BİLGİSAYAR MÜHENDİSLİĞİ VERİ MADENCİLİĞİ DERSİ ÖDEV 2

EFE GİRGİN 19011095

```
import numpy as np
import pandas as pd
from sklearn.model_selection import StratifiedKFold
from collections import Counter




# Euclidean uzaklık hesaplama fonksiyonu
def euclidean_distance(a, b):
    return sum((e1 - e2) ** 2 for e1, e2 in zip(a, b)) ** 0.5

# Cosine uzaklık hesaplama fonksiyonu
def cosine_distance(a, b):
    dot_product = sum(e1 * e2 for e1, e2 in zip(a, b))
    magnitude_a = sum(e1 ** 2 for e1 in a) ** 0.5
    magnitude_b = sum(e2 ** 2 for e2 in b) ** 0.5
    if not magnitude_a or not magnitude_b:
        return 1
    return 1 - dot_product / (magnitude_a * magnitude_b)

# k-NN algoritmasının uygulanması
def knn(data, test_point, k, distance_func):
    distances = [(euclidean_distance(test_point, row[:-1]),
row[-1]) for row in data] if distance_func == 'euclidean' \
        else [(cosine_distance(test_point, row[:-1]), row[-
1]) for row in data]
    k_nearest = sorted(distances, key=lambda x: x[0])[:k]
    labels = [label for _, label in k_nearest]
    most_common = Counter(labels).most_common(1)
    return most_common[0][0]

# Cross-validation ve k-NN performans değerlendirme
def cross_validate_knn(X, y, k_values, distance_func):
    skf = StratifiedKFold(n_splits=5)
    scores = {k: [] for k in k_values}

    for train_index, test_index in skf.split(X, y):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]
        data = np.column_stack((X_train, y_train))

        for k in k_values:
            correct = sum(knn(data, test_row, k,
distance_func) == true_label for test_row, true_label in
zip(X_test, y_test))
            accuracy = correct / len(y_test)
            scores[k].append(accuracy)

    for k in k_values:
        print(f"Distance: {distance_func}, k={k}, Accuracy:
{np.mean(scores[k]) * 100:.2f}%")

# Veri kümesini yükleyin ve hazırlayın
iris =
pd.read_csv("C:\\Users\\MONSTER\\Desktop\\odev2\\iris.csv")
X = iris.drop('Species', axis=1).values
y = iris['Species'].values

k_values = [1, 3, 5, 9, 15]

# Euclidean uzaklık ile cross validation
cross_validate_knn(X, y, k_values, 'euclidean')

# Cosine uzaklık ile cross validation
cross_validate_knn(X, y, k_values, 'cosine')
```

```
Distance: euclidean, k=1, Accuracy:
87.33%
Distance: euclidean, k=3, Accuracy:
87.33%
Distance: euclidean, k=5, Accuracy:
87.33%
Distance: euclidean, k=9, Accuracy:
86.67%
Distance: euclidean, k=15, Accuracy:
86.67%
Distance: cosine, k=1, Accuracy: 95.33%
Distance: cosine, k=3, Accuracy: 93.33%
Distance: cosine, k=5, Accuracy: 90.00%
Distance: cosine, k=9, Accuracy: 86.67%
Distance: cosine, k=15, Accuracy: 82.67%
```