

# Отчет.

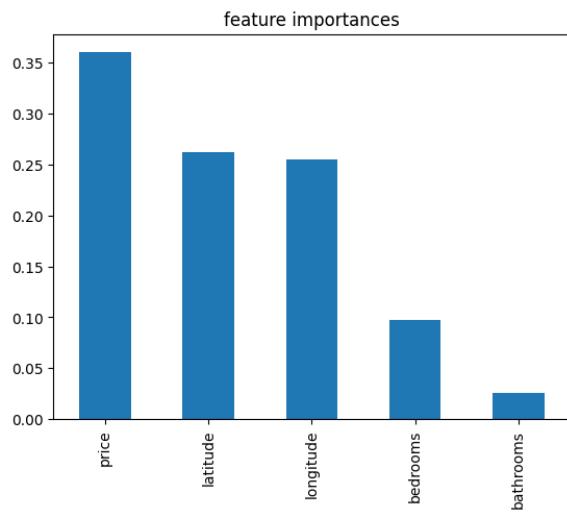
В ходе работы были проведены эксперименты с feature importance, все функции добавлены в библиотеку на `importance_lib`. Было проведено сравнение feature importance в реализации `sklearn` и `R`, `sklearn` с `R-ranger` и сравнение между реализациями: `Sklearn`, `rfpimp`, `treeinterpreter`, `R RandomForest`, `R-ranger`, `RandomForestSRC`, `Party cforest`, `Partykit`.

## Эксперименты.

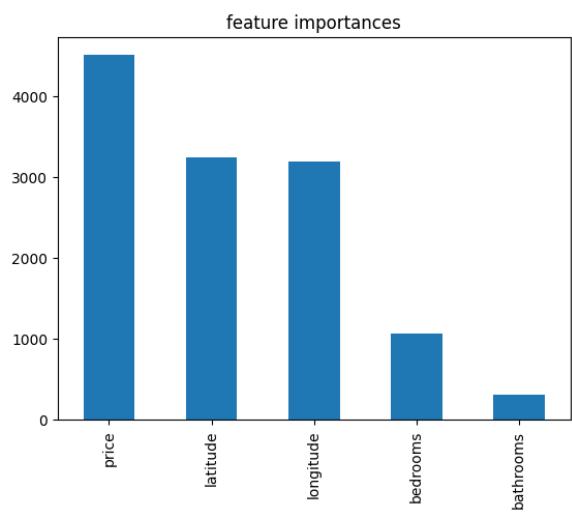
Использовался датасет `rent.csv`, рассмотрены случаи, когда признаки: 1) разнородны, 2) присутствуют коррелированные признаки, 3) присутствуют нерелевантные признаки.

## Sklearn vs R-randomforest

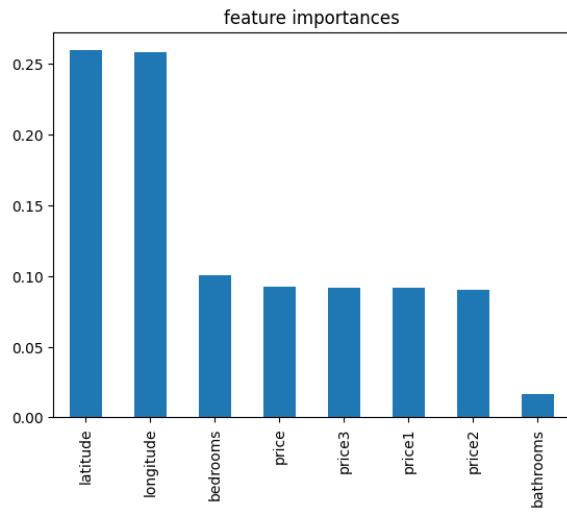
Подбор оптимальных гиперпараметров в реализации `sklearn` осуществлялся с помощью `Optuna`, и полученные параметры использовались в реализации на `R`.



