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

Лабораторная работа №3
По дисциплине: «ОМО»
Тема: «Сравнение классических методов классификации»

Выполнил:
Студент 3-го курса
Группы АС-66
Пекун М.С.
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.

Вариант 8
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.
• Seeds
• Классифицировать семена на три сорта пшеницы (Kama, Rosa, Canadian) на основе их геометрических параметров
• Задания:
1. Загрузите и стандартизируйте данные;
2. Разделите выборку на обучающую и тестовую;
3. Обучите три классификатора;
4. Сравните общую точность (accuracy) всех трех моделей;
5. Визуализируйте данные в 2D (например, с помощью PCA), раскрасив точки в соответствии с предсказаниями лучшей модели.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
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 accuracy_score
from sklearn.decomposition import PCA

# === 1. Загрузка данных ===
data_path = r"E:\БРГТУ 3
КУРС\ОМО\1\ml_as66\reports\Pekun\lab1\src\seeds_dataset.txt"

columns = [
    "area", "perimeter", "compactness", "length", "width",
    "asymmetry", "groove", "class"
]
```

```
df = pd.read_csv(data_path, sep=r"\s+", names=columns)

X = df.drop("class", axis=1)
y = df["class"]

# === 2. Стандартизация и разделение выборки ===
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


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

# === 3. Обучение моделей ===
models = {
    "DecisionTree": DecisionTreeClassifier(random_state=42),
    "SVM": SVC(kernel="rbf", random_state=42),
    "kNN": KNeighborsClassifier(n_neighbors=5)
}

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

# === 4. Анализ k-NN для разных k ===
k_values = range(1, 21)
knn_scores = []
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    knn_scores.append(accuracy_score(y_test, y_pred))

best_k = k_values[np.argmax(knn_scores)]
best_knn_acc = max(knn_scores)

# === 5. Итоги и визуализация ===
print("Точность моделей:")
print(f"DecisionTree: {results['DecisionTree']:.4f}")
print(f"SVM: {results['SVM']:.4f}")
print(f"kNN (best k={best_k}): {best_knn_acc:.4f}\n")
```

```
print("Точность k-NN при разных k:")
for i in [1, 3, 5, 7, 9, 11, 15]:
    print(f"k={i}: {knn_scores[i-1]:.4f}")

# === 6. Важность признаков для Decision Tree ===
importances = models["DecisionTree"].feature_importances_
feature_importance = pd.DataFrame({
    "feature": X.columns,
    "importance": importances
}).sort_values(by="importance", ascending=False)

print("\nТоп признаков по важности (DecisionTree):")
print(feature_importance.head(7).to_string(index=False))

# === 7. Лучшая модель ===
best_model_name = max(results, key=results.get)
best_model = models[best_model_name]
y_pred_best = best_model.predict(X_test)

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_test)

plt.figure(figsize=(6, 5))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y_pred_best, cmap="viridis")

plt.title(f"PCA визуализация предсказаний ({best_model_name})")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.colorbar(label="Класс предсказания")
plt.show()

# === 8. Вывод ===
print("\n=== Вывод ===")
print(f"Наилучшую точность ({results[best_model_name]:.4f}) показала модель
{best_model_name}.")
```

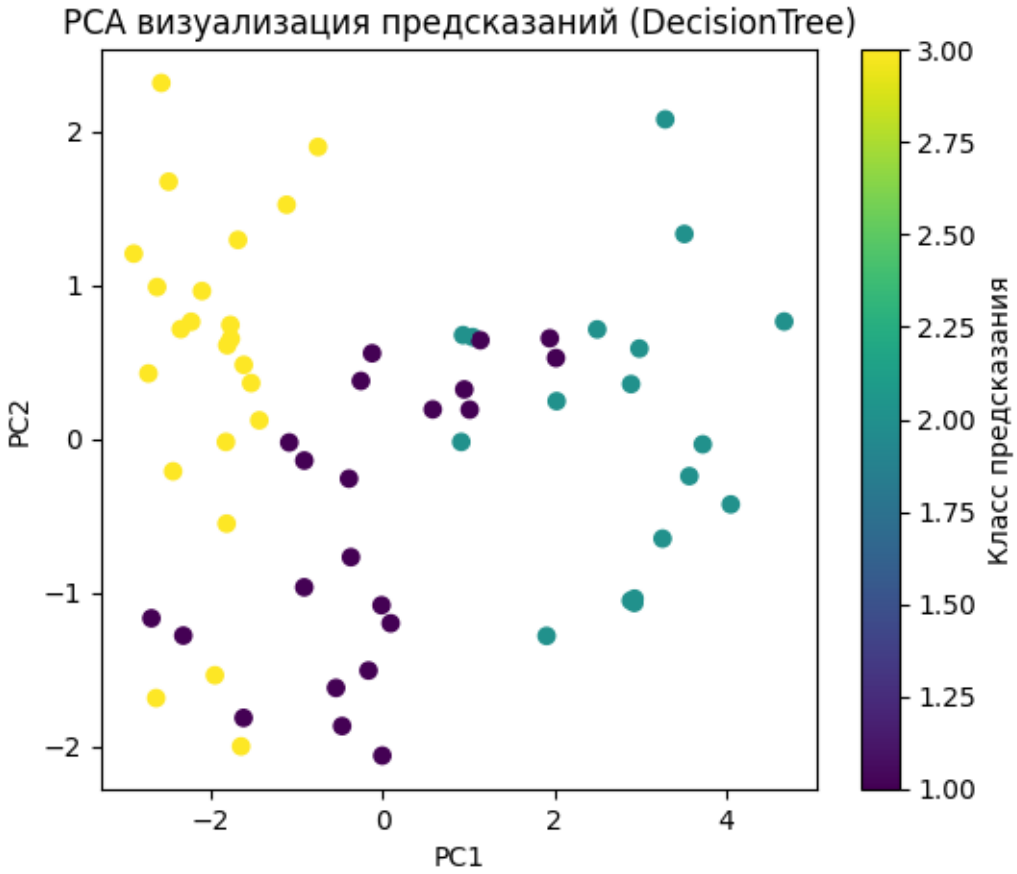
Результат:

```
Точность моделей:
DecisionTree: 0.8889
SVM: 0.8730
kNN (best k=1): 0.9048

Точность k-NN при разных k:
k=1: 0.9048
k=3: 0.8730
k=5: 0.8730
k=7: 0.8730
k=9: 0.8889
k=11: 0.8571
k=15: 0.8571

Топ признаков по важности (DecisionTree):
    feature  importance
     groove    0.491673
       area    0.345650
  asymmetry    0.087403
      width    0.056105
  perimeter    0.019169
compactness    0.000000
     length    0.000000

=== Вывод ===
Наилучшую точность (0.8889) показала модель DecisionTree.
```



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