INTERNSHIP PROGRESS REPORT FOR THE TASK - (I)

Report on:

Paid Apps: Revenue vs. Number of Installs Visualisation Scatter Plot

SUBMITTED BY:

Shubham Dubey

Under Guidance of

Registration Mentor:

Manjur Sir

Training Mentor:

Elavarasi Mam

Internship Progress Report

Intern Name: Shubham Dubey
Position: Data Analytics Intern
Organization: NullClass
Report Date: 05-Oct-2024
Problem Number: 1
Task Title: Paid Apps: Revenue vs. Number of Installs Visualisation Scatter Plot

Report on Paid Apps: Revenue vs. Number of Installs Visualisation Scatter Plot.

Introduction

The market of mobile applications is rather saturated and various developers try to explore factors influencing profitability of such applications. The overall rationale behind paid apps is during purchase so they tend to directly sell the products, hence, install and revenue stand in a direct relation for paid apps. Here the author provides an evaluation of this relationship through a graphic of paid app download data depicted by a scatter plot; colored by the type of app. The use of a trendline brings out possibilities of affiliations or relations, and distribution patterns between two or more categories.

Background

Among all the mobile applications, paid applications are another important business model. In contrast to the free applications that operate on the principle of sales of their additional services and products, paid applications receive money from the sales of their applications. The analysis of the basic relationship between installs and revenue for paid apps are important for the developers, who are willing to enhance the efficiency of their plans for monetization. It is possible that there is a difference in trends in the area of revenue and installs across the various categories and, therefore, an examination of whether there are certain sorts of apps that are likely to have low install numbers but high levels of revenue or high install numbers but low levels of revenue would be necessary.

Using visual data analysis, this report analyzes these relationships with reference to paid apps in order to gain insights into the market.

Learning Objectives

1. **Correlation Analysis:**
   Analyze cross tabulation between installs and revenue for the paid apps to find out what kind of trends could exist.

2. **Data Visualization:**
   When presenting and analyzing this relationship, one should use scatter diagrams and different trends.

3. **Categorical Insights:**
   It is essential to divide applications by the type in order to find out which categories are most effective in such criteria as install-to-revenue.

4. **Decision-Making:**
   Provide recommendations to app developers and interested parties about potential areas that they could devote their efforts in order to increase profitability of the app in question.

Activities and Tasks:

1. **Data Collection and Cleaning:**
   - Gather data on installs, revenue, and app categories from reliable sources.
   - Filter out free apps, concentrating solely on paid apps for data relevance.

2. **Scatter Plot Creation:**
   - Generate a scatter plot to visualize the relationship between installs (X-axis) and revenue (Y-axis) for paid apps.
   - Color-code data points by app category for enhanced comparison between different app types.

3. **Trendline Addition:**
   - Apply a trendline to emphasize the general correlation between installs and revenue.
   - Analyze different slopes across categories to identify outliers and high-performing segments.

4. **Categorical Analysis:**
   - Examine the scatter plot to observe performance disparities between categories like "Games," "Productivity," and "Lifestyle."
   - Investigate whether specific categories demonstrate a higher revenue potential relative to their install base.

Skills and Competencies:

1. **Data Analysis:**
   Applying data analysis skills in preprocessing and cleaning of the dataset with the help of the Python.

2. **Data Visualization:**
   Gain sensing of creating scatter plots and further improving the scatter plot using categorical color-coding and trendline using Matplotlib and obtaining the same through Seaborn.

3. **Statistical Analysis:**
   Use statistics in order to analyze correlation in elements of a set, slope and all kinds of patterns in analysis set.

4. **Problem-Solving:**
   Illustrate analysis and evaluation: address the data-related problems and modifying visualization characteristics for better understanding.

Feedback and Evidence:

1. **Peer and Expert Feedback:**

   - Share with other students and supervisors the created scatterplot and the related discussion.
   - Look for feedback that will focus on the colors used to categorise the apps, and how well the trendline is capable of representing correlation.eedback emphasizing the effectiveness of color-coding app categories and the clarity of the trendline in representing the correlation.

2. **Visual Evidence:**

   - Present the following; a scatter plot with color coded categories and a trendline as part of the analysis.
   - Ensure documentation of the analysis process that is taken in order to show a clear documentation of the process taken in data analysis and visualization.the steps taken during the analysis to demonstrate transparency in data processing and visualization.

