

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Звіт
з лабораторної роботи №4
з дисципліни « Машинне навчання »

Виконав:
Студент групи ФІТ 3-16
Пархоменко Іван Дмитрович.

Київ 2025

```
# Імпортуємо необхідні бібліотеки
import pandas as pd
import numpy as np
```

```

from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns

# --- Завдання 1: Breast Cancer ---

# 1. Завантаження датасету
data_bc = load_breast_cancer()
X_bc = pd.DataFrame(data_bc.data, columns=data_bc.feature_names)
y_bc = pd.Series(data_bc.target)

# 2. Аналіз
head_bc = X_bc.head()
info_bc = X_bc.info()
missing_bc = X_bc.isnull().sum().sum()
shape_bc = X_bc.shape

# 3. Розділення на навчальну і тестову
X_train_bc, X_test_bc, y_train_bc, y_test_bc = train_test_split(X_bc, y_bc,
test_size=0.2, random_state=42)

# 4. Моделі
models_bc = {
    'Logistic Regression': (LogisticRegression(max_iter=10000), {'C': [0.1, 1, 10]}),
    'Decision Tree': (DecisionTreeClassifier(), {'max_depth': [3, 5, 10]}),
    'Random Forest': (RandomForestClassifier(), {'n_estimators': [50, 100], 'max_depth':
[5, 10]})
}

results_bc = {}

for name, (model, params) in models_bc.items():
    clf = GridSearchCV(model, params, cv=5)
    clf.fit(X_train_bc, y_train_bc)
    y_pred = clf.predict(X_test_bc)
    results_bc[name] = {
        'best_params': clf.best_params_,
        'accuracy': accuracy_score(y_test_bc, y_pred),
        'conf_matrix': confusion_matrix(y_test_bc, y_pred),
        'report': classification_report(y_test_bc, y_pred, output_dict=True)
    }

# --- Завдання 2: Titanic ---

# 1. Завантаження titanic.csv (використаємо seaborn як джерело)
df_titanic = sns.load_dataset("titanic")

```

2. Попередній аналіз

```
df_titanic_clean = df_titanic[['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch',
'fare']].dropna()
df_titanic_clean['sex'] = df_titanic_clean['sex'].map({'male': 0, 'female': 1})

X_titanic = df_titanic_clean.drop(columns='survived')
y_titanic = df_titanic_clean['survived']

X_train_t, X_test_t, y_train_t, y_test_t = train_test_split(X_titanic, y_titanic,
test_size=0.2, random_state=42)

models_titanic = {
    'Logistic Regression': (LogisticRegression(max_iter=1000), {'C': [0.01, 0.1, 1]}),
    'Decision Tree': (DecisionTreeClassifier(), {'max_depth': [3, 5, 7]}),
    'Random Forest': (RandomForestClassifier(), {'n_estimators': [50, 100], 'max_depth':
[5, 10]})
}

results_titanic = {}

for name, (model, params) in models_titanic.items():
    clf = GridSearchCV(model, params, cv=5)
    clf.fit(X_train_t, y_train_t)
    y_pred = clf.predict(X_test_t)
    results_titanic[name] = {
        'best_params': clf.best_params_,
        'accuracy': accuracy_score(y_test_t, y_pred),
        'conf_matrix': confusion_matrix(y_test_t, y_pred),
        'report': classification_report(y_test_t, y_pred, output_dict=True),
        'sample_prediction': clf.predict(X_test_t.head(10))
    }

results_bc, results_titanic
```