Lijian Mei

CS3120 Project

So I using a Steam Game dataset to find what type of features impact sales of games, the type of factors that has the highest influence on if a game sells well.

Using Kaggle: gamalytic_steam_games.csv https://www.kaggle.com/datasets/safwaibrahim/gamalytic-steam-games-csv/data and Kaggle: https://www.kaggle.com/datasets/artermiloff/steam-games-dataset?select=games_march2025_cleaned.csv Merged the two datasets and removed NaN rows. The file read in Project_merge.csv which is readed into for Analysis.

The goal is to use a RandomForest model to find features that impacted the what type of games sold more copies. The was a lot of factors to take in to account.

In [8]:

```
import plotly as px
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import glob

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score
```

In [9]:

```
#Import data from Merge - created by the two datasets above
df = pd.read_csv('data3/Project_merge.csv', low_memory=False) #From Merge jupter no
print('Rows: ', df.shape[0], ' and ', df.shape[1], ' Columns' )
```

```
Rows:  84536  and  16  Columns
```

In [10]:

```
df.head(3)
```

Out[10]:

	appid	name	release_date	price	developers	publishers	categories	genres	po
0	20	Team Fortress Classic	1999-04-01	4.99	['Valve']	['Valve']	['Multi-player', 'PvP', 'Online PvP', 'Shared/...	['Action']	
1	240	Counter-Strike: Source	2004-11-01	9.99	['Valve']	['Valve']	['Multi-player', 'Cross-Platform Multiplayer',...	['Action']	17
2	300	Day of Defeat: Source	2010-07-12	9.99	['Valve']	['Valve']	['Multi-player', 'Cross-Platform Multiplayer',...	['Action']	2

Columns - ['developers', 'publishers'] is encode using LabelEncoders for size.

In [11]:

```
cols_to_encode = ['developers', 'publishers']
# Dictionary to store LabelEncoders
encoders = {}

for col in cols_to_encode:
    # Initialize encoder
    le = LabelEncoder()
    df[col + '_encoded'] = le.fit_transform(df[col])
    encoders[col] = le

numeric_df = df.select_dtypes(include=['number']).copy()
```

In [12]:

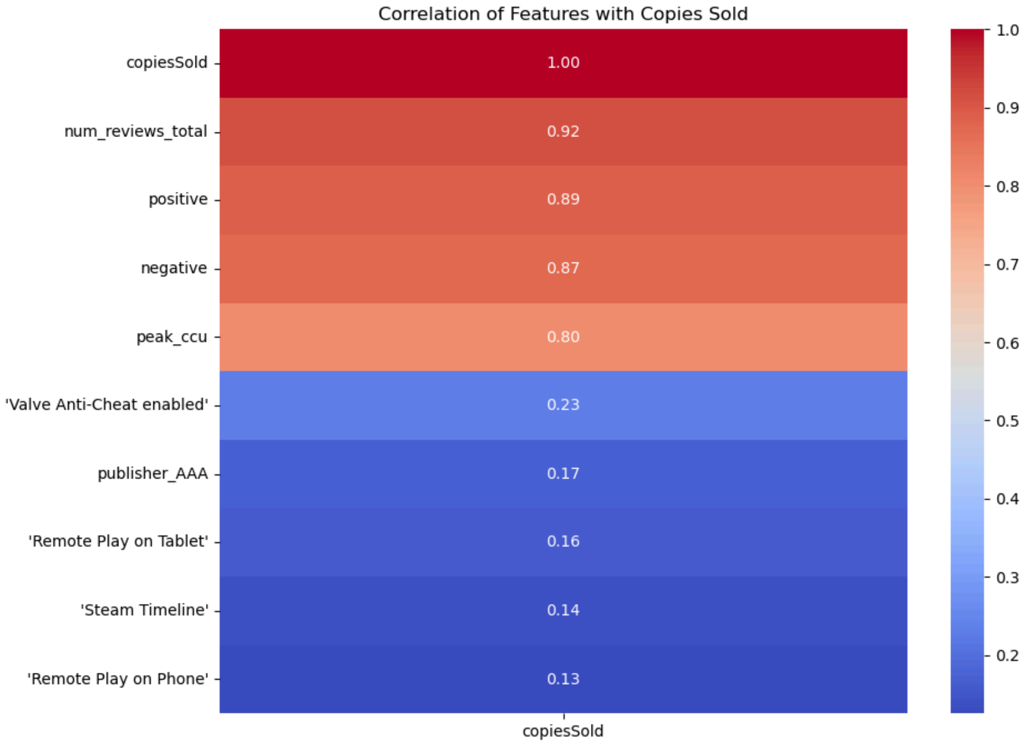
```
# Columns ['categories', 'genres', 'publisherClass] are converted one-hot encoded
print('Starting: ', numeric_df.shape)

df['genres'] = df['genres'].str.replace('[', '', regex=False).str.replace(']', '',
genre = df['genres'].str.get_dummies(sep=',')
df['categories'] = df['categories'].str.replace('[', '', regex=False).str.replace('
categories = df['categories'].str.get_dummies(sep=',')
publisher = pd.get_dummies(df['publisherClass'], prefix='publisher').astype(int)


numeric_df_final = pd.concat([numeric_df, genre, categories, publisher], axis=1)
print('End: ', numeric_df_final.shape)
```

```
Starting:  (84536, 11)
End:  (84536, 120)
```

