УДК 004

Применение методов машинного обучения для построения модели прогнозирования динамики цен на рынке автомобилей в среде Google Colaboratory

Андриенко Иван Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается применение методов машинного обучения для построения модели прогнозирования динамики цен на рынке автомобилей с использованием среды Google Colaboratory. Описан процесс сбора, предобработки данных и создания различных моделей машинного обучения, включая линейную регрессию, случайный лес, дерево решений, k-ближайших соседей и XGBoost. Особое внимание уделено анализу данных и оценке эффективности моделей с использованием метрик качества, таких как среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE) и коэффициент детерминации (R2). Приводится сравнительный анализ производительности моделей, показывающий, что модель случайного леса демонстрирует наилучшую точность прогнозирования.
Ключевые слова: Машинное обучение, прогнозирование цен, рынок автомобилей, Google Colaboratory, регрессионный анализ, линейная регрессия, случайный лес, дерево решений, k-ближайших соседей, XGBoost

Application of machine learning methods to build a model for predicting price dynamics in the car market in the Google Colaboratory environment

Andrienko Ivan Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses the application of machine learning methods to build a model for predicting price dynamics in the car market using the Google Colaboratory environment. The process of collecting, preprocessing data and creating various machine learning models, including linear regression, random forest, decision tree, k-nearest neighbors and XGBoost, is described. Special attention is paid to data analysis and evaluation of the effectiveness of models using quality metrics such as mean square error (MSE), mean absolute error (MAE) and coefficient of determination (R2). A comparative analysis of model performance is provided, showing that the random forest model demonstrates the best prediction accuracy.

Keywords: Machine learning, price forecasting, car market, Google Coollaboratory, regression analysis, linear regression, random forest, decision tree, k-nearest neighbors, XGBoost

1 Введение

1.1 Актуальность

Сегодня разработка систем машинного обучения занимает одно из центральных мест в области информационных технологий, математического анализа и статистики. Эти системы все более активно внедряются в нашу повседневную жизнь через продукты, основанные на методах искусственного интеллекта. Можно уверенно утверждать, что такие технологии будут и дальше развиваться, постепенно становясь неотъемлемой частью различных профессиональных сфер. Создание систем машинного обучения требует значительных временных и интеллектуальных затрат от высококвалифицированных специалистов как в области искусственного интеллекта, так и в конкретных предметных областях их применения. Примером использования машинного обучения является прогнозирование цен.

1.2 Обзор исследований

В статье К.А. Найденова, О.А. Невзорова рассматриваются проблемы применения машинного обучения для обработки естественного языка. Перечислены основные методы и технологии, такие как глубокие нейронные сети и методы обучения на больших данных, их применение для анализа и синтеза текстов. Оцениваются перспективы и вызовы в этой области. Особое внимание уделено методам повышения точности и эффективности обработки текстовой информации [1]. В статье М.В. Боброва, А.Е. Мастилин исследуются применения методов машинного обучения в кибербезопасности. Рассматриваются различные алгоритмы, такие как классификация аномалий и предсказание угроз, а также их применение в реальных системах безопасности. Описаны перспективы использования машинного обучения для повышения эффективности и точности обнаружения кибератак [2]. В работе М.В. Коротеева рассматриваются современные тенденции в технологии машинного обучения. Выполнен анализ новейших методологий, таких как использование предобученных моделей, мультизадачных систем и нейроэволюции. Особое внимание уделено автоматизированному машинному обучению, что позволяет значительно сократить долю человеческого участия в создании систем искусственного интеллекта [3]. Статья Н.Г. Полетаева предлагает классификацию различных систем машинного обучения и анализ их областей применения. Рассматриваются преимущества и недостатки различных алгоритмов и моделей, а также их потенциальное влияние на различные сферы деятельности. Подчеркнута важность выбора правильной методологии для конкретных задач [4]. В исследовании В.Е. Дементьев, С.Х. Киреев рассмотрели применение методов машинного обучения для оценки стоимости недвижимости. Описаны

методики сбора и обработки данных, построение регрессионных моделей и их валидация на реальных данных. Особое внимание уделено точности предсказаний и применению моделей в практике [5].

1.3 Цель исследования

Целью данного исследования является построения моделей прогнозирования цены на рынке автомобилей на основе регрессионного анализа.

2 Материалы и методы

Для решения данной цели подойдет бесплатная облачная среда Google Colaboratory, разработанная для написания программного кода с использованием различных библиотек машинного обучения.

3 Результаты и обсуждения

Для исследования были взяты данные машин. В датасете содержаться легковые машины разного года выпуска и разных комплектаций. Все данные занесены в таблицу MS Excel (рис. 1).

