Здравствуйте.

При выполнении домашнего задания на тему **точности модели** возникли следующие вопросы:

По пунктам:

***1а. Проверка* наличия пропущенных значений. Заполнение**

Для логической регрессионной модели решил оставить признак – **год рождения спортсмена**. Для этого предварительно выполнил следующий код:

df['dob\_year'] = df['dob'].apply(lambda x: pd.to\_datetime(x, format='%m/%d/%y').year)

Новый столбец успешно создан. Но при внимательной проверке обнаружил, что по никоторым спортсменам год конвертировался некорректно. У 65 спортсменов из 11 538 год определялся как 2068, 2067, 2056... Такая особенность отмечается у всех спортсменов год рождения которых старше 1969 г.

Почему таким образом обрабатывается дата, почему именно с 1968 дата отрабатывается неверно, а с 1969 – все даты отображаются корректно? Так же пробовал использовать библиотеку datetime. Результат тот же!

### 1б. Закодируйте категориальные переменные числовыми значениями по необходимости.

Не смог OneHotEncoder закодировать категориальные переменные.

Причина: OneHotEncoder просит внести все признаки(столбцы) датафрейма. Мой датафрейм содержит не только категориальные данные, но и количественные. Отдельно создавать DF с нужными признаками, а потом соединять DF с количественными признаками? Существует ли способ чтоб OneHotEncoder кодировал нужные столбцы датафрейма как pd.get\_dummies?

## **2. Разделение на обучающую и тестовую выборки**

Правильно ли я понимаю, что параметр **stratify=y** обеспечивает одинаковую долю распределения целевого признака по тестовым и обучающим выборкам? А если целевой признак сильно не сбалансирован?

Верно ли было решение выполнить **стандартизацию всех признаков?** Проверка показала, что без стандартизации модель работает некорректно…

## **5. Реализуйте подсчет метрик TPR, FPR «вручную», без использования готовых функций из библиотеки sklearn.**

Выполнил задание в два варианта. С первым вариантом возникли сложности с логическим решением. Второй вариант выполнить с помощью формирования датафрейма со столбцами: probability, y\_predict, actual. Для меня почему-то более понятным. Графики совпали!

**7. Постройте график Precision-Recall**

Так же получилось реализовать только с помощью датафрейма. С помощью цикла график получался неверным не совпадал с графиком, который реализован с помощью встроенных функций metrics библиотеки sklearn. График совпал если я использовал с помощью датафрейма.

Так же возникли сложности при делении на ноль. При некоторых расчётах получалось что происходит деление на ноль. Как с этим бороться, что значит «занулить»?

**8. Вычислите значение ROC-AUC метрики, используя метрики, посчитанные в п. 5.**

Я так и не понял, как выполнить это задание вручную….

Очень надеюсь на ответы т. к. очень сильно поможет на исправление и дальнейшего понимания темы.