Соковнин Игорь Леонидович, факультет Искусственного интелекта

Практическое задание к 3 уроку: Урок 3. Построение модели классификации.

1. **Для чего и в каких случаях полезны различные варианты усреднения для метрик качества классификации: micro, macro, weighted?**

Усреднение множественных метрик

Одним из важных вопросов построения метрик в случае классификации и фильтрации, является метод усреднения результатов. Этот вопрос зачастую остается не освещенным, несмотря на значительное влияние его на результаты оценки [3].

При микро-усреднении сначала характеристики усредняются по всем классам, а затем вычисляется итоговая двухклассовая метрика

При макро-усреднении сначала вычисляется итоговая метрика для каждого класса, а затем результаты усредняются по всем классам [4].

Хотя каждый метод усреднения имеет свои преимущества, одним из общих соображений при выборе соответствующего метода является дисбаланс класса.[1]

Наиболее часто используемой метрикой для оценки качества мультиклассовой классификации для несбалансированных наборов данных, является мультиклассовый вариант f-меры. Идея, лежащая в основе мультиклассовой f-меры, заключается в вычислении одной бинарной f-меры для каждого класса, интересующий класс становится положительным, а все остальные – отрицательными классами. Затем эти f-меры для каждого класса усредняются с использованием одной из следующих стратегий:

• "**macro**" усреднение вычисляет f-меры для каждого класса и находит их невзвешенное среднее. Всем классам, независимо от их размера, присваивается одинаковый вес.

• "**weighted**" усреднение вычисляет f-меры для каждого класса и находит их среднее, взвешенное по поддержке (количеству фактических примеров для каждого класса). Эта стратегия используется в классификационном отчете по умолчанию.

• "**micro**" усреднение вычисляет общее количество ложно положительных примеров (ложных срабатываний), ложно отрицательных примеров (ложных негативов) и истинно положительных примеров по всем классам, а затем вычисляет точность, полноту и f-меру с помощью этих показателей [2].

Если необходимо присвоить одинаковый вес каждому примеру, рекомендуется использовать микро-усреднение f1-меры.

Если вам необходимо присвоить одинаковый вес каждому классу, рекомендуется использовать макро-усреднение f1-меры, в остальных случаях weighted-усреднение.

Макроусреднение (macroaverage) характерно для оценки задач поиска, в которых важен результат в среднем по запросу, независимо от мощности ответа на этот запрос.

Микроусреднение (microaverage) нашло большее применение в оценке классификации и фильтрации, где необходимо учитывать «размеры» классов [3].

1. Оценка результатов автоматического эксперимента машинного обучения https://docs.microsoft.com/ru-ru/azure/machine-learning/how-to-understand-automated-ml

2. Андреас Мюллер, Сара Гвидо Введение в машинное обучение с помощью Python М.: 2016-2017, с. 322

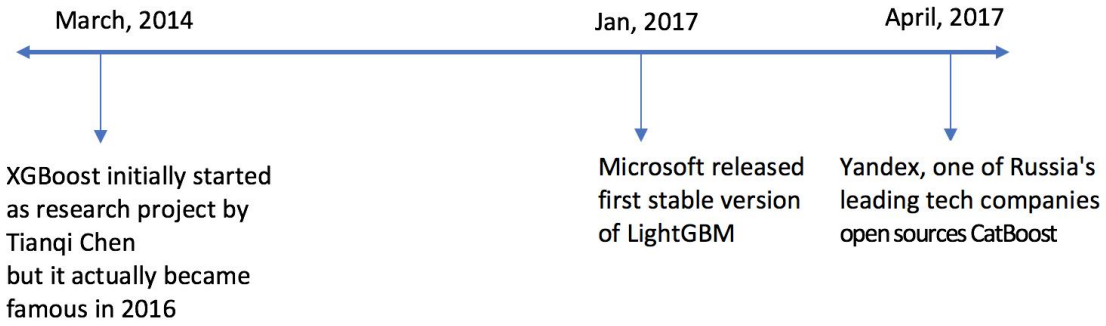
3. М. Агеев, И. Кураленок Официальные метрики РОМИП’2004, с. 3 - http://romip.ru/docs/romip_metrics.pdf

4. http://www.machinelearning.ru/wiki/images/1/1c/sem06_metrics.pdf, с.8-9

Библиотеки Python для Data Science: продолжение.

Урок 3. Построение модели классификации.

2. В чём разница между моделями xgboost, lightgbm и catboost или какие их основные особенности?



Структурные различия в LightGBM и XGBoost

XGBoost — это наиболее широко используемый бесплатный пакет программ с реализацией стохастического градиентного бустинга, который первоначально был разработан Тьянси Ченом (Tianqi Chen) и Карлосом Гестрином (Carlos Guestrin) в Вашингтонсоком университете [4].

