**Procrastination**

🔧 Step 1: Install Required Libraries

Libraries installed via pip:

pip install pandas numpy matplotlib scikit-learn

---

📥 Step 2: Import Libraries

Python libraries used:

pandas, numpy for data processing

matplotlib.pyplot for visualization

KMeans from sklearn.cluster for clustering

datetime for date-time operations

---

📄 Step 3: Load and Preprocess the Data

CSV File: procrastination\_data.csv

Converted Columns to datetime:

TaskStartTime, TaskEndTime, Deadline, SubmissionTime

New Calculated Columns:

TaskDuration = Task end - Task start (in minutes)

Delay = Submission time - Deadline (in minutes)

📌 Sample Processed Data:

UserID TaskID TaskDuration Delay

1 101 120.0 mins -5.0

1 102 120.0 mins -10.0

2 103 120.0 mins 30.0

2 104 120.0 mins 15.0

---

📈 Step 4: Analyze Procrastination Patterns

Grouped by UserID, calculated:

Average Distraction Duration

Average Delay

Created user\_stats dataframe with:

AvgDistractionDuration, AvgDelay

📌 Example Output:

UserID AvgDistractionDuration AvgDelay

1 75.0 -7.5

2 75.0 22.5

---

📊 Step 5: Visualize Procrastination Patterns

Scatter plot: AvgDistractionDuration vs. AvgDelay

Helped identify patterns: users who delay more tend to have longer distraction durations.

---

📌 Step 6: Clustering Users

Used K-Means Clustering (k=3) on user\_stats

Added a new Cluster column to classify users based on procrastination behavior

📌 Clustered Data Visualization:

Color-coded scatter plot showing users grouped into 3 clusters

X-axis: Avg. Distraction Duration

Y-axis: Avg. Delay

Each cluster represents a different procrastination pattern

---

✅ Outcome

Clear understanding of procrastination behavior across users

Visual and statistical insights on how distraction duration relates to delay

Users categorized into behavioral clusters for further targeted analysis or intervention

# Step 1: Install Required Libraries

# Run the following command in your terminal or command prompt:

# pip install pandas numpy matplotlib scikit-learn

# Step 2: Import Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from datetime import datetime

# Step 3: Load and Preprocess the Data

# Load the dataset

data = pd.read\_csv('procrastination\_data.csv')

# Convert time columns to datetime

time\_columns = ['TaskStartTime', 'TaskEndTime', 'Deadline', 'SubmissionTime']

for col in time\_columns:

data[col] = pd.to\_datetime(data[col])

# Calculate task duration and delay

data['TaskDuration'] = (data['TaskEndTime'] - data['TaskStartTime']).dt.total\_seconds() / 60

data['Delay'] = (data['SubmissionTime'] - data['Deadline']).dt.total\_seconds() / 60

# Display the processed data

print("Processed Data:")

print(data.head())

# Step 4: Analyze Procrastination Patterns

# Calculate average distraction duration and delay per user

user\_stats = data.groupby('UserID').agg({

'DistractionDuration': 'mean',

'Delay': 'mean'

}).reset\_index()

# Rename columns for clarity

user\_stats.rename(columns={

'DistractionDuration': 'AvgDistractionDuration',

'Delay': 'AvgDelay'

}, inplace=True)

# Display user statistics

print("\nUser Statistics:")

print(user\_stats)

# Step 5: Visualize Procrastination Patterns

# Plot average distraction duration vs. average delay

plt.figure(figsize=(10, 6))

plt.scatter(user\_stats['AvgDistractionDuration'], user\_stats['AvgDelay'], c='blue', alpha=0.7)

plt.title('Procrastination Patterns: Distraction vs. Delay')

plt.xlabel('Average Distraction Duration (minutes)')

plt.ylabel('Average Delay (minutes)')

plt.grid(True)

plt.show()

# Step 6: Clustering Users Based on Procrastination Behavior

# Prepare data for clustering

X = user\_stats[['AvgDistracti

onDuration', 'AvgDelay']]

# Perform K-Means clustering

kmeans = KMeans(n\_clusters=3, random\_state=42)

user\_stats['Cluster'] = kmeans.fit\_predict(X)

# Display clustered user statistics

print("\nClustered User Statistics:")

print(user\_stats)

# Visualize clusters

plt.figure(figsize=(10, 6))

for cluster in user\_stats['Cluster'].unique():

cluster\_data = user\_stats[user\_stats['Cluster'] == cluster]

plt.scatter(cluster\_data['AvgDistractionDuration'], cluster\_data['AvgDelay'], label=f'Cluster {cluster}')

plt.title('User Clusters Based on Procrastination Behavior')

plt.xlabel('Average Distraction Duration (minutes)')

plt.ylabel('Average Delay (minutes)')

plt.legend()

plt.grid(True)

plt.show()

Data set

UserID TaskID TaskStartTime TaskEndTime DistractionDuration Deadline SubmissionTime TaskDuration Delay

0 1 101 2023-10-01 10:00:00 2023-10-01 12:00:00 30 2023-10-01 18:00:00 2023-10-01 17:55:00 120.0 -5.0

1 1 102 2023-10-02 09:00:00 2023-10-02 11:00:00 120 2023-10-02 18:00:00 2023-10-02 17:50:00 120.0 -10.0

2 2 103 2023-10-01 14:00:00 2023-10-01 16:00:00 60 2023-10-01 18:00:00 2023-10-01 18:30:00 120.0 30.0

3 2 104 2023-10-02 10:00:00 2023-10-02 12:00:00 90 2023-10-02 18:00:00 2023-10-02 18:15:00 120.0 15.0