Научная статья
Original article
УДК 004.852

ПРОГНОЗИРОВАНИЕ СТОИМОСТИ НЕДВИЖИМОСТИ НА ОСНОВЕ МОДЕЛИ ЛИНЕЙНОЙ РЕГРЕССИИ

FORECASTING THE COST OF REAL ESTATE BASED ON THE MODEL LINEAR REGRESSION

Катковская Ксения Владимировна, старший преподаватель кафедры «Информационные системы», ФГБОУ ВО СГУВТ Россия, г. Новосибирск.
Дудко Антон Дмитриевич, студент 1 курса магистратуры, факультет «Электромеханический факультет», ФГБОУ ВО СГУВТ, Россия, г. Новосибирск

Katkovskaya Ksenia Vladimirovna, Senior Lecturer, Department "Information Systems", FGBOU VO SGUVT, Russia, Novosibirsk.
Dudko Anton Dmitrievich, 1st year master's student, faculty "Electromechanical Faculty", FGBOU VO SGUVT, Russia, Novosibirsk

Аннотация: В статье рассмотрено прогнозирование стоимости недвижимости с использованием алгоритма линейной регрессии на языке Python при помощи библиотеки Scikit-Learn. Показана важность прогнозирования сфере недвижимости, так как жилой рынок я системообразующей частью региональной экономики. Представлено описание метода линейной регрессии и алгоритма машинного обучения.

Приведены графики корреляции показателей прогнозирования и кривые ошибок машинного обучения. Предложены способы снижения тестовых ошибок и ошибок обучения с помощью кросс-валидации. Рассмотрены понятия гиппер-параметров обучения и их регуляризации.

Summary: The article discusses forecasting the value of real estate using a linear regression algorithm in Python using the Scikit-Learn library. The importance of forecasting the real estate sector is shown, since the residential market is a system-forming part of the regional economy. The description of the linear regression method and the machine learning algorithm is presented. Graphs of correlation of forecasting indicators and machine learning error curves are given. Methods of reducing test errors and learning errors using cross-validation are proposed. The concepts of hyper-parameters of learning and their regularization are considered.

Ключевые слова: Прогнозирование, Недвижимость, Стоимость, Линейная регрессия, Машинное обучение, Python.

Keywords: Prediction, Real Estate, Price, Linear Regression, Machine Learning, Python.

На сегодняшний день задача прогнозирования является одним из наиболее важных шагов в управлении, планировании и принятии решений в любой предметной области. Практически во всех случаях прогноз сводиться к разработке разных вариантов моделей развития объекта, из которых в дальнейшем необходимо выбрать наиболее подходящую по тем или иным критериям. В силу большого объема показателей и набора данных, необходимых для построения модели прогнозирования, целесообразно использовать информационные технологии, как механизм реализации вычислений и визуализации данных. В качестве таких информационных современных средств может выступать специальные среды разработки программ прогнозирования на языке программирования Python.

Прогнозирование представляет собой научно обоснованное суждение о будущих состояниях объекта и (или) об альтернативных путях достижения

этого состояния. Необходимость прогнозирования вызвана тем обстоятельством, что будущие состояния объекта имеют большое значение для решений, принимаемых в настоящий момент. Имеет место неопределенность, связанная с возможным развитием ситуации, которую полностью устранить невозможно [1].

Рынок жилой недвижимости является одним из системообразующих элементов региональной экономики, ее инвестиционным активом, фактором привлекательности территории. Таким образом, выбор темы данной статьи обусловлен ролью, которую жилая недвижимость играет в экономической и социальной жизни региона, а также наличием проблем, возникающих в процессе оценки стоимости объектов жилой недвижимости с учетом многих факторов [2].

