**IFT 511:** Analyzing Big Data

**Instructor:** Professor Asmaa Elbadrawy

**Project Step 4 -** Model Building and Evaluation

**Due:** April 30, 2024

**ABD Project Team 7**

Anagha Nitin Navale

Harsh Ketan Khona

Jayesh Ramesh Borkar

Stuti Pandey

**Code:**

from sklearn.preprocessing import StandardScaler

from sklearn.metrics.pairwise import cosine\_similarity, manhattan\_distances, euclidean\_distances

from scipy.stats import pearsonr

import pandas as pd

import numpy as np

from sklearn.model\_selection import KFold

# Load the dataset

data = pd.read\_csv('Data\_After\_Transformation.csv')

# Select features

features = data[['Zip Code', 'Student Enrollment',

'State Wide Assessment Results - 2023 : ELA (English Language Arts) : ALL ENROLLED - Minimally Proficient(%)',

'State Wide Assessment Results - 2023 : ELA (English Language Arts) : ALL ENROLLED - Partially Proficient(%)',

'State Wide Assessment Results - 2023 : ELA (English Language Arts) : ALL ENROLLED - Proficient(%)',

'State Wide Assessment Results - 2023 : ELA (English Language Arts) : ALL ENROLLED - Highly Proficient(%)',

'State Wide Assessment Results - 2023 All:MATH : ALL ENROLLED - Minimally Proficient(%)',

'State Wide Assessment Results - 2023 All:MATH : ALL ENROLLED - Partially Proficient(%)',

'State Wide Assessment Results - 2023 All:MATH : ALL ENROLLED - Proficient(%)',

'State Wide Assessment Results - 2023 All:MATH : ALL ENROLLED - Highly Proficient(%)',

'End of Year Promotion (%)']].values

# Normalize features

scaler = StandardScaler()

features\_scaled = scaler.fit\_transform(features)

# Function to compute similarity scores using various measures

def compute\_similarity\_scores(features, target\_index):

target\_features = features[target\_index]

similarity\_scores = {}

# Cosine similarity

cosine\_similarities = cosine\_similarity([target\_features], features)[0]

similarity\_scores['Cosine Similarity'] = cosine\_similarities

# Euclidean distance

euclidean\_dists = euclidean\_distances([target\_features], features)[0]

similarity\_scores['Euclidean Distance'] = euclidean\_dists

# Manhattan distance

manhattan\_dists = manhattan\_distances([target\_features], features)[0]

similarity\_scores['Manhattan Distance'] = manhattan\_dists

return similarity\_scores

# Define the number of folds for cross-validation

k = 5

# Initialize KFold cross-validator

kf = KFold(n\_splits=k)

# Iterate over each fold

for fold, (train\_index, test\_index) in enumerate(kf.split(features\_scaled), 1):

print(f"\nFold {fold}:")

# Split data into training and testing sets for the current fold

X\_train, X\_test = features\_scaled[train\_index], features\_scaled[test\_index]

y\_test = data.iloc[test\_index]['School']

# Perform similarity matching and evaluation for each similarity measure

similarity\_scores = compute\_similarity\_scores(X\_train, 0) # Target index is assumed to be 0

# Print the name of the target school

target\_school\_name = data.iloc[test\_index[0]]['School']

print(f"\nTarget School: {target\_school\_name}")

# Print similar schools recommended by each measure

for measure, scores in similarity\_scores.items():

sorted\_indices = np.argsort(scores)

top\_similar\_schools = sorted\_indices[1:6] # Exclude the target school itself

print(f"\nSimilar schools using {measure}:")

for i, idx in enumerate(top\_similar\_schools, 1):

school\_name = data.iloc[idx]['School']

print(f"{i}. {school\_name.strip()}")

# Calculate average accuracy for each measure

print("\nAverage Accuracy for Each Measure:")

for measure in similarity\_scores.keys():

accuracies = []

for fold, (train\_index, test\_index) in enumerate(kf.split(features\_scaled), 1):

X\_train, X\_test = features\_scaled[train\_index], features\_scaled[test\_index]

y\_test = data.iloc[test\_index]['School']

similarity\_scores = compute\_similarity\_scores(X\_train, 0)

sorted\_indices = np.argsort(similarity\_scores[measure])

top\_similar\_schools = sorted\_indices[1:6]

correct\_predictions = sum(data.iloc[top\_similar\_schools]['School'].isin(y\_test))

accuracy = correct\_predictions / len(top\_similar\_schools)

accuracies.append(accuracy)

average\_accuracy = np.mean(accuracies)

print(f"{measure}: {average\_accuracy:.2f}")

**Code explanation:**

The code is intended to assess the performance of various similarity metrics when proposing similar schools based on specific attributes.

