GAMEAWARDS

By Doron Tayar and Orel Rafailov

מכון
טכנולוגי חולון
Holon Institute
of Technology

פרוייקט סיום
מדעי הנתונים- שיטות וכלים

הקדמה

- מי לא אוהב משחקים?
- תחום שעבר צמיחה מאוד מהירה

מקורות הנתונים והרכשתן

- https://store.steampowered.com/
- חילוץ כל העמודים של המשחקים

משחק לדוגמה:


Install Steam
login | language
STEAM
STORE COMMUNITY ABOUT SUPPORT
Your Store
New & Noteworthy
Categories
Points Shop
News
Labs
search
All Games > Free to Play Games > PUBG: BATTLEGROUNDS
PUBG: BATTLEGROUNDS
Community Hub
VIKENDI REBORN
DECEMBER 6, 2022
Autoplay videos
2:05 / 2:07
PUBG
BATTLEGROUNDS
Play PUBG: BATTLEGROUNDS for free. Land on strategic locations, loot weapons and supplies, and survive to become the last team standing across various, diverse Battlegrounds. Squad up and join the Battlegrounds for the original Battle Royale experience that only PUBG: BATTLEGROUNDS c...
RECENT REVIEWS: Mixed (15,282)
ALL REVIEWS: Mixed (2,132,374)
RELEASE DATE: 21 Dec, 2017
DEVELOPER: KRAFTON, Inc.
PUBLISHER: KRAFTON, Inc.
Popular user-defined tags for this product:
Survival
Shooter
Battle Royale
Multiplayer
+
Sign in to add this item to your wishlist, follow it, or mark it as ignored

נתונים לדוגמה:

- סוג הביקורות הכללי
- מספר הביקורות
- שם המשחק
- הג'נר
- תאריך השיווק
- האם זכה בפרס או לא
- תמיכה בשפות

TITLE: PUBG: BATTLEGROUNDS
GENRE: Action, Adventure, Free to Play, Massively Multiplayer
DEVELOPER: KRAFTON, Inc.
PUBLISHER: KRAFTON, Inc.
RELEASE DATE: 21 Dec, 2017

RECENT REVIEWS: Mixed (15,282)
ALL REVIEWS: Mixed (2,132,374)
RELEASE DATE: 21 Dec, 2017
DEVELOPER: KRAFTON, Inc.
PUBLISHER: KRAFTON, Inc.
Popular user-defined tags for this product:
Survival | Shooter | Battle Royale | Multiplayer | +

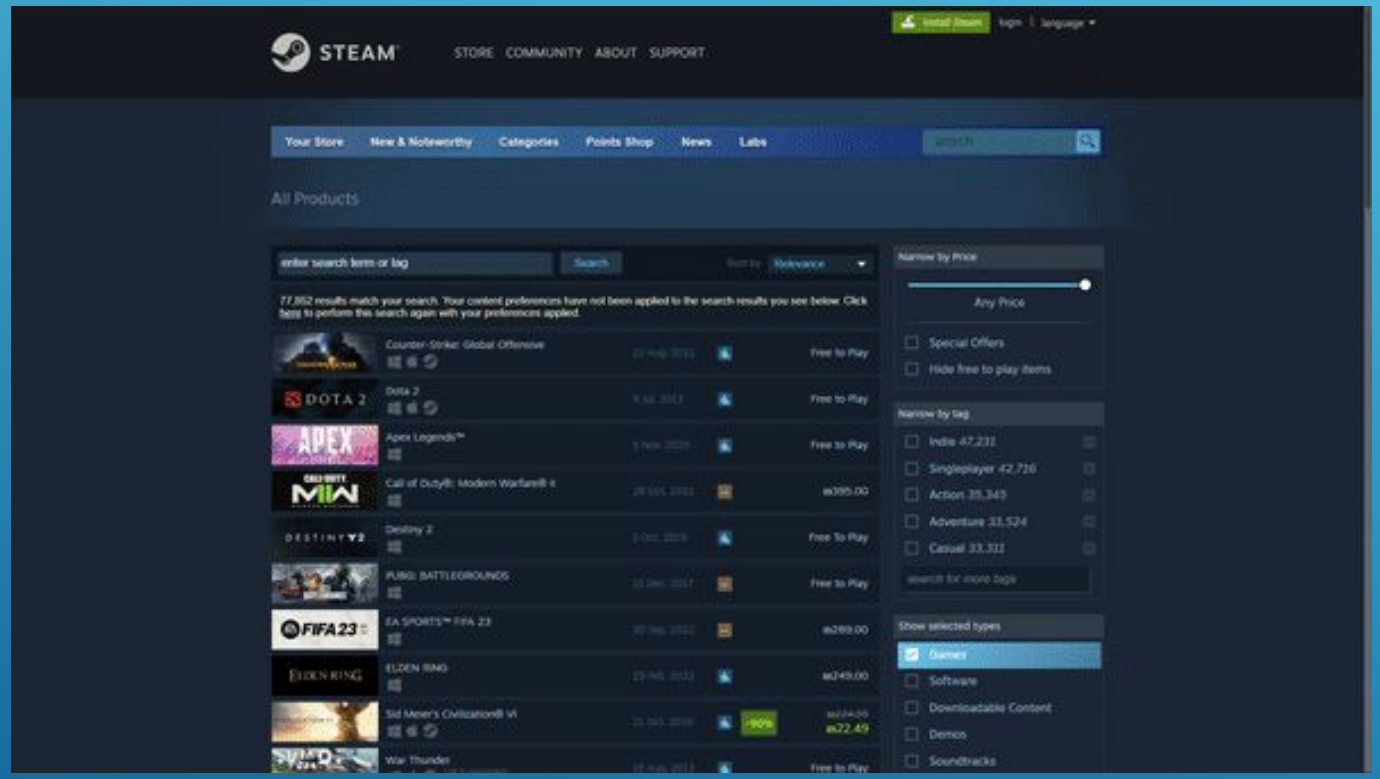
Overall Reviews:
Mixed (2,132,374 reviews)

Languages:

	Interface	Full Audio	Subtitles
English	✓		
Korean	✓		
Simplified Chinese	✓		
French	✓		
German	✓		

Awards

מקורות הנתונים והרכשה



- רוצים להרכיש את כל העמודים

קשיים בהרכשה:
העמוד טוען עוד משחקים רק
כשאנחנו מגיעים לתחתית העמוד.

