МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙАВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Кафедра 319 «Системы интеллектуального мониторинга»

КУРСОВАЯ РАБОТА

по дисциплине «Нейронные сети»

«Решение задачи нелинейной регрессии с использованием XGboost»

Студент _____ Муравьев И.А.

Группа М30-435Б-20

Руководитель _____ Нагибин С.Я.

Оценка _____ Дата защиты «___» декабря 2023 г.

Москва 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Кафедра 319 «Системы интеллектуального мониторинга»

З А Д А Н И Е

на курсовую работу по дисциплине

Нейронные сети

Студент М30-435Б-20, Муравьев Иван Алексеевич
(№ группы, Ф. И. О.)

Тема Решение задачи нелинейной регрессии с использованием XGboost

Перечень вопросов, подлежащих разработке в курсовой работе

Рекомендуемая литература

Задание выдано «___» 2023 г.
Руководитель Нагибин С.Я., профессор
(Ф. И. О., должность, подпись)

Студент _____
(подпись)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В теоретической части будут рассмотрены основы нелинейной регрессии и методики ее решения, с фокусом на использовании XGBoost – мощного алгоритма машинного обучения, который широко применяется в задачах регрессии и классификации. Будут рассмотрены основные принципы работы XGBoost, его возможности для моделирования нелинейных зависимостей и преимущества по сравнению с другими методами решения задачи регрессии.

В практической части будет представлен конкретный пример применения XGBoost в решении задачи нелинейной регрессии. Будет показан процесс выбора набора данных, обучение модели, анализ и интерпретацию результатов, что обеспечит понимание применения данного алгоритма в решении реальных задач нелинейной регрессии.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Нелинейная регрессия

Нелинейная регрессия – это метод анализа данных, который используется для моделирования отношений между зависимой переменной и одной или несколькими независимыми переменными. В отличие от линейной регрессии, нелинейная регрессия позволяет учитывать нелинейные связи между переменными, что делает ее более гибкой для описания разнообразных явлений.

Нелинейная регрессия широко применяется в различных областях, таких как экономика, биология, физика, медицина и многие другие, где взаимосвязи между переменными могут быть сложными и нелинейными.

1.2 Решение задачи нелинейной регрессии

Решение задачи регрессии в машинном обучении направлено на предсказание непрерывной переменной на основе входных данных. Это требует поиска функции, которая наилучшим образом описывает зависимость между входными признаками и выходным значением. В контексте нелинейной регрессии, где связь между переменными не является линейной, применяются различные методы.

Процесс решения задачи нелинейной регрессии обычно включает в себя следующие шаги:

1. Выбор модели: определение класса функций, которые могут описывать зависимость между входными и выходным значениями. Например, полиномиальные функции, экспоненциальные функции, логарифмические функции и другие.

2. Определение параметров модели: для выбранной модели нужно определить параметры. Например, если выбрана полиномиальная функция, необходимо определить коэффициенты полинома.

3. Оценка параметров: используя набор данных, проводится процесс обучения, в результате которого настраиваются параметры модели. Это может быть достигнуто с использованием методов оптимизации, таких как метод наименьших квадратов.

4. Оценка качества модели: после обучения модели ее качество оценивается на отдельном тестовом наборе данных. Это может включать в себя вычисление метрик регрессии, таких как среднеквадратичная ошибка (MSE) или коэффициент детерминации (R^2).

Существует множество методов для решения задачи нелинейной регрессии, и одним из широко используемых инструментов является библиотека XGBoost.

1.3 XGBoost

XGBoost (eXtreme Gradient Boosting) представляет собой мощный алгоритм машинного обучения, который широко используется для задач регрессии и классификации. Он основан на идее градиентного бустинга, который комбинирует слабые модели (обычно деревья решений) для создания более сильной и точной модели.

Принцип работы XGBoost заключается в итеративном обучении деревьев решений, где каждое новое дерево исправляет ошибки предыдущего. Он использует градиентный спуск для минимизации функции потерь, что позволяет алгоритму фокусироваться на тех примерах, где происходят наибольшие ошибки.

Работа алгоритма начинается с создания первого дерева решений, которое приближается к искомой функции. Затем вычисляются ошибки прогноза, и следующее дерево строится так, чтобы минимизировать эти ошибки. Процесс повторяется, и каждое новое дерево фокусируется на тех областях данных, где предыдущие ошибаются больше всего. Это позволяет XGBoost эффективно выявлять сложные зависимости и шаблоны в данных.

