Correlating descriptive tags and User Ratings of Visual Novel games using machine learning

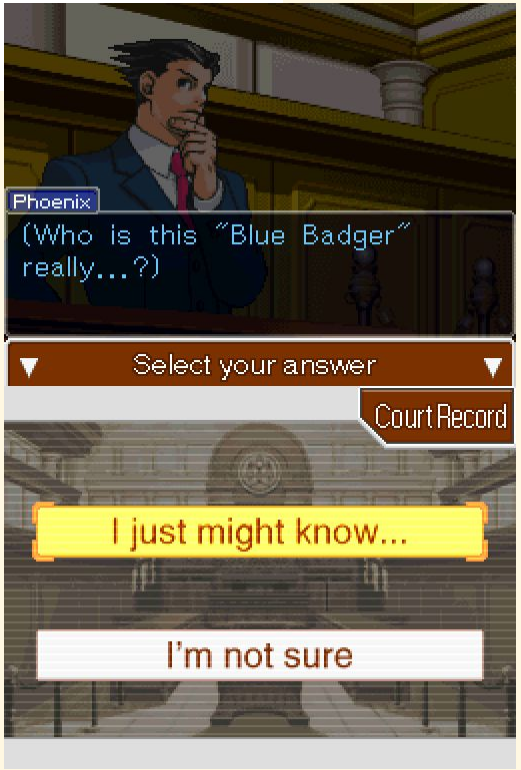
Max Gardos
UCSC Applied Mathematics Department
June 15, 2023

General Description and Hypothesis + Background

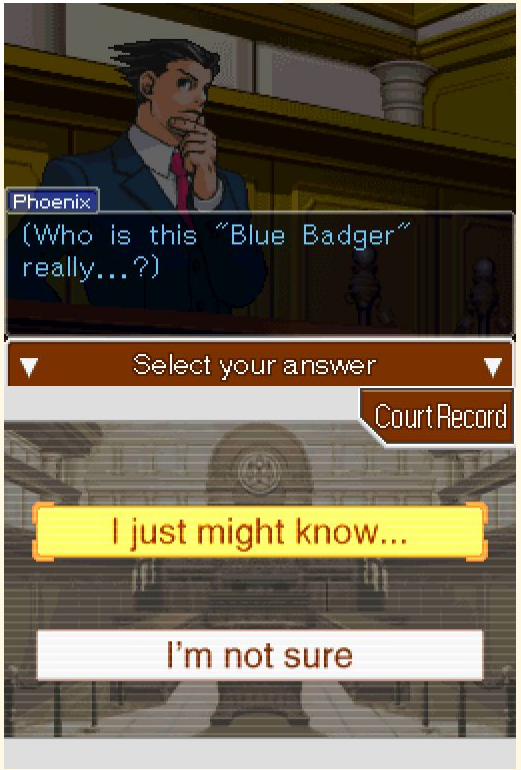
- Video game descriptors or “tags” are useful tools for players to find games that match their interests.

- Visual Novel games are minimally interactive games characterized by text-based dialogue accompanied by illustration and choices which often determine a branching storyline.

- Some high-grossing “gacha” games (like Fate Grand/Order) incorporate elements from Visual Novel games into their stories. It would be profitable for R&D teams to understand what features of these games might attract new players and retain current ones.



General Description and Hypothesis + Background



• On vndb.org, the most comprehensive public database for translated games in the Visual Novel genre, games commonly have upwards of 30 tags. For each game, users vote for how applicable they think the tag is to the game and for an overall rating of the game are counted and timestamped.

