


САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Направление: 02.03.02 «Фундаментальная информатика и информационные технологии»

ООП: Программирование и информационные технологии

ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Тема задания: Методы анализа социально-демографических характеристик аудитории социальных сетей

Выполнила: Тарелкина Анастасия Александровна 

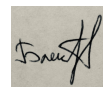
20.Б11-пу

Фамилия И. О.

номер группы

Руководитель научно-исследовательской работы: доцент кафедры технологии программирования, кандидат технических наук, Блеканов Иван Станиславович

должность, ученая степень, ФИО



Санкт-Петербург

2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Постановка задачи	5
1. Теоретическая часть	6
1.1. Обзор подходов к решению задачи	6
1.2. Метрики	6
1.3. Выбор модели	8
2. Практическая часть	11
2.1. Выгрузка и предобработка данных	11
2.2. Обучение алгоритмов	14
2.3. Результаты	15
ЗАКЛЮЧЕНИЕ	17
Список использованных источников	18

ВВЕДЕНИЕ

В современном мире социальные сети являются важнейшим сегментом интернета, привлекающим миллионы пользователей, поэтому для многих компаний размещение рекламы в социальных сетях – это одно из необходимых условий развития бизнеса. Однако даже качественный продукт необходимо продвигать тем, кому он понадобится.

В последние годы такое направление, как машинное обучение, получило развитие и широкое применение во многих областях. Не стала исключением и область сетевого маркетинга. Рекламодатели, как правило, хотят продвигать свои продукты, ориентируясь на целевую аудиторию, имеющую определенные социально-демографические признаки. Таким образом, компании могут привлечь больше клиентов и увеличить свою прибыль. Анализируя большие объемы данных о пользователях, можно выделить подходящий сегмент аудитории для каждого рекламного предложения. Для отбора аудитории социальной сети, удовлетворяющей предъявляемым к целевой аудитории требованиям, необходимо по имеющимся данным о пользователе предсказать его социально-демографические признаки.

Демографические данные – это информация о пользователях, такая как возраст, пол и место жительства. Необходимые для анализа данные могут также включать в себя и социально-экономические факторы, такие как профессия, семейное положение или доход. Демографические данные и интересы относятся к числу наиболее важных статистических данных веб-аналитики и анализа поведения потребителей.

Благодаря наличию больших объемов данных, задача определения демографических признаков пользователя может быть решена с использованием методов машинного обучения.

Одним из важнейших демографических признаков является возраст, который значительная часть пользователей предпочитает скрывать. Сегментация рынка по возрасту является важным видом сегментации из-за того, что представители разных возрастных групп значительно различаются между собой. На аудиторию определенного возрастного сегмента может быть ориентирован как продукт в целом, так и определенный вид рекламы.

В данной работе решение задачи определения демографических признаков пользователя рассматривается на примере задачи предсказания возраста по имеющимся данным о пользователе.

Постановка задачи

Целью данной работы является нахождение алгоритма, способного определять возрастной сегмент, к которому относится пользователь, с приемлемой точностью.

Для достижения цели работы были поставлены следующие задачи:

1. Рассмотреть существующие подходы к решению задачи предсказания демографических характеристик.
2. Собрать достаточное количество данных
3. Предобработать данные, исследовать полученные признаки
4. Сравнить различные современные подходы, адаптировать их к данной постановке задачи путем подбора параметров;

Поскольку для исследования аудитории социальной сети определение точного возраста пользователей не является необходимым, достаточно определять, к какой возрастной группе относится пользователь. Поэтому в терминах машинного обучения задачу можно определить, как задачу многоклассовой классификации.

1. Теоретическая часть

1.1. Обзор подходов к решению задачи

Существует несколько способов предсказания возраста по профилю пользователя.

1. Использовать текстовые признаки (статус, посты и т.д.) [1]
2. Использовать методы распознавания изображений (фотографии с профиля) [2]
3. Использовать социальные связи пользователя [3]
4. Использовать информацию из профиля пользователя.

1.2. Метрики

Для того, чтобы оценить качество моделей классификации и сравнить их между собой, необходимо выбрать метрики качества.

