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
Шлейхер А. 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 10

- Adult Census Income
- Предсказать, превышает ли доход человека $50 тыс. в год
- **Задания:**
    1. Загрузите данные, обработайте пропуски и категориальные признаки;

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

df = pd.read_csv("adult.csv")

df.replace("?", pd.NA, inplace=True)
df.dropna(inplace=True)

df['income'] = df['income'].apply(lambda x: 1 if x.strip() == ">50K" else 0)

cat_cols = df.select_dtypes(include=['object']).columns
encoder = LabelEncoder()
for col in cat_cols:
    df[col] = encoder.fit_transform(df[col])

print(df.head(), "\n", "\n")
```

	age	workclass	fnlwgt	education	education.num	marital.status	...
1	82	2	132870	11	9	6	...
3	54	2	140359	5	4	0	...
4	41	2	264663	15	10	5	...
5	34	2	216864	11	9	0	...
6	38	2	150601	0	6	5	...

    2. Разделите данные на обучающую и тестовую выборки;

```
X = df.drop('income', axis=1)
y = df['income']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

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

3. Обучите k-NN, Decision Tree и SVM;

```
# лучшее k для knn
best_k = 1
best_precision = 0

for k in range(1, 21):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train_scaled, y_train)
    y_pred = knn.predict(X_test_scaled)
    precision = precision_score(y_test, y_pred)

    print(f"k={k}: precision={precision:.4f}")

    if precision > best_precision:
        best_precision = precision
        best_k = k

print("\nBest k:", best_k)
print(f"Best kNN precision: {best_precision:.4f}\n")

# k-NN
knn_best = KNeighborsClassifier(n_neighbors=best_k)
knn_best.fit(X_train_scaled, y_train)
precision_knn = precision_score(y_test, knn_best.predict(X_test_scaled))

# Decision Tree
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
precision_dt = precision_score(y_test, dt.predict(X_test))

# SVM
svm = SVC(kernel='rbf')
svm.fit(X_train_scaled, y_train)
precision_svm = precision_score(y_test, svm.predict(X_test_scaled))
```

Определили лучшее количество соседей для k-NN алгоритма:

```
k=1: precision=0.6029
k=2: precision=0.7207
k=3: precision=0.6535
k=4: precision=0.7335
k=5: precision=0.6848
k=6: precision=0.7364
k=7: precision=0.6955
k=8: precision=0.7452
k=9: precision=0.7103
k=10: precision=0.7464
k=11: precision=0.7168

Best k: 10
Best kNN precision: 0.7464
```

4. Сравните модели по метрике precision для класса ">50K";
5. Определите, какой алгоритм лучше всего идентифицирует людей с высоким доходом.

```
print("Сравнение моделей (precision >50K)")
print(f"k-NN (best k={best_k}):\t{precision_knn:.4f}")
print(f"Decision Tree:\t\t{precision_dt:.4f}")
print(f"SVM:\t\t\t{precision_svm:.4f}")

best_model_name = max(
    {"kNN": precision_knn, "DecisionTree": precision_dt, "SVM":
precision_svm},
    key=lambda k: {"kNN": precision_knn, "DecisionTree": precision_dt, "SVM":
precision_svm}[k]
)

print(f"\nЛучший алгоритм для идентификации людей с высоким доходом:
{best_model_name}")
```

```
Сравнение моделей (precision >50K)
k-NN (best k=10):       0.7464
Decision Tree:          0.6208
SVM:                    0.7666
```

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.