Инженерный вестник Дона, №1 (2020)
ivdon.ru/ru/magazine/archive/n1y2020/6254

ГИС-технологии - современный помощник в подборе недвижимости.

М.Р. Степанова, А.В. Максимченко, К.С. Петров, Н.Ю. Невмывайченко, Е.А. Куклев, М.И. Кохан, А.А. Габриелян

Донской Государственный Технический Университет, Ростов-на-Дону

Аннотация: Авторы предлагают рассмотреть интересное решение поиска недвижимости с использованием ГИС - технологий. Этот метод может значительно сократить время на выбор объекта недвижимости. В этой статье показано, как возможности обработки данных научных вычислительных инструментов для Python и возможности визуализации и анализа геопространственных данных ArcGIS могут использоваться для построения модели, которая генерирует короткий список домов, соответствующих потребностям и желаниям потенциальных покупателей. И хотя выбор дома сугубо индивидуален, есть масса общих факторов, таких, как общая площадь, количество этажей и комнат, наличие гаража и многое другое. Также, покупая дом, вы ищете близость к таким видам услуг, как продукты питания, аптеки, пункты неотложной помощи, детские учреждения, парки и т.д. Модуль геокодирования ArcGIS API для Python можно использовать для поиска таких объектов на заданном расстоянии вокруг дома. Данный метод предлагает объединить все необходимые условия в одну систему, которая позволит значительно сократить список объектов недвижимости находящихся на рынке и выбрать самые оптимальные варианты.
Ключевые слова: ГИС, ArcGIS, недвижимость, Python, карты, гистограммы, фильтр, DataFrame, пространственное распределение, геокодирование.

Многие из нас на то, чтобы арендовать или купить дом, тратят большое количество времени. Обычно многие факторы, которые мы рассматриваем, находятся под сильным влиянием местоположения. В этой статье показано, как возможности обработки данных научных вычислительных инструментов для Python и возможности визуализации и анализа геопространственных данных ArcGIS могут использоваться для построения модели, которая генерирует короткий список домов, соответствующих потребностям и желаниям потенциальных покупателей.

Данные по жилью, собранные с популярного сайта недвижимости, представляются в нескольких файлах CSV разных размеров. Данные читаются с использованием электронной библиотеки «Pandas» в качестве объектов таблиц данных. Эти таблицы данных являются основой как для пространственного, так и для атрибутивного анализа. Файлы CSV

объединяются для получения первоначального списка - около 4200 объектов, которые выставлены на продажу [1-3].

Первым и критическим шагом в любом проекте анализа данных и машинного обучения является обработка и очистка данных. До этого данные страдают от повторений, недопустимых символов в именах столбцов и выбросов.

С помощью «Pandas» чрезвычайно легко очистить табличные данные. Показатели центральности, такие как среднее значение и медиана, используются для расчета отсутствующих показателей частоты в столбцах: «цена жилья», «цена за квадратный метр», «количество квадратных метров», тогда как показатели «mode» использовались для таких столбцов, как ZIP. Строки, в которых отсутствовали значения, такие как: «Спальные комнаты», «Ванные», «Цена», «Год постройки» и «Длительность» были отброшены, поскольку не было надёжного способа сохранить эти данные. После удаления этих записей 3652 объекта были доступны для анализа.

Из первых двух гистограмм (рис. 1) видно, что во всех домах одинаковое количество кроватей и ванн. Это просто не соответствует действительности и свидетельствует о том, что небольшое количество высоких значение (выбросов) искажает распределение.

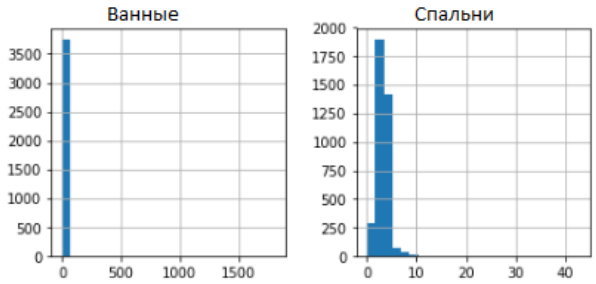

Рис. 1. Гистограммы количества ванных комнат и спален, которые показывают, что данные сильно искажены выбросами.

