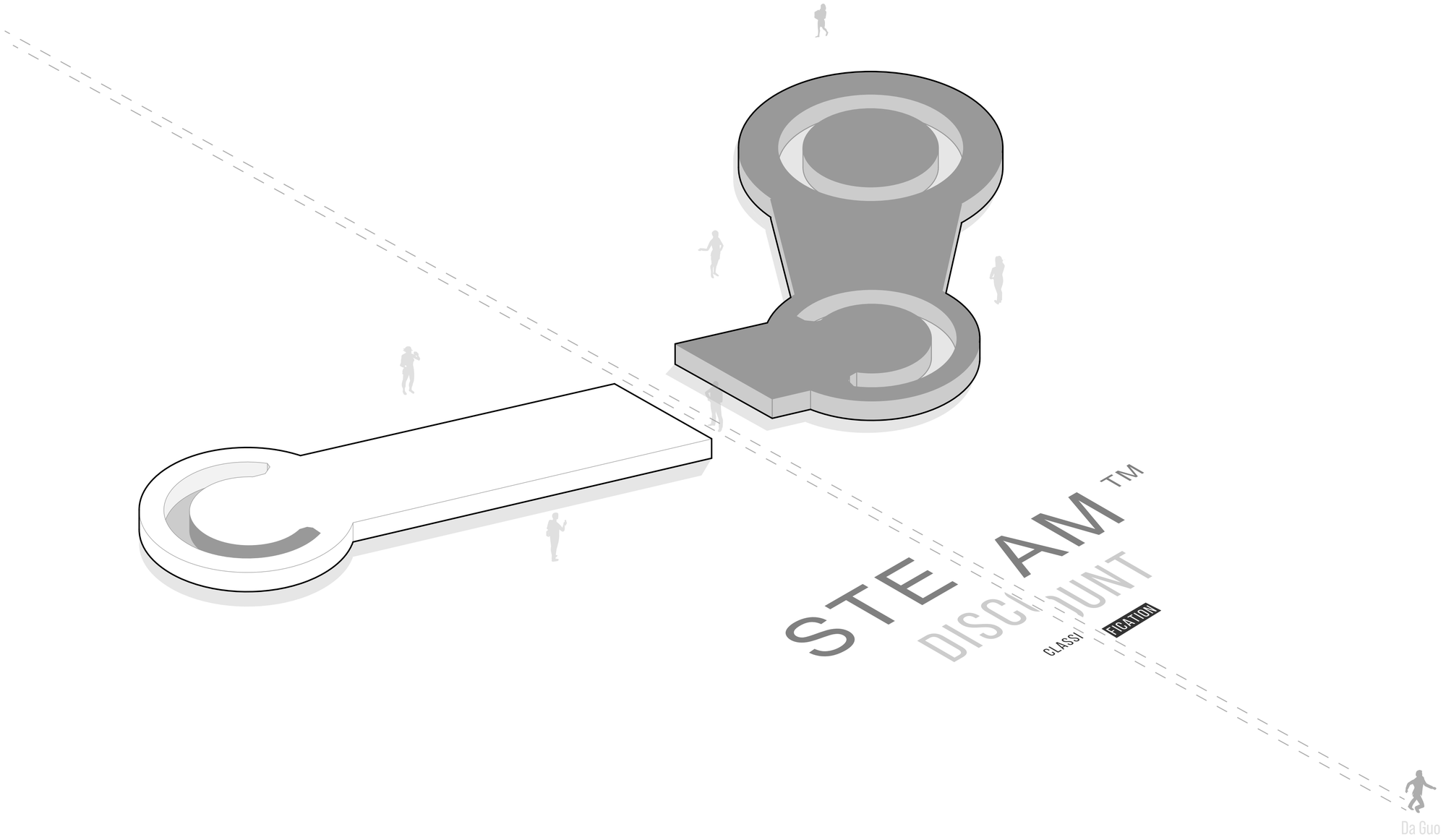


Da Guo

ASK:

Questions

Questions

1

If a strategy game on Steam would go on sale,

during a regular week, using classification models.

2/2 What determines the magnitude of strategy game discounts

ANSWERED IN PROJECT 2

on Steam during a regular week, using linear regression modeling.

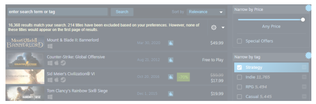
SEE:

Over | view

Overview

Data Scraping FROM PROJECT 2

Volumn: 14,806 Features: 12



```
In [4]: df.shape
Out[4]: (14806, 12)
```

Data Cleaning

Volumn: 8,665 Features: 5

Exclusion:

Free games (demos, upcomings)

	platform	reviewcount	positivepercent	releasedate	originalprice	discountpercentage
1	Win, Mac, Linux	68,938	79	2016-10-20	59.99	-70
2	Win	539,688	88	2015-12-01	19.99	[NULL]
3	Win	236	74	2017-03-14	9.99	[NULL]
4	Win, Mac	560	61	2015-01-20	129.99	-90

200 8,665

EDA FROM PROJECT 2

Known important features:

days_since_release

& % of positive reviews

Feature Engineering

Including:

feature transformation (polynomial, log)

, feature combination (multiply, divide)

& feature selection (by highest vote counts)

Modeling

Comparing predetermined metrics:

