***Отчет по Task2***

Первый этап — подготовка данных. Все значение были считаны из *.txt* файла и переделаны в *Dataframe* для дальнейшего удобства. Значения предикторов, по всей видимости, нормированы в промежутке *[0, 1]*, а прогнозируемая величина принимает целочисленные значения. Изучив детальнее все данные, увидел, что y — не только целочисленная переменная, но и значения, принимаемые ею — ограничены. Из чего можно сделать вывод, что задача может быть отнесена к классификации, а не регрессии. Также оказаллось, что относительно сильная корреляция есть только между y и предикторами *x\_1, x\_2, x\_4, x\_5*, с остальными же переменными корреляция не только слабая, но и обратная. Поэтому возмем обратное отношение этих предикторов, чтобы все корреляции были прямыми. Очевидно, что после данной операции значения соотвествующих переменных будут уже больше 1, поэтому, возможно, стоит отнормировать эти значения с помощью *MinMaxScalar*. После анализа окончателных данных понял, что даже после корректировки некоторых предикторов ситуация с корреляцией особо не поменялась, поэтому эта задача точно относится к классфикации, и далее нужно исследовать различные алгоритмы для соотвествующей задачи. Чтобы оценить качество набора данных, изучим его сбалансированность. Распредение данны по классам оказалось весьма неравномерным, для устранения этой особенностиы будем использовать готовые решения (в виде параметров алгоритмов классификации, например *class\_weights*).

Второй этап — тестирование различных алгоритмов классификации. Начнем с *K Nearest Neighboors*. Все предикторы лежат в диапозоне от 0 до 1, поэтому, предположу, необязательно производить стандартизацию и масштабирование этих переменных. Соотвественно, не нужно строить pipeline, а можно сразу приступать к созданию класса модели и дальнейшего подбора параметров путем прохода по сетке гиперпараметров.