1. Вычисление счетчиков

Довольно прямолинейная задача,

Категориальные признаки: medallion, hack_licence, vendor_id, rate_code, store_and_fwd_flag, passenger_count, pickup_place_id, dropoff_place_id, payment_type.

Для каждой строчки каждый из них преобразовывается в запись вида

KEY: (feature_index, feature_value)

VALUE: (

tip_amount,

fare_amount,

tip_amount/fare_amount,

tip_amount/fare_amount > .25,

1

)

На reduce-стадии, соответственно, все поля банально складываются.

Затем, уже в ML-коде каждый категориальный признак каждой строки заменяется на 4 числа, выведенные из счетчиков. Пусть текущие значения счетчика это TIP, FARE, RATIO, GOOD, COUNT (это суммы соответствующих значений для каждой строки с тем же значением текущего признака. GOOD, таким образом, это количество строк с данным значением текущего признака, в которых tip_amount/fare_amount > .25). Тогда новые признаки следующие:

TIP / FARE

RATIO / COUNT

GOOD / COUNT

TIP / COUNT / fare_amount

Таким образом, усреднения значений по строкам с тем же значением признака производится разными способами. Тип поля везде выходит некоторым отношением tip к fare.

2. ML

Поля pickup_datetime, dropoff_datetime удалены. Оставшиеся некатегориальные поля используются сырыми как числа.

Используется решающий лес из sklearn. Локально для выбора модели используется обычный score, а не AUC. Перебираемые параметры модели:

n_estimators

max_features

max_depth

criterion

Долгий и мучительный grid search дал следующие результаты:

1) max_features оптимально брать равным 12, умеренно-независимо от других параметров

2) max_depth тоже показывает улучшение качества при росте до 12 и ухудшение после

3) n_estimators на всём диапазоне даёт улучшенеи качества с ростом, однако и очень сильно увеличивает время обучения модели

4) criterion=gini равномерно лучше, чем entropy

После выбора max_features и max_depth при помощи GridSearchCV, n_estimators удваивался до тех пор, пока обычный i3 12GB мог считать модель за субъективно-конечное время. Начиная с 72 и до 576.

Финальная используемая модель:

```
clf=RandomForestClassifier(
    n_estimators=576,
    max_features=12,
    max_depth=16,
    random_state=80085,
    n_jobs=3
)
```