Unlocking Donor Insights:

A K-means Exploration of Crowdfunding Dynamics

Prepared by:

Sohaila Abdel-Moniem Ahmed

P-EM0236/23

Supervised by:

Ts. Dr. Normalini Kassim

School Of Management

Academic Session 2023/2024 | Semester 1

Master Of Business Analytics

Abw508d Analytics Lab

Table of Contents

Acknowledgement

I would like to express my sincere gratitude to Ts. Dr. Normalini Kassim for her unwavering support and invaluable guidance throughout this research journey. Her encouragement and the assistance she provided were instrumental in shaping the direction of this study. I would also like to express my gratitude to Eng. Heba Sherif, my Debi supervisor, for her caring and supportive nature throughout the entire process. I extend my appreciation to Professor T. Ramayah for imparting invaluable knowledge in the research methodology course during the first semester, laying a strong foundation for this project. I am also thankful to all the professors who, through their expertise and guidance, further enriched my knowledge in various aspects of my academic journey. Special thanks to my colleagues for their collaborative spirit and guidance, contributing to a dynamic and enriching research environment. I am deeply grateful to my family for their unwavering support and encouragement, providing the foundation for my academic pursuits. Additionally, heartfelt thanks to my friends for their understanding and encouragement, making this academic endeavor a collective achievement.

List of Figures

List of Tables

Abstract

This research delves into the evolving landscape of crowdfunding platforms, addressing the challenge of identifying optimal target populations for digital fundraising marketing campaigns. Traditional one-size-fits-all approaches are deemed inadequate, prompting a transition towards personalized interactions. Leveraging the K-means clustering algorithm, this study achieves a silhouette score of 0.396, indicative of well-defined donor segments. Notable implications emerge, unveiling distinct donor profiles based on age, device usage, and engagement categories. This nuanced understanding empowers organizations to tailor communication and outreach strategies effectively. Gender-based trends and device preferences uncovered in the clusters provide actionable insights for targeted appeals. The study holds significance for nonprofits, crowdfunding platforms, and marketing professionals, offering practical strategies to enhance donor segmentation, optimize user experiences, and tailor promotional efforts. The integration of clustering results with actionable recommendations enhances the potential impact of this study, revolutionizing marketing strategies on crowdfunding platforms and contributing to the field of data-driven marketing analytics.

INTRODUCTION

The evolution of crowdfunding demands a shift from one-size-fits-all marketing to personalized campaigns catering to donors' unique preferences. Identifying high-value segments within crowdfunding platforms has become crucial to maximizing outreach and campaign success. Traditional marketing strategies often prove inadequate, relying solely on broad demographics and neglecting the intricate behavioral patterns underlying donor activity.

This research proposes a data-driven approach employing the K-means clustering algorithm and Power BI visualization to unlock significant insights within vast donor datasets. By meticulously analyzing historical contributions, socio-demographic characteristics, and device usage, the aim is to segment crowdfunding audiences into distinct groups with clearly defined preferences and behaviors. This segmentation will empower platforms to design targeted campaigns that resonate deeply with each segment, fostering increased engagement and optimizing campaign efficacy.

The anticipated outcomes extend beyond revolutionizing crowdfunding marketing strategies. This research also contributes to the burgeoning field of data-driven marketing analytics, offering valuable insights applicable across diverse industries. By bridging the gap between data analysis and actionable strategies, this study paves the way for enhanced engagement, amplified campaign impact, and sustainable success in the dynamic world of online fundraising.

Building the foundation for this research, this chapter delves into the background, clearly outlines the problem statement, defines the research objectives, and questions, emphasizes the significance of the study, and maps out the overall structure. This comprehensive overview equips stakeholders with a clear understanding of the research goals and facilitates their engagement with the findings.

1.0 Background of the Study

Crowdfunding is a fundraising approach in which small financial donations are gathered from a potentially large number of backers over the Internet, often without the participation of traditional

financial intermediaries (Mollick, 2014). Crowdfunding can be divided into four categories. Reward-based, equity-based, lending-based, or donation/charity-based. Unlike the former three types, donation-based crowdfunding involves no expectation of monetary gain in exchange for participation (Li et al. 2020). Crowdfunding systems that perform transactions online leave a massive digital trace of their online users or donors. Because of the large volume of transactions, these platforms have a deep reservoir of data from which to collect information about their customers, segment them for generating insights, and increase user lifetime value (Lim & Wang, 2023).

Nowadays because of the internet and the advent of new Information Technologies and Communication, modern consumers are adopting new purchasing habits. The customer is becoming increasingly accountable and engaged, he does not consume more just to equip himself or meet basic requirements. Because of this societal shift, mass consumption is disrupted, and decision-makers are impacted by moral or political opinions. Therefore, Marketers must undertake a gradual shift to data-driven decision-making. To do this, businesses must concentrate on continuous data collection since customers leave valuable information about themselves and what they are seeking. Data gathered from outside sources should be used to provide a comprehensive picture of the customer and their demands. These strategies will make marketing campaigns more relevant (Abakouy et al., 2019).

The concept of data-driven marketing as demonstrated by Sundsøy et al., 2014, is the practice of using data to drive marketing. It helps marketers understand their consumers, build successful campaigns, and deliver them to relevant audiences, which leads the way for customer segmentation.

Customer segmentation is collecting and analyzing customer socio-demographics, psychographics, behavioral data, and other descriptive data. The assumption is that differences between customer segments are observable and explainable by demographics and socioeconomics including personal characteristics such as age, sex, ethnicity, occupation, and marital status. In other words, consumers who share similar observable characteristics are likely to respond in similar ways to marketing messages (Fader et al., 2005; Kumar, 2018).

For this research, the focus extends to the utilization of clustering techniques for customer segmentation in crowdfunding. Customer segmentation becomes essential in this context as it enables personalized targeting of contributors with similar preferences and behaviors. The significance lies in optimizing marketing strategies for crowdfunding campaigns and tailoring efforts to the distinct needs of segmented customer groups. This approach aims to maximize the effectiveness of promotional endeavors, enhance user engagement, and contribute to the success of crowdfunding initiatives.

1.1 Problem Statement

Crowdfunding platforms are vibrant ecosystems in the dynamic world of digital fundraising, effortlessly linking creative project creators with a worldwide audience ready to support unique ideas. As the popularity of crowdfunding grows, navigating this dynamic space becomes more difficult. The main challenge is determining the optimal target population for potential marketing campaigns.

Marketing strategies have always taken a one-size-fits-all approach, but the digital world necessitates a shift from this traditional strategy as donors now are increasingly looking for individualized interactions and campaigns that are targeted to their preferences.
To address this challenge, this paper's focus shifts to advanced clustering techniques, specifically the use of the K-means algorithm. The goal is to employ data-driven analysis to precisely identify segments and understand donor behavior within the context of online crowdfunding platforms.

1.2 Research Objectives

The main objective of this study is to identify the customers with the highest donation history for a crowdfunding platform. Therefore, the following objectives are the primary goals of this research.

- Identify distinct donor segments within crowdfunding platforms using K-means clustering.

- Analyze the characteristics and preferences of each donor segment (age, device, project category, gender, and donation amount).
- Provide crowdfunding platforms with actionable insights, enabling them to make tailored campaigns that resonate with the preferences of each donor segment.

1.3 Research Questions

To achieve the research objectives, these questions are formulated.

- How does the K-means algorithm contribute to the identification of specific donor segments on crowdfunding platforms?
- How do donor segments on crowdfunding platforms differ in terms of age, device preference, project category, gender, and donation amount?
- How can the identified donor segments on crowdfunding platforms inform the development of tailored campaigns to resonate with the preferences of each segment?

1.4 Significance Of the Research

The expected results aim to revolutionize marketing strategies on crowdfunding platforms. Insights derived from the results will inform targeted advertising efforts to make impactful crowdfunding campaigns and contribute to the growing field of data-driven marketing analytics.

1.5 Research Structure

This study was divided into five chapters, which are as follows: introduction, literature review, methodology, results, and conclusion.

The content of chapter one provides the introduction, background of the study, problem statements, research objective, significance of research, and research structure. In chapter two, the literature review will be illustrated, laying the conceptual foundation for this study by conducting a comprehensive review of relevant literature. Following that, in chapter three, the methodology will be provided, elaborating on the methodology and the model employed in the research. Results and

findings will be addressed in the fourth chapter, in addition to a discussion of the implications and limitations of the study. Finally, there will be a conclusion explaining the recommendations for future academics and an overall conclusion.

LITERATURE REVIEW

2.1 Crowdfunding:

According to Mollick (2014), crowdfunding is described as a tool enabling entrepreneurs to gather funds from a large online community, connecting them with backers willing to contribute small amounts of capital to selected projects. Hossain & Oparaocha (2017) define this internet-based funding method, highlighting the significance of time, specifically within a limited timeframe. Their definition underscores the collective effort of a group of individuals to support an initiative by making online pledges of modest monetary sums. In essence, crowdfunding is the process by which individuals or groups – known as initiators, founders, or entrepreneurs – gather monetary funds from the community using an internet platform in order to obtain support for a project. The number of people making a tiny or large monetary donation - known as supporters or funders - might vary depending on the level of individual support (Jáki et al., 2022). In academic discourse, crowdfunding is categorized into four distinct types. In donation-based crowdfunding, contributors do not receive any form of compensation for their financial support. This model is predominantly utilized by non-profit and non-governmental organizations (Hörisch, 2015; Lehner, 2013). Reward-based crowdfunding involves supporters receiving tangible or intangible rewards for their investments, often in the form of the product being funded. Mollick (2014) notes that reward-based crowdfunding is the most commonly employed form. Additionally, there are two investment-based crowdfunding types, both involving the distribution of monetary returns among investors. Equity-based crowdfunding, also known as crowdinvesting, grants financial returns to investors if the venture proves profitable (Mochkabadi & Volkmann, 2018). Similar to stock market investments, this crowdfunding type is linked to the highest risk for investors (Bapna, 2019). Lastly, lending-based crowdfunding, also referred to as debt-based crowdfunding or crowdlending, resembles a traditional bank loan. Here, supporters function as lenders and receive a predetermined interest rate within a specified timeframe (Bruton et al., 2015). Lending-based crowdfunding constitutes the largest portion of the global funding volume generated through crowdfunding (Massolution, 2015).

Crowdfunding provides several advantages for entrepreneurs, with its primary function being the funding of novel ideas or existing ventures (Lehner, 2013). Additionally, crowdfunding serves as a valuable tool for marketing purposes (Hörisch, 2018). It has the potential to enhance visibility among prospective customers, the broader public, and the media (Burtch et al., 2014; Lambert & Schwienbacher, 2010; Mollick, 2014). Moreover, crowdfunding serves as an effective market test, offering insights into the level of interest from potential users in a given crowdfunding campaign's offering (Bellefamme et al., 2014; Lam & Law, 2016).

The crowdfunding process involves two distinct funding phases, consistent across various crowdfunding types, as identified in the scientific literature (Jovanovic, 2019; Hörisch, 2019). These phases are the pre-funding phase and the post-funding phase. The pre-funding phase spans the duration until funding on the crowdfunding platform is complete. It encompasses campaign preparation, communication, marketing to target groups, and the active funding period. On the other hand, the post-funding phase commences once the crowdfunding campaign concludes. During this phase, project initiators are responsible for communicating outcomes to supporters, fulfilling promised returns, and, most importantly, executing the project by implementing the advertised initiatives.

In this paper, the emphasis will be on the pre-funding phase of the crowdfunding process, specifically directed towards optimizing the marketing campaign to effectively target the right audience.

2.2 Customer Segmentation

Customer segmentation is the process of leveraging various distinctive customer characteristics, aiding business professionals in tailoring marketing strategies, recognizing emerging trends, planning product development, executing advertising campaigns, and delivering products that resonate with the target audience. The customization of messages for individuals enhances communication effectiveness with specific target groups. Common attributes utilized in customer segmentation include location, age, gender, income, lifestyle, and past purchase behavior (Christy et al., 2021).

According to Sabuncu et al., (2020), customer segmentation serves as a pivotal mechanism for establishing efficient communication channels with customers. Through the partitioning process, characteristics of latent customer groups are identified within the dataset. This partitioning serves as a preliminary step for classifying the identified customer groups. Segmentation empowers marketers to optimize resource allocation and enhances their ability to identify opportunities effectively. Furthermore, customer segmentation is an unsupervised learning process employing various clustering methods to categorize customer data based on similarity. Similarity is gauged through an objective function like Euclidean distance. It is essential to recognize that customer behavior is a continual process, characterized by evolving needs, desires, and satisfaction. Consequently, organizational processes and underlying procedures need to be adaptable to accommodate this dynamic nature (Liu et al., 2009; Ding et al., 2019; Griva et al., 2021).

2.3 Implications of Clustering in Customer Segmentation

Employing clustering techniques is a highly effective approach for businesses and companies to classify distinct customer segments, facilitating precise targeting of potential user bases. Sembiring Brahmana et al., (2020) utilized three clustering algorithms, namely K-Means, K-Medoids, and DBSCAN, with the application of RFM (Recency, Frequency, Monetary) models in customer segmentation. These algorithms are commonly employed due to their simplicity and effectiveness in revealing diverse customer classes.

Another study conducted an RFM analysis on transactional data and utilized traditional K-means and Fuzzy C-means algorithms for clustering. The paper aimed to propose an efficient method for selecting initial centroids in the K-Means algorithm to segment customers with reduced iteration and time (Christy et al., 2021). Additionally, Li et al. (2022) aimed to develop a method for segmenting customers within the grape market in China. To enhance the precision of customer segmentation, the research introduces a K-means clustering algorithm incorporating adaptive learning particle swarm optimization (ALPSO). The ALPSO algorithm is customized by redesigning inertia weight, learning factors, and position update methods based on dynamic population analysis, aiming to elevate the optimization accuracy of particle swarm optimization (PSO). The adapted ALPSO algorithm is subsequently applied to optimize the original K-means cluster centers, mitigating the dependence of K-means on initial points.

Abdulhafedh, A. (2021) employed two approaches for customer segmentation for a credit card company that has collected data about their customers' accounts. The first approach involved utilizing all variables with clustering algorithms using Hierarchical clustering and K-means. The second approach incorporated dimensionality reduction through Principal Component Analysis (PCA), identifying the optimal number of clusters, and then conducting clustering analysis with the updated number of clusters.

Building on previous research, this paper will leverage clustering for customer segmentation, employing the k-means technique.

METHODOLOGY

3.0 Methodology

This section outlines the procedural framework used to carry out this study. The study was performed using Jupyter Notebook (version 6.4.12), installed on a personal computer equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz CPU and 8.00 GB RAM and Windows 10 Pro as an operating system. The Python programming language, version 3.9.13, is used in the study's implementation.

