ДОДАТОК А

Код програми

```
from sklearn.neural_network import MLPRegressor
import numpy as np
import skfuzzy as fuzz


class FuzzyNonlinearBaggingRegressor(BaggingRegressor):
    def __init__(self,
                 base_estimator=None,
                 n_estimators=10,
                 max_samples=1.0,
                 max_features=1.0,
                 bootstrap=True,
                 bootstrap_features=False,
                 oob_score=False,
                 warm_start=False,
                 n_jobs=None,
                 random_state=None,
                 verbose=0):
        super().__init__(
            base_estimator,
            n_estimators,
            max_samples,
            max_features,
            bootstrap,
            bootstrap_features,
            oob_score,
            warm_start,
            n_jobs,
            random_state,
            verbose)

    def fit(self, X, y):
        super().fit(X, y)
        self.errors = [mean_squared_error(estimator.predict(X), y) for
estimator in self.estimators_]
        self.fuzzy_weights = fuzz.sigmf(np.array(self.errors),
np.mean(self.errors), np.std(self.errors))

    def predict(self, X):
        predictions = np.array([estimator.predict(X) for estimator in
self.estimators_])
        return np.average(predictions, axis=0, weights=self.fuzzy_weights)

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Завантаження даних
data = pd.read_csv(ScimagoJR Journals - 2023-FORMATED.csv)


# Передпоцессинг даних (нормалізація)
scaler = StandardScaler()
data = scaler.fit_transform(data)

# Розділення даних на навчальні та тестові набори
X_train, X_test, y_train, y_test = train_test_split(data.drop('target',
```

```
axis=1), data['target'], test_size=0.2)

# Створення та навчання моделі
model = FuzzyNonlinearBaggingRegressor(base_estimator=MLPRegressor(),
n_estimators=10)
model.fit(X_train, y_train)

# Прогнозування
y_pred = model.predict(X_test)

from sklearn.metrics import accuracy_score, recall_score, f1_score
import pandas as pd

# Розрахунок метрик
accuracy = accuracy_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Створення DataFrame
results = pd.DataFrame({
    '20% вибірки': [np.nan, np.nan, np.nan],
    '50% вибірки': [np.nan, np.nan, np.nan],
    '100% вибірки': [accuracy, recall, f1]
}, index=['SVM', 'Градієнтний бустінг', 'Нелінійний беггінг'])

# Виведення результатів
print(results)

import pandas as pd

# Створити пустий DataFrame
results = pd.DataFrame(columns=['Метод', 'Точність', 'Віддача', 'F1-
оцінка'])

# Обчислити метрики для кожної моделі та додати їх до DataFrame
for sample_size in ['20%', '50%', '100%']:
    for method in ['SVM', 'Градієнтний бустінг', 'Нелінійний беггінг']:
        # Обчислення метрики для нашої моделі
        accuracy = accuracy_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred)
        f1 = f1_score(y_test, y_pred)

        # Додаємо результати до DataFrame
        results = results.append({
            'Метод': method,
            'Точність': accuracy,
            'Віддача': recall,
            'F1-оцінка': f1
        }, ignore_index=True)

# Показати результати
print(results)
```

ДОДАТОК Б

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	97 с.
	Інші документи	
2.	Презентаційні матеріали	__ плакатів

.	.	Прізвище та ініціали.	Підп.	Дата			
Розробив		Іванов Є.О.			Нелінійний бегінг у системах обчислювального інтелекту	Шифр групи	Код напр./спец.
Перевірив		Бодянський Є.В.				СШІм-22-1	122
Н.контр.		Малєєва І.А.				ХНУРЕ кафедра ШІ	
Затв.		Філатов В.О.					