Перед тем как переходить к метрикам классификации, следует привести возможные варианты соотношения предсказанного и истинного класса:

1. True Positive (TP) – классификатор верно отнес объект к нужному классу.
2. True Negative (TN) – классификатор верно не отнес объект в определенному классу.
3. False Positive (FP) – классификатор ошибся и отнес объект к неверному классу.
4. False Negative (FN) – классификатор ошибся и утверждает, что объект не относится к некоторому классу, хотя он на самом деле относится.

Precision (Точность) – это метрика, показывающая долю объектов, названными классификатором положительными и при этом действительно являющимися положительными.

Recall (Полнота) – это метрика, показывающая, какую долю объектов положительного класса алгоритм отметил как положительные.

Расчет метрик в нашем случае усложняется наличием нескольких классов. Так как задача классификации ставится как задача отделения одного класса от остальных, то существует два варианта получения итогового значения метрики из матриц ошибок [4]:

1. Усредняются элементы матрицы ошибок (TP, FP, TN, FN) между бинарными классификаторами, например, для True Positive по следующей формуле:

$$TP = \frac{1}{k} \sum_{i=1}^k TP_i.$$

Затем по одной усреднённой матрице ошибок считаются Precision, Recall. Такой подход называется микроусреднением.

2. Считаются Precision, Recall для каждого класса отдельно, а потом усредняются. Такой подход называется макроусреднением.

Стоит обратить внимание на то, что в первом случае Precision будет равен Recall [4]. Поэтому метрики Precision и Recall в данной работе рассчитываются вторым способом. Precision и Recall для одного класса вычисляются по следующей формуле:

$$Precision_k = \frac{TP_k}{TP_k + FP_k}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k}$$

Затем происходит усреднение метрик по классам следующим образом:

$$MacroAveragePrecision = \frac{\sum_{k=1}^K Precision_k}{K}$$

$$MacroAverageRecall = \frac{\sum_{k=1}^K Recall_k}{K}$$

Перечисленные метрики для выбранных моделей сравнивались с Baseline методами, например, с предсказанием наиболее популярного класса, для любых данных и случайным выбором класса.

1.3. Выбор модели

Для решения задачи многоклассовой классификации существует большое количество различных методов. Поэтому нужно выбирать модели, опираясь на специфику задачи. В задаче предсказания возраста по признакам профилей в социальной сети данные имеют большое количество категориальных признаков, также могут встречаться выбросы. Кроме того, как будет показано далее в работе, линейная зависимость между признаками и возрастом крайне незначительна, поэтому алгоритм классификации должен выделять нелинейные зависимости. Поэтому основной моделью обучения были выбраны решающие деревья.

Решающее дерево [5] предсказывает значение целевой переменной с помощью применения последовательности простых решающих правил (предикатов). Деревья решений нечувствительны к выбросам, поскольку разделение происходит на основе доли выборок в пределах диапазонов разделения, а не абсолютных значений.

Обобщающая способность (способность выделять закономерности из данных) решающих деревьев невысока, их предсказания вычисляются довольно просто, из-за чего решающие деревья часто используют как базовые модели для построения ансамблей – моделей, делающих предсказания на основе агрегации предсказаний других моделей.

Ансамблевые методы – это методы машинного обучения, которые объединяют несколько базовых моделей, чтобы создать одну оптимальную

модель. Система на основе ансамбля получается путем объединения различных моделей.

Наиболее популярными ансамблевыми методами являются бэггинг и бустинг.

1. Бэггинг

Основная идея бэггинга заключается в том, чтобы обучить несколько одинаковых моделей на разных выборках. Поскольку выборки генерируются случайным образом, деревья тоже получаются разными в процессе обучения. Процесс генерации подвыборок с помощью семплирования с возвращением называется бутстрепом (bootstrap), а модели часто называют базовыми.

2. Random Forest

Случайный лес – это ансамбль деревьев решений, которые обычно обучены посредством метода бэггинга [5].

Чтобы получить предсказание ансамбля на тестовом объекте, усредняются отдельные ответы деревьев (для регрессии) или берется самый популярный класс (для классификации).

3. Градиентный бустинг

Бустинг (boosting) – это ансамблевый метод, в котором строится множество базовых алгоритмов из одного семейства, объединяющихся затем в более сильную модель. Базовые алгоритмы обучаются последовательно.

Каждый следующий базовый алгоритм в бустинге обучается так, чтобы уменьшить общую ошибку всех своих предшественников.