Ранее в статье упоминалось, что одним из современных средств реализации алгоритмов прогнозирования является язык программирования Python. Данный язык имеет ряд удобных библиотек машинного обучения и научных расчетов: Pandas, NumPy, seaborn, Scikit-Learn, matplotlib.pyplot, которые позволяют построить рабочие модели. Библиотека Pandas отвечает за обработку и анализ данных, NumPy – за выполнение операций линейной алгебры и численных преобразований, seaborn и matplotlib.pyplot – отвечает за визуализацию данных, Scikit-Learn – является непосредственно библиотекой машинного обучения [3, 4, 5, 6, 7].

Машинное обучение – это методики анализа данных, которые позволяют аналитической системе обучаться в ходе решения множества сходных задач. Машинное обучение базируется на идее о том, что аналитические системы могут учиться выявлять закономерности и принимать решения с минимальным участием человека.

Алгоритмы машинного обучения помогают ответить на вопросы, ответ на которые сложно найти с помощью анализа, выполняемого вручную. С помощью алгоритмов линейной регрессии отображается либо прогнозируется связь между двумя переменными или факторами путем вписывания в

представление данных непрерывной прямой линии. Эта линия часто вычисляется с помощью функции Squared Error Cost (Стоимость среднеквадратической погрешности). Линейная регрессия — один из наиболее популярных типов регрессионного анализа.

Для работы использовались данные из набора (далее Датасет) встроенной библиотеки Scikit-Learn , показатели которых представлены ниже.

Основные характеристики набора данных:

- MedInc – Медианный доход MedInc в группе блоков;
- HouseAge – средний возраст дома в блочной группе;
- AveRooms – Среднее количество комнат на семью;
- AveBedrms – Среднее количество спален на семью;
- Population – Население, блок, группа, население;
- AveOccup – Среднее количество членов семьи;
- Latitude – Широта группы блоков широта;
- Longitude – Долгота группы блока долготы;
- TARGET – медианное значение стоимости занимаемых домов.

Для начала необходимо построить тепловую матрицу, для выбора оптимального показателя относительно которого будет реализовываться прогноз (рис.1). Затем необходимо посмотреть степень корреляции параметров между собой, чтобы подкрепить оптимальность выбора.