Существуют разные подходы к фильтрации выбросов. Популярным является 6-сигма-фильтр, который удаляет значения, превышающие 3 стандартных отклонений от среднего. Этот фильтр предполагает, что данные соответствуют нормальному распределению и использует среднее значение в качестве меры центральности. Однако, когда данные сильно страдают от выбросов, как в этом случае, среднее значение может быть искажено.

Фильтр Inter Quartile Range (IQR) использует медиану, которая является более надёжной мерой центральности. Он может более надёжно отфильтровывать выбросы, которые находятся на заданном расстоянии от медианы. После удаления выбросов с помощью IQR-файла распределение числовых столбцов выглядит намного правильнее. (рис.2)

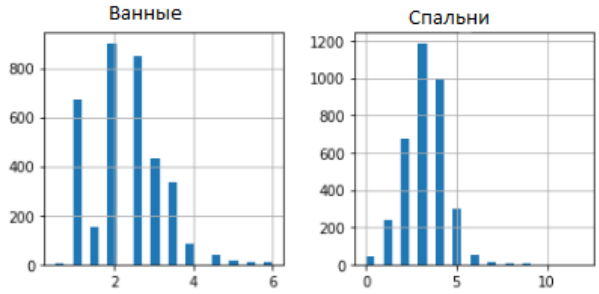

Рис. 2. Улучшенные гистограммы тех же числовых столбцов после внесения пропущенных значений и удаления выбросов.

Исследовательский анализ данных Pandas представляет эффективный API для изучения статистического распределения числовых столбцов. Чтобы изучить пространственное распределение этого набора данных, необходимо использовать ArcGIS API for Python. Код 1 имеет вид:

```
import pandas as pd
import matplotlib.pyplot as plt
from pprint import pprint
%matplotlib inline
import seaborn as sns
```

from arcgis.gis import GIS

from arcgis.features import Feature, FeatureLayer, FeatureSet, GeoAccessor, GeoSeriesAccessor

Классы GeoAccessor и GeoSeriesAccessor добавляют пространственные возможности к объектам DataFrame Pandas. Любой обычный объект, DataFrame со столбцами местоположения, может быть преобразован в пространственно-включенный DataFrame с использованием этих классов.

Аналогично построению статистической диаграммы из объекта DataFrame, пространственная диаграмма с использованием виджета интерактивной карты может быть построена из пространственно-включенного DataFrame. Рендеры, такие как тепловые карты, могут быть применены, чтобы быстро визуализировать плотность списка.

Построение пространственно-включенного DataFrame с визуализацией тепловой карты показывает наличие горячих точек. ArcGIS API для Python поставляется с набором сложных средств, которые помогают визуализировать пространственные изменения в столбцах, такие как «Цена», «Название», «Возраст здания» или «Площадь». Объединение карт со статистическими графиками дает более глубокое понимание и исследует общие предположения. (рис.3)

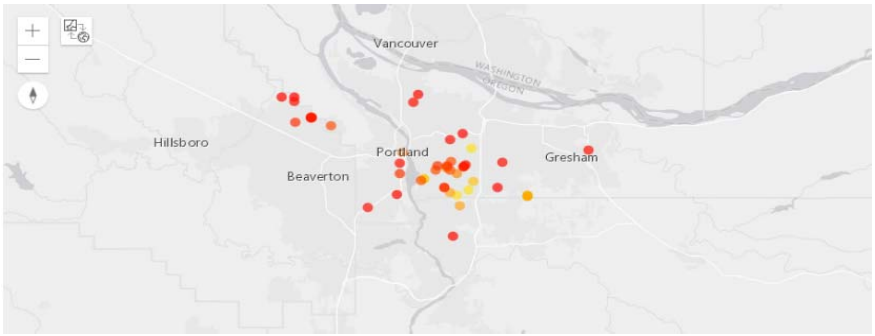

Рис. 3. Построение пространственно-включенного кадра данных с визуализацией тепловой карты показывает наличие горячих точек.