Алгоритм градиентного бустинга является достаточно популярным, и имеет несколько реализаций в различных пакетах для Python: LightGBM, XGBoost и CatBoost. В статье [6] проведен подробный сравнительный анализ этих алгоритмов. По результатам (рис 1.1) можно сделать вывод о том что CatBoost превосходит другие реализации по качеству работы с различными наборами данных.

	CatBoost	LightGBM	XGBoost
Adult	0.270 / 0.127	+2.4% / +1.9%	+2.2% / +1.0%
Amazon	0.139 / 0.044	+17% / +21%	+17% / +21%
Click	0.392 / 0.156	+1.2% / +1.2%	+1.2% / +1.2%
Epsilon	0.265 / 0.109	+1.5% / +4.1%	+11% / +12%
Appetency	0.072 / 0.018	+0.4% / +0.2%	+0.4% / +0.7%
Churn	0.232 / 0.072	+0.1% / +0.6%	+0.5% / +1.6%
Internet	0.209 / 0.094	+6.8% / +8.6%	+7.9% / +8.0%
Upselling	0.166 / 0.049	+0.3% / +0.1%	+0.04% / +0.3%
Kick	0.286 / 0.095	+3.5% / +4.4%	+3.2% / +4.1%

Рисунок 1.1: Сравнение CatBoost с другими популярными реализациями алгоритма (logloss / zero-one loss)

Поскольку решается задача многоклассовой классификации, в качестве функции потерь, которая оптимизируется при обучении будет использоваться перекрёстная энтропия.

Идея перекрёстной энтропии состоит в том, чтобы минимизировать расхождение между реальными метками классов и предсказанными вероятностями модели. Данная функция потерь вычисляется по следующей формуле:

$$- \sum_{n=1}^n \sum_{j=1}^c y_{i,j} \log(p_{i,j})$$

где:

- $y_{i,j}$ – истинная метка принадлежности примера j к классу i , в виде метки one-hot-encoding (равно 1 для истинного класса и 0 для остальных классов);
- $p_{i,j}$ – предсказанная вероятность принадлежности примера j к классу i ;
- c – количество примеров в обучающей выборке;
- n – количество классов.

2. Практическая часть

2.1. Выгрузка и предобработка данных

Для исследования было решено использовать данные социальной сети ВКонтакте, так как данная социальная сеть является достаточно крупной, ее используют миллионы человек, а также в этой соцсети пользователь может указать достаточно много разных данных о себе в профиле. Данные о пользователях выгружались с помощью VK API - интерфейса, который позволяет получать информацию из базы данных с помощью HTTP-запросов к серверу.

Выгрузка и обработка данных производилась скриптами на языке Python. Для отправки запросов к VK API использовалась библиотека requests, а для хранения и обработки данных – библиотека Pandas.

При работе с VK API возникло несколько проблем:

1. у API есть лимиты на количество запросов в секунду;
2. Токен для аутентификации в API действителен только 24 часа;
3. Данные выгружаются в виде Json, и для хранения в табличном виде их нужно распарсить.

Для решения проблемы с ограничением на число запросов, при получении ошибки API в ответ на запрос, запрос отправлялся еще раз через 3 секунды. Но, из-за ограничений на количество запросов, выгружать достаточно большое количество данных за время, которое действителен токен, не удавалось. Поэтому, во избежания дублирования данных при случайном выборе была использована следующая схема выгрузки:

1. Весь диапазон от 0 до 800000000, в котором находятся ID пользователей, был разбит на 80 равных промежутков;
2. Из каждого промежутка выбирались 10000 случайных ID пользователей, по ним делались запросы данных профиля;

Благодаря такой процедуре удалось выгрузить равное количество пользователей из каждого промежутка, избежав дублирования данных.

Ответ API в виде Json с несколькими уровнями вложенности преобразовывался к виду строки в таблице, при этом вложенные поля разворачивались в плоскую структуру и также включались в таблицу.

Всего удалось выгрузить 800000 профилей пользователей, из них около 50% не имели заполненной даты рождения, что еще раз доказывает необходимость предсказания возраста таких пользователей. Из выборки были удалены заблокированные и удаленные аккаунты, а также аккаунты, последняя активность которых была более 5 лет назад. Для обучения и тестирования использовались только профили с заполненной датой рождения, итоговый объем датасета составил около 220000 строк.