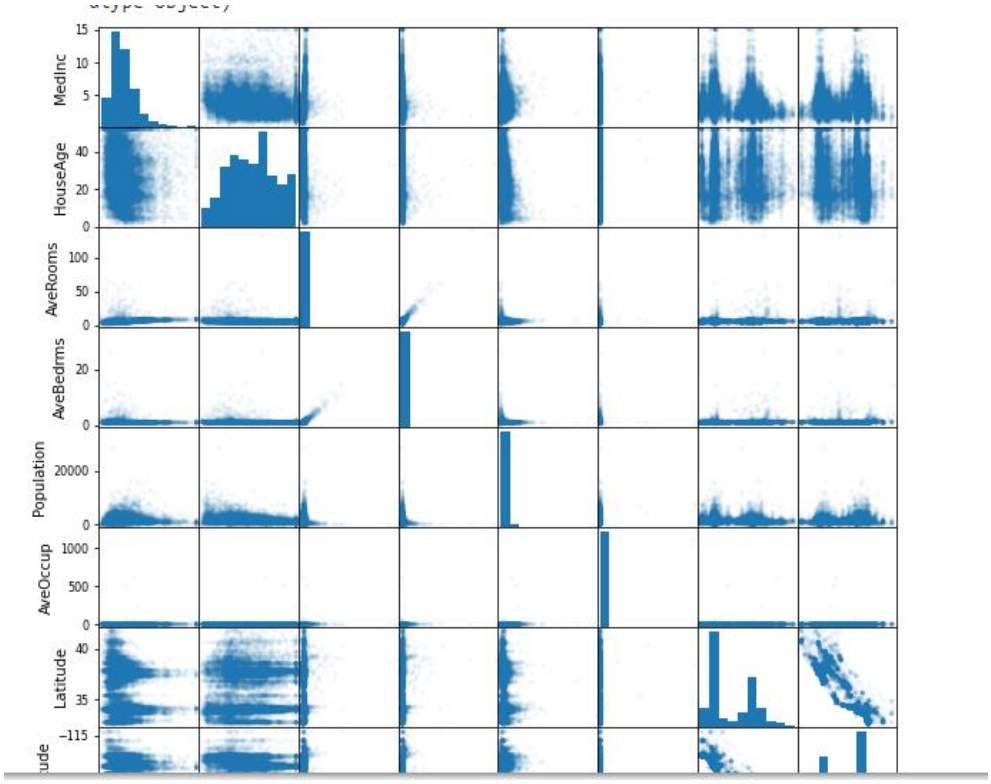


Рисунок 1 – Тепловая матрица

Поскольку набор данных не очень большой, используем метод corr () для вычисления стандартного коэффициента корреляции между каждой парой атрибутов, также известного как коэффициент корреляции.

```
Housing_df.corr()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
MedInc	1.000000	-0.119034	0.326895	-0.062040	0.004834	0.018766	-0.079809	-0.015176
HouseAge	-0.119034	1.000000	-0.153277	-0.077747	-0.296244	0.013191	0.011173	-0.108197
AveRooms	0.326895	-0.153277	1.000000	0.847621	-0.072213	-0.004852	0.106389	-0.027540
AveBedrms	-0.062040	-0.077747	0.847621	1.000000	-0.066197	-0.006181	0.069721	0.013344
Population	0.004834	-0.296244	-0.072213	-0.066197	1.000000	0.069863	-0.108785	0.099773
AveOccup	0.018766	0.013191	-0.004852	-0.006181	0.069863	1.000000	0.002366	0.002476
Latitude	-0.079809	0.011173	0.106389	0.069721	-0.108785	0.002366	1.000000	-0.924664
Longitude	-0.015176	-0.108197	-0.027540	0.013344	0.099773	0.002476	-0.924664	1.000000

Рисунок 2 – Стандартный коэффициент корреляции

Видно, что median_income сильно коррелирует с конечным результатом. Используем данный признак для прогнозирования TARGET.

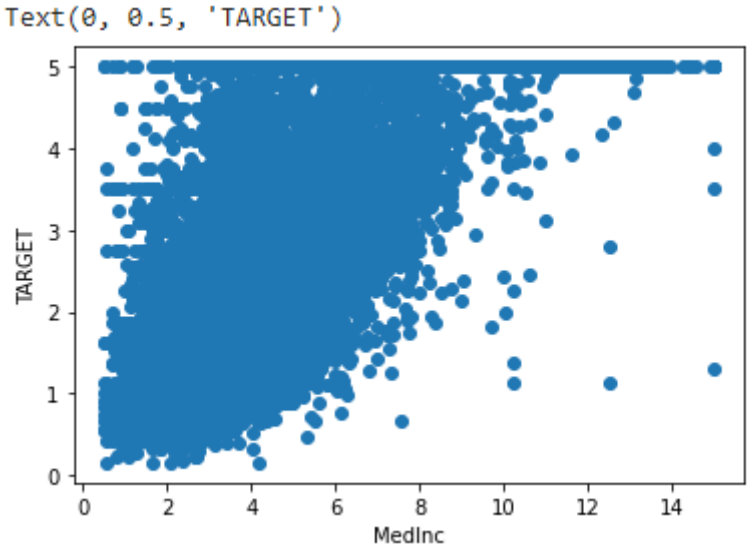


Рисунок 3 – Корреляция признаков MedInc и TARGET

По оси X расположен параметр MedInc, а по оси Y – TARGET. Наиболее многообещающим признаком для прогнозирования средней цены на жилье является средний доход т. к. корреляция очень высокая и можно четко видеть восходящую тенденцию, а точки данных не очень разбросаны.

Поэтому предсказание TARGET можно осуществлять на основе MedInc. В качестве оценочной метрики будем использовать среднюю квадратическую ошибку (meansquarederror – MSE). Необходимо, чтобы среднее расстояние между предсказанными и фактическими показателями было минимальным. Для этого рассмотрим модель линейной регрессии, которая имеет вид:

$$\widehat{y_i} = \beta_0 + \sum_j B_j x_j, i$$

Где j – количество имеющихся признаков (независимых переменных), B_j - это наши коэффициенты, β_0 – смещение.

Таким образом, у получим модель и функцию оценки. Наша задача состоит в том, чтобы найти бета-значения, которые минимизируют MSE для наших данных. Для этого будем использовать метод градиентного спуска.

Этот метод найдет глобальный минимум, если функция выпуклая. В противном случае мы можем быть уверены только в том, что достигнем локального минимума.