Значения кода 2, включающие в себя внутренние особенности домов, были использованы для создания короткого списка. Код 2 имеет вид:

```
>>> filtered_df = prop_sdf [(prop df [‘BEDS’]>=2) &
                            (prop_df [‘BATHS’]>1) &
                            (prop_df [‘HOA PER MONTH’] <=200) &
                            (prop_df [‘YEAR BUILT’]>=2000) &
                            (prop_df [‘SQUARE FEET’] > 2000) &
                            (prop_df [‘PRICE’] <=700000)]
>>> filtered_df.shape
(331, 23)
```

Шорт-лист сокращает число подходящих объектов недвижимости с 3624 до 331. При нанесении на карту эти объекты распределяются по городу [4-5].

Покупая дом, вы ищете близость к таким услугам, как магазины, аптеки, больницы и парки. Модуль геокодирования ArcGIS API для Python можно использовать для поиска таких объектов на заданном расстоянии вокруг дома, используя код 3:

```
restaurants = geocode('restaurant', search_extent=prop_buffer.extent, max_locations=200)
hospitals = geocode('hospital', search_extent=prop_buffer.extent, max_locations=50)
```

Еще одним важным фактором является время, необходимое для поездки на работу или в школу. Сетевой модуль ArcGIS API for Python предоставляет инструменты для расчета маршрутов и продолжительности поездки на основе исторической информации о трафике. Используя код 4 вычисляют направления между домом и офисом, а также время на дорогу, которое требуется обычным утром в понедельник в 8:00. Вы можете добавить

несколько остановок, которые вы делаете, как часть вашей поездки на работу. Эта информация может быть преобразована в DataFrame для Pandas и отображена в виде таблицы или гистограммы. Таким образом, дома можно сравнивать друг с другом по доступности к необходимым объектам. Код 4 имеет вид:

```
route_result = route_service.solve(stops_geocoded2, return_routes=True,
                                   return_stops=True, return_directions=True,
                                   impedance_attribute_name='TravelTime',
                                   start_time=644511600000,
                                   return_barriers=False, return_polygon_barriers=False,
                                   return_polyline_barriers=False)
```

Сравнение доступности проводится в пакетном режиме с каждым из 331 включенных в список объектов. Различные объекты соседства добавляются как новые столбцы в набор данных. Счетчик количества объектов (в пределах указанного расстояния) добавлен в качестве значения столбца. Если много объектов одного типа находятся рядом с собственностью, все они конкурируют за один и тот же рынок, снижая цены и улучшая качество обслуживания. Эти дома более привлекательны, чем остальные.

На основании гистограммы, многие из 331 домов находятся рядом с различными услугами. Благодаря этим шагам пространственного обогащения вы теперь можете рассматривать эти атрибуты на основе местоположения в дополнение к внутренним свойствам, таким как количество спальных комнат, санузлов и квадратным метрам [6-8].

Оценка домов - сугубо личный процесс. Разные покупатели оценивают разные характеристики домов. Не все аспекты рассматриваются одинаково, поэтому присвоение разных значений для объектов позволит вам получить

взвешенную сумму (балл) для каждого дома. Чем выше оценка, тем более желательным является дом для вас.

