Movie Recommendation System(Deliverable-2)

My movies dataset consists of users and the movies they have watched. Each user has mentioned five favorite movies. Since the dataset is textual and not visual, I applied **text-specific preprocessing techniques** instead of image-based ones.

Preprocessing Steps Applied:

- **Lowercasing**
  All movie titles were converted to lowercase to avoid mismatches due to capitalization (e.g., `"Inception"` vs `"inception"`).
- **Whitespace Removal**
  Leading and trailing whitespace was stripped from each movie title to ensure consistency.

```python
# 1. Load CSV
df = pd.read_csv('./dataset.csv')

# 2. Combine movie columns into a single list per user
movie_cols = ["Movie 1", "Movie 2", "Movie 3", "Movie 4", "Movie 5"]

# Normalize movies: lowercase, strip spaces
for col in movie_cols:
    df[col] = df[col].astype(str).str.strip().str.lower()
```

- **Typo Fixing and Normalization**
  A dictionary of common typos and alternate titles was created to standardize movie names.
  Example fixes:
    - `"muna bhi mbbs"` → `"munna bhai mbbs"`
    - `"top gun"` → `"top gun maverick"`

```python
# 3. Fixing typos from dataset
fixes = {
    "interstellars": "interstellar",
    "top gun": "top gun maverick",
    "muna bhi mbbs": "munna bhai mbbs",
    "bhjrngi bhai jan": "bajrangi bhaijaan",
    "money hiest korean": "money heist",
    "mission impossible series": "mission impossible",
    "the heirs": "heirs",
    "matrix": "the matrix",
    "cars": "cars (2006)",
    "tron": "tron legacy",
}

Tabnine | Edit | Test | Explain | Document
def fix_title(title):
    return fixes.get(title, title)

df[movie_cols] = df[movie_cols].applymap(fix_title)
```

- **Duplicate Removal**
  If a user listed the same movie multiple times, duplicates were removed using Python's `set` function.

```python
user_movies = {}
for i, row in df.iterrows():
    name = row['Name'].strip().lower()
    movies = [row[col] for col in movie_cols if pd.notna(row[col])]
    user_movies[name] = list(set(movies))  # remove duplicates per user
```

These steps were essential for preparing the dataset for clustering and recommendation without semantic conflicts.

Cross-Correlation Analysis

To mimic a **cross-correlation analysis**, I calculated the **pairwise similarity between users based on their movie choices**.