```
!pip uninstall -y numpy
!pip install numpy==1.23.5  # Compatible with scikit-surprise

!pip uninstall -y scikit-surprise
!pip install scikit-surprise  # Reinstall surprise for compatibility
```

```
Found existing installation: numpy 2.2.4
Uninstalling numpy-2.2.4:
  Successfully uninstalled numpy-2.2.4
Collecting numpy==1.23.5
  Downloading numpy-1.23.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Downloading numpy-1.23.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 49.3 MB/s eta 0:00:00
Installing collected packages: numpy
ERROR: pip's dependency resolver does not currently take into account all the package
xarray 2025.1.2 requires numpy>=1.24, but you have numpy 1.23.5 which is incompatible
treescope 0.1.9 requires numpy>=1.25.2, but you have numpy 1.23.5 which is incompatib
chex 0.1.89 requires numpy>=1.24.1, but you have numpy 1.23.5 which is incompatible.
pymc 5.21.1 requires numpy>=1.25.0, but you have numpy 1.23.5 which is incompatible.
albumentations 2.0.5 requires numpy>=1.24.4, but you have numpy 1.23.5 which is incom
blosc2 3.2.1 requires numpy>=1.26, but you have numpy 1.23.5 which is incompatible.
bigframes 1.42.0 requires numpy>=1.24.0, but you have numpy 1.23.5 which is incompati
albucore 0.0.23 requires numpy>=1.24.4, but you have numpy 1.23.5 which is incompatib
scikit-image 0.25.2 requires numpy>=1.24, but you have numpy 1.23.5 which is incompat
jax 0.5.2 requires numpy>=1.25, but you have numpy 1.23.5 which is incompatible.
imbalanced-learn 0.13.0 requires numpy<3,>=1.24.3, but you have numpy 1.23.5 which is
tensorflow 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 1.23.5 which is i
jaxlib 0.5.1 requires numpy>=1.25, but you have numpy 1.23.5 which is incompatible.
Successfully installed numpy-1.23.5
WARNING: The following packages were previously imported in this runtime:
  [numpy]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

```
Found existing installation: scikit-surprise 1.1.4
Uninstalling scikit-surprise-1.1.4:
  Successfully uninstalled scikit-surprise-1.1.4
Collecting scikit-surprise
  Using cached scikit_surprise-1.1.4-cp311-cp311-linux_x86_64.whl
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-package
Installing collected packages: scikit-surprise
ERROR: Operation cancelled by user
^C
```

```
import pandas as pd
import numpy as np
from surprise import SVD
from surprise import Dataset, Reader
from surprise.model_selection import cross_validate
```

```
url = "https://files.grouplens.org/datasets/movielens/ml-100k/u.data"

df = pd.read_csv(url, sep='\t', names=["user_id", "item_id", "rating", "timestamp"])
df = df[['user_id', 'item_id', 'rating']]  # Keep only relevant columns
df.head()
```

	user_id	item_id	rating
0	196	242	3
1	186	302	3
2	22	377	1
3	244	51	2
4	166	346	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
reader = Reader(rating_scale=(1, 5))  # Define rating scale (1 to 5)
data = Dataset.load_from_df(df, reader)  # Load dataset for Surprise


model = SVD()  # Initialize Singular Value Decomposition model
cross_validate(model, data, cv=5, verbose=True)  # Perform cross-validation
```

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9275  0.9360  0.9403  0.9367  0.9338  0.9349  0.0042
MAE (testset)     0.7316  0.7370  0.7413  0.7356  0.7368  0.7365  0.0031
Fit time          1.51    1.08    1.41    1.12    1.17    1.26    0.17
Test time         0.16    0.16    0.07    0.18    0.16    0.15    0.04
{'test_rmse': array([0.9275485 , 0.93596521, 0.94031885, 0.93673243, 0.93376633]),
 'test_mae': array([0.73157751, 0.73700602, 0.74128367, 0.73560312, 0.7367943 ]),
 'fit_time': (1.5103731155395508,
  1.0799105167388916,
  1.4051151275634766,
  1.1211130619049072,
  1.1729538440704346),
 'test_time': (0.15953397750854492,
  0.1647047996520996,
  0.0745539665222168,
  0.18459081649780273,
  0.1567833423614502)}
```

```
trainset = data.build_full_trainset()  # Use full dataset for training
model.fit(trainset)  # Train the model

# Predict rating for a specific user-item pair
user_id, item_id = 196, 302
predicted_rating = model.predict(user_id, item_id).est
print(f'Predicted Rating for User {user_id} and Item {item_id}: {predicted_rating:.
```

```
Predicted Rating for User 196 and Item 302: 3.91
```