פתרון:

Selenium

מקורות הנתונים והרכשה

- ישנם משחקים הדורשים אימות גיל

פתרון:

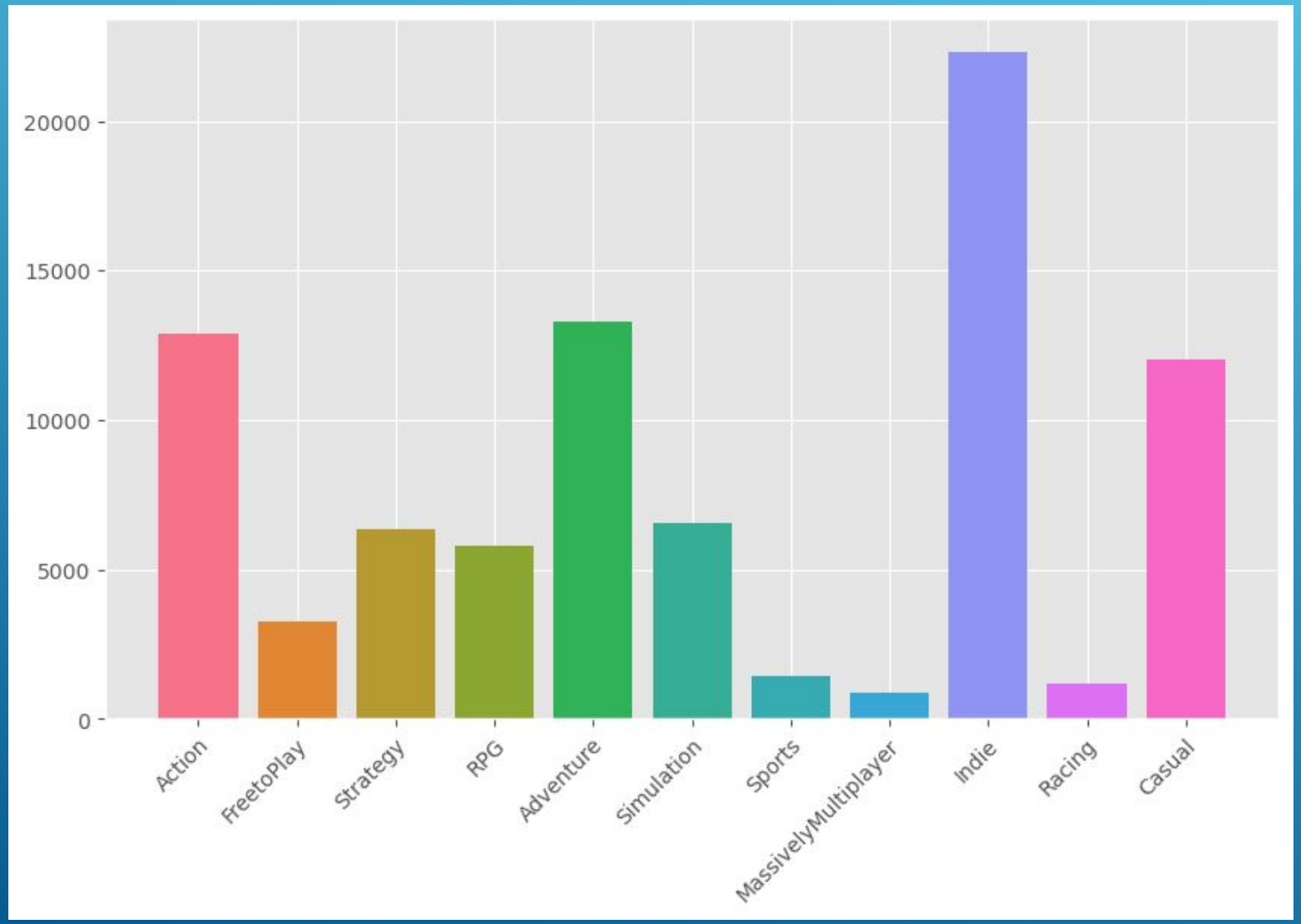
Selenium

ניתוח ראשוני וטיוב

- כל שורה היא משחק
- טיפול בדופליקציות (משחקים שהם Demo והמשחקים המלאים).
- הסרת המשחקים עם מחיר null.
- המרה של חלק מהמשחקים מדולארים לשקלים.
- משחקים עם ג'נרים ריקים מולאו ע"י התגים שהמשתמשים שמו.
- הסרת משחקים עם מפתח null.
- השלמת עמודת publisher עם developer.
- ועוד'.

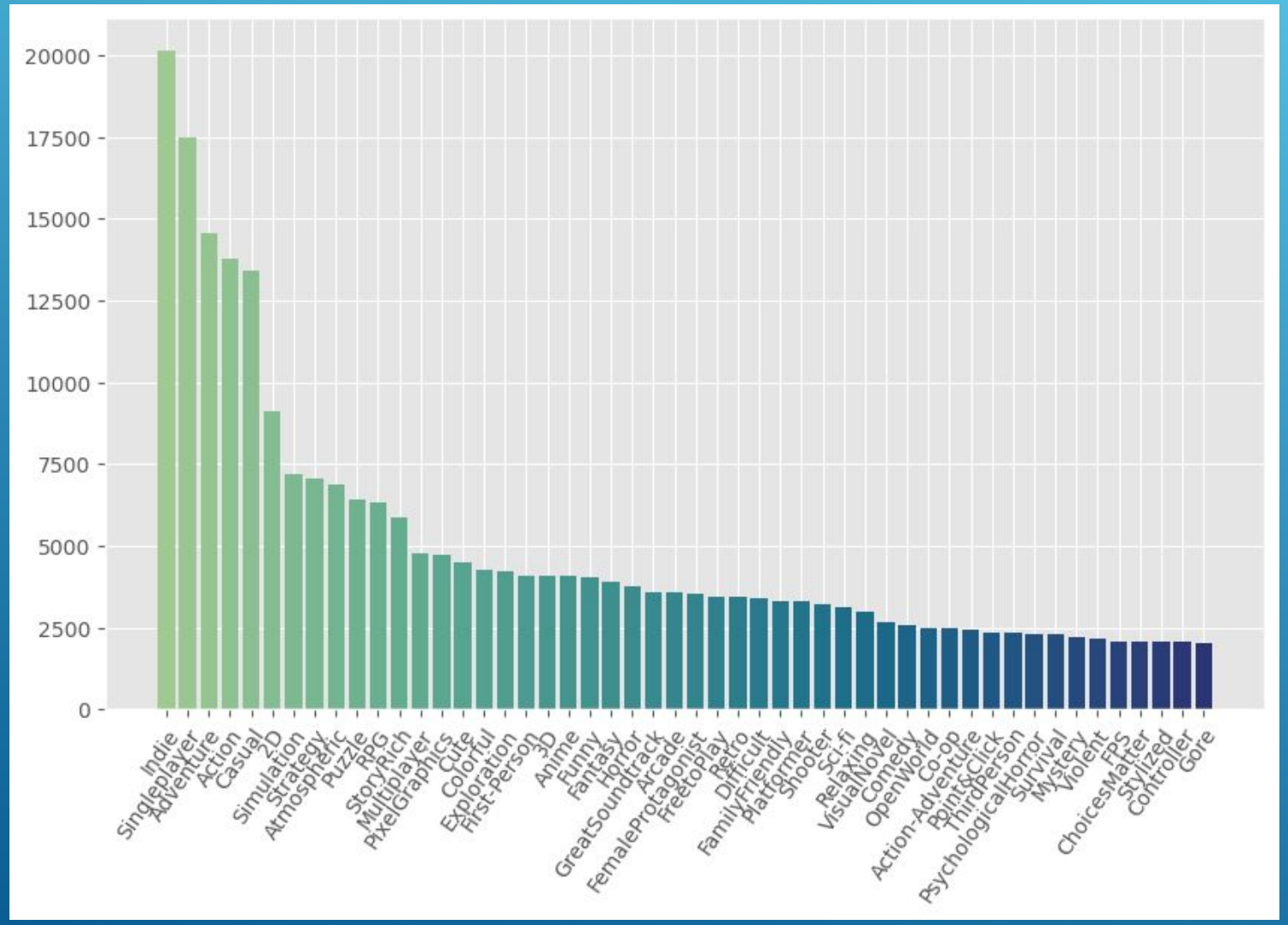
game_name	genres	franchise	developer	publisher	publication_date	user_tags	all_reviews	reviews_type	awards	price	game_features
Counter-Strike: Global Offensive	Action, Free to Play	NaN	Valve, Hidden Path Entertainment	Valve	21 Aug, 2012	['FPS', 'Shooter', 'Multiplayer', 'Competitive...	6833486.0	Very Positive	1.0	0	['Steam Achievements', 'Full controller suppor...
Dota 2	Action, Free to Play, Strategy	Dota	Valve	Valve	9 Jul, 2013	['Free to Play', 'MOBA', 'Multiplayer', 'Strat...	1903396.0	Very Positive	0.0	0	['Steam Trading Cards', 'Steam Workshop', 'Ste...
ELDEN RING	Action, RPG	Bandai Namco Entertainment	FromSoftware Inc.	FromSoftware Inc., Bandai Namco Entertainment	25 Feb, 2022	['Souls-like', 'Dark Fantasy', 'RPG', 'Open Wo...	457621.0	Very Positive	1.0	₪249.00	['Single-player', 'Online PvP', 'Online Co-op'...
Apex Legends™	Action, Adventure, Free to Play	Apex Legends	Respawn Entertainment	Electronic Arts	4 Nov, 2020	['Free to Play', 'Multiplayer', 'Battle Royale...	559929.0	Very Positive	1.0	0	['Online PvP', 'Online Co-op', 'Steam Achievem...
Call of Duty®: Modern Warfare® II	Action	Call of Duty	Infinity Ward, Raven Software, Beenox, Treyarc...	Activision	28 Oct, 2022	['FPS', 'Action', 'Shooter', 'Multiplayer', 'M...	160623.0	Mixed	0.0	₪395.00	['Single-player', 'Online PvP', 'Online Co-op'...