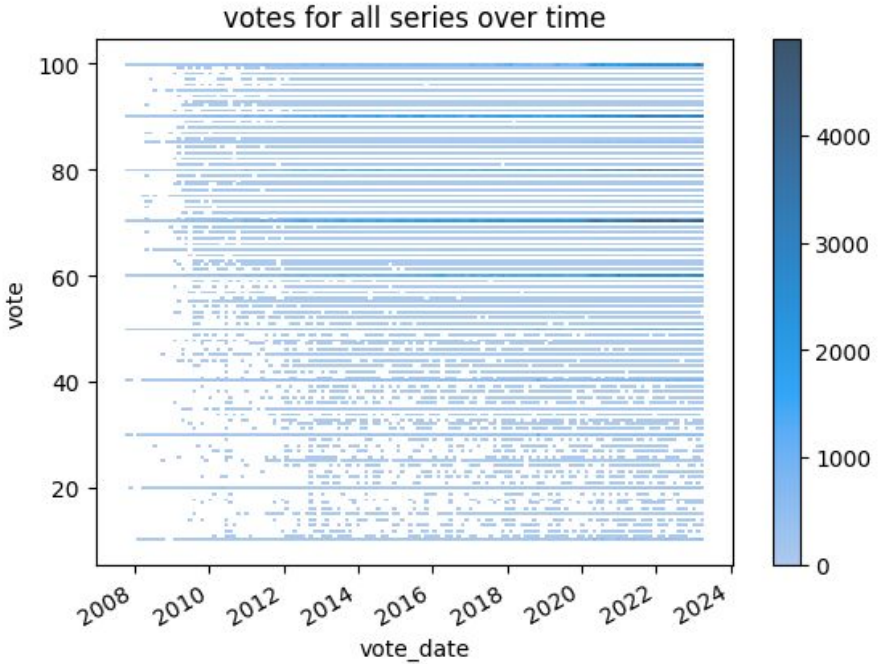
• We can use machine learning techniques to extract relevant tags (as features) across all games and to create a predictive neural net which aims to predict the reception of a game based on its tags.

Aims of the project

- Utilise LASSO feature selection to identify the most important features (tags).

- Create visualizations that connect these features and other information to time-series data to more easily observe how they change over time.

- Generate a neural net model for predicting user vote scores after LASSO Regularization based on descriptor voting data.



Model + Method

DATA:

- Dataset was converted to a dataframe where each row was a timestamped user vote for the score of a series where the features were that user's applicability scores for each of the tags of that particular game.
- Applicability scores can be any of the following numbers: [-1, 0, 1, 2, 3]
- Scores for the overall rating of the game can be integers in range [1,100]

Model + Method

	uid	vid	vid_vote_date	vid_vote	gid_vote															
tag					g10	g100	g1000	g1001	g1002	g1003	...	g990	g991	g992	g993	g994	g995	g996	g997	g998	g999
0	u10003	v2123	2011-06-30 00:00:00+00:00	70.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	u100051	v12402	2017-02-12 00:00:00+00:00	94.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	u100051	v18077	2019-10-26 00:00:00+00:00	67.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	u100051	v18636	2016-01-03 00:00:00+00:00	61.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	u100051	v430	2016-12-03 00:00:00+00:00	69.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34325	u99841	v20256	2019-10-30 00:00:00+00:00	75.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
34326	u99841	v13999	2019-06-23 00:00:00+00:00	70.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
34327	u99841	v28985	2020-12-12 00:00:00+00:00	82.0	0	0	0	0	0	0	...	0	0	0	2	0	0	0	0	0	0
34328	u99841	v21646	2019-10-30 00:00:00+00:00	77.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
34329	u99841	v26790	2020-04-30 00:00:00+00:00	78.0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	3	0	0

34330 rows × 2667 columns

Y

X

Model + Method

LASSO:

