Постановка задачи

Сведения о задаче:

https://www.kaggle.com/competitions/competative-data-science-course-by-data-feeling/overview

Предыстория: существует каршеринговая компания с крупным автопарком машин в нескольких городах. Машины ежедневно совершают тысячи поездок. Люди очень любят этот сервис. А основатели довольны метриками. Чтобы так продолжалось и дальше, за машиной необходимо обеспечить надлежащий и тщательный уход. Своевременный технический осмотр или мелкий ремонт позволяет предотвратить перемещение машины с линии ежедневной аренды на длительный период.

Проблема: Однако не все так просто. Автопарк очень большой, люди активно изнашивают, а иногда бывают в поездках в очень агрессивной машине. В результате машины в один момент могут покинуть здание в силу определения поломок или неполадок. А техническая бригада автопарка не позволяет объезжать все машины каждый день, чтобы предотвратить поломки, проводить превентивные меры.

Идея: Главный бригадир по ремонту машин обратился в ИТ-отдел и предложил составить приоритизированный список обходов машин. Тогда тех. бригаде не придется объезжать все подряд, а было бы достаточно обойти только те машины, которые, скорее всего, выдут из здания раньше. Чтобы определить такой список, необходимо уметь прогнозировать вид поломки машины.

Решение: Как же повезло, что в компании этого автопарка уже давно есть команда Data Engineer'ов, которая аккуратно собирает все данные по поездкам. А это значит, что у нас есть исторические данные, в которых есть информация о состоянии машины, которая предшествовала поломкам. Более того, в этой команде появился специалист по данным, главный герой должен решить эту задачу. Роль этого Data Scientist'а будешь играть ты!

Данные:

https://github.com/a-milenkin/Competitive_Data_Science/tree/main/data

Файлы, необходимые для построения и оценки точности модели классификации:
1) *car_train.csv* (основные данные о машинах);
2) *rides_info.csv*, *driver_info.csv*, *fix_info.csv* (дополнительные данные о машинах, необходимо предварительное преобразование данных для объединения с основными сведениями).

Основные данные
Главное описание машин с информацией о поломках для обучения / прогноза –в файле *car_train.csv*.
Пояснение столбцов:

car_id – идентификатор машины;
model / *car_type* / *fuel_type* – марка, класс и тип топлива машины;
car_rating / *riders* – общий рейтинг и общее число поездок к концу 2021-го кода;
year_to_start / *year_to_work* – года выпуска машины и начала работы в автопарке;
main_city – город пребывания машины (Москва или Питер);
target_reg – время до поломки (отсутствует в тесте);
target_class – класс поломки (всего 9 видов).

Дополнительные данные
Помимо описания машин, есть немаловажные данные о поездках на этих машинах за период трех месяцев, ремонтных работах и данных водителей. То есть, есть еще три дополнительные таблицы в файлах:
rides_info.csv – информация про поездки;
driver_info.csv – информация про водителей;
fix_info.csv – информация про ремонт машин.

Описание дополнительных данных
Информация про поездки – *rides_info.csv*.
Пояснение столбцов:

user_id / *car_id* / *ride_id* – идентификаторы водителя, машины, поездки соответственно;
ride_date / *rating* – дата поездки и рейтинг, поставленный водителем;
ride_duration / distance / ride_cost – длительность (время), пройденное расстояние, стоимость поездки;
speed_avg / speed_max – средняя и максимальная скорости поездки соответственно;
stop_times / refueling – количество остановок (паузы) и флаг - была ли дозаправка;
user_ride_quality – оценка манеры вождения в машины водителя, определенная скоринговой ML системой сервиса;
deviation_normal – общий показатель датчиков о состоянии машины, относительно эталонных показателей (нормы).

Информация про водителей – *driver_info.csv*.
Пояснение столбцов:

user_id / *age* / *sex* – идентификатор, возраст и пол водителя соответственно;
user_rating – общий рейтинг пользователя за все поездки к концу 2021-го года;
user_rides – общее количество поездок к концу 2021-го кода;
user_time_accident – число инцидентов (это могли быть аварии/штрафы/эвакуация машины);
first_ride_date – дата первой поездки;

Информация про ремонт машин – *fix_info.csv*.
Пояснение столбцов:

worker_id / *car_id* – идентификатор работника и машины соответственно;
work_type / *work_duration* – тип и длительность (в часах) проводимой работы;
destroy_degree – степень износа/поврежденности машины в случае поломки;
fix_date – время начала ремонта (время снятия машины с линии).

Что надо сделать?

1. Выбрать модель многоклассовой классификации
2. Построить модель классификации по основным данным и оценить качество модели с помощью метрики AUC. Убедиться, что качество модели плохое.
3. Построить модель классификации, используя основные данные и хотя бы один файл из дополнительных данных. Оценить качество модели с помощью метрики AUC. Убедиться, что качество модели улучшилось.

!!! Задание можно выполнять командой до 5 чел.

В качестве ответа необходимо прикрепить блокнот с решением (ссылку на блокнот или файл формата .ipynb), в комментарии явно указать значение метрики AUC.