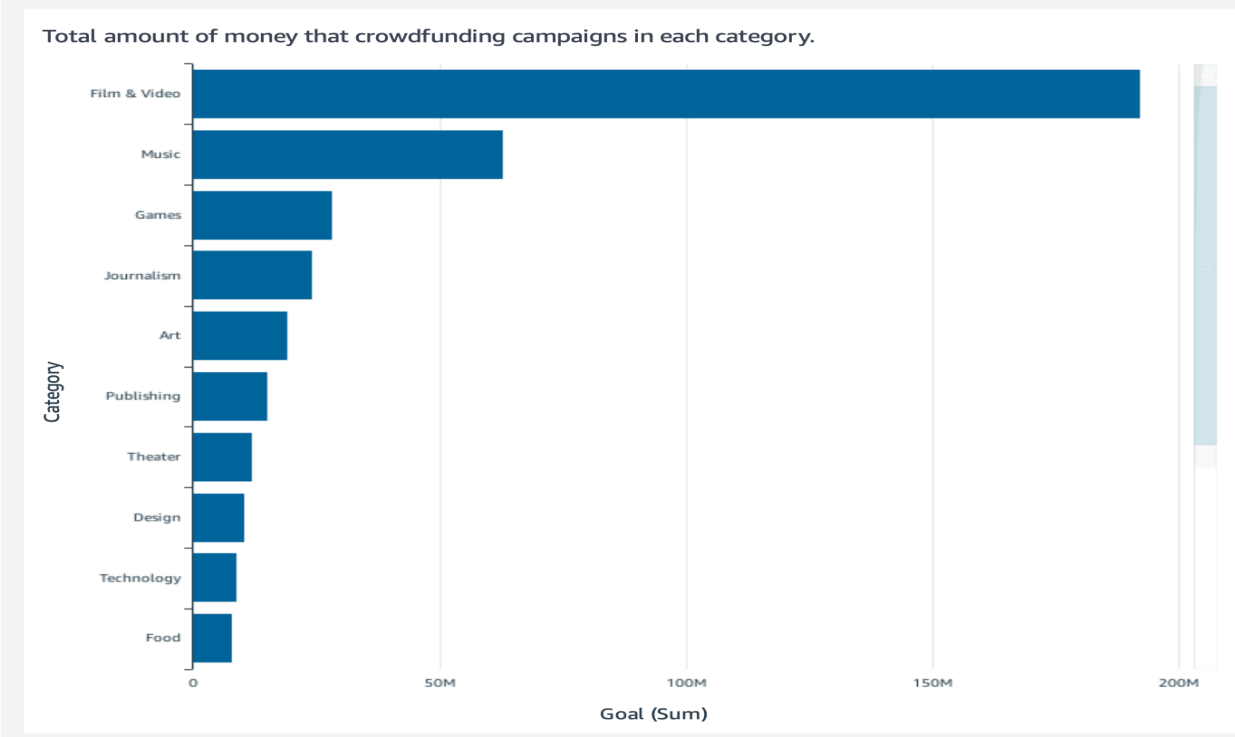
8. Project Outcome by Duration Range

[35]:
```
#Project Outcome by Duration Range
import pandas as pd
import matplotlib.pyplot as plt

# Load the data

# Create a new column for duration in days
df["Duration_in_Days"] = (pd.to_datetime(df["Deadline"]) - pd.to_datetime(df["Launched"])).dt.days

# Define the duration ranges
duration_bins = [0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480, 510, 540, 570, 600]

# Cut the duration data into the defined ranges
df["Duration_Range"] = pd.cut(df["Duration_in_Days"], bins=duration_bins)

# Create a cross-tabulation table of the duration ranges and project states
duration_state_counts = pd.crosstab(df["Duration_Range"], df["State"])

colors = ["#E74C3C", "#2ECC71", "#F1C40F", "#3498DB", "#9B59B6", "#34495E"]

# Plot a stacked bar chart of the cross-tabulation table with custom colors
duration_state_counts.plot(kind="bar", stacked=True, figsize=(10,6), color=colors)

# Set the chart title and axis labels
plt.title("Project Outcomes by Duration Range")
plt.xlabel("Duration Range (Days)")
plt.ylabel("Number of Projects")

plt.show()
```

