Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Курсовая работа по дисциплине «Методы машинного обучения» на тему: «Исследование продаж на рынке видеоигр»

ИСПОЛНИТЕЛЬ:

Мокренко Никита Валерьевич
ИУ5-34М

_____

"__"_____2021 г.

Москва 2021

Оглавление

Задание

1. Поиск и выбор набора данных для построения модели машинного обучения. На основе выбранного набора данных строится модель для задачи классификации.
2. Для выбранного датасета решить следующие задачи:
    a. устранение пропусков в данных;
    b. кодирование категориальных признаков;
    c. нормализацию числовых признаков;
    d. масштабирование признаков;
    e. обработку выбросов для числовых признаков;
    f. обработку нестандартных признаков (которые не является числовым или категориальным);
    g. отбор признаков, наиболее подходящих для построения модели;
3. Обучить модель и оценить метрики качества для двух выборок:
    a. исходная выборка, которая содержит только минимальную предобработку данных, необходимую для построения модели (например, кодирование категориальных признаков).
    b. улучшенная выборка, полученная в результате полной предобработки данных в пункте 2.
4. Построить модель с использованием произвольной библиотеки AutoML.
5. Сравнить метрики для трех полученных моделей.

Подготовка данных

Загрузка датасета

Ввод [5]:
```
data = pd.read_csv('./vgsales.csv', sep=',', encoding="utf-8")
data.head()
```

Out[5]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

Устранение пропусков данных

Пропуски присутствуют в двух столбцах. Так как оба признака являются категориальными и количество пропусков незначительно, можно выполнить замену пропусков одним значением («Unknown»), либо вообще исключить строки с пропусками.

Ввод [21]:
```
data = data.fillna(value="Unknown")
```

Ввод [22]:
```
data.isnull().sum()
```

Out[22]:
```
Rank            0
Name            0
Platform        0
Year            0
Genre           0
Publisher       0
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

Обработка категориальных признаков

Можно использовать Count Encoder для признаков Genre, Platform, Year и Publisher.

Ввод [86]:
```
data_trns = data
enc = c_enc.CountEncoder(cols=['Genre','Publisher','Platform','Year'], normalize=True)
data_trns = enc.fit_transform(data)
data_trns.sort_values('Global_Sales', ascending=False).head()
```

Out[86]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	0.079829	0.060730	0.141342	0.042355	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	0.005904	0.000843	0.053380	0.042355	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	0.079829	0.086034	0.075250	0.042355	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	0.079829	0.086215	0.141342	0.042355	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	0.005904	0.015845	0.089649	0.042355	11.27	8.89	10.22	1.00	31.37

Нормализация числовых признаков

Не провожу нормализацию, т.к. в данных много нулевых значений.

Масштабирование признаков

Использую MinMaxScaler для масштабирования признаков, относящихся к продажам.

Ввод [170]:
```
cols = ['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']
df = data_trns[cols]
data_scl = data_trns
scl = MinMaxScaler()
df = pd.DataFrame(mmscale.fit_transform(df), columns = cols)
df.head()
```

Out[170]:

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1.000000	1.000000	0.368885	0.800378	1.000000
1	0.700892	0.123363	0.666341	0.072848	0.486281
2	0.382020	0.443832	0.370841	0.313150	0.432854
3	0.379610	0.379394	0.320939	0.280038	0.398767
4	0.271632	0.306340	1.000000	0.094607	0.379064

Отбор признаков

Можно пока исключить признак Rank, т.к. это уникальное значение, обозначающее место в рейтинге. Гораздо разумнее предсказывать продажи.

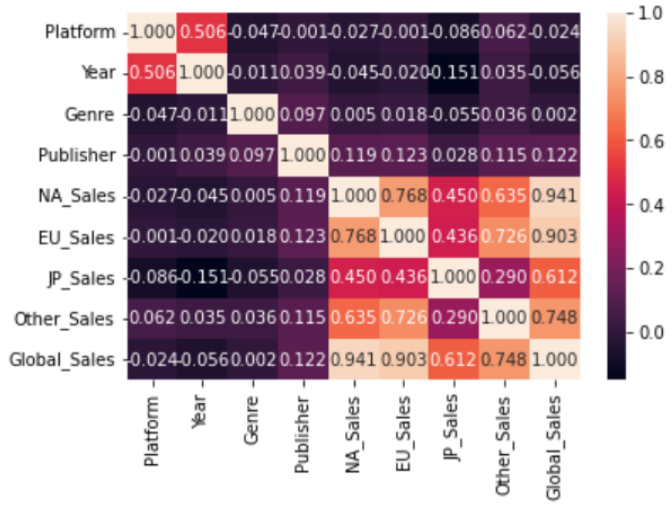
Проверим датасет на наличие константных признаков:

Ввод [178]:
```
df = data_trns.drop(columns=['Name', 'Rank'])
selector = VarianceThreshold(threshold=0.15)
selector.fit(df)
for i in range(len(selector.variances_)):
    print(selector.variances_[i].round(2), '\t', df.columns[i])
```