	A	B	C	D	E	F	G	H	I	J
1	name	model	transmission	color	Year of release	engine_type	Dvigatel	the body	cena	log10
2	Subaru	Outback	automatic	silver	2010	gasoline	2.5	universal	1800000	6,25527251
3	Subaru	Outback	automatic	blue	2002	gasoline	3.0	universal	450000	5,65321251
4	Subaru	Forester	automatic	red	2001	gasoline	2.0	suv	760000	5,88081359
5	LADA	Vesta	automatic	grey	2017	gasoline	1.6	sedan	930000	5,96848295
6	LADA	Largus	mechanical	silver	2014	gasoline	1.6	universal	720000	5,8573325
7	LADA	Vesta	mechanical	brown	2018	gasoline	1.6	universal	1150000	6,06069784
8	LADA	Largus	mechanical	black	2017	gasoline	1.6	universal	975000	5,98900462
9	УАЗ	452	mechanical	green	1978	gasoline	2.5	pickup	385000	5,58546073
10	УАЗ	Patriot	mechanical	black	2014	gasoline	2.7	suv	700000	5,84509804
11	Kia	Cerato	mechanical	black	2009	gasoline	1.6	sedan	700000	5,84509804
12	Kia	Cee'd	mechanical	black	2007	diesel	1.6	hatchback	877000	5,94299959
13	Kia	Sportage	automatic	black	2012	gasoline	2.0	suv	1450000	6,161368
14	Kia	Optima	automatic	black	2015	gasoline	2.4	sedan	1900000	6,2787536
15	Kia	Clarus	mechanical	blue	1999	gasoline	1.8	universal	104400	5,0187005
16	Kia	Rio	automatic	black	2011	gasoline	1.4	sedan	785000	5,89486966
17	Kia	Picanto	mechanical	blue	2005	gasoline	1.1	hatchback	575000	5,75966785
18	Opel	Corsa	mechanical	black	2009	diesel	1.3	hatchback	599000	5,77742682
19	Opel	Insignia	mechanical	black	2008	gasoline	1.3	liftback	429000	5,63245729
20	Opel	Omega	mechanical	brown	2000	gasoline	2.5	sedan	430000	5,63346846
21	Opel	Zafira	mechanical	grey	2001	diesel	2.0	minivan	544000	5,7355989
22	Opel	Vectra	mechanical	blue	1997	gasoline	1.6	universal	145000	5,161368
23	Москвич	408	mechanical	red	1972	gasoline	1.4	sedan	280000	5,44715803
24	Москвич	412	mechanical	white	1977	gasoline	1.5	sedan	469000	5,67117284
25	Москвич	412	mechanical	green	1984	gasoline	1.5	sedan	150000	5,17609126
26	Москвич	2140	mechanical	red	1986	gasoline	1.5	sedan	80000	4,90308999
27	Izh	2125	mechanical	blue	1988	gasoline	1.5	sedan	2500000	6,39794001
28	Alfa Romeo	GTV	mechanical	black	1998	gasoline	2.0	sedan	950000	5,97772361
29	Acura	MDX	automatic	grey	2014	gasoline	3.5	suv	2450000	6,38916608
30	Acura	MDX	automatic	black	2014	gasoline	3.5	suv	3400000	6,53147892
31	Acura	MDX	automatic	silver	2004	gasoline	3.5	suv	600000	5,77815125
32	Volkswagen	Golf	automatic	silver	2013	diesel	1.4	universal	1500000	6,17609126
33	Volkswagen	Polo	automatic	blue	2015	gasoline	1.6	sedan	1450000	6,161368
34	Volkswagen	Passat	mechanical	green	2007	gasoline	2.0	sedan	980000	5,99122608

Рис. 1. Фрагмент таблицы MS Excel с данными.

Используем облачную среду Google Colaboratory для создания регрессионной модели.

В первую очередь делаем импорт необходимых библиотек для загрузки библиотек и модулей, которые предоставляют инструменты для работы с данными (рис. 2). Код на рисунке отвечает за импорт необходимых библиотек и модулей, построения модели линейной регрессии и оценки ее качества. Библиотека pandas используется для работы с табличными

данными, numpy - для выполнения численных вычислений, а scikit-learn предоставляет инструменты для машинного обучения и оценки моделей. Импортированные функции и метрики оценки моделей используются для разделения данных на обучающий и тестовый наборы, создания модели линейной регрессии и вычисления метрик ее качества, таких как среднеквадратичная ошибка, средняя абсолютная ошибка и коэффициент детерминации. Библиотека matplotlib используется для визуализации данных.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
```

Рис. 2. Импорт необходимых библиотек

Далее выполняем загрузку данных из датасета carr.xlsx с использованием библиотеки pandas (рис.3). Функция используется для чтения данных из Excel-файла в формате DataFrame. После загрузки данных происходит проверка корректности загрузки и вывод первых нескольких строк данных с помощью метода. Это позволяет быстро ознакомиться с структурой данных и убедиться, что они были успешно загружены.

