**GameSense: Understanding Player Sentiment Through NLP**

# Problem statement

This project aims to analyze user reviews of video games and accurately predict sentiment scores using natural language processing, vectorizers and ML models. By classifying reviews into positive or negative sentiment and identifying key terms using techniques, we can provide actionable insights to different stakeholders:

**For players**: The sentiment analysis can help users quickly assess whether a game is worth their time and money, either by reviewing specific features mentioned in positive reviews or by looking at the overall sentiment score. Can also easily submit their own reviews knowing the review is rated accurately.

**For developers**: The analysis can highlight which features are consistently praised or criticized, allowing teams to improve game design. It can also assist in identifying recurring issues, especially if users leave informal bug reports within reviews.

Game reviews on their own are very unstructured and subjective, based of its raw content it makes it hard for developers to make sense of the review as users often start ranting or go onto off topic posts as well new players are looking for something simple and effective to help them make fast and smarter choices, and avoid disappointments.

# Industry/ domain

The video game industry is a rapidly growing and highly competitive global market, estimated to generate nearly $455 billion USD in revenue. With the rapid evolution of mobile devices and gaming technology making games more accessible than ever, the mobile gaming segment alone is projected to account for almost $100 billion USD in revenue in 2024. The industry is such at a large scale that organizations like GDQ (games done quick) are able to even raise money for charity, since 2010 they have raised over 50 million for charities such as Doctors without Borders and Prevent Cancer Foundation.

When new games are released, companies like IGN, Metacritic, and GameSpot publish professional reviews that serve as a baseline for quality, shaping public perception and influencing early sales. These expert evaluations are important, but they represent only a snapshot in time. In contrast, user-generated reviews offer a continuous stream of feedback that developers can use to improve games post-launch. By applying natural language processing to analyze these reviews, this project helps extract sentiment trends and recurring keywords, giving developers and players a more complete picture of a game’s strengths and weaknesses over time. There are also new games that release which fail and get dropped entirely within such a short time which makes you wonder what is causing this

This project can be transferred over to many other industries such as classrooms, movies, music, hospitality, e-commerce, mobile app stores, etc. Essentially anything which requires a user’s input about a certain product.

# Stakeholders

The primary stakeholders in this project are game developers, game publishers, and players. Developers and publishers are directly impacted by user feedback, as it provides insight into player satisfaction, bug reports, feature demand, and overall sentiment. Understanding which aspects of a game are well-received or criticized can guide future updates, bug fixes, or even design choices for upcoming titles. This makes sentiment analysis a valuable tool for refining user experience and increasing commercial success.

Players also are key stakeholders as they often rely on reviews to decide which game they’d like to invest the time and money in. In hopes of the project it will allow players to easily look at reviews and assess general perception of the game and decide if the game they decide is something that matters to them.

# Business question

What is the sentiment of user reviews for each game, and what specific features are associated with positive or negative experiences? Understanding review sentiment helps developers create new patches with fix bugs in their games and understand features to improve future games, while for players it helps make informed decisions about where to invest time and money. The project aims for a moderate f1-score around 70/80% as this is what is acceptable under sentiment classification. I measure F1-score as it balances both precision and recall across the 3 sentiments ensuring it is accurately classified.

A false positive could mislead players and developers into thinking a game is better than it is while a false negative would ruins a games reputation and the developers business. **F1 score** is especially important here as it accounts for both the accuracy of identifying positive reviews and the importance of not missing negative reviews, which could have significant business implications.

# Data question

Can user reviews be analysed using NLP to classify sentiment values and features? The main column needed is reviews. Title and genres help group and compare the results, while rating can be used as a secondary score check to check if the sentiment score aligns with the game score.