```
0.0     Platform
0.0     Year
0.0     Genre
0.0     Publisher
0.67    NA_Sales
0.26    EU_Sales
0.1     JP_Sales
0.04    Other_Sales
2.42    Global_Sales
```

Теперь посмотрим корреляционную матрицу:

```
Ввод [181]: sns.heatmap(df.corr(),annot = True, fmt = '.3f')
Out[181]: <AxesSubplot:>
```

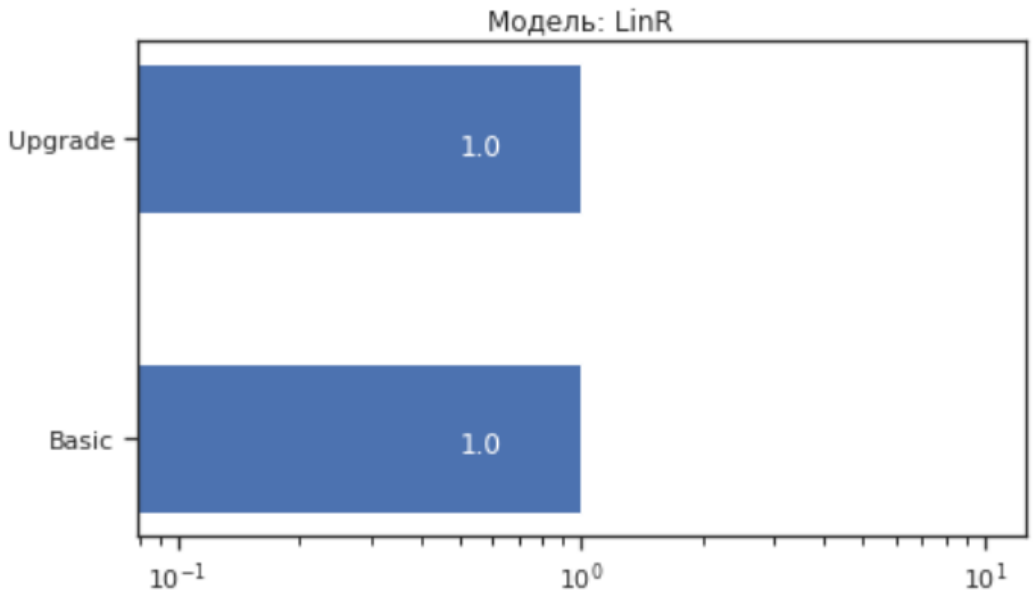


Видно, что признак продаж по миру сильно коррелирует с продажами в Европе и Северной Америке (что логично), так что при построении модели машинного обучения его можно будет убрать.

Результат работы моделей

In [125]:
```
clas_models_dict = {'LinR': LogisticRegression(),
                    'KNN_5':KNeighborsClassifier(n_neighbors=5),
                    'Tree':DecisionTreeClassifier(random_state=1),
                    'GB': GradientBoostingClassifier(random_state=1),
                    'RF':RandomForestClassifier(n_estimators=20, random_state=1)}
```

In [129]:
```
for model in clas_models_dict:
    logger.plot('Модель: ' + model, model, figsize=(7, 4))
```



AutoML

Обучение при помощи технологии AutoML

```
automl.fit(train[train.columns[:-1]], train['EU_Sales'])
```

```
Linear algorithm was disabled.
AutoML directory: AutoML_1
The task is regression with evaluation metric rmse
AutoML will use algorithms: ['Baseline', 'Decision Tree', 'Random Forest', 'Xgboost', 'Neural Network'
AutoML will ensemble available models
AutoML steps: ['simple_algorithms', 'default_algorithms', 'ensemble']
* Step simple_algorithms will try to check up to 2 models
1_Baseline rmse 0.615306 trained in 1.37 seconds
2_DecisionTree rmse 0.30605 trained in 10.36 seconds
* Step default_algorithms will try to check up to 3 models
3_Default_Xgboost rmse 0.260395 trained in 10.48 seconds
4_Default_NeuralNetwork rmse 0.021286 trained in 2.53 seconds
5_Default_RandomForest rmse 0.332865 trained in 6.68 seconds
* Step ensemble will try to check up to 1 model
Ensemble rmse 0.021286 trained in 0.32 seconds
AutoML fit time: 69.21 seconds
AutoML best model: 4_Default_NeuralNetwork
AutoML()
```