katmandoone / dsc-phase-3-project

forked from learn-co-curriculum/dsc-phase-3-project

0 stars 20 forks

Star Watch

Code Pull requests Actions Projects Wiki Security Insights Settings

main

This branch is 7 commits ahead, 1 commit behind learn-co-curriculum:main. Pull request Compare

katmandoone update … 23 seconds ago 18

View code

README.md

Predicting ESRB Ratings for Video Games

Introduction

The purpose of this project is to predict game ratings for PlayStation4, Xbox One, and Nintendo Switch games. The game data is gathered directly from the esrb.org website. By looking at the descriptors given to a game, I hope to be able to use a classification model to predict the given rating. I believe that this model will help to determine whether game ratings are 'fair,' or if there is some degree of ambiguity when a consumer consults the rating tag before deciding on a game purchase. In the modeling, I will look to maximize recall for M-rated games as well as accuracy across the board.

EVERYONE 10+
TM
Cartoon Violence
Crude Humor
10+
ESRB CONTENT RATING www.esrb.org
Online Interactions not
Rated by the ESRB

> Looking at the image above, we can see an example of and ESRB rating and its descriptors.

Obtaining Data

The data for this project was obtained from esrb.org using the Selenium library. I originally attempted to gather the data with BeautifulSoup as I was more familiar with that library, but I ran into some issues with the ESRB site requiring client-side loading of assets with Java. BeautifulSoup was not equipped to handle this, but Selenium allows for client-side web scraping.

The process for scraping the data is detailed in the data_gathering.ipynb notebook within this repository.

After scraping the data, everything was compiled into a pandas DataFrame

	title	consoles	descriptors	rating
0	Blizzard Arcade Collection	[PlayStation 4, Nintendo Switch, Xbox One]	[Blood, Fantasy Violence, Language, Use of Tob...	T
1	Rez Infinite	[PlayStation 4]	[Fantasy Violence]	E10plus
2	Hotshot Racing	[PlayStation 4, Nintendo Switch]	[Alcohol Reference, Language, Mild Violence]	E10plus
3	Sea of Solitude : The Director's Cut	[Nintendo Switch]	[Fantasy Violence, Language]	T
4	Ape Out	[Nintendo Switch]	[Blood and Gore, Violence]	T

Scrubbing the Data

There really wasn't much scrubbing involved with this data. I just needed to pull the values out of the lists in the descriptors column so that I could one-hot encode each descriptor as a categorical variable.

```
descriptors_ohe = pd.get_dummies(df.descriptors.apply(pd.Series).stack()).sum(level=0)
df_ohe = pd.concat([df.drop(columns=['descriptors']), descriptors_ohe], axis=1)
```

	title	consoles	rating	Alcohol Reference	Alcohol and Tobacco Reference	Animated Blood	Animated Blood and Gore	Animated Violence	Blood	Blood and Gore	...
0	Blizzard Arcade Collection	[PlayStation 4, Nintendo Switch, Xbox One]	T	0	0	0	0	0	1	0	...
1	Rez Infinite	[PlayStation 4]	E10plus	0	0	0	0	0	0	0	...
2	Hotshot Racing	[PlayStation 4, Nintendo Switch]	E10plus	1	0	0	0	0	0	0	...
3	Sea of Solitude : The Director's Cut	[Nintendo Switch]	T	0	0	0	0	0	0	0	...
4	Ape Out	[Nintendo Switch]	T	0	0	0	0	0	0	1	...

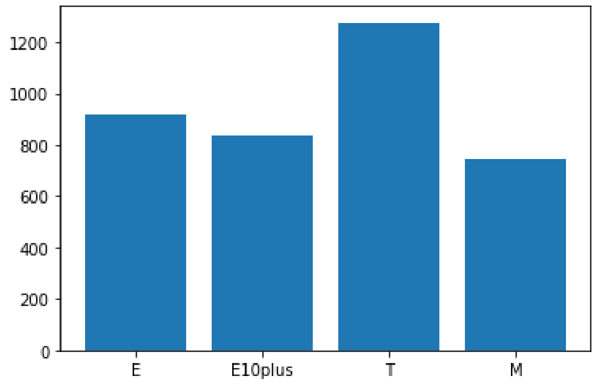
Then we rearranged the new DataFrame into a more intuitive order.

```
cols = list(df_ohe)
rating_col = cols.pop(2)
cols.append(rating_col)
df_ohe = df_ohe.reindex(columns=cols)
```