# Data

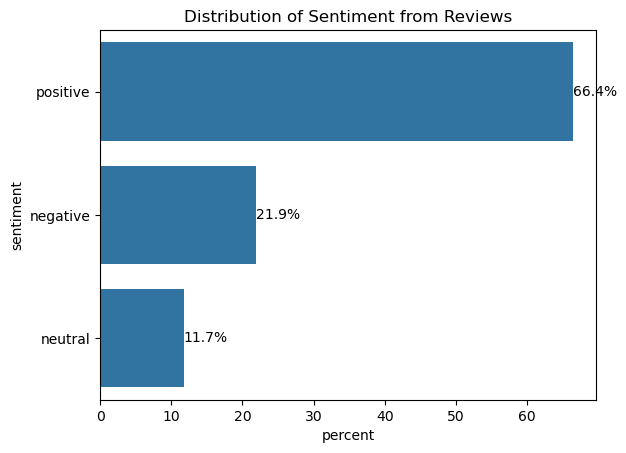
This data was sourced from kaggle which was scraped from backloggd.com. This dataset has 1512 samples and 13 features. The features are title, release date, team, rating, time listed, number of reviews, genres, summary, reviews, plays, playing, backlogs, and wishlist, our variable of interest here is reviews. There were some data issues with missing values, incorrect data type issues and duplicates which had been cleaned and removed. The data is reliable as reviews are inputted by user after creating an account on the website also behind a captcha, without an account a review can not be posted on games, although there may be some bias towards more popular games containing more reviews compared to niche games. The data is available at anytime and steadily growing as new games release more people play each game.

# Data science process

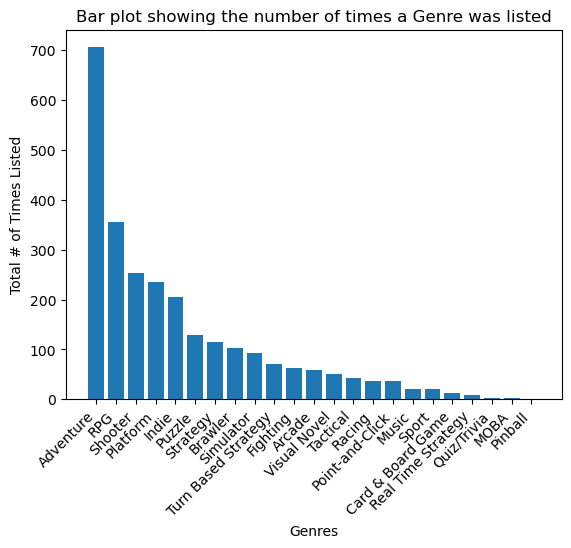
## Data analysis

I first cleaned the data by removing duplicates, missing values and adjusting the data types to the correct types. I separated the reviews in the reviews column and exploded the dataset so every column has one review. I removed all the reviews that are not English to remove any inaccuracy issues. I then processed and cleaned the raw content including lower-casing, punctuation removal, emoji removal and extra whitespace. The text was then vectorized into count vectorizer, wordlevel tf-idf, n-gram tf-idf and character level tf-idf. In my EDA I found the genre distribution imbalanced with Adventure being the most common genre and RPG being in second at half the titles. The sentiment distributed is imbalanced with 66.4% reviews positive, 21.9% reviews negative, and 11.7% reviews are neutral. Using the final model I was able to find the top 10 words that influences the reviews positively and negatively. The pipeline is reusable as all the steps are implemented programmatically, so they can be added onto new data as long as its kept in the same format. After cleaning the text, I stored the data in a new dataframe. The data was then split into train and test sets for our models. Each vectorizer was used with the data to form large matrices of numbers to be used as input for the models. SHAP values arrays were created to explain the feature importance of words for each review.