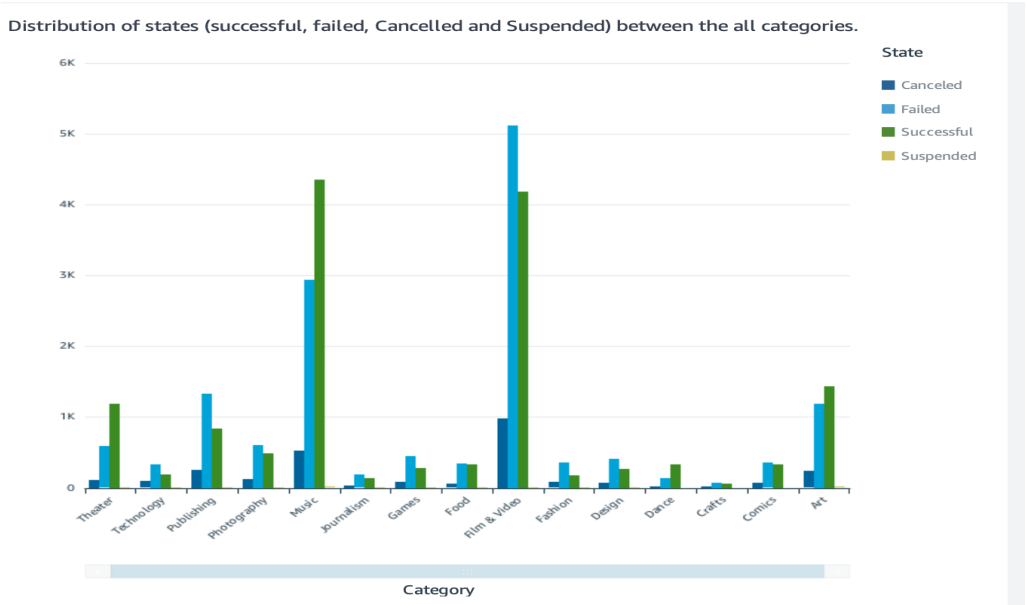


Data Visualization:

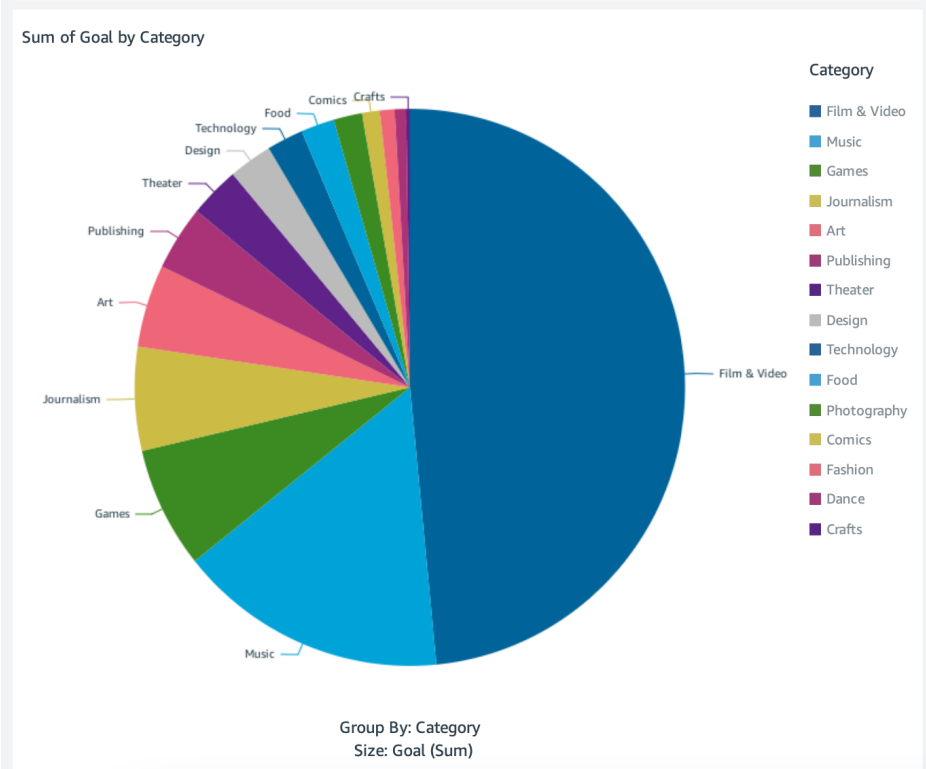
1.Total amount of money raised by crowdfunding campaigns in each category.



