Билеты, часть 1:

1. **Типы задач в машинном обучении, метрики**
2. **Типы задач в машинном обучении, функции потерь**
3. **Линейная регрессия**
4. **Логистическая регрессия**
5. **Полносвязная нейронная сеть и ее связь с линейной регрессией**
6. **Полносвязные нейронные сети, полносвязные слои**
7. **Полносвязные нейронные сети, функции активации**
8. **Обучение нейронных сетей, SGD**
9. **Обучение нейронных сетей, SGD + momentum**
10. **Обучение нейронных сетей, Adam**
11. **Обучение нейронных сетей, BatchNorm**
12. **Обучение нейронных сетей, инициализация весов**
13. **Обучение нейронных сетей, аугментации**
14. **Обучение нейронных сетей, гиперпараметры**
15. **Обучение нейронных сетей, регуляризация L1/L2**
16. **Обучение нейронных сетей, регуляризация dropout, drop connect**
17. **Сверточные нейронные сети, свертка, отличие от DWS свертки**
18. **Сверточные нейронные сети, DWS свертка, отличие от обычной свертки**
19. **Сверточные нейронные сети, связь сверточных нейронных сетей с FC**
20. **Сверточные нейронные сети, ResNet**
21. **Сверточные нейронные сети, MobileNet**
22. **Сверточные нейронные сети, Inception (v3)**
23. **Функции активации. Затухание и взрыв градиента**
24. **Метрики в задачах классификации (accuracy, precision, recall)**
25. **Метрика mAP, IOU**
26. **Метрика per pixel accuracy, IOU**

**Билеты, часть 2, “продвинутые” вопросы:**

1. **Обратное распространение ошибки, общая формула**
2. **Обратное распространение ошибки, производная по FC слою**
3. **Обратное распространение ошибки, производная по conv слою**
4. **Задача детекции, архитектура SSD**
5. **Задача детекции, архитектура Yolo-v5**
6. **Задача детекции, архитектура FasterRCNN**
7. **Задача детекции, архитектура RCNN**
8. **Задача детекции, архитектура FastRCNN**
9. **Задача сегментации, deconvolution, ее связь с со сверткой и т.д.**
10. **Задача сегментации, UNet- подобные архитектуры**
11. **Задача Seq2Seq, LSTM архитектура, обучение**
12. **Задача Seq2Seq, LSTM + attention**
13. **Задача Seq2Seq, архитектура Transformer для задачи переводов**
14. **Внимание, базовый блок архитектуры Transformer**
15. **Процедура дистилляции, как влияет на повышение точности**
16. **Процедура квантования, как работает, пример способа квантования**

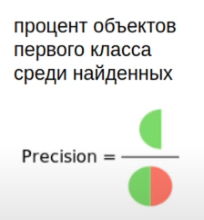
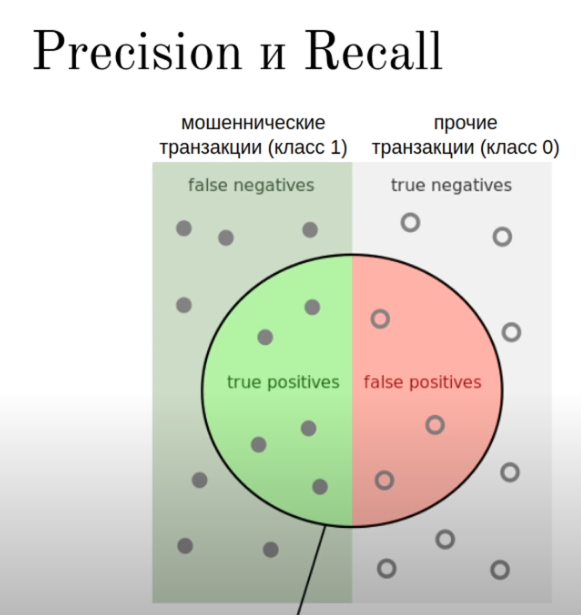
# **Типы задач в машинном обучении, метрики**

Метрика помагает в себе объеденить такие параметры как точность и полноте

Метрика – показатель качества работы алгоритма для данной задачи (1 – принимает на вход правильный ответ и ответ алгоритма, 2 – чем больше метрика чем точнее алгоритм предсказывает правильный ответ)

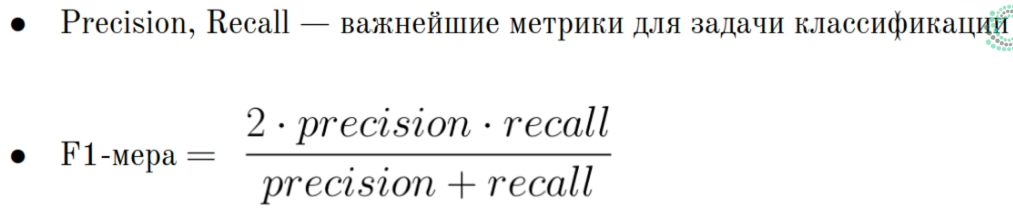
**Метрика качества (метрика)** - функция, которая показывает, насколько сильно полученные предсказания, выдаваемые моделью, похожи на правильные ответы. Выбирается из соображений целесообразности (от бизнеса), а не из-за её совместимости с какими-то моделями (Может быть недифференцируемой).