	title	consoles	Alcohol Reference	Alcohol and Tobacco Reference	Animated Blood	Animated Blood and Gore	Animated Violence	Blood	Blood and Gore	Cartoon Violence	...
0	Blizzard Arcade Collection	[PlayStation 4, Nintendo Switch, Xbox One]	0	0	0	0	0	1	0	0	...
1	Rez Infinite	[PlayStation 4]	0	0	0	0	0	0	0	0	...
2	Hotshot Racing	[PlayStation 4, Nintendo Switch]	1	0	0	0	0	0	0	0	...
3	Sea of Solitude : The Director's Cut	[Nintendo Switch]	0	0	0	0	0	0	0	0	...
4	Ape Out	[Nintendo Switch]	0	0	0	0	0	0	1	0	...

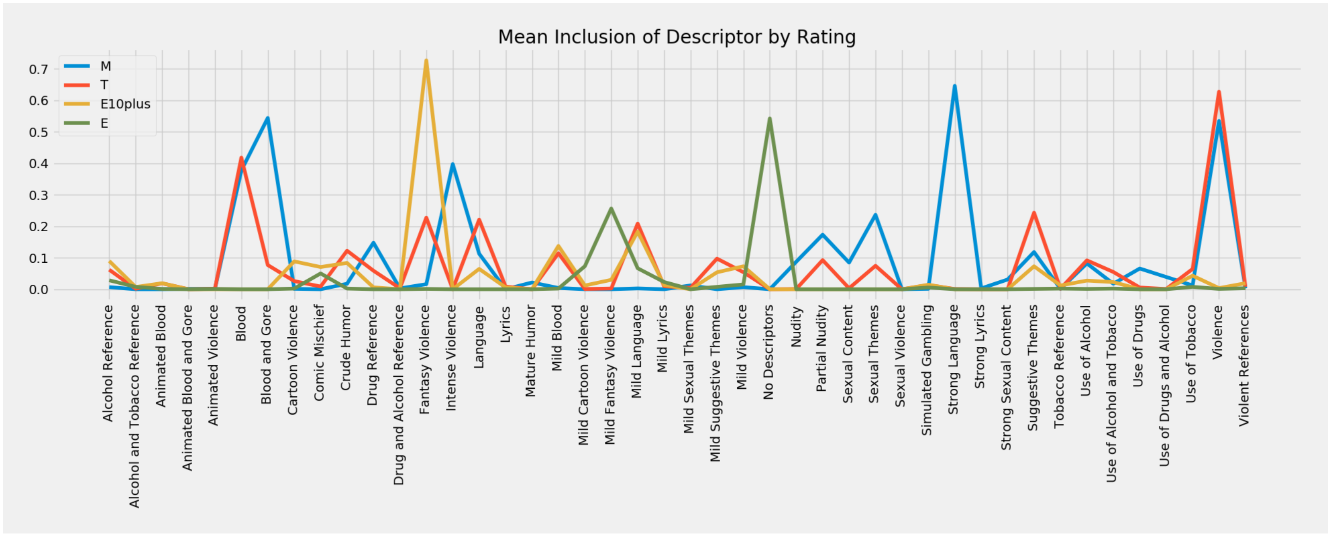
Exploring the Data

After doing a stratified 75/25 train/test split, I checked the balance of the data.



The imbalance wasn't too bad, and apparently SMOTE doesn't support categorical data, so I moved on.

I wanted to know which descriptors seemed most indicative, so I plotted each descriptor against its mean value for each rating. A higher mean value for a single rating in a single descriptor would show that that descriptor is indicative of that rating.



> This graph shows strong indications for 'Blood and Gore', 'Fantasy Violence', 'Intense Violence', 'Mild Fantasy Violence', 'No Descriptors', and 'Strong Language'.

