|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**Отчёт по лабораторной работе №6**

По дисциплине:

«Технологии машинного обучения»

Выполнил:

Студент группыИУ5-62 **\_\_\_ \_ \_ \_\_\_\_ \_ Андреев И.М.\_**

(Подпись, дата) (Фамилия И.О.)

Проверил:

**\_\_\_\_\_\_\_\_\_ \_ Гапанюк Ю. Е.\_**

(Подпись, дата) (Фамилия И.О.)

Москва, 2020

**Задание**

1. Выберите набор данных (датасет) для решения задачи классификации или регресии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода train\_test\_split разделите выборку на обучающую и тестовую.
4. Обучите две ансамблевые модели. Оцените качество моделей с помощью одной из подходящих для задачи метрик. Сравните качество полученных моделей.

**Jupyter Notebook**

Импорт библиотек и подготовка данных для моделей была произведена в лабораторной работе №4.

**Случайный лес**

In [66]:

**from** **sklearn.ensemble** **import** RandomForestClassifier, ExtraTreesClassifier

In [67]:

forest = RandomForestClassifier(n\_estimators=50, oob\_score=**True**, random\_state=1)

forest.fit(X\_train, y\_train)

Out[67]:

RandomForestClassifier(bootstrap=True, ccp\_alpha=0.0, class\_weight=None,

criterion='gini', max\_depth=None, max\_features='auto',

max\_leaf\_nodes=None, max\_samples=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=50,

n\_jobs=None, oob\_score=True, random\_state=1, verbose=0,

warm\_start=False)

In [68]:

forest.score(X\_test,y\_test)

Out[68]:

0.8491620111731844

In [69]:

y\_predicted\_forest = forest.predict(X\_test)

In [70]:

classification\_report(y\_predicted\_forest,y\_test, output\_dict=**True**)

Out[70]:

{'0': {'precision': 0.875,

'recall': 0.8828828828828829,

'f1-score': 0.8789237668161436,

'support': 111},

'1': {'precision': 0.8059701492537313,

'recall': 0.7941176470588235,

'f1-score': 0.8,

'support': 68},

'accuracy': 0.8491620111731844,

'macro avg': {'precision': 0.8404850746268657,

'recall': 0.8385002649708532,

'f1-score': 0.8394618834080718,

'support': 179},

'weighted avg': {'precision': 0.8487763695489036,

'recall': 0.8491620111731844,

'f1-score': 0.8489415537239774,

'support': 179}}

**Градиентный бустинг**

In [71]:

**from** **sklearn.ensemble** **import** AdaBoostClassifier

ab1 = AdaBoostClassifier(n\_estimators=50, algorithm='SAMME', random\_state=1)

ab1.fit(X\_train, y\_train)

Out[71]:

AdaBoostClassifier(algorithm='SAMME', base\_estimator=None, learning\_rate=1.0,

n\_estimators=50, random\_state=1)

In [72]:

ab1.score(X\_test,y\_test)

Out[72]:

0.8324022346368715

In [73]:

y\_predicted\_ab = ab1.predict(X\_test)

In [74]:

classification\_report(y\_predicted\_ab,y\_test, output\_dict=**True**)

Out[74]:

{'0': {'precision': 0.8214285714285714,

'recall': 0.9019607843137255,

'f1-score': 0.8598130841121495,

'support': 102},

'1': {'precision': 0.8507462686567164,

'recall': 0.7402597402597403,

'f1-score': 0.7916666666666667,

'support': 77},

'accuracy': 0.8324022346368715,

'macro avg': {'precision': 0.8360874200426439,

'recall': 0.8211102622867329,

'f1-score': 0.8257398753894081,

'support': 179},

'weighted avg': {'precision': 0.8340400948172149,

'recall': 0.8324022346368715,

'f1-score': 0.8304987034233104,

'support': 179}}

Вывод: построенная модель случайного леса дает лучшие результаты по всем метрикам качества модели.