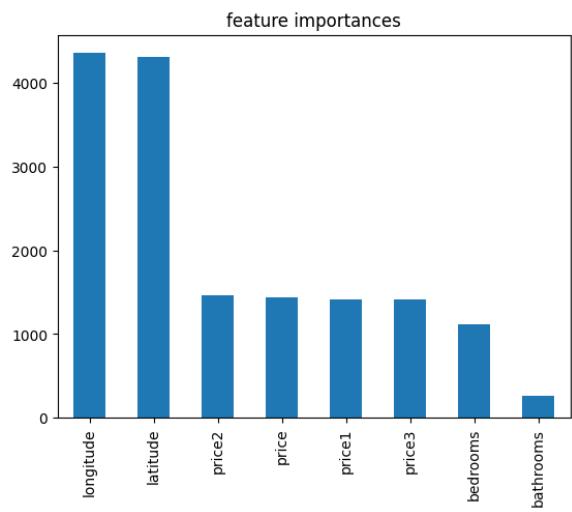
(a) Python: Исходный датасет



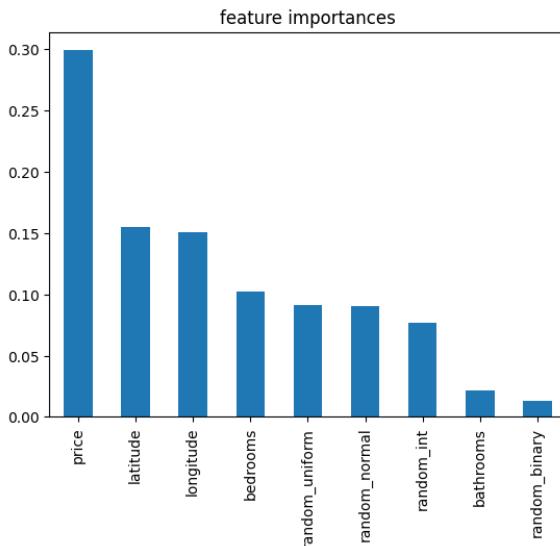
(a) R: Исходный датасет



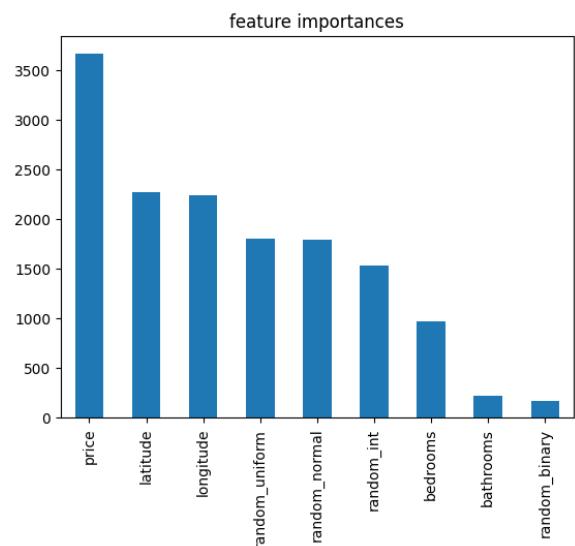
(b) Python: добавлены коррелированные признаки



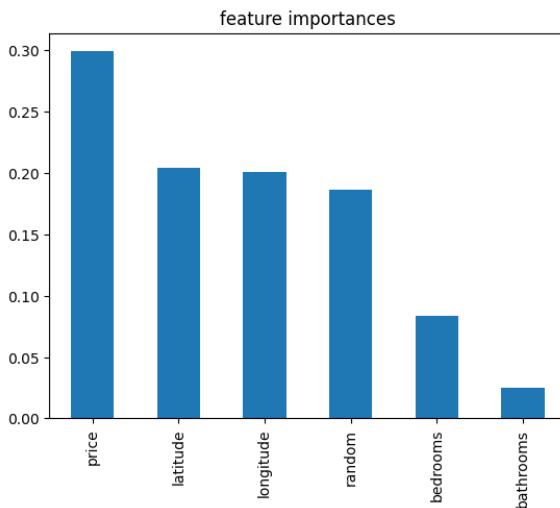
(b) R: добавлены коррелированные признаки



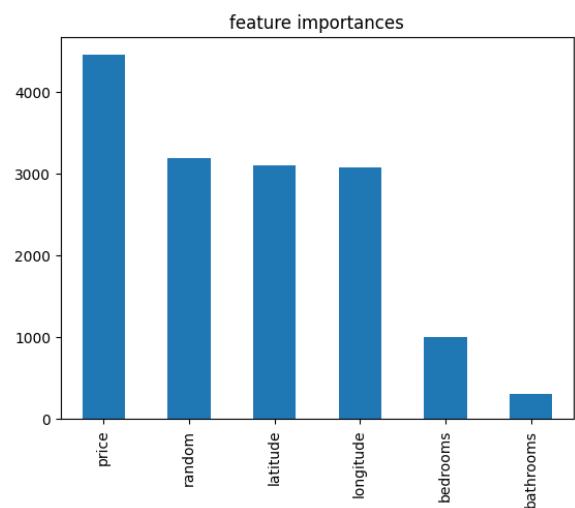
(c) Python: много нерелевантных признаков



(c) R: много нерелевантных признаков



(d) Python: 1 нерелевантный признак



(d) R: 1 нерелевантный признак

Как можно заметить, результаты в некоторых экспериментах неожиданные.