Для построения графика мы взяли диапазон значений коэффициентов для MedInc, и для каждого из них рассчитали MSE на основе наших данных. Затем отобразили полученные значения и получили приведенную выше выпуклую кривую Таким образом, наша функция MSE с нашей моделью линейной регрессии всегда будет выпуклой. Это означает: мы можем использовать градиентный спуск, чтобы найти оптимальные коэффициенты для нашей модели.

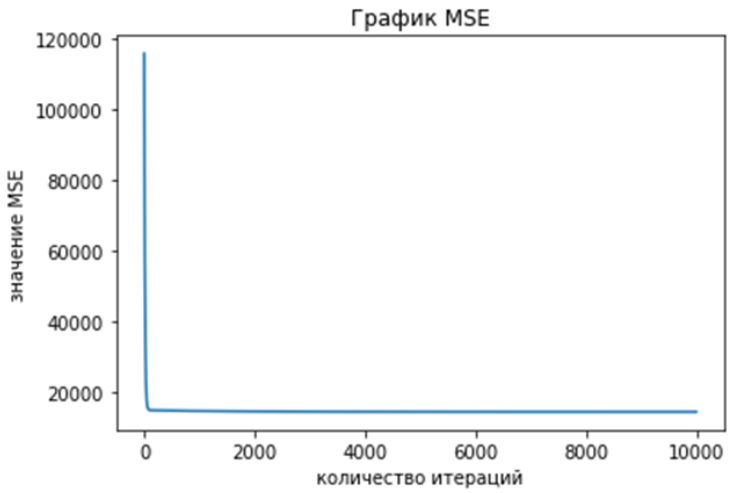


Рисунок 4 - График MSE

График показывает, что MSE уменьшается со временем по мере выполнения алгоритма. Это означает, что мы постоянно приближаемся к оптимальному решению.

Построим на одном рисунке фактические и прогнозные значения TARGET.

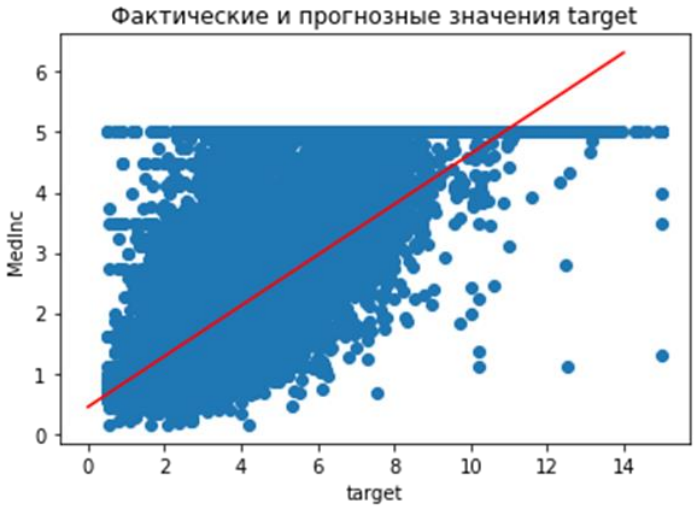


Рисунок 5 - Фактические и прогнозные значения TARGET

Запуск градиентного спуска показал, что оптимальное смещение составляет 0.44, а оптимальный наклон линии регрессии равен 0.41. На приведенном выше графике эта линия показана поверх наших данных, она выглядит как аппроксимирующая прямая.

Воспользуемся библиотекой scikit-learn, чтобы построить линейную регрессию по нашим данным. SGDRegressor строит линейную регрессию с использованием градиентного спуска.

Рассматривая построенный выше график, мы можем заметить, что между данными показателями существует полиномиальная связь. Линейная регрессия хорошо подходит в случае линейной зависимости, но, если мы добавим полиномиальные показатели, такие как MedInc, мы сможем установить более сложные отношения.

Функция PolynomialFeaturesc генерировала новую матрицу показателей, состоящую из всех их полиномиальных комбинаций со степенью меньше или равной указанной (в нашем примере – 1). Затем мы нормализовали эти данные и передали их нашей модели. Так мы получили улучшенную тренировочную RMSE, равную 142.0973622358076.

Посмотрим на коэффициенты, которым научилась наша модель:

('HouseAge', -1.2734039225808715),

('Population', -1.992007181192062),

('MedInc', 5.100484198363296),

('AveBedrms', 6.391608747578457),

('AveRooms', -9.571767992150356),

('Latitude', 10.347183104860004),

('Longitude', 10.832471671312222),

('AveOccup', -70.12352781871152)

Каждый коэффициент представляет собой среднее изменение цены на жилье при изменении соответствующего показателя на единицу с условием, что все остальные показатели остаются неизменными.