ויזואליזציה וEDA



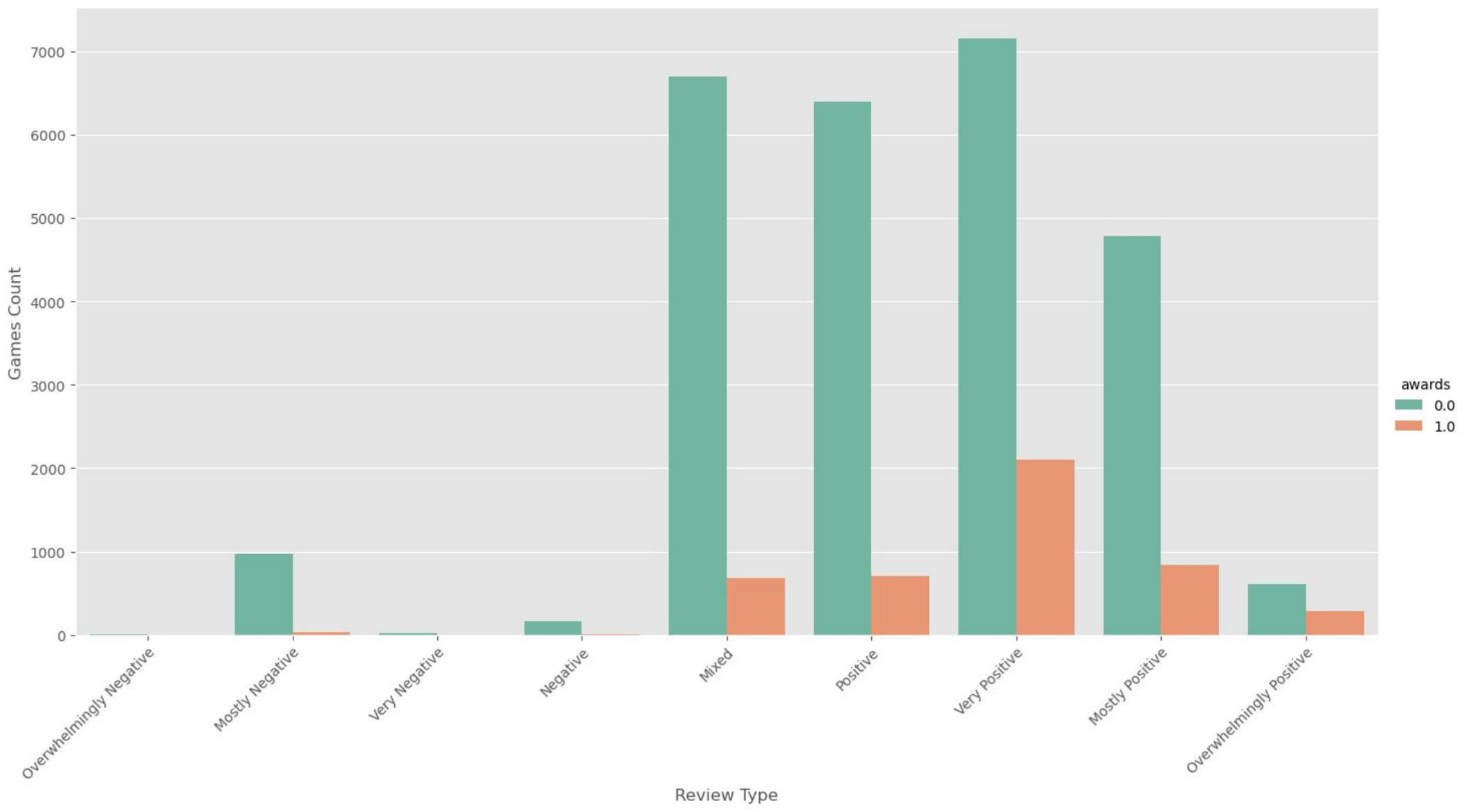
הערה:משחק יכול להכיל יותר מג'אנר אחד

ויזואליזציה ו EDA

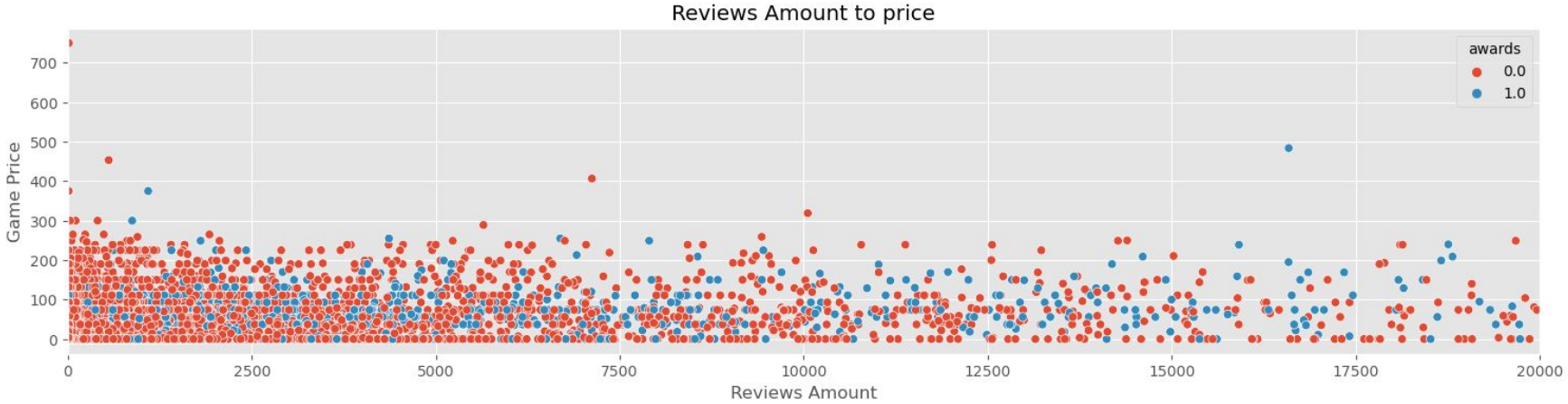


הערה: משחק יכול להכיל
יותר מתג אחד

Review Type to Awards




Reviews Amount to price
Game Price
Reviews Amount
1e6
awards
0.0
1.0


Reviews Amount to price
Game Price
Reviews Amount
awards
0.0
1.0


Reviews Amount to price
Game Price
Reviews Amount
awards
0.0
1.0

now we can see that there are more games that won awards with more then 5k reviews compared to those with below 5k reviews

In [73]:
```
print("amount of games below 5k reviews:{}".format(len(gCopy[gCopy["all_reviews"]<5000])))
print("amount of games below 5k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]<5000][gCopy[gCopy["all_reviews

print("amount of games below 3k reviews:{}".format(len(gCopy[gCopy["all_reviews"]<3000])))
print("amount of games below 3k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]<3000][gCopy[gCopy["all_reviews

print("amount of games below 2k reviews:{}".format(len(gCopy[gCopy["all_reviews"]<2000])))
print("amount of games below 2k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]<2000][gCopy[gCopy["all_reviews

print("amount of games below 1k reviews:{}".format(len(gCopy[gCopy["all_reviews"]<1000])))
print("amount of games below 1k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]<1000][gCopy[gCopy["all_reviews

print("amount of games with more then 5k reviews:{}".format(len(gCopy[gCopy["all_reviews"]>5000])))
print("amount of games with more then 5k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]>5000][gCopy[gCopy["al

print("amount of games with more then 10k reviews:{}".format(len(gCopy[gCopy["all_reviews"]>10000])))
print("amount of games with more then 10k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]>10000][gCopy[gCopy["

print("amount of games with more then 20k reviews:{}".format(len(gCopy[gCopy["all_reviews"]>20000])))