In [13]:

```
print(numeric_df_final.head(3))
```

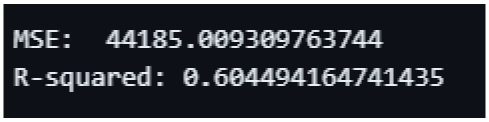
```
   appid  price  positive  negative  peak_ccu  pct_pos_total  \
0     20   4.99    7500.0    1121.0      46.0           86.0
1    240   9.99  172801.0    6697.0   14426.0           96.0
2    300   9.99   20604.0    1878.0     285.0           90.0

   num_reviews_total  copiesSold  reviewScore  developers_encoded  ...  \
0             6482.0    378635.0         87.0               46087  ...
1           124438.0  15468468.0         96.0               46087  ...
2            15155.0   1172320.0         92.0               46087  ...

   'Stats'  'Steam Achievements'  'Steam Cloud'  'Steam Trading Cards'  \
0        0                     0              0                      0
1        0                     0              0                      0
2        0                     0              0                      0

   'Tracked Controller Support'  'VR Only'  publisher_AA  publisher_AAA  \
0                             0          0             0              1
1                             0          0             0              1
2                             0          0             0              1

   publisher_Hobbyist  publisher_Indie
0                   0                0
1                   0                0
2                   0                0

[3 rows x 120 columns]
```



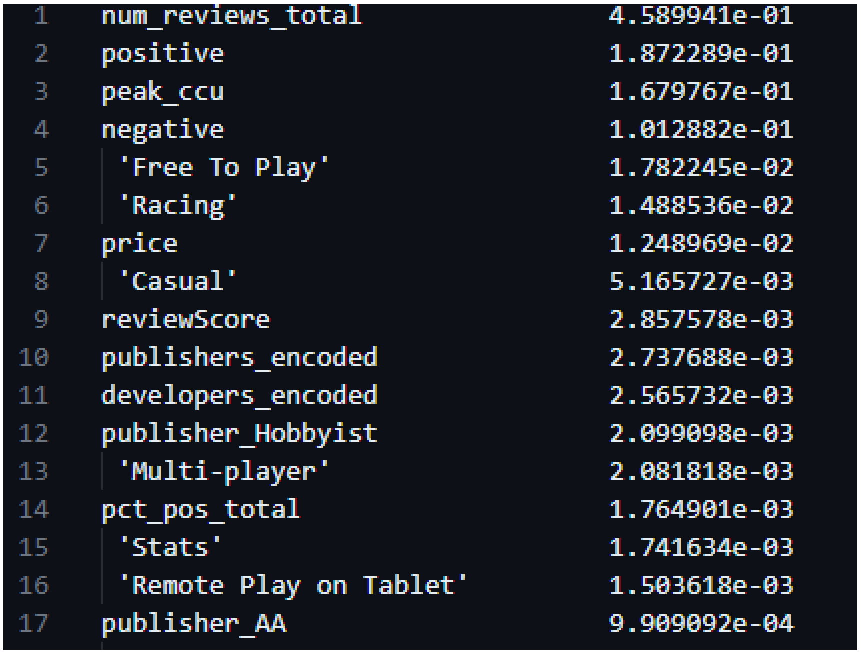
Using EDA these are the basic Corr

My model was RandomForest Regression to find CopiesSold.

