Data Science for Business - Team Project Final Report

Steam Genre Prediction

Team 44 - Sriya Vemuri, Amna Mahmood, Samir Dar, Kerry Chen, Calida Mathias

Dataset Used: Steam apps metadata (one row per app)

Goal: Predict multi-label genres from app metadata to help developers and publishers position new titles and plan distribution.

Business Understanding

- **Business Problem:**

  Steam is the world's largest digital distribution platform for PC gaming, developed by Valve Corporation. It hosts over 50,000 games and serves as the main storefront where developers publish and market their games. When a studio is getting ready to launch a new game, one of the most important choices is which genre tags to put on the store page, since those tags shape how easily players find the game, who sees the ads, which bundles it fits, and which influencers to brief. Nowadays, that decision is mostly gut feel and quick competitor checks, which can miss useful secondary genres, create inconsistency across teams, and slow launches.

- **How a data mining solution addresses it:**

  We built and evaluated a data driven assistant that reads early game details such as platform, price plan or free to play status, screenshots, trailers and demos, and basic developer or publisher history. It recommends the most likely genre tags with confidence and turns those facts into a short ranked list. It also provides confidence scores and a simple threshold so teams can choose between higher precision or more reach. It further highlights which inputs mattered most such as platform flags, price, and media richness. This cuts risk, speeds go to market, and surfaces secondary genres to test for broader reach.

Data Understanding

The data comes from a merged Steam dataset on Kaggle that includes information about 66,928 games available on the Steam platform. Each row represents one unique app/game and contains 31 variables describing the game's characteristics (e.g. title, genre, developer, publisher, pricing details, media content, and system

requirements). The original dataset was uncleaned and contained mixed data types, missing values, and several semi-structured fields.

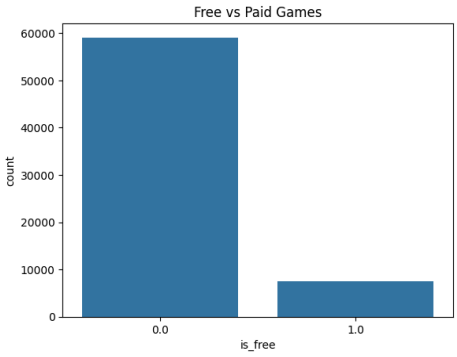
Key variables include steam_appid, name, genres, price_overview, is_free, required_age, platforms, supported_languages, developers, publishers, screenshots, and movies. Many of these fields were stored as text or JSON-like objects and needed to be parsed before analysis. Some columns, such as controller_support, dlc, and legal_notice, had a large amount of missing data.

This dataset supports our goal of predicting game genres from early-stage metadata. Features such as platform, price, and media richness are available before launch and align directly with the decisions studios make when positioning new titles. However, the data also shows certain biases: Windows games dominate the sample compared to Mac and Linux-based applications, popular genres like Action and Indie are overrepresented, and tagging practices vary across developers and over time. These patterns create class imbalance and a bias toward common genres, which we address later during data preparation and modeling

Data Preparation and Exploratory Data Analysis

Our raw steam dataset contained several inconsistencies, malformed text, missing values and anomalies. Using Python and Pandas, we removed redundant and mostly empty columns, standardized column names, and cleaned text fields by stripping HTML tags and special characters. We also parsed nested JSON data (such as genres, categories, and developers) into readable lists and converted relevant fields into proper numeric, date, and boolean formats. The cleaned and well-structured dataset was then exported as an Excel file for further analysis.

We initiated our project with basic exploratory analysis. *df.describe()* is another function we utilized to uncover descriptive statistics of the numerical columns. We found that approximately 11% of the games were free (is_free mean came out to 0.113), meaning that paid games mostly dominate the Steam platform. Moreover, the required_age field showed a mean of 1.75 with an extremely high standard deviation of 387, showing us the presence of



extreme outliers in this column. This exercise emphasized the need for outlier treatment.

