Задача 4

Условие

Набор данных: [BankChurners.csv](https://www.kaggle.com/sakshigoyal7/credit-card-customers)

Тип классификатора - *DecisionTreeClassifier* (решающее дерево)

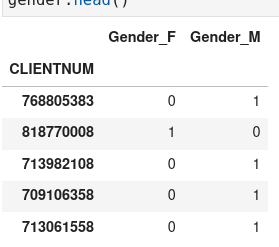
Классификация по столбцу - *Credit Limit* (выше 10 000 – класс 0, ниже или совпадает – класс 1)

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 BankChurners.csv из подготовим датасет к дальнейшему использованию классификатором:

* Удалим строки со значениями NaN и ‘unknown’
* Преобразуем поле *Credit\_Limit* согласно условию: 0, если значение поля >= 10000, 1 в противном случае
* Отделим *Credit\_Limit* от остальных данных, теперь это отдельный датафрейм
* Переведем категориальные переменные (*Education\_Level, Marital\_Status, Attrition\_Flag, Gender, Income\_Category, Card\_Category*) в One hot encoding, теперь каждая представляет собой отдельный датафрейм вида:



Присоединим эти переменные к основным данным

* Разделим датасет на тестовую и обучающую выборку со следующими параметрами: размер тестовый выборки составляет 0.33 от общей выборки, параметр *random\_state* равен 42 (необходим, чтобы при разных запусках блокнота данные выборок не менялись)

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

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

* *f1* ~ 0.996
* *precision* ~ 0.997
* *recall* ~ 0.996

Это невероятно хорошая метрика, мы можем назвать модель успешной

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

* *f1* ~ 0.994
* *precision* ~ 0.991
* *recall* ~ 0.995

Эту метрику тоже можно назвать хорошей, а модель успешной

Выводы

Оба классификатора показывают показывают очень хороший значения метрик (значение каждой метрики >= 0,99), но значения решающего дерева чуть лучше значений случайного леса

Приложение 1



