УДК 004.85

АНСАМБЛИРОВАНИЕ РЕГРЕССИОННЫХ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ НА ПРИМЕРЕ ЗАДАЧИ HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES С РЕСУРСА KAGGLE.COM

Мисюра В. В., Пирко Д. В.
Донской государственный технический университет, Ростов-на-Дону, Российская Федерация
vvmisyura2011@gmail.com
dmitwl2000@gmail.com

Рассмотрено построение эффективных алгоритмов получения прогнозов с помощью регрессионных моделей машинного обучения на примере решения задачи с портала Kaggle.com, а именно построение алгоритма прогнозирования цен на частные дома.

Ключевые слова: Python, анализ данных, предобработка данных, машинное обучение, стекинг, блендинг, ансамблирование.

UDC 004.85

ENSEMBLING OF MACHINE LEARNING REGRESSIONAL ALGORITHMS ON THE EXAMPLE OF THE COMPETITION “HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES’’ FROM KAGGLE.COM

Misyura V. V., Pirko D. V.
Don State Technical University, Rostov-on-Don, Russian Federation
vvmisyura2011@gmail.com
dmitwl2000@gmail.com

The paper considers the construction of effective algorithms for obtaining forecasts using regression models of machine learning on the example of solving the problem with Kaggle.com portal, namely the construction of an algorithm for predicting prices for private homes.

Keywords: Python, data analysis, data preprocessing, machine learning, stacking, blending, ensemble

Введение. В современном мире ученые и практики сталкиваются с анализом огромного количества данных, а также построения моделей на их. Данная работа посвящена построению эффективных алгоритмов получения прогнозов с помощью регрессионных моделей машинного обучения на примере решения задачи с портала Kaggle.com, а именно построение алгоритма прогнозирования цен на частные дома. Ансамблирование алгоритмов машинного обучения позволит улучшить прогнозы, полученные с помощью регрессионных моделей.

Для решения поставленных задач были использованы методы машинного обучения, язык программирования Python вместе с библиотеками NumPy, Pandas, Mathplotlib, Seaborn, Scipy.

Постановка задачи. Kaggle — это платформа для проведения конкурсов по машинному обучению. Победителя определяет автоматическая система метрики, назначенная компанией-организатором. За победу в некоторых соревнованиях можно получить денежное вознаграждение либо место работы в компании. Задача, рассматриваемая в данной работе — House Prices: Advanced Regression Tech-niques. Необходимо предсказать цены домов в городе Эймс штата Айова. Даны значения 79 переменных, представляющих практически все характеристики каждого дома. К ним прилагается подробное описание. Конкурс предполагает использование регрессионных алгоритмов машинного обучения.

В качестве инструмента решения задачи будет использован высокоуровневый язык программирования общего назначения Python [1]. Будут использованы также следующие библиотеки: NumPy, Pandas, Matplotlib, Seaborn, Scipy.

Библиотека NumPy предоставляет общие математические и числовые операции в виде прескомпилированных, быстрых функций. NumPy предоставляет базовые методы для манипуляции с большими массивами и матрицами.

Высокоуровневая библиотека Pandas предназначена для анализа и обработки данных, построена поверх NumPy, что значительно увеличивает её производительность.

Matplotlib — массивная библиотека для визуализации данных двумерной графикой.

Seaborn — библиотека, основанная на Matplotlib, специализирована на визуализации статистических данных.

Scikit-learn — библиотека, построенная на SciPy, представляет собой реализацию ряда алгоритмов машинного обучения [2].

На ресурсе Kaggle.com к задаче прилагаются две выборки: обучающая и тестовая. Также есть описание каждой переменной, представленной в них. Разница между выборками в том, что в обучающей есть целевой аргумент — SalePrice, который используется для построения модели, и по условию задачи необходимо научиться его предсказывать. На рис. 1 представлен фрагмент обучающей выборки.

