In [ ]:

```
from IPython.display import Image
Image(filename='/content/drive/MyDrive/2560px-Banco_Santander_Logotipo.svg.png', width=1200, height=200)
```

Out[ ]:

Santander Product Recommendation

Can you pair products with people?

1. Introduction

Santander Bank is one of the North America’s top retails banks by deposits and a wholly owned subsidiary of one of the most respected banks in the world: Banco Santander.It’s parent company, Santander Group, serves more than 100 million customers in the United Kingdom, Latin America, and Europe.Santander Bank offers a lending hand to their customers through personalized product recommendation to support needs for a range of financial decisions. *Under their current system, a small number of Santander’s customers receive many recommendations while many others rarely see any, resulting in an uneven customer experience*.

With a more effective recommendation system in place, Santander can better meet the individual needs of all customers and ensure their satisfaction no matter where they are in life.

2. Business problem

Santander want us to **predict which products their existing customers will use in the next month based on their past behavior and that of similar customers.**

For this they have provided their Spanish customer’s data month-wise. It has information about customer’s demographics and features explaining customer relationships with Santander. It also has information about customer’s past product purchase history.We are given a dataset which consists of **1.5 years** customer behavior data. Approximately there **1 million customers** and the timeline for the data is from **January 2015 to May 2016**. There are approximately **13.5 million records** which gives us the information about the customer and the products purchased.

The objective is to **recommend top 7 products** to each customer. *For Santander, the purpose of this model is to recommend the most appropriate products to the customers, so that the conversion rates are good and customers get relevant products offered.*

By having a precise and strong recommendation system, the sales of the bank can be maximized. At the same time, the right products can also help the customers utilized their financial plan.

3. ML formulation of the business problem

1. As this problem deals with recommendation system, we can solve it by Collaborative Filtering, Content based filtering Matrix Factorization (User-User and Product-Product Similarity Matrix)

2. Given the number of users and products, creating User-User and Product-Product similarity matrix will increase time complexity and will also require better computational resource.

3. Alternatively, we have customer profile data along with past purchase behaviour using which we can create user vector and product vector. There are 24 different products for a given customer, which can be our class labels.

4. Given that we can proceed with Hybrid approach by making a linear combination of Content Based Filtering (for creating user vector) and Rule based approach which considers customers' purchase sequences over time as well as their purchase data for latest period.

5. We will pose this as **Multi class classification problem** where we predict the *probability of customer buying an individual product and*

recommened top seven.

4. Business Constraints

1. Multi-class probability estimates.
2. Probabilities along with class labels are useful in recommending most likely product first.
3. Cost of recommending irrelevant will affect customer experience.
4. Interpretability is partially important.
5. No strict latency concerns.

5. Data Overview

5.1 Source

Get data from https://www.kaggle.com/c/santander-product-recommendation/data

5.2 Dataset Information

1. Train dataset will be in train_ver2.csv and test_ver2.csv
2. Train Data
   - Total records ➡ 13,647,309
   - Total unique customers ➡ 956,645
   - Time Range ➡ Jan 2015 to May 2016
   - Total Columns ➡ 48 columns
   - First set of 24 columns are Customer Infomation as follows ➡
     A. fecha_dato : The table is partitioned for this column
     B. ncodpers : Customer code
     C. ind_empleado : Employee index
        - A active
        - B ex employed
        - F filial
        - N not employee
        - P pasive
     D. pais_residencia : Customer's Country residence
     E. sexo : Customer's sex
     F. age : Age
     G. fecha_alta : The date in which the customer became as the first holder of a contract in the bank
     H. ind_nuevo : New customer Index
        - 1 if the customer registered in the last 6 months.
     I. antiguedad Customer seniority (in months)
     J. indrel :
        - 1 (First/Primary)
        - 99 (Primary customer during the month but not at the end of the month)
     K. ult_fec_cli_1t : Last date as primary customer (if he isn't at the end of the month)
     L. indrel_1mes : Customer type at the beginning of the month
        - 1 (First/Primary customer)
        - 2 (co-owner)
        - P (Potential)
        - 3 (former primary)
        - 4 (former co-owner)
     M. tiprel_1mes : Customer relation type at the beginning of the month
        - A (active)
        - I (inactive)
        - P (former customer)
        - R (Potential)
     N. indresi : Residence index
        - S (Yes)

