`

Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Отчет по лабораторной работе 5

Специальность ИИ-23

Выполнил:

Гавришук В.Р.

Студент группы ИИ-23

Проверил:

Андренко К. В.
Преподаватель-стажёр
Кафедры ИИТ,
«___» _____2025 г.

Брест 2025

`

Лабораторная работа №5. **Деревья решений**

Цель: на практике сравнить работу нескольких алгоритмов одиночного дерева решений, случайного леса и бустинга для деревьев решений.

Задачи:

1. Загрузить датасет по варианту;
2. Разделить данные на обучающую и тестовую выборки;
3. Обучить на обучающей выборке одиночное дерево, случайный лес и реализовать бустинг для решающих деревьев (AdaBoost, CatBoost, XGBoost);
4. Оценить точность каждой модели на тестовой выборке;
5. Сравнить результаты, сделать выводы о применимости каждого метода для данного набора данных.

Задание по вариантам

Вариант 4

- Digits
- Распознать, какая цифра (от 0 до 9) изображена на картинке 8x8 пикселей
- **Задания:**
    1. Загрузите встроенный в scikit-learn набор данных digits;
    2. Разделите данные на обучающую и тестовую выборки;
    3. Обучить на обучающей выборке одиночное дерево, случайный лес и реализовать бустинг для решающих деревьев (AdaBoost, CatBoost, XGBoost);
    4. Для каждой модели выведите classification_report (sklearn.metrics), содержащий основные метрики для каждого класса;
    5. Сравните общую точность моделей и определите, какая из них лучше всего подходит для этой задачи.

Ход работы:

```
warnings.filterwarnings("ignore")

digits = load_digits()
X, y = digits.data, digits.target

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

models = {
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
    "AdaBoost": AdaBoostClassifier(
        estimator=DecisionTreeClassifier(max_depth=2),
        n_estimators=100,
        random_state=42
    ),
```

```
    "CatBoost": CatBoostClassifier(
        iterations=200,
        depth=6,
        learning_rate=0.1,
        verbose=False,
        random_state=42
    ),
    "XGBoost": XGBClassifier(
        n_estimators=200,
        learning_rate=0.1,
        max_depth=6,
        use_label_encoder=False,
        eval_metric='mlogloss',
        random_state=42
    ),
}

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    results[name] = acc
    print(f"\n{name}")
    print(classification_report(y_test, y_pred))
    print(f"Accuracy: {acc:.4f}")

fig, axes = plt.subplots(2, 5, figsize=(10, 5))
fig.suptitle("Примеры изображений и предсказаний (Decision Tree)", fontsize=14)

y_pred_sample = models["Decision Tree"].predict(X_test)
for i, ax in enumerate(axes.flat):
    ax.imshow(X_test[i].reshape(8, 8), cmap="gray")
    ax.set_title(f"Pred: {y_pred_sample[i]}\nTrue: {y_test[i]}")
    ax.axis("off")

plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 5))
plt.barh(list(results.keys()), list(results.values()), color="skyblue")
plt.xlabel("Accuracy")
plt.title("Сравнение точности моделей")
plt.xlim(0, 1)
for i, v in enumerate(results.values()):
    plt.text(v + 0.01, i, f"{v:.3f}", va="center")
plt.show()
```

Результат работы программы:

```
Decision Tree
              precision    recall  f1-score   support

           0       0.94      0.91      0.92        54
           1       0.87      0.71      0.78        55
           2       0.78      0.81      0.80        53
           3       0.91      0.91      0.91        55
           4       0.85      0.87      0.86        54
```

```
           5       0.89      0.87      0.88        55
           6       0.96      0.89      0.92        54
           7       0.89      0.91      0.90        54
           8       0.72      0.83      0.77        52
           9       0.75      0.81      0.78        54

    accuracy                           0.85       540
   macro avg       0.86      0.85      0.85       540
weighted avg       0.86      0.85      0.85       540

Accuracy: 0.8519

 Random Forest
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        54
           1       0.93      1.00      0.96        55
           2       1.00      1.00      1.00        53
           3       0.96      0.98      0.97        55
           4       0.96      0.98      0.97        54
           5       1.00      0.98      0.99        55
           6       1.00      0.98      0.99        54
           7       0.90      1.00      0.95        54
           8       0.96      0.88      0.92        52
           9       0.96      0.89      0.92        54

    accuracy                           0.97       540
   macro avg       0.97      0.97      0.97       540
weighted avg       0.97      0.97      0.97       540

Accuracy: 0.9667

 AdaBoost
              precision    recall  f1-score   support

           0       1.00      0.89      0.94        54
           1       0.76      0.64      0.69        55
           2       0.79      0.64      0.71        53
           3       0.93      0.91      0.92        55
           4       0.86      0.94      0.90        54
           5       0.87      0.82      0.84        55
           6       0.92      0.89      0.91        54
           7       0.82      0.93      0.87        54
           8       0.56      0.77      0.65        52
           9       0.80      0.80      0.80        54

    accuracy                           0.82       540
   macro avg       0.83      0.82      0.82       540
```

`

weighted avg	0.83	0.82	0.82	540

Accuracy: 0.8222

CatBoost

	precision	recall	f1-score	support
0	1.00	0.96	0.98	54
1	0.93	0.98	0.96	55
2	1.00	1.00	1.00	53
3	0.96	1.00	0.98	55
4	0.96	0.96	0.96	54
5	1.00	0.95	0.97	55
6	1.00	0.98	0.99	54
7	0.96	1.00	0.98	54
8	0.96	0.92	0.94	52
9	0.95	0.96	0.95	54
accuracy			0.97	540
macro avg	0.97	0.97	0.97	540
weighted avg	0.97	0.97	0.97	540