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig
0	1	60	RL	65.000	8450	Pave	NaN	Reg	Lvl	AllPub	Inside
1	2	20	RL	80.000	9600	Pave	NaN	Reg	Lvl	AllPub	FR2
2	3	60	RL	68.000	11250	Pave	NaN	IR1	Lvl	AllPub	Inside
3	4	70	RL	60.000	9550	Pave	NaN	IR1	Lvl	AllPub	Corner
4	5	60	RL	84.000	14260	Pave	NaN	IR1	Lvl	AllPub	FR2

Рис. 1. Фрагмент обучающей выборки

Для каждой переменной был определен ее тип: категориальная это переменная или количественная. Целевая переменная SalePrice является количественной.

Особое внимание обращено на факторы: OverallCond — общая оценка состояния дома, YearBuilt — год постройки, TotalBsmtSF — площадь подвальной части дома, GrLivArea — площадь жилой части дома над землёй.

Преобразование текстовых переменных. Для нормальной работы алгоритмов машинного обучения в выборках должны отсутствовать текстовые переменные: их нужно закодировать числовыми значениями. Авторы это сделали с помощью объекта LabelEncoder() модуля preprocessing библиотеки SciPy.

Заполнение пустых полей в выборках. Временно объединив две выборки и опустив переменную SalePrice, определяют процентное соотношение пропущенных данных в общей выборке. Обратившись к документации с описанием переменных, заменяют пропущенные данные на None — для категориальных, на 0 — для количественных признаков соответственно. Это связано с тем, что наличие пустого поля означает отсутствие соответствующего признака о доме как такового. Например, переменная PoolQC (качество бассейна). Большая часть (более 99%) оказалась пустой. Из описания данных следует, что в таком случае бассейна у дома нет. Переменные MSZoning, Electrical, KitchenQual, Exterior1st, Exterior2nd, SaleType заполняют модами, так как у них пропущено крайне малое количество данных, и они категориальные. Переменную Utilities можно удалить, так как во всех записях она равна одному значению, за исключением трех случаев. Соответственно при построении модели она ничем не поможет. LotFrontage (описывает примыкание улиц) будет сгруппировано по Neighborhood (район, в котором располагается дом), и в каждой группе будут заменены пустые значения LotFrontage на медиану группы. Это исходит из логики, что дома в одном районе примерно одинаково соединены с улицами.

Корреляционный анализ. Результатом анализа связи между целевой переменной и факторами явились графики. В случае количественных признаков это будут простые графики (рис. 2), а в случае категориальных — «ящики с усами» (рис. 3).

Рис. 2. График взаимоотношения целевой переменной SalePrice и GrLivArea

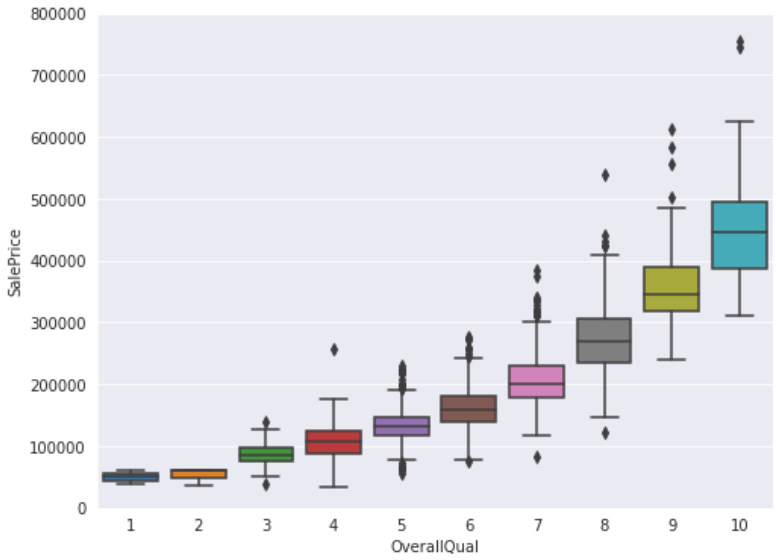

Рис. 3. График «ящиков с усами», показывающих взаимоотношения целевой переменной SalePrice и OverallQual

В выборке представлено около 70 переменных. Построив для них корреляционную матрицу, представленную на рис. 4, авторы пришли к следующим выводам. Присутствует сильная корреляция переменных TotalBsmtSF и 1stFlrSF (зачастую площадь подвала и первого этажа в домах приблизительно равны), корреляция переменных GarageCars и GarageArea является достаточно сильной (действительно, с увеличением площади гаража количество машин, которые можно поместить, растёт). Корреляция между этими переменными может рассматриваться как ситуация с наличием мультиколлинеарности.