В примере (b) скопирован признак *price* 3 раза, и получилось, что важность этого признака уменьшилась. Вообще, при добавлении коррелированных признаков, важности этих признаков будут уменьшаться.

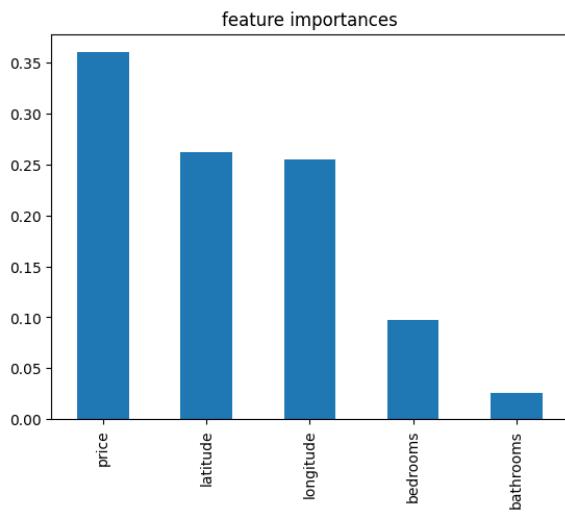
В примерах (c) и (d) были созданы несколько признаков, которые никак не связаны с target.

Возникают вопросы, глядя на рисунки с нерелевантными признаками разного типа: 1) Почему эти признаки имеют важность? 2) Почему признаки *random\_uniform*, *random\_normal*, *random\_int* имеют важность больше, чем признак *random\_binary*?

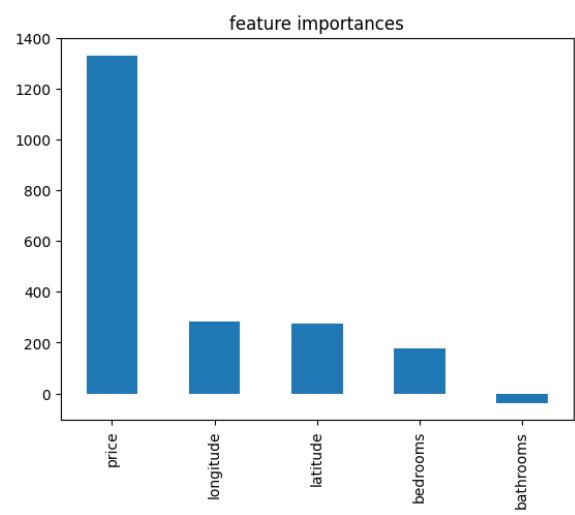
Ниже приведена таблица с оптимальными параметрами для каждого из примеров (a)-(d):

Пример	<i>n_estimators</i>	<i>max_depth</i>	<i>min_samples_leaf</i>	<i>max_features</i>
(a)	1760	26	5	<i>sqrt</i>
(b)	1289	14	3	1.0
(c)	1064	24	13	0.5
(d)	1096	19	3	0.3333333333333333

