

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  
Степанова Д. 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.

## Вариант 7

- Telco Customer Churn
- Предсказать, откажется ли клиент от услуг телеком-оператора
- **Задания:**

1. Загрузите данные, обработайте категориальные признаки и пропуски;

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

df = pd.read_csv("Telco-Customer-Churn.csv")

df["TotalCharges"] = pd.to_numeric(df["TotalCharges"], errors="coerce")
df["TotalCharges"] = df["TotalCharges"].fillna(df["TotalCharges"].median())

df["Churn"] = df["Churn"].map({"Yes": 1, "No": 0})

df = df.drop("customerID", axis=1)

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

X = pd.get_dummies(X, drop_first=True)

print(X.head())
```

...	PaperlessBilling_Yes	PaymentMethod_Credit card (automatic)	PaymentMe
...	True		False
...	False		False
...	True		False
...	False		False
...	True		False

2. Разделите данные на обучающую и тестовую выборки;

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42, stratify=y
)
```

3. Обучите модели k-NN, Decision Tree и SVM;

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

```

# k-NN
f1_scores = []
k_values = range(1, 51)

print("Подбор k для k-NN:")
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train_scaled, y_train)
    preds = knn.predict(X_test_scaled)
    score = f1_score(y_test, preds)
    f1_scores.append(score)

    if k % 10 == 0:
        print(f"k={k}, F1={score:.4f}")

best_f1 = max(f1_scores)
best_k = k_values[f1_scores.index(best_f1)]
print(f"\nЛучший k: {best_k}")
print(f"F1-score при k={best_k}: {best_f1:.4f}")

knn_best = KNeighborsClassifier(n_neighbors=best_k)
knn_best.fit(X_train_scaled, y_train)
pred_knn = knn_best.predict(X_test_scaled)
f1_knn = f1_score(y_test, pred_knn)

# Decision Tree
dt = DecisionTreeClassifier(max_depth=6, random_state=42)
dt.fit(X_train, y_train)
pred_dt = dt.predict(X_test)
f1_dt = f1_score(y_test, pred_dt)

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

```

```

Подбор k для k-NN:
k=10, F1=0.5248
k=20, F1=0.5636
k=30, F1=0.5853
k=40, F1=0.5890
k=50, F1=0.5907

Лучший k: 49
F1-score при k=49: 0.6013

```

#### 4. Сравните модели по метрике F1-score для класса "отток";

```

print("\nСравнение моделей")
print("=====")
print(f"F1-score k-NN (k={best_k}): {f1_knn:.4f}")
print(f"F1-score Decision Tree: {f1_dt:.4f}")
print(f"F1-score SVM: {f1_svm:.4f}")

```

```

Сравнение моделей
=====
F1-score k-NN (k=49): 0.6013
F1-score Decision Tree: 0.5290
F1-score SVM: 0.5586

```

5. Сделайте вывод о применимости каждого из методов для решения бизнес задачи по удержанию клиентов.

Лучшей моделью для решения задачи удержания является k-NN, так как она лучше всего выявляет клиентов, склонных к уходу. Модели Decision Tree и SVM показали качество ниже чем k-NN.

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