CS 210 - Term Project Report
Emre Soylu - 26582

Motivation & Introduction

In the realm of digital gaming, Steam stands as a ubiquitous platform, serving as a virtual playground for millions of gamers worldwide. As an ardent participant in this digital ecosystem, I embarked on a personal project to explore and analyze my own Steam data. This project delves into the vast expanse of my gaming history, employing Exploratory Data Analysis (EDA) and visualization techniques to unearth meaningful insights. Steam, with its comprehensive data records, offers a unique opportunity to scrutinize personal gaming trends, preferences, and milestones. EDA serves as the investigative backbone, allowing for the systematic exploration of diverse facets within the data. Visualization, on the other hand, transforms the raw information into compelling visual representations, making patterns and correlations readily apparent. This report aims to elucidate the significance of both Steam data and the analytical tools applied, illustrating how the amalgamation of personal gaming data, EDA, and visualization contributes to a richer understanding of one's digital gaming journey.

Data Source

The data for this project originates from my existing Steam account, and I retrieved the necessary information by generating an API Key through Steam's 'Steam Web API' (available at <https://developer.valvesoftware.com/wiki/Steam_Web_API>). Using Python scripts that I personally crafted, as showcased on my GitHub page, I harnessed the power of the Steam Web API to extract pertinent data. This API Key allowed me secure access to my account's details, enabling the retrieval of valuable insights such as;

- Game ID of a game ***<game_id>***
- Game name of a given game_id ***<gameName>***
- Total achievement count that I acquired ***<ach_count>***
- Total playtime as minute <playtime_forever>
- Last played time of a game ***<date-time>*** & ***<date>*** & ***<time>***
- Genre of a game ***<type>***
- Price of a game as MENA USD currency ***<price>***
    - MENA USD exists on Steam as a currency tailored for the Middle East and North Africa region. It is designed to provide a more

user-friendly and locally relevant currency option for gamers in that geographic area, facilitating transactions and pricing in a way that aligns with regional economic factors.

- Steam Id ***<steamId>***
- Achievement name ***<achievement name>***

The scripts, available in the respective sections of my GitHub repository, facilitated the systematic extraction of this data. Subsequently, I amalgamated all the gathered information into a comprehensive xlsx file. This cohesive dataset serves as the foundation for the Exploratory Data Analysis (EDA) and visualizations conducted in this project, offering a detailed exploration of my gaming history and patterns.