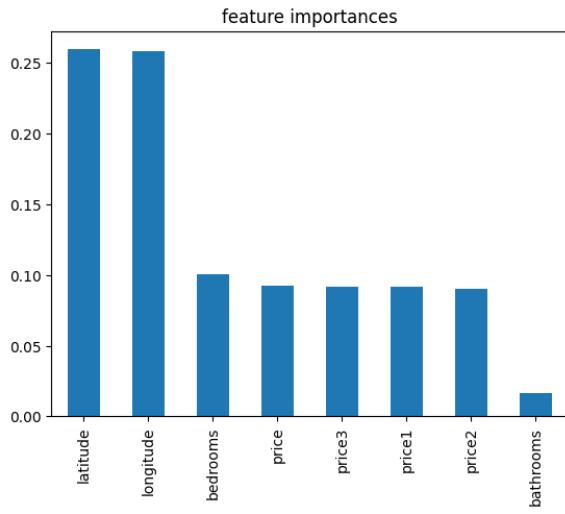
# Sklearn vs R-ranger



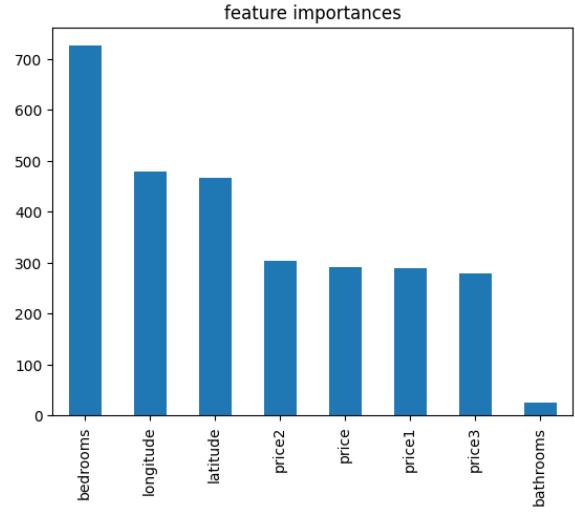
(a) Sklearn: Исходный датасет



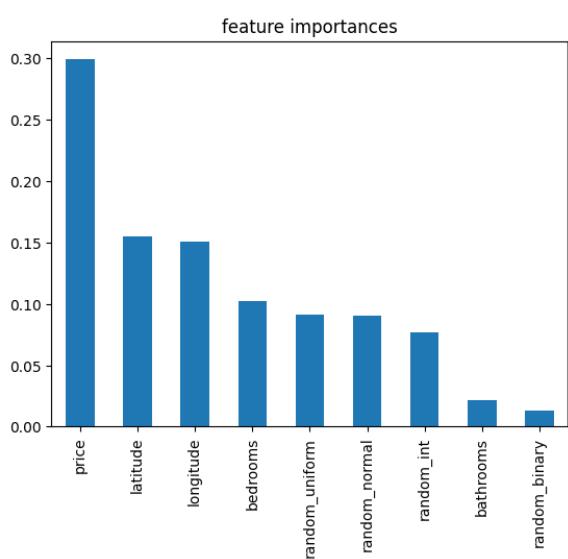
(a) R-ranger: Исходный датасет



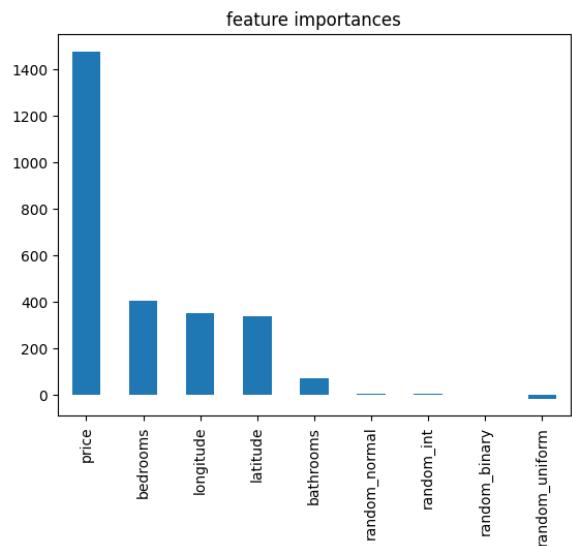
(b) Sklearn: добавлены коррелированные признаки



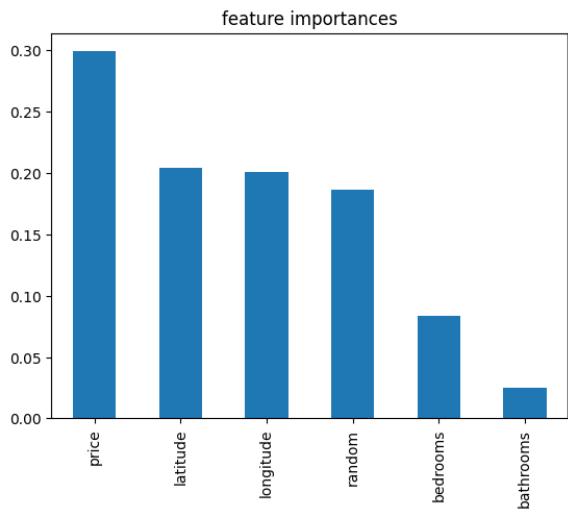
(b) R-ranger: добавлены коррелированные признаки