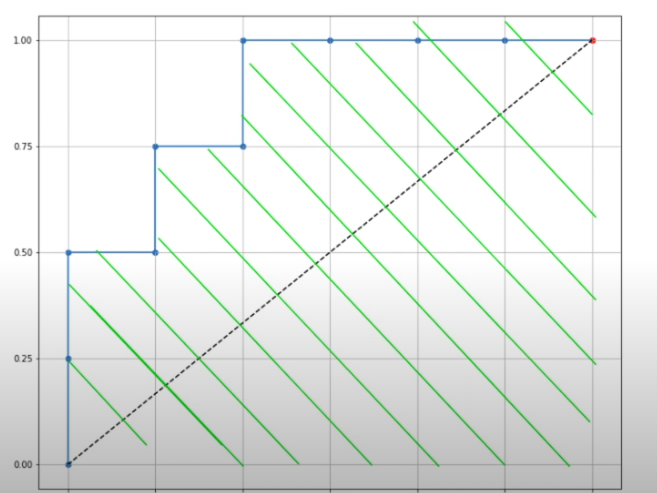
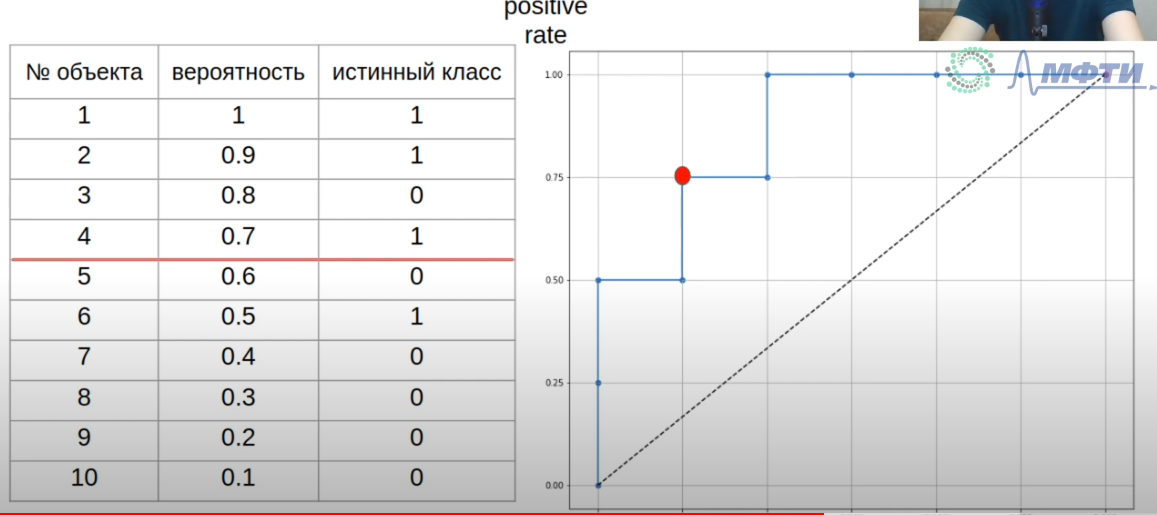
Accuracy метрика не специфична к различным классам задачь (деление на 2 класса)