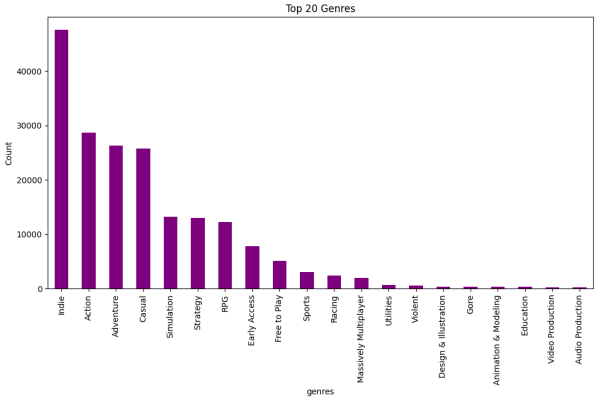
We then attempted to capture our data through some informative visualizations. We started out with a simple barchart visualization comparing the distribution of free versus paid games in our data set. It revealed that the Steam marketplace is dominated by paid titles. This aligns with the platform's major revenue model, where free titles drive user acquisition and initial engagement, while premium releases remain the core business.

Consequently, to capture the qualitative user feedback for all the entries, we cleaned the reviews column and visualized most repeated words using a word cloud. Custom stopwords were strictly used as guardrails to exclude trivial or expected terms such as "*game*", "*play*", "*steam*" etc. The resulting word cloud reflected prominent user emotions (e.g. "*fun*", "*experience*", "*great*", "*good*"), key elements (e.g. "*rock*", "*character*" , "*gameplay*") and popular genres (e.g. "*indie*", "*adventure*"). The popularity in presence of features like *story* and *character* suggest that narrative-driven game elements strongly correlate with player satisfaction and game popularity.

Next, we shifted our focus to the genre composition of our dataset. Since the genres field was stored in a semi-structured string consisting of all the genres in a game, it required cleaning and consistency which was achieved by removing brackets, quotes and escape sequences. After cleaning, each genre list was "exploded" such that each genre had its own row. Using Vader sentiment analyzer, we computed an average sentiment score

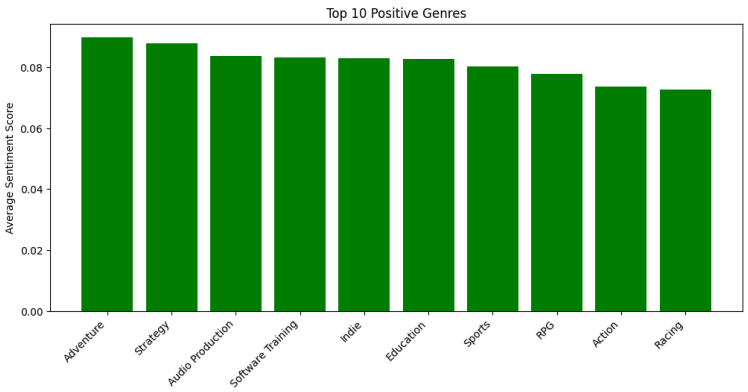
for each genre, while ensuring that each genre had at least 5 reviews for robustness purposes. Resultingly, we plotted a barchart of the 10 most positively received and 10 most negatively received genres.

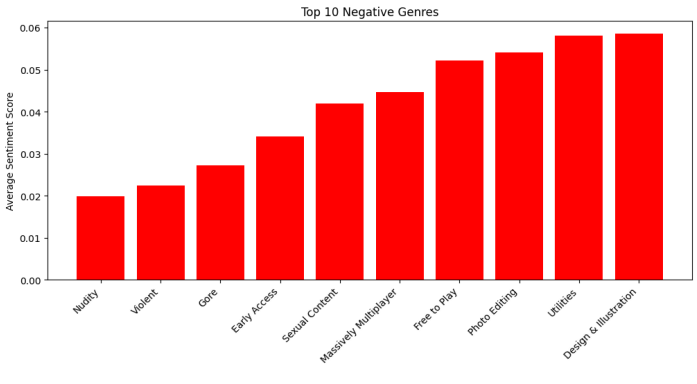
The resulting visualizations suggest that genres that combine creativity, problem-solving and mental skills, such as “adventure”, and “strategy” are perceived very positively, alongside game-play enhancing features like audio production. On the contrary, incomplete products and controversial content such as “*nudity*”, “violence” and “*gore*” are negatively perceived (eg. in the case of “*early access*”). Game developers could, thus, be careful about content sensitivity and focus on positively perceived genres.