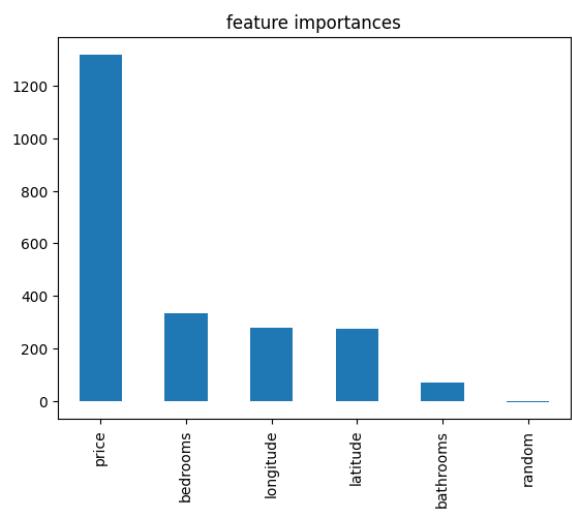
(c) Sklearn: много нерелевантных признаков



(c) R-ranger: много нерелевантных признаков

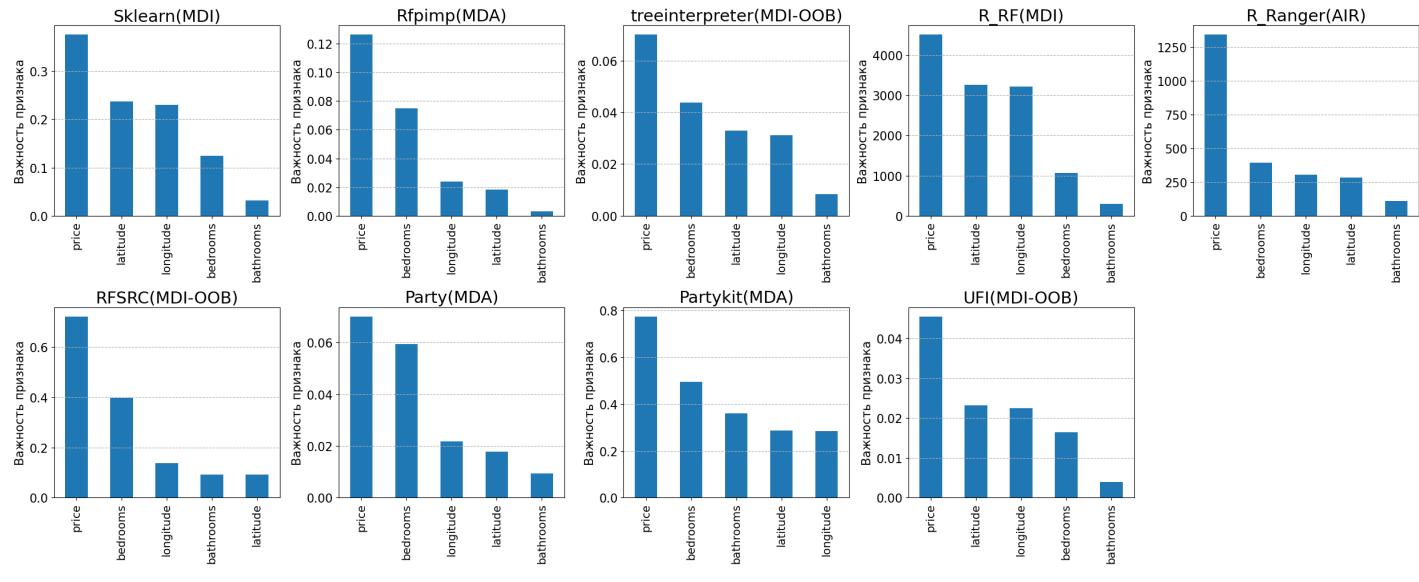


(d) Sklearn: 1 нерелевантный признак