**Model Description:**

1. **Data Loading and Preprocessing:**

* The dataset is loaded from a CSV file containing transformed data.
* Relevant features are selected for similarity computation.

1. **Similarity Computation:**

* The features are normalized using StandardScaler to ensure uniformity in scale.
* Similarity scores are computed using three different measures: Cosine Similarity, Euclidean Distance, and Manhattan Distance.
* For each similarity measure, the similarity scores between a target school (assumed to be the first school in the dataset) and all other schools are calculated.

1. **Cross-Validation:**

* The dataset is divided into K folds for cross-validation.
* For each fold:
  + The data is split into training and testing sets.
  + Similarity scores are computed for the training set.
  + Similarity-based recommendations are made for the testing set using each similarity measure.
  + The top 5 similar schools recommended by each measure are printed.

1. **Evaluation:**

* The accuracy of each similarity measure is evaluated using K-fold cross-validation.
* For each measure, the average accuracy across all folds is calculated and printed.

1. **Model Performance:**

* The average accuracy for each similarity measure is computed and printed.
* This provides insights into how well each measure performs in recommending similar schools.

1. **Interpretation:**

* Based on the average accuracy values, the effectiveness of each similarity measure can be compared.
* A higher average accuracy indicates better performance in recommending similar schools.

**Screenshot:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Output:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

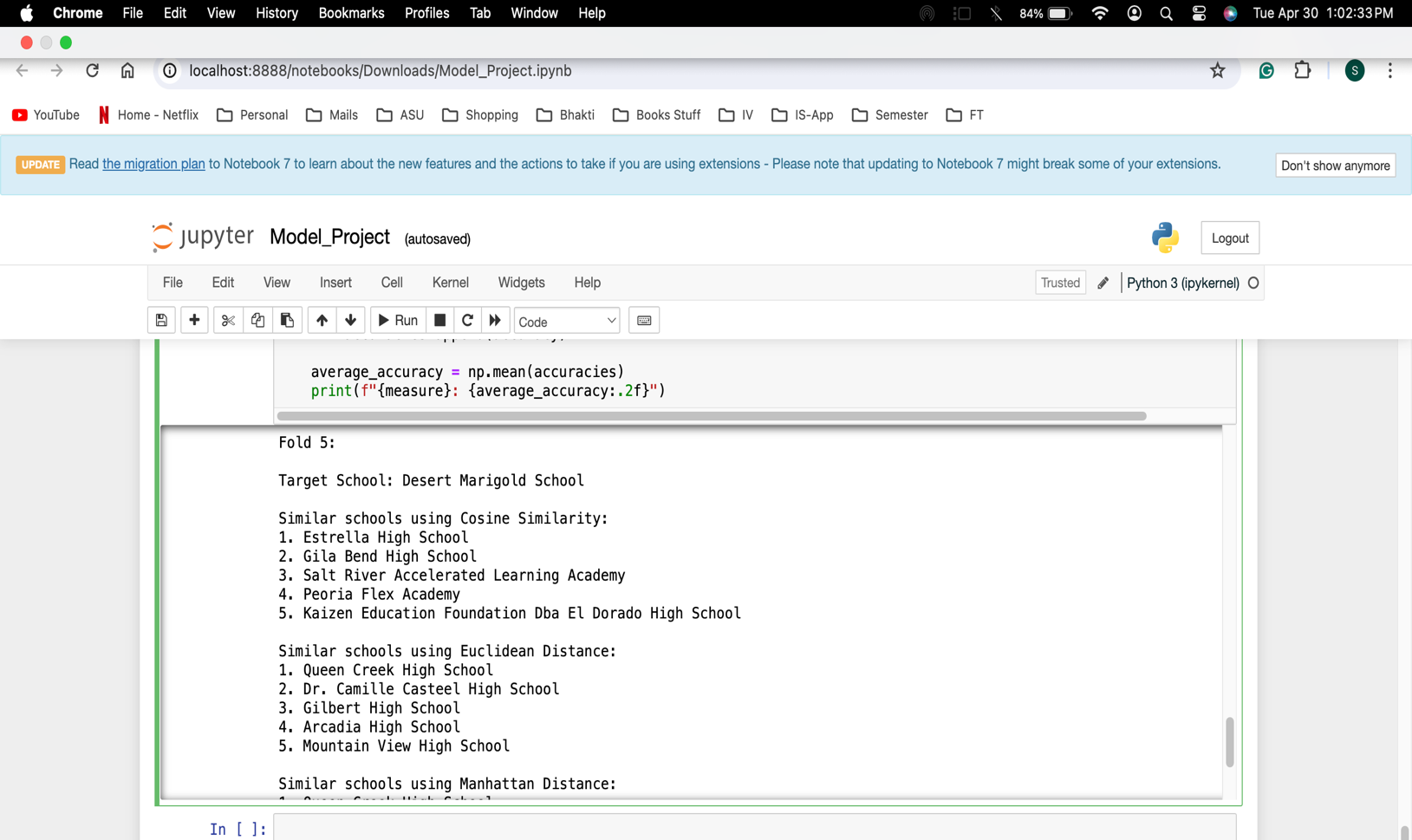
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

