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»

Выполнила:
Студентка 3 курса
Группы АС-65
Сергиевич М.А.
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.

Ход работы

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.

Код программы (1):

```
import pandas as pd
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 recall_score, confusion_matrix

# 1. Загрузка данных и стандартизация
df = pd.read_csv('pima-indians-diabetes.csv', header=None)
df.columns = ['pregnancies', 'glucose', 'blood_pressure', 'skin_thickness',
              'insulin', 'bmi', 'diabetes_pedigree', 'age', 'outcome']
print("Датасет загружен. Размер:", df.shape)
print("Распределение классов:")
print(df['outcome'].value_counts())
# 2. Разделение выборки 70/30
X = df.drop('outcome', axis=1)
y = df['outcome']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)
# Стандартизация данных
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# 3. Обучение моделей с заданными параметрами
print("\nОбучение моделей...")
# k-NN с k=5
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train_scaled, y_train)
knn_pred = knn_model.predict(X_test_scaled)
knn_recall = recall_score(y_test, knn_pred)

# Decision Tree с max_depth=4
dt_model = DecisionTreeClassifier(max_depth=4, random_state=42)
dt_model.fit(X_train, y_train)
dt_pred = dt_model.predict(X_test)
```

```
dt_recall = recall_score(y_test, dt_pred)

# SVM с линейным ядром
svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train_scaled, y_train)
svm_pred = svm_model.predict(X_test_scaled)
svm_recall = recall_score(y_test, svm_pred)

# 4. Сравнение производительности по recall
print("\n" + "="*50)
print("СРАВНЕНИЕ МОДЕЛЕЙ ПО RECALL")
print("="*50)

results = {
    'k-NN (k=5)': knn_recall,
    'Decision Tree (max_depth=4)': dt_recall,
    'SVM (linear)': svm_recall
}

for model_name, recall in results.items():
    print(f"{model_name}: Recall = {recall:.4f}")

# Находим лучшую модель
best_model = max(results, key=results.get)
best_recall = results[best_model]

print(f"\nЛучшая модель: {best_model} с Recall = {best_recall:.4f}")

# 5. Обоснование для медицинской задачи
print("\n" + "="*60)
print("ОБОСНОВАНИЕ ВЫБОРА ДЛЯ МЕДИЦИНСКОЙ ЗАДАЧИ")
print("="*60)

print(f"""
Для медицинской задачи предсказания диабета наиболее важной метрикой
является RECALL (полнота), так как пропуск заболевания (False Negative)
гораздо опаснее ложной тревоги (False Positive).

РЕЗУЛЬТАТЫ:
- k-NN (k=5): Recall = {knn_recall:.4f}
- Decision Tree (max_depth=4): Recall = {dt_recall:.4f}
- SVM (linear): Recall = {svm_recall:.4f}

ВЫВОД:
Для данной медицинской задачи наиболее предпочтительной является
модель {best_model}, так как она обеспечивает наивысший показатель
recall ({best_recall:.4f}), что означает максимальное выявление
реальных случаев диабета и минимальное количество пропущенных
заболеваний.
""")
```

```
Датасет загружен. Размер: (768, 9)
Распределение классов:
outcome
0    500
1    268
Name: count, dtype: int64

Обучение моделей...

==================================================
СРАВНЕНИЕ МОДЕЛЕЙ ПО RECALL
==================================================
k-NN (k=5): Recall = 0.4938
Decision Tree (max_depth=4): Recall = 0.6049
SVM (linear): Recall = 0.4938

Лучшая модель: Decision Tree (max_depth=4) с Recall = 0.6049

============================================================
ОБОСНОВАНИЕ ВЫБОРА ДЛЯ МЕДИЦИНСКОЙ ЗАДАЧИ
============================================================

Для медицинской задачи предсказания диабета наиболее важной метрикой
является RECALL (полнота), так как пропуск заболевания (False Negative)
гораздо опаснее ложной тревоги (False Positive).

РЕЗУЛЬТАТЫ:
- k-NN (k=5): Recall = 0.4938
- Decision Tree (max_depth=4): Recall = 0.6049
- SVM (linear): Recall = 0.4938

ВЫВОД:
Для данной медицинской задачи наиболее предпочтительной является
модель Decision Tree (max_depth=4), так как она обеспечивает наивысший показатель
recall (0.6049), что означает максимальное выявление
реальных случаев диабета и минимальное количество пропущенных
заболеваний.
```

Вывод: Мы сравнили работу нескольких алгоритмов классификации, таких как: метод k-ближайших соседей (k-NN), деревья решений и метод опорных векторов(SVM). Также я научилась подбирать гиперпараметры моделей и оценивать их влияние на результат.