LightGBM использует новую технику односторонней выборки на основе градиента (a novel technique of Gradient-based One-Side Sampling (GOSS)) для фильтрации экземпляров данных для нахождения значения разделения, в то время как XGBoost использует предварительно отсортированный алгоритм и алгоритм на основе гистограммы (presorted algorithm & Histogram-based algorithm) для вычисления наилучшего разделения. [1].

Как каждая модель относится к категориальным переменным?

CatBoost

CatBoost отлично работает **с категориальными функциями**:

Позволяет задавать индексы категориальных столбцов, с использованием one_hot_max_size (используйте кодирование в одно касание для всех функций с числом различных значений, меньшим или равным данному значению параметра).

Если ничего не передается в аргументе cat_features, CatBoost будет обрабатывать все столбцы как числовые переменные.

Для оставшихся категориальных столбцов, которые имеют уникальное количество категорий, превышающее one_hot_max_size, CatBoost использует эффективный метод кодирования, который аналогичен среднему кодированию, но уменьшает переопределение.

LightGBM

Как и в CatBoost, LightGBM также **может обрабатывать категориальные функции**, вводя имена функций. Он не конвертируется в одноразовое кодирование и намного быстрее, чем одноразовое кодирование. LGBM использует специальный алгоритм, чтобы найти значение разделения категориальных признаков.

XGBoost

В отличие от CatBoost или LGBM, XGBoost не может обрабатывать категориальные функции сам по себе, он **принимает только числовые значения**. Поэтому перед подачей категориальных данных в XGBoost необходимо выполнить различные кодировки, такие как кодирование меток, среднее кодирование или однократное кодирование.

Библиотеки Python для Data Science: продолжение.

Урок 3. Построение модели классификации.

Сходство в гиперпараметрах

Все эти модели имеют множество параметров для настройки, но мы рассмотрим только самые важные. Ниже приведен список этих параметров в соответствии с их функциями и их аналогами в разных моделях.

Function	XGBoost	CatBoost	Light GBM
Important parameters which control overfitting	1. **learning_rate or eta** – optimal values lie between 0.01-0.2 2. **max_depth** 3. **min_child_weight:** similar to min_child leaf; default is 1	1. **Learning_rate** 2. **Depth** - value can be any integer up to 16. Recommended - [1 to 10] 3. No such feature like min_child_weight 4. **l2-leaf-reg**: L2 regularization coefficient. Used for leaf value calculation (any positive integer allowed)	1. **learning_rate** 2. **max_depth**: default is 20. Important to note that tree still grows leaf-wise. Hence it is important to tune **num_leaves** (number of leaves in a tree) which should be smaller than 2^(max_depth). It is a very important parameter for LGBM 3. **min_data_in_leaf**: default=20, alias= min_data, min_child_samples
Parameters for categorical values	Not Available	1. **cat_features**: It denotes the index of categorical features 2. **one_hot_max_size**: Use one-hot encoding for all features with number of different values less than or equal to the given parameter value (max – 255)	1. **categorical_feature:** specify the categorical features we want to use for training our model
Parameters for controlling speed	1. **colsample_bytree**: subsample ratio of columns 2. **subsample**: subsample ratio of the training instance 3. **n_estimators**: maximum number of decision trees; high value can lead to overfitting	1. **rsm:** Random subspace method. The percentage of features to use at each split selection 2. No such parameter to subset data 3. **iterations**: maximum number of trees that can be built; high value can lead to overfitting	1. **feature_fraction:** fraction of features to be taken for each iteration 2. **bagging_fraction:** data to be used for each iteration and is generally used to speed up the training and avoid overfitting 3. **num_iterations:** number of boosting iterations to be performed; default=100

CatBoost supports the following types of features [2]:

- **Numerical**. Examples are the height (“182”, “173”), or any binary feature (“0”, “1”).
- **Categorical** (cat). Such features can take one of a limited number of possible values. These values are usually fixed. Examples are the musical genre (“rock”, “indie”, “pop”) and the musical style (“dance”, “classical”).
- **Text**. Such features contain regular text (for example, “Music to hear, why hear'st thou music sadly?”).

XGBoost принимает только числовые данные.

1. https://www.machinelearningmastery.ru/catboost-vs-light-gbm-vs-xgboost-5f93620723db/, оригинал https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db.

2. Overview of CatBoost. Transforming categorical features to numerical features. - https://catboost.ai/docs/concepts/algorithm-main-stages_cat-to-numberic.html#algorithm-main-stages_cat-to-numberic

3. Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, Andrey Gulinю CatBoost: unbiased boosting with categorical features 2019 - https://arxiv.org/abs/1706.09516

4. Брюс, П. Практическая статистика для специалистов Data Science. Пер. с англ. СПб.: БХВ-Петербург, 2018. с. 254