5. Кривые валидации и обучения

Мы уже получили представление о проверке модели, кросс-валидации и регуляризации.  
Теперь рассмотрим главный вопрос:

Если качество модели нас не устраивает, что делать?

* Сделать модель сложнее или упростить?
* Добавить больше признаков?
* Или нам просто нужно больше данных для обучения?

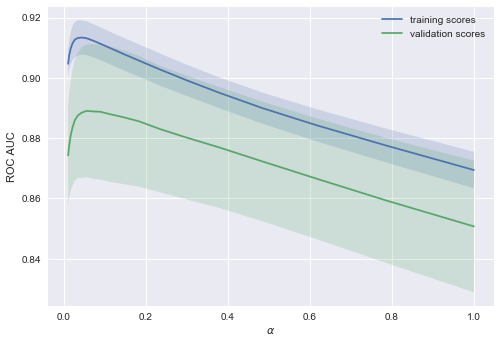
Ответы на данные вопросы не всегда лежат на поверхности. В частности, иногда использование более сложной модели приведет к ухудшению показателей. Либо добавление наблюдений не приведет к ощутимым изменениям. Способность принять правильное решение и выбрать правильный способ улучшения модели, собственно говоря, и отличает хорошего специалиста от плохого.

Будем работать со знакомыми данными по оттоку клиентов телеком-оператора.

Импорт библиотек и чтение данных

Логистическую регрессию будем обучать стохастическим градиентным спуском. Пока объясним это тем, что так быстрее, но далее в программе у нас отдельная статья про это дело. Построим валидационные кривые, показывающие, как качество (ROC AUC) на обучающей и проверочной выборке меняется с изменением параметра регуляризации.

Код



Тенденция видна сразу, и она очень часто встречается.

* Для простых моделей тренировочная и валидационная ошибка находятся где-то рядом, и они велики. Это говорит о том, что модель **недообучилась**: то есть она не имеет достаточное кол-во параметров.
* Для сильно усложненных моделей тренировочная и валидационная ошибки значительно отличаются. Это можно объяснить **переобучением**: когда параметров слишком много либо не хватает регуляризации, алгоритм может "отвлекаться" на шум в данных и упускать основной тренд.

**Сколько нужно данных?**

Известно, что чем больше данных использует модель, тем лучше. Но как нам понять в конкретной ситуации, помогут ли новые данные? Скажем, целесообразно ли нам потратить N$ на труд асессоров, чтобы увеличить выборку вдвое?

Поскольку новых данных пока может и не быть, разумно поварьировать размер имеющейся обучающей выборки и посмотреть, как качество решения задачи зависит от объема данных, на котором мы обучали модель. Так получаются **кривые обучения** (**learning curves**).

Идея простая: мы отображаем ошибку как функцию от количества примеров, используемых для обучения. При этом параметры модели фиксируются заранее.

Давайте посмотрим, что мы получим для линейной модели. Коэффициент регуляризации выставим большим.

Код



Типичная ситуация: для небольшого объема данных ошибки на обучающей выборке и в процессе кросс-валидации довольно сильно отличаются, что указывает на переобучение. Для той же модели, но с большим объемом данных ошибки "сходятся", что указывается на недообучение.

Если добавить еще данные, ошибка на обучающей выборке не будет расти, но с другой стороны, ошибка на тестовых данных не будет уменьшаться.

Получается, ошибки "сошлись", и добавление новых данных не поможет. Собственно, это случай – самый интересный для бизнеса. Возможна ситуация, когда мы увеличиваем выборку в 10 раз. Но если не менять сложность модели, это может и не помочь. То есть стратегия "настроил один раз – дальше использую 10 раз" может и не работать.

Что будет, если изменить коэффициент регуляризации (уменьшить до 0.05)?

Видим хорошую тенденцию – кривые постепенно сходятся, и если дальше двигаться направо (добавлять в модель данные), можно еще повысить качество на валидации.



А если усложнить модель ещё больше (alpha=10−4)?

Проявляется переобучение – AUC падает как на обучении, так и на валидации.



Строя подобные кривые, можно понять, в какую сторону двигаться, и как правильно настроить сложность модели на новых данных.

**Выводы по кривым валидации и обучения**

* Ошибка на обучающей выборке сама по себе ничего не говорит о качестве модели
* Кросс-валидационная ошибка показывает, насколько хорошо модель подстраивается под данные (имеющийся тренд в данных), сохраняя при этом способность обобщения на новые данные
* **Валидационная кривая** представляет собой график, показывающий результат на тренировочной и валидационной выборке в зависимости от **сложности модели**:
* если две кривые распологаются близко, и обе ошибки велики, — это признак *недообучения*
* если две кривые далеко друг от друга, — это показатель *переобучения*
* **Кривая обучения** — это график, показывающий результаты на валидации и тренировочной подвыборке в зависимости от количества наблюдений:
* если кривые сошлись друг к другу, добавление новых данных не поможет – надо менять сложность модели
* если кривые еще не сошлись, добавление новых данных может улучшить результат.