1. ROC AUC score (skillfulness)
2. F1 (in relation to Precision vs Recall balance)

ex. prec : 0.8 recall : 0.02 ✗

prec : 0.2 recall : 0.3 ✓

Source: https://www.kaggle.com/mlwhiz/feature-selection-using-football-data

CLASSIFICATION:

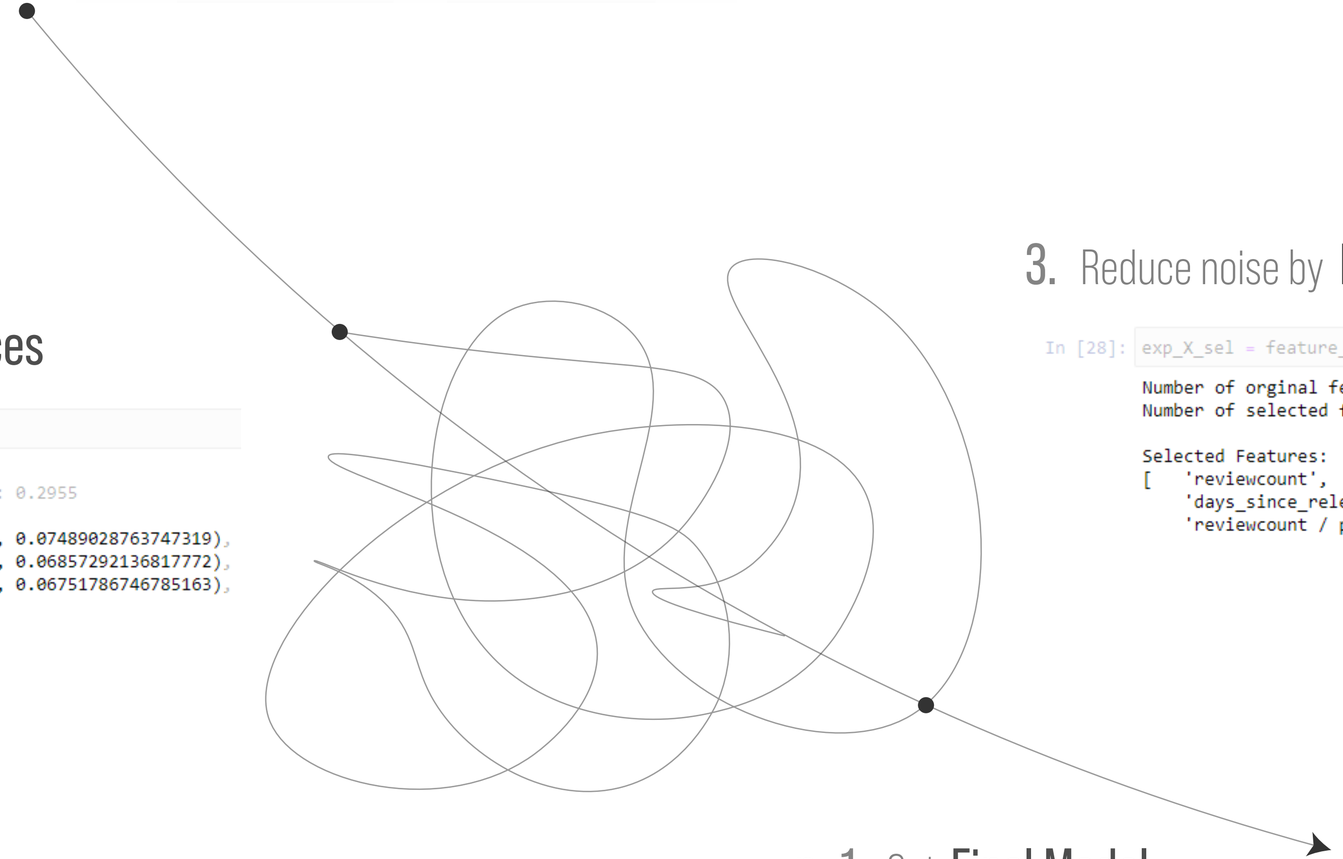
Feature | Model

1. Establishing a **baseline** Feature: 5

```
In [92]: baseline_model = classification(X, y, {0: 1, 1: 15})

         Best model is  GradientBoostingClassifier
         With f1 score of 0.2727 and ROC AUC of 0.6950
```

2. Exploratory **F.E.** & Check **Feature Importances**

```
In [22]: explore_X = explore_fe(X, y)

         Accuracy:  0.9347711730668069
         Precision: 0.7647,   Recall: 0.1831,   f1: 0.2955

         [   ('days_since_release * multiplatform', 0.07489028763747319),
             ('originalprice * days_since_release', 0.06857292136817772),
             ('days_since_release / multiplatform', 0.06751786746785163),
                                    ⋮
```



3. Reduce noise by **Feature Selection**

```
In [28]: exp_X_sel = feature_selection(explore_X, y)

         Number of orginal features:  25
         Number of selected features:  12

         Selected Features:
         [   'reviewcount',
             'days_since_release',
             'reviewcount / positivepercent',
                                    ⋮
```

n+1. Get **Final Model**

```
In [95]: best_model=classification(exp_Xpoly_sel_5, y, {0: 1, 1: 15})

         Best model is  GradientBoostingClassifier
         With f1 score of 0.2746 and ROC AUC of 0.7097
```

▲ 2.12%

LIVE DEMO:

Website | Tableau

steam-discount-predictor.herokuapp.com

THANK YOU