print("amount of games with more then 20k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]>20000][gCopy[gCopy["

print("amount of games with more then 40k reviews:{}".format(len(gCopy[gCopy["all_reviews"]>40000])))
print("amount of games with more then 40k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]>40000][gCopy[gCopy["

print("amount of games with more then 100k reviews:{}".format(len(gCopy[gCopy["all_reviews"]>100000])))
print("amount of games with more then 100k reviews with awards:{}".format(len(gCopy[gCopy["all_reviews"]>100000][gCopy[gCopy
```

```
amount of games below 5k reviews:29879
amount of games below 5k reviews with awards:4101
amount of games below 3k reviews:29277
amount of games below 3k reviews with awards:3935
amount of games below 2k reviews:28642
amount of games below 2k reviews with awards:3751
amount of games below 1k reviews:27226
amount of games below 1k reviews with awards:3408
amount of games with more then 5k reviews:1574
amount of games with more then 5k reviews with awards:548
amount of games with more then 10k reviews:937
amount of games with more then 10k reviews with awards:365
amount of games with more then 20k reviews:529
amount of games with more then 20k reviews with awards:226
amount of games with more then 40k reviews:291
amount of games with more then 40k reviews with awards:136
amount of games with more then 100k reviews:105
amount of games with more then 100k reviews with awards:48
```

As we can see below 5k reviews the percentage of award winning games compared to the rest of the games are somewhere close to 14%.
As we can see above 5k reviews the percentage of award winning games compared to the rest of the games are somewhere close to 35%.
As we can see above 100k reviews the percentage of award winning games compared to the rest of the games are somewhere close to 46%.
There might be a connection between having more reviews and winning awards?

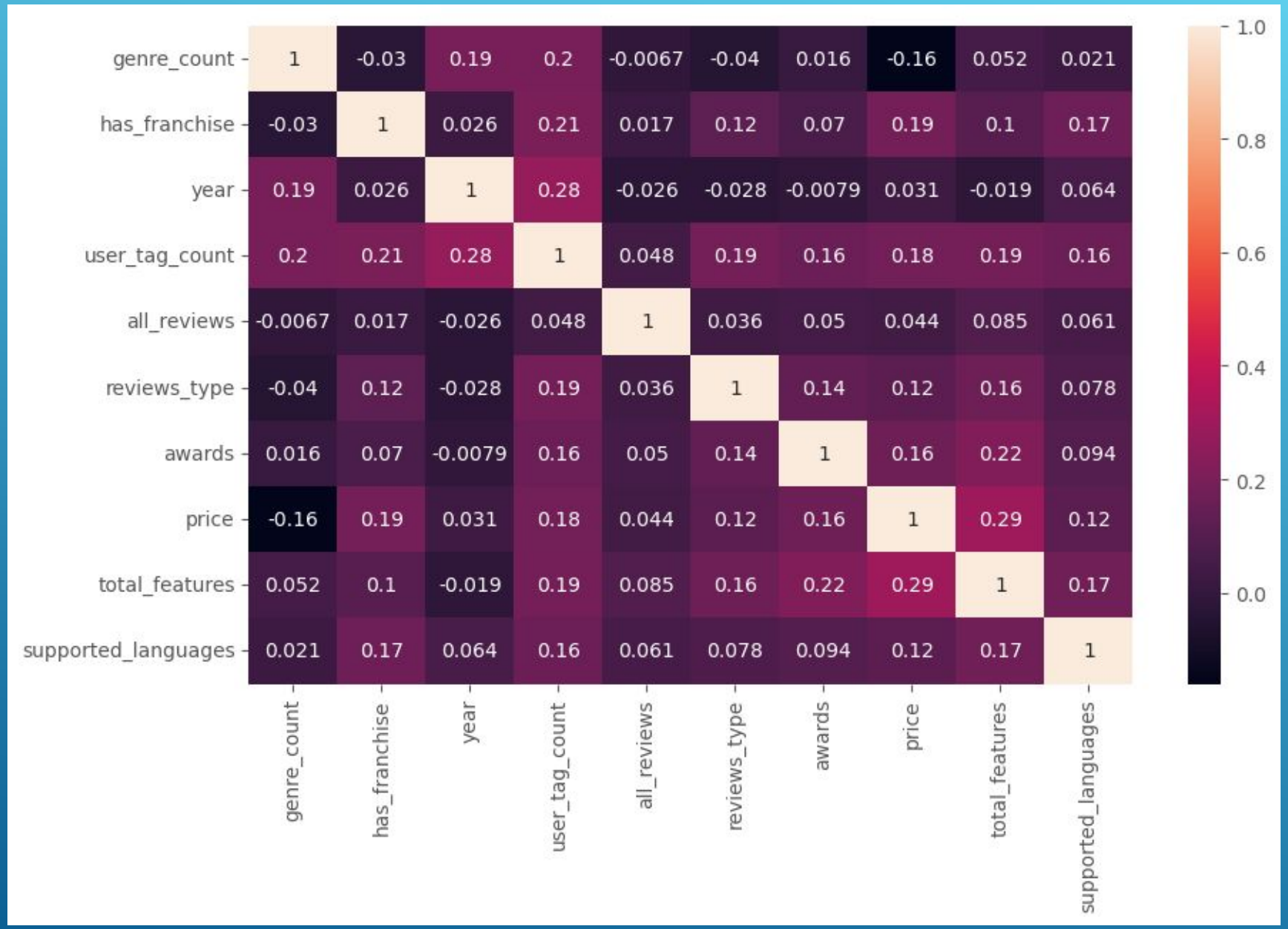
לפני ניתוח נתונים מתקדם עשינו כמה דברים :

