Introduction:

To identify those who are likely to spend money in their game after finishing the tutorial, I decided to develop a 'profile' of a likely spender. This profile gives insight and shows common characteristics of a potential spender. To determine this profile, I will using clustering in PCA, XGboost and Visualization.

Approach:

Data_investigation.ipynb

First data is preprocessed. **Load_and_Combine** joins data from three separate sources on the user_id. Since the number of rows across all files is equal, the type of join is not relevant. We then filter by those who completed the tutorial. This is the population of interest. Next is data investigation.

Filter_Data checks for and counts nulls. If the count is negligible in relation to data size then rows with nulls are removed from the dataset. In **Split_Col_Types,** we also check for all datatypes within the dataframe. Objects, floats and integer types are grouped. This breakdown is important because the numerical type is used for PCA while the objects will be "one-hot-encoded" for XGboost. Recall that One-hot-encoding creates additional columns, so those with large amounts of unique values drastically increase the dataframe dimension. In **Show_Unique_Values** and **Count_Unique**, I investigate how many distinct values are present. Due to time constraints, it is easier to remove these columns. Another feasible solution will be to group said distinct values. For example, all those whose text contain Intel are grouped as Intel. Regex and the nlp module nltk will accomplish the groupings.

ML_PCA.ipynb

The purpose of PCA here is to identify which variables capture the most similar information to the numerical response variable **Total_spend**. PCA is more than just a dimension reduction algorithm; by using the loading score we can cluster the variables in relation to the two largest dimensions (PC1, PC2). Plotting the loading scores in these two dimensions, we can see which variables are most similar to Total_spend. Granted that this only captures linear relationships and uses numerical variables. We will make use of a more robust algorithm to validate this.

ML.ipynb

Decision tree algorithms are extremely useful for tabular data; they are very robust. Of the various ensemble tree based algorithms, I believe that XGboost works best. A great boosting algorithm that handles complex intricacies of large dataset. We can also use the variable importance feature of the model.

Although grid search is a plausible method for tuning the model, I used trial and error to find suitable hyper-parameters due to its minimal memory cost. The important variables found here and in PCA are then further investigated using visualization.

Data_Profile.ipynb

Using conditional plots and the most useful variables from the ML models, a profile was determined.
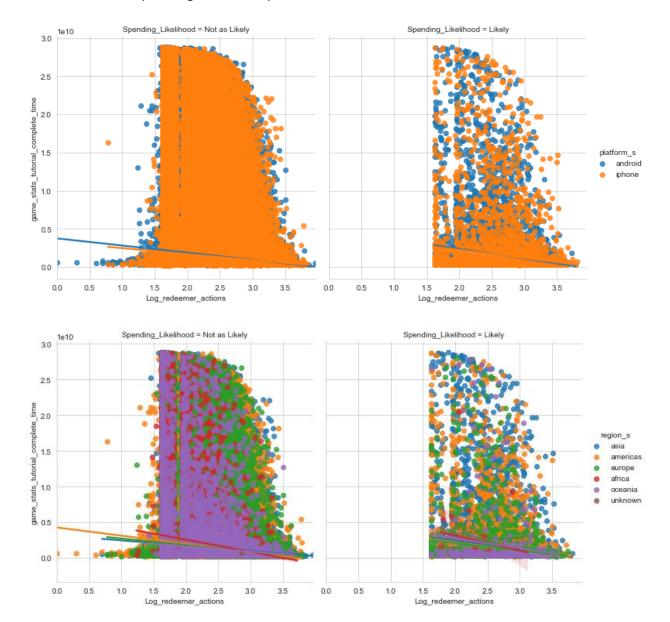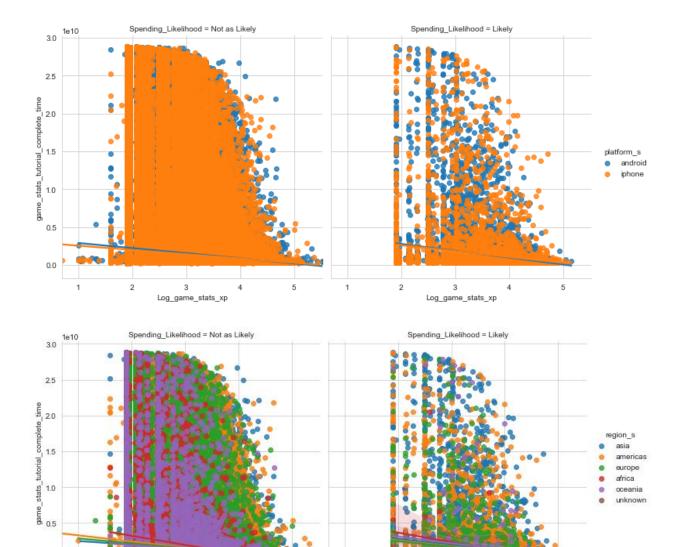
A User likely to spend will:
- Have redeemer actions above 31
    - Extremely true for those in Oceania region
- 11.7% faster at completing the tutorial upon install than those who are not likely (Median value)
    - Specifically from Oceania Region a median complete time of under 527 seconds
- Complete the tutorial under a median value of 571 seconds.
    - They are 20% faster if they use an IPhone
- Have at least 80 experience points accumulated by first time interval from install of
- Device memory of at least 794 for both apple and android
    - Android users have 8% more memory than iPhone so longer tutorials may be pushed for them
- Have at least 31.6 experience points accumulated by second time interval from install
- Have Over 115 scribe actions taken since install

Summary

Prices can be increased for users who fit these criteria, as they are more likely to spend. Key metrics are time taken to finish tutorials, memory available on device, amount of experience points accumulated within several period after install, platform, region, redeemer and scribe actions.

Attached below are outputs to generate the profile

Spending_Likelihood = Not as Likely   **Spending_Likelihood = Likely**

platform_s
- android
- iphone

Spending_Likelihood = Not as Likely   **Spending_Likelihood = Likely**

region_s
- asia
- americas
- europe
- africa
- oceania
- unknown

Spending_Likelihood = Not as Likely

Spending_Likelihood = Likely

platform_s
● android
● iphone

Spending_Likelihood = Not as Likely

Spending_Likelihood = Likely

region_s
● asia
● americas
● europe
● africa
● oceania
● unknown

Spending_Likelihood = Not as Likely — Spending_Likelihood = Likely

region_s
- asia
- americas
- europe
- africa
- oceania
- unknown

Spending_Likelihood = Not as Likely — Spending_Likelihood = Likely

platform_s
- android
- iphone