

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  
Савинец М.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.

## Вариант 11

- Bank Marketing
- Предсказать, подпишется ли клиент на срочный вклад
- Задания:
  1. Загрузите данные и преобразуйте категориальные признаки;
  2. Разделите выборку;
  3. Обучите три классификатора;
  4. Сравните модели по F1-score из-за дисбаланса классов;
  5. Определите, какая модель предлагает лучший компромисс для выявления потенциальных клиентов.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import f1_score

data = pd.read_csv("bank.csv")
data.columns = data.columns.str.strip()

target_col = "deposit"

for col in data.select_dtypes(include=["object"]).columns:
    if col != target_col:
        data[col] = LabelEncoder().fit_transform(data[col])

X = data.drop(target_col, axis=1)
y = LabelEncoder().fit_transform(data[target_col])

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)

k_values = range(1, 21)
f1_scores_knn = []
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train_scaled, y_train)
    y_pred = knn.predict(X_test_scaled)
    f1_scores_knn.append(f1_score(y_test, y_pred))

best_k = k_values[f1_scores_knn.index(max(f1_scores_knn))]
best_knn_score = max(f1_scores_knn)

dt = DecisionTreeClassifier(random_state=42, max_depth=6)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
```

```

f1_dt = f1_score(y_test, y_pred_dt)

svm = SVC(kernel="rbf", C=1, gamma="scale")
svm.fit(X_train_scaled, y_train)
y_pred_svm = svm.predict(X_test_scaled)
f1_svm = f1_score(y_test, y_pred_svm)

print("\nСравнение моделей по F1-score:")
print(f"k-NN (k={best_k}): {best_knn_score:.3f}")
print(f"Decision Tree: {f1_dt:.3f}")
print(f"SVM: {f1_svm:.3f}")

best_model = max(
    [("k-NN", best_knn_score), ("Decision Tree", f1_dt), ("SVM", f1_svm)],
    key=lambda x: x[1]
)
print(f"\nЛучшая модель: {best_model[0]} (F1-score={best_model[1]:.3f})")

```

### Результат:

```

Сравнение моделей по F1-score:
k-NN (k=5): 0.766
Decision Tree: 0.808
SVM: 0.810

Лучшая модель: SVM (F1-score=0.810)

```

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.