Appendix A documents the entire procedural details, which include data retrieval, pre-processing, exploration, modeling, and evaluation. This documentation provides a thorough reference to the study's step-by-step process, assuring transparency and consistency in the research methodology.

3.1 Data Retrieval

The crowdfunding dataset used in this study is available and obtained from DataCamp, encompassing a wide range of categories. It is acquired in a comma-separated values (CSV) file format and consists of five features and 20658 records. The dataset was imported into Python using the read_csv() function from the Python Pandas package, converting raw data to a structured DataFrame, providing a foundation for systematic analysis with properly designated axes. The dataset's features are described in the below table.

Table 1: Data dictionary

Name of the feature	Description
Category	The type of crowdfunding project (e.g., Sports, Fashion, Technology).
Device	The type of device used for donation (desktop, mobile, tablet).
Gender	The gender of the donor (male, female).
Age	An age bracket representing the donor's age group (18-24, 25-34, 35-44, 45-54, 55+).
Amount	The donation amount in Euros.

This dataset retrieval process ensures that the raw data is seamlessly integrated into the analytical framework, setting the stage for subsequent exploration and interpretation.

3.2 Data Pre-processing

Data pre-processing is a fundamental phase in preparing the raw data for machine learning models. In this project, pre-processing encompasses several steps, including handling missing values, type conversion, handling inconsistent data, and encoding categorical variables.

3.2.1 Handling Missing Values

The function df.isnull().sum() is used to find if there are missing values in the dataset, as a result, there are no missing values in this dataset as illustrated in Figure 1.

```
# Checking for missing values
df.isnull().sum()
```

```
category    0
device      0
gender      0
age         0
amount      0
dtype: int64
```

Figure 1: checking missing values.

3.2.2 Type Conversion

Checking the data type for each feature is essential to ensure that all the features are in the appropriate data type, df.info() function has been employed for this purpose. As shown in Figure 2 below, the inspection reveals that every feature is in the appropriate data type, and no adjustments are required.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20658 entries, 0 to 20657
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   category  20658 non-null  object
 1   device    20658 non-null  object
 2   gender    20658 non-null  object
 3   age       20658 non-null  object
 4   amount    20658 non-null  float64
dtypes: float64(1), object(4)
memory usage: 807.1+ KB
```

Figure 2: checking features datatypes.

3.2.3 Handling Inconsistent Data

Data recoding is typically done to handle missing or ambiguous values in categorical variables. In the case of this dataset, the gender column has three values as shown in Figure 3, F for female, M for male, and U for unknown or unidentified. As It is a common practice to make the data more consistent and suitable for analysis, as some machine learning algorithms may struggle with missing or undefined values, the ‘’U’’ variable was replaced with ‘’F’’ which is the mode of the gender column, or the most frequent variable, using the function df['gender'].replace('U', 'F', inplace=True).

```
gender : ['F' 'M' 'U'] ( 3 )
```

Figure 3: variables of the gender column.

As a continuation of this step, checking for misspelled variables was also necessary to ensure data consistency, .unique() function is used to show the unique values of each variable in order to identify the values with the same meaning which are represented differently. Figure 4 shows the result, there was consistency and no misspelled variables.

```
for cat_col in df.select_dtypes(include='object_'):
    print(cat_col, ':' , df[cat_col].unique(), '(' ,df[cat_col].nunique(), ')')
```

```
category : ['Fashion' 'Sports' 'Technology' 'Games' 'Environment'] ( 5 )
device : ['iOS' 'android'] ( 2 )
gender : ['F' 'M'] ( 2 )
age : ['45-54' '18-24' '35-44' '55+' '25-34'] ( 5 )
```

Figure 4: unique values of each column.

3.2.4. Encoding Categorical Variables

Structured datasets are often made up of a combination of numerical and category columns. Nonetheless, machine learning algorithms are designed to interpret numerical data, which presents a barrier when dealing with categorical characteristics. To bridge this gap, the categorical columns must be encoded in order to be converted into numerical representations. The Scikit-learn pre-processing library's LabelEncoder module was used for this purpose. Based on alphabetical ordering, the LabelEncoder assigns a unique number to each label. This encoding approach converts categorical labels into numeric values ranging from 0 to n-1, where 'n' is the number of different classes in the category variable.

3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a necessary step in developing an efficient machine-learning model, it has a pivotal role in unveiling valuable insights. In this study, a comprehensive EDA was conducted, encompassing descriptive analysis of numerical and categorical data. Followed by univariate analysis allowed for a focused exploration of individual variables and bivariate analysis provided a glimpse into relationships between pairs of variables.

3.3.1 Descriptive Statistics

Descriptive statistics play a pivotal role in distilling meaningful insights from both numerical and categorical variables. For numerical variables, the implementation of the .describe() function furnishes a comprehensive summary encompassing central tendencies (mean, median), spread (standard deviation), and the overall distribution of the data. Meanwhile, when used to categorical variables, .describe() produces a frequency distribution summary. This includes the count, unique

categories, top category, and frequency of occurrence. Tables 2 and 3 below show a review of the output.

Table 2: Statistical Summary of Numerical Variables

	amount
count	20658.000000
mean	39.407009
std	14.913658
min	1.000000
25%	29.000000
50%	39.000000
75%	50.000000
max	101.000000

Table 3: Statistical Summary of Categorical Variables

	category	device	gender	age
count	20658	20658	20658	20658
unique	5	2	2	5
top	Sports	iOS	F	18-24
freq	4179	13459	11087	10439

3.3.2 Univariate Distributions

For the numerical variables, a combination of box plots and histograms was employed to gain insights into the distribution and central tendency of the 'amount' feature. Functions like sns.histplot() and sns.boxplot() were created to generate a side-by-side display of both a box plot and a histogram. The box plot provides an overview of the data's dispersion and identifies potential outliers, while the histogram visualizes the frequency distribution.

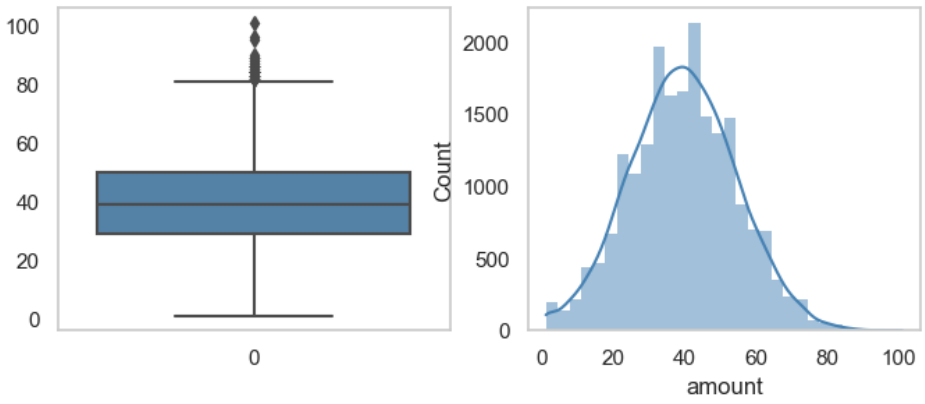


Figure 5: Univariate distribution for the amount column.

Figure 5 above shows that the boxplot and the histogram provide an overview of the distribution of donation amounts. They show that there is a mix of small and large donations, with a few very large outliers. The distribution is skewed to the right, which means that there are more small donations than large donations.

For categorical columns “category”, “device”, “age”, and “category”, a count plot was employed to visualize the distribution of each category within the variable. The sns.countplot() function was created to generate these count plots. Count plots help in understanding the frequency or count of each category within a categorical variable.

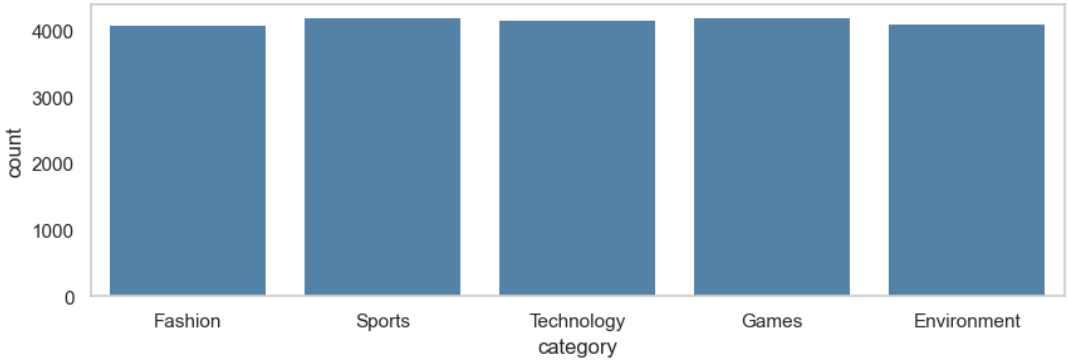


Figure 6: count plot for the category column.

As shown in Figure 6, the most popular fields are sports and games. However, the distribution of interest in the types of crowdfunding projects is relatively even across the five fields, with no one field having a significantly higher or lower number of people interested. This suggests that there is a diverse range of interests represented in the data.

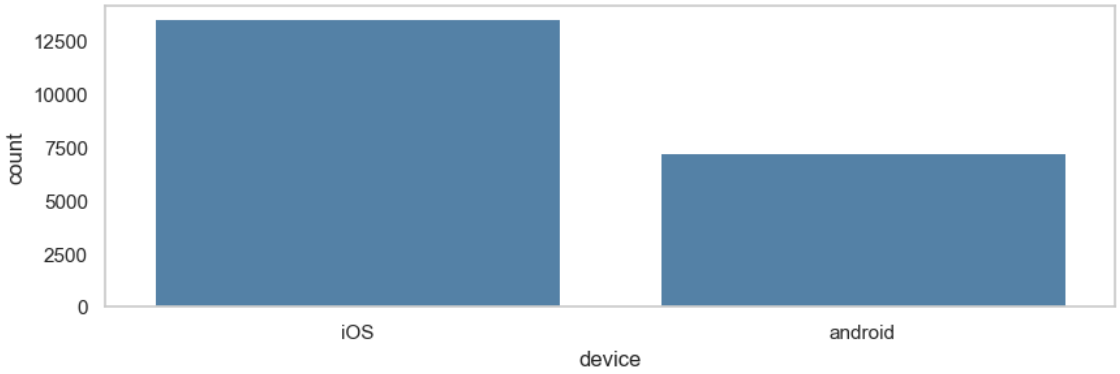


Figure 7: count plot for the device column.

Figure 7 illustrates that the most common device used is iOS, with over 12,500 devices, followed by Android at around 7,500 devices. This suggests that the majority of users access the crowdfunding platform through iOS devices.

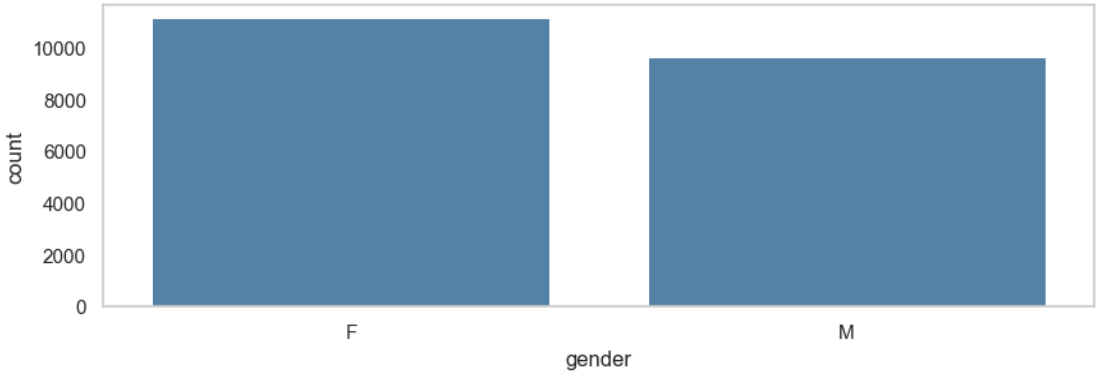


Figure 8: count plot for the gender column.

The data appears to be skewed towards females, with around 11,500 female users compared to about 9,000 male users as suggested by Figure 8 above.

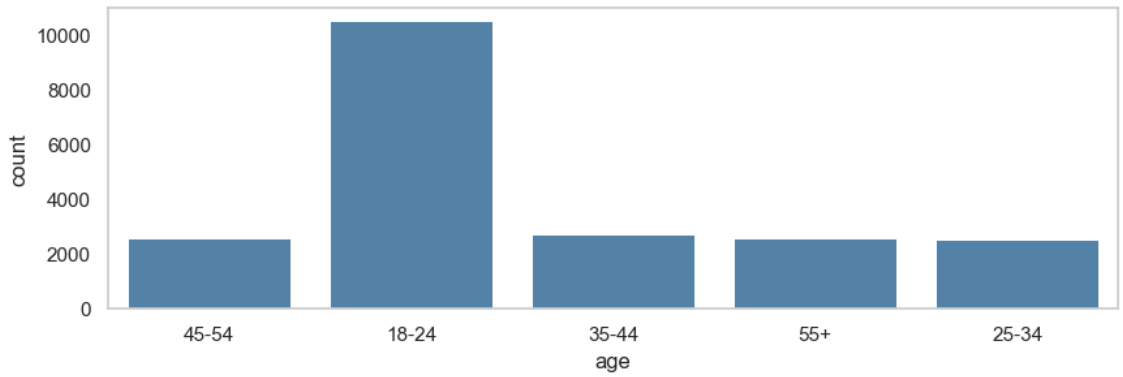


Figure 9: count plot for the gender column.

The age count plot in Figure 9 shows that the highest bar is for the "18-24" age group, most of the platform users' ages are in the range between 18 to 24.

3.3.3 Bivariate Distributions

Next bivariate analysis was conducted to understand the relationships and interactions between the variables. In this study, bivariate analysis was conducted for each categorical variable ('category', 'gender', 'device', 'age') in conjunction with the 'amount' column. The biplot() function was utilized to generate scatter plots illustrating the relationship between the categorical variable and the 'amount' column.

3.3.4 Exploratory Visualization

In research, exploratory visualization serves as a crucial step, directing data exploration without predetermined aims. It is a flexible tool that may be used by both unfamiliar and professional researchers who want to understand the possibilities and restrictions of visualization. In this context, our exploratory visualizations investigate the distribution of categories, donor device preferences, and differences in contribution quantities among age groups. This hands-on approach helps us to uncover hidden insights and patterns, resulting in a more thorough knowledge of the dataset. In this study, a combination of Power BI and Python's plotly.express library are used for data visualization, Appendix A and B include all of the exploratory visualization.

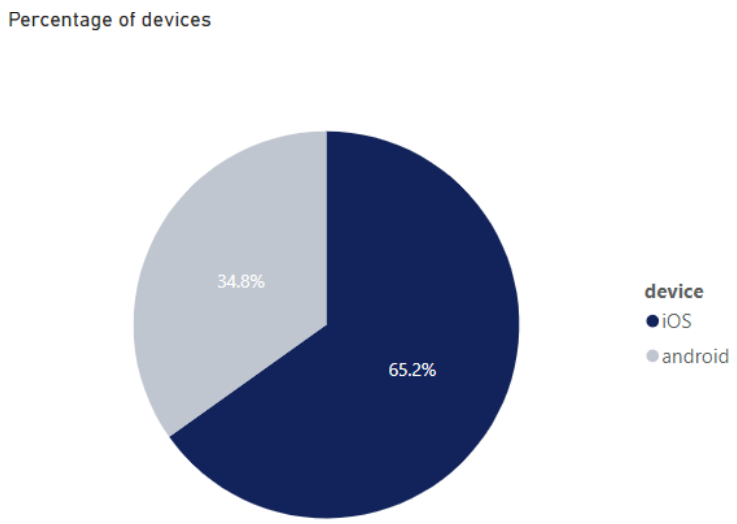


Figure 10: Pie chart for percentage of devices.

Figure 10 shows that the percentage of iOS users is 65.2%, and the percentage of Android users is 34.8%.

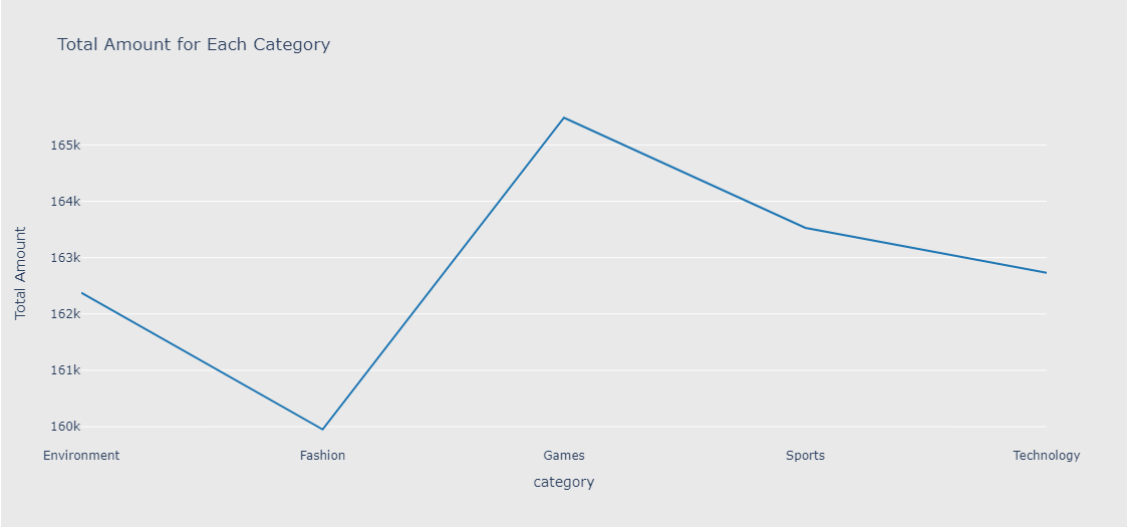


Figure 11: Line chart of total amount for each category.

Figure 11 illustrates that the highest donation amount goes for games projects with around 165,500 Euros, followed by sports and technology projects respectively. The lowest donation amount is for the fashion category.

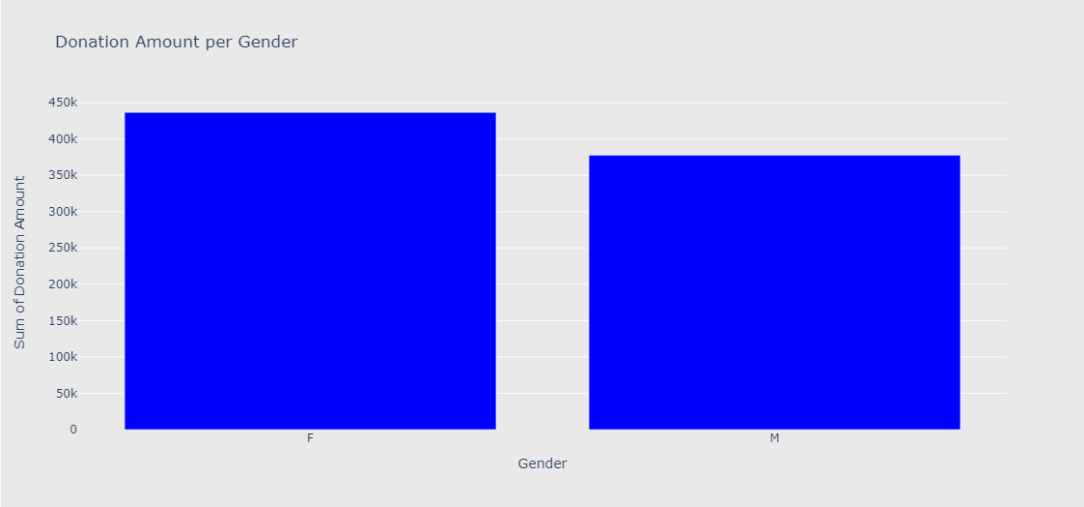


Figure 12:Bar chart of donation amount per gender.

Female users are the highest donation group compared to the male groups, with donation amount of 436,517 Euros as shown in Figure 12 above.

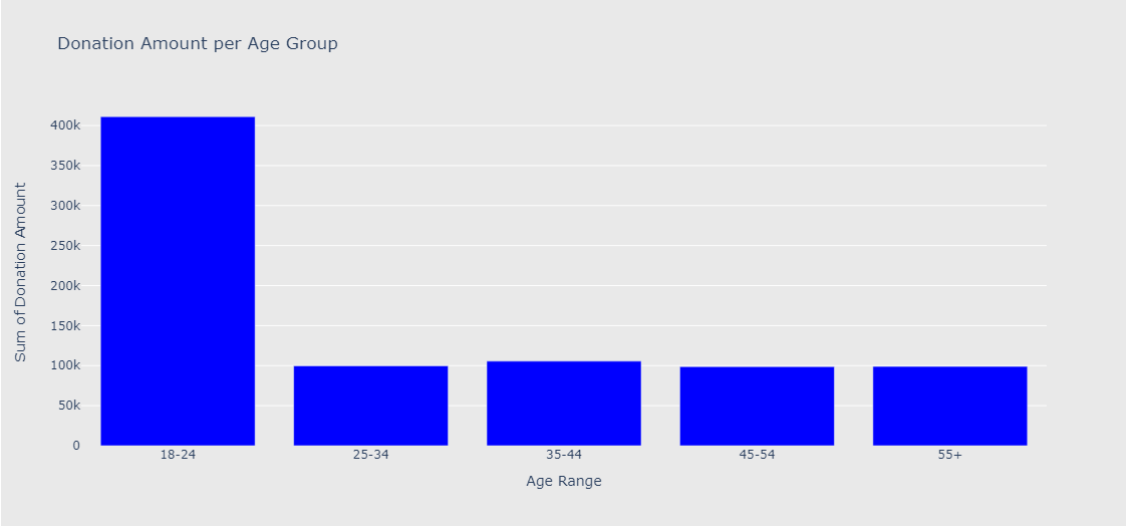


Figure 13: Bar chart of donation amount per age group.

Figure 13 shows that the highest age group is 18-24 with a total donation amount of 411,000 approximately. Followed by 35-44 group.

3.4 Data Modeling

In the data modeling phase, the K-means clustering algorithm was applied for unsupervised learning. Preprocessing involved Label Encoding for categorical columns using LabelEncoder() and standardization of the 'amount' column using StandardScaler() and fit_transform(). The optimal number of clusters (K) was determined using the Elbow Method. The 'kmeans_cluster' variable assigned each data point to a specific cluster. Evaluation metrics, including the Silhouette Score, were computed for clustering quality assessment. Descriptive statistics and visualizations, such as scatter plots and pair plots, were utilized to gain insights into each cluster's characteristics. This methodology provided a comprehensive exploration of K-means clustering results and their implications within the study context.

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import silhouette_score
from kmodes.kmodes import KModes
from sklearn.decomposition import PCA
```