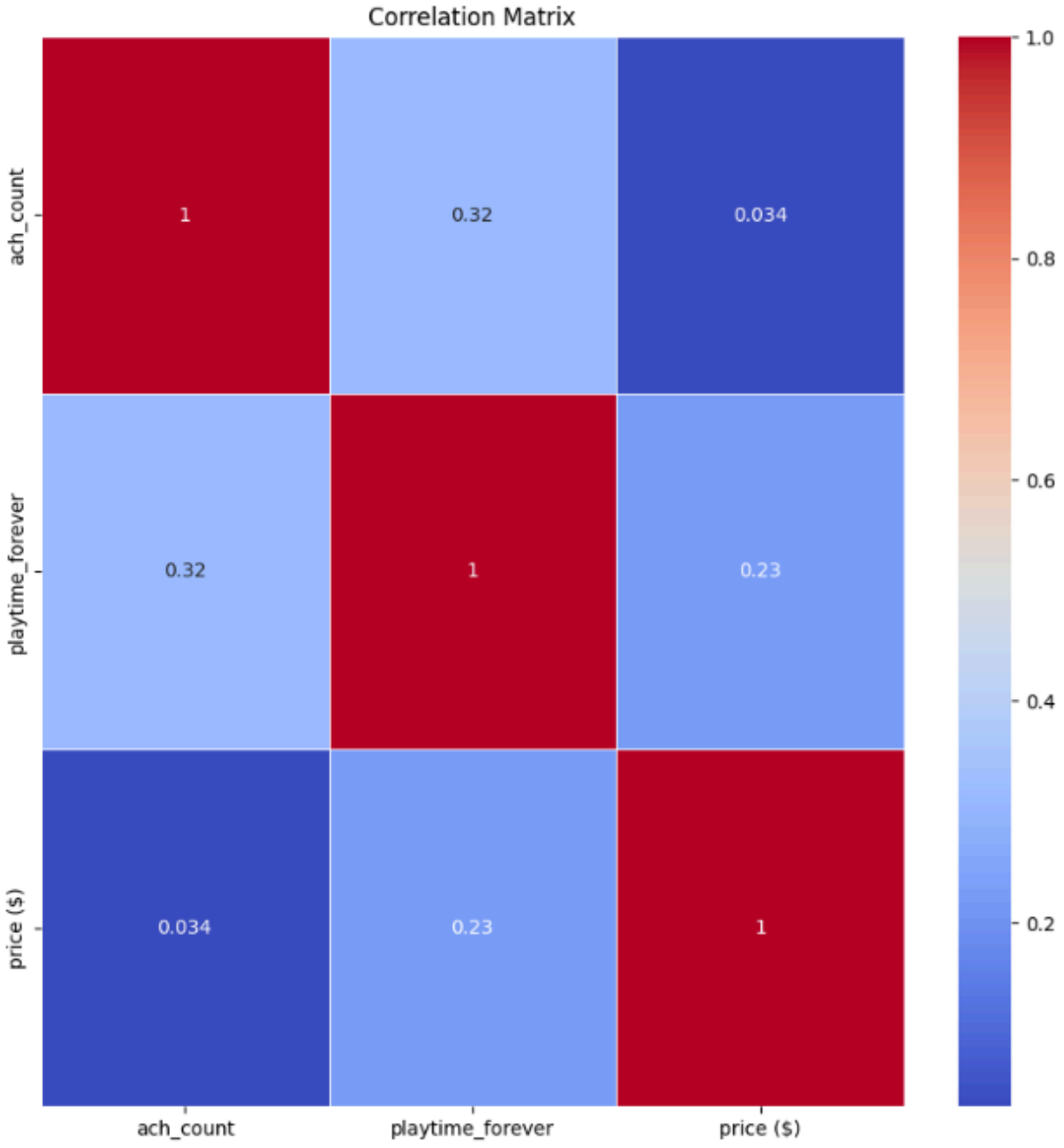
Data Analysis

- Descriptive analysis
    - Measures of Central Tendency (Mean, median, mod)
        - Mean of total achievement count for all games : The average number of achievements earned per game in my collection of 76 games is approximately 9.7.
            - ```
              mean of ach_count: 9.789473684210526
              ```
        - Mean of total playtime for all games :The total playtime per game in my collection of 76 games is approximately 710 minutes per game.
            - ```
              mean of playtime_forever: 710.1315789473684
              ```
        - Mean of game price :The average price of all the games in my possession.
            - ```
              mean of price: 19.62592105263158
              ```
        - Mode of same game statistics
            - ```
              median of ach_count: 3.0
              median of playtime_forever: 128.5
              median of price: 19.99
              ```
        - Meidan of same game statistics
            - ```
              mode of ach_count: 3.0
              mode of playtime_forever: 128.5
              mode of price: 19.99
              ```

In the 76 games I own, each priced at approximately 19 MENA USD, I achieved an average of 9.7 in-game accomplishments per
              ```

game over an average playtime of 710 minutes. This analysis provides insights into measures of central tendency for both the cost and achievement distribution within my gaming collection.

