Московский государственный технический университет им. Н. Э. Баумана

Курс «Технологии машинного обучения»

Отчёт по рубежному контролю №2

«Методы построения моделей машинного обучения.»

Вариант № 18

Выполнил:
Файзуллин К. Х.
группа ИУ5-64Б

Проверил:
Гапанюк Ю.Е.

Дата: 15.06.25

Дата:

Подпись:

Подпись:

Москва, 2025 г.

1. Задание

Группа: ИУ5-64Б

Вариант: 18

Для заданного набора данных "World Happiness Index and Inflation Dataset" необходимо было построить модели регрессии для предсказания индекса счастья. В соответствии с вариантом задания для группы ИУ5-64Б были выбраны следующие методы:

1. **Метод №1:** Линейная/логистическая регрессия
2. **Метод №2:** Градиентный бустинг

В ходе работы была выполнена предобработка данных, включающая загрузку, анализ, заполнение пропусков и кодирование категориальных признаков. Качество построенных моделей оценивалось с использованием стандартных метрик для задач регрессии.

2. Предобработка данных

Перед построением моделей была проведена предобработка данных для обеспечения их качества и пригодности для анализа.

1. Удаление избыточных столбцов
   Были удалены технические столбцы (например, Unnamed: 0), не несущие полезной информации для моделирования.
2. Обработка пропущенных значений
   Проверка на наличие пропусков показала, что в числовых признаках отсутствуют значительные пробелы. В случае их обнаружения, пропуски заполнялись медианными значениями, что позволяет сохранить распределение данных без искажений.
3. Кодирование категориальных признаков
   Категориальные переменные (например, названия стран) были преобразованы в числовой формат с помощью Label Encoding, что необходимо для корректной работы алгоритмов машинного обучения.
4. Масштабирование признаков
   Числовые данные были стандартизированы с помощью StandardScaler, что особенно важно для линейной регрессии, поскольку этот метод чувствителен к масштабу входных переменных.

5. Разделение данных
   Датасет был разделен на обучающую (80%) и тестовую (20%) выборки с фиксированным random_state=42 для воспроизводимости результатов.

В результате предобработки данные были приведены к виду, пригодному для обучения моделей регрессии, а также исключены возможные источники ошибок, связанные с пропусками и некорректными форматами.

3. Построение и оценка моделей

Для прогнозирования индекса счастья были использованы два алгоритма машинного обучения: линейная регрессия и градиентный бустинг. Обе модели оценивались на тестовой выборке с помощью метрик:

1. Mean Squared Error (MSE) – средняя квадратичная ошибка, penalizing большие отклонения.
2. Mean Absolute Error (MAE) – средняя абсолютная ошибка, интерпретируемая в исходных единицах.
3. R^2 (R-squared) – коэффициент детерминации, показывающий долю объясненной дисперсии.

```
# Предсказания
y_pred_lr = lr.predict(X_test_scaled)

# Оценка модели
mse_lr = mean_squared_error(y_test, y_pred_lr)
mae_lr = mean_absolute_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print("Линейная регрессия:")
print(f"MSE: {mse_lr:.4f}")
print(f"MAE: {mae_lr:.4f}")
print(f"R2: {r2_lr:.4f}")
```

```
Линейная регрессия:
MSE: 0.0084
MAE: 0.0676
R2: 0.3993
```