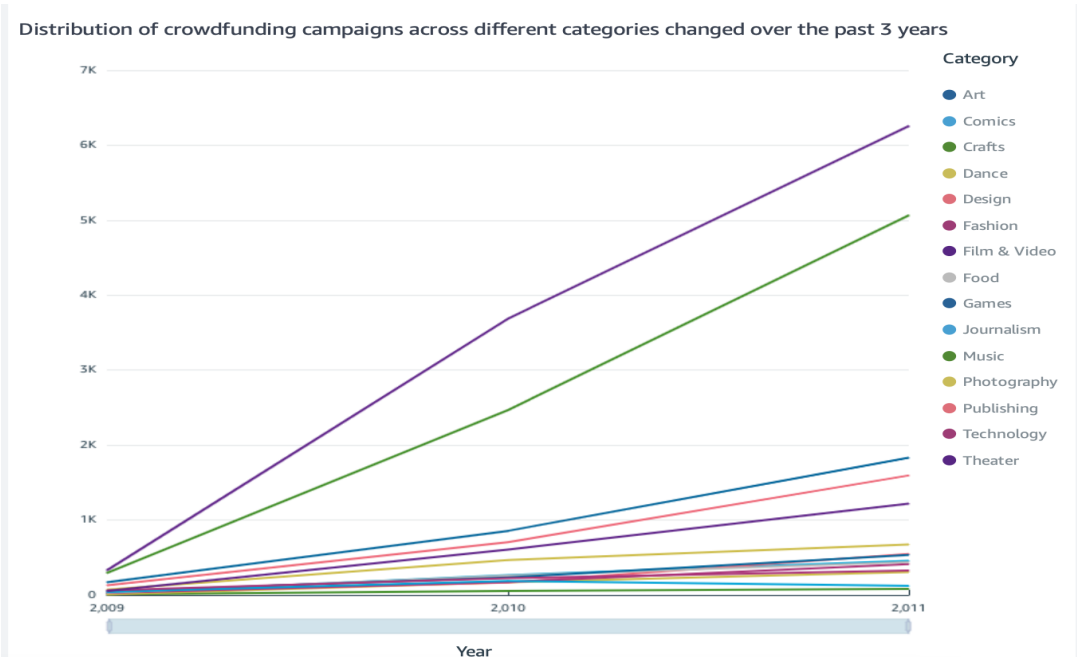
2. Distribution of states (successful, failed, Cancelled and Suspended) between the all categories.