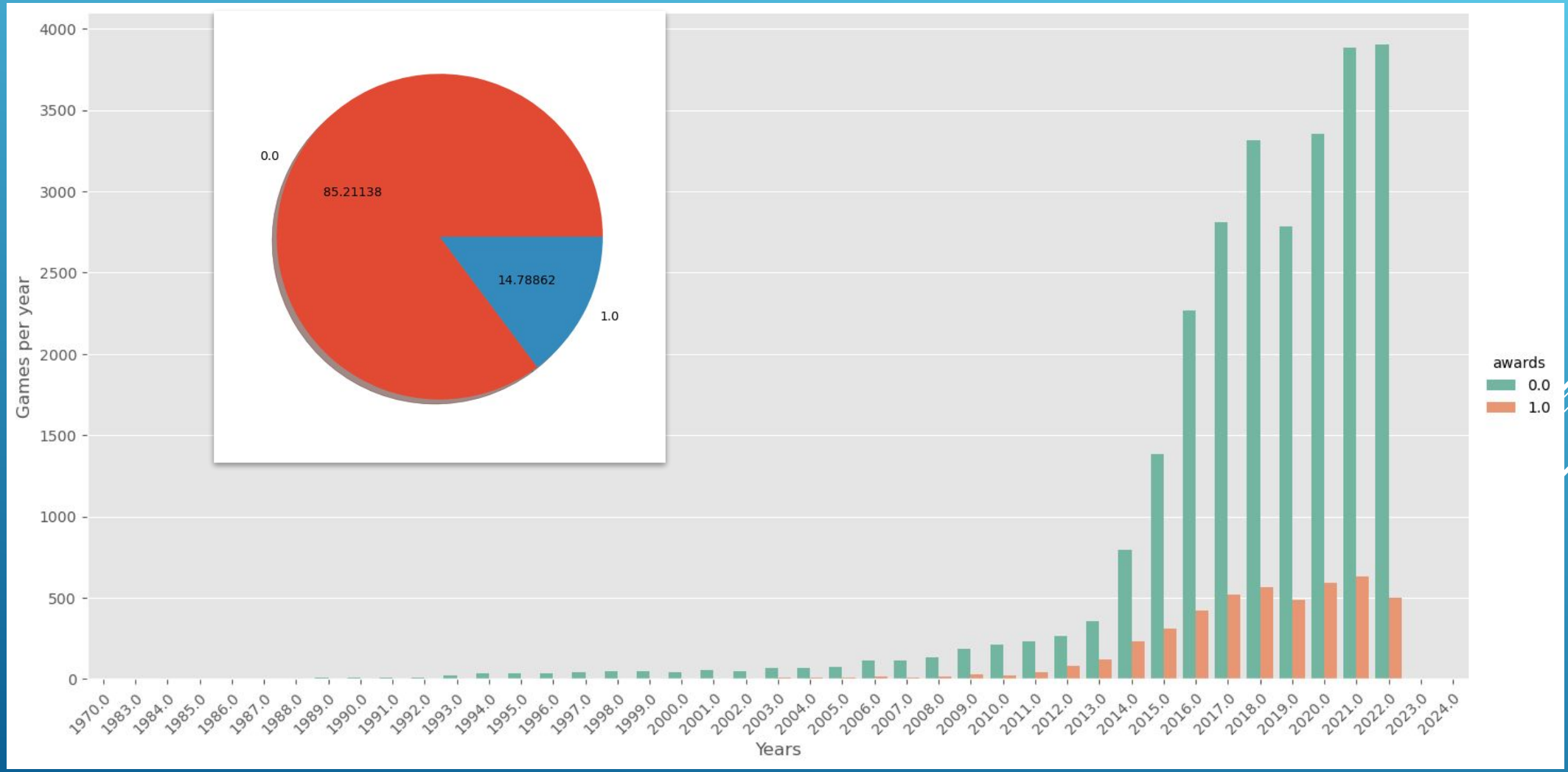
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31389 entries, 0 to 31388
Data columns (total 22 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   game_name           31389 non-null  object
 1   genres              31389 non-null  object
 2   genre_count         31389 non-null  int64
 3   franchise           31389 non-null  object
 4   has_franchise       31389 non-null  int64
 5   developer           31389 non-null  object
 6   publisher           31389 non-null  object
 7   day_month           31389 non-null  object
 8   year                31389 non-null  float64
 9   popular_tag         31389 non-null  object
 10  user_tag_count      31389 non-null  int64
 11  user_tags           31389 non-null  object
 12  all_reviews         31389 non-null  float64
 13  reviews_type        31389 non-null  int64
 14  awards              31389 non-null  float64
 15  price               31389 non-null  float64
 16  game_features       31389 non-null  object
 17  total_features      31389 non-null  int64
 18  languages           31389 non-null  object