Figure 14: imported libraries for the k-means model.

3.4.1 Elbow Method

Before employing the K-means algorithm, the optimal number of clusters (K) was determined by using the elbow method. Ensuring the stability of the algorithm's centroid requires careful consideration of the chosen K value. However, a drawback arises concerning the initialization of K points. To address this issue, the algorithm's performance is assessed across various centroid numbers. During evaluation, once convergence is achieved, the distance between each cluster's centroid and the data points is calculated. The sum of these distances serves as a performance metric. With an increasing number of cluster centroids, the objective function's magnitude decreases. Typically, the elbow method is employed to determine the optimal K value (Cui, M., 2020).

In this research the maximum number of (k) was assigned to ten clusters, specifying the range to be from one cluster to ten clusters, for each K value, a K-means clustering model is created using the K-Means class from scikit-learn. The model is fitted to the standardized and encoded dataset.

The inertia_ attribute of the K-means model is then calculated, representing the within-cluster sum of squares. This value indicates the total distance of data points within each cluster to their respective centroids. Figure 15 below shows that the optimal number of (K) is four clusters.

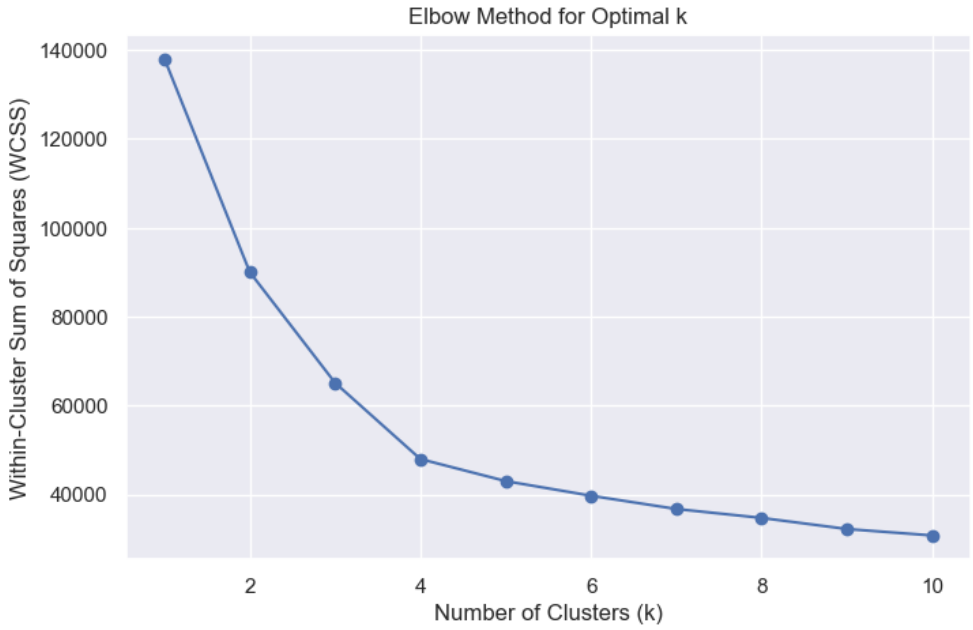


Figure 15: Elbow method to specify the optimal (k).

3.4.2 K-means

After initializing the optimal number of (K) to four clusters, the K-means model was employed. In preparation for the K-means clustering algorithm, the categorical features in the dataset, including 'category,' 'device,' 'gender,' and 'age,' were encoded into numerical representations using the LabelEncoder from scikit-learn. This transformation was necessary as K-means operates exclusively with numerical values. The encoded dataset, denoted as 'df_encoded,' was then used for clustering analysis to identify meaningful patterns within the data.

K-means aims to categorize data points into distinct groups or clusters, ensuring that members within the same cluster exhibit maximum similarity, while those in different clusters are maximally dissimilar. In this clustering method, the center of each cluster, known as the centroid, is identified, representing the mean of the values of observations assigned to that specific cluster (Lantz, 2019).

```
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
df['kmeans_cluster'] = kmeans.fit_predict(df_encoded)
```

Figure 16: Build the K-means model.

3.4.3 Model Evaluation

The evaluation metrics applied to quantify the clusters' quality is the silhouette score. Scikit-learn's silhouette score function calculates the average silhouette coefficient across all samples. The silhouette coefficient is derived by considering the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each data point. For an individual sample, the silhouette coefficient is computed as (b - a)/max(a, b).

A silhouette score close to +1 indicates that the data point is well-placed within its cluster, while a score near 0 suggests potential ambiguity in cluster assignment. Conversely, a silhouette score near -1 signifies that the data point might be misallocated to an incorrect cluster (Shahapure, K. R., & Nicholas, C., 2020).

```
silhouette_avg = silhouette_score(df_encoded, df_encoded['kmeans_cluster'])
print(f"Silhouette Score for K-Means: {silhouette_avg}")
```

Figure 17: Silhouette score code snippet.

RESULTS AND DISCUSSION

4.0 Results

This chapter presents a comprehensive analysis of the performance outcomes of the K-means clustering model applied to the dataset. The results section delves into the evaluation metrics, with a primary focus on the silhouette score, providing insights into the model's ability to identify distinct and meaningful clusters within the data, in addition to, applying hyperparameter tuning using GridSearchCV. Additionally, visual representations of the clustering results and any noteworthy observations will be discussed, contributing to a thorough understanding of the K-means model's effectiveness in uncovering underlying patterns in the dataset.

4.1 Performance Results

4.1.1 Silhouette Score

As mentioned previously, the silhouette score provides a measure of how well-defined and separated the identified clusters are within the dataset. For our K-means model, the obtained silhouette score was found to be 0.396. A silhouette score near +1 indicates that the data points are appropriately placed within their respective clusters, signifying a well-defined clustering structure. In this case, the silhouette score of 0.396 suggests a substantial degree of separation between clusters, affirming the efficacy of the K-means algorithm in grouping similar data points.