В датасете присутствуют признаки заполнения профиля пользователя, которые можно разделить на несколько типов:

1. Категориальные признаки (пол, религия, наличие детей и т.д.)
2. Количественные признаки (счетчики количества друзей, групп и т.д.)
3. Текстовые признаки (поля в профиле, которые заполняются в свободной форме, такие как мировоззрение, интересы и т.д.)
4. Технические признаки (ID пользователя, размер фотографии и т.д.)

Поскольку алгоритмы классического машинного обучения не могут напрямую работать с текстовыми признаками, данные признаки были заменены признаками, показывающими, заполнено ли пользователем соответствующее поле. В следующих работах будет рассмотрен вопрос обработки текстовых признаков моделями естественного языка. Технические признаки также не были использованы.

Была исследована корреляция между признаками и целевой переменной. Коэффициенты корреляции оказались достаточно малы, максимум по абсолютному значению составил 0.12, а среднее по абсолютным

значениям – 0.03, что свидетельствует о низкой линейной взаимосвязи возраста и признаков из профиля пользователя. Поэтому для классификации использовались не линейные модели, а ансамблевые методы, которые способны выявлять нелинейные зависимости.

Данные о дате рождения были преобразованы в возраст. Как видно на *Рисунке 2.1*, возраста свыше 60 лет встречаются редко и не попали в промежуток

$$[Q1 - 1.5 \cdot IQR; Q3 + 1.5 \cdot IQR],$$

где $Q1$, $Q3$ – первый и третий квартили соответственно, а IQR – межквартильный размах.

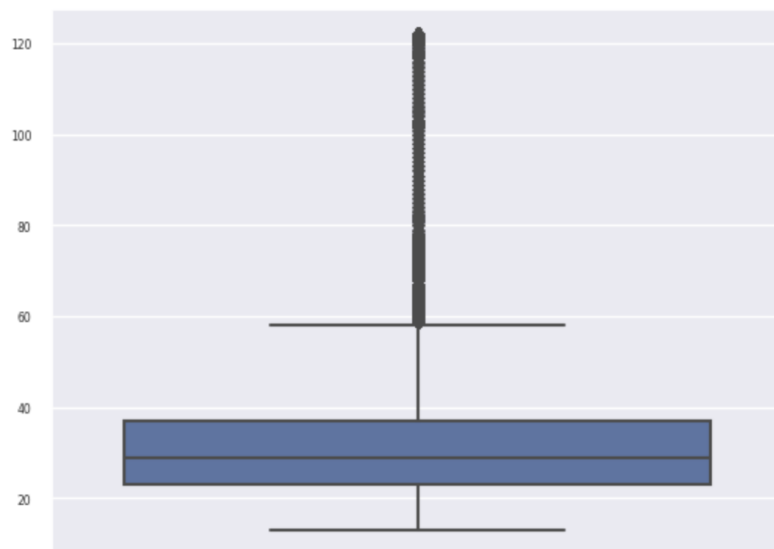


Рисунок 2.1: График распределения возраста

Поэтому возраста свыше 60 лет были признаны выбросами и исключены из выборки. Возраст был преобразован в возрастные группы: 14 – 18, 19 – 24, 25 – 35, 36 – 60 лет. Как можно заметить на *Рисунке 2.2*, классы получились несбалансированными в силу естественного распределения возрастов.

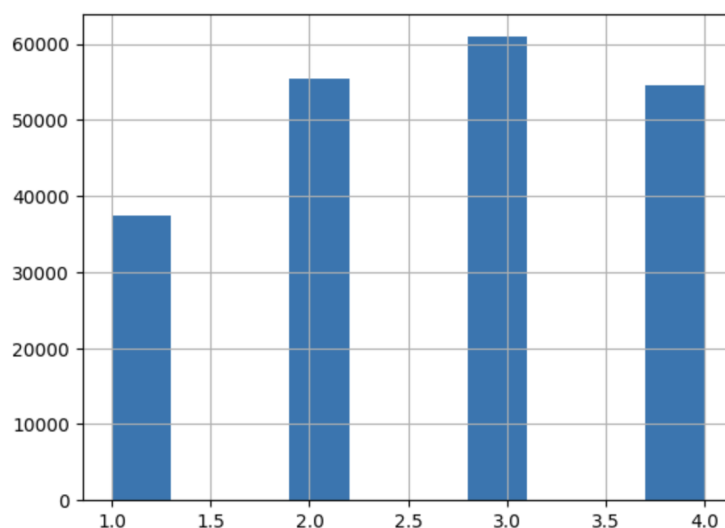


Рисунок 2.2: Распределение возрастных групп

Были проведены эксперименты с балансировкой классов с помощью оверсемплинга, однако на качество моделей это не оказало существенного влияния. Количественные признаки также были отмасштабированы путем деления на стандартное отклонение и вычитания среднего значения.