        - N (No) if the residence country is the same than the bank country
    O. indext : Foreigner index
        - S (Yes)
        - N (No) if the customer's birth country is different than the bank country
    P. conyuemp : Spouse index
        - 1 if the customer is spouse of an employee
    Q. canal_entrada : channel used by the customer to join
    R. indfall : Deceased index
        - N
        - S
    S. tipodom : Addres type
        - 1, primary address
    T. cod_prov : Province code (customer's address)
    U. nomprov : Province name
    V. ind_actividad_cliente : Activity index
        - 1, active customer
        - 0, inactive customer
    W. renta : Gross income of the household
    X. segmento : segmentation
        - 01 - VIP
        - 02 - Individuals
        - 03 - college graduated
- Second set of 24 columns are product data with boolean values 0/1 ➡
    A. ind_ahor_fin_ult1 : Saving Account
    B. ind_aval_fin_ult1 : Guarantees
    C. ind_cco_fin_ult1 : Current Accounts
    D. ind_cder_fin_ult1 : Derivada Account
    E. ind_cno_fin_ult1 : Payroll Account
    F. ind_ctju_fin_ult1 : Junior Account
    G. ind_ctma_fin_ult1 : Más particular Account
    H. ind_ctop_fin_ult1 : particular Account
    I. ind_ctpp_fin_ult1 : particular Plus Account
    J. ind_deco_fin_ult1 : Short-term deposits
    K. ind_deme_fin_ult1 : Medium-term deposits
    L. ind_dela_fin_ult1 : Long-term deposits
    M. ind_ecue_fin_ult1 : e-account
    N. ind_fond_fin_ult1 : Funds
    O. ind_hip_fin_ult1 : Mortgage
    P. ind_plan_fin_ult1 : Pensions
    Q. ind_pres_fin_ult1 : Loans
    R. ind_reca_fin_ult1 : Taxes
    S. ind_tjcr_fin_ult1 : Credit Card
    T. ind_valo_fin_ult1 : Securities
    U. ind_viv_fin_ult1 : Home Account
    V. ind_nomina_ult1 : Payroll
    W. ind_nom_pens_ult1 : Pensions
    X. ind_recibo_ult1 : Direct Debit

3. Test Data
    - Total Records ➡ 929,615
    - Total Unique Customers ➡ 929,615
    - Timeframe ➡ June 2016
    - Total Columns ➡ Same first set of 24 columns as train data (*which are customer information*)
4. Very important information mentioned by the company which will help us in deciding training and testing sample set. **You will predict what a customer will buy in addition to what they already had at 2016-05-28**
5. As we can see all the column names are in Spanish

5.3 Example Data Point

Train Data Point

fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	fecha_alta	ind_nuevo	antiguedad	indrel	ult_fec_cli_1t	indrel_1mes	tiprel_1m

2015-01-28	1375586	N	ES	H	35	42016	0	6	1	NaN	1	A
2015-01-28	1050611	N	ES	V	23	41131	0	35	1	NaN	1	I
2015-01-28	1050612	N	ES	V	23	41131	0	35	1	NaN	1	I
2015-01-28	1050613	N	ES	H	22	41131	0	35	1	NaN	1	I
2015-01-28	1050614	N	ES	V	23	41131	0	35	1	NaN	1	A

6. Performance Metrics

a. Evaluation Metric : **Mean Average Precision @ 7 (MAP@7)**

- Source : https://www.kaggle.com/c/santander-product-recommendation/overview/evaluation

b. Performance Metric : **Multi class log-loss**

7. Research Section

7.1 What is MAP@K ?

A. Sources

1. http://fastml.com/what-you-wanted-to-know-about-mean-average-precision/
2. https://makarandtapaswi.wordpress.com/2012/07/02/intuition-behind-average-precision-and-map/
3. https://www.kaggle.com/c/santander-product-recommendation/discussion/24908

B. My understanding

From the above blogs and discussion I got a good understanding of MAP@K and how the order of the correct recommended product is useful in getting high MAP score.

