ПОДГОТОВКА МАШИННОГО ОБУЧЕНИЯ

ВЫБОР МОДЕЛЕЙ

Выбор модели опирируется на множество факторов. Рассмотрим все ключевые факторы выбора модели машинногообучения и относительно них выберим модели.

Представленная задача прогнозирования успеваемости студента относится к виду задач “обучение с учителем”. Целевой прогнозирующей переменной будут данные об среднем балле по сто бальной шкале студент за последний семестр.

Обучение с учителем (Supervised learning) – это вид обучения модели, при котором обучающая выборка имеет метки.

Тип задачи ВКР, прогнозирование успеваемости студента, относится к задачи регрессии.

Y = R (Y = RM) Регрессия (Regression) – это задача предсказания вещественного числа.

Что касается выбора модели относительно данных, то данных немного в связи с этим использование нейронных сетей и тяжёлых моделей не допустимо, нужно выбрать простые модели, которые имеют хорошою способность репрезинтативности.

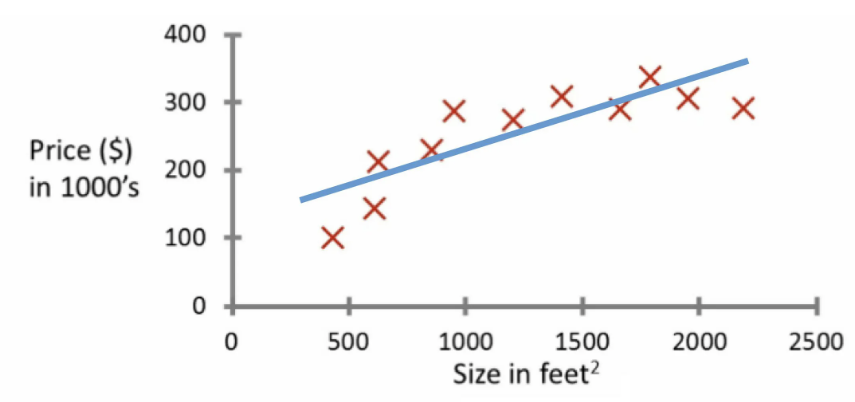
По поводу интерпретации, модель должна обладать хорошей интерпретацией для того, чтобы было возможно объяснить работу алгоритмов.

Также модель должна быть устойчива к переобучению, чтобы в дальнейшем модель качественно прогнозировала относительно новых данных.

Относительно всех вышеозвученных требований нам подходят следующие модели:

ЛИНЕНЙНАЯ МОДЕЛЬ

Линейная модель (Linear model) – это модель, которая является линейной по численным признакам.



Линейная функция имеет вид:

 или 

Форма записи облегчающая работу с матрицой признаков:



где:

y – **целевая переменная (Target variable)**;

(x1, x2, x3, …, xD) – объект выборки, **вектор признаков (Vector of features)**;

*Например*: объект Игрок № 6.

w1, w2, w3, …, wD,w0 – **вектор весов (Vector of weights)**.

**Вес (Weight)** – это числовое значение, показывающие важность признака.

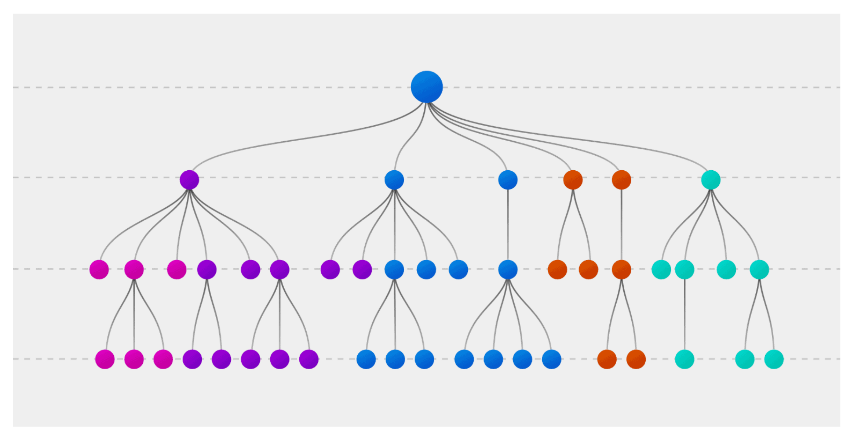
Линейные зависимости могут использовать сложные компоненты: логарифмы, экспоненту, модуль, ...