Для обучения и тестирования моделей датасет был разбит на обучающее и тестовое множество, размер тестового множества составил 25% от всего датасета.

2.2. Обучение алгоритмов

Для решения задачи классификации пользователей были протестированы алгоритмы Random Forest в реализации из библиотеки scikit-learn и градиентный бустинг в реализации CatBoost. В качестве метрик качества классификации использовались метрики Precision и Recall, рассчитанные для каждого класса и усредненные.

Подбор гиперпараметров алгоритмов производился с помощью поиска по сетке параметров. Для каждого сочетания параметров из заданного диапазона с помощью кросс-валидации обучался алгоритм и тестировалось его качество. Для алгоритма Random Forest были подобраны следующие значения гиперпараметров: `n_estimators: 1200`, `min_samples_split: 10`, `min_samples_leaf: 2`, `max_features: auto`, `max_depth: 30`, `bootstrap: True`. Для

CatBoost были получены параметры border_count: 100, depth: 6, l2_leaf_reg: 1, learning_rate: 0.1.

2.3. Результаты

Метрики алгоритмов с гиперпараметрами по умолчанию, оптимизированными значениями гиперпараметров и значения, при использовании Baseline методов представлены в *Таблице 2.1*.

Модель	Precision	Recall
Random Forest	0.426	0.424
Optimized Random Forest	0.444	0.432
CatBoost	0.455	0.454
Optimized CatBoost	0.456	0.456
Baseline (Popular class)	0.073	0.25
Baseline (Random)	0.253	0.253