3.Sum of Goals by Category

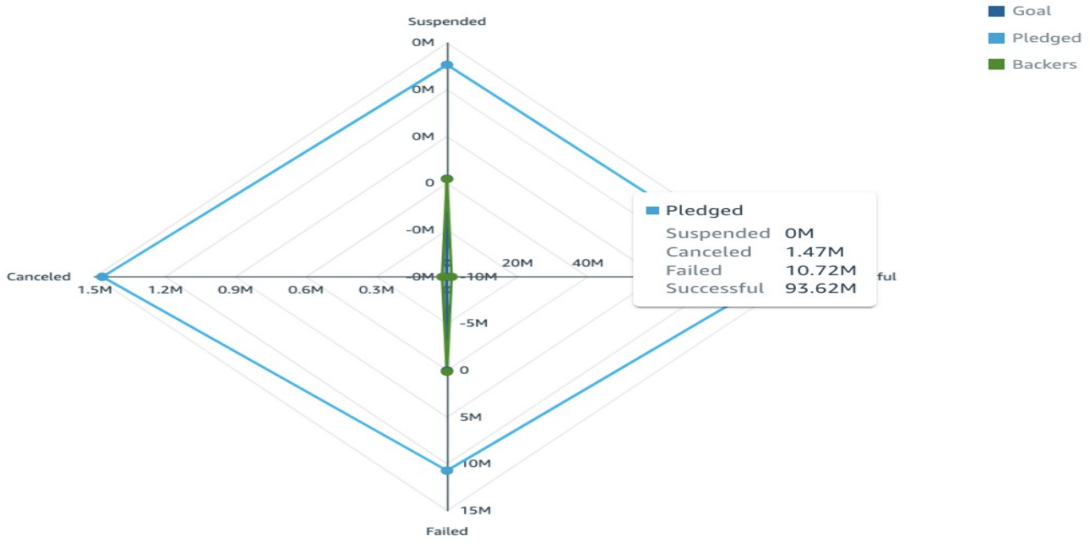


4.Distribution of crowdfunding campaigns across different categories changed over the past 3 years

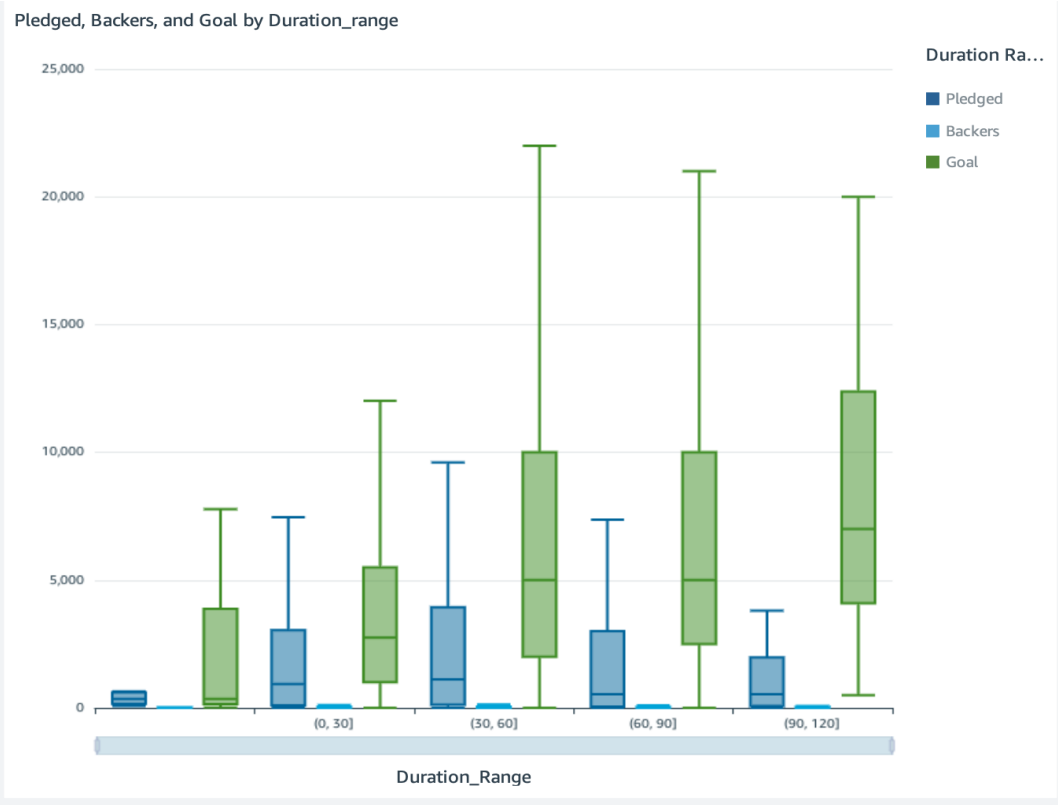


5.Count of Records by Subcategory.

