

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "toc_visible": true
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "eoOch6ULgUua",
        "outputId": "2b02d731-0b5e-4f91-f29e-cee1f9a6654d"
      },
      "source": ""
    }
  ]
}
```

```

"outputs": [

{
    "output_type": "stream",
    "name": "stdout",
    "text": [
        "Recommended movies for User1\n",
        "Movie4  2\n",
        "Movie3  0\n",
        "Name: User2, dtype: int64\n"
    ]
}

],

"source": [

"\n",

"import pandas as pd\n",

"from sklearn.metrics.pairwise import cosine_similarity\n",

"\n",

"# Sample user-movie ratings\n",

"data = {\n",

"    'User1': [5, 3, 0, 0, 2],\n",

"    'User2': [4, 0, 0, 2, 3],\n",

"    'User3': [1, 1, 0, 5, 0],\n",

"    'User4': [1, 0, 4, 4, 0],\n",

"    'User5': [0, 1, 5, 4, 0],\n",

"}\n",

"movies = ['Movie1', 'Movie2', 'Movie3', 'Movie4', 'Movie5']\n",

"\n",

```

```

"df = pd.DataFrame(data, index=movies)\n",
"\n",
"# Transpose and compute similarity between users\n",
"user_sim = cosine_similarity(df.T)\n",
"user_sim_df = pd.DataFrame(user_sim, index=df.columns, columns=df.columns)\n",
"\n",
"# Recommend movies to User1 based on User2's similarity\n",
"target_user = 'User1'\n",
"similar_users = user_sim_df[target_user].sort_values(ascending=False)[1:]\n",
"most_similar = similar_users.index[0]\n",
"\n",
"# Recommend a movie\n",
"recommendations = df[df[target_user] == 0][most_similar].sort_values(ascending=False)\n",
"print(\"Recommended movies for\", target_user)\n",
"print(recommendations.head())"
]
},
{
"cell_type": "code",
"source": [],
"metadata": {
"id": "ikboHQs1guWH"
},
"execution_count": null,
"outputs": []
}
]

```

}