Корреляция между целевой переменной SalePrice и факторами рассмотрена с помощью корреляционной матрицы, в которую включены первые 10 переменных, наиболее сильно коррелирующих с указанной (рис. 5).

Видна сильная корреляция SalePrice с факторами OverallQual, GrLivArea, TotalBsmtSF. Также наблюдается корреляция с переменными, описывающими наличие гаража в доме: GarageCars и GarageArea.

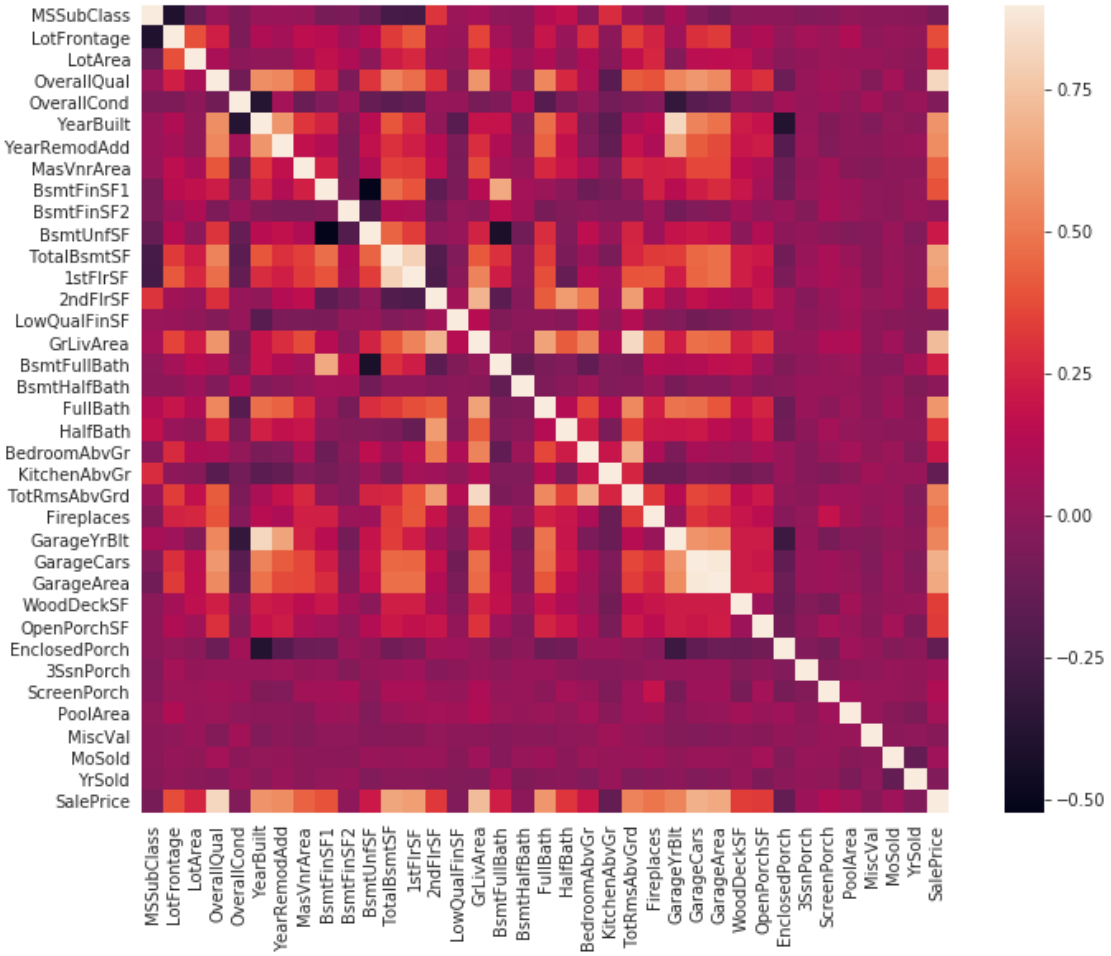

Рис. 4. Корреляционная матрица переменных исходной выборки

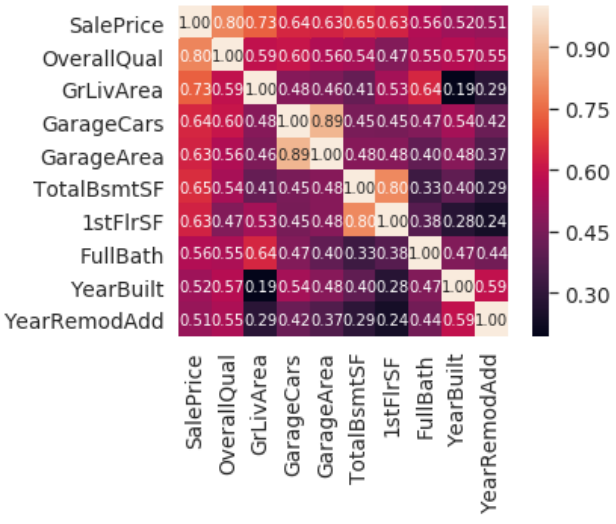

Рис. 5. Корреляционная матрица первых 10 переменных исходной выборки, наиболее сильно коррелирующих с SalePrice

Обработка выбросов. Выбросы — элементы выборки, которые не попадают под общее распределение. Их обработке стоит уделять большое внимание, так как многие алгоритмы машинного обучения неустойчивы к ним. В документации к данным содержится информация о их присутствии в обучающей выборке. На графиках зависимостей целевой переменной и переменных GrLivArea и TotalBsmtSF можно их наблюдать в правом нижнем углу. Принимается решение об их

удалении, так как при огромной площади двух домов (GrLivArea) они дешевые, такая же ситуация с выбросом TotalBsmtSF (площадь подвальной части).

Стоит отметить, что в обучающей выборке возможно нахождение и других, менее значимых выбросов, но также важно понимать, что они могут находиться и в тестовой выборке. Вместо удаления всех этих выбросов нужно сделать алгоритмы более устойчивыми к ним.

Нормализация переменных. Для статистического анализа целевой переменной было выполнено построение выборочной функции распределения (рис. 6). На графике отображены черной линией теоретическое нормальное распределение, а также значения коэффициентов асимметрии и эксцесса.

Рис. 6. График распределения переменной SalePrice

Очевидно, что распределение переменной SalePrice отклоняется от нормального и потребует нормализации. Это важно, так как алгоритмы работают значительно лучше с нормализованными переменными. Для проверки будет использована также гистограмма распределения переменной и QQ-plot. QQ-plot используется для сравнения двух распределений путем построения их квантилей относительно друг друга. В данном случае это будут квантили теоретического распределения и распределения переменной. В случае нормального распределения оно на QQ-plot должно повторять диагональ теоретического. Будет получена гистограмма и QQ-plot целевая переменная SalePrice (рис. 7).

