

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по курсу

«Data Science»

Слушатель: Быков Илья Павлович

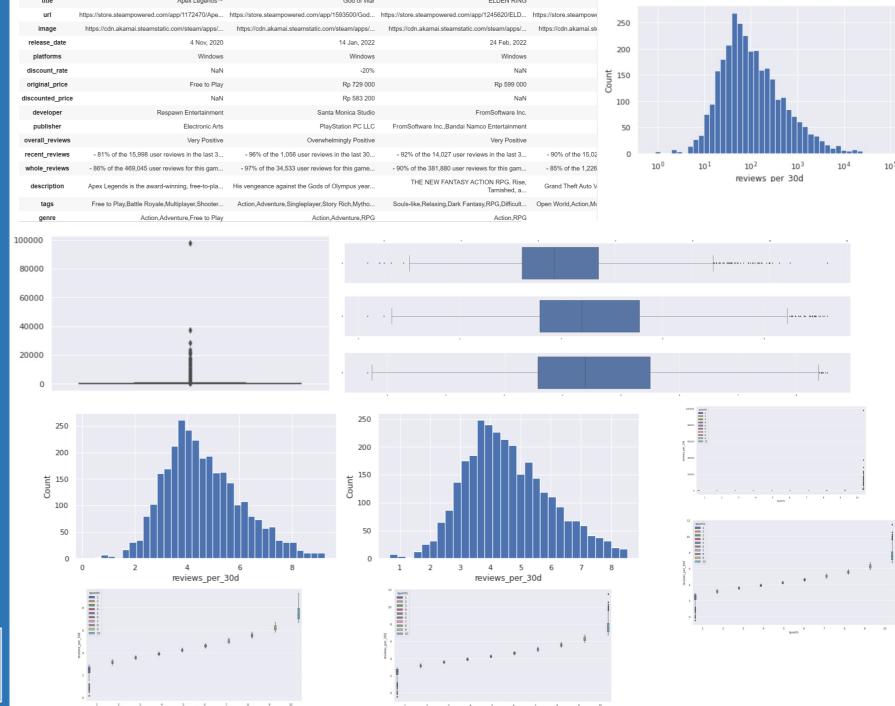
Разведочный анализ:

- Первичная обработка данных
- Избавление от дубликатов
- Избавление от пропусков
- Создание новых полей
- Исследования методов работы с выбросами
- Подготовка к работе с моделями

Построение графиков

- Построение гистограмм
- Построение диаграмм «ящик с усами»
- Построение графиков разбивки на квантили





Загрузка и исследование датасета:

- Делаем все нужные импорты
- Загружаем датасет
- Изучаем датасет
- Удаляем все файлы где меньше 3-х меток
- Удаляем все игры где нет отзывов
- Выявлем дубликаты по меткам и целевой переменной, удаляем
- Выявляем дубликаты только по меткам, удаляем их с заменой значения целевой переменной на среднее