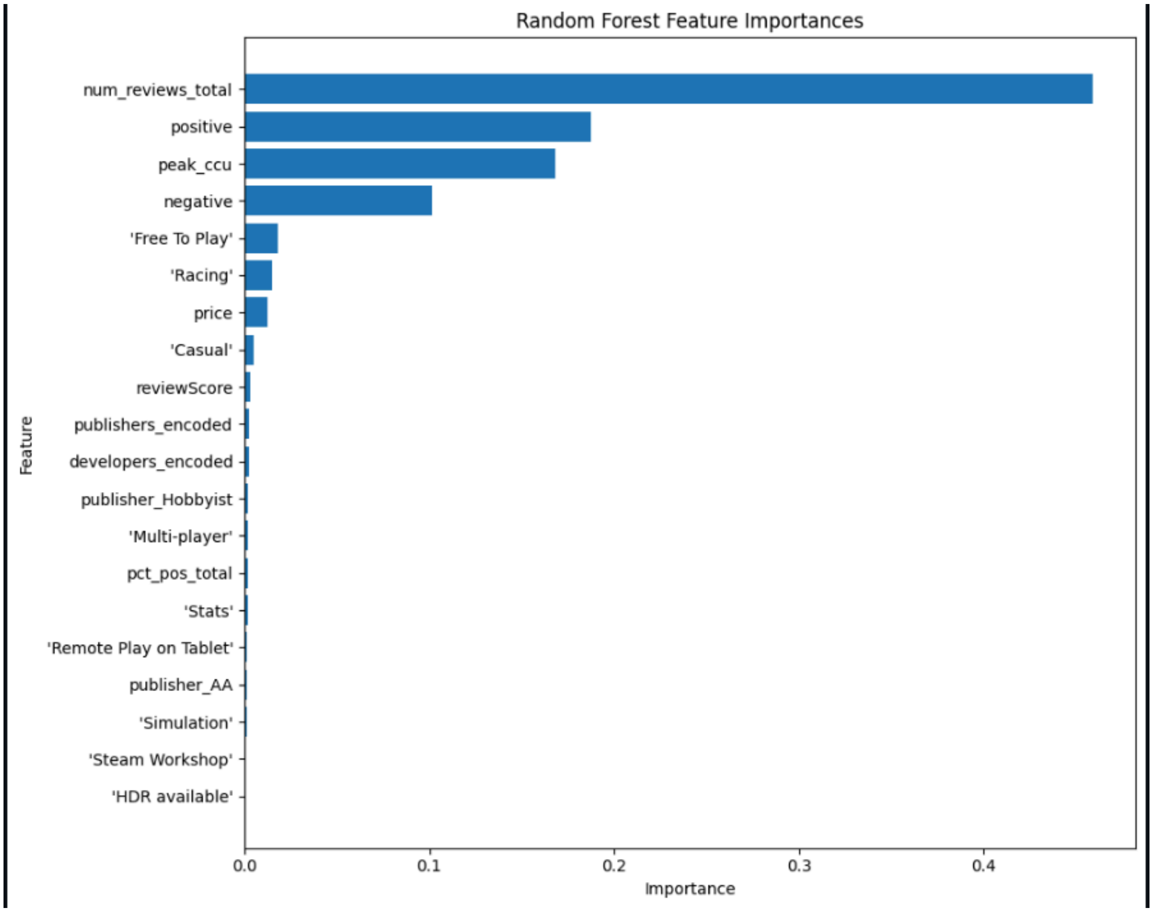
Split my data to test size 0.2



```
MSE:   44185.009309763744
R-squared:  0.604494164741435
```

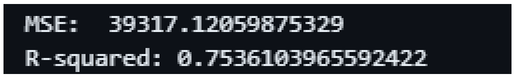


```
num_reviews_total                  4.589941e-01
positive                           1.872289e-01
peak_ccu                           1.679767e-01
negative                           1.012882e-01
 'Free To Play'                    1.782245e-02
 'Racing'                          1.488536e-02
price                              1.248969e-02
 'Casual'                          5.165727e-03
reviewScore                        2.857578e-03
publishers_encoded                 2.737688e-03
developers_encoded                 2.565732e-03
publisher_Hobbyist                 2.099098e-03
 'Multi-player'                    2.081818e-03
pct_pos_total                      1.764901e-03
 'Stats'                           1.741634e-03
 'Remote Play on Tablet'           1.503618e-03
publisher_AA                       9.909092e-04
```