Таблица 2.1: Сравнение метрик качества алгоритмов

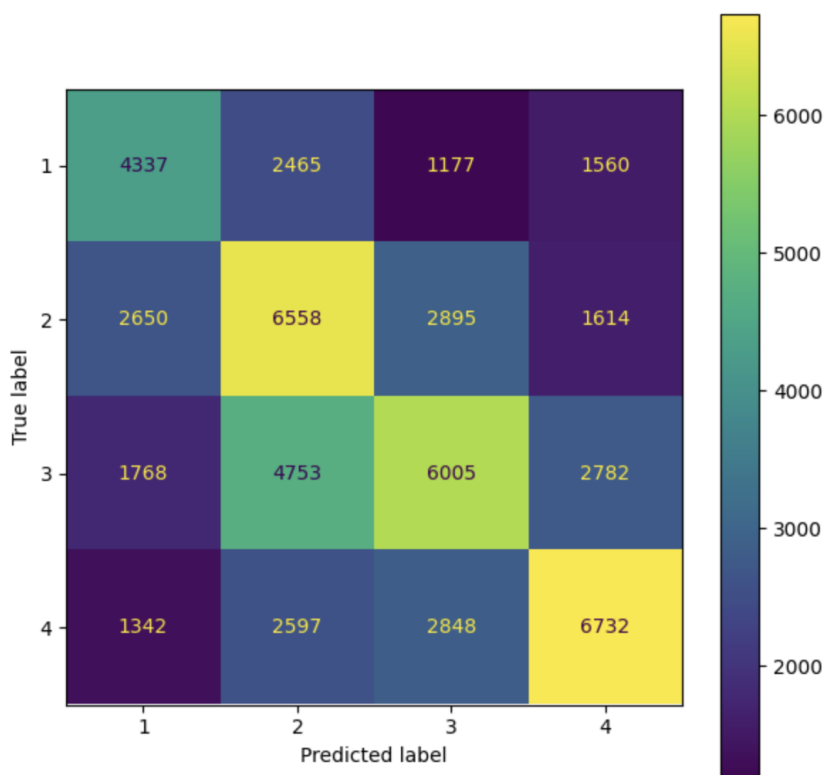


Рисунок 2.3: Confusion matrix модели Optimized CatBoost

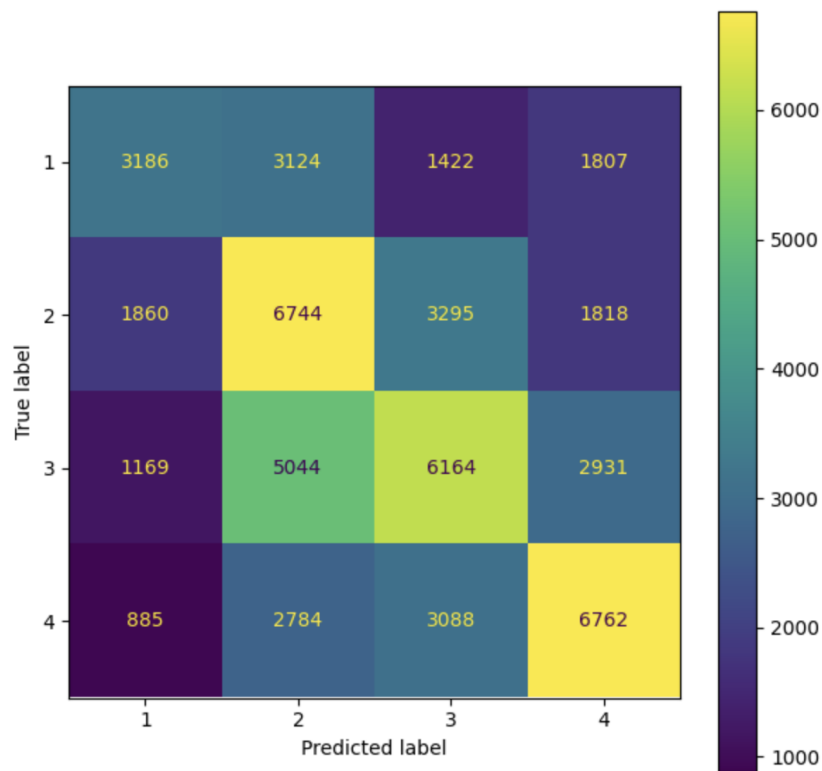


Рисунок 2.4: *Confusion matrix модели Optimized Random Forest*

Качество работы моделей получилось не очень высоким, вероятно, из-за того, что признаки достаточно слабо коррелируют с целевой переменной, а также из-за того, что пользователи классифицируются по нескольким возрастным группам, но при этом у пользователей относительно близких возрастов, но при этом относящихся к разным группам, профили заполнены похожим образом. Также играет роль и тот фактор, что в предложенном решении не учитывается смысловой аспект текстовых признаков, который может являться крайне важной характеристикой пользователей. Стоит отметить, что подбор гиперпараметров заметно улучшил качество Random Forest, но мало повлиял на CatBoost из-за того, что в этом алгоритме параметры по умолчанию автоматически подбираются под задачу различными эвристиками.

ЗАКЛЮЧЕНИЕ

В данной работе представлено решение задачи предсказания возрастной группы пользователей социальной сети различными методами многоклассовой классификации. Был проведен обзор литературы, собран датасет, данные были исследованы и предобработаны, модели были выбраны, оптимизированы и протестированы. Из-за того, что поставленная задача оказалась сложна для решения с использованием имеющихся данных, качество моделей получилось недостаточно высоким. В дальнейшем планируется продолжать исследования в этой сфере, изучить другие подходы к прогнозированию, в частности используя методы распознавания изображений и естественного языка.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Antonio A. Morgan-López, Annice E. Kim, Rob Chew, Paul Ruddle. [Predicting age groups of Twitter users based on language and metadata features](#)
2. Wenzhi Cao, Vahid Mirjalili, Sebastian Raschka. [Rank consistent ordinal regression for neural networks with application to age estimation](#)
3. Гомзин А.Г., Кузнецов С.Д. [Метод автоматического определения возраста пользователей с помощью социальных связей](#)
4. Margherita Grandini, Enrico Bagli, Giorgio Visani. [Metrics for Multi-Class Classification: an Overview](#)
5. Orelin Geron. Applied Machine Learning with Scikit-Learn and TensorFlow
6. Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, Andrey Gulin. [CatBoost: unbiased boosting with categorical features](#)