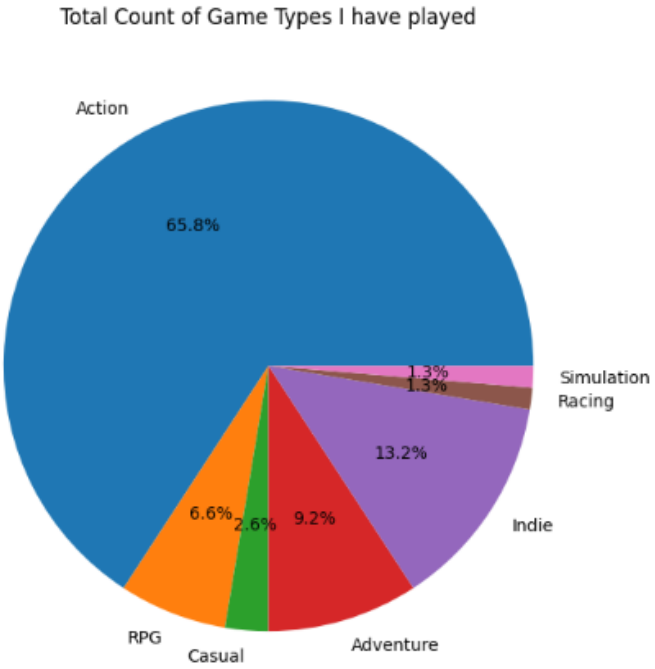
- Correlation analysis



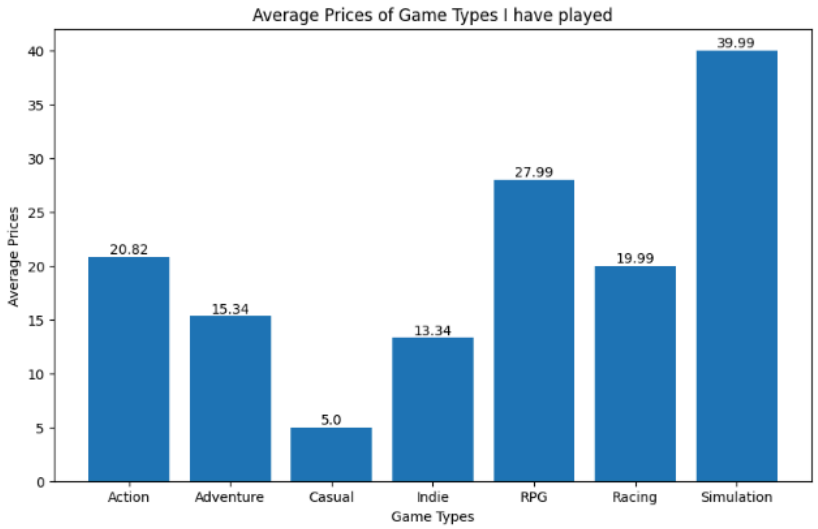
I conducted a correlation analysis to understand whether there is a correlation between the in-game achievements, total playtime, and the cost of the games in my possession. As evident in the analysis table, there is a 32% correlation between my playtime and the achievements obtained. Similarly, as the game price increases, there is a 23% correlation in the number of in-game achievements. On the other hand, I observed a very small correlation of 3.4% between the game price and in-game achievements.

- Visualization: In the visualization section, I utilized bar and pie charts to depict which genres of games I played the most, the prices among the most-played genres, and the average in-game achievement count within each genre. These visualizations offer a clear representation of my gaming preferences, expenditure patterns, and achievement pursuits across various game genres.

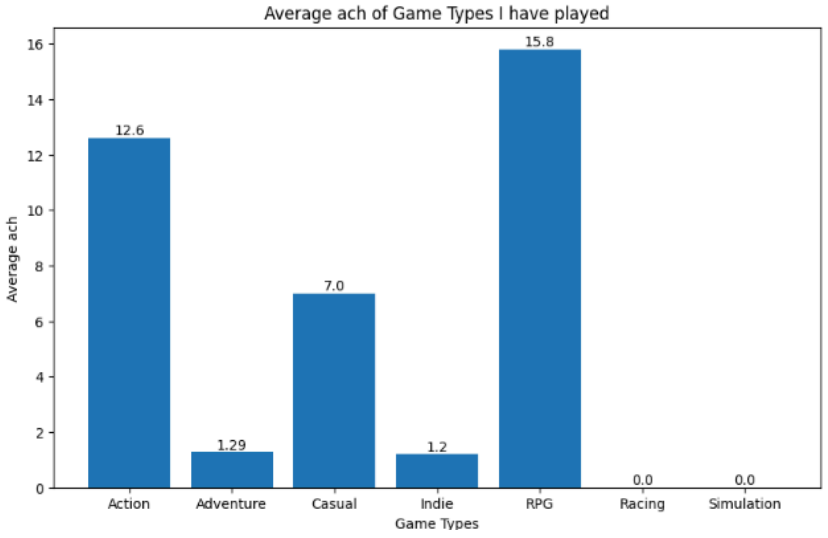
    - Total percentage of game genres that I have played: As evident, I have played games belonging to the action genre by a significant margin, with a dominating percentage of 65.8% among all my games.
        - Action → 65.8
        - Indie → 13.2
        - Adventure → 9.2
        - RPG → 6.6
        - Casual → 2.6
        - Simulation → 1.3
        - Racing → 1.3



