**Notes on papers/literature**

InCarMusic: Context-Aware Music Recommendations in a Car

* Considering individual perceptions of the users about the influence of context on their decision
* Can only map one context at a time (never more than one)
* Make a rating
* Limited to what the user thinks their situation is
* Categories can be a bit too broad (countryside vs mountains vs urban)
* Uses **collaborative filtering** in building the RS. Explicit ratings given by a population used to predict ratings for items not yet evaluated by the users.
* Four steps:
  + Context factors relevance assessment
  + In-context acquisition of ratings
  + Context-aware rating prediction (classic matrix factorisation)
  + Context aware recommendation generation and visualisation for user
* Channels enables a user to quickly switch to a certain type of music whenever they want.
* Rating happens dynamically as the user is listening to the track (“in-context”)
  + This rating is immediately uploaded to the recommender server component and can be exploited for the computation of the next recommendations.
* No immediately ready to use application for collecting ratings
* Factors (driving style, landscape, etc) are assumed to be independent to get a traceable mathematical model
* No genre definition, but used a set defined in a certain paper.
* For the average user who did not like the tracks presented at all, there was no context that could change this attitude
* For users that did enjoy the music, you could find several contextual factors that had a positive influence on the ratings.
* Extension of matrix factorization (which I think I will be doing tbh because that’s what webtech used!!)
* Indicate the general deviation of the rating of a user for an item from the global average. Baseline is positive if it refers to a user that tends to rate higher than the average users’ population.
* MAE – you basically want this to be as low as possible!