Fold 1:

Target School: Boulder Creek High School

Similar schools using Cosine Similarity:

1. Great Hearts Academies - Scottsdale Prep

2. Arizona Charter Academy

3. Tolleson Union High School

4. Camelback High School

5. Kino Junior High School

Similar schools using Euclidean Distance:

1. Asu Preparatory Academy-Polytechnic High School

2. Arizona Agribusiness & Equine Center - Estrella

3. Dysart High School

4. Desert Edge High School

5. Buckeye Union High School

Similar schools using Manhattan Distance:

1. Asu Preparatory Academy-Polytechnic High School

2. Arizona Agribusiness & Equine Center - Estrella

3. Basis Phoenix

4. Dysart High School

5. Desert Edge High School

Fold 2:

Target School: Mountain Ridge High School

Similar schools using Cosine Similarity:

1. Estrella High School

2. Gila Bend High School

3. Peoria High School

4. Cortez High School

5. Heritage Academy Laveen

Similar schools using Euclidean Distance:

1. Basis Phoenix Central Primary

2. Bostrom Alternative Center

3. Gilbert High School

4. East Valley Academ

5. Valley Vista High School

Similar schools using Manhattan Distance:

1. Basis Phoenix Central Primary

2. Gilbert High School

3. East Valley Academ

4. Bostrom Alternative Center

5. Basis Mesa

Fold 3:

Target School: Stapley Junior High School

Similar schools using Cosine Similarity:

1. Estrella High School

2. Gila Bend High School

3. Peoria High School

4. Cortez High School

5. Kaizen Education Foundation Dba El Dorado High School

Similar schools using Euclidean Distance:

1. Basis Phoenix Central Primary

2. Bostrom Alternative Center

3. Gilbert High School

4. Valley Vista High School

5. Mountain View High School

Similar schools using Manhattan Distance:

1. Basis Phoenix Central Primary

2. Gilbert High School

3. Bostrom Alternative Center

4. Sandra Day O'Connor High School

5. Mountain Ridge High School

Fold 4:

Target School: New School for the Arts

Similar schools using Cosine Similarity:

1. Estrella High School

2. Gila Bend High School

3. Cortez High School

4. Peoria Flex Academy

5. Kaizen Education Foundation Dba El Dorado High School

Similar schools using Euclidean Distance:

1. Gilbert High School

2. Arcadia High School

3. Valley Vista High School

4. Mountain View High School

5. Queen Creek Virtual Academy

Similar schools using Manhattan Distance:

1. Gilbert High School

2. Arcadia High School

3. Sandra Day O'Connor High School

4. Mountain Ridge High School

5. Sunrise Mountain High School

Fold 5:

Target School: Desert Marigold School

Similar schools using Cosine Similarity:

1. Estrella High School

2. Gila Bend High School

3. Salt River Accelerated Learning Academy

4. Peoria Flex Academy

5. Kaizen Education Foundation Dba El Dorado High School

Similar schools using Euclidean Distance:

1. Queen Creek High School

2. Dr. Camille Casteel High School

3. Gilbert High School

4. Arcadia High School

5. Mountain View High School

Similar schools using Manhattan Distance:

1. Queen Creek High School

2. Gilbert High School

3. Arcadia High School

4. Dr. Camille Casteel High School

5. Sandra Day O'Connor High School

Average Accuracy for Each Measure:

Cosine Similarity: 0.12

Euclidean Distance: 0.32

Manhattan Distance: 0.24​

**Result Explanation:**

In this case, Euclidean Distance has the highest average accuracy (0.32), followed by Manhattan Distance (0.24) and Cosine Similarity (0.12).

Based on the average accuracy values, the model's performance is better with Euclidean Distance compared to Cosine Similarity and Manhattan Distance.