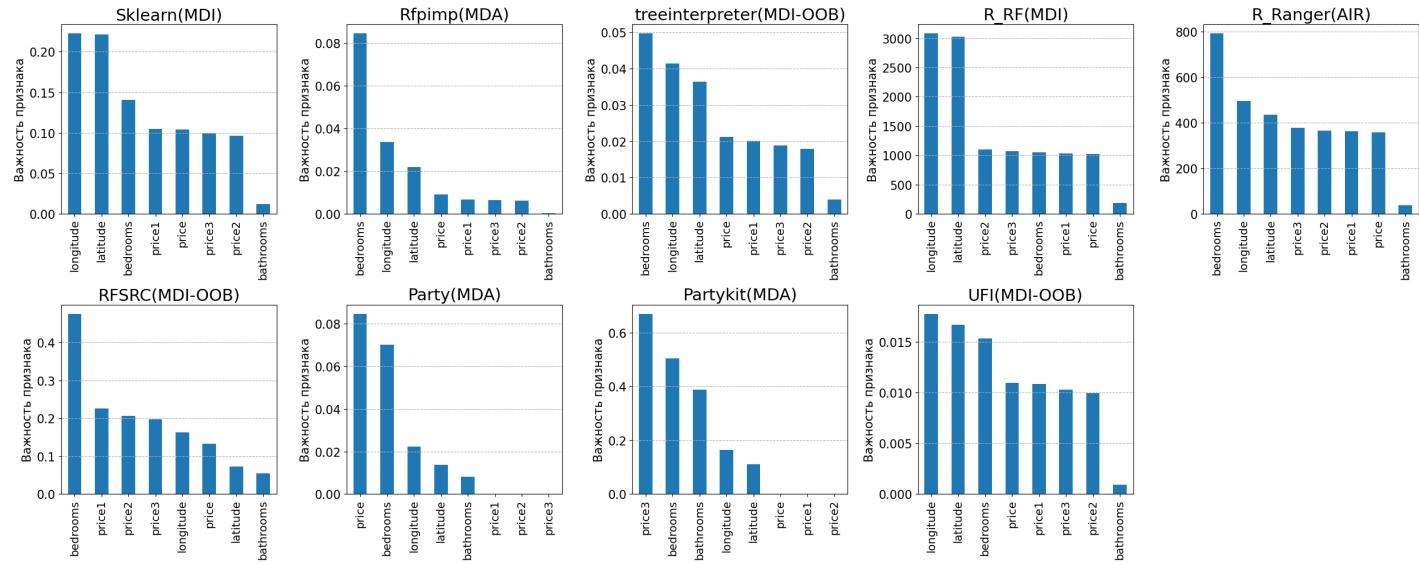


(d) R-ranger: 1 нерелевантный признак

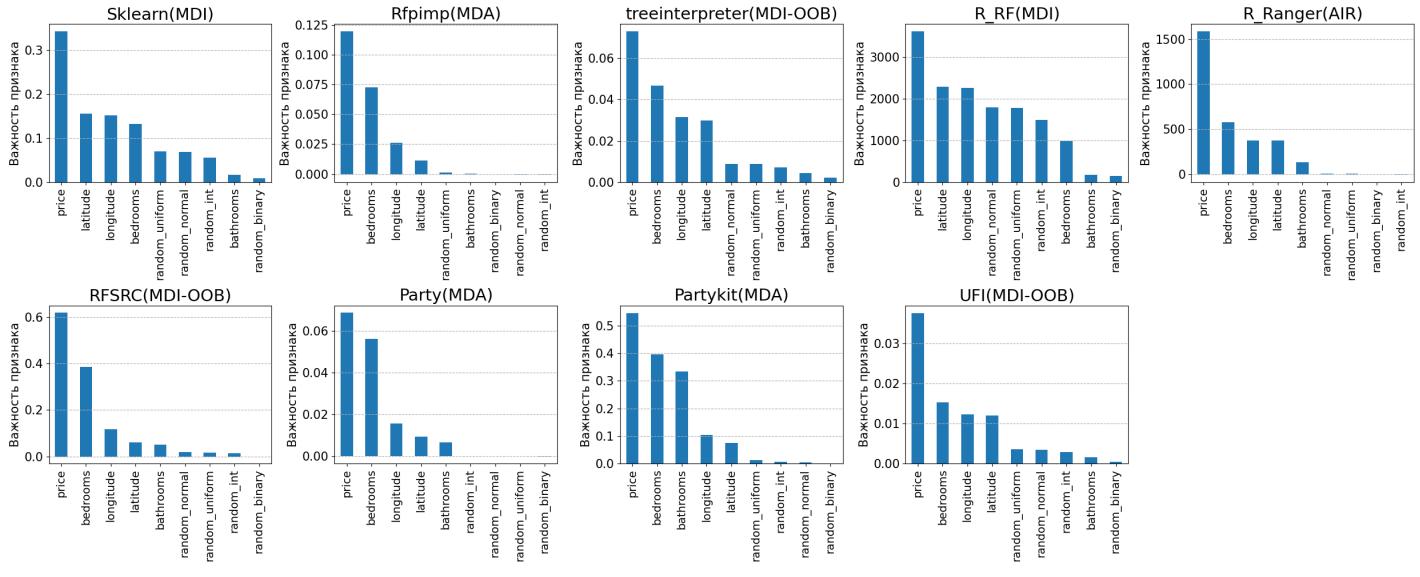
# Сравнение библиотек: Sklearn, rfpimp, treeinterpreter, R RandomForest, R-ranger, RandomForestSRC, Party cforest, Partykit