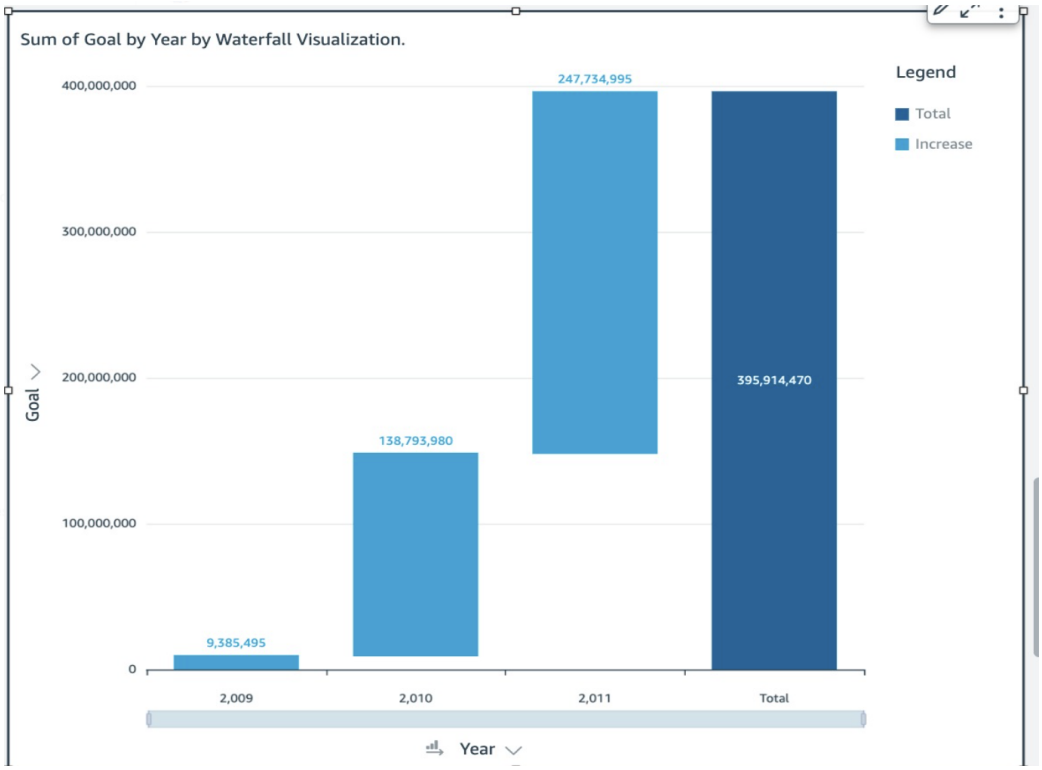
6.Count of Goal, Sum of Pledged, and Sum of Backers by State