```python
# Загрузка данных из файла carr.xlsx
data = pd.read_excel('carr.xlsx')

# Проверка загрузки и вывод первых нескольких строк данных
print(data.head())
```

```
     name     model transmission   color  Year of release engine_type  \
0  Subaru   Outback    automatic  silver             2010    gasoline
1  Subaru   Outback    automatic    blue             2002    gasoline
2  Subaru  Forester    automatic     red             2001    gasoline
3   LADA     Vesta    automatic    grey             2017    gasoline
4   LADA    Largus   mechanical  silver             2014    gasoline

   Dvigatel   the body     cena     log10
0       2.5  universal  1800000  6.255273
1       3.0  universal   450000  5.653213
2       2.0        suv   760000  5.880814
3       1.6      sedan   930000  5.968483
4       1.6  universal   720000  5.857332
```

Рис. 3. Загрузка данных из датасета

Данный датасет представляет собой набор данных о различных автомобилях. В наборе содержится информация о следующих столбцах:

1. name: название марки автомобиля.

2. model: модель автомобиля.

3. transmission: тип коробки передач (например, автоматическая или механическая).

4. color: цвет автомобиля.

5. Year of release: год выпуска автомобиля.

6. engine_type: тип двигателя (например, бензиновый).

7. Dvigatel: объем двигателя.

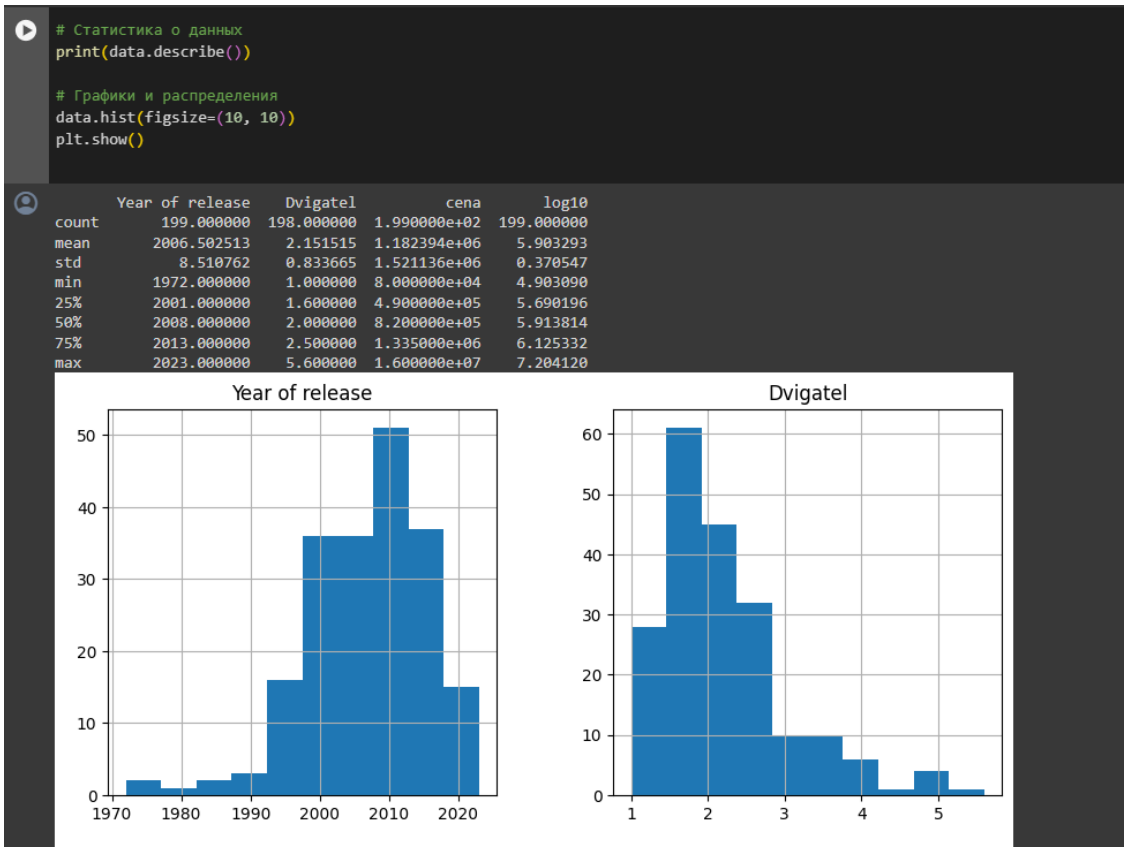
8. the body: тип кузова автомобиля.

9. cena: цена автомобиля.

10. log10: логарифм цены по основанию 10.

Этот датасет содержит информацию о характеристиках автомобилей, таких как модель, тип коробки передач, цвет, год выпуска, тип двигателя, объем двигателя, тип кузова и цена. Кроме того, в датасете присутствует столбец log10, который представляет логарифм цены по основанию 10.

Затем выполняем анализ данных. Сначала вычисляются основные статистические показатели для каждого числового столбца с помощью метода, который выводит среднее значение, стандартное отклонение, минимальное и максимальное значения, квартили и т. д. Затем создаются гистограммы для каждого числового столбца. Параметр «figsize» определяет размеры создаваемой фигуры. После этого вызывается функция для отображения гистограмм. Это помогает понять распределение данных и выявить возможные аномалии или особенности в данных.



	Year of release	Dvigatel	cena	log10
count	199.000000	198.000000	1.990000e+02	199.000000
mean	2006.502513	2.151515	1.182394e+06	5.903293
std	8.510762	0.833665	1.521136e+06	0.370547
min	1972.000000	1.000000	8.000000e+04	4.903090
25%	2001.000000	1.600000	4.900000e+05	5.690196
50%	2008.000000	2.000000	8.200000e+05	5.913814
75%	2013.000000	2.500000	1.335000e+06	6.125332
max	2023.000000	5.600000	1.600000e+07	7.204120

Рис. 4. Анализ данных

Необходимо выполнить предобработку данных для моделирования (рис. 5). Код кодирует категориальные переменные, выбирает признаки и целевую переменную, а затем логарифмирует целевую переменную для улучшения ее распределения, что может быть полезно для моделей, таких как линейная регрессия.

