

Министерство образования Республики Беларусь  
Учреждение образования  
“Брестский государственный технический университет”  
Кафедра интеллектуально-информационных технологий

Интеллектуальный анализ данных  
Лабораторная работа №5  
Деревья решений

Выполнила:  
студентка 4 курса  
группы ИИ-24  
Алешко А. В.  
Проверила:  
Андренко К. В.

Брест-2025

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

### **Задание:**

Вариант 1

Iris

Определить вид ириса (setosa, versicolor, virginica) по измерениям его цветка

Задания:

1. Загрузите данные и ознакомьтесь с ними;
2. Разделите выборку на обучающую и тестовую;
3. Обучить на обучающей выборке одиночное дерево, случайный лес и реализовать бустинг для решающих деревьев (AdaBoost, CatBoost, XGBoost);
4. Оцените точность (accuracy) каждой модели на тестовой выборке;
5. Сравните результаты и сделайте вывод, какая модель лучше всего справилась с этой задачей.

### **Код программы(вариант 1):**

```
import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,
AdaBoostClassifier
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
import warnings
warnings.filterwarnings('ignore')

iris = datasets.load_iris()
X = iris.data
y = iris.target

print("Информация о датасете Iris")
print(f"Размерность признаков: {X.shape}")
print(f"Размерность целевой переменной: {y.shape}")
print(f"Названия классов: {iris.target_names}")
#Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
```

```

)
print(f"\nРазмер обучающей выборки: {X_train.shape[0]}")
print(f"Размер тестовой выборки: {X_test.shape[0]}")

#Обучение моделей
#Одиночное дерево решений
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
#Случайный лес
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
#AdaBoost
ada = AdaBoostClassifier(n_estimators=100, random_state=42)
ada.fit(X_train, y_train)
#XGBoost
xgb = XGBClassifier(n_estimators=100, random_state=42,
eval_metric='mlogloss')
xgb.fit(X_train, y_train)
#CatBoost
cat = CatBoostClassifier(iterations=100, random_state=42,
verbose=0)
cat.fit(X_train, y_train)

#Оценка точности
models = {
    'Decision Tree': dt,
    'Random Forest': rf,
    'AdaBoost': ada,
    'XGBoost': xgb,
    'CatBoost': cat
}

results = {}
print("\nТочность моделей на тестовой выборке")
for name, model in models.items():
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    results[name] = acc
    print(f"{name}: {acc:.4f}")

#Сравнение и вывод
print("\nСравнение моделей")
results_df = pd.DataFrame(list(results.items()),
columns=['Модель', 'Accuracy'])
results_df = results_df.sort_values(by='Accuracy',
ascending=False)
print(results_df)

best_model = results_df.iloc[0]['Модель']
best_acc = results_df.iloc[0]['Accuracy']

```

```
print(f"\nВывод: Лучшая модель — {best_model} с точностью  
{best_acc:.4f}.")  
print("Все модели показали высокую точность, что ожидаемо для  
простого датасета Iris.")  
print("Бустинговые методы (CatBoost, XGBoost, AdaBoost) и  
Random Forest обычно дают лучшие результаты, чем одиночное  
дерево.")
```

Результат работы программы:

Информация о датасете Iris

Размерность признаков: (150, 4)

Размерность целевой переменной: (150,)

Названия классов: ['setosa' 'versicolor' 'virginica']

Размер обучающей выборки: 105

Размер тестовой выборки: 45

Точность моделей на тестовой выборке

Decision Tree: 0.9333

Random Forest: 0.8889

AdaBoost: 0.9333

XGBoost: 0.9333

CatBoost: 0.9111

Сравнение моделей

	Модель	Accuracy
0	Decision Tree	0.933333
2	AdaBoost	0.933333
3	XGBoost	0.933333
4	CatBoost	0.911111
1	Random Forest	0.888889

Лучшая модель — Decision Tree с точностью 0.9333.

Все модели показали высокую точность, что ожидаемо для простого датасета Iris.

Бустинговые методы (CatBoost, XGBoost, AdaBoost) и Random Forest обычно дают лучшие результаты, чем одиночное дерево.

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