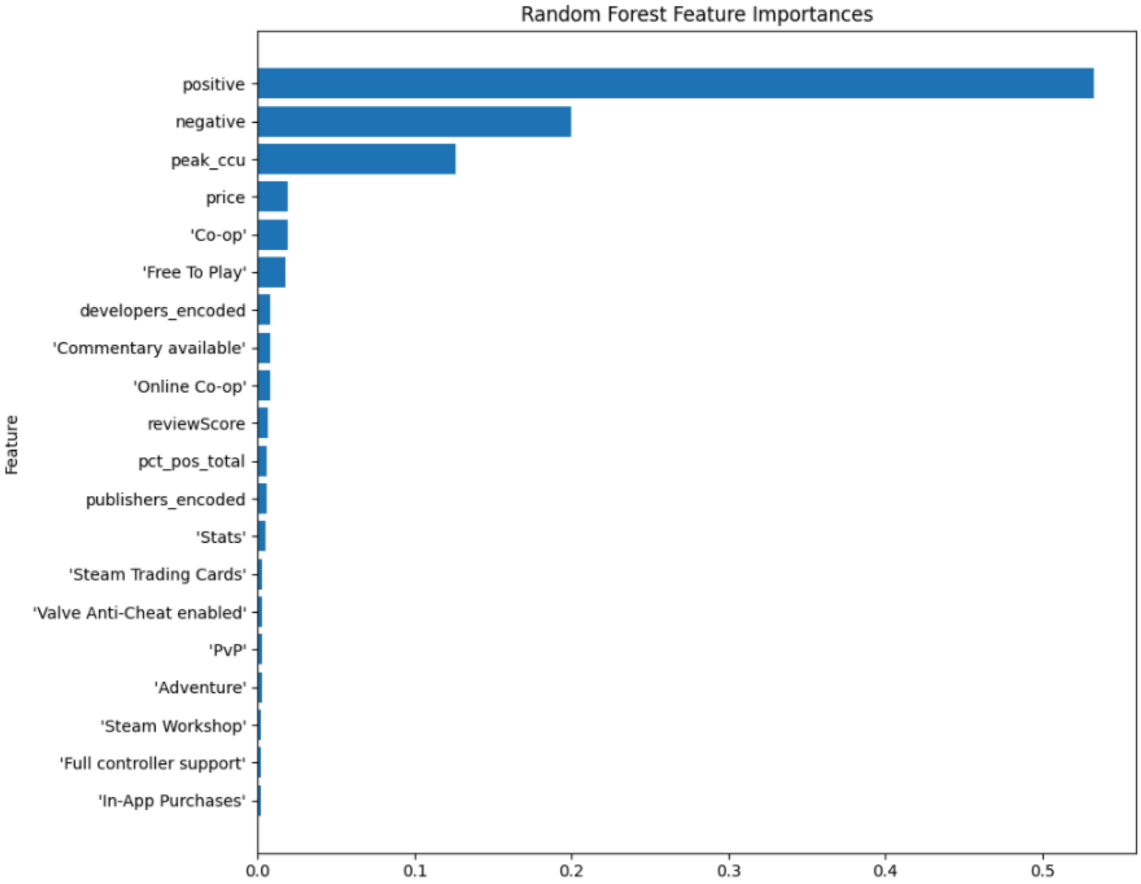


Number of reviews had a large impact but not as large as the basic correlation graph. So what if we removed the num_reviews

rf2 = RandomForestRegressor(n_estimators=200, max_depth=200)



```
MSE:  39317.12059875329
R-squared: 0.7536103965592422
```


Random Forest Feature Importances
Feature
positive
negative
peak_ccu
price
'Co-op'
'Free To Play'
developers_encoded
'Commentary available'
'Online Co-op'
reviewScore
pct_pos_total
publishers_encoded
'Stats'
'Steam Trading Cards'
'Valve Anti-Cheat enabled'
'PvP'
'Adventure'
'Steam Workshop'
'Full controller support'
'In-App Purchases'