```
# Кодирование категориальных переменных
data_encoded = pd.get_dummies(data)

# Выбор признаков и целевой переменной
X = data_encoded[['Year of release']]
y = data_encoded['cena']

# Логарифмирование целевой переменной (для линейной регрессии)
y_log = np.log10(y)
```

Рис.5. Предобработка данных для моделирования

Выполним разделение данных на обучающий и тестовый наборы. Параметр test_size = 0.2 указывает, что 20% данных будут использоваться для тестирования, а оставшиеся 80% - для обучения модели. Создаем модель линейной регрессии и обучаем ее на обучающих данных. После выполнения этого кода, модель будет готова для предсказаний на новых данных (рис. 6).

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y_log, test_size=0.2, random_state=42)

# Создание и обучение модели
model = LinearRegression()
model.fit(X_train, y_train)

▾ LinearRegression
LinearRegression()
```

Рис. 6. Обучение модели

Выполним предсказание целевой переменной на тестовом наборе данных с использованием обученной модели. После получения прогнозов, вычисляются различные метрики оценки качества модели, такие как среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE), корень из среднеквадратичной ошибки (RMSE) и коэффициент детерминации (R2 score). Затем значения этих метрик выводятся на экран. Эти метрики позволяют оценить точность и эффективность модели на тестовых данных (рис. 7).

```
# Предсказание на тестовом наборе
y_pred = model.predict(X_test)

# Метрики оценки модели
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("MSE:", mse)
print("MAE:", mae)
print("RMSE:", rmse)
print("R2 score:", r2)
```

```
MSE: 0.04194030378333637
MAE: 0.15757435159818203
RMSE: 0.20479331967458406
R2 score: 0.5820495982226115
```

Рис. 7. Вывод метрик

Выполним предсказания с двумя фичами. В данном случае создадим новый датафрейм, который содержит только два выбранных признака: год выпуска автомобиля и тип двигателя. Выполняем предсказание целевой переменной на тестовом наборе данных с использованием модели, построенной на двух выбранных признаках. Затем вычислим различные метрики оценки качества модели с использованием только этих двух признаков (рис. 8).

```
# Предсказание на тестовом наборе для модели с двумя фичами
y_pred2 = model2.predict(X_test2)

# Метрики оценки модели с двумя фичами
mse2 = mean_squared_error(y_test2, y_pred2)
mae2 = mean_absolute_error(y_test2, y_pred2)
rmse2 = np.sqrt(mse2)
r2_2 = r2_score(y_test2, y_pred2)

print("MSE (две фичи):", mse2)
print("MAE (две фичи):", mae2)
print("RMSE (две фичи):", rmse2)
print("R2 score (две фичи):", r2_2)
```

```
MSE (две фичи): 0.04292756162597193
MAE (две фичи): 0.15960935065647935
RMSE (две фичи): 0.20718967548112027
R2 score (две фичи): 0.5722112142633766
```

Рис. 8. Предсказание с двумя фичами

Попробуем использовать для прогнозирования и другие модели. Создадим модель случайного леса для решения задачи. Импортируем соответствующий класс. Затем создаем экземпляр модели с заданным параметром, который обеспечивает воспроизводимость результатов. Далее обучаем модель и выводим метрики (рис. 9).