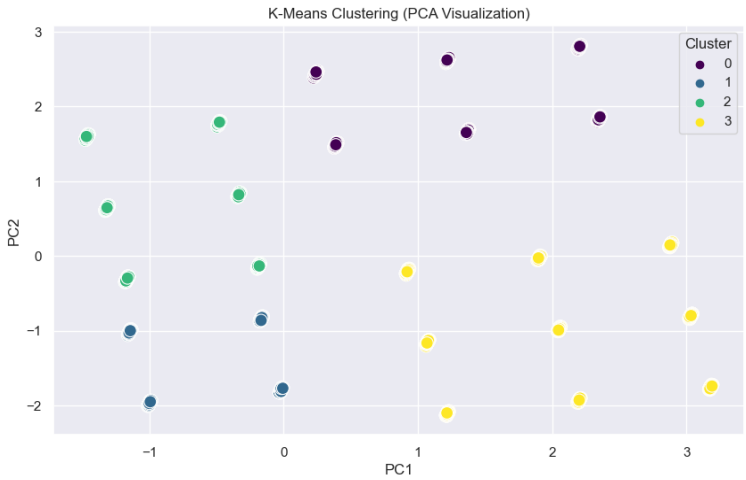


Figure 18: PCA scatter plot for cluster visualization.

Clusters visualized in the scatterplot (Figure 18) using PCA, Principal Component Analysis (PCA) is a statistical method employed in unsupervised learning for purposes such as dimensionality reduction, feature extraction, and data visualization. It works by examining patterns within the data, allowing for the reduction of dataset dimensions while retaining crucial information. The significance of PCA becomes apparent in scenarios where datasets possess a multitude of features, potentially leading to overfitting. Through its dimensionality reduction capabilities, PCA proves to be a valuable and efficient technique for describing and visualizing complex datasets (Abdulhafedh, A., 2021).

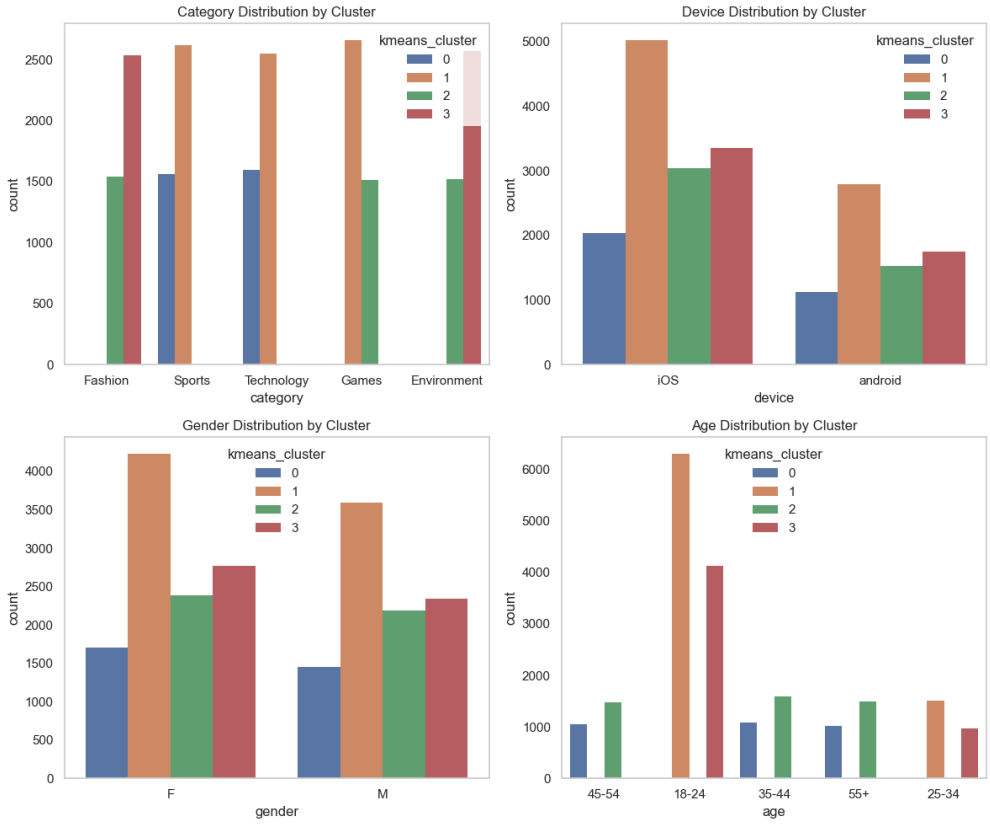


Figure 19: Category, device, gender, and age distributions by clusters.

Table 4: characteristics of the four clusters.

	Cluster	Category	Device	Gender	Age Range	Mean Amount	Median Amount	Min Amount	Max Amount	Std Amount	Count
0	0	Technology	iOS	F	35-44	39.112837	39.0	1.0	101.0	14.954463	3155

	Cluster	Category	Device	Gender	Age Range	Mean Amount	Median Amount	Min Amount	Max Amount	Std Amount	Count
1	1	Environment	iOS	F	18-24	39.616725	40.0	1.0	89.0	14.885202	5106
2	2	Games	iOS	F	18-24	39.412058	39.0	1.0	89.0	14.800680	7829
3	3	Fashion	iOS	F	35-44	39.367119	39.0	1.0	96.0	15.110029	4568

Figure 19 and Table 4 show that Cluster 0 predominantly consists of individuals in the age range of 35-44, using iOS devices, engaging in the Technology category, with a mean donation amount of $39.11. Cluster 1 is characterized by individuals aged 18-24, using iOS devices, inclined towards the Environment category, and exhibiting a mean donation amount of $39.62. Cluster 2 encompasses individuals aged 18-24, predominantly using iOS devices, engaging in the Games category, with a mean donation amount of $39.41. Lastly, Cluster 3 is distinguished by individuals aged 35-44, using iOS devices, showing an affinity for the Fashion category, and presenting a mean donation amount of $39.37, Notably, the majority of users in all four clusters are female. These findings shed light on diverse donor profiles, allowing for tailored strategies to enhance engagement and contribution within specific demographic segments.

The integration of clustering results with insights gained during the exploratory visualization phase establishes a robust and coherent narrative, reinforcing the validity of the findings. By drawing connections between the initial observations derived from graphical representations in both Power BI and Python and the detailed clusters uncovered by the K-means model, a comprehensive understanding of donor behaviors is provided. This approach not only validates earlier interpretations but also offers a deeper insight into the nuanced characteristics of each cluster. The alignment of initial insights with the clustering results enhances the credibility of the research, presenting a unified and holistic perspective on donor segmentation.

4.1.2. Gird Search

Grid search is a systematic hyperparameter tuning technique widely employed in machine learning to optimize the performance of algorithms. Specifically, when applied to the k-means clustering algorithm, grid search involves the exploration of various combinations of hyperparameter values to identify the most effective configuration for the task at hand. In the context of k-means, crucial parameters include the number of clusters (k), the initialization method, and the convergence criterion. By exhaustively searching through a predefined grid of possible values for these parameters, grid search aims to find the optimal set that maximizes the algorithm's efficacy in clustering the data. This method is instrumental in enhancing the robustness and accuracy of k-means clustering, ultimately leading to more meaningful and reliable results in various applications, including donor segmentation in the realm of fundraising and charitable contributions.

In order to increase the silhouette score of the K-means model grid search is employed to optimize parameters for the k-means model, employing the silhouette score as the evaluation metric. The pipeline integrates a standard scaler and a k-means model with various cluster configurations. Figure 20 below shows the output reveals that the optimal setup involves 2 clusters, resulting in a silhouette score of approximately 0.3333. Notably, this score is slightly lower than the original k-means model's silhouette score of 0.396. Despite the marginal decrease, the original k-means model is preferred for its superior performance. This comparison highlights the delicate balance between model complexity and clustering quality, emphasizing the relevance of the chosen parameters in influencing clustering outcomes.

```
Best Parameters: {'kmeans__n_clusters': 2}
Best Silhouette Score: 0.33333532365399887
```

Figure 20: Grid search results.

4.2 Discussion

In this section, the discussion delves into the implications of the obtained results. Additionally, the limitations inherent in the methodology are addressed, providing a transparent evaluation of the study's constraints.

4.2.1 Potential Implications

The findings of this research carry several noteworthy implications for the field of donor segmentation and fundraising strategies. Firstly, the identified clusters reveal distinct donor profiles based on age, device usage, and engagement categories. This nuanced understanding of donor characteristics provides valuable insights for organizations aiming to tailor their communication and outreach strategies. For instance, recognizing that Cluster 0, characterized by individuals aged 35-44 using iOS devices and engaging in the Technology category, represents a significant donor segment, allows organizations to customize campaigns and initiatives to better resonate with this specific group.

Secondly, the predominance of female users across all four clusters suggests a gender-based trend in donation behavior. This insight can guide organizations in designing gender-specific appeals and targeted campaigns. Additionally, the preference for iOS devices, as evident in the clustering results, underscores the importance of optimizing fundraising platforms and communication channels for iOS users. Moreover, the mean donation amounts associated with each cluster provide a quantitative basis for resource allocation and goal setting. Organizations can strategically allocate resources based on the identified clusters' donation patterns, directing efforts towards clusters with higher mean donation amounts.

This study holds particular relevance for nonprofit organizations, crowdfunding platforms, and marketing professionals. Nonprofit organizations can leverage these insights to enhance their donor segmentation strategies, thereby fostering more personalized and effective donor interactions. Crowdfunding platforms can optimize their user experience based on the identified preferences, creating a more engaging and donor-friendly environment. Marketing professionals can utilize the gender and device-based trends to tailor their promotional efforts, ensuring maximum impact.

4.2.2 Limitations

Despite the valuable insights generated, it is essential to acknowledge the limitations of this study. The dataset used may not capture the entirety of donor behavior, and external factors influencing donations were not considered. Additionally, the chosen clustering algorithm and parameters may

not be universally optimal for all datasets. Future research could explore alternative algorithms and include a broader set of features for a more comprehensive analysis. Furthermore, the interpretation of clusters is based on statistical patterns and may not capture nuanced motivations behind donor behavior. As with any machine learning model, the results should be viewed as tools for informed decision-making rather than definitive representations of donor characteristics. The study's scope is limited to the specific dataset, and potential biases or incomplete information in the data may impact the generalizability of the findings.

Furthermore, it is important to acknowledge that while the silhouette score serves as a valuable metric for assessing the efficacy of clustering, its potential improvement is constrained by the inherent limitations of the dataset which contained only five features. The quality of the silhouette score is contingent upon the presence of well-defined clusters in the data, and variations in donor behavior influenced by external, unobserved factors may impact the clustering accuracy. Therefore, the silhouette score attained in this study, while indicative of the model's performance, may not fully capture the intricacies of donor segmentation. Future researchers should consider refining the dataset or incorporating additional relevant features to enhance the silhouette score and provide a more comprehensive understanding of donor clusters.

RECOMMENDATIONS AND CONCLUSION

5.0 Recommendations & Conclusion

This chapter encapsulates recommendations for future research as well as a concluding summary that addresses the outlined objectives and research questions that are mentioned in the first chapter.

5.1 Recommendations

In directing future researchers, several recommendations emerge from the present study on donor segmentation within crowdfunding platforms.

Firstly, researchers are encouraged to leverage the insights gained here in the exploration of actual crowdfunding datasets, thereby validating, and extending the applicability of the findings to real-world scenarios. Furthermore, there exists an opportunity for refining clustering model performance by focusing on the enhancement of the silhouette score. This can be achieved through the exploration of alternative clustering algorithms, fine-tuning hyperparameters, and employing advanced techniques.

Additionally, expanding the dataset with additional variables, such as user engagement metrics and project-specific features, promises a more comprehensive understanding of donor behavior. Finally, recognizing the influence of external factors on donation patterns, future studies should explore and integrate these elements for a more nuanced analysis.

5.2. Conclusion

This research employed the K-means clustering algorithm and successfully identified four distinct donor segments within crowdfunding platforms. The findings revealed nuanced profiles based on age, device usage, project preferences, and donation amounts. The actionable insights derived from these segments offer a valuable foundation for crowdfunding platforms to tailor campaigns effectively, enhancing engagement within specific demographic clusters. The study's limitations, including dataset constraints and the inherent complexity of donor behavior, should be acknowledged. Nevertheless, the research contributes practical implications for nonprofit

organizations, crowdfunding platforms, and marketing professionals, guiding future efforts in personalized donor interactions and fundraising strategies. The concise integration of clustering results with actionable recommendations addresses provide a meaningful contribution to the field of donor segmentation and data-driven marketing analytics.

References

Abakouy, R., En-naimi, E. M., Haddadi, A. E., & Lotfi, E. (2019, October 2). Data-driven marketing. *Proceedings of the 4th International Conference on Smart City Applications*. https://doi.org/10.1145/3368756.3369024

Bapna, S. (2019, February). Complementarity of Signals in Early-Stage Equity Investment Decisions: Evidence from a Randomized Field Experiment. *Management Science*, *65*(2), 933–952. https://doi.org/10.1287/mnsc.2017.2833

Belleflamme, P., Lambert, T., & Schwienbacher, A. (2014, September). Crowdfunding: Tapping the right crowd. *Journal of Business Venturing*, *29*(5), 585–609. https://doi.org/10.1016/j.jbusvent.2013.07.003

Burtch, G., Ghose, A., & Wattal, S. (2014). Cultural Differences and Geography as Determinants of Online Prosocial Lending. *MIS Quarterly*, *38*(3), 773–794. https://doi.org/10.25300/misq/2014/38.3.07

Christy, A. J., Umamakeswari, A., Priyatharsini, L., & Neyaa, A. (2021, December 1). *RFM ranking – An effective approach to customer segmentation*. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2018.09.004

Crowdfunding Industry Statistics 2015 2016 - CrowdExpert.com. (2016, March 21). CrowdExpert.com. http://crowdexpert.com/crowdfunding-industry-statistics/

Cui, M. (2020). Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance, 1*(1), 5-8.