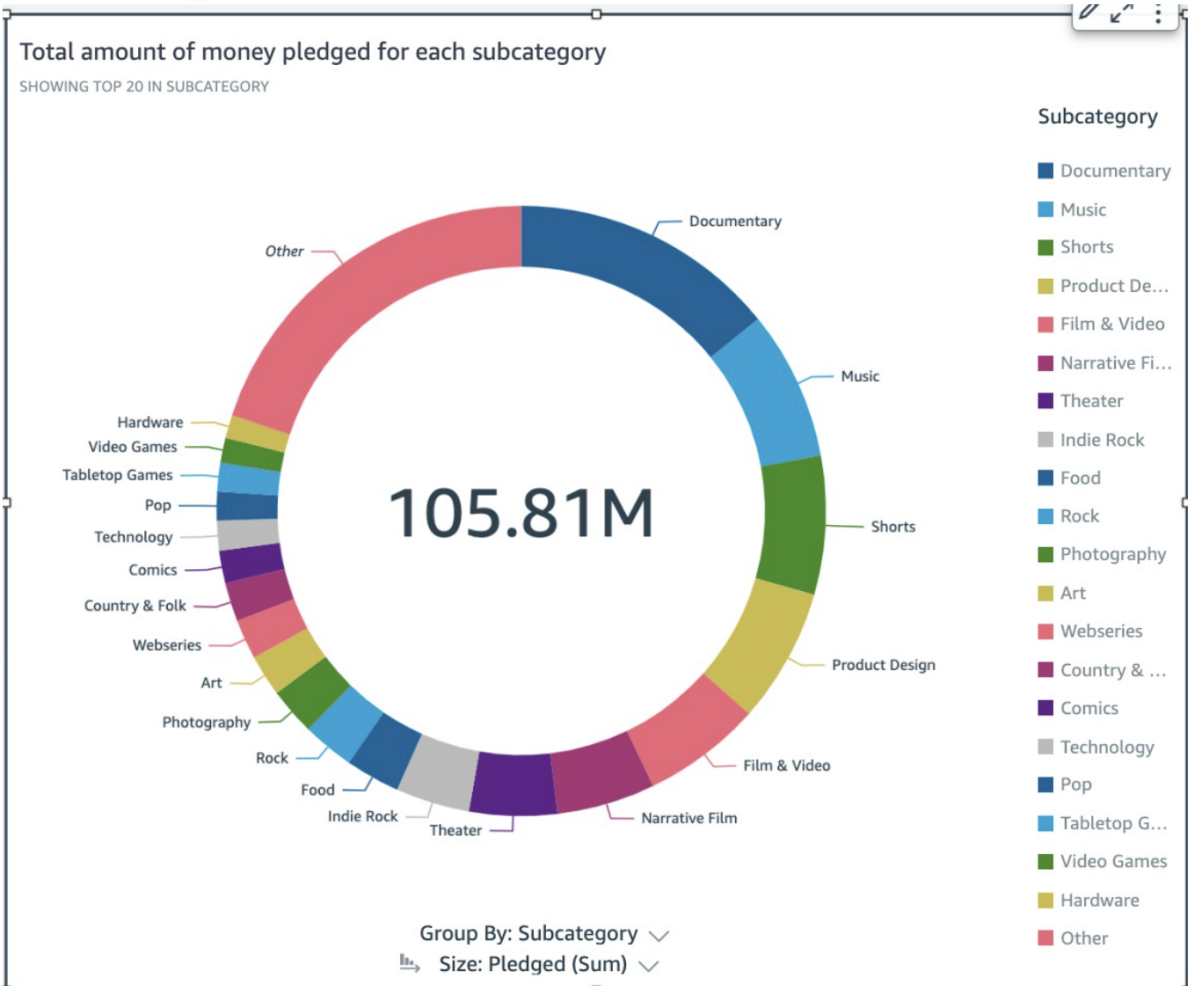
7.Pledged, Backers, and Goal by Duration_range



8.Sum of Goal by Year by Waterfall Visualization.



9.How many different campaign durations are there for each category.