**w0 – свободный коэффициент, сдвиг (Bias)** – это значение, на которое сдвигается график по оси ox.

**СЛУЧАЙНЫЙ ЛЕС**

**RF (random forest)** — это множество решающих деревьев. В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по следующей схеме:

* Выбирается подвыборка обучающей выборки размера samplesize (м.б. с возвращением) – по ней строится дерево (для каждого дерева — своя подвыборка).
* Для построения каждого расщепления в дереве просматриваем max\_features случайных признаков (для каждого нового расщепления — свои случайные признаки).
* Выбираем наилучшие признак и расщепление по нему (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление.



Понятно, что такая схема построения соответствует главному принципу ансамблировнаия (построению алгоритма машинного обучения на базе нескольких, в данном случае решающих деревьев): базовые алгоритмы должны быть хорошими и разнообразными (поэтому каждое дерево строится на своей обучающей выборке и при выборе расщеплений есть элемент случайности).

**ГРАДИЕНТНЫЙ БУСТИНГ НАД РЕШАЮЩИМИ ДЕРЕВЬЯМИ**

Бустинг, использующий деревья решений в качестве базовых алгоритмов, называется градиентным бустингом над решающими деревьями, Gradient Boosting on Decision Trees, GBDT. Он отлично работает на выборках с «табличными», неоднородными данными. Примером таких данных может служить описание пользователя Яндекса через его возраст, пол, среднее число поисковых запросов в день, число заказов такси и так далее. Такой бустинг способен эффективно находить нелинейные зависимости в данных различной природы. Этим свойством обладают все алгоритмы, использующие деревья решений, однако именно GBDT обычно выигрывает в подавляющем большинстве задач. Благодаря этому он широко применяется во многих конкурсах по машинному обучению и задачах из индустрии (поисковом ранжировании, рекомендательных системах, таргетировании рекламы, предсказании погоды, пункта назначения такси и многих других).

Алгоритм градиентного бустинга

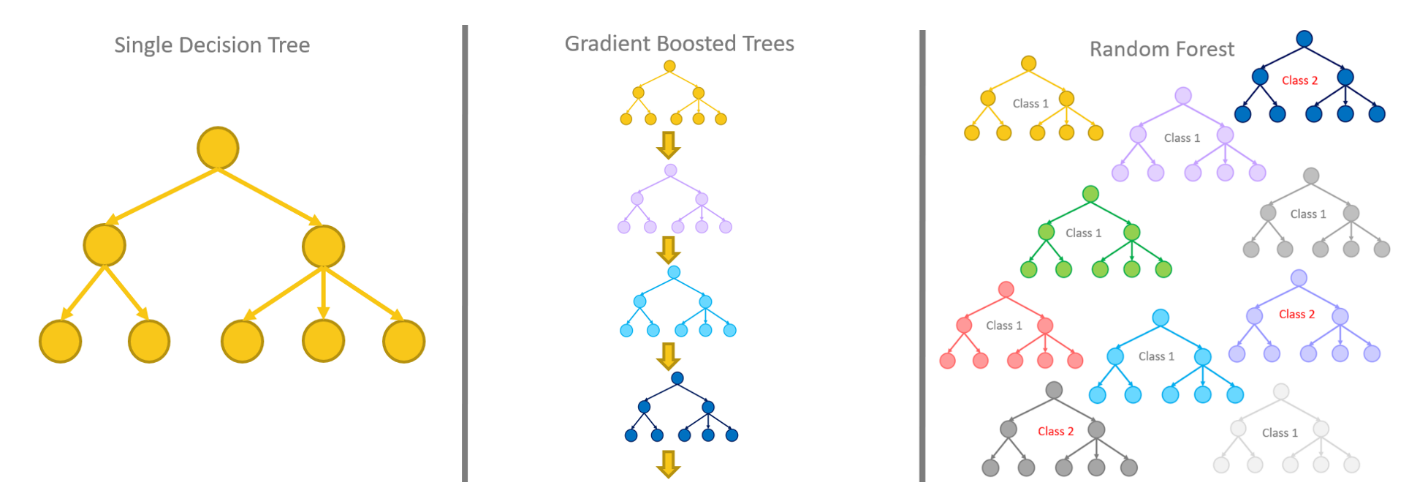
**Градиентный бустинг** — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений.

