МЕТОДЫ МНОЖЕСТВЕННОЙ РЕГРЕССИИ В ПРОГНОЗИРОВАНИИ ЦЕН НА ЖИЛЫЕ ПОМЕЩЕНИЯ

ЛЕВИЧЕВА Надежда Борисовна
магистрант
ФГБОУ ВО «Ростовский государственный экономический университет (РИНХ)»
г. Ростов-на-Дону, Россия

В статье рассматриваются методы прогнозирования цен на жилую недвижимость с помощью машинного обучения на примере решения задачи множественной регрессии с помощью Python. В исследовании используются такие методы, как линейная регрессия, лес принятия решений, метод опорных векторов с линейным ядром, метод k-ближайших соседей.
Ключевые слова: множественная регрессия, метод k-ближайших соседей, ценообразование, машинное обучение.

Каждый экономический показатель всегда формируется под влиянием множества различных категорий и существующих фактов. Так, для определения уровня спроса на определенный товар/услугу необходимо выявить кроме цены также и другие показатели, такие как покупательская способность, цены на подобные и дополнительные товары/услуги и так далее. В такой ситуации стоит применять множественную регрессию:

$$\hat{y} = f\left(x_1, x_2, ..., x_p\right)$$

Множественную регрессию можно применять для большого спектра задач: определения макроэкономических показателей, спроса, прибыльности вложений, оптимизации расходов и решения других вопросов [2, с. 24]. Множественная регрессия предназначена прежде всего для получения многофакторной мо-

дели с выявлением доли влияния каждого конкретного фактора и при этом степени их общего влияния на искомый показатель. Так, с учетом того, что в формировании прогноза цен на жилье влияние оказывают многие факторы (дата, класс недвижимости, территориальный кластер, место нахождения объекта внутри/за МКАД, доступность метро/МЦД пешком и на автомобиле), задачу ценообразования в данном сегменте рынка можно решить с помощью множественной регрессии.

В случае, когда зависимая переменная имеет взаимосвязь с независимой, парный регрессионный анализ может быть заменен множественным. Это необходимо по причине увеличения объема данных, которые должны быть задействованы в регрессионной модели. Здесь возникает несколько вопросов, требующих решения, основные из которых перечислены ниже:

1. Нужно определить, каким образом оказывает влияние независимая переменная на зависимую. При этом есть потребность в том, чтобы идентифицировать, в каких случаях имеет место результат её влияния, а в каких – других переменных.

2. Для выбора характеристик модели следует принять решение о том, какие факторы нужно оставить для регрессии. Вероятно, что не все факторы будут полезны для проведения регрессионного анализа и решения поставленной задачи.

Чтобы спрогнозировать цены на жилые помещения и транспортной доступности используем следующие методы решения задачи множественной регрессии с помощью Python:

1. Линейная регрессия – метод «LinearRegression»;

2. Лес принятия решений (случайный лес) – метод «RandomForestRegressor»;

3. Метод опорных векторов с линейным ядром – метод «SVR»;

4. Метод k-ближайших соседей – метод «KneighborsRegressor».

В целях сравнения эффективности вышеперечисленных методов для разных признаков из имеющегося набора данных рассмотрим их на примере прогнозирования цен на квартиры и доступности метро/МЦД.

Для разработки и обучения моделей прогнозные значения отделены от выборки:

trg = df_MinMax[['Цена, руб./кв.м.', 'Доступность метро/МЦД, минут пешком']]

trn = df_MinMax.drop(['Цена, руб./кв.м.', 'Доступность метро/МЦД, минут пешком'], axis=1)

Далее все модели были помещены в один список для удобства дальнейшего анализа (рисунок 1):

```
models = [LinearRegression(), # линейная регрессия
          RandomForestRegressor(n_estimators=100, max_features ='sqrt'), # случайный лес
          KNeighborsRegressor(n_neighbors=10), # метод ближайших соседей
          SVR(kernel='linear'), # метод опорных векторов с линейным ядром
          ]
```

Рисунок 1. **Формирование списка моделей для прогнозирования цен**

Для целей оценки качества моделей при обучении выделили 30 процентов всего объема данных, при этом на оставшейся части проводилось оценка качества моделей после их обучения. Выбор такого процентного соотношения обусловлен размерами выборки и установлен как самый эффективный по результатам тестирования в различных пропорциях. После этого по каждому используемому параметру (цены на жилые помещения и доступность метро/МЦД пешком) сформированы регрессионные модели следующим образом:

from sklearn.model_selection import KFold, cross_val_score, train_test_split

Xtrn, Xtest, Ytrn, Ytest = train_test_split(trn, trg, test_size=0.3)

Прогнозируем 2 параметра ('Цена, руб./кв.м.', 'Доступность метро/МЦД, минут пешком') и строим регрессию для каждого из них. Для дальнейшего ана-