 19  english_support     31389 non-null  int64
 20  supported_languages 31389 non-null  int64
 21  os_compatibility    31389 non-null  object
dtypes: float64(4), int64(7), object(11)
memory usage: 5.3+ MB
```

עמודת שבהן הנתונים היו קטגוריאלים ו\או מרובי משתנים
הפכנו לנומרים והוספנו אותן כעמודות חדשות:
has_franchise - אם יש לו franchise אז 1 אחרת 0
user_tag_count - כמות הuser_tags שיש למשחק
total_features - כמות ה פיצ'רים שיש למשחק
supported_languages - בכמה שפות המשחק תומך

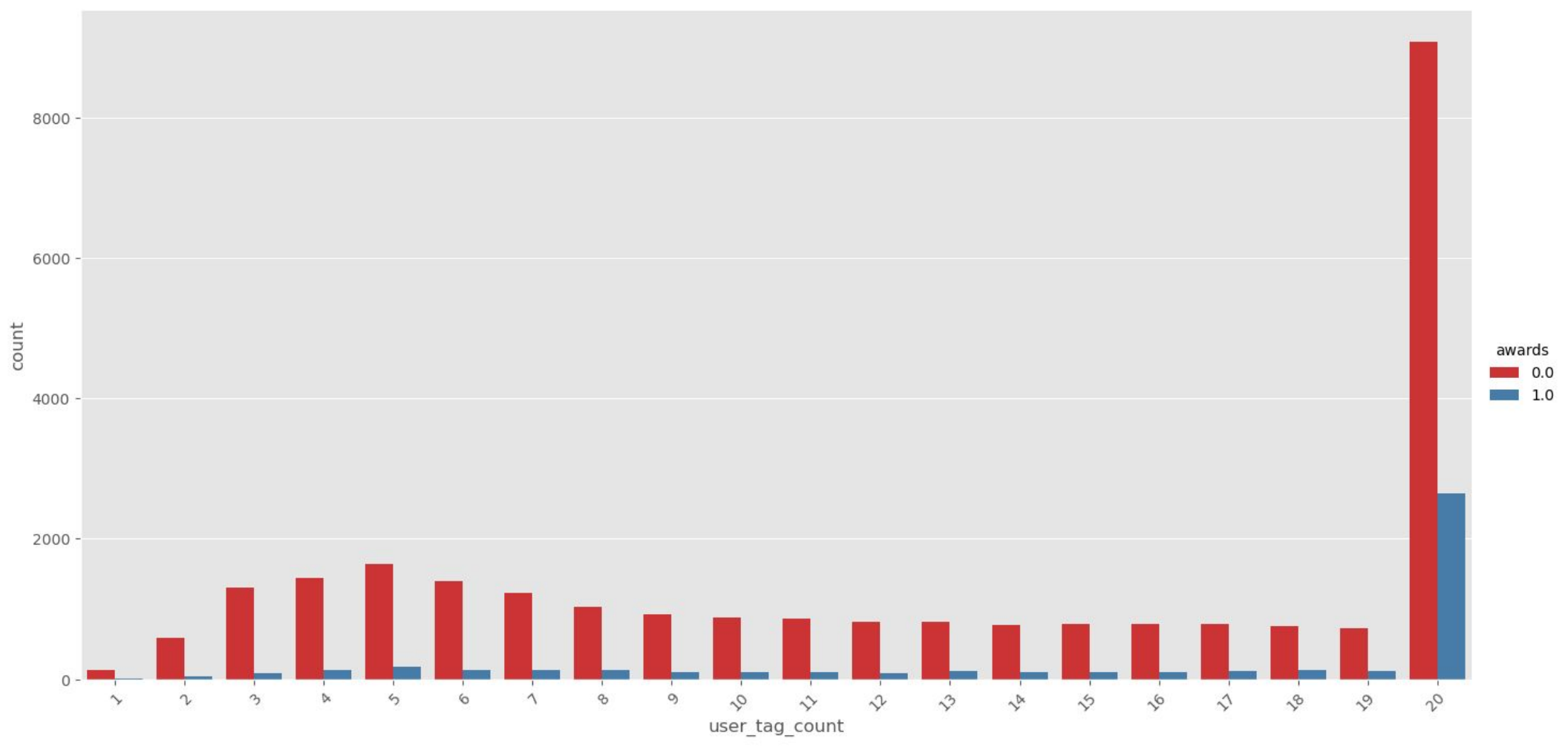
עמודות שהיות צריכים לסדר\ לפרק את הנתונים:
day_month - יום+חודש של פרסום המשחק
year - שנה פרסום של המשחק
popular_tag - הuser_tag הפופולרי ביותר של המשחק

ניתוח נתונים מתקדם:




0.0
85.21138
14.78862
1.0
Games per year
Years
awards
0.0
1.0

ניתוח נתונים מתקדם:



בחירת למידת המכונה: DECISION TREES

מודל למידת המכונה שבחרנו הוא עץ החלטה. ראינו לנכון לבחור במודל זה מהסיבות הבאות

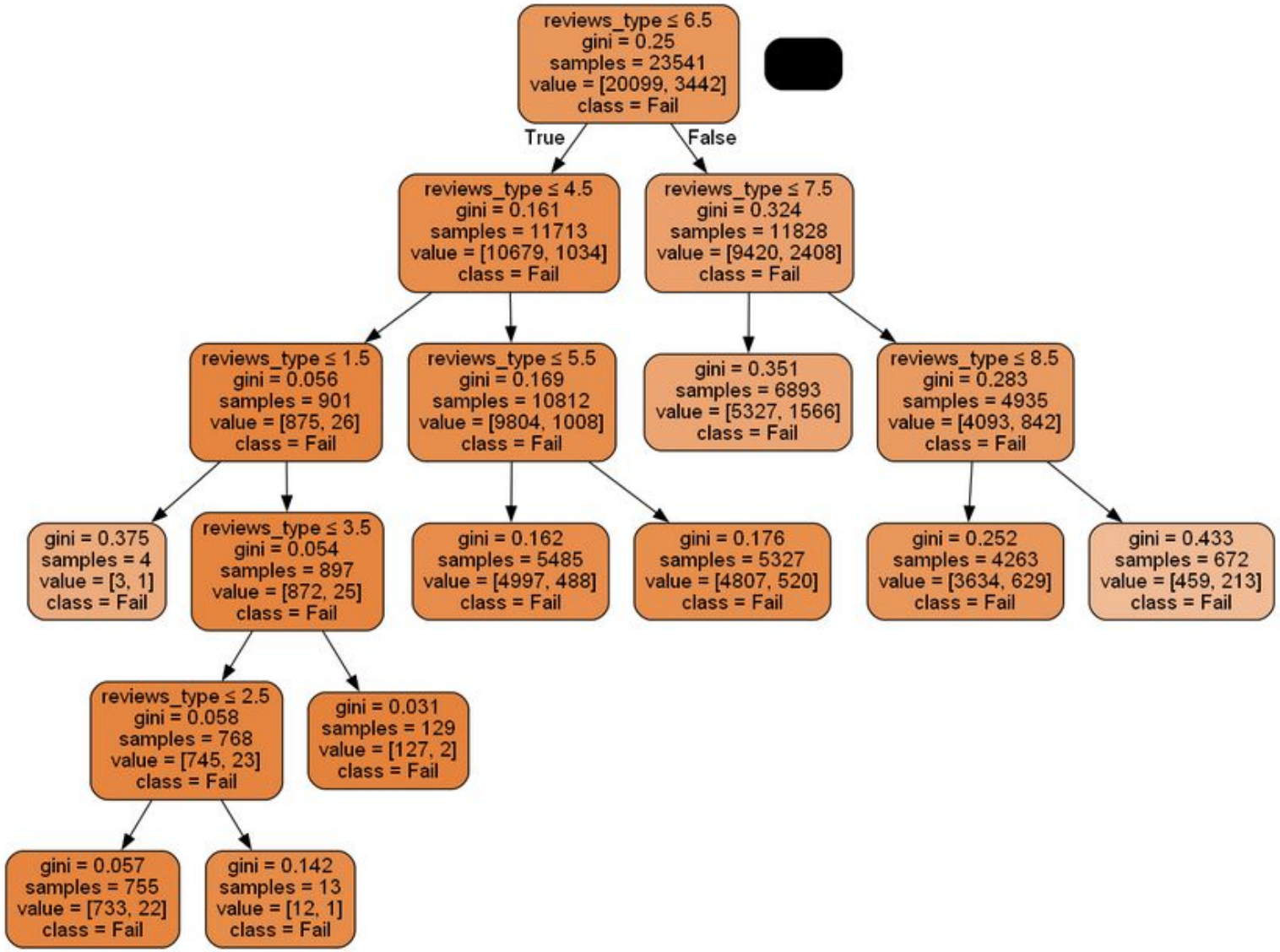
זה מודל קלאסיפיקציה

נוכל להשתמש בהמשך ב
Random forest
לביצוע Cross Validation

הוא פשוט יחסית ואינטואיטיבי

First Model: Relying only on Review Type to Predict Awards