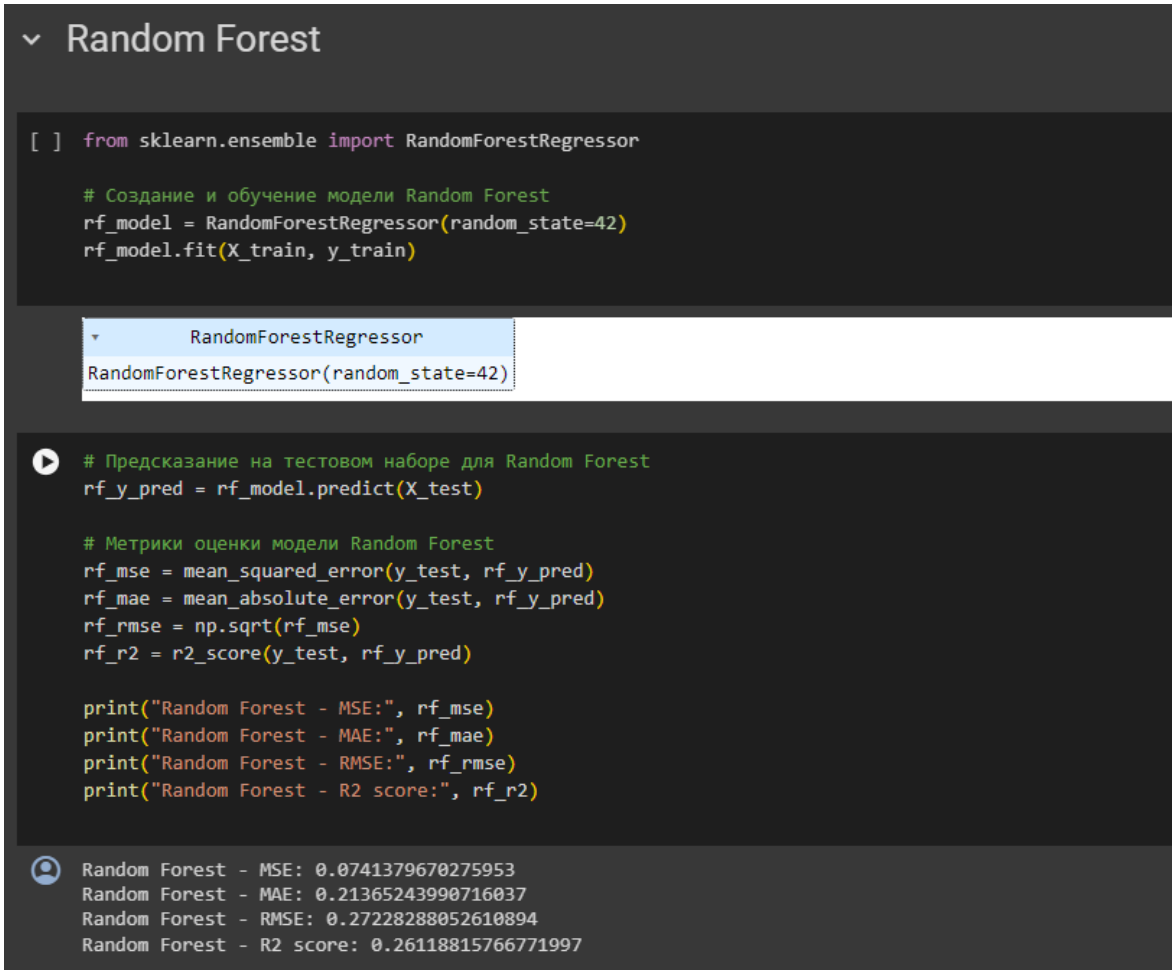


Рис. 9. Использование модели случайного леса

Далее код создает и обучает модель дерева решений для задачи регрессии. Затем выполняется предсказание целевой переменной на тестовом наборе данных, а затем вычисляются и выводятся метрики оценки качества модели, такие как MSE, MAE, RMSE и R2 (рис. 10).