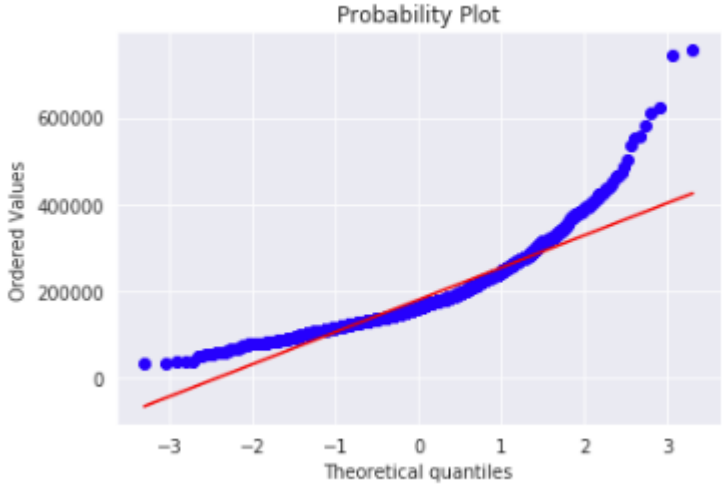

Рис. 7. График QQ-plot переменной SalePrice

Можно наблюдать правостороннюю асимметрию на гистограмме. На QQ-plot распределение не соответствует нормальному.

Для нормализации будет применено логарифмирование, и получится результат, представленный на рис. 8–9.

Рис. 8. График распределения переменной ln(SalePrice)

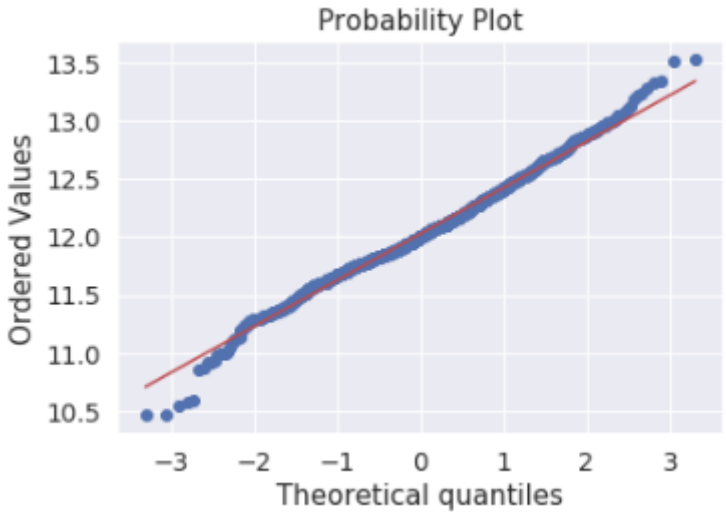

Рис. 9. График QQ-plot переменной ln(SalePrice)

Асимметрия переменной SalePrice была исправлена, и её распределение теперь близко к нормальному. Также необходимо проверить и исправить распределения других переменных, которые сильно коррелируют с SalePrice и важны для моделей. Для этого к переменным с высокой асимметрией (коэффициент больше 0,75) применяется логарифмирование.

Обзор алгоритмов машинного обучения, использующихся для построения модели. В данной задаче используется трансдуктивное обучение. При таком обучении делаются выводы о тестовых данных на основе данных обучения.

Будут использоваться алгоритмы, основанные на регрессии. В линейной регрессии алгоритм может стать нестабильным, часто появляется проблема переобучения. Переобучение — ситуация, когда модель хорошо описывает обучающую выборку, но даёт низкие результаты на других. Решить данную проблему можно путём наложения ограничений на регрессионную модель, то есть её регуляризацией [3]. Для построения моделей будут использоваться следующие алгоритмы

машинного обучения: LASSO Regression, Kernel Ridge Regression, Elastic Net Regression, Gradient Boosting Regression, XGBoost, LightGBM.

LASSO Regression — регрессия, при которой добавляется 1 параметр регуляризации, имеющий смысл штрафа за сложность. При этом некоторые коэффициенты переменных становятся равными нулю, что делает модель более совершенной, оставляя информативные признаки.

Kernel Ridge Regression — усовершенствованная линейная регрессия, при которой на коэффициенты параметров накладываются некоторые ограничения, что приближает результат к реальности. Также применяется метод для борьбы с избыточностью данных в случаях, когда наблюдается мультиколлинеарность данных.

Elastic Net Regression — обобщение регрессии с регуляризацией. Устанавливаются два штрафных параметра, объединяются Ridge Regression и LASSO Regression.