```
Accuracy on training data =  0.8537870098976255
Accuracy on test data =  0.8470948012232415
```



Relying on more features to Predict Awards

```
Accuracy on training data =  0.8685272503292129
Accuracy on test data =  0.8412334352701325
f_measure score=  0.2420924574209246
dot: graph is too large for cairo-renderer bitmaps. Scaling by 0.817948 to fit
```

```
features = ["genre_count"
            ,"has_franchise"
            ,"year"
            ,"popular_tag"
            ,"user_tag_count"
            ,"all_reviews"
            ,"reviews_type"
            ,"total_features"
            ,"supported_languages"
            ,"price"]
```

Creating a new column to improve the model

```
features = ["genre_count"
            ,"has_franchise"
            ,"year"
            ,"popular_tag"
            ,"user_tag_count"
            ,"all_reviews"
            ,"reviews_type"
            ,"total_features"
            ,"supported_languages"
            ,"price"
            ,"game_developer_awards_count"]
```

הוספנו עמודה שמכילה בתוכה את מספר הפרסים שאותו מפתח זכה בהם בשביל לנסות ולשפר את המודל.

OLD

```
Accuracy on training data =  0.8685272503292129
Accuracy on test data =  0.8412334352701325
f_measure score=  0.2420924574209246
```

```
Accuracy on training data =  0.9228579924387239
Accuracy on test data =  0.9051987767584098
f_measure score=  0.7111801242236024
```

ניסינו גם לעשות שינויים בפרמטרים אחרים כמו עומק מקסימלי ומספר דוגמיות פיצול מינימאלי

We will try changing the depth and the min sample split to try and improve the prediction accuracy.

In [93]:
```
parameters = {'max_depth': range(2, 15), "min_samples_split": range(5, 50) }
dt = tree.DecisionTreeClassifier()

clf = GridSearchCV(dt, parameters, scoring=make_scorer(metrics.accuracy_score, greater_is_better=True))
clf.fit(XTrain, yTrain)

print("best parameter set is:",clf.best_params_," and its score was",clf.best_score_)
```

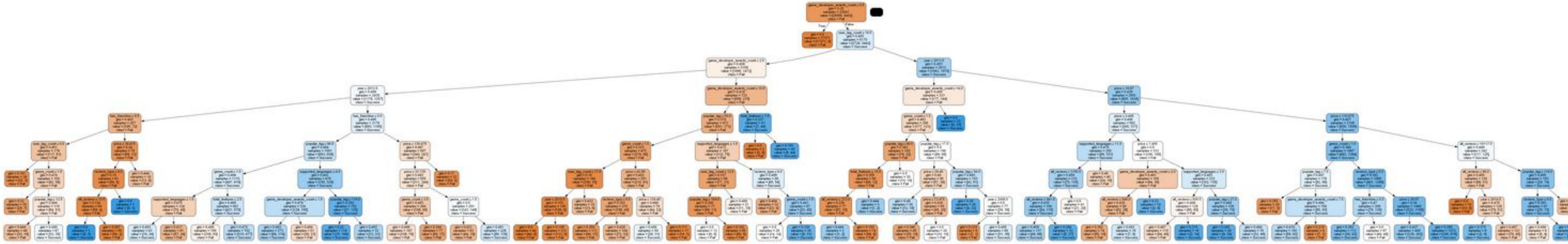
```
best parameter set is: {'max_depth': 8, 'min_samples_split': 20}  and its score was 0.9109636854751102
```

Let's try predicting again with the suggested max depth and see if there's any improvement.

```
Accuracy on training data =  0.9228579924387239
Accuracy on test data =  0.9051987767584098
f_measure score=  0.7111801242236024
```

אבל ראינו שינוי מאוד מזערי

```
Accuracy on training data =  0.9211163501975277
Accuracy on test data =  0.9064729867482161
f_measure score=  0.7135050741608118
```



We can see there is **very** slight improvement

הגענו לaccuracy,f_measure יחסית גבוהים.
האם אפשר להגיע לתוצאות טובות יותר?

מודל: RANDOM FOREST

למה לבחור בrandom forest?

- יש לו accuracy יותר טוב מDecision Tree
- סיכוי נמוך יותר לOverfitting
- נותן משקלים לfeatures ונוח לבחירת features מועילים למודל

מודל: RANDOM FOREST

FIRST RUN

```
#using random forest
features = ["genre_count"
            ,"has_franchise"
            ,"year"
            ,"popular_tag"
            ,"user_tag_count"
            ,"all_reviews"
            ,"reviews_type"
            ,"total_features"
            ,"supported_languages"
            ,"price"
            ,"game_developer_awards_count"]

XTrain, XTest, yTrain, yTest = splitData(processed_data, features, ["awards"],42)

forest =  RandomForestClassifier(bootstrap=True, n_estimators=50, random_state=0)

trained_forest = forest.fit(XTrain, yTrain.ravel())

y_pred_train = trained_forest.predict(XTrain)
print('Accuracy on training data= ', metrics.accuracy_score(y_true = yTrain, y_pred = y_pred_train))

y_pred = trained_forest.predict(XTest)
print('Accuracy on test data= ', metrics.accuracy_score(y_true = yTest, y_pred = y_pred))
print('f_measure score= ',  f1_score(yTest, y_pred))
```

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9119520897043832
f_measure score=  0.7208080808080808
```

```
Accuracy on training data =  0.9228579924387239
Accuracy on test data =  0.9051987767584098
f_measure score=  0.7111801242236024
```

מודל: RANDOM FOREST

ראינו אם אפשר לשפר את המודל ע"י הסרת feature עם המשקל הכי נמוך.

	features	weights
10	game_developer_awards_count	0.465165
5	all_reviews	0.088350
9	price	0.080252
3	popular_tag	0.065792
2	year	0.060076
7	total_features	0.053232
8	supported_languages	0.053053
4	user_tag_count	0.050701
0	genre_count	0.039778
6	reviews_type	0.030144
1	has_franchise	0.013457

מודל: RANDOM FOREST

	features	weights
10	game_developer_awards_count	0.465165
5	all_reviews	0.088350
9	price	0.080252
3	popular_tag	0.065792
2	year	0.060076
7	total_features	0.053232
8	supported_languages	0.053053
4	user_tag_count	0.050701
0	genre_count	0.039778
6	reviews_type	0.030144
1	has_franchise	0.013457

ראינו אם אפשר לשפר את המודל ע"י הסרת feature עם המשקל הכי נמוך.

OLD

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9119520897043832
f_measure score=  0.7208080808080808
```

רואים שAccuracy,f_measure ירדו.
לא נראה שההסרה של feature הועילה במיוחד.

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9087665647298675
f_measure score=  0.7101214574898786
```

we can see that removing the feature with the lowest weight , didnt really improve our model

מודל: RANDOM FOREST

```
revType={
    "OverwhelminglyNegative":1 ,
    'MostlyNegative':2,
    "VeryNegative":3,
    "Negative":4,
    "Mixed":5,
    "Positive":6,
    "VeryPositive":7,
    'MostlyPositive':8,
    "OverwhelminglyPositive":9
}
```

המרנו את review_type לערכים נומריים מההכי גרוע 1 עד ההכי טוב 9.