Ding L, Han B, Wang S, Li X, Song B (2019) User-centered recommendation using US-ELM based on dynamic graph model in ecommerce. Int J Mach Learn Cybern 10(4):693–703. https://doi.org/10. 1007/s13042-017-0751-z

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005, November). RFM and CLV: Using Iso-Value Curves for Customer Base Analysis. *Journal of Marketing Research*, *42*(4), 415–430. https://doi.org/10.1509/jmkr.2005.42.4.415

Griva A, Bardaki C, Pramatari K, Doukidis G (2021) Factors afecting customer analytics: evidence from three retail cases. Inf Syst Front. https://doi.org/10.1007/s10796-020-10098-1

Hörisch, J. (2015, November). Crowdfunding for environmental ventures: an empirical analysis of the influence of environmental orientation on the success of crowdfunding initiatives. *Journal of Cleaner Production*, *107*, 636–645. https://doi.org/10.1016/j.jclepro.2015.05.046

Hörisch, J. (2018). Think Big or Small is Beautiful. An empiricalanalysis of characteristics and determinants ofsuccess of sustainable crowdfunding projects. *International Journal of Entrepreneurial Venturing*, *10*(1), 1. https://doi.org/10.1504/ijev.2018.10008386

Hörisch, J. (2019, June). Take the money and run? Implementation and disclosure of environmentally-oriented crowdfunding projects. *Journal of Cleaner Production*, *223*, 127–135. https://doi.org/10.1016/j.jclepro.2019.03.100

Hossain, M., & Oparaocha, G. O. (2017, January 1). *Crowdfunding: Motives, Definitions, Typology and Ethical Challenges*. Entrepreneurship Research Journal. https://doi.org/10.1515/erj-2015-0045

Jáki, E., Csepy, G., & Kovács, N. (2022, September 21). *Conceptual framework of the crowdfunding success factors – Review of the academic literature*. Acta Oeconomica. https://doi.org/10.1556/032.2022.00028

Jovanović, T. (2019, February). Crowdfunding: What Do We Know So Far? *International Journal of Innovation and Technology Management*, *16*(01). https://doi.org/10.1142/s0219877019500093

Kumar, V. (2018, January). A Theory of Customer Valuation: Concepts, Metrics, Strategy, and Implementation. *Journal of Marketing*, *82*(1), 1–19. https://doi.org/10.1509/jm.17.0208

Lam, P. T., & Law, A. O. (2016, July). Crowdfunding for renewable and sustainable energy projects: An exploratory case study approach. *Renewable and Sustainable Energy Reviews*, *60*, 11–20. https://doi.org/10.1016/j.rser.2016.01.046

Lambert T, Schwienbacher A (2010) An empirical analysis of crowdfunding. Social Sci Res Netw 1578175:1–23

Lantz, B. (2019, April 15). *Machine Learning with R*. Packt Publishing Ltd. http://books.google.ie/books?id=iNuSDwAAQBAJ&printsec=frontcover&dq=Brett+Lantz.+2019.+Machine+Learning+with+R.+Packt+Publishing+Ltd.&hl=&cd=2&source=gbs_api

Lehner, O. M. (2014, May 27). The formation and interplay of social capital in crowdfunded social ventures. *Entrepreneurship & Regional Development*, *26*(5–6), 478–499. https://doi.org/10.1080/08985626.2014.922623

Lehner, O. M., & Kansikas, J. (2013, July). Pre-paradigmatic Status of Social Entrepreneurship Research: A Systematic Literature Review. *Journal of Social Entrepreneurship*, *4*(2), 198–219. https://doi.org/10.1080/19420676.2013.777360

Li, Y. M., Wu, J. D., Hsieh, C. Y., & Liou, J. H. (2020, February). A social fundraising mechanism for charity crowdfunding. *Decision Support Systems*, *129*, 113170. https://doi.org/10.1016/j.dss.2019.113170

Li, Y., Qi, J., Chu, X., & Mu, W. (2022, January 9). Customer Segmentation Using K-Means Clustering and the Hybrid Particle Swarm Optimization Algorithm. *The Computer Journal*, *66*(4), 941–962. https://doi.org/10.1093/comjnl/bxab206

Lim, C. S. L., & Wang, Z. (2023). A Systematic Approach to Segmentation Analysis Using Machine Learning for Donation-Based Crowdfunding. *Marketing and Smart Technologies*, 125–146. https://doi.org/10.1007/978-981-19-9099-1_10

Liu D-R, Lai C-H, Lee W-J (2009) A hybrid of sequential rules and collaborative fltering for product recommendation. Inf Sci 179(20):3505–3519. https://doi.org/10.1016/j.ins.2009.06.004

Mochkabadi, K., & Volkmann, C. K. (2018, August 4). Equity crowdfunding: a systematic review of the literature. *Small Business Economics*, *54*(1), 75–118. https://doi.org/10.1007/s11187-018-0081-x

Mollick, E. (2014). The dynamics of crowdfunding: An exploratory study. Journal of Business Venturing, 29(1), 1–16. https://doi.org/10.1016/j.jbusvent.2013.06.005

Sabuncu, B., Türkan, E., & Polat, H. (2020, April 25). *CUSTOMER SEGMENTATION AND PROFILING WITH RFM ANALYSIS*. Turkish Journal of Marketing. https://doi.org/10.30685/tujom.v5i1.84

Sembiring Brahmana, R. W., Mohammed, F. A., & Chairuang, K. (2020, April 30). Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *11*(1), 32. https://doi.org/10.24843/lkjiti.2020.v11.i01.p04

Shahapure, K. R., & Nicholas, C. (2020, October). Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)* (pp. 747-748). IEEE

Sundsøy, P., Bjelland, J., Iqbal, A. M., Pentland, A. S., & de Montjoye, Y. A. (2014). Big Data-Driven Marketing: How Machine Learning Outperforms Marketers' Gut-Feeling. Social Computing, Behavioral-Cultural Modeling and Prediction, 367–374. https://doi.org/10.1007/978-3-319-05579-4_45

Appendix A (Notebook)

Importing Required Libraries

In [2]:
```
# Install the kmodes library using pip
!pip install kmodes
```

```
Requirement already satisfied: kmodes in c:\users\sohaila\anaconda3\lib\site-packages (0.12.2)
Requirement already satisfied: numpy>=1.10.4 in c:\users\sohaila\anaconda3\lib\site-packages (from kmodes) (1.21.5)
Requirement already satisfied: scipy>=0.13.3 in c:\users\sohaila\anaconda3\lib\site-packages (from kmodes) (1.9.1)
Requirement already satisfied: joblib>=0.11 in c:\users\sohaila\anaconda3\lib\site-packages (from kmodes) (1.1.0)
Requirement already satisfied: scikit-learn>=0.22.0 in c:\users\sohaila\anaconda3\lib\site-packages (from kmodes) (1.0.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sohaila\anaconda3\lib\site-packages (from scikit-learn>=0.22.0->kmodes) (2.2.0)
```

In [35]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import plotly.express as px
import plotly.graph_objects as go


# Import KMeans from the sklearn libarary and additional modules
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline
```

Data Acquisition and Exploration

In [4]:
```
df = pd.read_csv("crowdfunding.csv")
df.head()
```

Out[4]:

	category	device	gender	age	amount
0	Fashion	iOS	F	45-54	61.0
1	Sports	android	M	18-24	31.0
2	Technology	android	M	18-24	39.0
3	Technology	iOS	M	18-24	36.0
4	Sports	android	M	18-24	40.0

In [5]:
```
df.shape
```

Out[5]:
```
(20658, 5)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20658 entries, 0 to 20657
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   category  20658 non-null  object 
 1   device    20658 non-null  object 
 2   gender    20658 non-null  object 
 3   age       20658 non-null  object 
 4   amount    20658 non-null  float64
dtypes: float64(1), object(4)
memory usage: 807.1+ KB
```

Data Preproccessing

In [7]:
```
# Checking for missing values
df.isnull().sum()
```

Out[7]:
```
category    0
device      0
gender      0
age         0
amount      0
dtype: int64
```

In [8]:
```
#checking whether the categorical variables have typos.

# Filter categorical variables
num_cols = df._get_numeric_data().columns
cols = df.columns
cat_cols = list(set(cols) - set(num_cols))
cat_cols

#Value counts
for col in cat_cols:
    if col in ['category','device','gender', 'age']:
        print(df[col].value_counts())
```

```
iOS        13459
android     7199
Name: device, dtype: int64
F    9682
M    9571
U    1485
Name: gender, dtype: int64
18-24    10439
35-44     2676
45-54     2532
55+       2515
25-34     2496
Name: age, dtype: int64
Sports          4179
Games           4173
Technology      4144
Environment     4889
Fashion         4873
Name: category, dtype: int64
```

In [9]:
```
# Replace 'U' (Unknown/Unspecified) with 'F' (Female) in the 'gender' column
df['gender'].replace('U', 'F', inplace=True)
```

In [10]:
```
#Updated Categorical Variable Exploration after Gender replacing
for cat_col in df.select_dtypes(include='object_'):
    print(cat_col, ':' , df[cat_col].unique(), '(' ,df[cat_col].nunique(), ')')
```

