Министерство образования Республики Беларусь
Учреждение образования
«Брестский Государственный технический университет»
Кафедра ИИТ

Лабораторная работа №3
По дисциплине «Основы машинного обучения»
Тема: **«Сравнение классических методов классификации»**

Выполнил:
Студент 3 курса
Группы АС-65
Нестюк Н. С.
Проверил:
Крощенко А. А.

Брест 2025

Цель работы: На практике сравнить работу нескольких алгоритмов классификации, таких как метод k-ближайших соседей (k-NN), деревья решений и метод опорных векторов (SVM). Научиться подбирать гиперпараметры моделей и оценивать их влияние на результат.

Вариант 2

Ход работы:

Задание:

• Breast Cancer Wisconsin

• Определить, является ли опухоль злокачественной (malignant) или доброкачественной (benign);

• Задания:

1. Загрузите данные и выполните стандартизацию признаков;
2. Разделите данные на обучающую и тестовую части;
3. Обучите три классификатора: k-NN, Decision Tree и SVM;
4. Для каждой модели постройте матрицу ошибок и рассчитайте метрики precision, recall и F1-score для класса "злокачественная опухоль";
5. Сравните модели и укажите, какая из них наиболее надежна для минимизации ложноотрицательных прогнозов (когда злокачественная опухоль определяется как доброкачественная).

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score,
accuracy_score

df = pd.read_csv('breast_cancer.csv')
df = df.drop(['id', 'Unnamed: 32'], axis=1, errors='ignore')
df['diagnosis'] = df['diagnosis'].map({'M': 1, 'B': 0})

X = df.drop('diagnosis', axis=1)
y = df['diagnosis']

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

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print("ИССЛЕДОВАНИЕ k-NN С РАЗНЫМ K:")

k_values = range(1, 21)
best_accuracy = 0
best_k_accuracy = 1
best_fn = 100
best_k_fn = 1

for k in k_values:
    knn_temp = KNeighborsClassifier(n_neighbors=k)
    knn_temp.fit(X_train_scaled, y_train)
    y_pred = knn_temp.predict(X_test_scaled)
```

```
    accuracy = accuracy_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    fn = cm[1, 0]

    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_k_accuracy = k

    if fn < best_fn:
        best_fn = fn
        best_k_fn = k

print(f"Лучшая точность: k={best_k_accuracy}, точность={best_accuracy:.4f}")
print(f"Меньше всего ложноотрицательных: k={best_k_fn}, FN={best_fn}")

knn = KNeighborsClassifier(n_neighbors=best_k_fn)
dt = DecisionTreeClassifier(random_state=42)
svm = SVC(random_state=42)

knn.fit(X_train_scaled, y_train)
dt.fit(X_train, y_train)
svm.fit(X_train_scaled, y_train)

y_pred_knn = knn.predict(X_test_scaled)
y_pred_dt = dt.predict(X_test)
y_pred_svm = svm.predict(X_test_scaled)

models = {
    f'k-NN (k={best_k_fn})': y_pred_knn,
    'Decision Tree': y_pred_dt,
    'SVM': y_pred_svm
}

print("\nМЕТРИКИ ДЛЯ ЗЛОКАЧЕСТВЕННЫХ ОПУХОЛЕЙ:")

for name, y_pred in models.items():
    cm = confusion_matrix(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    print(f"\n{name}:")
    print(f"Матрица ошибок:\n{cm}")
    print(f"Precision: {precision:.4f}, Recall: {recall:.4f}, F1: {f1:.4f}")
    print(f"Ложноотрицательные: {cm[1,0]}")

print("\nСРАВНЕНИЕ МОДЕЛЕЙ:")

false_negatives = {}
for name, y_pred in models.items():
    cm = confusion_matrix(y_test, y_pred)
    false_negatives[name] = cm[1,0]

for name, fn in sorted(false_negatives.items(), key=lambda x: x[1]):
    print(f"{name}: {fn} ложноотрицательных")

best_model = min(false_negatives.items(), key=lambda x: x[1])
print(f"\nЛУЧШАЯ МОДЕЛЬ: {best_model[0]} ({best_model[1]} FN)")
```

```
ИССЛЕДОВАНИЕ k-NN С РАЗНЫМ K:
Лучшая точность: k=5, точность=0.9649
Меньше всего ложноотрицательных: k=1, FN=6

МЕТРИКИ ДЛЯ ЗЛОКАЧЕСТВЕННЫХ ОПУХОЛЕЙ:

k-NN (k=1):
Матрица ошибок:
[[103   4]
 [  6  58]]
Precision: 0.9355, Recall: 0.9062, F1: 0.9206
Ложноотрицательные: 6

Decision Tree:
Матрица ошибок:
[[100   7]
 [ 10  54]]
Precision: 0.8852, Recall: 0.8438, F1: 0.8640
Ложноотрицательные: 10

SVM:
Матрица ошибок:
[[107   0]
 [  7  57]]
Precision: 1.0000, Recall: 0.8906, F1: 0.9421
Ложноотрицательные: 7

СРАВНЕНИЕ МОДЕЛЕЙ:
k-NN (k=1): 6 ложноотрицательных
SVM: 7 ложноотрицательных
Decision Tree: 10 ложноотрицательных

ЛУЧШАЯ МОДЕЛЬ: k-NN (k=1) (6 FN)
```

Вывод: На практике сравнил работу нескольких алгоритмов классификации, таких как метод k-ближайших соседей (k-NN), деревья решений и метод опорных векторов (SVM). Научитлся подбирать гиперпараметры моделей и оценивать их влияние на результат.