באמצעות המרה זו, הוספנו עמודה חדשה עם הממוצע של סוג הביקורת של מפתח שעבד על המשחק. קראנו לו developers_by_reviewsType_mean.

מודל: RANDOM FOREST

```
revType={
    "OverwhelminglyNegative":1 ,
    'MostlyNegative':2,
    "VeryNegative":3,
    "Negative":4,
    "Mixed":5,
    "Positive":6,
    "VeryPositive":7,
    'MostlyPositive':8,
    "OverwhelminglyPositive":9
}
```

המרנו את review_type לערכים נומריים מההכי גרוע 1 עד ההכי טוב 9.

באמצעות המרה זו, הוספנו עמודה חדשה עם הממוצע של סוג הביקורת של מפתח שעבד על המשחק. קראנו לו developers_by_reviewsType_mean.

OLD

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9087665647298675
f_measure score=  0.7101214574898786
```

NEW

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9290265035677879
f_measure score=  0.77181482998771
```

הגענו לשיפור של כ5% בf_measure וכ2% בaccuracy

מודל: RANDOM FOREST

ניסינו לשפר את המודל עוד קצת ע"י שינוי פרמטרים:

```
In [155]: # some hyper-params tuning
parameters = {
    'bootstrap':[True],
    'n_estimators':[50, 51, 53,55, 100, 101, 501, 1000],
    'random_state':[0],
    'max_features':['sqrt', 'log2', 'auto']
}
rf = RandomForestClassifier()
clf = GridSearchCV(rf, parameters, scoring=make_scorer(metrics.accuracy_score, greater_is_better=True))
clf.fit(XTrain, yTrain.ravel())
print("best parameter set is:",clf.best_params_," and its score was",clf.best_score_)

best parameter set is: {'bootstrap': True, 'max_features': 'sqrt', 'n_estimators': 1000, 'random_state': 0}  and its score was
0.9355591969173439
```

מודל: RANDOM FOREST

ניסינו לשפר את המודל עוד קצת ע"י שינוי פרמטרים:

OLD

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9290265035677879
f_measure score=  0.77181482998771
```

NEW

```
Accuracy on training data=  1.0
Accuracy on test data=  0.9297910295616718
f_measure score=  0.7761072734660707
```

ניתן לראות שיפור קטן בf_measure_score
(של כ~0.05%)

מודל: RANDOM FOREST

ניסינו לשפר את המודל עוד קצת ע"י שינוי פרמטרים:

ניתן לראות שיפור קטן בf_measure_score
(של כ~0.05%)

OLD

NEW

מודל: RANDOM FOREST

יכול להיות שבטעות בחרנו את הפרמטרים שיתנו לנו תוצאות כי טובות?
כדי לבדוק שלא במקרה בחרנו את הפרמטרים הכי טובים הפעלנו את המודל ל[illegible] ביבלנו תוצא[illegible] יותר.

```
random_state=  0
Accuracy on training data=  0.9999575209209465
Accuracy on test data=  0.9357798165137615
f_measure score=  0.7840616966580978
--------
--------
random_state=  50
Accuracy on training data=  1.0
Accuracy on test data=  0.9342507645259939
f_measure score=  0.7798634812286689
--------
--------
random_state=  80
Accuracy on training data=  0.9999575209209465
Accuracy on test data=  0.9288990825688074
f_measure score=  0.7680798004987532
--------
--------
random_state=  100
Accuracy on training data=  1.0
Accuracy on test data=  0.9357798165137615
f_measure score=  0.7882352941176473
--------
--------
random_state=  125
Accuracy on training data=  1.0
Accuracy on test data=  0.9323394495412844
f_measure score=  0.779759435918706
```

```
random_state=  150
Accuracy on training data=  1.0
Accuracy on test data=  0.933868501529052
f_measure score=  0.7843788948899045
--------
--------
random_state=  200
Accuracy on training data=  1.0
Accuracy on test data=  0.9346330275229358
f_measure score=  0.7795444778685002
--------
--------
random_state=  250
Accuracy on training data=  0.9999575209209465
Accuracy on test data=  0.9353975535168195
f_measure score=  0.7819354838709678
--------
--------
random_state=  333
Accuracy on training data=  1.0
Accuracy on test data=  0.9332313965341489
f_measure score=  0.7811194653299917
--------
--------
random_state=  500
Accuracy on training data=  1.0
Accuracy on test data=  0.9331039755351682
f_measure score=  0.7870182555780934
```

מודל: RANDOM FOREST

יכול להיות שבטעות בחרנו את הפרמטרים שיתנו לנו תוצאות כי טובות?

כדי לבדוק שלא במקרה בחרנו את הפרמטרים הכי טובים הפעלנו את המודל ע[illegible]לנו תו[illegible] יותר.

```
random_state=  555
Accuracy on training data=  1.0
Accuracy on test data=  0.9305555555555556
f_measure score=  0.7752577319587629
--------
--------
random_state=  600
Accuracy on training data=  1.0
Accuracy on test data=  0.9332313965341489
f_measure score=  0.7840065952184667
--------
--------
random_state=  666
Accuracy on training data=  1.0
Accuracy on test data=  0.9343781855249745
f_measure score=  0.7829751369574378
--------
--------
random_state=  700
Accuracy on training data=  1.0
Accuracy on test data=  0.9384556574923547
f_measure score=  0.8004956629491945
--------
--------
random_state=  777
Accuracy on training data=  1.0
Accuracy on test data=  0.9334862385321101
f_measure score=  0.7753872633390706
```

```
random_state=  800
Accuracy on training data=  1.0
Accuracy on test data=  0.9308103975535168
f_measure score=  0.7631923244657653
--------
--------
random_state=  850
Accuracy on training data=  1.0
Accuracy on test data=  0.9343781855249745
f_measure score=  0.7803837953091685
--------
--------
random_state=  900
Accuracy on training data=  1.0
Accuracy on test data=  0.9366717635066258
f_measure score=  0.7926574885273259
--------
--------
random_state=  950
Accuracy on training data=  1.0
Accuracy on test data=  0.9337410805300713
f_measure score=  0.7844112769485904
--------
--------
random_state=  1000
Accuracy on training data=  1.0
Accuracy on test data=  0.9331039755351682
f_measure score=  0.780242779405609
```

סיכום ומסקנות:

בשביל לחזות אם משחק יזכה בפרס או לא, היינו צריכים היכרות עם הנתונים.
למשל: לדעת מה המחיר המשחק הממוצע בשביל לאתר נתונים חריגים ולבחור בפעולה הולמת.
בנוסף, נדרשת יצירתיות על מנת למצוא דרכים להוציא את המרב מהנתונים.
לדוגמה: המצאת עמודות חדשות עם ממוצע פרסים של מפתח וכלל המשחקים שלו והקשר ביניהם לבין הסיכוי שהמשחק שלו יזכה בפרס שוב פעם.
מהתבוננות לאחור על המודלים לחיזוי שלנו, אנחנו מאמינים כי הגענו לתוצאות טובות עם סיכוי חיזוי יחסית גבוה.

FIN