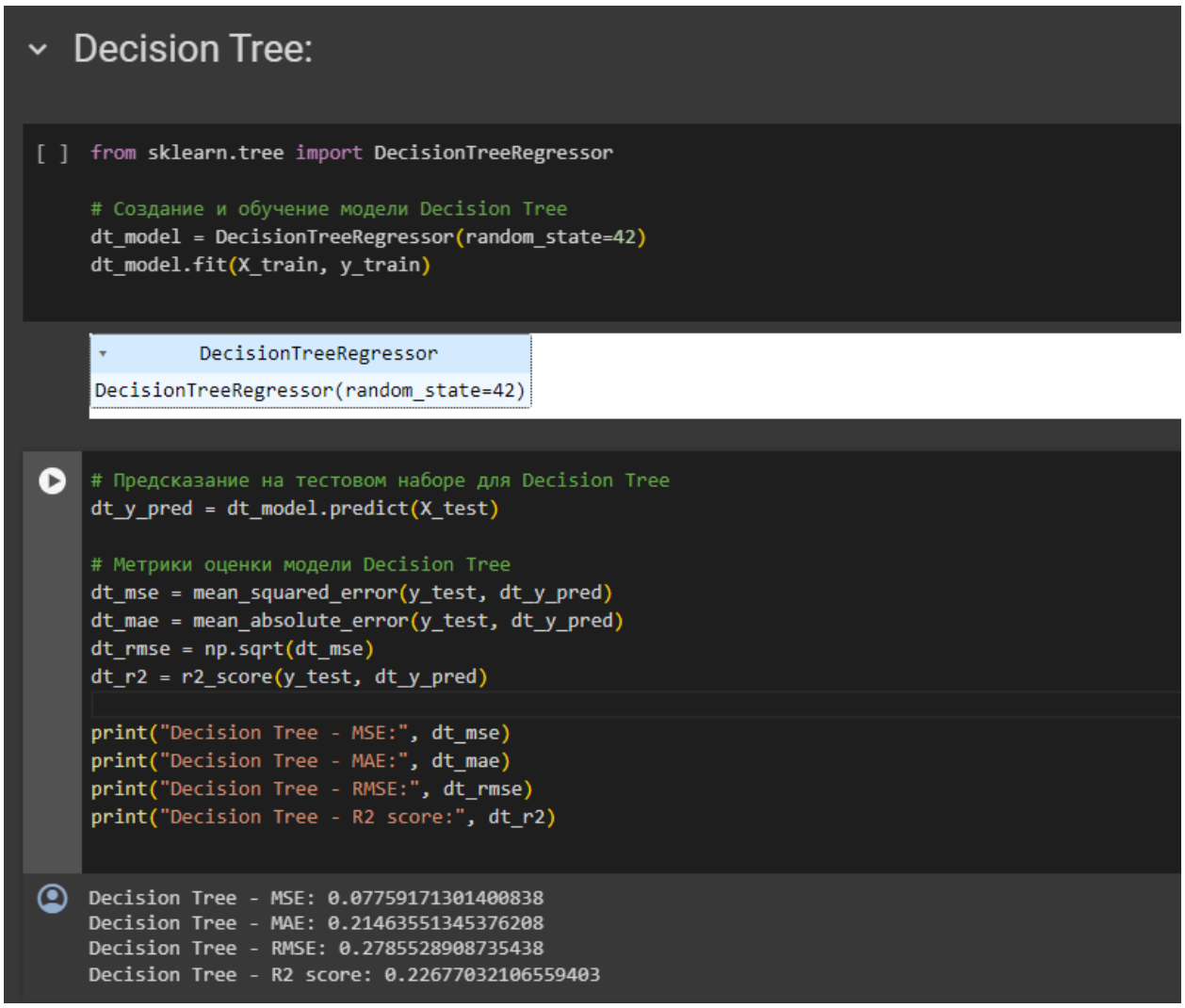


Рис. 10. Использование модели дерева решений

Также используем модель k-ближайших соседей (KNN) для задачи регрессии. Код выполняет предсказание целевой переменной на тестовом наборе данных, и вычисляются метрики оценки качества модели. Полученные значения метрик выводятся на экран для оценки эффективности модели KNN на тестовых данных (рис. 11).

