

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
Янчук А.Ю.
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 13

- Car Evaluation (UCI)
- Оценить безопасность автомобиля (классы: unacc, acc, good, vgood)
- Задания:
 1. Загрузите данные. Все признаки категориальные, поэтому используйте OrdinalEncoder или OneHotEncoder;
 2. Разделите выборку;
 3. Обучите три классификатора;
 4. Сравните общую точность моделей;
 5. Проанализируйте, какие признаки дерево решений посчитало наиболее важными (feature_importances_).

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

df = pd.read_csv("car_evaluation.csv", header=None)

df.columns = ["buying", "maint", "doors", "persons", "lug_boot", "safety",
              "class"]

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

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

categorical_features = X.columns.tolist()
encoder = OneHotEncoder(handle_unknown="ignore")

models = {
    "DecisionTree": DecisionTreeClassifier(random_state=42),
    "LogisticRegression": LogisticRegression(max_iter=2000, random_state=42),
    "RandomForest": RandomForestClassifier(random_state=42)
}

results = {}

for name, model in models.items():
    pipe = Pipeline(steps=[
        ("encoder", ColumnTransformer(
            transformers=[("cat", encoder, categorical_features)],
            remainder="drop"
        )),
        ("clf", model)
    ])
```

```

    ])

    pipe.fit(X_train, y_train)
    y_pred = pipe.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    results[name] = (pipe, acc)

print("Точность моделей:")
for name, (_, acc) in results.items():
    print(f"{name}: {acc:.4f}")

dt_model = results["DecisionTree"][0].named_steps["clf"]
ohe =
results["DecisionTree"][0].named_steps["encoder"].named_transformers_["cat"]

feature_names = ohe.get_feature_names_out(categorical_features)
importances = dt_model.feature_importances_

feat_imp = pd.DataFrame({
    "feature": feature_names,
    "importance": importances
}).sort_values(by="importance", ascending=False)

print("\nТоп-10 признаков по важности (DecisionTree):")
print(feat_imp.head(10))

```

Результат:

Точность моделей:

DecisionTree: 0.9740

LogisticRegression: 0.9017

RandomForest: 0.9682

Топ-10 признаков по важности (DecisionTree):

	feature	importance
12	persons_2	0.230072
19	safety_low	0.157644
17	lug_boot_small	0.082438
5	maint_low	0.075773
20	safety_med	0.063853
7	maint_vhigh	0.051521
15	lug_boot_big	0.047748
6	maint_med	0.046280
18	safety_high	0.041452
8	doors_2	0.041116

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.