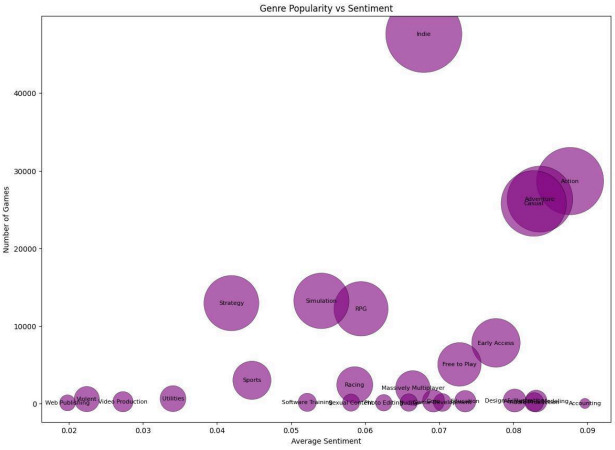
A frequency distribution of the top 20 most common genres in the data set was also plotted on a histogram; this reveals that the steam platform has a heavy concentration of “indie,” “action,” “adventure,” and “casual” games while genres like “education” remain unexplored.

A co-occurrence matrix was plotted to investigate how often genres overlap across games in our data. The resulting heatmap shows that the most co-occurring genres are “action” and “indie,” reflecting that smaller game developers probably adopt action-mechanics in games. They were followed by “*casual*” and “*indie*”, signalling the popularity of easy-to-play experiences amongst independent developers. These patterns can be





used for game positioning and genre-tagging decisions to improve visibility, marketing and optimizing recommendations for target player segments.

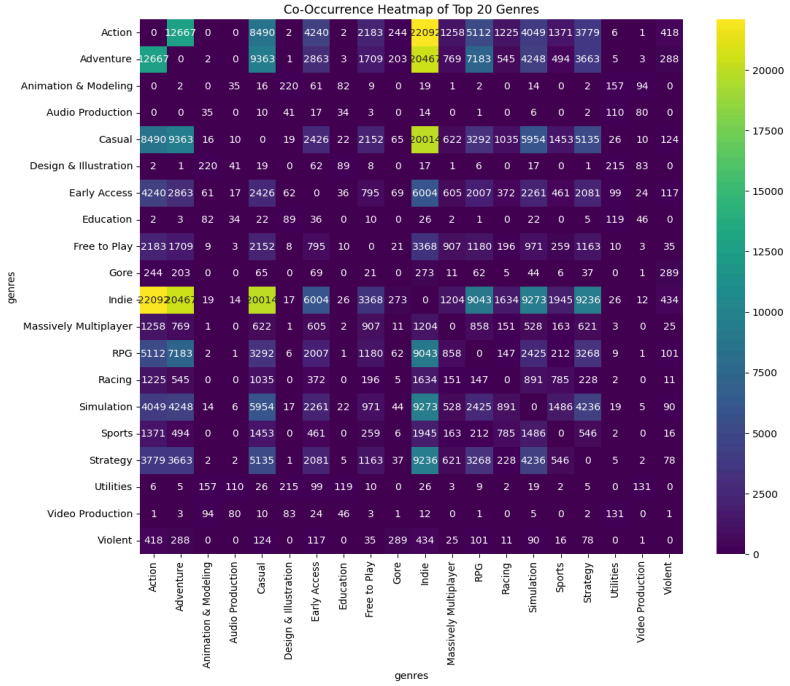


Lastly, we examined the relationship of genre popularity (proxied by number of games) and average sentiment about the genre. The result highlights "*indie*" as the most common genre, while "*action*", "*adventure*" and "*casual*" have the most positive sentiment score and a decent number of games. These explorations help developers understand which genres to explore, and how to position new titles, considering how exhausted the genre is and how well it performs in terms of user experience and sentiment.