```
category : ['Fashion' 'Sports' 'Technology' 'Games' 'Environment'] ( 5 )
device : ['iOS' 'android'] ( 2 )
gender : ['F' 'M'] ( 2 )
age : ['45-54' '18-24' '35-44' '55+' '25-34'] ( 5 )
```

Exploratory Data Analysis (EDA) - Numerical Variables

In [11]:
```
df.describe()
```

Out[11]:

	amount
count	20658.000000
mean	39.407009
std	14.913658
min	1.000000
25%	29.000000
50%	39.000000
75%	50.000000
max	101.000000

Exploratory Data Analysis (EDA) - Categorical Variables

In [12]:
```
# Computing summary statistics for the categorical columns
df[['category', 'device','gender', 'age']].describe()
```

Out[12]:

	category	device	gender	age
count	20658	20658	20658	20658
unique	5	2	2	5
top	Sports	iOS	F	18-24
freq	4179	13459	11087	10439

In [13]:
```
# Iterating through each categorical column in the DataFrame
for cat_col in df.select_dtypes(include='object_'):
    # Print the column name, unique values, and the number of unique values
    print(cat_col, ':' , df[cat_col].unique(), '(' ,df[cat_col].nunique(), ')')
```

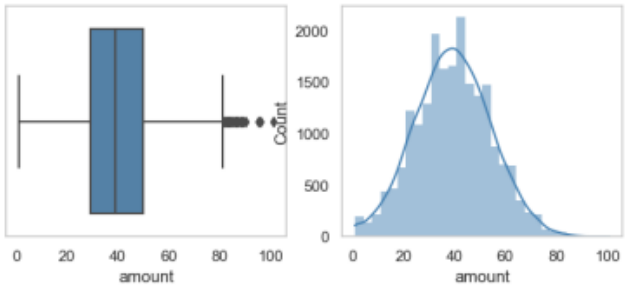
```
category : ['Fashion' 'Sports' 'Technology' 'Games' 'Environment'] ( 5 )
device : ['iOS' 'android'] ( 2 )
gender : ['F' 'M'] ( 2 )
age : ['45-54' '18-24' '35-44' '55+' '25-34'] ( 5 )
```

Univariate Distribution of Features

Univariate Distribution of Numerical Variables

In [21]:
```
#Creating a function to plot Box plot and Histogram
def hist_box_plot(df, feature, fig_num):
    sns.set(color_codes='Blue', style="whitegrid")
    sns.set_style("whitegrid", {'axes.grid': False})
    sns.set_context(rc={'patch.linewidth': 0.0})
    fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(8, 3))
    filtered = df.loc[~np.isnan(df[feature]), feature]
    sns.boxplot(x=filtered, ax=ax1, color='steelblue')  # boxplot
    sns.histplot(x=filtered, kde=True, color='steelblue', ax=ax2, bins=30)  # histogram
    plt.show()
```

In [22]:
```
fig_num = 1
for col in df.select_dtypes(include=[np.number]).columns:
    if col in ['amount']:
        hist_box_plot(df, col, fig_num)
        fig_num += 1
```



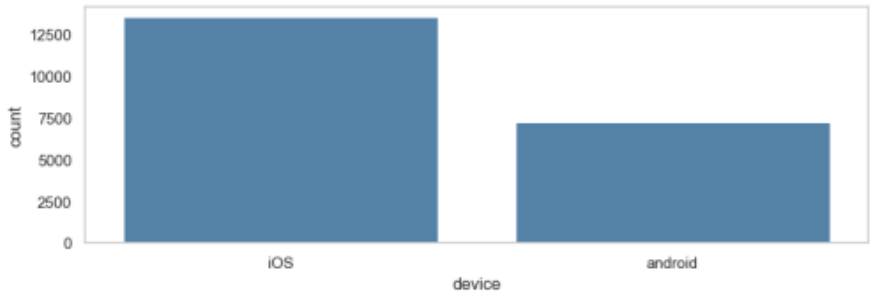
Univariate Distribution of Categorical Variables

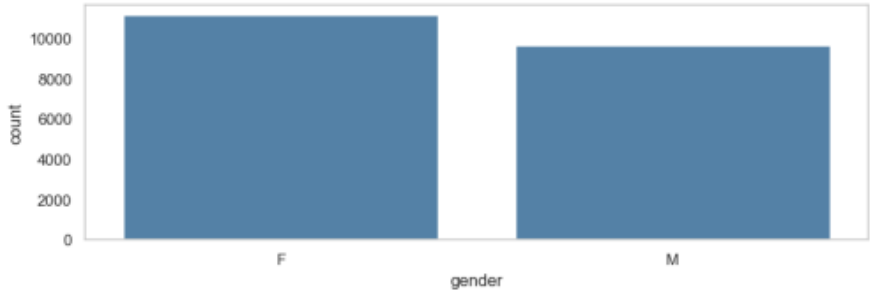
In [23]:
```
#Creating a function to plot Count plot
def count_plot(df,feature):
    sns.set(color_codes = 'Blue', style="whitegrid")
    sns.set_style("whitegrid", {'axes.grid' : False})
    sns.set_context(rc = {'patch.linewidth': 0.0})
    fig = plt.subplots(figsize=(10,3))
    sns.countplot(x=feature, data=df, color = 'steelblue') # countplot
    plt.show()
```

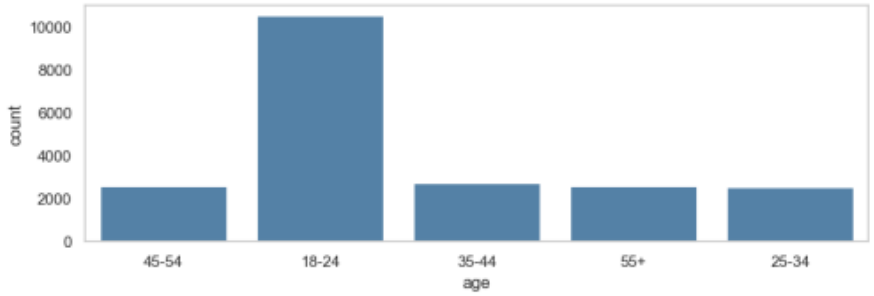
In [24]:
```
# Filter categorical variables
num_cols = df._get_numeric_data().columns
cols = df.columns
cat_cols = list(set(cols) - set(num_cols))
cat_cols
```

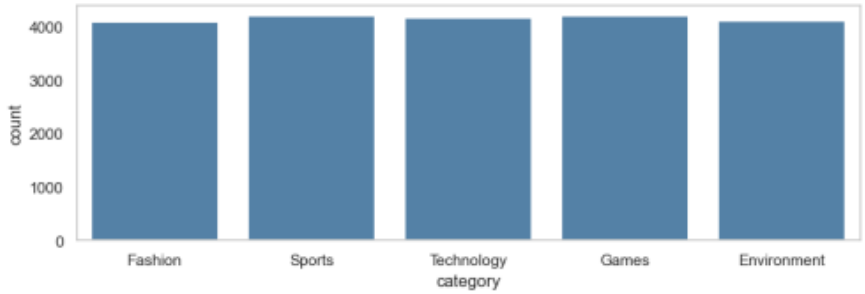
Out[24]:
```
['device', 'gender', 'age', 'category']
```

In [25]:
```
for col in cat_cols:
    if col in ['category','device','gender', 'age']:
        count_plot(df,col)
```





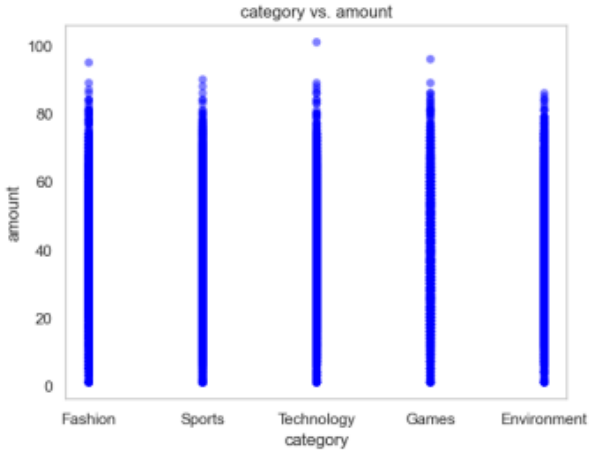




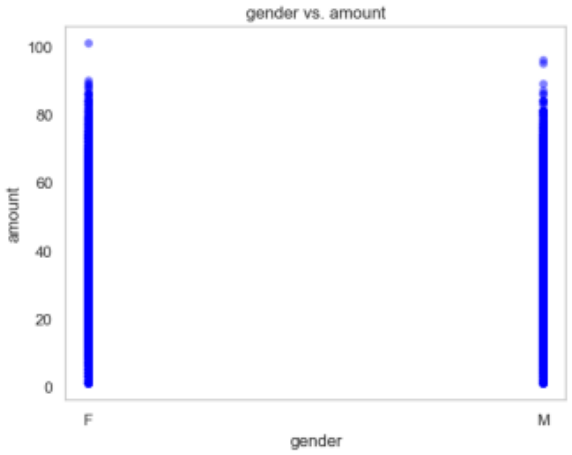
Bivariate Distribution of Features

In [26]:
```
def biplot(df, x_name, y_name):
    fig, ax = plt.subplots()
    ax.grid(False)
    x = df[x_name]
    y = df[y_name]
    plt.scatter(x,y,c='blue', edgecolors='none',alpha=0.5)
    plt.xlabel(x_name)
    plt.ylabel(y_name)
    plt.title('{x_name} vs. {y_name}'.format(x_name=x_name, y_name=y_name))
    plt.show()
```

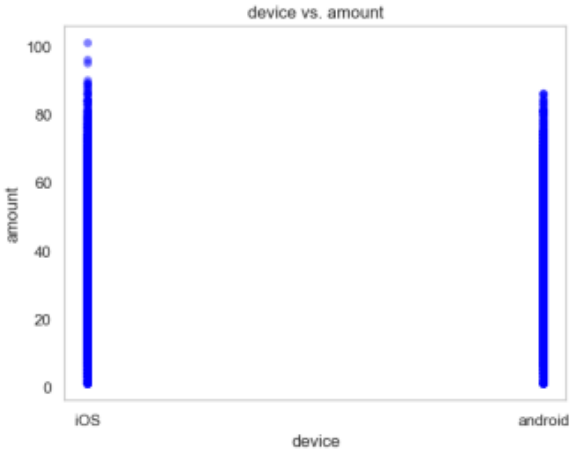
In [27]:
```
biplot(df=df,x_name='category',y_name='amount')
```



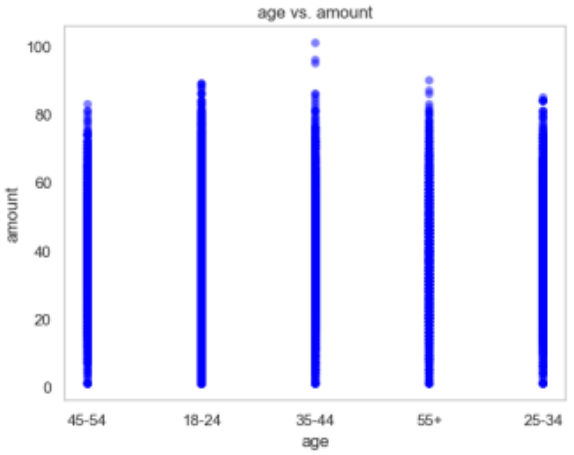
In [28]:
```
biplot(df=df,x_name='gender',y_name='amount')
```



In [29]:
```
biplot(df=df,x_name='device',y_name='amount')
```

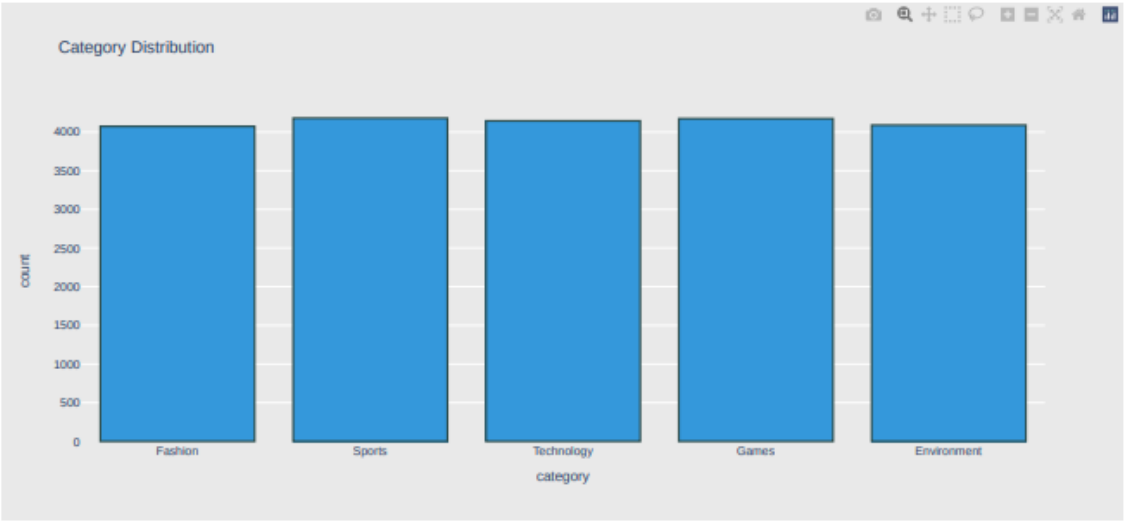


In [30]:
```
biplot(df=df,x_name='age',y_name='amount')
```

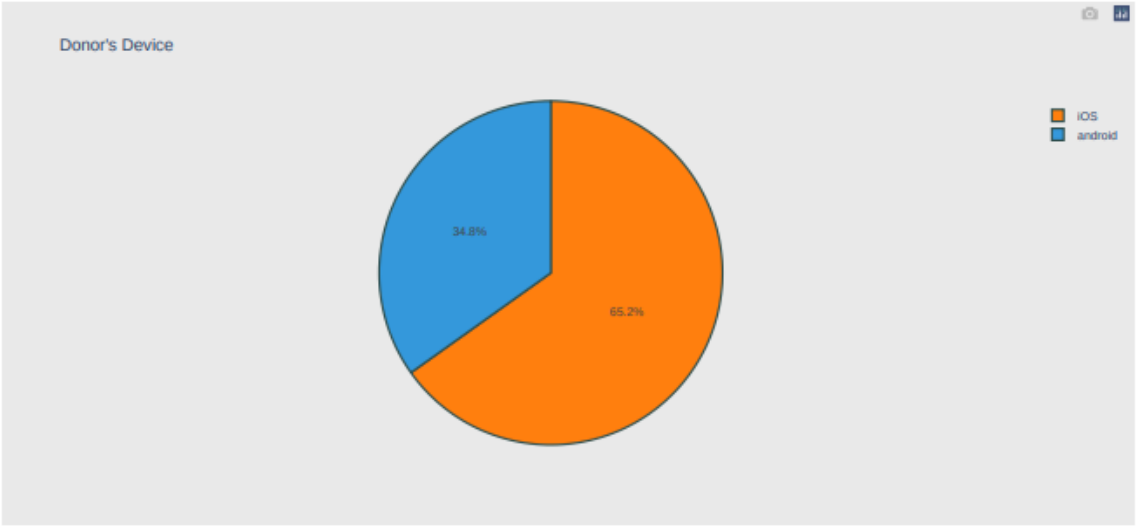


Exploratory Visualizations

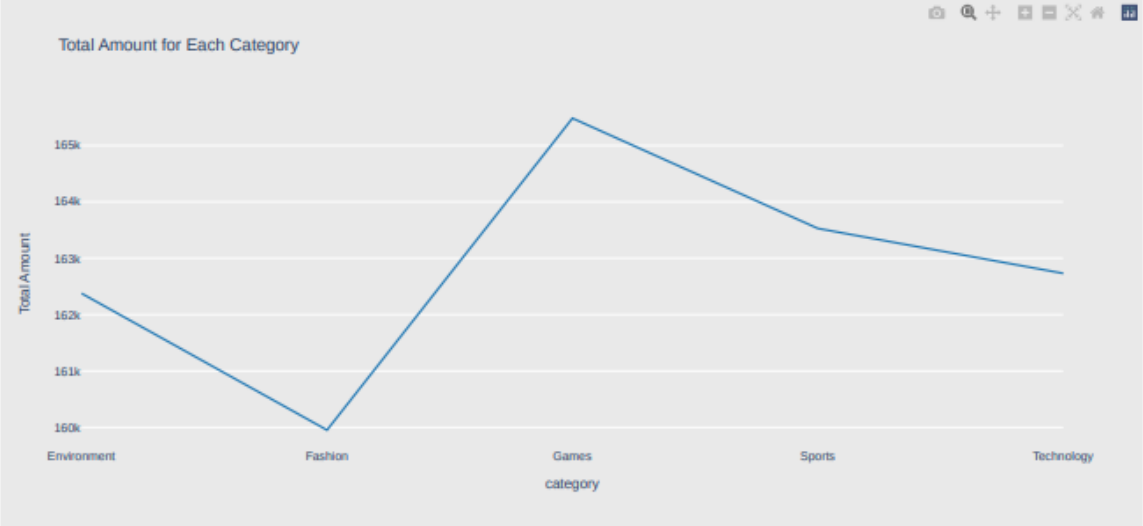
In [31]:
```
fig = px.histogram(df, x='category',
                   title='Category Distribution',
                   color_discrete_sequence=['#3498db'],  # Setting custom color
                   )
fig.update_traces(marker=dict(line=dict(width=2, color='DarkSlateGrey')))

# format the layout
fig.update_layout(
    xaxis=dict(showgrid=False, zeroline=False),
    yaxis=dict(zeroline=False, gridcolor='white'),
    paper_bgcolor='rgb(233,233,233)',
    plot_bgcolor='rgb(233,233,233)',
)
fig.show()
```



In [32]:
```
fig = px.pie(df, names='device',
             title='Donor\'s Device',
             color_discrete_sequence=['#ff7f0e', '#3498db'],# Setting custom color
            )

# format the layout
fig.update_layout(
    xaxis=dict(showgrid=False, zeroline=False),
    yaxis=dict(zeroline=False, gridcolor='white'),
    paper_bgcolor='rgb(233,233,233)',
    plot_bgcolor='rgb(233,233,233)',
)

fig.update_traces(marker=dict(line=dict(width=2, color='DarkSlateGrey')))


# Show the pie chart
fig.show()
category_amount = df.groupby('amount').sum().reset_index()
```

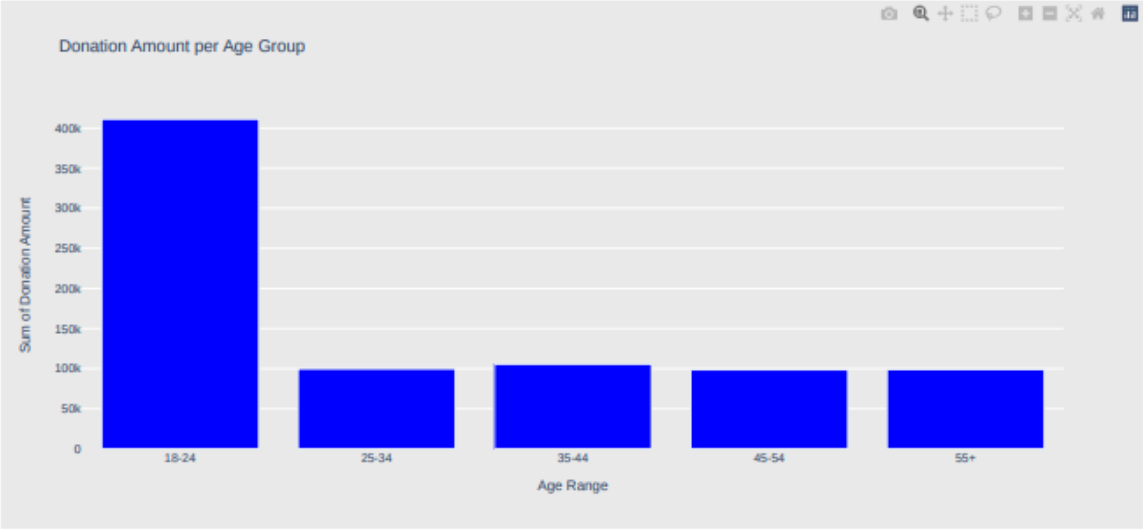


In [33]:
```
category_amount = df.groupby('category')['amount'].sum().reset_index()


fig = px.line(category_amount, x='category', y='amount',
              title='Total Amount for Each Category',
              labels={'amount': 'Total Amount'},
              color_discrete_sequence=['#1f77b4'])


# format the layout
fig.update_layout(
    xaxis=dict(showgrid=False, zeroline=False),
    yaxis=dict(zeroline=False, gridcolor='white'),
    paper_bgcolor='rgb(233,233,233)',
    plot_bgcolor='rgb(233,233,233)',
)