XGBoost имеет несколько ключевых преимуществ:

1. Высокая производительность и эффективность: XGBoost обладает выдающейся производительностью благодаря своей параллельной обработке и высокооптимизированным структурам данных. Он способен обрабатывать большие объемы данных и решать сложные задачи с высокой точностью.

2. Регуляризация: XGBoost включает в себя механизмы регуляризации, такие как L1 (Lasso) и L2 (Ridge), что помогает предотвратить переобучение модели и повысить ее обобщающую способность.

3. Обработка отсутствующих данных: Алгоритм XGBoost способен эффективно работать с отсутствующими данными, позволяя модели обучаться на неполных наборах данных без необходимости предварительной обработки.

4. Поддержка различных функций потерь: XGBoost поддерживает разнообразие функций потерь, что позволяет выбирать наилучший вариант в

зависимости от конкретной задачи. Это включает в себя логистическую регрессию для классификации и квадратичную функцию потерь для регрессии.

5. Автоматический выбор переменных: Алгоритм XGBoost способен автоматически выбирать наиболее важные переменные, что упрощает процесс выбора признаков и может повысить обобщающую способность модели.

6. Интерпретируемость: XGBoost предоставляет инструменты для анализа важности признаков, что помогает понимать, какие факторы вносят наибольший вклад в прогнозирование модели. Это помогает в интерпретации результатов и принятия обоснованных решений.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

В практической части данного исследования будет рассмотрено применение XGBoost для решения конкретной задачи нелинейной регрессии. Для этого будет выбран набор данных, который представляет собой разнообразный набор факторов, оказывающих влияние на целевую переменную. Это позволит оценить способность XGBoost выявлять сложные нелинейные закономерности в данных.

Полный исходный текст программы находится в Приложении А.

2.1 Выбор набора данных

При выборе набора данных для решения задачи нелинейной регрессии важно учитывать разнообразие факторов, способных оказывать влияние на целевую переменную. Идеальный датасет должен содержать достаточное количество наблюдений, разнообразные признаки и представлять реальные условия, в которых модель будет применяться.

В данной работе был выбран набор данных, в котором пять различных факторов (X1-X5) влияют на целевую переменную (Y). Данный выбор обусловлен желанием продемонстрировать способность XGBoost работать с несколькими признаками и выявлять сложные нелинейные зависимости между ними.

Набор данных содержит 1000 строк, что обеспечивает достаточный объем данных для обучения и тестирования модели.

Пример данных из набора представлен в таблице 1.

Таблица 1 – Пример используемых данных

x1	x2	x3	x4	x5	y
14.47878143	5.11241001	552.7056502	7888.926227	1.543017072	8466.847538
30.0889382	4.840943149	572.2144672	8938.384509	0.717611137	9553.683909
19.8594994	5.296732317	28.47946612	3019.026376	1.656854509	3123.311226

2.2 Обучение модели

Для обучения модели выбранный датасет был разделен на независимые и зависимую переменные, после чего они были разделены на обучающую и тестовую выборки. Это необходимо для последующей оценки результата обучения модели.

В Листинге 2.1 представлен код, использующий библиотеку scikit-learn для такого разделения.

Листинг 2.1 – Разделение данных на обучающую и тестовую выборки

```
from sklearn.model_selection import train_test_split

X = data[["x1", "x2", "x3", "x4", "x5"]]
y = data["y"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
  test_size=0.2)
```

Затем было проведено обучение модели на обучающих данных (X_train и y_train), а также предсказание значений для факторов из тестовой выборки. Код данного процесса представлен в Листинге 2.2.

