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

Лабораторная работа №3
По дисциплине «ОМО»

Выполнил:
Студент 3-го курса
Группы АС-65
Грущинский Д.Д.
Проверил:
Крощенко А.А

Брест 2025

Цель: 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.

Задачи:

1. Загрузить датасет по варианту;
2. Разделить данные на обучающую и тестовую выборки;
3. Обучить на обучающей выборке три модели: k-NN, Decision Tree и SVM;
4. Для модели k-NN исследовать, как меняется качество при разном количестве соседей (k);
5. Оценить точность каждой модели на тестовой выборке;
6. Сравнить результаты, сделать выводы о применимости каждого метода для данного набора данных.

Вариант 2

Датасет: 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).

Ход работы

Код программы данной лабораторной работы:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import confusion_matrix, classification_report,
precision_recall_fscore_support
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import recall_score
data = pd.read_csv(r'reports\Грущинский\3\src\breast_cancer.csv')
label_encoder = LabelEncoder()
data['diagnosis'] = label_encoder.fit_transform(data['diagnosis'])
```

```
x, y = data.drop(['id', 'diagnosis'], axis=1, inplace=False),
data['diagnosis']
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
x_train, x_test, y_train, y_test = train_test_split(
    x_scaled, y, test_size=0.2, random_state=42, stratify=y
)
knn = KNeighborsClassifier()
dt = DecisionTreeClassifier(random_state=42)
svm = SVC(kernel='rbf', probability=True, random_state=42)
knn.fit(x_train, y_train)
dt.fit(x_train, y_train)
svm.fit(x_train, y_train)
k_range = range(1, 21)
recall_scores = []
for k in k_range:
    knn_k = KNeighborsClassifier(n_neighbors=k)
    knn_k.fit(x_train, y_train)
    y_pred = knn_k.predict(x_test)
    recall = recall_score(y_test, y_pred, pos_label=1)
    recall_scores.append(recall)
best_k = k_range[np.argmax(recall_scores)]
print(f'Лучшее k по recall: {best_k}')
knn_best = KNeighborsClassifier(n_neighbors=best_k)
knn_best.fit(x_train, y_train)
models = {
    'k-NN': knn_best,
    'Decision Tree': dt,
    'SVM': svm
}
for name, model in models.items():
    y_pred = model.predict(x_test)
    cm = confusion_matrix(y_test, y_pred)
    report = classification_report(y_test, y_pred, target_names=['Benign',
'Malignant'])
    precision, recall, f1, _ = precision_recall_fscore_support(y_test,
y_pred, pos_label=1)
    print(f"\n{name}:\nМатрица ошибок:\n{cm}\n")
    print(f"Репорт классификации:\n{report}")
    print(f"Precision (malignant): {precision}")
    print(f"Recall (malignant): {recall}")
    print(f"F1-score (malignant): {f1}")
recall_scores_for_malignant = {}
for name, model in models.items():
    y_pred = model.predict(x_test)
    recall_malignant = recall_score(y_test, y_pred, pos_label=1)
    recall_scores_for_malignant[name] = recall_malignant
best_model_name = max(recall_scores_for_malignant,
key=recall_scores_for_malignant.get)
best_recall = recall_scores_for_malignant[best_model_name]
print(f"Модель, максимально минимизирующая
ложноотрицательные:\n{best_model_name}")
print(f"Recall (злокачественная опухоль) = {best_recall:.2f}")
```

```
best_model = models[best_model_name]
y_pred_best = best_model.predict(x_test)
cm_best = confusion_matrix(y_test, y_pred_best)
print(f"\nМатрица ошибок для лучшей модели ({best_model_name}):\n{cm_best}")
```

Вывод программы:

```
Лучшее k по recall: 5

k-NN:
Матрица ошибок:
[[71  1]
 [ 4 38]]

Репорт классификации:
              precision    recall  f1-score   support

      Benign       0.95      0.99      0.97        72
   Malignant       0.97      0.90      0.94        42

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114

Precision (malignant): [0.94666667 0.97435897]
Recall (malignant): [0.98611111 0.9047619 ]
F1-score (malignant): [0.96598639 0.9382716 ]

Decision Tree:
Матрица ошибок:
[[68  4]
 [ 4 38]]

Репорт классификации:
              precision    recall  f1-score   support

      Benign       0.94      0.94      0.94        72
   Malignant       0.90      0.90      0.90        42

    accuracy                           0.93       114
   macro avg       0.92      0.92      0.92       114
weighted avg       0.93      0.93      0.93       114

Precision (malignant): [0.94444444 0.9047619 ]
Recall (malignant): [0.94444444 0.9047619 ]
```

```
F1-score (malignant): [0.94444444 0.9047619 ]

SVM:
Матрица ошибок:
[[72  0]
 [ 3 39]]

Репорт классификации:
              precision    recall  f1-score   support

      Benign       0.96      1.00      0.98        72
   Malignant       1.00      0.93      0.96        42

    accuracy                           0.97       114
   macro avg       0.98      0.96      0.97       114
weighted avg       0.97      0.97      0.97       114

Precision (malignant): [0.96 1.  ]
Recall (malignant): [1.         0.92857143]
F1-score (malignant): [0.97959184 0.96296296]
Модель, максимально минимизирующая ложноотрицательные:
SVM
Recall (злокачественная опухоль) = 0.93

Матрица ошибок для лучшей модели (SVM):
[[72  0]
 [ 3 39]]
```

Вывод: мы научились решать задачи классификации многими методами, строить матрицы ошибок, а также укрепили свои знания в области машинного обучения.