

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: 2.10.22

Оценка:

Москва 2022

Мягкое и жесткое голосование

Мягкое голосование

Применим мягкое голосование для обученных моделей на обучающих данных.

```
accuracy_score на обучающей выборки = 0.8569321533923304
recall_score на обучающей выборки = 0.09902912621359224
precision_score на обучающей выборки = 0.7083333333333334
roc_auc_score на обучающей выборки = 0.5458623891937527
confusion_matrix на обучающей выборки =
[[2854  21]
 [ 464  51]]
```

Применим мягкое голосование для обученных моделей на тестовых данных.

```
accuracy_score на тестовой выборки = 0.8525943396226415
recall_score на тестовой выборки = 0.06976744186046512
precision_score на тестовой выборки = 0.6428571428571429
roc_auc_score на тестовой выборки = 0.5314066694698709
confusion_matrix на тестовой выборки =
[[714  5]
 [120  9]]
```

Жесткое голосование

Применим жесткое голосование для обученных моделей на обучающих данных.

```
accuracy_score на обучающей выборки = 0.8563421828908555
recall_score на обучающей выборки = 0.06990291262135923
precision_score на обучающей выборки = 0.8181818181818182
roc_auc_score на обучающей выборки = 0.5335601519628536
confusion_matrix на обучающей выборки =
[[2867  8]
 [ 479  36]]
```

Применим жесткое голосование для обученных моделей на тестовых данных

```
accuracy_score на тестовой выборки = 0.847877358490566
recall_score на тестовой выборки = 0.046511627906976744
precision_score на тестовой выборки = 0.5
roc_auc_score на тестовой выборки = 0.5190833522010545
confusion_matrix на тестовой выборки =
[[713  6]
 [123  6]]
```

Выводы

- Мягкое голосование показало себя лучше

Дерево решений

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

Собственная модель

```
accuracy_score на обучающей выборки = 0.868141592920354
recall_score на обучающей выборки = 0.1650485436893204
precision_score на обучающей выборки = 0.8333333333333334
roc_auc_score на обучающей выборки = 0.5795677501055297
confusion_matrix на обучающей выборки =
[[2858  17]
 [ 430  85]]
accuracy_score на тестовой выборки = 0.8254716981132075
recall_score на тестовой выборки = 0.023255813953488372
precision_score на тестовой выборки = 0.12
roc_auc_score на тестовой выборки = 0.4963288805511531
confusion_matrix на тестовой выборки =
[[697  22]
 [126   3]]
```

Модель из sklearn

```
accuracy_score на обучающей выборки = 1.0
recall_score на обучающей выборки = 1.0
precision_score на обучающей выборки = 1.0
roc_auc_score на обучающей выборки = 1.0
confusion_matrix на обучающей выборки =
[[2875   0]
 [   0 515]]
accuracy_score на тестовой выборки = 0.7747641509433962
recall_score на тестовой выборки = 0.24806201550387597
precision_score на тестовой выборки = 0.25396825396825395
roc_auc_score на тестовой выборки = 0.5586624402971396
confusion_matrix на тестовой выборки =
[[625  94]
 [ 97  32]]
```

Выводы

- Моя модель показала себя хуже на обучающей выборке, но лучше на тестовой
- Обе модели могут хорошо применяться в задачах бинарной классификации, где слабо представлен один из классов

Случайный лес

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

Собственная модель

```

accuracy_score на обучающей выборки = 0.8489675516224189
recall_score на обучающей выборки = 0.005825242718446602
precision_score на обучающей выборки = 1.0
roc_auc_score на обучающей выборки = 0.5029126213592233
confusion_matrix на обучающей выборки =
[[2875  0]
 [ 512  3]]
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 = 1.0
recall_score на обучающей выборки = 1.0
precision_score на обучающей выборки = 1.0
roc_auc_score на обучающей выборки = 1.0
confusion_matrix на обучающей выборки =
[[2875  0]
 [  0 515]]
accuracy_score на тестовой выборки = 0.8443396226415094
recall_score на тестовой выборки = 0.03875968992248062
precision_score на тестовой выборки = 0.38461538461538464
roc_auc_score на тестовой выборки = 0.5138165626246618
confusion_matrix на тестовой выборки =
[[711  8]
 [124  5]]

```

Выводы

- Обе модели не могут хорошо применяться в задачах бинарной классификации, где слабо представлен один из классов

GradientBoostingClassifier

Обучим GradientBoostingClassifier и выведем метрику на тестовой и обучающей выборки

```

accuracy_score на обучающей выборки = 0.879646017699115
recall_score на обучающей выборки = 0.21553398058252426
precision_score на обучающей выборки = 0.9652173913043478
roc_auc_score на обучающей выборки = 0.607071338117349
confusion_matrix на обучающей выборки =
[[2871  4]
 [ 404 111]]
accuracy_score на тестовой выборки = 0.839622641509434
recall_score на тестовой выборки = 0.06976744186046512
precision_score на тестовой выборки = 0.36
roc_auc_score на тестовой выборки = 0.5237571562570753
confusion_matrix на тестовой выборки =
[[703 16]
 [120  9]]

```