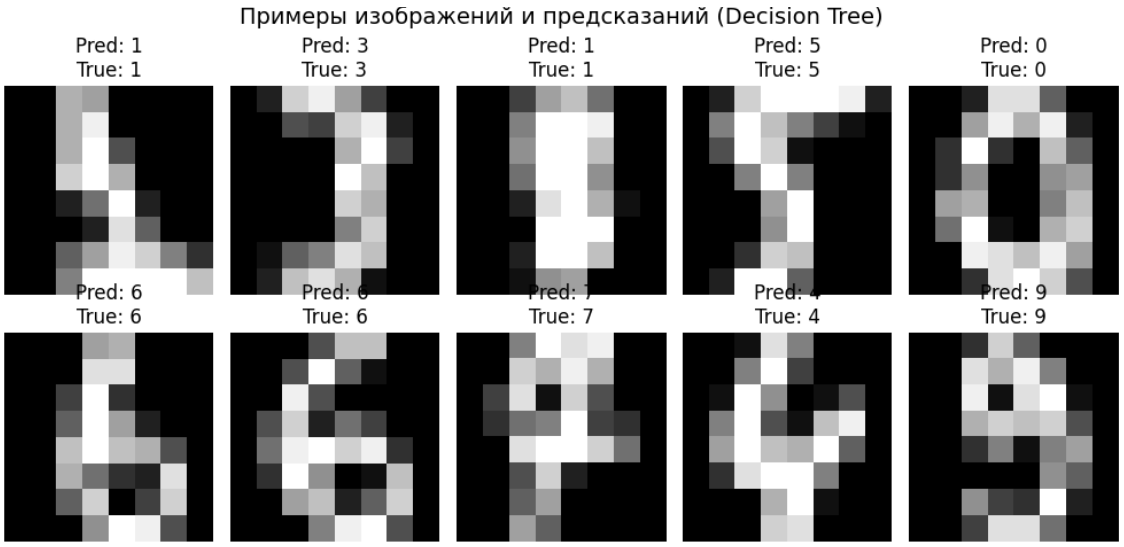
Accuracy: 0.9722

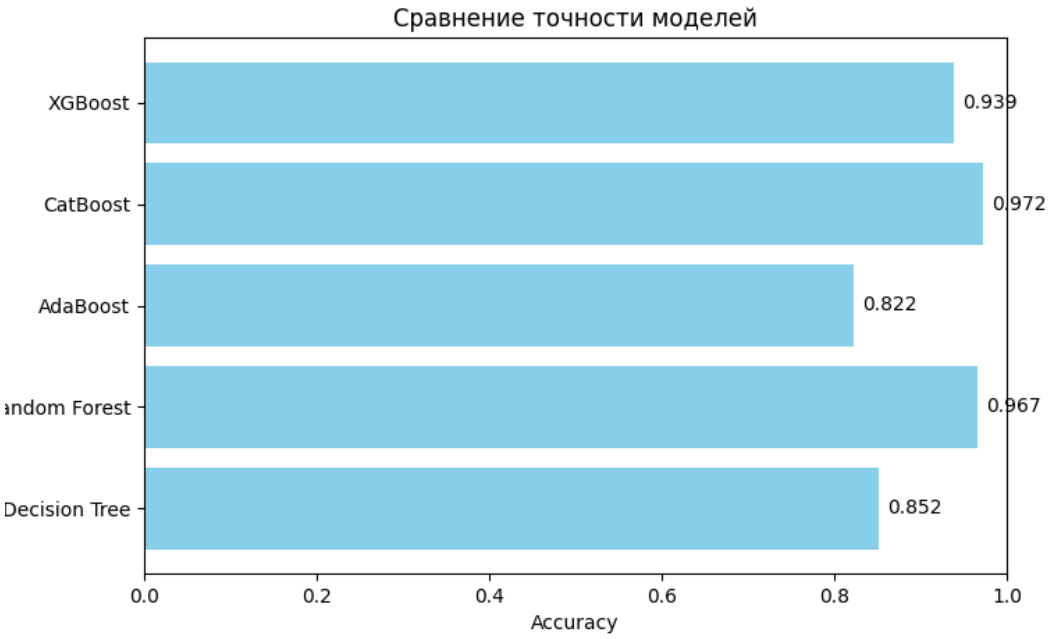
XGBoost

	precision	recall	f1-score	support
0	0.96	0.91	0.93	54
1	0.89	0.87	0.88	55
2	0.96	0.98	0.97	53
3	0.92	0.98	0.95	55
4	0.95	0.96	0.95	54
5	0.93	0.96	0.95	55
6	1.00	0.96	0.98	54
7	0.95	1.00	0.97	54
8	0.90	0.87	0.88	52
9	0.94	0.89	0.91	54
accuracy			0.94	540
macro avg	0.94	0.94	0.94	540
weighted avg	0.94	0.94	0.94	540

Accuracy: 0.9389

`





Можно сделать вывод: наилучшим оказался CatBoost по следующим причинам:

1. Используется современный бустинг, который оптимизирован под малые и средние табличные данные;
2. Обеспечивается баланс между скоростью и точностью обучения;
3. Легко справляется с нелинейными зависимостями между признаками (пикселями);
4. Требуется минимальная настройка гиперпараметров для хорошего результата.

Вывод: на практике сравнил работу нескольких алгоритмов одиночного дерева решений, случайного леса и бустинга для деревьев решений.