Цель любого алгоритма [обучения с учителем](https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/) — определить функцию потерь и минимизировать её. Давайте обратимся к математике градиентного бустинга. Пусть, например, в качестве функции потерь будет среднеквадратичная ошибка (MSE):

Мы хотим, чтобы построить наши предсказания таким образом, чтобы MSE была минимальна. Используя [градиентный спуск](https://neurohive.io/ru/osnovy-data-science/gradientyj-busting/%D0%B3%D1%80%D0%B0%D0%B4%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B9%20%D0%B1%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B3https:/neurohive.io/ru/osnovy-data-science/gradient-descent/) и обновляя предсказания, основанные на скорости обучения (learning rate), ищем значения, на которых MSE минимальна.

Итак, мы просто обновляем предсказания таким образом, что сумма наших отклонений стремилась к нулю и предсказанные значения были близки к реальным.

Визуализация всех моделей.



ВЫБОР метрики

В задачах регрессии целевая метка у нас имеет потенциально бесконечное число значений. И природа этих значений, обычно, связана с каким-то процессом измерений:

* величина температуры в определенный момент времени на метеостанции
* количество прочтений статьи на сайте
* количество проданных бананов в конкретном магазине, сети магазинов или стране
* дебит добывающей скважины на нефтегазовом месторождении за месяц и т.п.

Мы видим, что иногда метка это целое число, а иногда произвольное вещественное число. Обычно случаи целочисленных меток моделируют так, словно это просто обычное вещественное число. При таком подходе может оказаться так, что модель A лучше модели B по некоторой метрике, но при этом предсказания у модели A могут быть не целыми. Если в бизнес-задаче ожидается именно целочисленный ответ, то и оценивать нужно огрубление.

Общая рекомендация такова: оценивайте весь каскад решающих правил: и те «внутренние», которые вы получаете в результате обучения, и те «итоговые», которые вы отдаёте бизнес-заказчику.

Например, вы можете быть удовлетворены, что стали ошибаться не во втором, а только в третьем знаке после запятой при предсказании погоды. Но сами погодные данные измеряются с точностью до десятых долей градуса, а пользователь и вовсе может интересоваться лишь целым числом градусов.

Итак, напомним постановку задачи регрессии: нам нужно по обучающей выборке (��,��)�=1�, где ��∈� построить модель f(x).

Величину ��=�(��)−�� называют ошибкой на объекте i или регрессионным остатком.

Весь набор ошибок на отложенной выборке может служить аналогом матрицы ошибок из задачи классификации. А именно, когда мы рассматриваем две разные модели, то, глядя на то, как и на каких объектах они ошиблись, мы можем прийти к выводу, что для решения бизнес-задачи нам выгоднее взять ту или иную модель. И, аналогично со случаем бинарной классификации, мы можем начать строить агрегаты от вектора ошибок, получая тем самым разные метрики.

### MAE

Использовать RMSE для сравнения моделей на выборках с большим количеством выбросов может быть неудобно. В таких случаях прибегают к также знакомой вам в качестве функции потери метрике **MAE** (**mean absolute error**):

