Spotify tracks popularity prediction

Решение Ани

Типы признаков:

- NUMERICAL_FEATURES = ["danceability", "energy", "loudness", "speechiness", "mode", "acousticness", "instrumentalness", "liveness", "tempo", "duration_ms", "time_signature"] и новые добавленные
- CAT_FEATURES = ["composer", "album", "time_signature", "key", "tonality"]

Обработка и построение признаков

- Tonality = Key + “:” + Mode — эта пара описывает тональность трека
- Выделяем группы: groups = [["album"], ["composer"], ["album", "composer"]]
  - Склеиваем X_all = X_train + X_test
  - В рамках каждого элемента группы (group) считаем статистики по X_all для всех NUMERICAL_FEATURES
  - Статистики: mean, std, median
  - X_all.groupby(group).agg({f: [“mean”, “std”, “median”] for f in NUMERICAL_FEATURES}
  - Добавляем посчитанные статистики отдельными признаками в рамках своей group

Обработка и построение признаков

- Для каждого признака f из NUMERICAL_FEATURES:
  - Добавляем признак log(abs(f) + 1e-6)
  - Считаем разницу, модуль разницы, отношение значения признака и его среднего значения в рамках каждой группы из groups
- Для всех пар различных признаков из NUMERICAL_FEATURES:
  - Добавляем их произведение и отношение
- Из тренировочного множества удаляем те пары (composer, album), для которых их средняя популярность по X_train = 0
  - Спойлер: кажется это немного привело к оверфиту
  - Удаление применялось только для ML-решений

Решение 1. CatboostRegressor

- X_train_part_reg, X_val_reg, y_train_part_reg, y_val_reg = train_test_split(X_train, y_train, test_size=0.05)
- regr = CatBoostRegressor(cat_features=CAT_FEATURES, iterations=2000, depth=6)
- regr.fit(X_train_pool_reg, eval_set=X_val_pool_reg)

Решение 2. RandomForest

- CAT_FEATURES преобразовывались в вещественные при помощи счетчиков
- RandomForestRegressor(n_estimators=50, max_depth=13)

Решение 3. CatBoostClassifier + CatBoostRegressor

- Раз ошибка предсказания находится около 7, то можно разбить таргет на бины и попробовать предсказывать их
- Бины:
  - y < 5 => 0
  - y >= 5 && y < 10 => 1
  - y >= 10 && y < 15 => 2
  - y >=15 => y // 15 + 2
- Значения y распределены экспоненциально, поэтому на маленьких значениях бины более частые
- Многоклассовая классификация CatBoostClassifier(cat_features=CAT_FEATURES, iterations=300, depth=6, loss_function="MultiClass")

Решение 3. CatBoostClassifier + CatBoostRegressor

- Делим X_train на несколько тренировочных датасетов в соответствии с бинами
- На каждом датасете обучаем CatBoostRegressor(cat_features=CAT_FEATURES, iterations=300, depth=6)
- Предсказание: сначала предсказываем класс, затем соответствующим данному классу регрессору предсказываем популярность

Решение 4. “Умное” среднее

- Считаем средний таргет (по трейну) по каждой паре (альбом, композитор)
- Предсказываем посчитанное среднее значение для каждой пары (если пары в тренировочной выборке не было, то ставилось среднее от множества ненулевых средних популярностей по (альбом, композитор))

Скоры

- “Умное” усреднение —
  - Если в “умном” среднем был 0, то и итоговое предсказание тоже было 0
  - Делалась срезка, что если предсказание получалось < 0, то оно выставлялось в 0

Решение	**Паблик**	**Приват**
CatboostRegr	8	8.47
RandomForest	7.57	8.02
CatboostClf + CatboostRegr	8.18	8.87
“Умное” среднее	7.56	8.03
“Умное” усреднение всех 4 предсказаний	7.17	7.72

Пример предсказаний

	index	popularity_clf	popularity_stupid	popularity_regr	popularity_rf	popularity
0	0	2.604066	8.380952	8.656034	6.951776	6.648207
1	4	2.556066	0.000000	1.945420	8.854114	0.000000
2	8	3.017787	5.166667	4.733197	6.085935	4.750896
3	9	0.624451	1.888889	10.168770	2.046442	3.682138
4	13	3.206935	7.625000	4.594223	10.373066	6.449806
...	...	...	...	...	...	...
31008	19390	79.011786	83.000000	72.040898	76.843604	77.724072
31009	19777	79.041250	68.083333	64.554687	66.578157	69.564357
31010	21780	79.268125	83.000000	77.558160	77.407968	79.308563
31011	25715	79.006346	69.666667	67.414782	71.255579	71.835843
31012	25873	79.181500	75.250000	74.609364	78.593904	76.908692

Решение Сережи

- “Умное” среднее + отсечение по нулю – 7.56 на паблике

- Дополнительные признаки: число альбомов / треков у композитора, число треков в альбоме
- Важная идея 1: Будем считать статистики по конкатенации трейн+тест
- Агрегации признаков с плавающей точкой: минимум, максимум, среднее, стандартное отклонение
- ~ 70 признаков

Диффы

- Важная идея 2: будем предсказывать диффы между “умным средним”

- Паблик 7.32577, приват 7.80463

Итоговое решение авторов

- Усреднение решений Ани и Сережи + отсечение по нулю

- 7.18032 паблик, 7.69604 приват

- 2 дня не очень активного решения на каждого :)