Код 5 создает систему оценки, которая отражает относительную важность каждой функции в доме:

```
def set_scores(row):
    score = ((row['PRICE']*-1.5) + # penalize by 1.5 times
            (row ['BEDS']*1)+
            (row ['BATHS']*1)+
            (row['SQUARE FEET']*1)+
            (row['LOT SIZE']*1)+
            (row['YEAR BUILT']*1)+
            (row['HOA PER MONTH']*-1)+ # penalize by 1 times
            (row['grocery_count']*1)+
            (row['restaurant_count']*1)+
            (row['hospitals_count']*1.5)+ # reward by 1.5 times
            (row ['coffee_count']*1)+
            (row['bars_count']*1)+
            (row['shops_count']*1)+
            (row['travel_count']*1.5)+ # reward by 1.5 times
            (row['parks_count']*1)+
            (row['edu_count']*1)+
            (row['commute_length']*-1)+ # penalize by 1 times
            (row['commute_duration']*-2)  # penalize by 2 times
    return score
```

Несмотря на то, что система оценки может быть чрезвычайно удобной при сравнении характеристик объектов, включенных в короткий список, при непосредственном применении (без какого-либо масштабирования) она

возвращает набор оценок, на которые сильно влияет небольшое количество атрибутов, имеющих численно большие значения. Например, такой атрибут, как цена недвижимости, имеет тенденцию быть большим числом (сотни тысяч) по сравнению с количеством спален (которое, вероятно, будет меньше 10). Без масштабирования цена недвижимости будет доминировать над оценкой, превышающей ее выделенное значение. Результаты, рассчитанные без масштабирования, кажутся чрезвычайно коррелированными с переменной ценой собственности. Хотя цена на недвижимость является важным фактором для большинства покупателей, она не может быть единственным критерием, который определяет ранг недвижимости.

Чтобы исправить это и вычислить новый набор баллов, все числовые столбцы масштабируются до единого диапазона 0-1, используя функцию MinMaxScaler из библиотеки scikit-learn. На рисунке 4 гистограмма и диаграмма справа показывают результаты масштабирования: оценки выглядят нормально распределенными, а разброс между ценой на недвижимость и оценками показывает только слабую корреляцию.

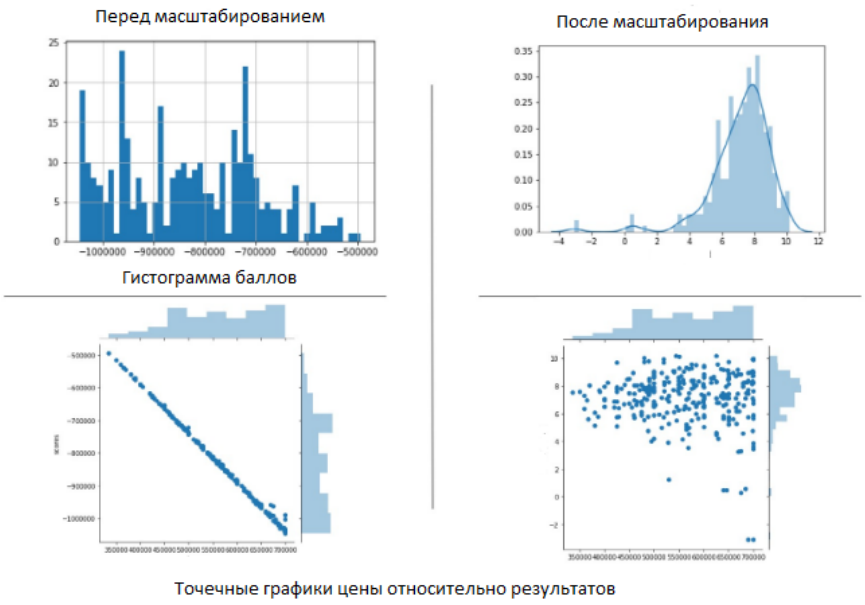

Рис. 4. Графики свойств до и после масштабирования

После того, как свойства были оценены, они сортируются в порядке убывания, создавая уточненный список домов, которые можно было бы посетить. В этом примере топ-50 домов распределены по всему городу без какой-либо сильной пространственной кластеризации. Цены на недвижимость, с другой стороны, появляются в кластерах.