лиза можно записать полученные результаты во временный DataFrame. Сделать это можно так (рисунок 2):

```
#создаем временные структуры
from sklearn.metrics import r2_score
TestModels = pd.DataFrame()
tmp = {}
#для каждой модели из списка
for model in models:
    #получаем имя модели
    m = str(model)
    tmp['Model'] = m[:m.index('(')]
    #для каждого столбца результирующего набора
    for i in range(Ytrn.shape[1]):
        #обучаем модель
        model.fit(Xtrn, Ytrn.iloc[:,i].values)
        #вычисляем коэффициент детерминации
        tmp['R2_Y%s'%str(i+1)] = r2_score(Ytest.iloc[:,0].values, model.predict(Xtest))
    #записываем данные и итоговый DataFrame
    TestModels = TestModels.append([tmp])
#делаем индекс по названию модели
TestModels.set_index('Model', inplace=True)
```

Рисунок 2. **Построение моделей для прогнозирования цен в Python**

Эффективность моделей определена путем вычисления коэффициентов детерминации, позволяющих оценить степень соответствия прогнозируемых моделями данных тем, на которых было проведено обучение. Этот критерий оценки считается наиболее эффективным по отношению к моделям, решающим задачу линейной регрессии. Коэффициенты детерминации помогли выявить присущую определяющему признаку степень вариации при воздействии на факторный признак. Так, если функциональная взаимосвязь существует, то коэффициент детерминации – единица, а если отсутствует – ноль.

Построим графики и посмотрим какая модель показала лучший результат (рисунок 3):

```
fig, axes = plt.subplots(ncols=2, figsize=(10,4))
TestModels.R2_Y1.plot(ax=axes[0], kind='bar', color='orange', title='R2_Цена, руб./кв.м.')
TestModels.R2_Y2.plot(ax=axes[1], kind='bar', color='green', title='Доступность метро/МЦД, минут пешком')
```

Рисунок 3. **Построение графиков для визуальной оценки результатов моделей**

Полученные результаты моделей показаны на рисунке 4:

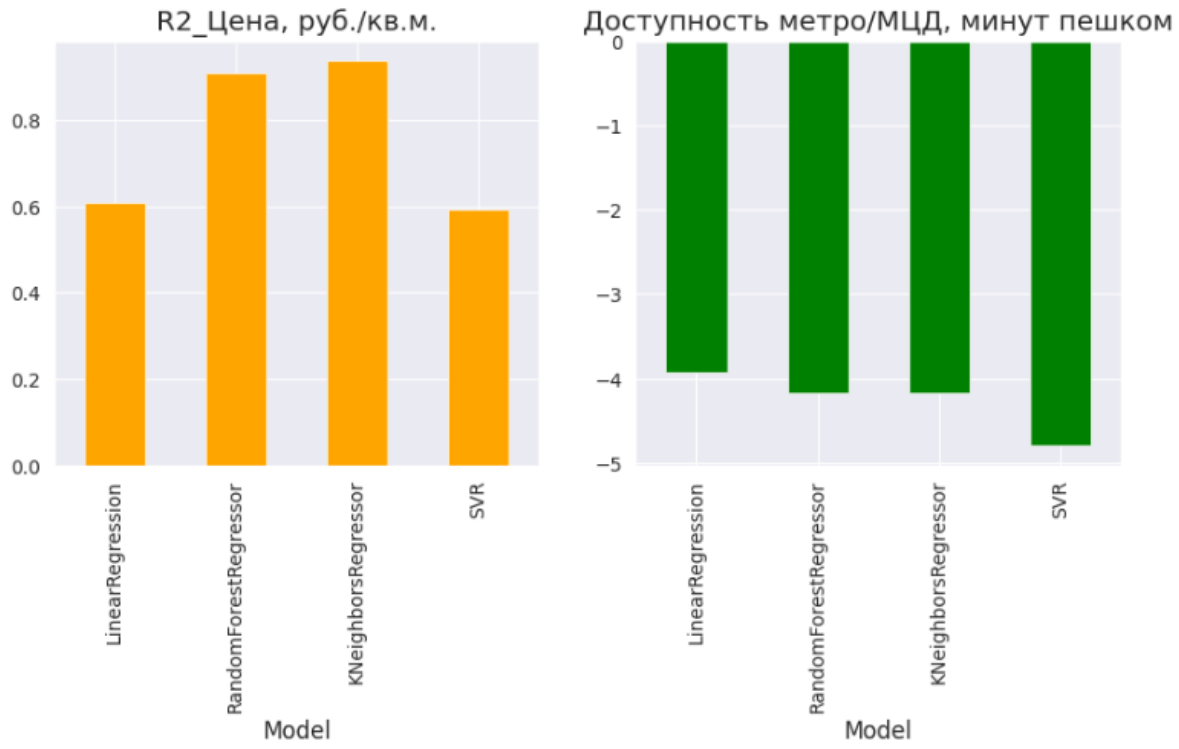

Рисунок 4. **Результаты моделей для решения задач множественной регрессии**

Из графиков видно, что для прогнозирования цен на жилые помещения наиболее приемлемые результаты получены с помощью моделей k-ближайших соседей и случайного леса принятия решений. При этом хуже всего показывают себя методы опорных векторов с линейным ядром и линейная регрессия.