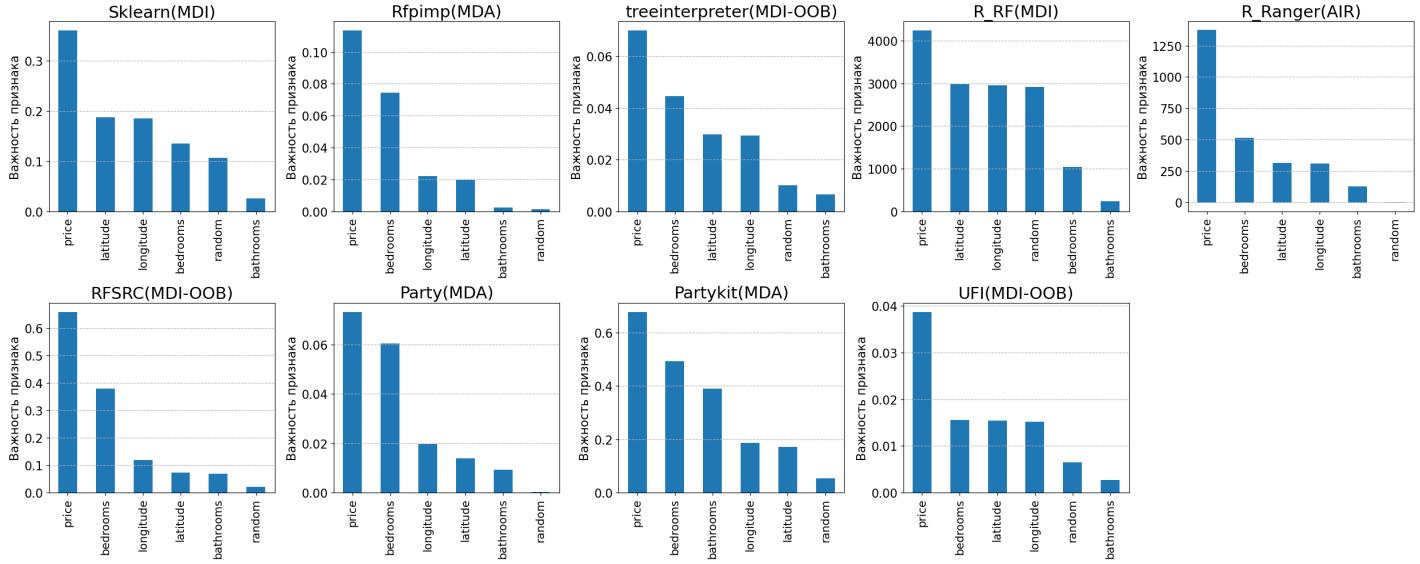
(a) Исходный датасет



(b) Добавлены коррелированные признаки



(c) Много нерелевантных признаков



(d) 1 нерелевантный признак

Из эксперимента (b) видно, что важность одного из 4 продублированных признаков в пакетах party и partykit осталась примерно такой же, как в первом эксперименте, а у оставшихся трех важность равна нулю, в остальных реализациях важности этих признаков примерно одинаковая и примерно в 4 раза меньше важности признака price в 1 эксперименте.

Из эксперимента (c) видно, что важность нерелевантных признаков в пакетах rfpimp, ranger, randomforestsrc, party и partykit практически нулевая, в остальных реализациях эти признаки имеют ненулевую важность, причем у категориального признака random\_binary важность заметно ниже, по сравнению с другими нерелевантными признаками, которые принимают намного больше значений.

в эксперименте (d) аналогично, важность признака random имеет большую важность в пакетах sklearn, r randomforest.