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



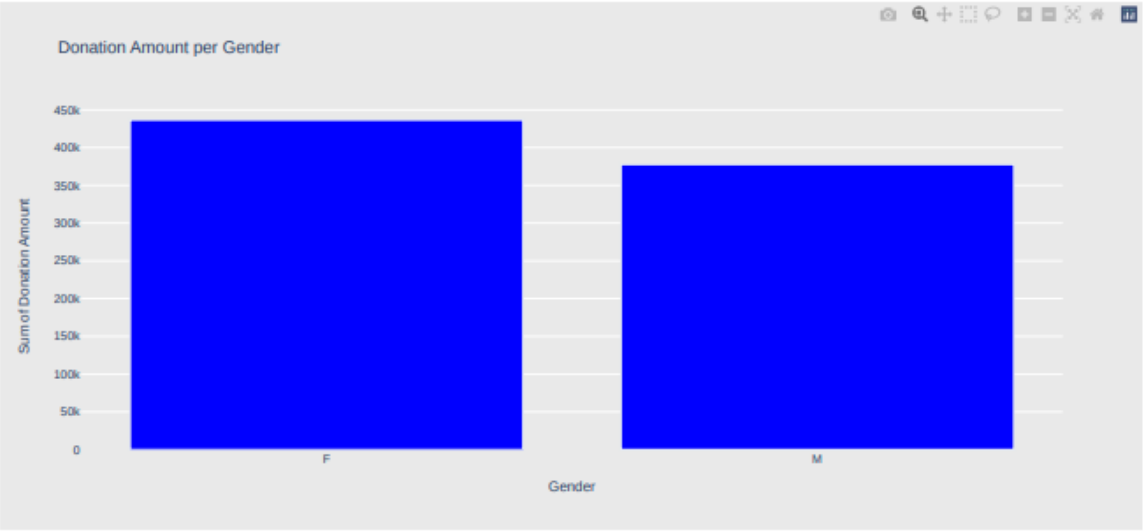
In [36]:

```
age_range = df["age"].value_counts().keys()
sum_amounts = df.groupby("age")["amount"].sum()

# Sort age_range for meaningful order on the x-axis
age_range = sorted(age_range)

# Create a bar plot using Plotly
fig = go.Figure(data=[go.Bar(x=age_range, y=sum_amounts[age_range], marker_color='blue')])

# Update layout
fig.update_layout(
    xaxis=dict(showgrid=False, zeroline=False),
    yaxis=dict(zeroline=False, gridcolor='white'),
    paper_bgcolor='rgb(233, 233, 233)',
    plot_bgcolor='rgb(233, 233, 233)',
    title="Donation Amount per Age Group",
    xaxis_title="Age Range",
    yaxis_title="Sum of Donation Amount"
)

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



In [37]:

```
gender = df["gender"].value_counts().keys()
sum_amounts = df.groupby("gender")["amount"].sum()

# Sort age_range for meaningful order on the x-axis
gender = sorted(gender)

# Create a bar plot using Plotly
fig = go.Figure(data=[go.Bar(x=gender, y=sum_amounts[gender], marker_color='blue')])

# Update layout
fig.update_layout(
    xaxis=dict(showgrid=False, zeroline=False),
    yaxis=dict(zeroline=False, gridcolor='white'),
    paper_bgcolor='rgb(233, 233, 233)',
    plot_bgcolor='rgb(233, 233, 233)',
    title="Donation Amount per Gender",
    xaxis_title="Gender",
    yaxis_title="Sum of Donation Amount"
)

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



K-means Model

In [38]:
```
# Encode categorical columns using Label Encoding
le = LabelEncoder()
df_encoded = df.copy()
categorical_columns = ['category', 'device', 'gender', 'age']
for col in categorical_columns:
    df_encoded[col] = le.fit_transform(df[col])

# Display the encoded dataset
df_encoded.head()
```

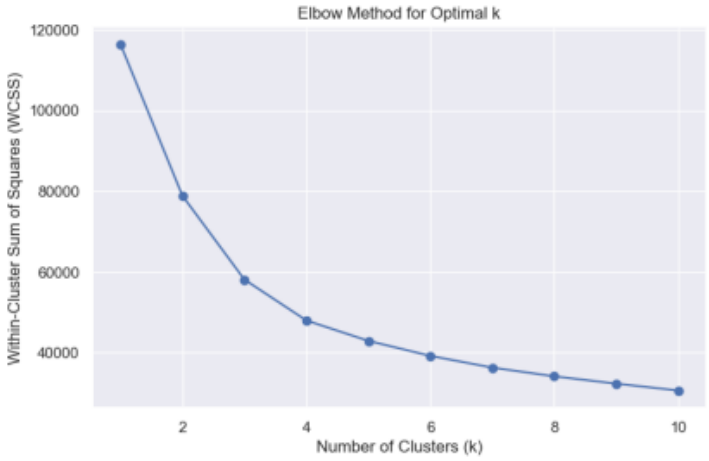
Out[38]:

	category	device	gender	age	amount
0	1	1	0	3	61.0
1	3	0	1	0	31.0
2	4	0	1	0	39.0
3	4	1	1	0	36.0
4	3	0	1	0	40.0

In [39]:
```
# Standardize the 'amount' column using StandardScaler
scaler = StandardScaler()
df_encoded['amount_scaled'] = scaler.fit_transform(df_encoded[['amount']])
```

In [40]:
```
# Drop the original 'amount' column
df_encoded = df_encoded.drop('amount', axis=1)
```

In [41]:
```
#Elbow method
wcss = []
max_k = 10

for k in range(1, max_k+1):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(df_encoded)
    wcss.append(kmeans.inertia_)

# Plot the Elbow Method graph
plt.figure(figsize=(8, 5))
plt.plot(range(1, max_k+1), wcss, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.show()
```



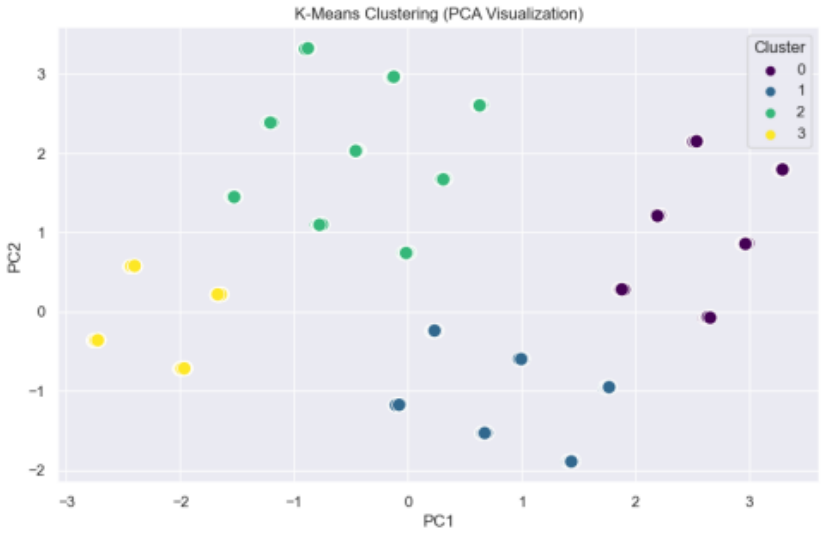
In [56]:
```
optimal_k = 4
```

In [71]:
```
# perform k-means model
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
df_encoded['kmeans_cluster'] = kmeans.fit_predict(df_encoded)
```

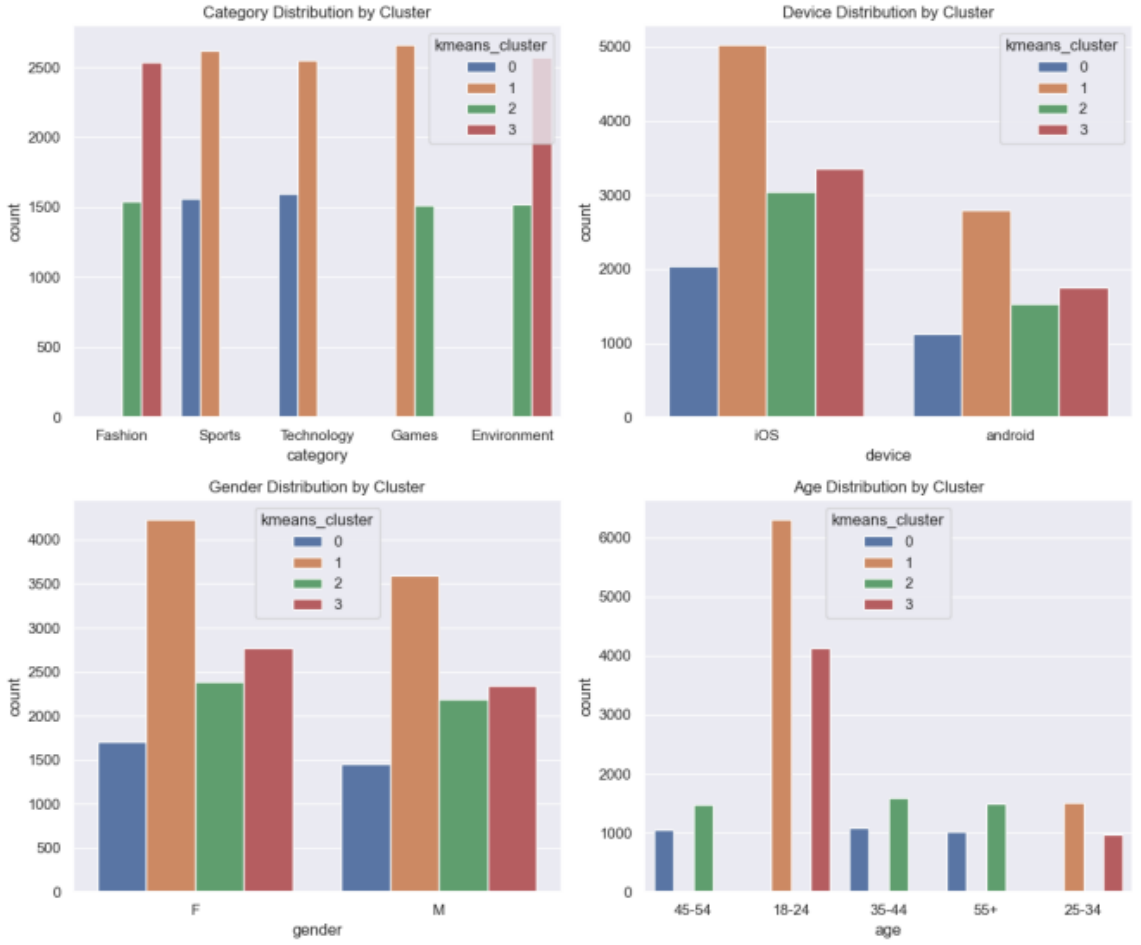
In [73]:
```
# Evaluate the model using Silhouette Score
silhouette_avg = silhouette_score(df_encoded, df_encoded['kmeans_cluster'])
print(f"Silhouette Score for K-Means: {silhouette_avg}")
```

```
Silhouette Score for K-Means: 0.39596976125543576
```

In [74]:
```
# Visualize the clusters using PCA for dimensionality reduction
pca = PCA(n_components=2)
pca_result = pca.fit_transform(df_encoded)

# Add the cluster labels to the PCA result
pca_df = pd.DataFrame(data=pca_result, columns=['PC1', 'PC2'])
pca_df['Cluster'] = df_encoded['kmeans_cluster']

# Plot the clusters using a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='PC1', y='PC2', hue='Cluster', data=pca_df, palette='viridis', s=100)
plt.title('K-Means Clustering (PCA Visualization)')
plt.show()
```



In [75]:
```
# Visualize distributions of each column by clusters
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

for i, column in enumerate(['category', 'device', 'gender', 'age']):
    sns.countplot(x=column, hue='kmeans_cluster', data=df, ax=axes[i//2, i%2])
    axes[i//2, i%2].set_title(f'{column.capitalize()} Distribution by Cluster')

plt.tight_layout()
plt.show()
```



In [76]:
```
# Calculate cluster statistics
cluster_stats = df.groupby('kmeans_cluster').agg({
    'category': lambda x: x.value_counts().index[0],
    'device': lambda x: x.value_counts().index[0],
    'gender': lambda x: x.value_counts().index[0],
    'age': lambda x: x.value_counts().index[0],
    'amount': ['mean', 'median', 'min', 'max', 'std', 'count']
}).reset_index()

# Display cluster statistics
cluster_stats.columns = ['Cluster', 'Category', 'Device', 'Gender', 'Age Range', 'Mean Amount', 'Median Amount', 'Min Amount', 'Max Amount', 'Std Am
cluster_stats.sort_values(by='Cluster')
```

Out[76]:

	Cluster	Category	Device	Gender	Age Range	Mean Amount	Median Amount	Min Amount	Max Amount	Std Amount	Count
0	0	Technology	iOS	F	35-44	39.112837	39.0	1.0	101.0	14.954463	3155
1	1	Games	iOS	F	18-24	39.412058	39.0	1.0	89.0	14.800680	7829
2	2	Fashion	iOS	F	35-44	39.367119	39.0	1.0	96.0	15.110029	4568
3	3	Environment	iOS	F	18-24	39.616725	40.0	1.0	89.0	14.885202	5106

In [77]:
```
# Cluster sampling
for cluster_id in df['kmeans_cluster'].unique():
    cluster_data = df[df['kmeans_cluster'] == cluster_id].sample(5, random_state=42)

    print(f"\nCluster {cluster_id} - Sample Data:")
    print(cluster_data[['category', 'device', 'gender', 'age', 'amount']])
```