Interestingly, the **Accounting** genre received the highest average sentiment score, despite having close to 0 games in the dataset - perhaps the ultimate niche fanbase!

Modeling

To predict the genres of games, we framed the task as a multi-label classification problem, since each game can belong to more than one genre. We built and compared three types of models: Logistic Regression, Random Forest, and Linear Support Vector Classifier (SVC). Each model was trained and evaluated using ten-fold



cross-validation to ensure consistent performance across different splits of the data. We chose these algorithms because they are well-known, interpretable, and work efficiently on structured tabular data like ours.

- Logistic Regression is simple, fast, and produces clear coefficients that show how each feature influences the prediction.
- Random Forest captures non-linear relationships between variables and can handle noisy or missing data fairly well while providing useful feature-importance measures.
- Linear SVC, on the other hand, focuses on finding the best separating boundary between genres and performs well even with high-dimensional feature spaces.

We also considered more complex models, such as Gradient Boosting (XGBoost or LightGBM) and deep learning approaches using text embeddings. While these could potentially improve performance, they require heavier computation and make the model less transparent, which goes against our goal of providing clear business insights.

This modeling approach helps solve the business problem by transforming early game metadata into actionable predictions. The assistant can automatically recommend a ranked list of likely genres along with confidence scores, helping studios make faster, data-driven tagging decisions. Random Forest, in particular, provides insights into which inputs—like price, platform, or number of media assets—matter most for genre prediction. These results directly support business goals by reducing manual effort, speeding up store setup, improving discovery for players, and helping teams understand what aspects of their games influence positioning success.

Evaluation

We evaluated the models using ten-fold cross-validation to ensure stable and reliable performance across different splits of the data. Because this is a multi-label classification problem, we focused on metrics that capture both per-label accuracy and overall prediction quality.

The key evaluation metrics used to assess our models are as follows:

- **Micro-F1:** Evaluates overall model performance by balancing precision and recall across all labels together.

- **Macro-F1:** Measures performance on each label separately, giving equal weight to both common and rare ones.
- **Weighted-F1:** Similar to Macro-F1 but gives more importance to frequently occurring labels for a realistic overall score.
- **Hamming Loss:** Captures the proportion of labels the model predicts incorrectly — either missing or adding extra labels.
- **Tuned Threshold:** The probability cut-off used to decide when to assign a label, helping balance precision and recall.

Key metrics (means ± std across folds):

- **Random Forest**

  Micro-F1: 0.5203 ± 0.0285 (best)

  Macro-F1: 0.1661 ± 0.0174

  Weighted-F1: 0.5093 ± 0.0318

  Hamming loss: 0.1317 ± 0.0099 (best / lowest)