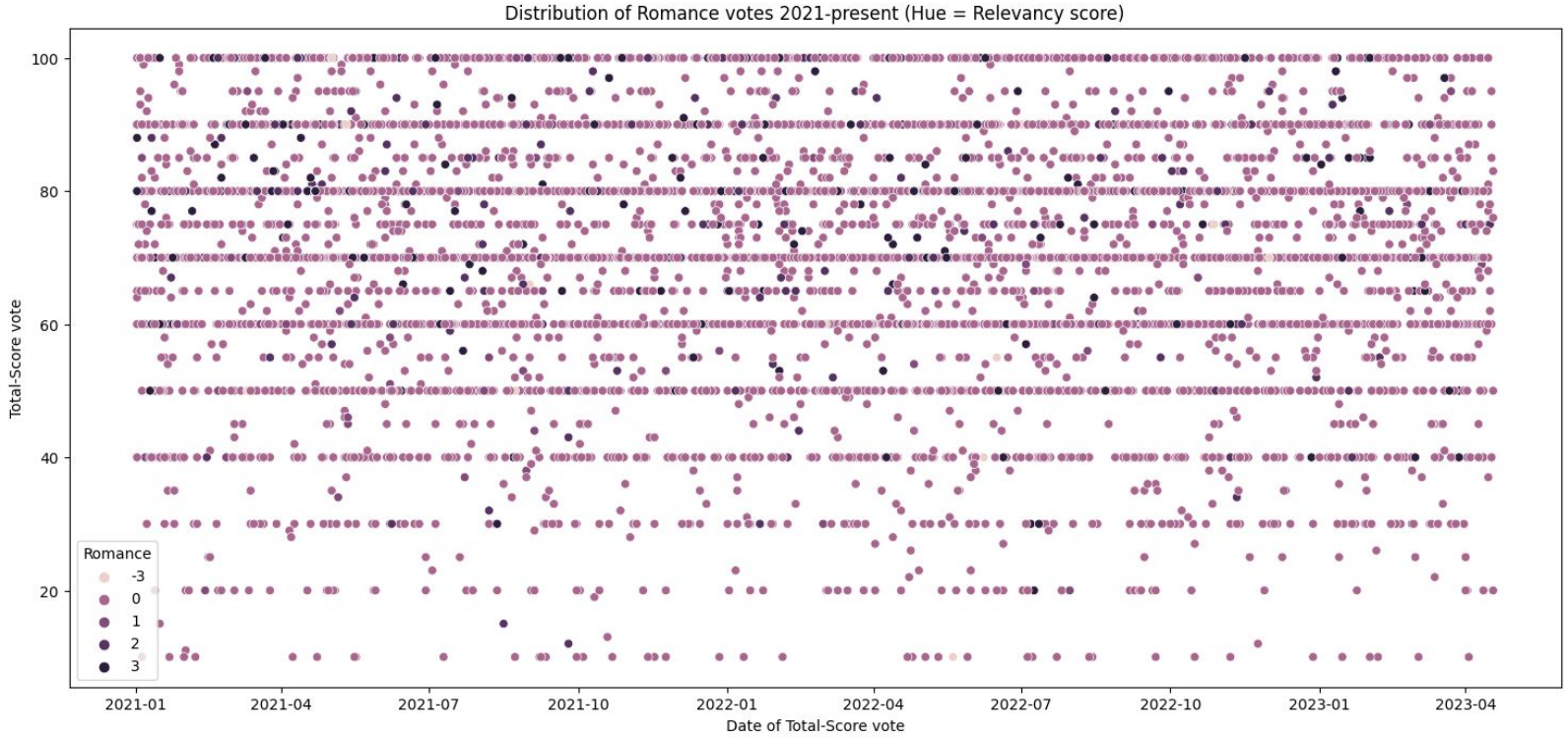
- Used a Standard Scaler (removing the mean and scaling to unit variance) on X.

- Used LASSO regression with 5-fold cross validation to find the best value of alpha, and then fit a final LASSO model using that value of alpha to the data.

- Extracted selected features (using the absolute value of the coefficients) from X.

Model + Method

LASSO:



Model + Method

NEURAL NET:

- Split X (keeping only those non-zero features from LASSO regularization) and y into training and test sets and performed standard scaling on them.