Challenges and Solutions:

1. **Data Filtering:**

   - Challenge: This means that paid application has a small importance in the first data sets that may include the free applications.
   - Solution: Continue the query so that all the free applications are eliminated from the sample; it is in this way safer that the examination just embraces paid applications**.**

2. **Trendline Interpretation:**

   - Challenge: The given overall trendline may not give information in case of specific categories.
   - Solution: Extend the trendlines adding more specific categories to make the vision of each kind of application

3. **Sparse Data for Certain Categories:**

   - Challenge: There may a situation, in which certain categories may contain very few numbers and it may be difficult to come up with results.
   - Solution: Gather like categories together and concentrate the analysis around more solid data sets that would increase reliability of results.

Outcomes and Impact:

1. **Positive Correlation:**
   - Quantify the relationship between installs and revenue for most categories and find that the strength of this relationship differs by app type..

2. **Category Performance:**

   - Recognise list items such as “Productivity” and “Games” with a more inclined trendline as the represent less number of apps installed but higher revenue.o Acknowledging categories such as ‘Lifestyle’ that has relatively low slopes indicating that more installations are required to generate a similar amount of revenues.installs.

   - Recognize categories like "Lifestyle" with flatter slopes, suggesting a need for more installs to achieve comparable revenue.

3. **Practical Insights:**

   - Provide recommendations to the app developers targeting categories with high install to revenue ratios to increase user acquisition by keeping it simple when monetizing.

   - Suggest developers who aim at categories having higher RVI to increase additional monetization means, for instance, purchases inside the application or monthly fees.s.

   - Advise developers targeting categories requiring more installs to consider additional monetization strategies like in-app purchases or subscription models.

Code implementation photos:

jupyter Untitled Last Checkpoint: 9 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[3]:
```
#Task - 1
```

[4]:
```
print('Create a scatter plot to visualize the relationship between revenue and the number of installs for paid apps only. Add a trendline to show the cor
```

```
Create a scatter plot to visualize the relationship between revenue and the number of installs for paid apps only. Add a trendline to show the correlatio
n and color-code the points based on app categories.
```

