

Московский авиационный институт

(Национальный исследовательский университет)

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Искусственный интеллект»

Студент: Перцев Д. М.

Группа: 301

Преподаватель:

Дата: 27.09.2022

Оценка:

Москва 2022

Logistic Regression

Обучим собственную модель и модель из sklearn. Посчитаем метрики и сделаем выводы.

Собственная модель

```
accuracy_score на обучающей выборки = 0.8480825958702065
recall_score на обучающей выборки = 0.0
precision_score на обучающей выборки = 0.0
roc_auc_score на обучающей выборки = 0.5
confusion_matrix на обучающей выборки =
[[2875  0]
 [ 515  0]]
accuracy_score на тестовой выборки = 0.847877358490566
recall_score на тестовой выборки = 0.0
precision_score на тестовой выборки = 0.0
roc_auc_score на тестовой выборки = 0.5
confusion_matrix на тестовой выборки =
[[719  0]
 [129  0]]
```

Модель из sklearn

```
accuracy_score на обучающей выборки = 0.8545722713864307
recall_score на обучающей выборки = 0.07572815533980583
precision_score на обучающей выборки = 0.6964285714285714
roc_auc_score на обучающей выборки = 0.5349075559307725
confusion_matrix на обучающей выборки =
[[2858  17]
 [ 476  39]]
accuracy_score на тестовой выборки = 0.8561320754716981
recall_score на тестовой выборки = 0.08527131782945736
precision_score на тестовой выборки = 0.7333333333333333
roc_auc_score на тестовой выборки = 0.5398540177464394
confusion_matrix на тестовой выборки =
[[715  4]
 [118 11]]
```

Выводы

- Исходя из метрик, можно сделать вывод, что моя модель плохо применима в ситуации, когда один из классов плохо представлен в выборке
- Переобучение не наблюдается

SVM

Обучим собственную модель и модель из sklearn. Посчитаем метрики и сделаем выводы.

Собственная модель

```

recall_score на обучающей выборки = 0.0
precision_score на обучающей выборки = 0.0
roc_auc_score на обучающей выборки = 0.5
confusion_matrix на обучающей выборки =
[[2875  0]
 [ 515  0]]
accuracy_score на тестовой выборки = 0.847877358490566
recall_score на тестовой выборки = 0.0
precision_score на тестовой выборки = 0.0
roc_auc_score на тестовой выборки = 0.5
confusion_matrix на тестовой выборки =
[[719  0]
 [129  0]]
Wall time: 2h 34min 44s

```

Модель из sklearn

```

accuracy_score на обучающей выборки = 0.35575221238938054
recall_score на обучающей выборки = 0.4446601941747573
precision_score на обучающей выборки = 0.1076633756464504
roc_auc_score на обучающей выборки = 0.39224314056563947
confusion_matrix на обучающей выборки =
[[ 977 1898]
 [ 286 229]]
accuracy_score на тестовой выборки = 0.37264150943396224
recall_score на тестовой выборки = 0.4186046511627907
precision_score на тестовой выборки = 0.10567514677103718
roc_auc_score на тестовой выборки = 0.39149982210434386
confusion_matrix на тестовой выборки =
[[262 457]
 [ 75  54]]

```

Выводы

- Исходя из метрик, можно сделать вывод, что моя модель плохо применима в ситуации, когда один из классов плохо представлен в выборке
- Точность моей модели выше чем модели из sklearn
- Переобучение не наблюдается

KNN

Обучим собственную модель и модель из sklearn. Посчитаем метрики и сделаем выводы.

Собственная модель

```
accuracy_score на обучающей выборки = 0.8648967551622418
recall_score на обучающей выборки = 0.13398058252427184
precision_score на обучающей выборки = 0.8518518518518519
roc_auc_score на обучающей выборки = 0.5649033347403968
confusion_matrix на обучающей выборки =
[[2863  12]
 [ 446  69]]
accuracy_score на тестовой выборки = 0.8384433962264151
recall_score на тестовой выборки = 0.05426356589147287
precision_score на тестовой выборки = 0.3181818181818182
roc_auc_score на тестовой выборки = 0.5167006285646516
confusion_matrix на тестовой выборки =
[[704  15]
 [122   7]]
```

Модель из sklearn

```
accuracy_score на обучающей выборки = 0.8648967551622418
recall_score на обучающей выборки = 0.13398058252427184
precision_score на обучающей выборки = 0.8518518518518519
roc_auc_score на обучающей выборки = 0.5649033347403968
confusion_matrix на обучающей выборки =
[[2863  12]
 [ 446  69]]
accuracy_score на тестовой выборки = 0.8384433962264151
recall_score на тестовой выборки = 0.05426356589147287
precision_score на тестовой выборки = 0.3181818181818182
roc_auc_score на тестовой выборки = 0.5167006285646516
confusion_matrix на тестовой выборки =
[[704  15]
 [122   7]]
```

Выводы

- Исходя из метрик, можно сделать вывод, что моя модель может применяться в задачах бинарной классификации, когда один из классов слабо представлен
- На удивление метрики моей модели и модели из sklearn совпали
- Переобучение не наблюдается

Наивный байсовский классификатор

Обучим собственную модель и модель из sklearn. Посчитаем метрики и сделаем выводы.

Собственная модель

```
accuracy_score на обучающей выборки = 0.8480825958702065
recall_score на обучающей выборки = 0.0
precision_score на обучающей выборки = 0.0
roc_auc_score на обучающей выборки = 0.5
confusion_matrix на обучающей выборки =
[[2875  0]
 [ 515  0]]
accuracy_score на тестовой выборки = 0.847877358490566
recall_score на тестовой выборки = 0.0
precision_score на тестовой выборки = 0.0
roc_auc_score на тестовой выборки = 0.5
confusion_matrix на тестовой выборки =
[[719  0]
 [129  0]]
```

Модель из sklearn

```
accuracy_score на обучающей выборки = 0.827433628318584
recall_score на обучающей выборки = 0.18640776699029127
precision_score на обучающей выборки = 0.366412213740458
roc_auc_score на обучающей выборки = 0.5643343182777543
confusion_matrix на обучающей выборки =
[[2709 166]
 [ 419  96]]
accuracy_score на тестовой выборки = 0.8349056603773585
recall_score на тестовой выборки = 0.23255813953488372
precision_score на тестовой выборки = 0.4225352112676056
roc_auc_score на тестовой выборки = 0.5877672477924767
confusion_matrix на тестовой выборки =
[[678 41]
 [ 99 30]]
```

Выводы

- Исходя из метрик, можно сделать вывод, что моя модель плохо применима в ситуации, когда один из классов плохо представлен в выборке
- Переобучение не наблюдается