Gradient Boosting Regression — используется множество слабых алгоритмов регрессии для создания более точного, стабильного и надежного.

XGBoost — используется множество слабых линейных алгоритмов и алгоритмов, основанных на деревьях решений, для создания более точного, стабильного и надежного. LightGBM — аналог XGBoost, но используются только решающие деревья

Построение базовых моделей. При помощи функций библиотек Scikit-learn строятся простые модели, основанные на вышеизложенных алгоритмах. Стоит отметить, что при проверке модели нужно использовать кросс-валидацию, чтобы избежать переобучения.

Результаты моделей оцениваются при помощи метода среднеквадратической ошибки (RMSD) (1). Чем эта ошибка меньше, тем лучше модель решает задачу.

$$RMSD = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}} \quad (1)$$

После оценки моделей получают результаты, представленные на рис. 10.

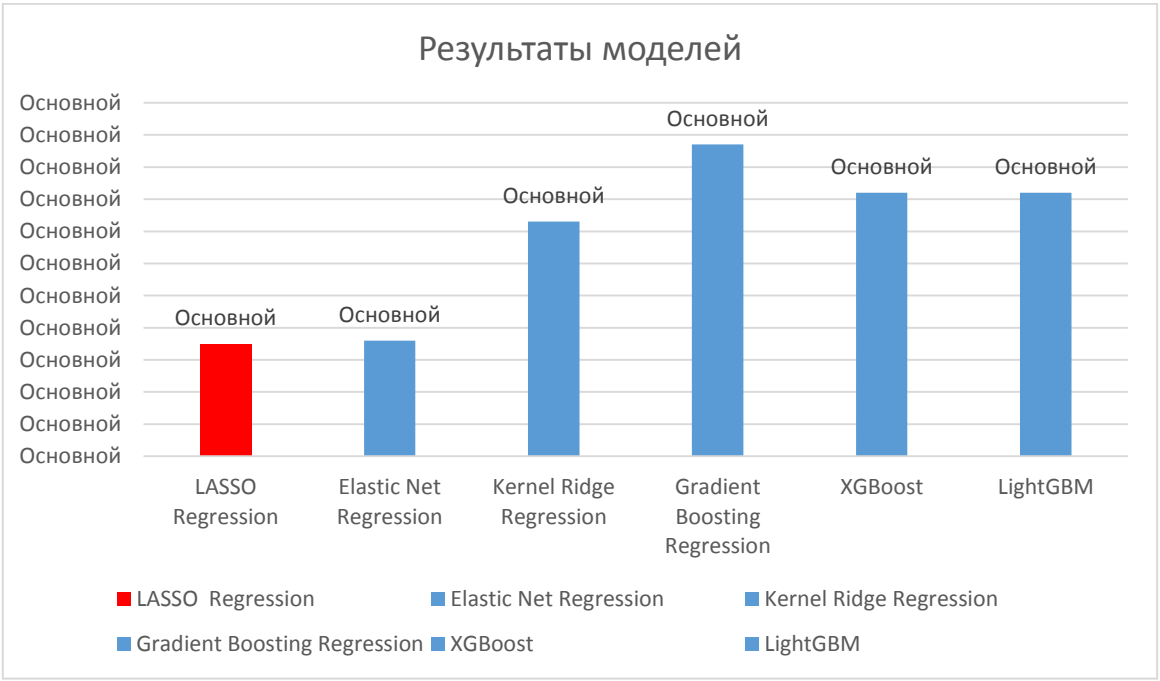

Рис. 10. Гистограмма результатов базовых алгоритмов машинного обучения. Лучше всех справился с задачей алгоритм LASSO Regression

Ансамблирование алгоритмов машинного обучения. Ансамблирование — это использование нескольких алгоритмов для решения одной задачи машинного обучения. Авторы будут ис-

пользовать стекинг, который является одним из популярных способов ансамблирования алгоритмов. Суть данного метода заключается в следующем: при обучении нескольких алгоритмов в задачах регрессии используется их среднее, а в задачах классификации — голосование большинству, но можно вместо данных операций усреднения и голосования использовать другой алгоритм, то есть метаалгоритм.