Часто в машинном обучении полезно знать доверительный интервал для оценок. Есть разные способы сделать это, но одним из общих методов является использование bootstrap. Bootstrap – это случайная выборка на основе наших данных, и эта выборка того же размера, что и исходные данные. Так мы можем создать несколько представлений одних и тех же данных. Создадим 1000 bootstrap-семплов (bootstrap-образцов, примеров) наших данных.

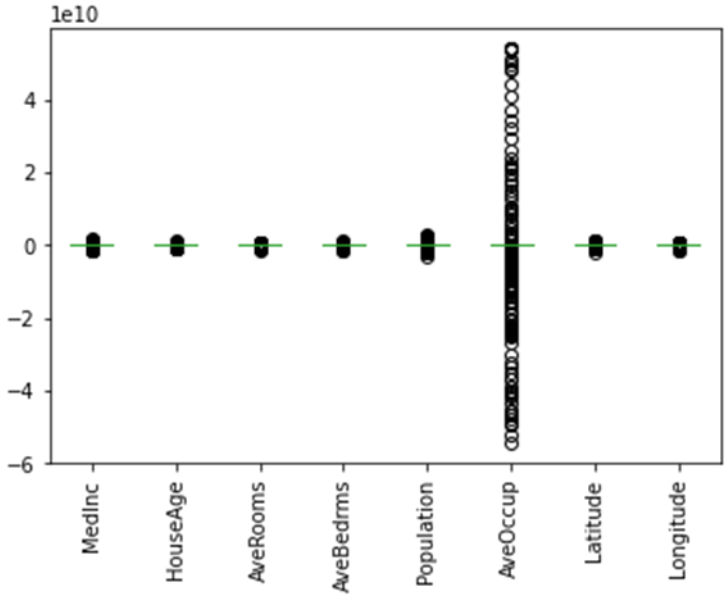


Рисунок 6 - Диаграмма размаха

На полученной диаграмме размаха показан диапазон значений коэффициентов, которые мы получили для каждого показателя для всех моделей, которые мы обучили. AveOccup – особенно интересен, потому что значения коэффициентов были как положительными, так и отрицательными,

что является хорошим признаком того, что, вероятно, нет никакой связи между количеством людей в семье и стоимостью.

До этого момента мы тренировались на всех имеющихся данных. По этой причине нам трудно оценить, насколько хорошо работает наша модель.

Разделим наши данные на две части: обучающий и тестовый наборы.

Train MSE: 941631.3998402633

Test MSE: 13085.051140369533

Видим, что MSE для тренировочных данных и для данных тестирования очень разные. Большие значения предполагают наличие высокого смещения(bias). Чтобы разобраться в этом, построим график обучения, который будет отображать функцию ошибок MSE с различными объемами данных, используемых для тренировки.

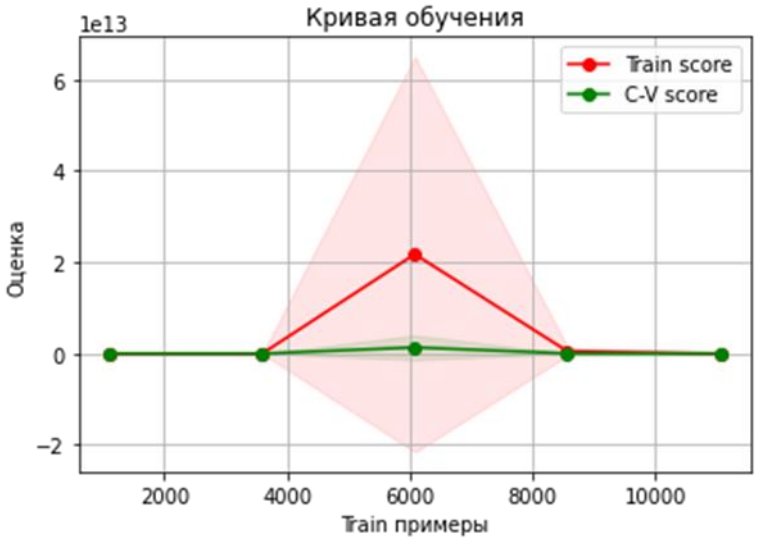


Рисунок 7 - Кривая обучения

Полученный график больше похож на проблему с большим смещением (bias – показывает асимметрию выборки данных), поскольку две наши кривые очень близки и сглаживаются.

Устранение проблем высоких bias/variance может достигаться двумя путями:

Для высокой дисперсии:

- Получите больше данных для обучения
- Попробуйте меньший набор показателей

- Используйте менее сложную модель
- Добавьте регуляризацию;

Для высокого смещения:

- Попробуйте увеличить число показателей
- Перейдите на более сложную модель;

Для устранения понадобится третий набор данных – Датасет для валидации. По сути, мы можем разбить наши обучающие данные на две части: обучающий и проверочный датасеты. Все модели будут обучены на тренировочном множестве, а затем протестированы на нашем проверочном наборе. Затем мы выберем модель, которая лучше всего справляется с проверкой, и посмотрим, насколько удачно она пройдет тестирование. Результаты тестирования покажут, как хорошо наша модель будет работать с незнакомыми данными, и на этом мы завершим процесс разработки.