- Average Prices of Game Types I have played: In terms of the games I've played, I have made the highest expenditure per genre on games belonging to the Simulation genre.



- Average achievement of game genres I have played



Data analysis with descriptive tables:

- Table I: Total playtime / Price
    - In this table, I aimed to determine the entertainment efficiency by taking the ratio of the minutes played to the total playtime for each paid game. This allows me to see which paid games provide a higher entertainment yield per minute of gameplay.

	efficiency	gameName	type	price ($)	playtime_forever
0	221.721722	PAYDAY 2	1	9.99	2215
10	215.402567	Battlefield™ 2042	1	59.99	12922
37	141.883768	Euro Truck Simulator 2	4	4.99	708
29	69.251337	Call to Arms	1	3.74	259
18	55.527764	The Forest	1	19.99	1110

- Table II: A table is created to identify which in-game achievement comes at a higher cost, thus indirectly revealing which game does not provide cost-effectiveness in terms of price/performance.

29	499.833333	Sherlock Holmes: The Devil's Daughter	1	29.99	6
31	499.750000	Mount&Blade: Warband	1	19.99	4
19	352.882353	Cyberpunk 2077	5	59.99	17
17	333.277778	Call of Duty: WWII	1	59.99	18
35	333.000000	Garry's Mod	3	9.99	3

- Machine learning

  Machine learning methods can be applied in two main ways: regression and classification. In this context, regression analysis has been performed to examine machine learning models. The target variable chosen is the "ach count." In machine learning, numerical values such as total playtime, price, and the converted numerical values of the game genre have been selected as intrinsic features. These numerical features are considered as independent variables.
    - Two machine learning algorithms have been individually tested. These algorithms are artificial neural networks and decision tree

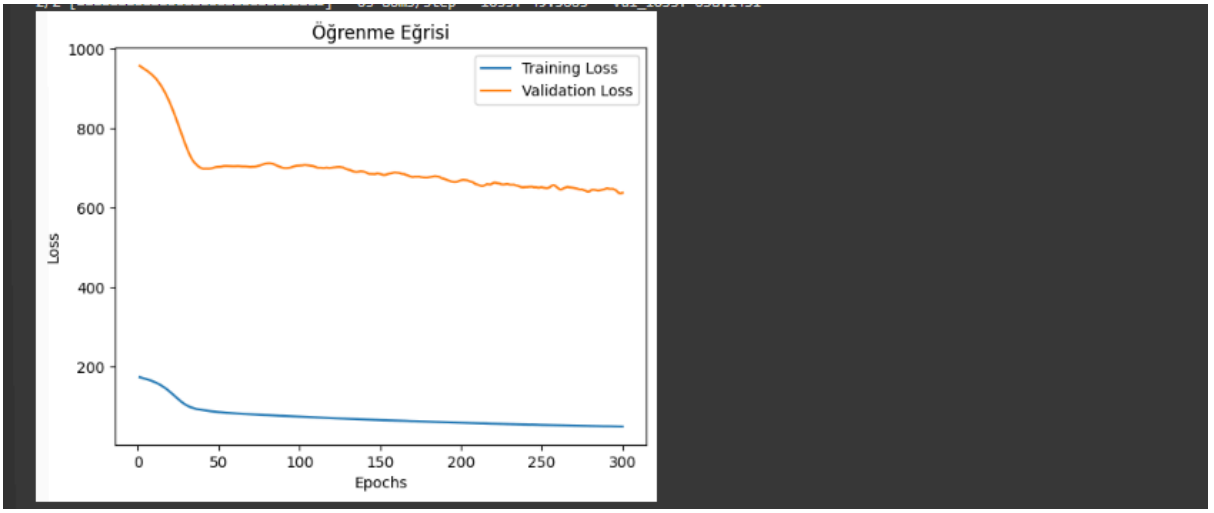
algorithms. The data has been divided into 80% training data and 20% test data. The results obtained for artificial neural networks and decision tree algorithms are provided below.

