DST PROJECT

VIDEO GAME PLAYTIME ANALYSIS

Pradyumn Bagry - 2020UCD2113
Ayush Kumar - 2020UCD2104

MENTORS - Dr.Geetanjali, Ms.Faraj Chisti

PROBLEM STATEMENT

Analyzing the trends observed in video game playtime over the years, preference for single-player or multiplayer games, and what genres are becoming popular.

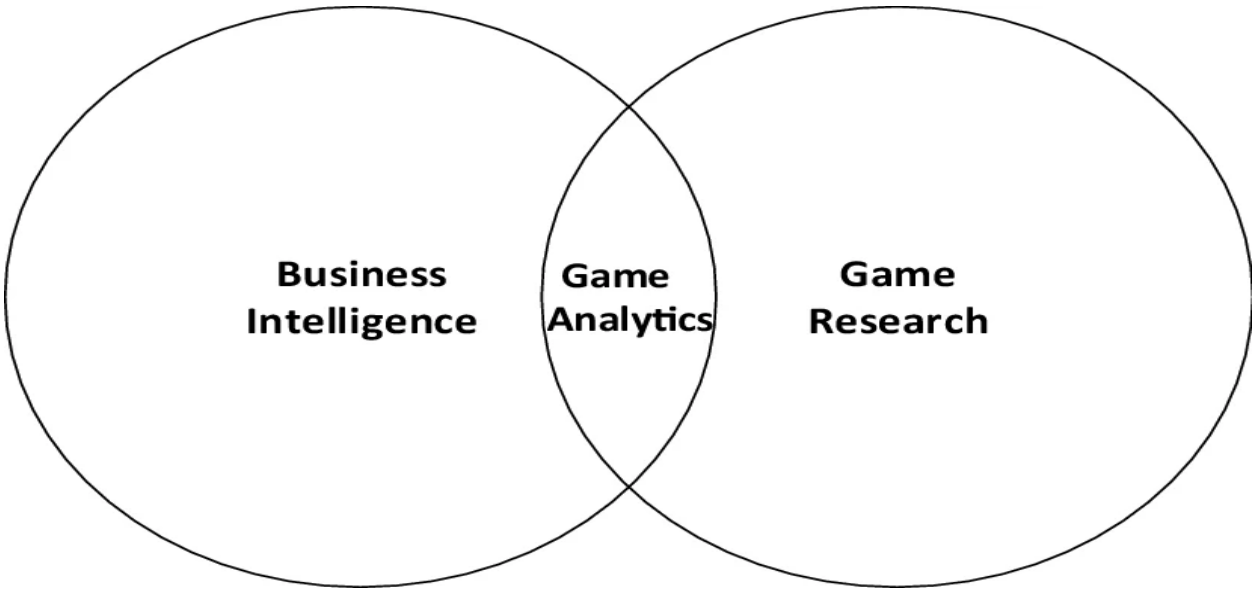
OBJECTIVE

As a business model, the essence of games is to provide a service to satisfy the player experience. From a business perspective, development in the game industry has led to the application of Business Intelligence becoming more and more extensive in identifying trending genres, single or multiplayer games, and even what consoles are

favorites. However, related research needs more systematic examination and precise classification. This paper provides a comprehensive review of the game industry, focusing primarily on game analytics.

INTRODUCTION

The game is a kind of service that provides game players with different experiences. With the continuous development of the game industry, more and more interdisciplinary knowledge and theories are being used. Game analytics derives from Business Intelligence (BI). It reflects the combination of BI with game research.



Venn Diagram for establishing the relationship

The purpose of analytics is to solve problems, make predictions in business, help decision-making, promote optimization actions, and improve business performance. Game analytics is the process of identifying and communicating meaningful patterns that can be used for game decision-making.