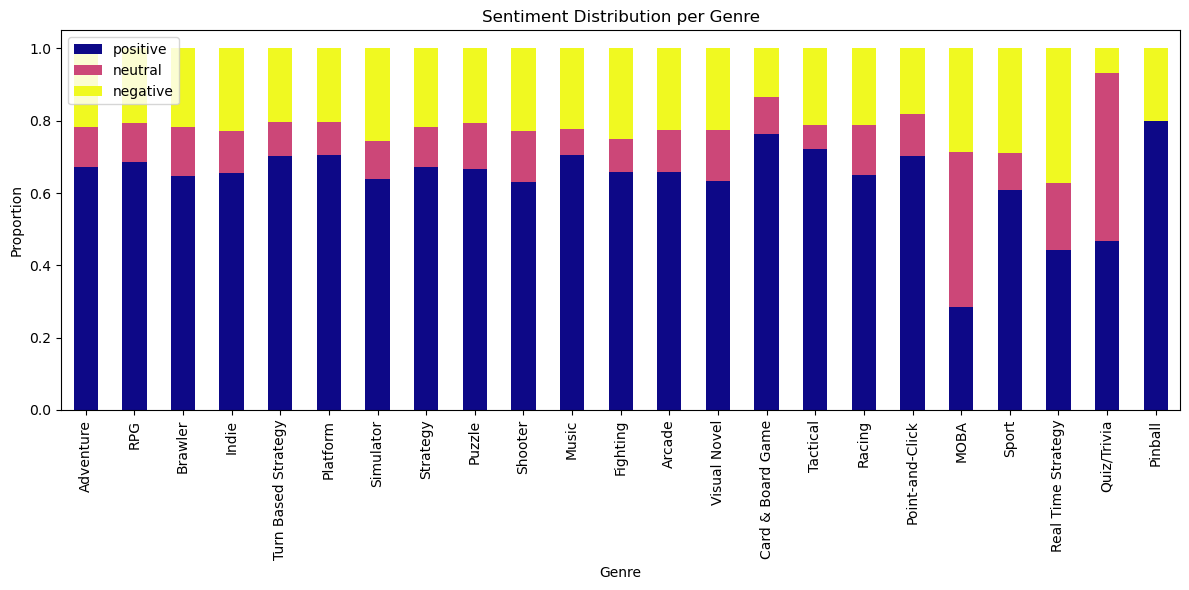
*Sentiment Distribution:*

**

*Genre Distribution:*

**

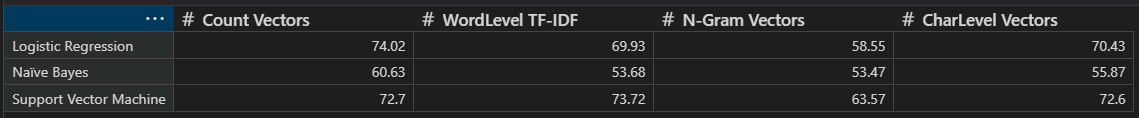
## *Sentiment Distribution per Genre:*

**

## Modelling

The main feature used in this project is ‘Reviews’, however when I created the dataframe I also used title, genres, rating, number of reviews, and summary. I found interesting features using SHAP to define which words are important per each review, and was able to find which words positively and negatively impacted each sentiment. The feature engineering techniques that were used are text preprocessing (Cleaning text), VADER (sentiment classification) and the 4 vectorizers that are wordlevel tf-idf, character level tf-idf, n-gram level tf-idf and count vectorization. The 3 models I used are logistic regression, Multinomial Naive Bayes and Support Vector Machine. The training time of each model did not take long, just a few seconds for all of them since the dataset isnt large and the models are efficient. For tools I used pandas, numpy, matplotlib, seaborn, regex, sklearn, nltk and langdetect. Code was created on Visual Studio Code. From each model I retrieved the classification report and used the f1-score from the results. The final model selected is Logistic Regression with WordLevel TF-IDF after tuning.