На практике вместо создания единого множества для проверки мы часто используем k-блочную кросс-валидацию.

Лучшая оценка MSE в итоге выглядит следующим образом: – 0.5230197179680892

Здесь фактически использовался рандомизированный поиск (RandomizedSearchCV), который обычно лучше, чем поиск по всем возможным значениям. Мы также установили cv=3, чтобы иметь 3 блока и использовали отрицательную MSE, потому что функции CV в scikit-learn пытаются максимизировать значение.

В качестве средства борьбы с высокой дисперсией используют регуляризацию. Для линейной регрессии она принимает форму трех популярных подходов. Все эти методы сосредоточены на идее ограничения того, насколько большими могут быть коэффициенты наших показателей.

Идея состоит в том, что если мы переоцениваем влияние предсказателя (большое значение коэффициента), то, вероятно, мы переобучаемся. Примечание: у нас все еще могут быть просто большие коэффициенты.

Регуляризация говорит о том, что уменьшение MSE должно оправдывать увеличение значений коэффициентов.

- Регуляризация L1 (Lasso): добавляется сумма абсолютных значений коэффициентов к функции оценки. Этот метод может принудительно обнулить коэффициенты, что затем может быть средством выбора показателей.
- Регуляризация L2 (Ridge): добавляется сумму квадратов значений коэффициентов к функции оценки.
- Эластичная сетка: добавляются обе и выбирается, способ утяжеления.

Каждый из этих методов принимает весовой множитель, который говорит, насколько сильное влияние регуляризация будет иметь на функцию оценки. В scikit-learn такой параметр называется альфа. Альфа равный 0 не добавит штрафа, в то время как высокое его значение будет сильно наказывать модель за наличие больших коэффициентов. Возможно использовать кросс-валидацию, чтобы найти хорошее значение для альфы.

В результате тестовые и тренировочные оценки выглядят следующим образом: Train MSE: 0.5814129001512113, Test MSE: 0.585057155561447.

На основании проделанной работы можно сделать следующий вывод: после кросс-валидации и настройки гиппер-параметров, а также регуляризации, тестовые ошибки и ошибки обучения уменьшились и стали более близки к друг другу. Следовательно, проблема с прогнозом цен на недвижимость во встроенном наборе данных, действительно заключалась в большой дисперсии.

Литература

1. Раков, В. И. Системный анализ (начальные понятия) / В. И. Раков. – учебное пособие – М.: Академия Естествознания, 2012. – 239 с. – URL : https://monographies.ru/ru/book/section?id=5972.