METHODOLOGIES

The methodology for a video game playtime and choice analysis project using data science would typically involve the following steps:

- **Data Collection**: The first step would be to identify relevant data sources and collect data. This includes data from game time surveys i.e. from Kaggle.

```
data = []
with open('hltb.jsonlines') as lines:
        for line in lines:
                data.append(json.loads(line))
```

```
import pandas as pd
df = pd.DataFrame(data)
df.head()
```

- **Data Cleaning/Transformation**: Once the data has been collected, it needs to be cleaned and preprocessed to ensure that it is suitable for analysis. This may involve removing duplicates, addressing missing values, and standardizing the data format.

```
def convert(list_dicts):
    total_dict = {}
    for h in list_dicts.keys():
        dict_1 = list_dicts[h].items()
        for i,j in dict_1:
            for k,l in j.items():
                total_dict.update({f'{h}_{i}_{k}': l})
    return  total_dict

#convert(test)
```

```
total =  df.Stats.apply(lambda x: convert(x)).to_list()
```

- **Exploratory Data Analysis**: Exploratory data analysis (EDA) involves visualizing and summarizing the data to gain insights and identify patterns. This may include creating graphs and charts to understand the distribution of game time and choice-related variables.

```
#Let's find all the avaialable platforms
total =  df.Stats.apply(lambda x: convert(x)).to_list()
consoles_list = set([g.split('_')[1] for i in total for g in i.keys()  if "Platform" in g])
consoles_list
```

- **Model Development**: Once the data has been cleaned, it's time to develop a predictive model. This may include machine learning models, such as logistic regression or decision trees, that can predict the likelihood of choice or other game-related outcomes.

```
def hours(x):
    total = []
    if x and isinstance(x, str):
        split = x.split()
        if 'h' in x and len(x.split()) <2:
            return int(x.replace('h', ''))
        elif 'm' in x and len(x.split()) <2:
            return int(x.replace('m', ''))/60
        elif len(x.split()) > 1:
            clean  = x.split()
            clean = int(clean[0].replace('h', '')) +  int(clean[1].replace('m', '')) / 60
            total.append(clean)
        return sum(total)
    else:
        return np.nan
```

- **Model Evaluation**: The next step is to evaluate the performance of the model using metrics such as accuracy, precision, and recall. This can help determine whether the model is effective in predicting game choice and game-related outcomes.