*Results Before Tuning:*

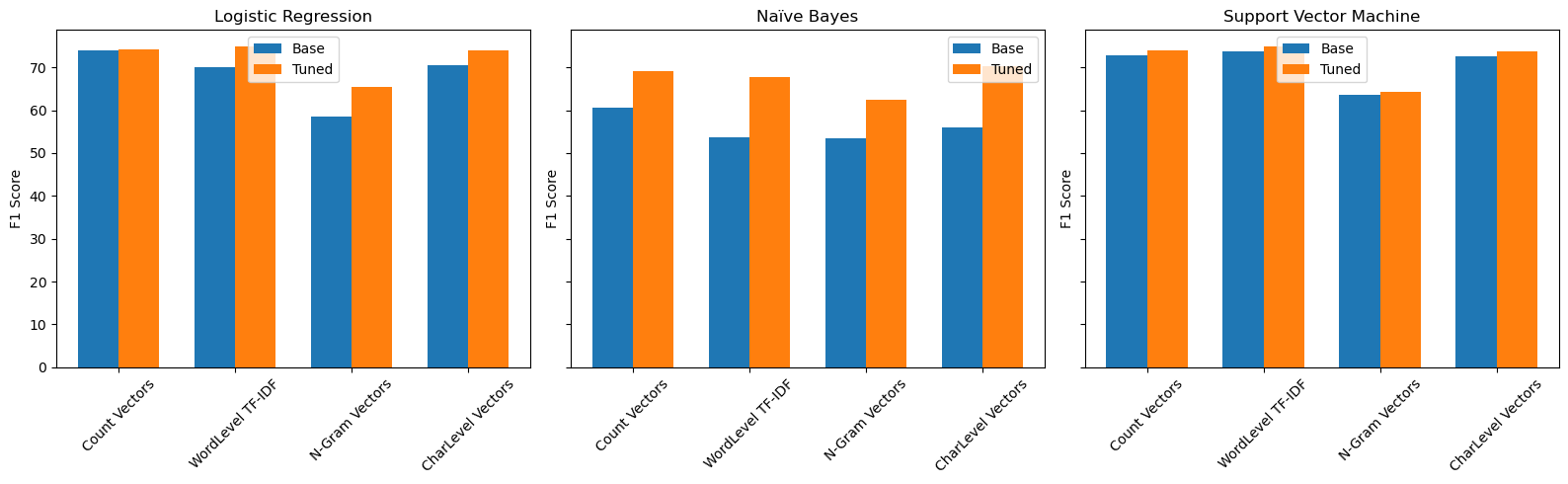
**

*Results After Tuning:*

## Outcomes

From this project I can highlight that logistic regression with WordLevel TF-IDF vectorizer is the best performing model to predict sentiment in player reviews achieving the highest score of 74.98%. A solid alternative model behind logistic regression is support vector machine. Hyperparameter improved the results across all the models naive bayes showing the most improvement. N-gram vectorizer needed more tuning as it was the lowest performing in all areas. SHAP analysis was used to interpret which words gave the most predictive power to each review, this offers valuable insights to stakeholders. Using this project and findings I can confirm that it is possible to reliably predict player sentiment using text alone.

*Final Comparison:*

**

## Implementation

The model needs to be integrated into a scalable pipeline that can handle incoming reviews in real time or a batch process. This requires the text processing and vectorizations steps are consistent, reusable and reliable. The performance of the model needs to be consistently monitored to detect any performance drift or degradation overtime as features in reviews are constantly changing such as new slang. The time it takes for a review to be processed into the final results must be processed in minimal time. Additionally SHAP analysis needs to be optimised as it could be computationally expensive. The final deployment must follow all rules, guidelines, privacy and ethical concerns.

# Data answer

The question of if the model can accurately predict player sentiment using player reviews was satisfactorily answered. Through the various model and vectorizers, I was able to achieve a strong result of 74.98% with Logistic Regression with Word-Level TF-IDF, which means the performance goal between 70-80%. Through testing with cross validation and train/test splits I have a high level of confidence in my results. Furthermore, I use SHAP analysis which clearly defines which features are relevant to the prediction of the sentiment which supports my reliability.

# Business answer

The question of if I can create a model that can determine sentiment of user reviews for each game, and what specific features are associated with positive or negative experiences was satisfactorily answered. I built an accurate final model which gave strong performance but also I was able to use that model to gather insights for each review, to see which features gave the most predictive power of positive, neutral and negative reviews. Given my F1-score of 74.98% and testing through cross validation, I have a high level of confidence in my results, although more improvements can be made with more data and tuning.

# Response to stakeholders

The key takeway is player sentiments can be predicted using NLP, vectorizers and ML models, for this project being Logistic Regression with Word-Level TF-IDF giving the strongest performance. Using this model we can identify which features are driving the prediction positively and negatively using SHAP analysis. Using this insight stakeholders can clearly identify common pain points or praised features, adjust features as needed and design marketing strategies more effectively. It is ideal to keep investing insights and gather more data to get better performance from the model.

# End-to-end solution

We must first gather the raw user reviews and preprocess the data, such as tokenization, cleaning, removing non-english reviews, and transformation using Word-Level TF-IDF. The data is fed into the model to predict the sentiment of each review as positive, neutral or negative. SHAP is used to interpret the text, to pull the key words driving the prediction. The data science process can be put into a pipeline where new data can be inputted in real time and analysed. This enables both our players and developers to monitor player feedbacks, and understand what features are driving the predictions. In turn this helps players make informed decisions on what games to spend time and money into, and for developers to understand the criticizing features on their games, what to improve on or use for future.

# References

* <https://www.statista.com/topics/868/video-games/>
* https://backloggd.com
* [https://gamesdonequick.com](https://gamesdonequick.com/)
* [https://pandas.pydata.org](https://pandas.pydata.org/)
* [https://numpy.org](https://numpy.org/)
* [https://matplotlib.org](https://matplotlib.org/)
* [https://seaborn.pydata.org](https://seaborn.pydata.org/)
* <https://pypi.org/project/langdetect/>
* <https://docs.python.org/3/library/re.html>
* scikit-learn.org
* [https://www.nltk.org](https://www.nltk.org/)
* <https://www.ubisoft.com/xdefiant-our-message-to-players>