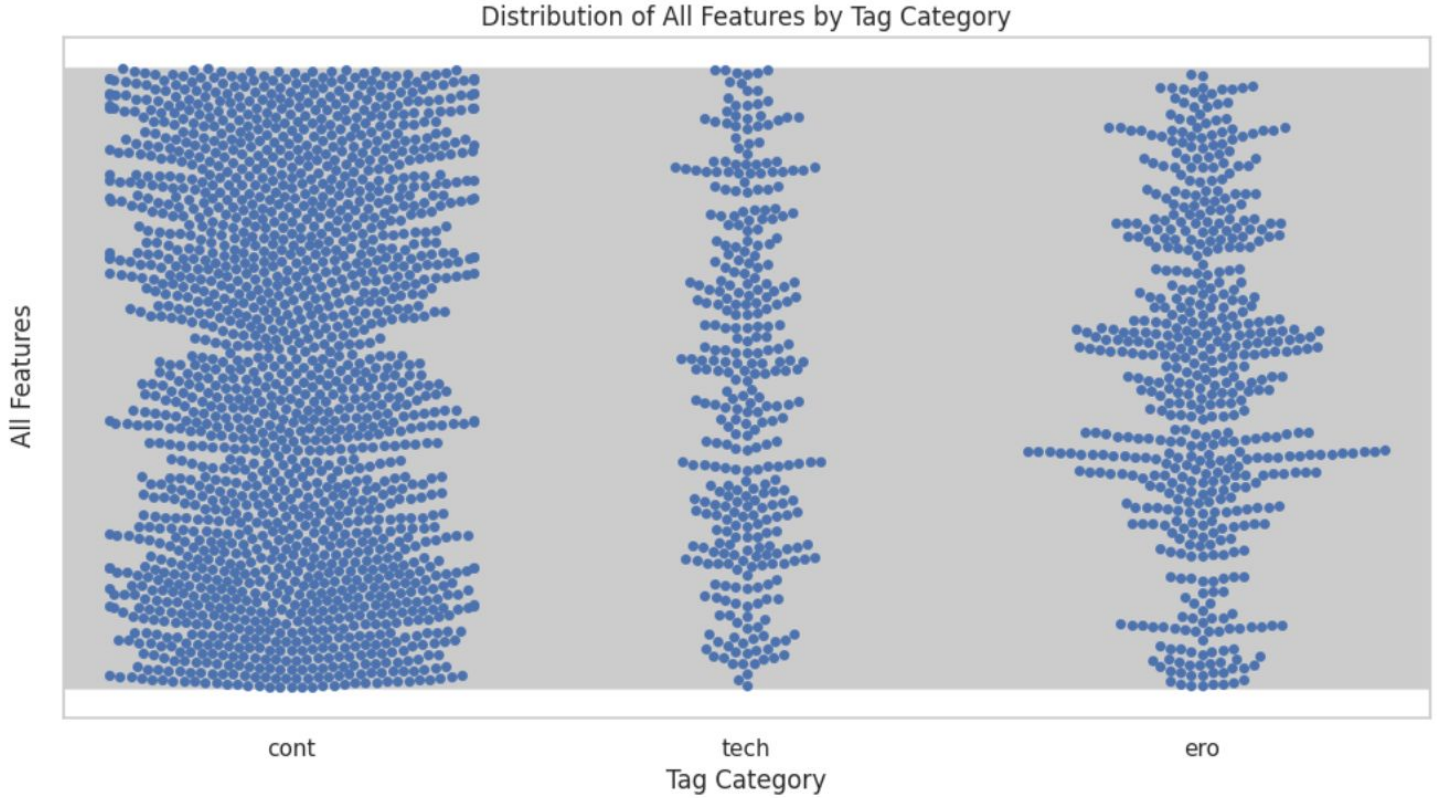
- A simple neural net (see right) was fit onto the training set with `epochs, batch-size = 50, 32`

```
model = Sequential()
model.add(Dense(128, activation='relu',
                input_shape=(X_train_scaled.shape[1],)))
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='linear'))

learning_rate = 0.001
optimizer = Adam(learning_rate=learning_rate)
model.compile(loss='mean_squared_error',
              optimizer=optimizer,
              metrics=['mean_squared_error'])
```