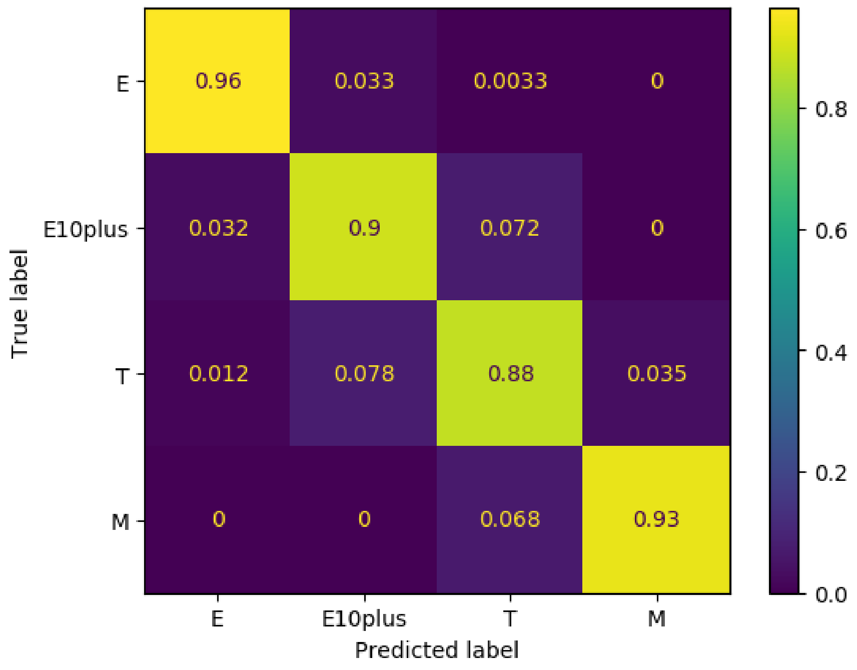
Modeling

I ran the data through a few different classifiers (DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier, and SVC). Some quick scoring checks on those vanilla models indicated that the RandomForestClassifier was the best starting point.

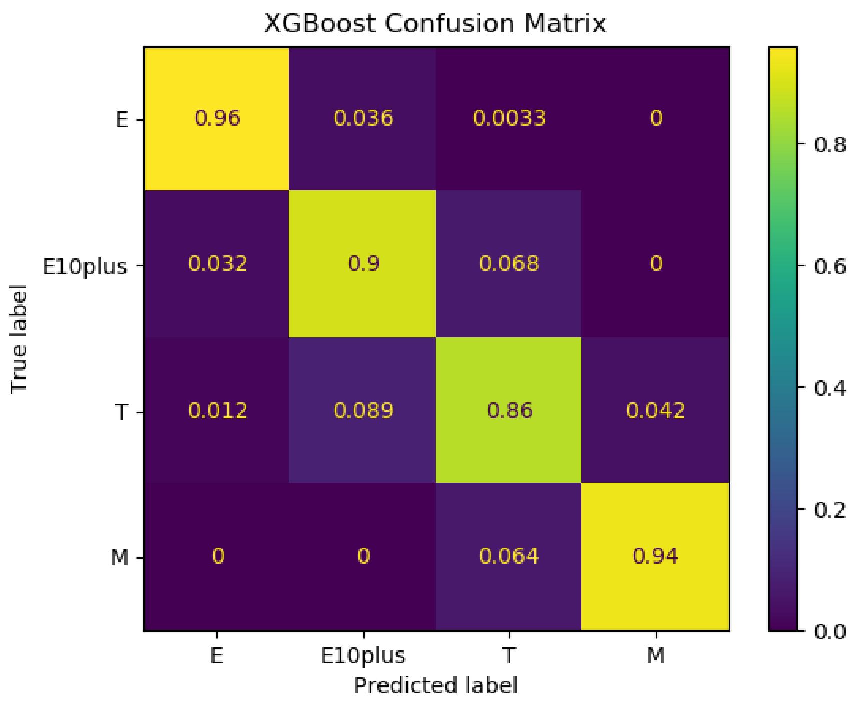
```
DecisionTreeClassifier()
Training: 0.9305960264900662, Test: 0.8983320095313742

RandomForestClassifier()
Training: 0.9305960264900662, Test: 0.9126290706910246

KNeighborsClassifier()
Training: 0.8908609271523179, Test: 0.8721207307386815

SVC()
Training: 0.9189403973509934, Test: 0.9070691024622717
```

```
RandomForestClassifier()
Training Data:
             precision    recall  f1-score   support

           E       0.98      0.97      0.98       917
     E10plus       0.87      0.94      0.90       837
           T       0.93      0.89      0.91      1276
           M       0.95      0.93      0.94       745

    accuracy                           0.93      3775
   macro avg       0.93      0.93      0.93      3775
weighted avg       0.93      0.93      0.93      3775

--------------------------------------------------
Test Data:
             precision    recall  f1-score   support

           E       0.96      0.96      0.96       306
     E10plus       0.84      0.90      0.87       279
           T       0.91      0.88      0.89       425
           M       0.94      0.93      0.94       249

    accuracy                           0.91      1259
   macro avg       0.91      0.92      0.92      1259
weighted avg       0.91      0.91      0.91      1259
```