2. Использование регрессионного анализа в оценке стоимости объектов регионального рынка недвижимости: публикация И. В. Бурова

[Электронный ресурс]. URL: https://cyberleninka.ru/article/n/ispolzovanie-regressionnogo-analiza-v-otsenke-stoimosti-obektov-regionalnogo-rynka-nedvizhimosti/viewer (дата обращения: 25. 05.2022).

3. Обрушение к странице библиотеки Pandas [Электронный ресурс]. URL: https://pandas.pydata.org/ (дата обращения: 25. 05.2022).
4. Обрушение к странице библиотеки NumPy [Электронный ресурс]. URL: https://numpy.org/ (дата обращения: 25. 05.2022).
5. Обрушение к странице библиотеки Seaborn [Электронный ресурс]. URL: https://seaborn.pydata.org/ (дата обращения: 25. 05.2022).
6. Обрушение к странице библиотеки Scikit-Learn [Электронный ресурс]. URL: https://scikit-learn.org/stable/#/ (дата обращения: 25. 05.2022).
7. Обрушение к странице библиотеки Matplotlib [Электронный ресурс]. URL: https://matplotlib.org/ (дата обращения: 25. 05.2022).

Literature

1. Rakov, V. I. System analysis (initial concepts) / V. I. Rakov. - textbook - M .: Academy of Natural Sciences, 2012. - 239 p. – URL: https://monographies.ru/ru/book/section?id=5972.
2. The use of regression analysis in assessing the value of objects of the regional real estate market: publication by IV Burov [Electronic resource]. URL: https://cyberleninka.ru/article/n/ispolzovanie-regressionnogo-analiza-v-otsenke-stoimosti-obektov-regionalnogo-rynka-nedvizhimosti/viewer (date of access: 05/25/2022).
3. Crash to the Pandas library page [Electronic resource]. URL: https://pandas.pydata.org/ (accessed 25.05.2022).
4. Crash to NumPy library page [Electronic resource]. URL: https://numpy.org/ (date of access: 05/25/2022).
5. Collapse to the Seaborn library page [Electronic resource]. URL: https://seaborn.pydata.org/ (accessed 25.05.2022).
6. Collapse to the Scikit-Learn library page [Electronic resource]. URL: https://scikit-learn.org/stable/#/ (accessed 25.05.202

7. Collapse to the Matplotlib library page [Electronic resource]. URL: https://matplotlib.org/ (accessed 25.05.2022).

© Катковская К.В., Дудко А.Д. 2022 Научно-образовательный журнал для студентов и преподавателей «StudNet» №6/2022.

Для цитирования: Катковская К.В., Дудко А.Д. ПРОГНОЗИРОВАНИЕ СТОИМОСТИ НЕДВИЖИМОСТИ НА ОСНОВЕ МОДЕЛИ ЛИНЕЙНОЙ РЕГРЕССИИ// Научно-образовательный журнал для студентов и преподавателей «StudNet» №6/2022.