Results

LASSO:

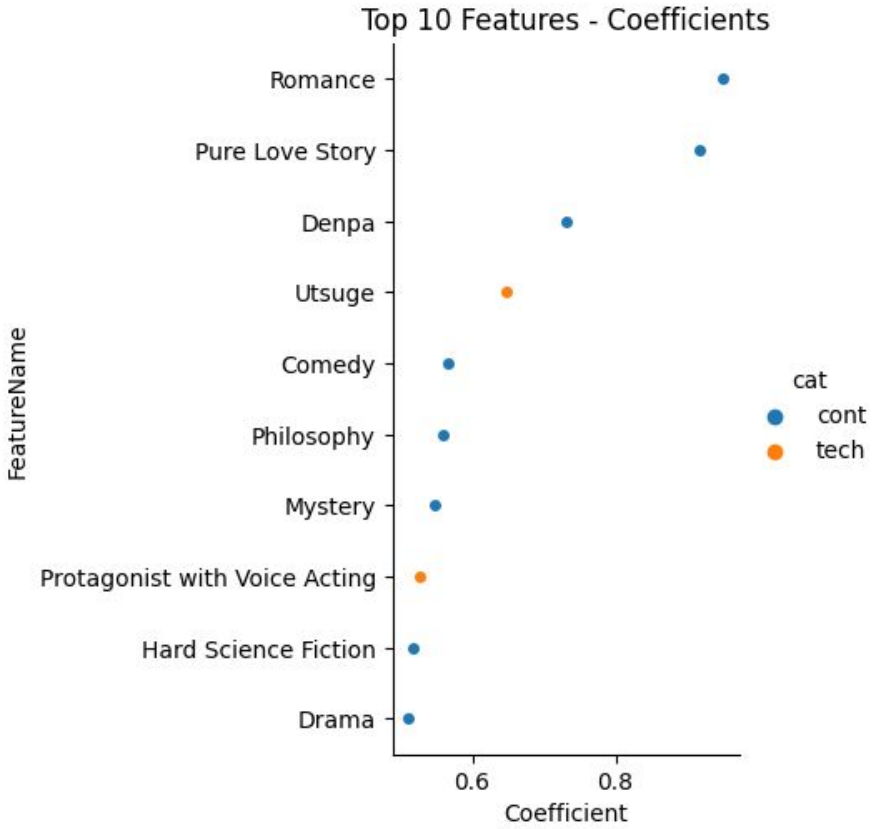


Results

LASSO:

Results

LASSO:




```
best_alpha = 0.020239087171219486
```

Results

NEURAL NET:

- Though training error appeared to converge, test error was `1.0289033651351929`, invalidating the ability of the model to be a reliable predictor of score. Every iteratively tested combination of learning rate, epoch count, and batch size performed just as badly if not worse.

Conclusion

- LASSO Feature selection removed %14 of the tags, and identifying the most influential coefficients improved the interpretability of the data.

- Failure of the Neural Net showed that even with feature selection, data may be too complex in its current form to be able to glean a meaningful prediction from.

Assessment

Limitations:

• Though the LASSO regularization may have helped to narrow down the number of important features (≈2,500 -> ≈2,100), this number may still be too high given the relatively small number of user votes (≈35,000). This complexity may be the reason for the poor performance of the neural net.

• Rigid categorical nature of the tag scoring system may be getting overlooked, as this project treats it as numerical data.

• A core non-technical limitation - The VNDB.org data set is highly biased towards users that are interested enough in visual novel games to make an account on a forum specifically for them, which may not align with a more general audience.

Thanks!

Acknowledgements

Dataset sourced from the vndb.org publicly available database dump.

https://vndb.org/d14