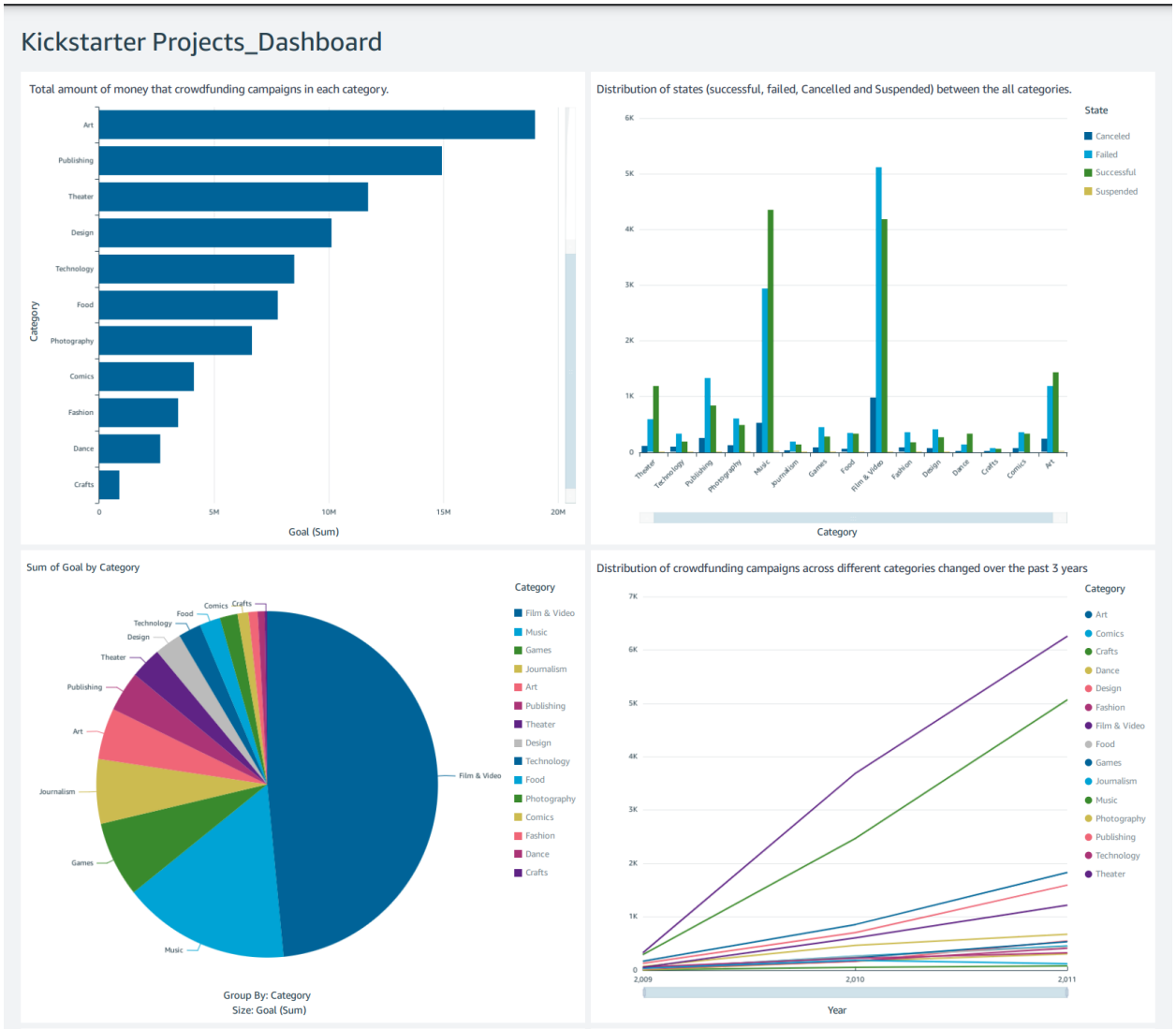


Dashboard:

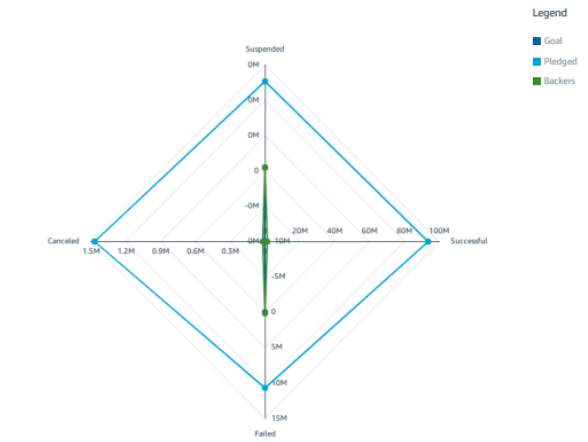
Dashboard instances displaying visualizations like pie charts, bar charts, word clouds, line graphs, outliers, and correlation graphs were made using AWS QuickSight. Research objectives are evaluated using these visualizations. Additionally, other plots were also produced using an AWS Amazon SageMaker notebook.