Большинство домов в списке 50 лучших имеют два санузла, четыре спальных комнаты (даже если критерием отбора было минимум две спальни), имеют площадь менее 762 квадратных метров и были построены за последние четыре года. Большинство домов имеют хороший доступ к инфраструктуре и находятся в пределах 16 км от центра города, до них можно добраться за 25 минут.

Функция масштабирования обеспечила отсутствие единой функции, которая доминирует в оценке свойства, превышающей его выделенное значение. Тем не менее, могут быть некоторые особенности, которые имеют тенденцию коррелировать с оценками. Чтобы визуализировать это, функция pairplot () из библиотеки seaborn используется для создания диаграммы распределения каждой переменной друг против друга. (рис.5).

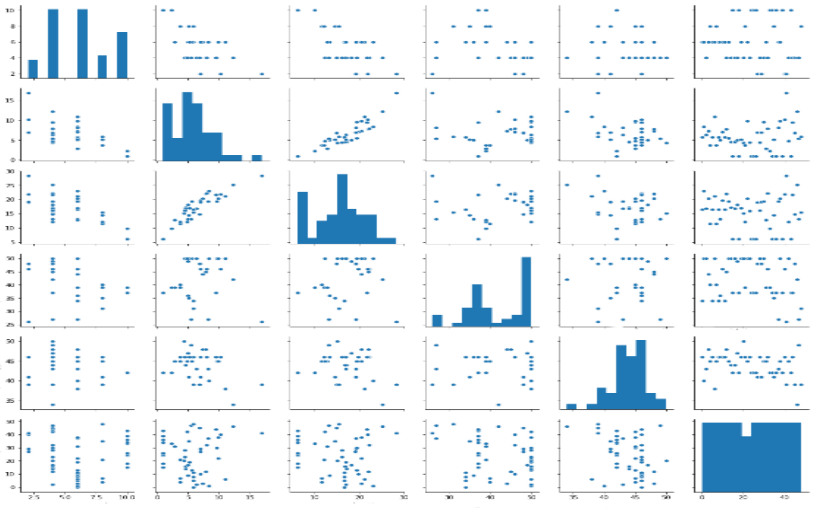

Рис. 5. Функция pairplot использовалась для создания этих диаграмм распределения.

Диагонали матрицы представляют собой гистограммы соответствующей переменной.

До настоящего времени набор данных был сконструирован с внутренними и пространственными атрибутами. Значения для различных функций были явно определены, чтобы можно было оценивать свойства. На самом деле процесс принятия решений для покупателей, хотя и логичен, но менее просчитан и немного размыт. Покупатели, скорее всего, будут довольны определенными недостатками, если они сильно впечатлены какой-то другой характеристикой. Если покупатели просто предпочитают одни дома и заносят в черный список другие, вы можете позволить модели машинного обучения определить их предпочтения.

Так как подобный тип обучающих данных трудно собрать для большого числа объектов, был синтезирован фиктивный набор данных с использованием 50 лучших домов в качестве любимой группы, а оставшиеся 281 - в черном списке. Эти данные были переданы в модель логистической регрессии машинного обучения.

Поскольку эта модель учится на данных обучения, она пытается присвоить значение каждой переменной-предиктору (внутренние и пространственные характеристики) и предсказать, будет ли покупатель предпочитать этот дом. Когда новая недвижимость появится на рынке, эта модель может предсказать, понравится ли она покупателю, и представить только релевантные результаты.