The vanilla model gives and overall accuracy of 0.91 and a 0.93 recall for M-rated games. I tried a couple variations of GridSearchCV to try to fine-tune the hyperparameters of the Random Forest, but in most cases, the default values were the most effective, and our scores remained unchanged.



To try and get a little bit of a boost in the scores, I switched to the XGBoost Classifier. Its default values were not an imrovement on the Random Forest model, but with a couple tweaks from GridSearchCV, I was able to get a model with 0.91 accuracy and 0.94 recall on the test data.



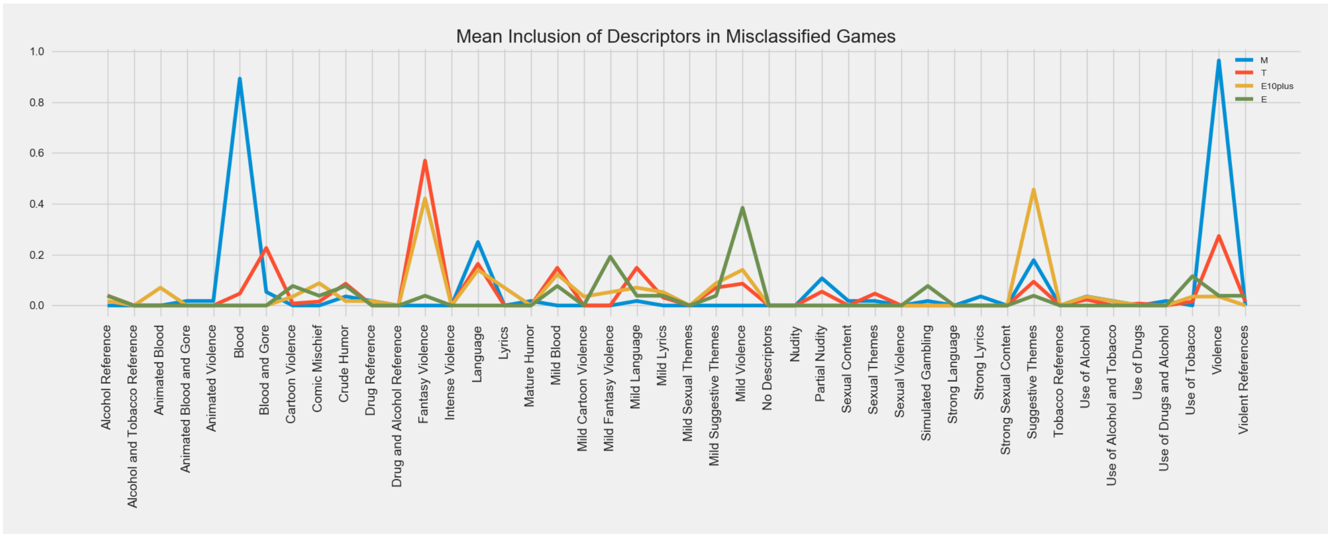
Interpreting Results

```
Best Model: XGBClassifier(learning_rate=0.3, max_depth=7, n_estimators=500,
              objective='multi:softprob')
              precision    recall  f1-score   support

           E       0.95      0.96      0.96       306
     E10plus       0.84      0.90      0.87       279
           T       0.91      0.86      0.88       425
           M       0.93      0.94      0.93       249

    accuracy                           0.91      1259
   macro avg       0.91      0.91      0.91      1259
weighted avg       0.91      0.91      0.91      1259
```

The XGBoost model gave the best scores, but all the tweaking gave minimal gains. What about the data is preventing better classification? I made a new DataFrame of miclassified games to try to find any anomalies.

```
training_with_preds = training_data.copy()
training_with_preds['prediction'] = grid_result.predict(X_train)

wrong_df = training_with_preds[training_with_preds['rating'] !=
                               training_with_preds['prediction']]
```