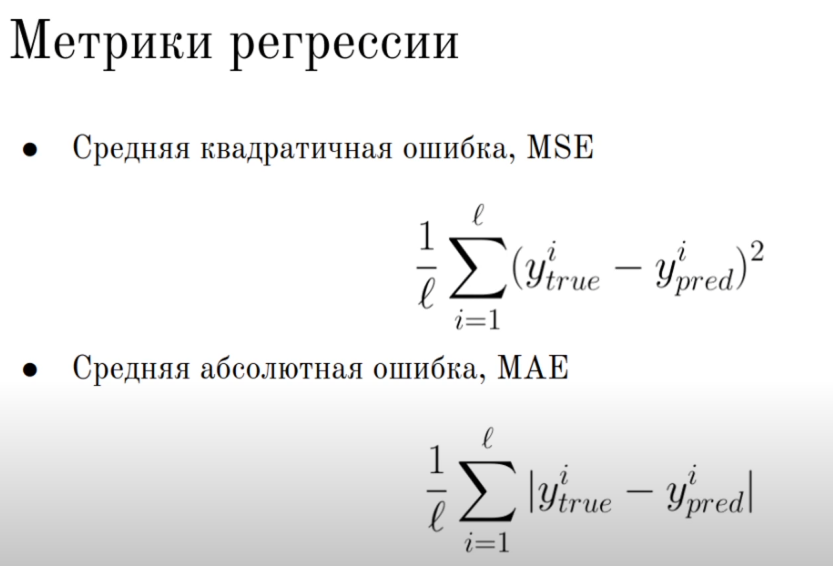
****

**1 метрика Precision = % объекта 1-го класса среди найденых**

**2 метрика Recall = % найденных объектов1-го класса (сколько % плохих операций мы нашли)**

**При поиске наиболее оптимальной оперции используют  
**

**3 метрика ROC-AUC метрика для задачь класивикации, работает с предсказанными вер. Классов. В отсличии от верхних (предсказанными лейблами классов). Данная метрика равна площади рок кривой  
** ****

****

# **Типы задач в машинном обучении, функции потерь**

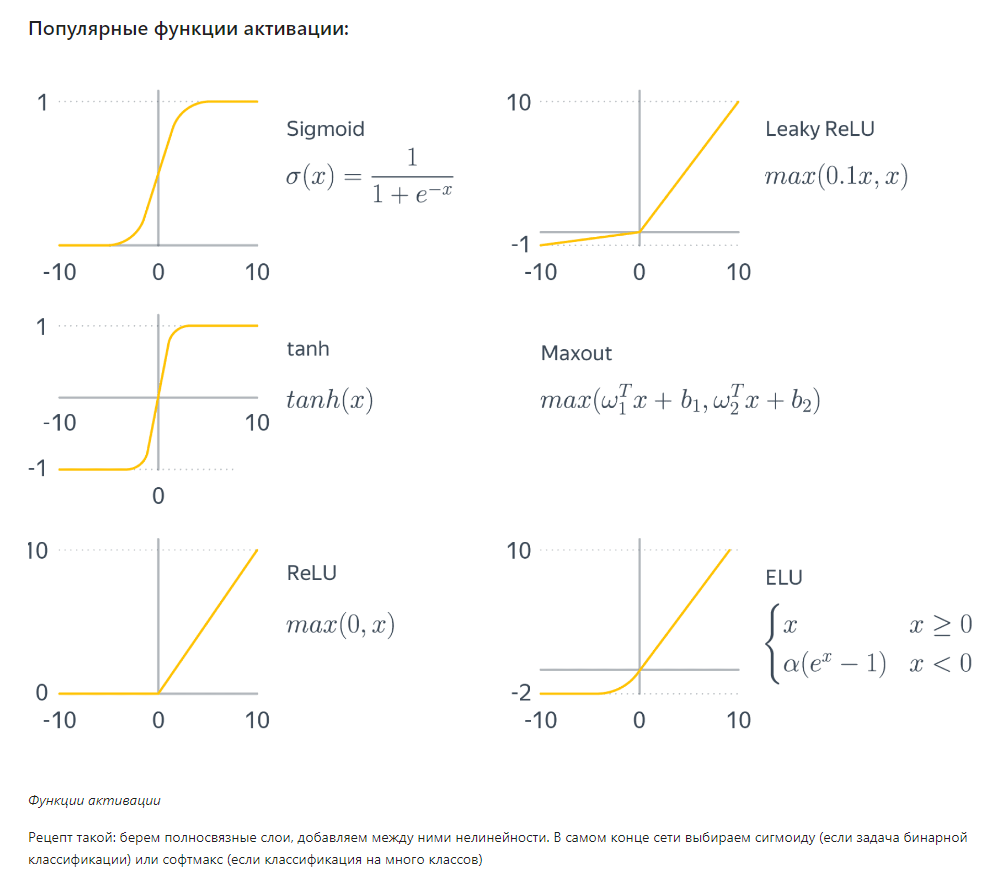
**1)Задача регрессии**– прогноз на основе выборки объектов с различными признаками. На выходе должно получиться вещественное число (2, 35, 76.454 и др.), к примеру цена квартиры, стоимость ценной бумаги по прошествии полугода, ожидаемый доход магазина на следующий месяц, качество вина при слепом тестировании.  
  
**2)Задача классификации** – получение категориального ответа на основе набора признаков. Имеет конечное количество ответов (как правило, в формате «да» или «нет»): есть ли на фотографии кот, является ли изображение человеческим лицом, болен ли пациент раком.  
  