The PDF document contains several graphs generated using AWS QuickSight. Please find the link to access the PDF document below.

https://github.com/PavanKalyanReddyM/BDA_Project/blob/main/Dashboard_KickstarterProjects.pdf

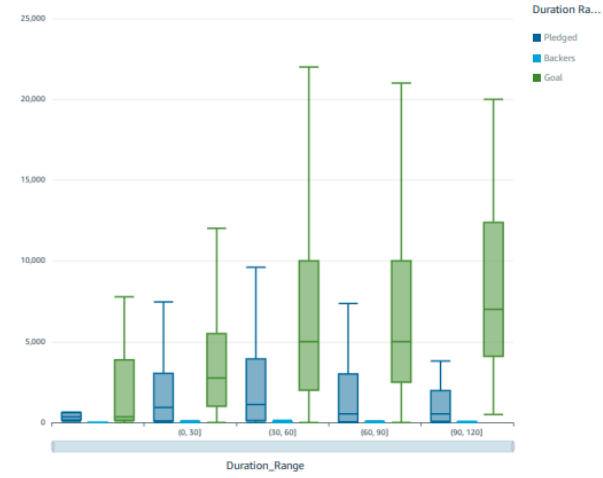


Count of Goal, Sum of Pledged, and Sum of Backers by State

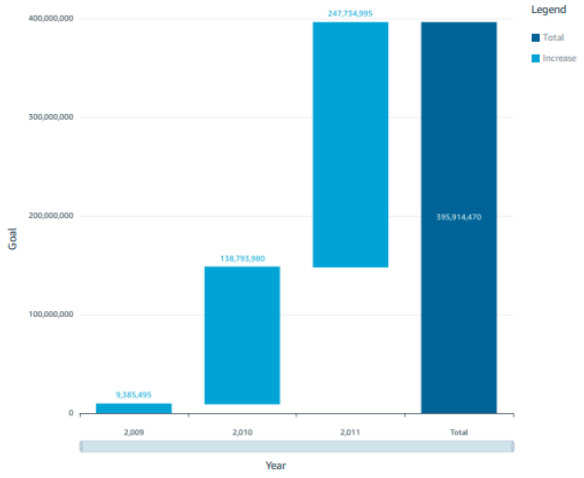


Count of Records by Subcategory

Pledged, Backers, and Goal by Duration_range

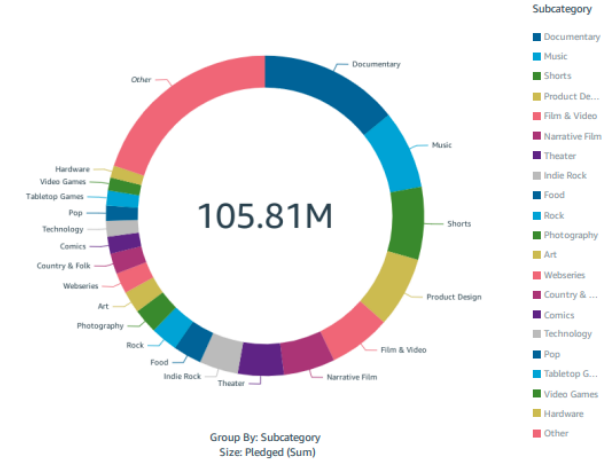


Sum of Goal by Year by Waterfall Visualization.



Total amount of money pledged for each subcategory

SHOWING TOP 20 IN SUBCATEGORY



Group By: Subcategory
Size: Pledged (Sum)