```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "cells": [
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "gehbO3Rinm4K",
    "outputId": "a79c6a75-0eea-4092-8a18-318e46d28692"
   },
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Enter a movie you like: Inception\n",
      "\n",
      "If you liked 'Inception', you might also enjoy: The Matrix\n"
     ]
    }
   ],
   "source": [
    "import pandas as pd\n",
    "from sklearn.feature_extraction.text import TfidfVectorizer\n",
    "from sklearn.metrics.pairwise import linear_kernel\n",
    "\n",
    "# Sample dataset (can be extended)\n",
```

```
      "data = {\n",
      "    'title': [\n",
      "        'The Matrix', 'Titanic', 'The Avengers', 'Interstellar',\n",
      "        'The Notebook', 'Inception', 'Toy Story', 'The Dark Knight'\n",
      "    ],\n",
      "    'genres': [\n",
      "        'Action Sci-Fi', 'Romance Drama', 'Action Adventure Sci-Fi', 'Sci-Fi Drama',\n",
      "        'Romance Drama', 'Action Sci-Fi Thriller', 'Animation Comedy Family', 'Action Crime
Drama'\n",
      "    ]\n",
      "}\n",
      "\n",
      "# Create DataFrame\n",
      "df = pd.DataFrame(data)\n",
      "\n",
      "# Vectorize genres using TF-IDF\n",
      "tfidf = TfidfVectorizer(stop_words='english')\n",
      "tfidf_matrix = tfidf.fit_transform(df['genres'])\n",
      "\n",
      "# Compute cosine similarity\n",
      "cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)\n",
      "\n",
      "# Function to return a single similar movie\n",
      "def recommend_one_movie(title, cosine_sim=cosine_sim):\n",
      "    idx = df[df['title'].str.lower() == title.lower()].index\n",
      "    if idx.empty:\n",
      "        return f\"Movie '{title}' not found in the dataset.\"\n",
      "\n",
      "    idx = idx[0]\n",
      "    sim_scores = list(enumerate(cosine_sim[idx]))\n",
      "    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)\n",
      "\n",
      "    # Skip the first one (it's the same movie), return the next one\n",
      "    for i in sim_scores[1:]:\n",
      "        recommended_idx = i[0]\n",
      "        return df['title'].iloc[recommended_idx]\n",
      "\n",
      "# Example usage\n",
      "user_input = input(\"Enter a movie you like: \")\n",
      "result = recommend_one_movie(user_input)\n",
      "print(f\"\\nIf you liked '{user_input}', you might also enjoy: {result}\")"
    ]
  }
```

```
  ]
}
```