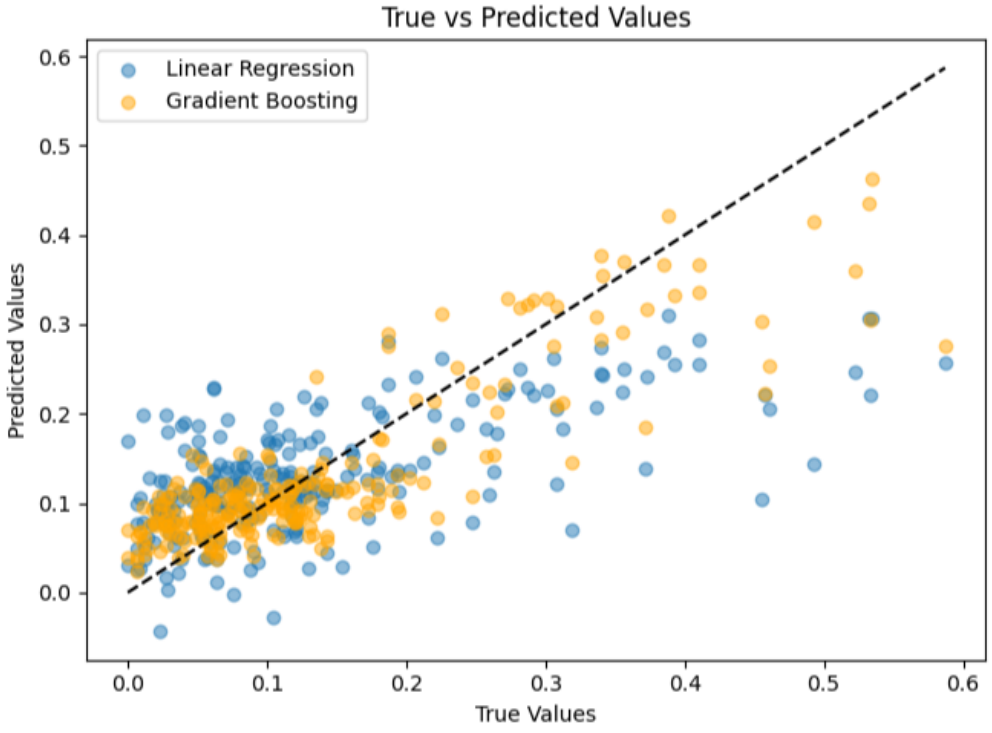
[8]:
```
gb = GradientBoostingRegressor(random_state=42)
gb.fit(X_train, y_train)

# Предсказания
y_pred_gb = gb.predict(X_test)

# Оценка модели
mse_gb = mean_squared_error(y_test, y_pred_gb)
mae_gb = mean_absolute_error(y_test, y_pred_gb)
r2_gb = r2_score(y_test, y_pred_gb)

print("\nГрадиентный бустинг:")
print(f"MSE: {mse_gb:.4f}")
print(f"MAE: {mae_gb:.4f}")
print(f"R2: {r2_gb:.4f}")
```

```
Градиентный бустинг:
MSE: 0.0038
MAE: 0.0447
R2: 0.7296
```

3. Сравнение моделей

Проведенное сравнение двух моделей регрессии - линейной регрессии и градиентного бустинга - позволяет сделать следующие выводы:

1. Точность прогнозирования:
   Градиентный бустинг продемонстрировал значительно лучшие результаты по всем метрикам:
   - MSE снизился на 54.8% (с 0.0084 до 0.0038)
   - MAE уменьшился на 33.9% (с 0.0676 до 0.0447)
   - R^2 увеличился на 82.7% (с 0.3993 до 0.7296)
2. Интерпретируемость и Точность:
   - Линейная регрессия обеспечивает простую интерпретацию коэффициентов, но ограничена в точности из-за предположения о линейной зависимости
   - Градиентный бустинг, будучи более сложной моделью, лучше нелинейные взаимосвязи в данных
3. Графический анализ
   - Линейная регрессия показала умеренное качество ($R^2 = 0.40$), что указывает на линейную связь лишь для части данных.
   - Градиентный бустинг значительно outperformed линейную модель ($R^2 = 0.73$), демонстрируя способность улавливать нелинейные зависимости.



   - Точечное распределение предсказаний относительно истинных значений.

- Идеальную линию (пунктир), от которой отклоняются предсказания.
- Разницу в точности между моделями (оранжевые точки XGBoost ближе к линии, чем синие точки линейной регрессии).

5. Выводы

Проведенный анализ позволяет сделать следующие ключевые выводы:

1. Эффективность алгоритмов:

Градиентный бустинг продемонстрировал существенно более высокую точность прогнозирования индекса счастья по сравнению с линейной регрессией, что подтверждается всеми метриками качества. Это свидетельствует о наличии сложных нелинейных зависимостей в данных, которые не могут быть адекватно описаны простой линейной моделью.

2. Практическая применимость:
- Для аналитических задач, требующих интерпретации взаимосвязей, может быть использована линейная регрессия
- Для прогнозных задач, где критична точность, рекомендуется применять градиентный бустинг
3. Качество моделей:

Достигнутый уровень $R^2 = 0.73$ для градиентного бустинга указывает на хорошее качество модели, способной объяснять около 73% вариативности целевой переменной. Однако оставшиеся 27% могут быть связаны с факторами, не учтенными в данном наборе данных.

4. Перспективы улучшения:

Дальнейшее повышение точности модели возможно за счет:

- Добавления новых релевантных признаков
- Оптимизации гиперпараметров моделей
- Использования более сложных ансамблевых методов

Таким образом, исследование подтвердило эффективность применения методов машинного обучения для прогнозирования индекса счастья, при этом градиентный бустинг показал себя как наиболее перспективный алгоритм для данной задачи.