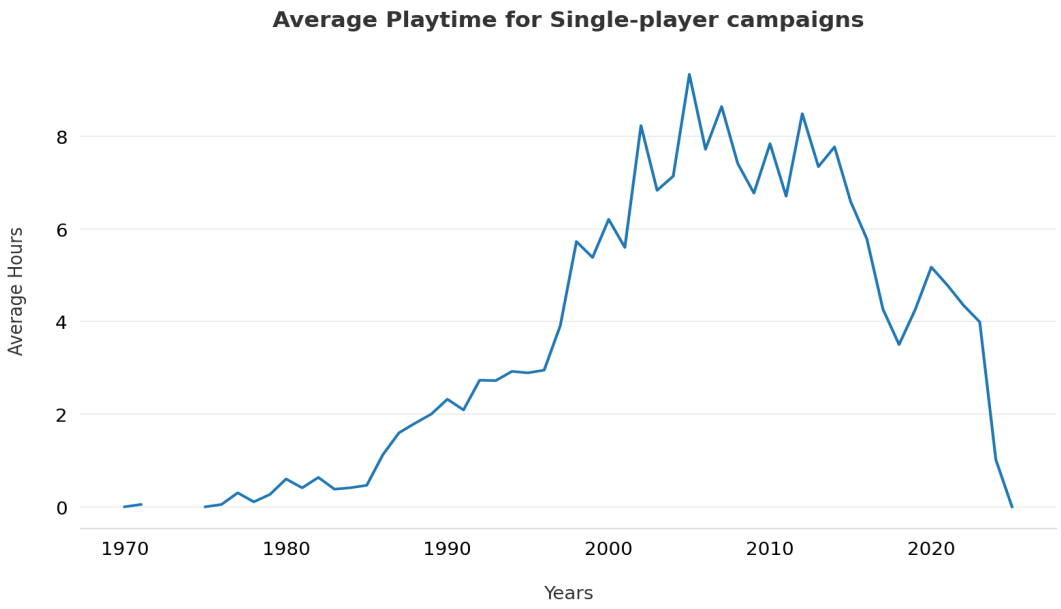
```
df.Genres.replace('',np.nan).dropna().value_counts().head(10).sort_values(ascending=True).plot(kind='barh')
plt.ylabel("Frequency", size = 12)
plt.title("Genres", size = 14)
```

```
plt.rcParams['figure.figsize'] = (8, 5)

fig, ax = plt.subplots()

ax.plot(
    df_final[['Release_date','Single-Player_Main Story_Average']].set_index(
        'Release_date').resample('Y').mean()
)
# Remove the top, right and left spines
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['bottom'].set_color('#DDDDDD')

# Second, remove the ticks as well.
ax.tick_params(bottom=False, left=False)
# Add a horizontal grid (but keep the vertical grid hidden).
# Color the lines a light gray as well.
ax.set_axisbelow(True)
ax.yaxis.grid(True, color='#EEEEEE')
ax.xaxis.grid(False)
# Add labels and a title.
# extra space between the text and the tick labels.
ax.set_xlabel('Years', labelpad=15, color='#333333')
ax.set_ylabel('Average Hours', labelpad=15, color='#333333')
ax.set_title('Average Playtime for Single-player campaigns', pad=15, color='#333333'
             weight='bold')


fig.tight_layout()
```

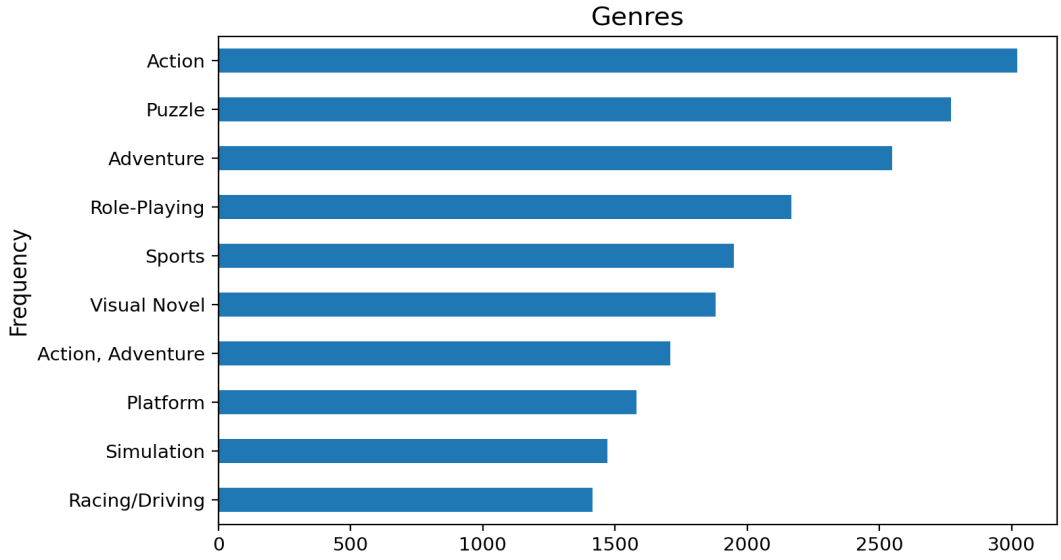
- **Model Deployment**: Finally, the model can be deployed for use in real-world scenarios. This may include integrating it into a game analysis app or choice-based gaming library to provide personalized support to individuals and companies.

ANALYSIS OUTCOMES

- Time Spent on Single-Player Games



- Genre Analysis



- Console Analysis