Листинг 2.2 – Обучение модели XGBRegressor и предсказание значений

```
from xgboost import XGBRegressor

model = XGBRegressor()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

2.3 Оценка качества модели

Следующим этапом после обучения модели является проверка соответствия предсказанных значений реальным на тестовой выборке. Для этого используются различные метрики, в данной работе будут использованы следующие:

- среднеквадратическая ошибка (MSE, среднее значение квадрата отклонения предсказанного значения от истинного)
- средняя абсолютная и абсолютная процентная ошибки (MAE и MAPE, среднее абсолютных и относительных значений отклонения предсказанного значения от истинного соответственно)
- R-квадрат (коэффициент детерминации, инвариантный к масштабу данных показатель, который показывает долю дисперсии зависимой переменной, объяснённой с помощью данной регрессионной модели)
- коэффициент корреляции (статистическая мера, которая вычисляет силу связи между относительными движениями двух переменных)

Код, отвечающий за вычисление вышеуказанных метрик, представлен в Листинге 2.3.

Листинг 2.3 – Вычисление метрик

```
from sklearn.metrics import mean_squared_error, r2_score, mean_ab-
  solute_error
import numpy as np

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mape = np.mean(np.abs(y_test - y_pred) / np.abs(y_pred))
r2 = r2_score(y_test, y_pred)
correlation = np.corrcoef(y_test, y_pred)[0, 1]
```

Результаты вычисления данных метрик показаны на Рисунке 1. На основе полученных значений можно сделать вывод, что полученная модель обладает очень сильной предсказательной способностью на данном наборе данных.

```
Mean Squared Error: 6271.243324020101
Mean Absolute Error: 59.81231861320314
Relative Absolute Error: 0.009937208508144104
R-squared: 0.999264012034646
Correlation Coefficient: 0.9996358218193421
```

Рисунок 1 – Метрики полученной модели

ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы были изучены основы нелинейной регрессии и методы ее решения, с акцентом на применении алгоритма XGBoost. В теоретической части представлен обзор основных принципов и преимуществ XGBoost в контексте регрессионного моделирования, его способность эффективно моделировать сложные зависимости в данных.

В практической части был представлен конкретный пример использования XGBoost для решения задачи нелинейной регрессии. Процесс включал в себя выбор набора данных, обучение модели и оценку ее качества. Анализ результатов позволил получить практические навыки в работе с XGBoost и понимание его применимости в контексте реальных данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 XGBoost Documentation — xgboost 2.0.3 documentation / [Электронный ресурс] // XGBoost Documentation — xgboost 2.0.3 documentation : [сайт]. — URL: https://xgboost.readthedocs.io/en/stable/ (дата обращения: 20.12.2023).

2 Олег Седухин CatBoost, XGBoost и выразительная способность решающих деревьев / Олег Седухин [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/companies/ods/articles/645887/ (дата обращения: 20.12.2023).

3 Дмитрий Кирьянов Машинное обучение — 2. Нелинейная регрессия и численная оптимизация / Дмитрий Кирьянов [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/companies/nerepetitor/articles/252571/ (дата обращения: 20.12.2023).

4 Метрики классификации и регрессии / [Электронный ресурс] // Яндекс Образование : [сайт]. — URL: https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii (дата обращения: 20.12.2023).

5 Орешков Вячеслав Метрики качества линейных регрессионных моделей | Loginom / Орешков Вячеслав [Электронный ресурс] // Loginom : [сайт]. — URL: https://loginom.ru/blog/quality-metrics (дата обращения: 20.12.2023).

ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ

Файл cw.ipynb

```
import pandas as pd

data = pd.read_csv("cw.csv")

from sklearn.model_selection import train_test_split

X = data[["x1", "x2", "x3", "x4", "x5"]]
y = data["y"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
  test_size=0.2)

from xgboost import XGBRegressor

model = XGBRegressor()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
from sklearn.metrics import mean_squared_error, r2_score, mean_ab-
  solute_error
import numpy as np

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mape = np.mean(np.abs(y_test - y_pred) / np.abs(y_pred))
r2 = r2_score(y_test, y_pred)
correlation = np.corrcoef(y_test, y_pred)[0, 1]

print(f"Mean Squared Error: {mse}")
print(f"Mean Absolute Error: {mae}")
print(f"Relative Absolute Error: {mape}")
print(f"R-squared: {r2}")
print(f"Correlation Coefficient: {correlation}")

import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred)
plt.plot(
    [min(y_test), max(y_test)],
    [min(y_test), max(y_test)],
    linestyle="--",
    color="red",
    linewidth=2,
    label="Ideal Line",
)
plt.title("True vs Predicted Values")
plt.xlabel("True Values")
plt.ylabel("Predicted Values")
plt.legend()
plt.show()
```