**3)Задача кластеризации** – распределение данных на группы: разделение всех клиентов мобильного оператора по уровню платёжеспособности, отнесение космических объектов к той или иной категории (планета, звёзда, чёрная дыра и т. п.).  
  
**4)Задача уменьшения размерности** – сведение большого числа признаков к меньшему (обычно 2–3) для удобства их последующей визуализации (например, сжатие данных).  
  
**5)Задача выявления аномалий** – отделение аномалий от стандартных случаев. На первый взгляд она совпадает с задачей классификации, но есть одно существенное отличие: аномалии – явление редкое, и обучающих примеров, на которых можно натаскать машинно обучающуюся модель на выявление таких объектов, либо исчезающе мало, либо просто нет, поэтому методы классификации здесь не работают. На практике такой задачей является, например, выявление мошеннических действий с банковскими картами.

**Функция потерь** - прокси-метрика, в процессе оптимизации которой улучшается и исходная метрика. Используется при поиске оптимальной модели при помощи градиентного спуска. (Нужна дифференцируемая)

**Обобщающая способность модели** - способность модели выучить закономерности не только на обучающем датасете, но и давать адекватные результаты на новых данных.

**Переобученный алгоритм** - алгоритм, избыточно подстроившийся под данные. С увеличением сложности модели ошибка на обучающей выборке падает, при этом ошибка на тестовых данных может расти.

* На train только тренируемся
* Мы градиентным спуском "подкручиваем" веса так, чтобы метрика на train была высокой. Поэтому, если что-то не пошло сильно не так, она всегда будет на train достаточно хорошей, и это об обобщающей способности модели не будет ничего говорить
* Нужно валидироваться (мерить метрику) на другой подвыборке (на val)
* Уже по валидации можно судить, натренировалась и не перетренировалась ли модель
* В самом конце, когда уже прекратили обучение, по-хорошему надо еще раз померить метрику на test. А то вдруг мы переобучились под val



Функции активации

Рецепт такой: берем полносвязные слои, добавляем между ними нелинейности. В самом конце сети выбираем сигмоиду (если задача бинарной классификации) или софтмакс (если классификация на много классов)

# **Линейная регрессия**

# **Логистическая регрессия**

# **Полносвязная нейронная сеть и ее связь с линейной регрессией**

# **Полносвязные нейронные сети, полносвязные слои**

**Полносвязные нейронные сети, функции активации**

**Обучение нейронных сетей, SGD**

**Обучение нейронных сетей, SGD + momentum**

**Обучение нейронных сетей, Adam**

**Обучение нейронных сетей, BatchNorm**

**Обучение нейронных сетей, инициализация весов**

**Обучение нейронных сетей, аугментации**

**Обучение нейронных сетей, гиперпараметры**

**Обучение нейронных сетей, регуляризация L1/L2**

**Обучение нейронных сетей, регуляризация dropout, drop connect**

**Сверточные нейронные сети, свертка, отличие от DWS свертки**

**Сверточные нейронные сети, DWS свертка, отличие от обычной свертки**

**Сверточные нейронные сети, связь сверточных нейронных сетей с FC**

**Сверточные нейронные сети, ResNet**

**Сверточные нейронные сети, MobileNet**

**Сверточные нейронные сети, Inception (v3)**

[https://github.com/pytorch/vision/blob/main/torchvision/models/inception.py](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fgithub.com%2Fpytorch%2Fvision%2Fblob%2Fmain%2Ftorchvision%2Fmodels%2Finception.py)

**Функции активации. Затухание и взрыв градиента**

**Метрики в задачах классификации (accuracy, precision, recall)**

**Метрика mAP, IOU**

**Метрика per pixel accuracy, IOU**

**Билеты, часть 2, “продвинутые” вопросы:**

**Обратное распространение ошибки, общая формула**

**Обратное распространение ошибки, производная по FC слою**

**Обратное распространение ошибки, производная по conv слою**

**Задача детекции, архитектура SSD**

**Задача детекции, архитектура Yolo-v5**

**Задача детекции, архитектура FasterRCNN**

**Задача детекции, архитектура RCNN**

**Задача детекции, архитектура FastRCNN**

**Задача сегментации, deconvolution, ее связь с со сверткой и т.д.**

**Задача сегментации, UNet- подобные архитектуры**

**Задача Seq2Seq, LSTM архитектура, обучение**

**Задача Seq2Seq, LSTM + attention**

**Задача Seq2Seq, архитектура Transformer для задачи переводов**

**Внимание, базовый блок архитектуры Transformer**

**Процедура дистилляции, как влияет на повышение точности**

**Процедура квантования, как работает, пример способа квантования**