Код 6 показывает точность модели для этого набора данных. Точность относится к способности модели правильно определять, является ли данное свойство любимым или нет. F1-оценка вычисляет среднее значение гармоники точности и отзыва, чтобы получить комбинированную оценку точности модели. Код 6 имеет вид:

```
classification_report(y_test, test_predictions, target_names=['not fav','fav']))
('             precision    recall  f1-score
 '\n'
 '    not fav       0.94      0.98      0.96
 '        fav       0.88      0.71      0.79
 '\n'
 'avg / total       0.93      0.93      0.92
```

Данные обучения, использованные в этом тематическом исследовании, невелики по современным стандартам и не сбалансированы, потому что меньше объектов, которые являются фаворитами по сравнению с черными списками (50 против 281). Тем не менее, модель работает довольно хорошо с высокими показателями f1 для устранения объектов, которые, вероятно, будут в черном списке.

Значения, присвоенные регрессионной моделью, показаны в нижней части рисунка 6. На основе данных обучения модель имеет лишь незначительное отклонение в виде цены имущества и продолжительности поездки. Она имеет взвешенные характеристики, такие как количество магазинов, парков и учебных заведений отрицательно, а остальные - положительно. Такие функции, как количество больниц, кафе, баров и заправочных станций, получили большее значение, чем при выборе вручную.

Ручные значения

```
coeff = log_model.coef_.round(5).tolist()[0]
list(zip(X_train.columns, coeff))
```

```
[('PRICE', -0.4817),
 ('BEDS', 0.56799),
 ('BATHS', 0.65258),
 ('SQUARE FEET', 0.09618),
 ('LOT SIZE', -0.10108),
 ('YEAR BUILT', 0.86107),
 ('HOA PER MONTH', 0.02129),
 ('grocery_count', -0.7736),
 ('restaurant_count', -1.22493),
 ('hospitals_count', 1.38967),
 ('coffee_count', 1.27494),
 ('bars_count', 2.9728),
 ('gas_count', 1.16501),
 ('shops_count', -0.71489),
 ('travel_count', 0.24195),
 ('parks_count', -1.02031),
 ('edu_count', -0.45057),
 ('commute_length', -0.58029),
 ('commute_duration', -0.59949)]
```

Предполагаемые значения

```
def set_scores(row):
    score = ((row['PRICE']*-1.5) + # penalize by 1.5 times
             (row['BEDS']*1)+
             (row['BATHS']*1)+
             (row['SQUARE FEET']*1)+
             (row['LOT SIZE']*1)+
             (row['YEAR BUILT']*1)+
             (row['HOA PER MONTH']*-1)+  # penalize by 1 times
             (row['grocery_count']*1)+
             (row['restaurant_count']*1)+
             (row['hospitals_count']*1.5)+  # reward by 1.5 times
             (row['coffee_count']*1)+
             (row['bars_count']*1)+
             (row['shops_count']*1)+
             (row['travel_count']*1.5)+  # reward by 1.5 times
             (row['parks_count']*1)+
             (row['edu_count']*1)+
             (row['commute_length']*-1)+  # penalize by 1 times
             (row['commute_duration']*-2)  # penalize by 2 times
            )
    return score
```

Рис. 6. Сравнение значений, выбранных вручную и проанализированных регрессионной моделью.

Таким образом, тип механизма рекомендаций, созданный в этом исследовании, называется контентной фильтрацией, поскольку он использует только внутренние и пространственные характеристики, разработанные для прогнозирования. Для рекомендаций такого типа необходим обучающий набор, который был бы слишком большим для создания вручную.

На практике используется другой тип фильтрации по рекомендациям на основе сообщества. Он использует функции, разработанные для свойств, в сочетании с данными избранного и черного списка, чтобы найти сходство между большим количеством покупателей. Затем он объединяет учебный комплект от аналогичных покупателей для создания большого учебного набора.

В этом тематическом исследовании входной набор данных был пространственно обогащен информацией о доступе к различным средствам. Это может быть расширено за счет разработки социально-экономических функций, таких как возраст, доход, уровень образования и множество других параметров, с помощью модуля геообогащения ArcGIS API для Python. Авторитетные данные, которыми обмениваются местные органы власти в рамках инициативы открытых данных, также могут быть включены на местном уровне [9-10].