Простейшая схема стекинга — блендинг: обучающую выборку делят на две части. На первой обучают базовые алгоритмы. Затем получают их ответы на второй части и на тестовой выборке. Ответ каждого алгоритма можно рассматривать как новый признак (метапризнак). На метапризнаках второй части обучения настраивают метаалгоритм. Затем запускают его на метапризнаках теста и получают ответ.

Самый большой недостаток блендинга — деление обучающей выборки. Получается, что ни базовые алгоритмы, ни метаалгоритм не используют всего объёма обучения (каждый — только свой кусочек).

Способом использования всей обучающей выборки при ансамблировании алгоритмов является реализация классического стекинга. Выборку разбивают на части (так называемые фолды), затем, последовательно перебирая фолды, обучают базовые алгоритмы на всех фолдах, кроме одного, а на оставшемся получают ответы базовых алгоритмов и трактуют их как значения соответствующих признаков на этом фолде. Для получения метапризнаков объектов тестовой выборки базовые алгоритмы обучают на всей обучающей выборке и берут их ответы на тестовой.

Главный недостаток стекинга в том, что метапризнаки на обучении и на тесте разные. Например, в рассматриваемой задаче метапризнак на обучающей выборке — это не ответы какого-то конкретного регрессора, он состоит из кусочков, которые являются ответами разных регрессий. А метапризнак на контрольной выборке вообще является ответом совсем другой регрессии, настроенной на всём обучении.

Зачастую с указанными недостатком борются обычной регуляризацией или добавлением к метапризнакам нормального шума [4].

Для ансамблирования алгоритмов машинного обучения был создан класс, с помощью которого авторы провели ансамблирование регрессионных алгоритмов методом стекинга для получения одной модели, а затем добавили алгоритмы, использующие бустинг, для получения новой, более совершенной модели. После оценки этих моделей будут получены результаты, представленные на рис. 11.

Рис. 11. Гистограмма результатов ансамблирования алгоритмов машинного обучения

Стекинг значительно улучшил результаты прогноза. Поэтому целесообразно применить данный алгоритм для построения модели по тестовой выборке, которую и отправили для оценки на Kaggle.com.

Заключение. В результате модель авторов заняла 399 место из 4477 возможных со счетом 0.11549 (RMSE). Таким образом, можно сделать вывод, что задача по построению модели на основе ансамблирования регрессионных алгоритмов машинного обучения была решена успешно. Язык программирования Python показал свою эффективность в решении задач, связанных с анализом и обработкой данных, а также с машинным обучением. Его библиотеки позволяют решать прикладные задачи профессионального уровня.

Библиографический список

1. Python 3.8.0 documentation [Электронный ресурс] / Python 3.8.0 documentation. — Режим доступа : https://docs.python.org/3/ (дата обращения : 01.02.2019).

2. Ten handy python libraries for (aspiring) data scientists [Электронный ресурс] / Big Data. — Режим доступа : https://bigdata-madesimple.com/ten-handy-python-libraries-for-aspiring-data-scientists/ (дата обращения : 01.02.2019).

3. Шитиков, В. К. Классификация, регрессия, алгоритмы Data Mining с использованием R / В. К. Шитиков, С. Э. Мастицкий [Электронный ресурс] / Электронная книга. — Режим доступа : https://ranalytics.github.io/data-mining/ (дата обращения : 01.03.2019).

4. Дьяконов, А. Стекинг (Stacking) и блендинг (Blending) / А. Дьяконов [Электронный ресурс] / Анализ малых данных. КвазиНаучный блог Александра Дьяконова. — Режим доступа : https://dyakonov.org/2017/03/10/c%D1%82%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3-stacking-%D0%B8-%D0%B1%D0%BB%D0%B5%D0%BD%D0%B4%D0%B8%D0%BD%D0%B3-blending/ (дата обращения : 01.03.2019).