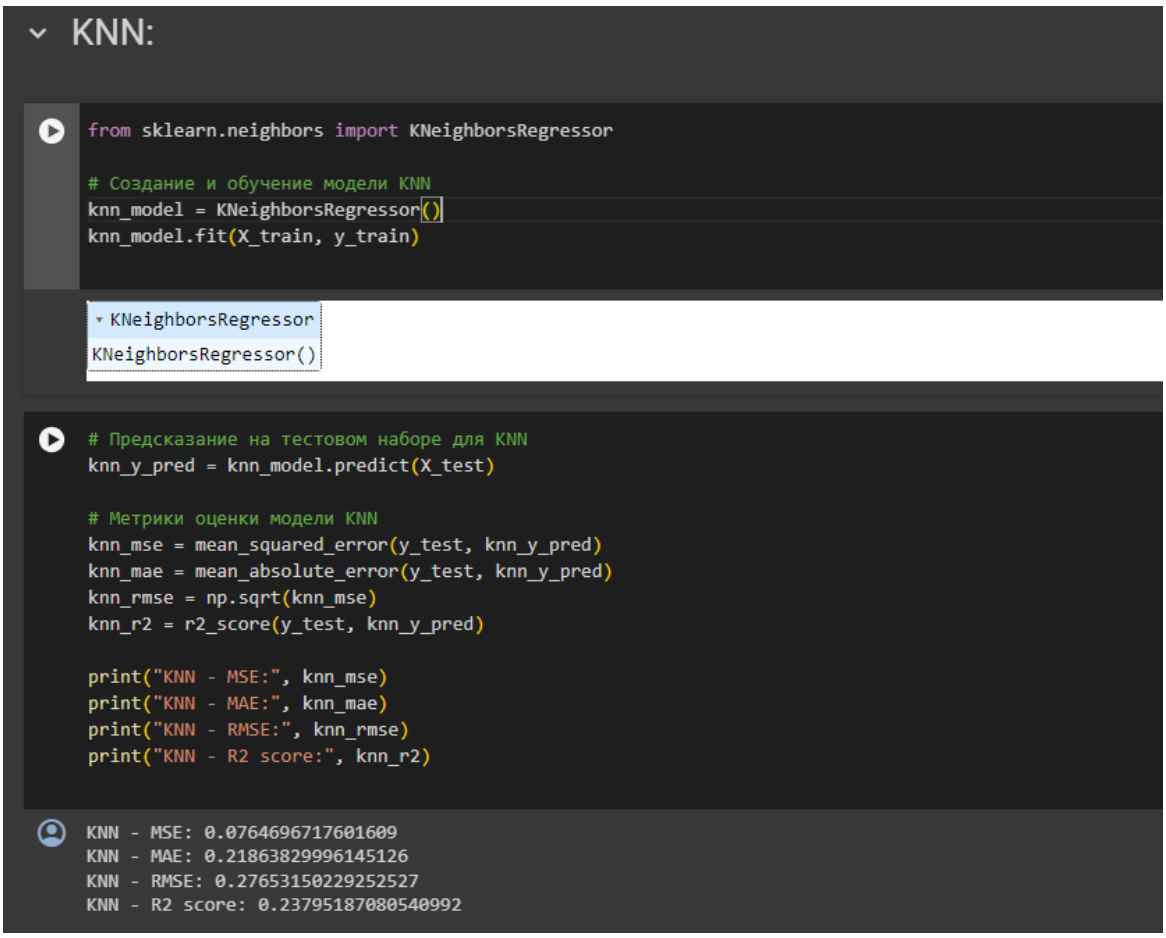


Рис.11. Использование модели k-ближайших соседей

Создадим и обучим последнюю модель XGBoost. Затем выполняется предсказание целевой переменной на тестовом наборе данных, и вычисляются метрики оценки качества модели. Полученные значения метрик выводятся на экран для оценки эффективности модели XGBoost на тестовых данных.

XGBoost:

```
from xgboost import XGBRegressor

# Создание и обучение модели XGBoost
xgb_model = XGBRegressor(random_state=42)
xgb_model.fit(X_train, y_train)
```

```
▸ XGBRegressor
```

```
[ ] # Предсказание на тестовом наборе для XGBoost
    xgb_y_pred = xgb_model.predict(X_test)

    # Метрики оценки модели XGBoost
    xgb_mse = mean_squared_error(y_test, xgb_y_pred)
    xgb_mae = mean_absolute_error(y_test, xgb_y_pred)
    xgb_rmse = np.sqrt(xgb_mse)
    xgb_r2 = r2_score(y_test, xgb_y_pred)

    print("XGBoost - MSE:", xgb_mse)
    print("XGBoost - MAE:", xgb_mae)
    print("XGBoost - RMSE:", xgb_rmse)
    print("XGBoost - R2 score:", xgb_r2)
```

```
XGBoost - MSE: 0.07741793504630061
XGBoost - MAE: 0.21441964946051
XGBoost - RMSE: 0.2782407860941681
XGBoost - R2 score: 0.22850208180339726
```

Рис.12. Использование модели XGBoost

Добавим дополнительные признаки для анализа, создавая новый датафрейм, который содержит три выбранных признака: год выпуска автомобиля, тип двигателя и тип трансмиссии. Далее разделяем данные на обучающий и тестовый наборы. Создаем ранее используемые модели с новыми фичами (рис. 13, 14, 15, 16).

```
# Предсказание на тестовом наборе для Random Forest
rf_y_pred3 = rf_model3.predict(X_test3)

# Метрики оценки модели Random Forest
rf_mse3 = mean_squared_error(y_test3, rf_y_pred3)
rf_mae3 = mean_absolute_error(y_test3, rf_y_pred3)
rf_rmse3 = np.sqrt(rf_mse3)
rf_r23 = r2_score(y_test3, rf_y_pred3)

print("Random Forest - MSE:", rf_mse3)
print("Random Forest - MAE:", rf_mae3)
print("Random Forest - RMSE:", rf_rmse3)
print("Random Forest - R2 score:", rf_r23)
```

```
Random Forest - MSE: 0.06401756835351638
Random Forest - MAE: 0.18438170136834003
Random Forest - RMSE: 0.25301693293832406
Random Forest - R2 score: 0.36204161597746365
```