Data preprocessing algorithm is as follows; there are 51 training data and 13 test data, as seen below.

```python
# Data Preprocessing
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split

dependent = ["playtime_forever","type","price ($)"]
independent = ['ach_count']
shuffled_data = shuffle(df_ml_data, random_state=61)

y = shuffled_data[independent]
X = shuffled_data[dependent]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=61)
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (51, 3)
X_test shape: (13, 3)
y_train shape: (51, 1)
y_test shape: (13, 1)
```

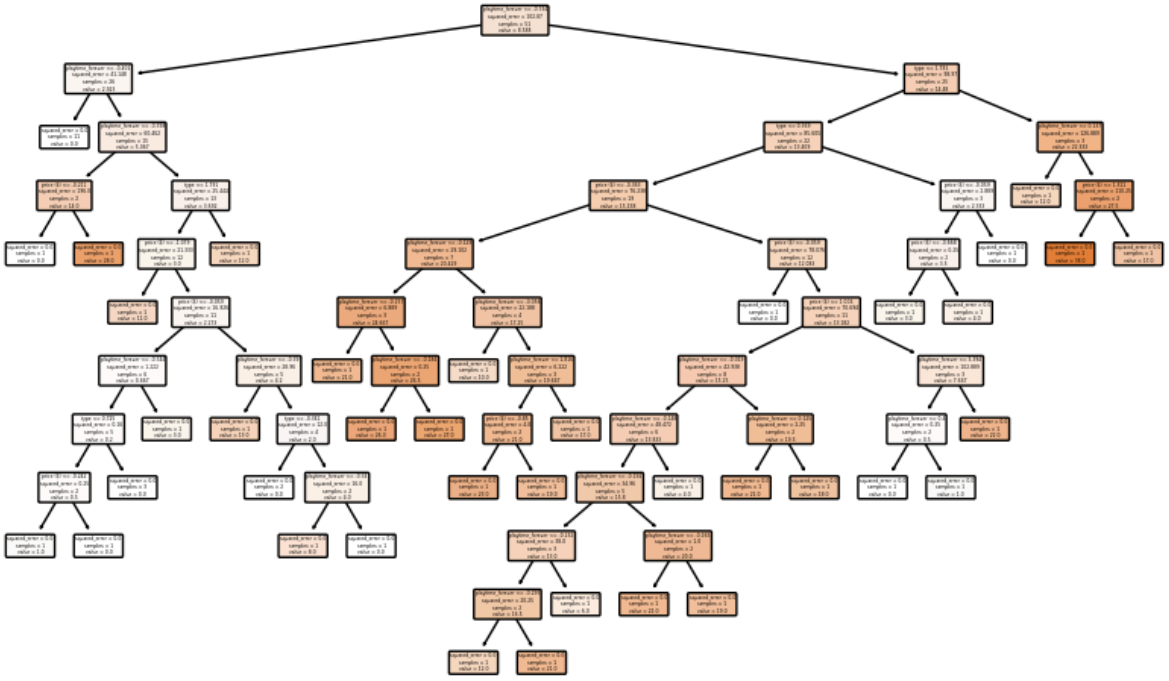
For the artificial neural networks regression model, firstly, the training and test data were scaled. Subsequently, an artificial neural network model was constructed, consisting of an input layer with 64 neurons, two hidden layers with 32 neurons each, and one output layer. The learning algorithm used was Adam, and the Loss function was set as mean_squared_error. The model was trained for 300 epochs with a batch_size of 32, and the change in the loss function for both training and test data is illustrated in the graph below.