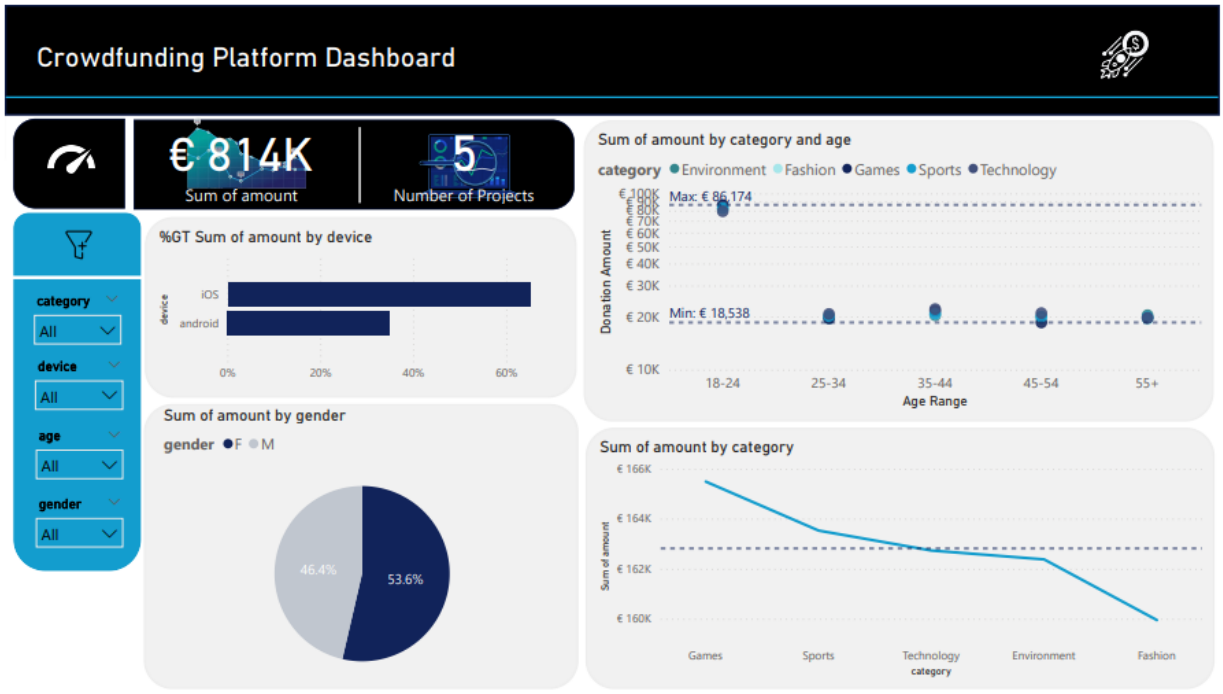
```
Cluster 2 - Sample Data:
          category device gender    age  amount
7148         Games    iOS      F  45-54    41.0
110          Games    iOS      M    55+    30.0
1224   Environment    iOS      F  45-54    48.0
12256        Games    iOS      F  35-44    44.0
18029  Environment    iOS      F    55+    34.0

Cluster 1 - Sample Data:
      category device gender    age  amount
3087     Games    iOS      M  25-34    25.0
12066   Sports    iOS      M  18-24    36.0
6791     Games    iOS      M  25-34    36.0
4451     Games    iOS      M  18-24    32.0
13090   Sports    iOS      M  18-24    50.0

Cluster 0 - Sample Data:
          category   device gender    age  amount
19972       Sports  android      F  35-44    63.0
17658   Technology      iOS      M  35-44    57.0
19406       Sports      iOS      F  35-44    45.0
20429       Sports      iOS      F    55+    65.0
960     Technology  android      F    55+    40.0

Cluster 3 - Sample Data:
           category   device gender    age  amount
16817       Fashion  android      F  18-24    28.0
2182        Fashion      iOS      M  18-24    62.0
2033   Environment  android      M  25-34    55.0
901    Environment      iOS      M  18-24    46.0
5298   Environment  android      F  25-34    34.0
```

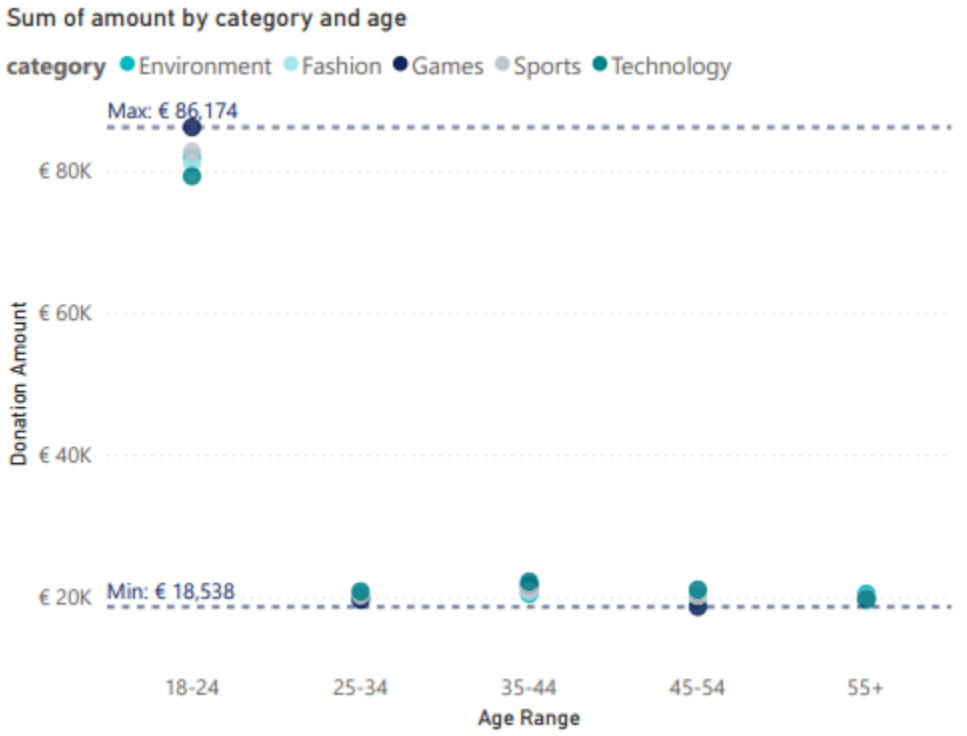
Hyperparameter Tuning - Grid Search

In [78]:
```
# Define the silhouette scorer function
def silhouette_scorer(estimator, X):
    labels = estimator.fit_predict(X)
    return silhouette_score(X, labels)

# Create a pipeline with a scaler and k-means
kmeans = KMeans(random_state=42, max_iter=15,n_init=50)
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('kmeans', kmeans)
])

# Define the parameter grid for GridSearchCV
param_grid = {
    'kmeans__n_clusters': [2, 3, 4, 5, 6]
}

# Create GridSearchCV with the silhouette scorer
grid = GridSearchCV(pipeline, param_grid, scoring=silhouette_scorer, cv=2)

# Fit the model
grid.fit(df_encoded)

# Get the best parameters and silhouette score
best_params = grid.best_params_
best_score = grid.best_score_

print("Best Parameters:", best_params)
print("Best Silhouette Score:", best_score)
```

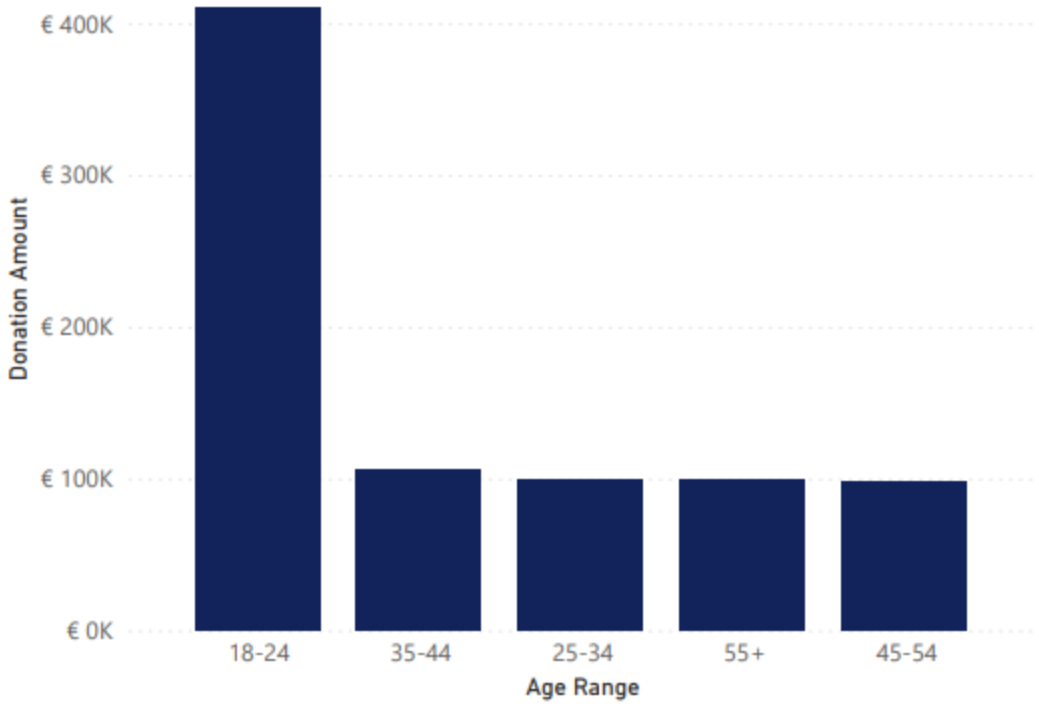
```
Best Parameters: {'kmeans__n_clusters': 2}
Best Silhouette Score: 0.333335323653971
```

Appendix B (Power BI Dashboard)

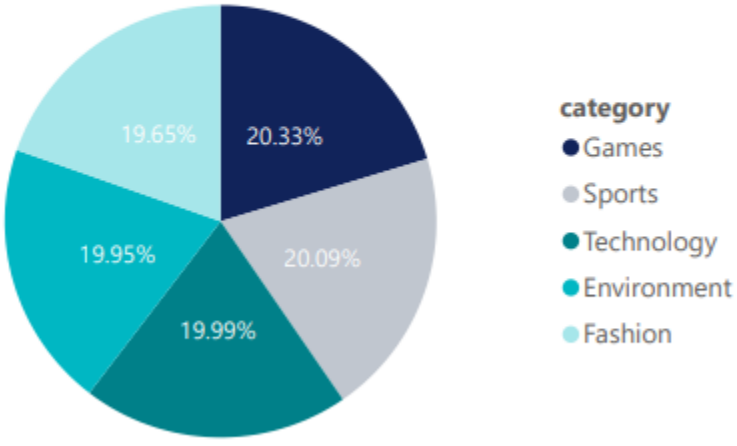


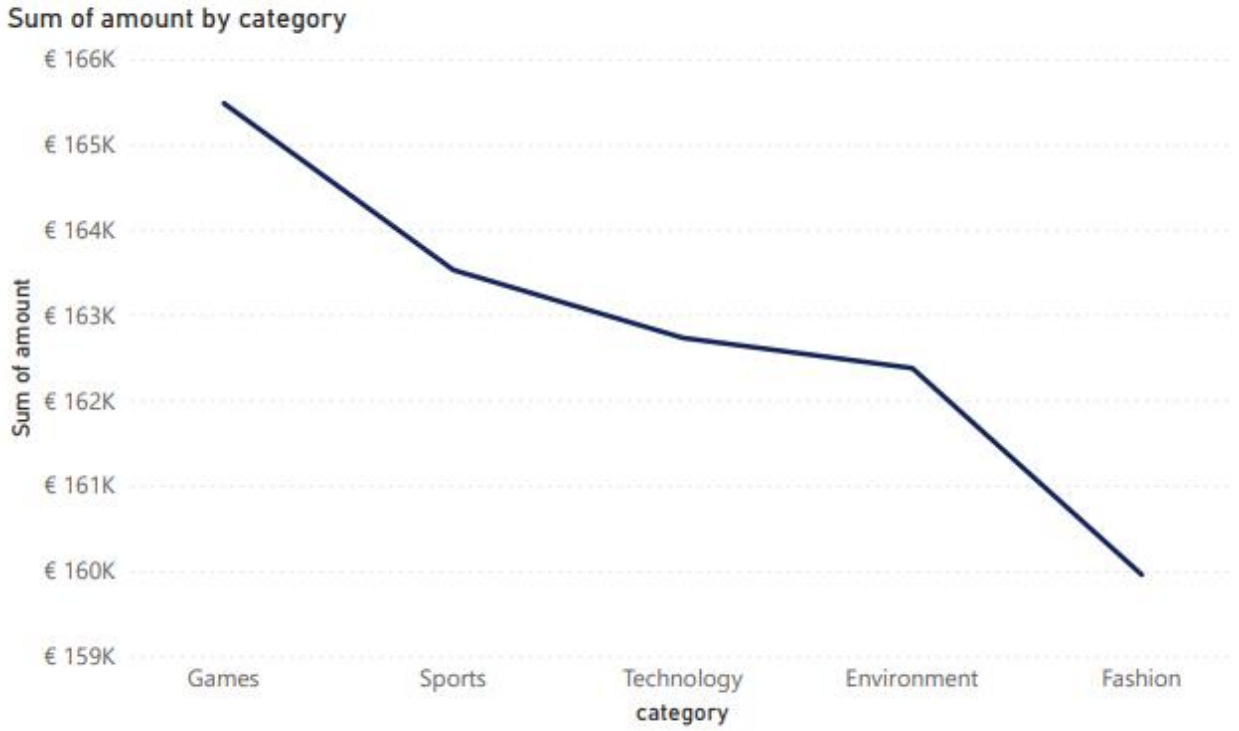


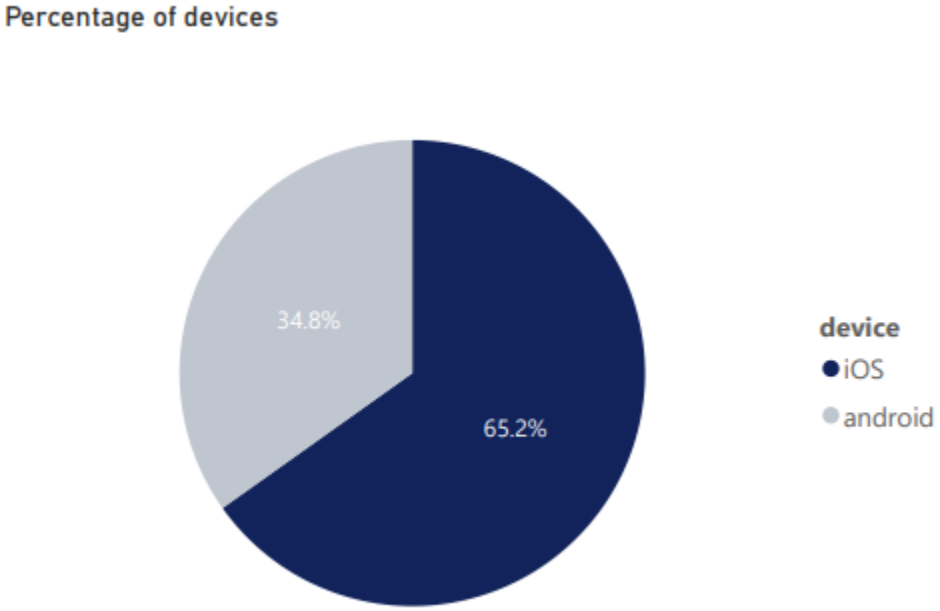
Sum of amount by age



Sum of amount by category




Sum of amount by category
€ 166K
€ 165K
€ 164K
€ 163K
€ 162K
€ 161K
€ 160K
€ 159K
Sum of amount
Games
Sports
Technology
Environment
Fashion
category


Percentage of devices
34.8%
65.2%
device
iOS
android