Эта статья демонстрирует, как может быть применен анализ данных и машинное обучение. Хотя покупка дома - это личный процесс, многие решения сильно зависят от местоположения. Библиотеки Python, такие как Pandas, могут использоваться для визуализации и статистического анализа.

Литература

1. Тальников Д.М., Степанова М.Р., Ажиба М.О., Сеферян Л.А. Применение BIM-технологий в оценке недвижимости. EBIM // Инженерный вестник Дона, 2019, №3. URL: ivdon.ru/ru/magazine/archive/n3y2019/5793

2. Заставной Д.А. Представление атрибутивной информации в ГИС WinMap и язык WMSL**.** // Инженерный вестник Дона, 2011, №1. URL: ivdon.ru/ru/magazine/archive/n1y2011/337

3. Серая Е.С., Шеина С.Г., Петров К.С., Матвейко Р.Б. Интеллектуальная городская среда. Интеграция ГИС и BIM // Инженерный вестник Дона, 2019, №1. URL: ivdon.ru/ru/magazine/archive/n1y2019/5495

4. Hollands Robert G. Will the real smart city please stand up? // City. 2008. V. 12. №3. pp. 303-320.

5. Петров К.С., Швец Ю.С., Корнилов Б.Д. и др. Применение BIM технологий при проектировании и реконструкции зданий и сооружений //

Инженерный вестник Дона, 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5255.

6. Amir H. Razavi. Arcview Gis //Avenue Developer's Guide. - OnWord Press. 1999. – 452 p.

7. Deakin M., Al Waer H. From Intelligent to Smart Cities // Intelligent Buildings International. 2011. V. 3. № 3. pp. 133-139.

8. Musa S. Smart Cities - A Roadmap for Development // J Telecommun Syst Manage. 2016. V. 5. № 3. pp. 144-146.

9. Петров К.С., Кузьмина В.А., Федорова К.В. Проблемы внедрения программных комплексов на основе технологий информационного моделирования (BIM-технологии) // Инженерный вестник Дона, 2017, №2 URL: ivdon.ru/ru/magazine/archive/N2y2017/4057.

10. Аленичева Е.В. Методы оценки объектов недвижимости. Тамбов: ТГТУ, 2005, 25 с.

References

1. Tal'nikov D.M., Stepanova M.R., Azhiba M.O., Seferyan L.A. Inzhenernyj vestnik Dona 2019, №3. URL: ivdon.ru/ru/magazine/archive/N2y2017/4114

2. Zastavnoj D.A. Inzhenernyj vestnik Dona, 2011, №1. URL: ivdon.ru/ru/magazine/archive/n1y2011/337

3. Seraya E.S., SHeina S.G., Petrov K.S., Matvejko R.B. Inzhenernyj vestnik Dona, 2019, №1. URL: ivdon.ru/ru/magazine/archive/n1y2019/5495

4. Ballon P., Glidden J., Kranas P. and others. eChallenges e-2011 Conference Proceedings. 2011. DOI: 10.13140 / 2.1.5062.4965

5. Petrov K.S., SHvets YU.S., Kornilov B.D. i dr. Inzhenernyj vestnik Dona, 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5255.

6. Amir H. Razavi. OnWord Press. 1999. 452 p.

7. Deakin M., Al Waer H. Intelligent Buildings International. 2011. Vol. 3. № 3. pp. 133-139.

8. Musa S. Smart Cities - A Roadmap for Development J Telecommun Syst Manage. 2016. Vol. 5. №3. pp. 144-146.

9. Petrov K.S., Kuz'mina V.A., Fedorova K.V. Inzhenernyj vestnik Dona, 2017, №2. URL: ivdon.ru/ru/magazine/archive/N2y2017/4057

10. Alenicheva E.V. Metody otsenki ob"yektov nedvizhimosti. [Methods of property evaluation]. Tambov: TGTU, 2005, 25 p.