Рассмотрим применение описанных выше методов в реальной задаче – предсказать, зарабатывает ли человек больше $50 тыс. Загрузим библиотеки и данные, для удобства оставив только численные признаки:

* age – возраст
* fnlwgt (final weight) – примерная оценка количества людей, которое представляет каждая строка данных
* educational-num – длительность обучения
* capital-gain – прирост капитала
* capital-loss – потеря капитала
* hours-per-week – количество рабочих часов в неделю

Посмотрим точность на кросс-валидации и важность признаков для случайного леса:

scores = [0.82427915 0.82290796 0.83106668 0.8192637 0.83155106]

mean score = 0.82581 +/- 0.00478

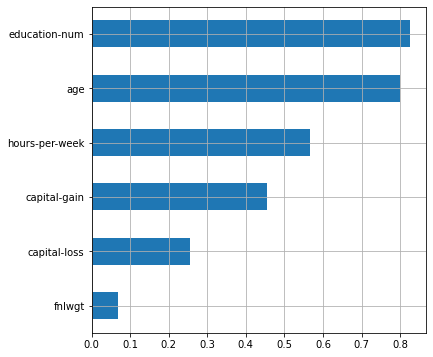


Самым важным признаком для случайного леса является fnlwgt. Это можно интерпретировать как то, что главным фактором того, что человек зарабатывает больше $50 тыс. является количество людей с такими же характеристиками. Такая интерпретация выглядит нелогичной, и происходит это потому, что модели с деревьями могут выдавать сильно смещённую оценку признаков. Притом, чем хуже настроена модель, тем сильнее может быть смещение, поэтому доверять оценкам таких моделей надо с осторожностью.

Повторим процедуру для линейной модели (с L1-регуляризацией). Для нормализации данных будем использовать метод PowerTransformer.

scores = [0.82034993 0.83000963 0.8348707 0.81787667 0.83548066]

mean score = 0.82772 +/- 0.00732



Создадим 12 шумовых признаков, элементами которых будут некоррелируемые случайные числа из выборок с нормальным, равномерным и Лапласовым распределениями. Параметры каждого распределения подбираются случайным образом независимо друг от друга.



Проведём кросс-валидацию на зашумлённых данных и посмотрим важность признаков:

scores = [0.8522425 0.85382173 0.86249657 0.84897581 0.85443027]

mean score = 0.85439 +/- 0.00447

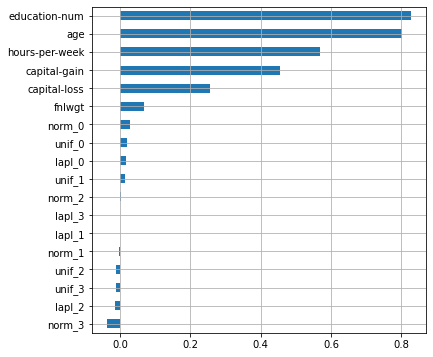


Несмотря на большое количество добавленных шумовых признаков, точность модели на кросс-валидации значительно возросла как на каждом фолде, так и в среднем! Кроме этого, все шумовые признаки имеют высокую важность, сравнимую с двумя оригинальными. Очевидно, что наша модель переобучена, однако в реальных задачах такие ситуации бывает очень сложно распознать, особенно когда при удалении некоторых признаков (про которые неизвестно – шумовые они, или нет) падает валидационная точность. Кроме того, часто бывает сложно подобрать пороговое значение важности признаков для исключения их из модели.

Посмотрим результаты для регрессии.

scores = [0.81993058 0.83005516 0.83446553 0.81763029 0.83543145]

mean score = 0.82750 +/- 0.00738



После добавления шумовых признаков модель не преобучилась, к тому же эти признки имеют значительно меньшие коэффициенты, чем оригинальные. Отметим, что распределение коэффициентов в линейных моделях часто зависит от способа нормализации или масштабирования признаков.

Проведём отбор признаков статистическими методами, для чего будем использовать обобщённый вариант SelectKBest и SelectPercentile, который называется [GenericUnivariateSelect](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.GenericUnivariateSelect.html" \l "sklearn.feature_selection.GenericUnivariateSelect). Он принимает на вход 3 параметра – функцию оценки, режим отбора и его характеристики. В качестве функции оценки будем использовать взаимную информацию.

|  |  |  |
| --- | --- | --- |
|  | **score** | **support** |
| **capital-gain** | 0.080221 | True |
| **age** | 0.065703 | True |
| **education-num** | 0.064743 | True |
| **hours-per-week** | 0.043655 | True |
| **capital-loss** | 0.033617 | True |
| **fnlwgt** | 0.033390 | True |
| **norm\_3** | 0.003217 | False |
| **unif\_3** | 0.002696 | False |
| **norm\_0** | 0.002506 | False |
| **norm\_2** | 0.002052 | False |
| **lapl\_3** | 0.001201 | False |
| **unif\_1** | 0.001144 | False |
| **lapl\_1** | 0.000000 | False |
| **unif\_2** | 0.000000 | False |
| **lapl\_2** | 0.000000 | False |
| **lapl\_0** | 0.000000 | False |
| **unif\_0** | 0.000000 | False |
| **norm\_1** | 0.000000 | False |

Сгенерированные нами признаки имеют низкое значение оценочной функции (scores\_), поэтому в дальнейшем селектор не будет их использовать (get\_support()=False).

В реальной задаче (когда количество шумовых признаков неизвестно) параметры GenericUnivariateSelect можно находить на кросс-валидации вместе с другими гиперпараметрами модели. Посмотрим, как изменится точность классификаторов после подбора их гиперпараметров, а также количества признаков селектора:

scores = [0.8632776968200635, 0.8683443340928604, 0.8710308000627435, 0.8615748939138762, 0.8693334091828478],

mean score = 0.86671 +/- 0.00364

best params = {'rf\_\_max\_depth': 12, 'rf\_\_max\_features': 0.3, 'selector\_\_param': 5}

Для случайного леса средняя точность на кросс-валидации значительно выросла, а лучший результат получился всего для 5 признаков:



Этот результат был получен после удаления шумовых признаков и признака fnlwgt, который при первоначальной оценке был самым значимым для модели. Однако из всех оригинальных признаков он имел наименьшее значение оценочной функции в GenericUnivariateSelect. Результаты оценки важности признаков после их отбора и настройки модели имеют более логичную интерпретацию – на заработок человека влияют именно характеристики человека, а не параметры самой выборки. Таким образом, статистический отбор признаков бывает полезен для увеличения точности некоторых типов моделей и получения менее смещённой оценки при интерпретации их результатов.

Посмотрим, как изменятся коэффиценты у признаков после подбора коэффициента регуляризации у логистической регрессии.

scores = [0.820445329307105, 0.829874053687009, 0.8346493482101578, 0.8177211039148669, 0.8354590546776963],

mean score = 0.82763 +/- 0.00729

best params = {'lr\_\_C': 0.01}

Средняя точность на кросс-валидации почти не изменилась, но скорректировались коэффициенты у шумовых признаков. Отметим, что сильная регуляризация (L1) может занулить излишнее количество признаков.

Отобранные линейной моделью признаки можно подать на вход другой модели. Для этого воспользуемся методом [SelectFromModel](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html" \l "sklearn.feature_selection.SelectFromModel), дадим ему на вход нашу настроенную линейную модель и обозначим граничное значение для отбора коэффициентов.