[12]:
```
# Step 0: Install necessary libraries (if not installed)
pip install pandas matplotlib seaborn
```

```
Requirement already satisfied: pandas in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (2.2.2)
Requirement already satisfied: matplotlib in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (0.13.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.1.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.9.0.pos
t0)
Requirement already satisfied: pytz>=2020.1 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (4.54.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (3.1.4)
Requirement already satisfied: six>=1.5 in c:\users\nice\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas)
(1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[ ]:
```
# Step 1: Import necessary libraries
import pandas as pd  # For data manipulation and analysis
import seaborn as sns  # For data visualization
import matplotlib.pyplot as plt  # For customizing and displaying plots
```

[25]:
```
# Step 2: Load the dataset
df = pd.read_csv('data.csv')
# check the data are loaded or not
df.head()
```

[25]:

	App	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Installs	Price	Revenue	Category	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13	Ur
0	Photo Editor & Candy Camera & Grid & ScrapBook	NaN	NaN	NaN	NaN	10000.0	10.0	0.0	ART_AND_DESIGN	NaN	NaN	NaN	NaN	NaN	
1	Coloring book moana	NaN	NaN	NaN	NaN	500000.0	11.0	1.0	ART_AND_DESIGN	NaN	NaN	NaN	NaN	NaN	
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	NaN	NaN	NaN	NaN	5000000.0	12.0	2.0	ART_AND_DESIGN	NaN	NaN	NaN	NaN	NaN	
3	Sketch - Draw & Paint	NaN	NaN	NaN	NaN	50000000.0	13.0	3.0	ART_AND_DESIGN	NaN	NaN	NaN	NaN	NaN	
4	Pixel Draw - Number Art Coloring Book	NaN	NaN	NaN	NaN	100000.0	14.0	4.0	ART_AND_DESIGN	NaN	NaN	NaN	NaN	NaN	

[16]:
```
# Step 3: Convert the 'Price' column to numeric, non-numeric values will become NaN
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
```

[17]:
```
# Step 4: Filter the dataset for paid apps only (Price > 0)
paid_apps = df[df['Price'] > 0]
```

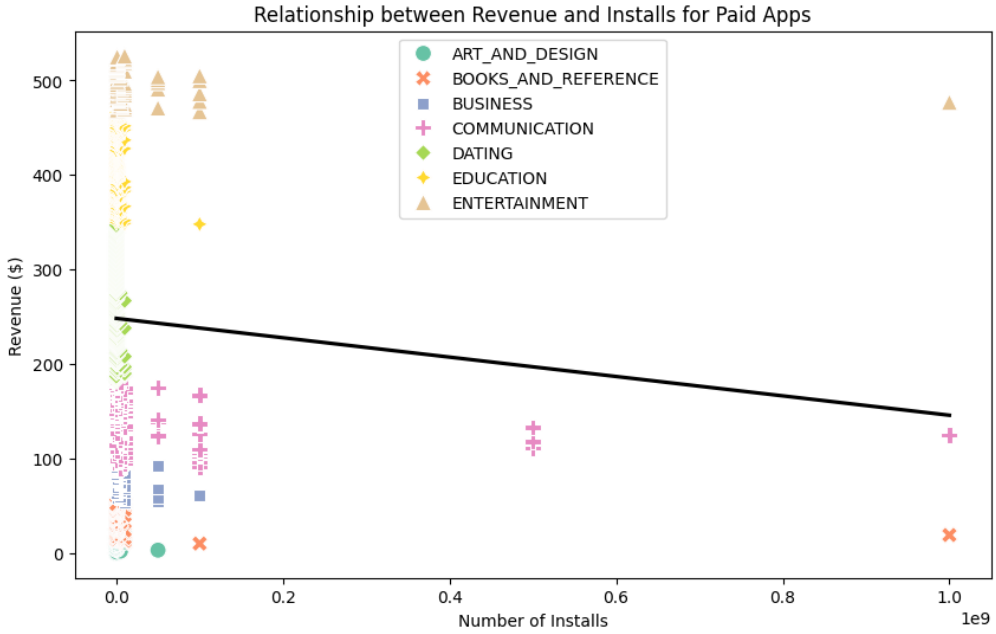
[18]:
```
# Step 5: Select relevant columns for visualization: 'Revenue', 'Installs', and 'Category'
paid_apps = paid_apps[['Revenue', 'Installs', 'Category']]
```

[24]:
```
# Step 6: Create the scatter plot, color-coded by app category
# This scatter plot shows the relationship between the number of installs and revenue, with different colors for different app categories
plt.figure(figsize=(10, 6))  # Set the figure size for the plot
scatter_plot = sns.scatterplot(
    data=paid_apps,
    x='Installs',  # X-axis: Number of installs
    y='Revenue',  # Y-axis: Revenue generated
    hue='Category',  # Points are colored by app category
    palette='Set2',  # Use the 'Set2' color palette for visual appeal
    style='Category',  # Optional: different marker styles for each category
    s=100  # Size of the scatter plot points
)

# Step 7: Add a trendline (regression line)
# This trendline shows the overall relationship between installs and revenue
sns.regplot(
    data=paid_apps,
    x='Installs',
    y='Revenue',
    scatter=False,  # Don't add scatter points here (they are already in the scatterplot)
    color='black',  # Trendline color
    line_kws={'label': 'Trendline'},  # Label the trendline for clarity
    ci=None  # Disable the confidence interval around the trendline for simplicity
)

# Step 8: Add labels and title
# These labels help describe the axes and the plot's purpose
plt.xlabel('Number of Installs')  # X-axis label
plt.ylabel('Revenue ($)')  # Y-axis label
plt.title('Relationship between Revenue and Installs for Paid Apps')  # Plot title

# Step 9: Show the legend and display the plot
plt.legend()  # Show the legend for the category and trendline
plt.show()  # Display the final plot
```



[ ]:

Conclusion:

This report presents an analysis of the correlation that exists between the installs and the revenue generated by paid apps. Using the scatter plot to analyze data and grouping applications by type, we have determined patterns that may be helpful for developers in terms of choosing the right model of monetization. The inclusion of a trendline has provided deeper insights that show that while most of the app categories receive increased installs, categories like “Games” and “Productivity” are examples of more app categories that get better revenues when fewer users install the apps. The findings from this study can help those who develop apps to better structure their future development and marketing efforts to increase profit margins.

Thank You