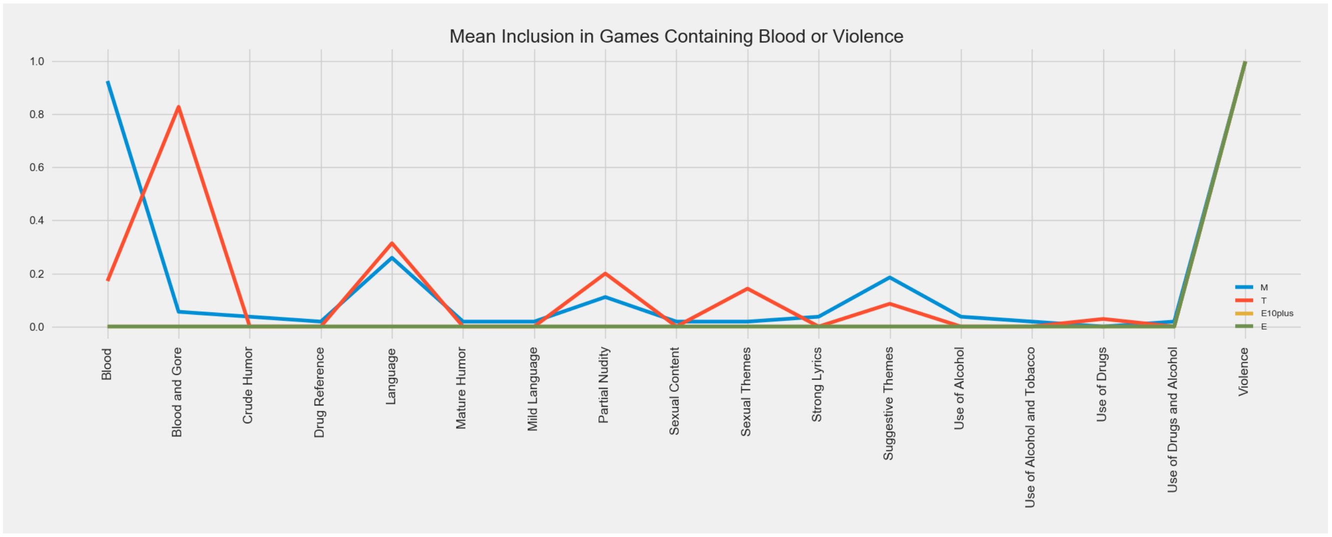
I found that in the training set, 267 games had been misclassified. I plotted the mean inclusion again to see what descriptors were common in misclassified games.



> 80% of misclassified M-rated games have the 'Blood' descriptor, and nearly 100% have the 'Violence' descriptor. Almost half of misclassified E10plus-rated games have the 'Suggestive Themes' descriptor.

This suggests to me that in spite of the ESRB using many different descriptors for games, they use some of them more frequently than they should. This relies on a potential buyer using the rating to determine how extreme the descriptors are rather than using the descriptors to determine how mature the content of them game is.

When looking only at misclassified game containing the 'Blood' or 'Violence' descriptor, the chart showed that most true M-rated games had the 'Blood' descriptor while most true T-rated games had the 'Blood and Gore' descriptor. Taking into account that 100% of these misclassified games had the 'Violence' descriptor, it suggests that the ESRB is applying content warnings and ratings inconsistently. Out of 177 games in the training set where 'Blood' and 'Violence' were the only descriptors present, 152 were correctly identified as being rated T, while 25 M-rated games were misclassified as T-ratings.



Recommendations

After reviewing the results, I believe predictions could be improved if lesser-used descriptors were used more frequently (e.g. 'Intense Violence' or 'Mild Violence' in place of 'Violence' when appropriate). It would also be helpful if more descriptors were used. In the case I presented previously, where both T- and M-rated games had only 'Blood' and 'Violence' descriptors, and extra descriptor or two could act as something of a tie-breaker, helping the model to make a better prediction as well as helping the consumer understand more fully why a game has its given rating.

Future work

In the future, I would like to approach this problem further by training a model to look at box art or screenshots of games to see if that adds a helpful degree of context in making predictions.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● **Jupyter Notebook** 100.0%