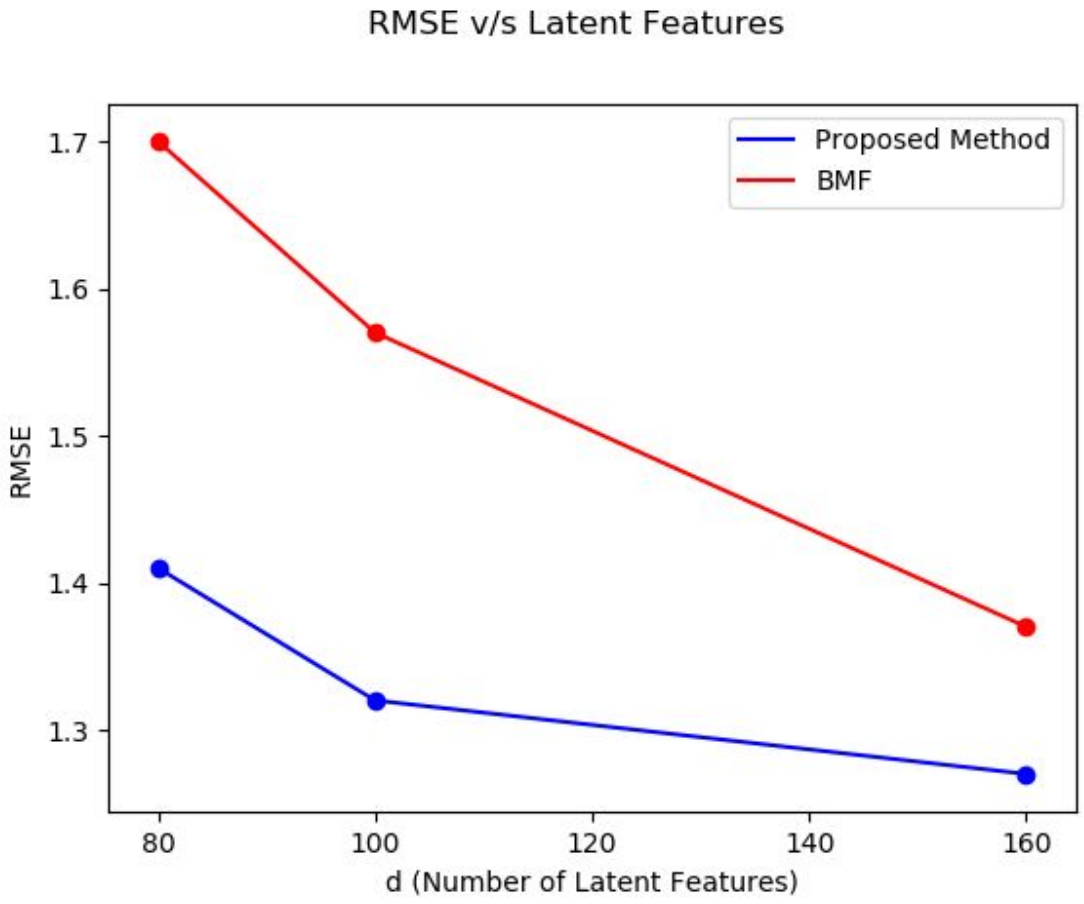
Experimental Results :

Task 1 - Ratings Prediction :

We used the proposed method and compared it with Bias Matrix Factorization with their RMF values.
In both the methods, error reduced as we learnt more latent vectors but in general, the proposed method, where regularization was done using the sound-sound similarity matrix, the error was lesser than BMF.
Here is the plot for the same :



(Graph of RMSE v/s Latent Features)

Task 2 - Top 10 (or n songs) Recommendation :

We also tried to find the songs similar to a given song in the projected Euclidean Space using cosine similarity values and here are some sample outputs for the same:

```
song_name = u"The Boy Looked At Johnny"
n = 10
l = song2vec.most_similar(song_name,topn=n)
print "Top 10 Songs Similar to: "+song_name
print "----------------------------------------"
i = 10
for song, sim in l:
    print i,song
    i -= 1
```

```
Top 10 Songs Similar to: The Boy Looked At Johnny
----------------------------------------
10 Radio America
9 Music When The Lights Go Out (Demo)
8 Babyshambles1
7 Cola Queen
6 The Man Who Would Be King
5 The Road To Ruin 2
4 Albion 1
3 Death On The Stairs (New Recording)
2 Playing Around With A Piano
1 Lots Of Songs
```

```
song_name = "Gang Of Gin"
n = 10
l = song2vec.most_similar(song_name,topn=n)
print "Top 10 Songs Similar to: "+song_name
print "----------------------------------------"
i = 10
for song, sim in l:
    print i,song
    i -= 1
```

```
Top 10 Songs Similar to: Gang Of Gin
----------------------------------------
10 Fuck Forever (Clean)
9 Lust Of The Libertines
8 Fuck Forever (Original)
7 Killamangiro
6 Do You Know Me
5 Á Rebours
4 What Katy Did Next
3 Fixing Up To Go
2 Beg Steal Or Borrow
1 Pipe Down
```

Thus we can accurately recommend Top N songs based on their similarities obtained from the proposed song2vec model.