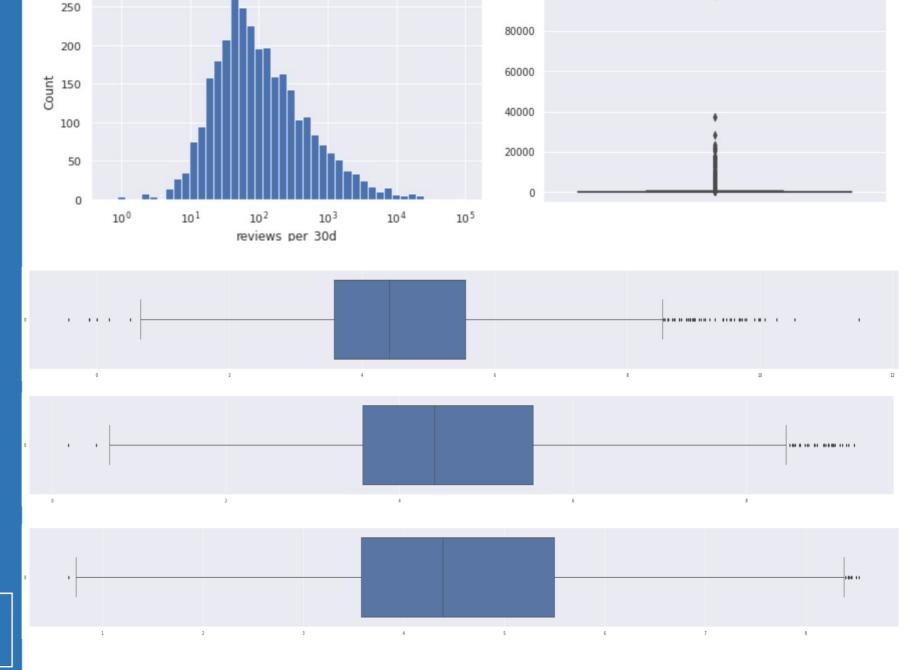


	reviews_count			dupl_che	ck		
2794	203	StrategyWarWorld War II			r II		
3577	203	StrategyWarWorld War II			r II		
3669	112856	StrategyTurn-BasedTurn-Based Strategy			gy		
5316	112856	StrategyTurn-Based	Turn-	Based Strate	gy		
5283	74015	StrategyTov	wer D	efenseZombi	es		
4008	54	2D Platformer					
3726	279				day	/s_since_release	dup1_check
4365	279		1639	5275		784	StrategyTurn-BasedTurn-Based Tactics
3115	1132	1990'sAdv	626 4633	3188		1296	StrategyTurn-BasedTurn-Based Tactics
4151	1132	1990'sAdv	2077	56957 3875		2329 5718	StrategyTurn-BasedTurn-Based Strategy StrategyTurn-BasedTurn-Based Strategy
			3435	112856		4291	StrategyTurn-BasedTurn-Based Strategy
			2109	146		733	2D FighterActionFighting
			867	4333		265	2D FighterActionAnime
			1345	2566		944	2D FighterActionAnime
			4632	4687		830	2D FighterActionAnime
			561	1787		97	2D FighterActionAnime

2336 rows × 3 columns

Создание и исследование целевой переменной:

- Исследование вопроса о том, что использовать в качестве целевой переменной
- Было решено использовать количество отзывов в месяц
- Исследование целевой переменной, построение графиков ее логарифма
- Исследование методов избавления от выбросов
- Решение не избавляться от выбросов



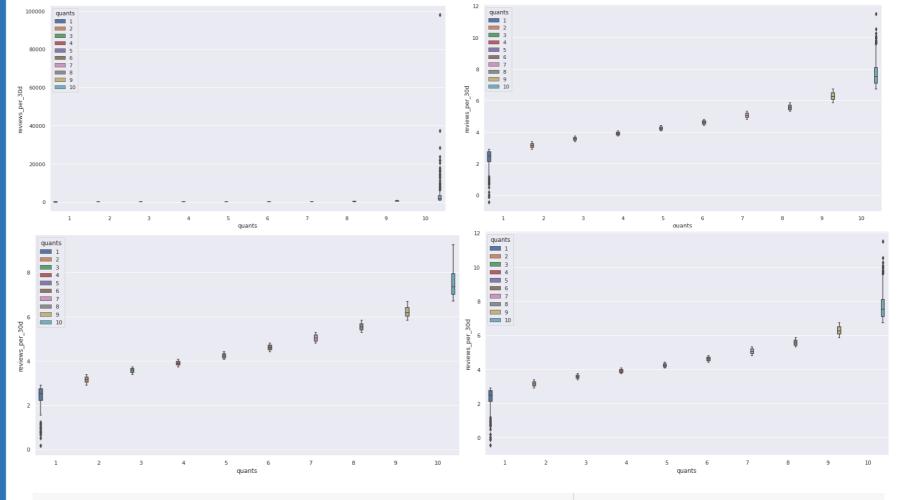
100000



Разбивка целевой переменной на квантили и поиск целевого класса

- Разбивка целевой переменной на квантили, и построения графика «Ящик с усами» для всех квантилей
- То же самое но с логарифмом целевой переменной
- Проверка способов избавления от выбросов zscore и quantile
- Окончательное решение не избавляться от выбросов
- Определение целевого класса как двух последних квантилей целевой переменной





```
quants = dst.reviews_per_30d.quantile([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
dst["rating"] = pd.cut(dst.reviews_per_30d, quants, labels = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], right=True, include_lowest=True)
```

```
#Используем последние 2 квантиля, как бинарный класс определяющий "успех" dst['is_successful'] = dst.rating > 8 dst['is_successful'].sum()
```

