# Problem Definition

Let’s say we have 8 users, and they have rated 8 different albums on a scale of 1 to 5. Note that not all users have rated all albums.

songData3 = {"Angelica": {"Blues Traveler": 3.5, "Broken Bells": 2.0, "Norah Jones": 4.5, "Phoenix": 5.0, "Slightly Stoopid": 1.5, "The Strokes": 2.5, "Vampire Weekend": 2.0},

"Bill":{"Blues Traveler": 2.0, "Broken Bells": 3.5, "Deadmau5": 4.0, "Phoenix": 2.0, "Slightly Stoopid": 3.5, "Vampire Weekend": 3.0},

"Chan": {"Blues Traveler": 5.0, "Broken Bells": 1.0, "Deadmau5": 1.0, "Norah Jones": 3.0, "Phoenix": 5, "Slightly Stoopid": 1.0},

"Dan": {"Blues Traveler": 3.0, "Broken Bells": 4.0, "Deadmau5": 4.5, "Phoenix": 3.0, "Slightly Stoopid": 4.5, "The Strokes": 4.0, "Vampire Weekend": 2.0},

"Hailey": {"Broken Bells": 4.0, "Deadmau5": 1.0, "Norah Jones": 4.0, "The Strokes": 4.0, "Vampire Weekend": 1.0},

"Jordyn": {"Broken Bells": 4.5, "Deadmau5": 4.0, "Norah Jones": 5.0, "Phoenix": 5.0, "Slightly Stoopid": 4.5, "The Strokes": 4.0, "Vampire Weekend": 4.0},

"Sam": {"Blues Traveler": 5.0, "Broken Bells": 2.0, "Norah Jones": 3.0, "Phoenix": 5.0, "Slightly Stoopid": 4.0, "The Strokes": 5.0},

"Veronica": {"Blues Traveler": 3.0, "Norah Jones": 5.0, "Phoenix": 4.0, "Slightly Stoopid": 2.5, "The Strokes": 3.0}

}

# Requirement for this Assignment

You have been provided with a framework for this assignment: “Assignment 3 – Framework.py”.

The Framework defines a Class called ***similarity***. You must use the class **as-is** with no changes. The class includes a class initialization method which takes two rating dictionaries ***ratingP*** and ***ratingQ*** as parameters. The class has two methods:

* ***minkowski*** which takes a single parameter ***r***, and returns the Minkowki Distance between the two dictionaries (that the Class object is instantiated with).
* ***pearson*** which takes no parameters, and returns the Pearson Correlation between the two dictionaries (that the Class object is instantiated with).

Given a userX, write code to find the NN (k=1) recommendations for userX based on the Euclidean Similarity measure. Pseudo-code has been provided to you in the framework. So, you essentially need to plug in appropriate code for steps 1 through 5 in the framework.

Since this is the same data as used in the Recommendation Systems Lecture, you should check your output against the Lecture for accuracy.

# Assignment Submission

Some things to keep in mind as you code:

* Make your code readable – for instance, use meaningful variable names and comments.
* Make your code elegant – for instance, balance the number of variables you introduce – too many or too few make your code difficult to debug, read, and maintain.
* Make your output readable and user-friendly

Once you have written up the script, save it as follows. Submit the script by uploading your python script. Note: upload the actual script – DO NOT attach a screenshot of the script!

<FirstName><LastName>Assignment3.py.

*[Example: HinaAroraAssignment3.py]*

The submitted script will be run **as-is** for grading. **I will be plugging in different users for userX to see if your code is giving me the correct recommendations.**

Points will be deducted for scripts that:

* are difficult to read/follow
* don’t compile/run
* don’t have all the various pieces of code required
* have hard-code values instead of using variables
* have logical errors
* don’t result in the expected output
* don’t have user-friendly output