Для прогнозирования доступности метро/МЦД наиболее точный прогноз предоставляет метод линейной регрессии. Его нельзя назвать эффективным, т.к. показатели точности данной модели принимают отрицательные значения. В этом случае необходимо продолжить поиск другого метода прогнозирования доступности до метро/МЦД.

Для понимания, что из себя представляет показавший наиболее приемлемые результаты метод k-ближайших соседей, обратимся к основам его функционирования. Идея состоит в том, что похожие значения целевой переменной

могут быть получены от объектов, обладающих схожими признаками, например, находящимися в одном классе. Порядок, в котором модель выполняет свою работу, следующий:

1) производится вычисление расстояния между тренировочным и обучающими объектами выборки;

2) на следующем этапе выбирают k-ближайших объектов к тестовому, при этом количество k определяется изначально;

3) окончательным результатом из числа отобранных k-объектов при регрессии станет среднее арифметическое значение, а в случае задачи классификации – мода;

4) все вышеперечисленные этапы будут пройдены для каждого тренировочного объекта.

Задачи регрессии с помощью метода k-ближайших соседей можно решать в том случае, когда имеются непрерывные данные. Так, путем нахождения среднего значения ближайших соседей определяют искомую метку объекта.

В библиотеке Scikit-learn существует регрессор «KNeighborsRegressor», который обучает на основе метода k-ближайших соседей все точки запроса. Лицо, осуществляющее данный запрос, должно ввести целочисленное значение k. При формировании модели на примере ценообразования наилучшее решение было получено с указанием k = 10.

При использовании регрессии на основе метода ближайших соседей используют похожие веса: каждая точка в локальном пространстве приносит подобный вклад в классификацию точки запроса. В некоторых случаях может быть полезным взвешивать точки таким образом, чтобы рядом находящиеся точки влияли на регрессию в большей степени, нежели дальние точки. Добиться такого результата можно применяя слово «weights». Присваивание всем точках одинаковых весов происходит с помощью значения по умолчанию weights = «uniform». Веса, обратно пропорциональные промежутку от точки запроса, можно установить так: weights = «distance» [1]. Таким образом, для определе-

ния весов возможно самостоятельно задавать функцию дистанции в зависимости от конкретной ситуации и условий.

Так как лучше всего себя показала модель на основе метода k-ближайших соседей, была проведена оценка её эффективности путём вычисления cross_val_score, т.к. это наиболее легкий способ применять перекрестную проверку:

```
from sklearn.model_selection import cross_val_score
from sklearn import svm
scores = cross_val_score(model, Xtrn, Ytrn, cv=10)
```

Для любой задачи регрессии имеется датасет, который применяется при обучении моделей, а также неизвестные данные, которые позволяют тренировать полученные модели. Для проверки эффективности модели используют перекрестную проверку. Она позволяет определить точность предсказаний, полученных на неизвестных ранее этой модели данных, выявить проблемы (например, переобучение модели). С помощью перекрестной проверки также реализуют повторную выборку и разделение выборки, что помогает обучать модель на разных временных промежутках и задействовать разные группы выборки для тренировки модели. Такую проверку применяют тогда, когда требуется оценка практических получаемых моделью прогнозов. На данном примере прогнозирования цен на жилые помещения получаем точность модели k-ближайших соседей: 0,97.

В целях поиска более точной модели прогнозирования цен на недвижимость необходимо протестировать все известные методы машинного обучения. При этом среди методов множественной регрессии наиболее эффективным является метод k-ближайших соседей.

СПИСОК ЛИТЕРАТУРЫ

1. Ближайшие соседи [Электронный ресурс] – URL: https://scikit-learn.ru/1-6-nearest-neighbors/ (Дата обращения: 24.06.2024).

2. Джалилов Ш.А. Метод расчета параметров множественной линейной регрессии // Достижения науки и образования. – 2020. - № 3 (57). – С. 24–28.
3. Жидченко Т.В., Середина М.Н., Серёгина В.В., Удинцова Н.М. Методы множественной регрессии и корреляции в экономических исследованиях // Методология развития управления, экономики и образования. – Пенза: Изд-во: АНО «Приволжский Дом знаний», 2023. – С. 130–153.
4. Кузнецов И.Н. Пример решения задачи множественной регрессии с помощью Python [Электронный ресурс]: – URL: https://habr.com/ru/post/206306/ (дата обращения: 24.06.2024).

MULTIPLE REGRESSION METHODS IN FORECASTING RESIDENTIAL PRICES

LEVICHEVA Nadezhda Borisovna
Graduate student
Rostov State University of Economics (RINH)
Rostov-on-Don, Russia

The article discusses methods for predicting residential real estate prices using machine learning on the example of solving the problem of multiple regression using Python. The research uses such methods as linear regression, decision forest, the method of support vectors with a linear kernel, and the k-nearest neighbor method.
Keywords: multiple regression, k-nearest neighbor method, pricing, machine learning.