Рис.13. Использование модели Random Forest

```
# Предсказание на тестовом наборе для Decision Tree
dt_y_pred3 = dt_model3.predict(X_test3)

# Метрики оценки модели Decision Tree
dt_mse3 = mean_squared_error(y_test3, dt_y_pred3)
dt_mae3 = mean_absolute_error(y_test3, dt_y_pred3)
dt_rmse3 = np.sqrt(dt_mse3)
dt_r23 = r2_score(y_test3, dt_y_pred3)

print("Decision Tree - MSE:", dt_mse3)
print("Decision Tree - MAE:", dt_mae3)
print("Decision Tree - RMSE:", dt_rmse3)
print("Decision Tree - R2 score:", dt_r23)
```

```
Decision Tree - MSE: 0.07086133811507966
Decision Tree - MAE: 0.20573232361354177
Decision Tree - RMSE: 0.2661979303358305
Decision Tree - R2 score: 0.293840957783151
```

Рис.14. Использование модели Decision Tree

```
# Предсказание на тестовом наборе для KNN
knn_y_pred3 = knn_model3.predict(X_test3)

# Метрики оценки модели KNN
knn_mse3 = mean_squared_error(y_test3, knn_y_pred3)
knn_mae3 = mean_absolute_error(y_test3, knn_y_pred3)
knn_rmse3 = np.sqrt(knn_mse3)
knn_r23 = r2_score(y_test3, knn_y_pred3)

print("KNN - MSE:", knn_mse3)
print("KNN - MAE:", knn_mae3)
print("KNN - RMSE:", knn_rmse3)
print("KNN - R2 score:", knn_r23)
```

```
KNN - MSE: 0.06689625753775882
KNN - MAE: 0.17950401136500005
KNN - RMSE: 0.25864310842889054
KNN - R2 score: 0.33335442982973174
```

Рис. 15. Использование модели KNN

```
# Предсказание на тестовом наборе для XGBoost
xgb_y_pred3 = xgb_model3.predict(X_test3)

# Метрики оценки модели XGBoost
xgb_mse3 = mean_squared_error(y_test3, xgb_y_pred3)
xgb_mae3 = mean_absolute_error(y_test3, xgb_y_pred3)
xgb_rmse3 = np.sqrt(xgb_mse3)
xgb_r23 = r2_score(y_test3, xgb_y_pred3)

print("XGBoost - MSE:", xgb_mse3)
print("XGBoost - MAE:", xgb_mae3)
print("XGBoost - RMSE:", xgb_rmse3)
print("XGBoost - R2 score:", xgb_r23)
```

```
XGBoost - MSE: 0.06738253874799972
XGBoost - MAE: 0.1936822821531006
XGBoost - RMSE: 0.2595814684217649
XGBoost - R2 score: 0.32850846046468696
```

Рис.16. Использование модели XGBoost

Сравним метрики моделей (рис. 17).

Random Forest	
MSE	0.06401756835351638
MAE	0.18438170136834003
RMSE	0.25301693293832406
R2	0.36204161597746365

Decision Tree	
MSE	0.07086133811507966
MAE	0.20573232361354177
RMSE	0.2661979303358305
R2	0.293840957783151

KNN	
MSE	0.06689625753775882
MAE	0.17950401136500005
RMSE	0.25864310842889054
R2	0.33335442982973174

XGBoost	
MSE	0.06738253874799972
MAE	0.1936822821531006
RMSE	0.2595814684217649
R2	0.32850846046468696

Рис.17. Сравнение моделей

В данном случае модель Random Forest демонстрирует лучшую производительность с точки зрения всех оценочных метрик, таких как среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE), корень из среднеквадратичной ошибки (RMSE) и коэффициент детерминации (R2), что указывает на более точные предсказания цен по сравнению с другими рассмотренными моделями.

Выводы

В результате проведенного исследования полученные модели прогнозирования, созданные в среде Google Colaboratory можно использовать для прогнозирования, а данные при прогнозе будут довольно достоверными.

Библиографический список

1. Найденова К.А., Невзорова О.А. Машинное обучение в задачах обработки естественного языка: обзор современного состояния исследований // Научные исследования. 2019. № 3 (1). С. 45-60.
2. Боброва М.В., Мастилин А.Е. Машинное обучение в кибербезопасности // Компьютерная безопасность. 2020. № 2 (3). С. 78-92.
3. Коротеев М.В. Современные тенденции в технологии машинного обучения // Технологии искусственного интеллекта. 2018. № 1 (1). С. 26-34.
4. Полетаева Н.Г. Классификация систем машинного обучения // Информационные технологии. 2019. № 4 (2). С. 14-29.
5. Дементьев В.Е., Киреев С.Х. Выбор алгоритмов машинного обучения для классификации текстовых документов // Экономика и прогнозирование. 2021. № 2 (5). С. 112-130.