Задача 4

Условие

Набор данных: [StudentsPerformance.csv](https://www.kaggle.com/spscientist/students-performance-in-exams)

Тип классификатора - *DecisionTreeClassifier* (решающее дерево)

Классификация по столбцу - *Test preparation course*

1. Обработайте набор данных набор данных, указанный во втором столбце таблицы 4.1, подготовив его к решению задачи классификации. Выделите целевой признак, указанный в последнем столбце таблицы, и удалите его из данных, на основе которых будет обучаться классификатор. Разделите набор данных на тестовую и обучающую выборку. Постройте классификатор типа, указанного в третьем столбце, для задачи классификации по параметру, указанному в последнем столбце. Оцените точность построенного классификатора с помощью метрик precision, recall и F1 на тестовой выборке.
2. Постройте классификатор типа Случайный Лес (*Random* *Forest*) для решения той же задачи классификации. Оцените его качество с помощью метрик precision, recall и F1 на тестовой выборке. Какой из классификаторов оказывается лучше?

Решение

1. Считаем данные из StudentsPerformance.csv и подготовим датасет к обработке:
   * + Удалим строки, имеющие значения NaN, это повысит точность классификатора
     + Преобразуем данные столбцов, которые имеют только два значения в бинарное представление (значение или 1, или 0)
     + Отделим *test\_preparation\_course*, переменную, значение которой мы предсказываем, от основных данных
     + Переведем категориальные переменные *race* и *parental\_level\_of\_education* в One hot encoding, сейчас каждая из них представляет собой отдельный датафрейм. Удалим оригинальные переменные из основных данных и присоеденим к основным данным датафреймы переменных в One hot encoding
     + Разделим датасет на тестовую и обучающую выборку. Размер тестовой выборки равен 33% от общей выборки, переменная *random\_state* равна 1

Построим классификатор типа решающее дерево (*decision tree classifier*), далее оценим

его эффективность с помощью метрик *precision, recall* и *F1*:

* f1: 0.507
* precision: 0.468
* recall: 0.463

Все метрики имеют плохие значения из чего мы можем сделать вывод, что данный классификатор является неэффективным

1. Построим классификатор типа случайный лес *(random forest classifier),* далее оценим его эффективность с помощью метрик *precision, recall* и *F1*:

* f1: 0.395
* precision: 0.493
* recall: 0.353

Метрика *precison* имеет чуть более высокое значение, чем в классификаторе *random forest*, но *F1* и *recall* значительно снизились. Этот классификатор тоже является крайне неэффективным

Выводы

Классификатор *random forest* показывает очень плохие значения для всех метрик, он крайне неэффективен.

Значения метрик классификатора *decision tree* выше чем у *random forest,* но все равно являются очень низкими. Этот классификатор тоже очень неэ5ффективен.

Приложение 1