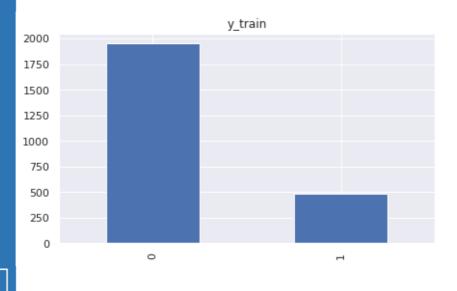
Предобработка данных перед подачей в модели:

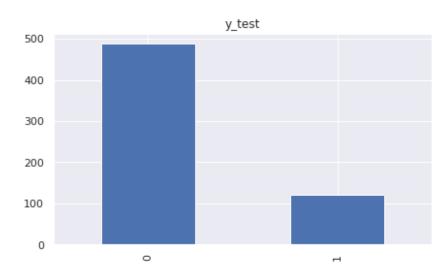
- Выборка всех меток из датасета, и создание датасета состоящего исключительно из трех первых меток всех записей
- Кодировка меток методом TargetEncoder, превращение их в численные значения
- Смесь апостериорной вероятности целевой переменной с заданным конкретным категориальным значением и априорной вероятности цели по всем обучающим данным
- Исследование дисбаланса классов
- Разбитие на тренировочную и тестовую выборки с применением стратификации

	ОБРАЗОВАТЕЛЬНЫЙ
1830	ЦЕНТР мгту им. Н. Э. Баумана

	tag1	tag2	tag3
0	Free to Play	Battle Royale	Multiplayer
1	Action	Adventure	Singleplayer
2	Souls-like	Relaxing	Dark Fantasy
3	Open World	Action	Multiplayer
4	Racing	Open World	Driving

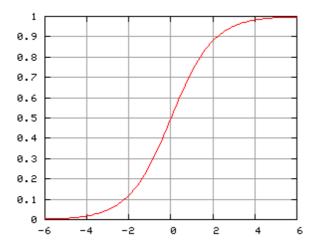
	tag1	tag2	tag3
177	0.210210	0.255002	0.429923
3007	0.270230	0.325690	0.126232
2955	0.345940	0.389890	0.244699
2389	0.261258	0.130153	0.211574
2602	0.244699	0.272970	0.171630

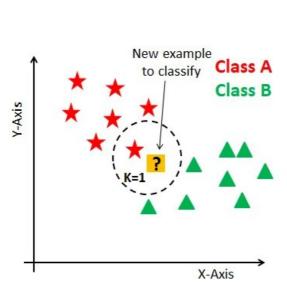


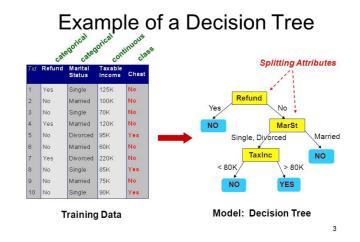


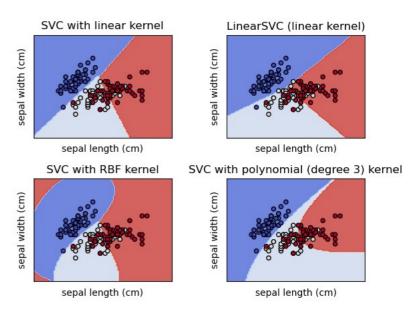
Исследование методов машинного обучения

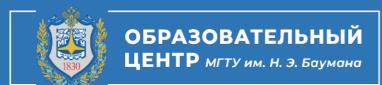
- Логистическая регрессия (LogisticRegression)
- Гауссовский байесовский классификатор (GaussianNB)
- Метод К-ближайших соседей (KNeighborsClassifier)
- Дерево решений(DecisionTreeClassifier)
- Метод опорных векторов (SVC)
- Случайный лес (Random forest)