As seen in the graph, there is a difference between the loss values in the training data and the test data. This observation indicates that the model excessively fits the training set, suggesting that it has lost its generalization ability and has overfitted. This could be attributed to the limited amount of data available for generalization. Overfitting implies that the model achieves lower performance when confronted with unseen data.

Decision Tree:
Another regression method employed is the decision tree. The regression model obtained from the decision tree is provided in the code below. The model yields a mean squared error of 702 for the test data. When compared to the artificial neural networks model, this error value appears to be similar. However, it may be considered high in comparison to a learned model, suggesting the model's inability to generalize, possibly due to overfitting.

```python
from sklearn.tree import DecisionTreeRegressor
# Karar Ağacı Regresyon Modeli Oluşturma
model = DecisionTreeRegressor(random_state=42)
model.fit(X_train, y_train)

# Modelin Performansını Değerlendirme
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Karar Ağacını Görselleştirme (Opsiyonel)
from sklearn.tree import export_text
tree_rules = export_text(model, feature_names=list(X.columns))
print("Decision Tree Rules:\n", tree_rules)

# Karar Ağacını Görselleştirme (Opsiyonel)
from sklearn.tree import plot_tree
plt.figure(figsize=(10, 6))
plot_tree(model, feature_names=X.columns, filled=True, rounded=True)
plt.show()
```



Findings

Exploring my gaming dataset has unearthed intriguing patterns and preferences, providing valuable insights into my gaming habits. The central tendencies reveal that, on average, I accomplish around 9.7 in-game achievements per game, dedicating approximately 710 minutes to each gaming session. This blend of time investment and achievement pursuit underscores the depth of my engagement with each gaming experience.

Correlation analysis has spotlighted interesting connections within the data. The positive correlation of 32% between total playtime and in-game achievements suggests that as I invest more time, my success in achieving in-game goals also increases. Similarly, a 23% correlation between game cost and achievements implies a nuanced relationship, indicating that more expensive games tend to yield a higher number of in-game accomplishments. However, a modest 3.4% correlation between game price and achievements suggests that the cost alone does not significantly impact in-game success.

Diving into genre preferences, action-packed adventures dominate my gaming landscape, constituting an impressive 65.8% of my played games. This preference is reflected in the higher expenditure on action and RPG genres, showcasing a willingness to invest more in immersive and dynamic gaming experiences.

The detailed data tables provide a closer look at the efficiency of entertainment derived from each paid game, revealing intriguing metrics such as minutes played per unit of currency. Additionally, the analysis of in-game achievement costs exposes which games might lack cost-effectiveness, potentially prompting a reevaluation of my gaming choices.

In essence, this comprehensive examination of my gaming profile offers a roadmap for future gaming endeavors. Whether optimizing my gaming budget, seeking specific genres, or balancing time and achievements, these insights empower me to curate a gaming experience tailored to my preferences and goals.

Limitations

The limitation posed by the Steam Web API, which does not make all the available data on Steam open source, has constrained the diversity within the study. For instance, the inability to access the game inventory and in-game achievement count of a user who has made their games and achievements public and is in my friends' list significantly hinders my ability to establish correlations between the games played by that user and the achievements earned. Furthermore, while working with my own data, the limited number of games and concentration on a specific genre have introduced algorithmic challenges and impacted the accuracy percentage when training the machine learning algorithm.

Future Works

In the future, with an increased number of games, the machine learning algorithm could potentially yield a lower mean squared error and achieve better learning. Therefore, repeating the same experiment with a larger number of games in the future could be presented as a part of the future work.