  Tuned threshold (mean): 0.3300 ± 0.0350

- **Logistic Regression**

  Micro-F1: 0.3953 ± 0.0153

  Macro-F1: 0.2015 ± 0.0134

  Weighted-F1: 0.5470 ± 0.0190

  Hamming loss: 0.3012 ± 0.0266

  Tuned threshold (mean): 0.4500 ± 0.0236

- **Linear SVC**

  Micro-F1: 0.3902 ± 0.0215

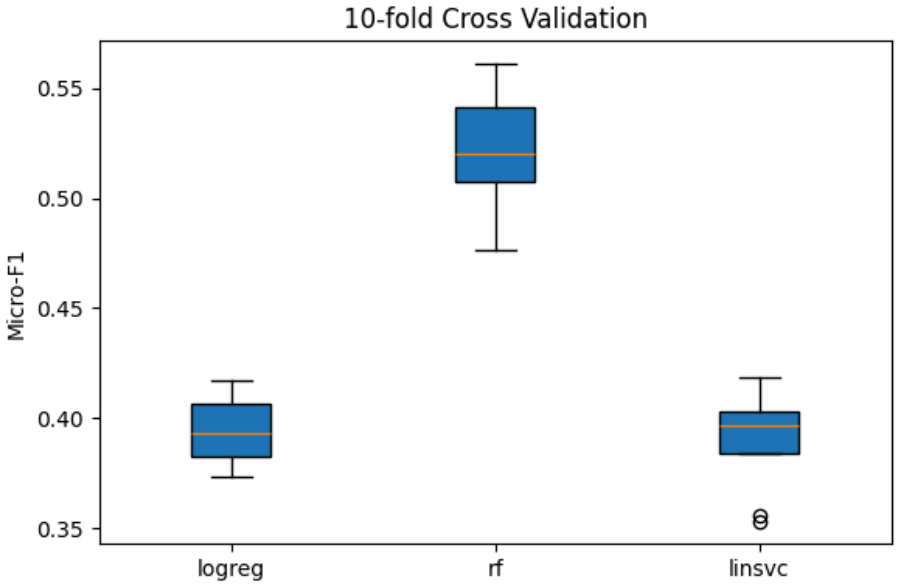
  Macro-F1: 0.2043 ± 0.0144

  Weighted-F1: 0.5502 ± 0.0234

  Hamming loss: 0.3088 ± 0.0445

Tuned score cut (mean): −0.0955 ± 0.0723

- **Interpretation**



  - RF is the best overall tagger by Micro-F1 and Hamming—i.e., it makes the fewest per-label mistakes overall.
  - Macro-F1 is low for all models (≈0.17–0.20) due to extreme class imbalance; rare genres have near-zero recall.
  - Weighted-F1 favors LogReg/SVC because common labels dominate the weighting.

Deployment

- **How the result of the data mining will be deployed:**

  Our model will be used as a "Genre Tagging Assistant" that helps game studios and publishers decide which genres best fit a new game before launch. When a developer enters early game details—like price, platform, and available media—the tool will automatically suggest a few likely genres, along with how confident it is in each suggestion. This makes the process faster, more consistent, and less dependent on guesswork. Over time, the model can be retrained with new Steam data to include upcoming games and changing trends, keeping the recommendations fresh and relevant.

- **Issues the firm should be aware of regarding deployment:**

  Before using the tool widely, the firm should be aware of a few practical challenges. The model's

accuracy depends on the quality and completeness of input data—missing information like screenshots or pricing can affect the model's predictions. It also needs regular updates as new game types and genres appear. Finally, the tool must be easy for developers to understand and trust: it should clearly show which features (e.g. price or platform) influenced each recommendation. This transparency will help developers use it confidently.

- **Ethical considerations:**

  The model should be used as a support tool, not as the final decision-maker. Developers should still review the suggested genres before publishing. Since the model learns from existing games, it might naturally favor popular genres like Action or Indie; it is important to review results for niche games carefully. All data used for training is publicly available Steam metadata, ensuring no privacy or ethical concerns. Being open about how predictions are made helps maintain fairness and trust.

- **Risks and mitigation:**

  The main risks are that the model could become outdated, biased toward common genres, or used without proper human checks. To reduce these risks, the model should be retrained regularly, especially as new trends appear. Developers should always review and approve final tags instead of relying only on the model. Bias can be reduced by balancing data during training, and technical risks can be managed by keeping the tool simple, documented, and easy to maintain. With these precautions, the Genre Tagging Assistant can be a reliable and practical aid for studios.

Appendix: Team Contributions

This project was a collective effort by all five members of **Team 44**, with responsibilities shared across different stages of the data science process.

- **Amna Mahmood and Kerry Chen** worked jointly on **data cleaning, preprocessing, and exploratory data analysis**, ensuring the dataset was structured, consistent, and ready for modeling.

- **Sriya Vemuri, Samir Dar, and Calida Mathias** collaborated on the **modeling and deployment** stages, including building, evaluating, and interpreting the machine learning models, as well as discussing how the solution could be implemented in practice.
- Together, the team co-developed the **final report and presentation deck**, aligning the technical outputs with the business problem and narrative.

All members contributed equally to discussions, decision-making, and review of the final deliverables.