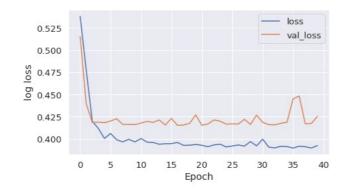
Обучение и тестирование моделей:

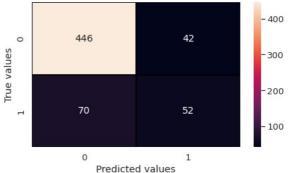
- Accuracy не работает с несбалансированными данными
- fl_score среднее гармоническое между precision и recall
- Precision точность определения позитивных классов
- Recall полнота охвата позитивных классов
- matthews_corr_coef корреляция между предсказанным и правдой, хорошо работает с несбалансированными данными
- Лучший алгоритм K-Nearest neighbor, потому что лучший matthews_corr_coef
- Построена нейронная сеть для бинарной классификации



	model_name	f1_score	precision	recall	matthews_corr_coef
0	Logistic regression	0.509202	0.406863	0.680328	0.366585
1	Decision Tree Classifier	0.355731	0.343511	0.368852	0.187627
2	Gaussian Naive Bayes	0.341969	0.464789	0.270492	0.240256
3	K-Nearest neighbors	0.445652	0.661290	0.336066	0.387900
4	SVC	0.474138	0.500000	0.450820	0.351781
5	Random forest classifier	0.464000	0.453125	0.475410	0.326104
6	Neural network	0.391061	0.614035	0.286885	0.332317

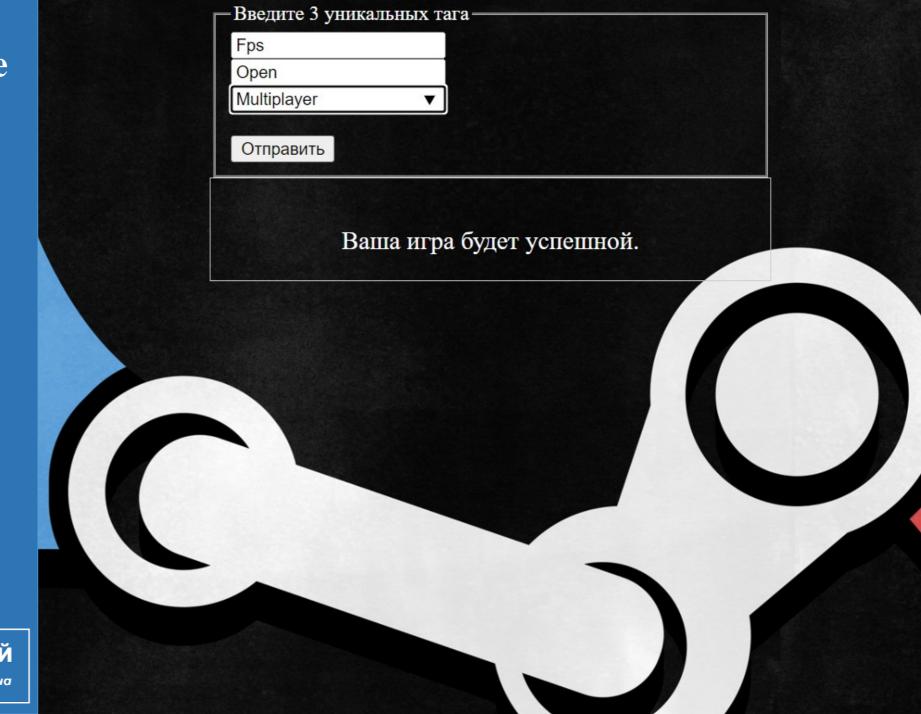
```
model = Sequential()
model.add(Dense(512, activation='relu', input_dim=3))
model.add(Dense(512, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy')
```





Построено приложение на базе фреймворка Flask

- Принимает на вход 3 метки игры
- Предсказывает ждет ли ее успех





Спасибо за Внимание!