Let's understand using examples. Take 2 cases where we have to predict 7 products where 1 means correctly predicted product and 0 is miss. We have to evaluate MAP@5

1. Case 1 : [0, 1, 1, 0, 1, 1, 1]

> Precision@i = [0, 1/2, 2/3, 0, 3/5, 4/6, 5/7]
> Change in Recall = [0, 1/5, 1/5, 0, 1/5, 1/5, 1/5]
> MAP@5 = (0)*(0) + (1/2)*(1/5) + (2/3)*(1/5) + 0*0 + (3/5)*(1/5) + (4/6)*(1/5) + (5/7)*(1/5) = 0.62

2. Case 2 : [1, 1, 1, 1, 0, 0, 0]

> Precision@i = [1, 1, 1, 1, 0, 0, 0]
> Change in Recall = [1/5, 1/5, 1/5, 1/5, 0, 0, 0]
> MAP@5 = 1*(1/5) + (1)*(1/5) + (1)*(1/5) + (1)*(1/5) + 0*0 + 0*0 + 0*0 = 0.8

We can see that though we predicted more correct products in Case 1 than Case 2, yet we got more MAP@5 of Case 2 impling order of correctly predicted products matters.

C.Usage in this case study

The model is evaluated based on Mean Average Precision@7. The intuition behind this scoring metric is that :

- it pays to submit all 7 recommendations
- we are not penalized for bad guesses
- repeating predictions gives no benefit if we repeat predictions, we lose the opportunity to predict a different product
- order matters, i.e. we get more points if purchased item is at the top of the list of recommendations. So it’s better to submit more certain recommendations first, followed by recommendations we are less sure about.
- The maximum score we can get for any one row is 1, and the minimum is 0.

7.2 Recommendations Based on Purchase Patterns

A. Sources

1. http://www.ijmlc.org/papers/462-C015.pdf

B. My understanding

- This reseach paper is about Location based recommendation system on mobiles,which can be useful for location based business such as restaurants or tourist attractions.
- Assuming that a user is located near a set of available coupon deals, it aims to predict the correct category of the user's next purchase, so as to recommend the deals that are the most likely to be accepted.
- The paper is targeting at a problem with limited amount of user information. User profiles, preferences, location histories, item ratings are not available. Hence techniques such as content-based filtering, collaborative filtering, rule-based and hybrid approaches are difficult to apply.
- It is solely focused on category information of the transaction. Thus the purchase history in this context is just the category sequence.
- Sequential Pattern Analysis (SPA) is a classical approach where frequently occurring patterns extracted from subsequences of the given set of sequences are studied. Drawback of using SPA fo this problem is that SPA extracts patterns relating distant transactions as well. However, such patterns may not be relevant because coupon deals are time limited and recency is important.
- Instead, the approch followed in this paper uses consecutive subsequences and a threshold on length of subsequences to maintain recency. Also, in the subsequences the category information is replaced by the order of appearance, such that the purchase patterns extracted can better represent user behaviors.
- Conditional Probability model is proposed where model computes the conditional probability of a category matching the next purchase based on a given sequence.
- It is defined as probability of the event of a category 'c' matching the next purchase given the event of event of category sequence 's' being the purchase history. Denoted as P(N=c|H=s). For eg. let
  - x = occurrences of pattern "2"
  - y = occurrences of pattern "112"
  - probability of "2" matching next purchase given a purchase history "112" is
    P(N="2" | H="112") = y / x
- Research paper uses the training data extract purchase patterns and build conditional probability model. In the test data set, the user's transaction history is considered as purchase history and it is used by the conditional probability model to make recommendations. The actual next purchase considered as recommendation ground truth.
- Precision of recommendation is used as performance metrics. SPA was also performed on the same data, however the conditional model outperformed SPA.

C.Usage in this case study

- One of the key takeaway is that while recommending products, recency matters.
  It could be because :
  - user's prefrence has changed with time
  - product purchased in past might be in active or discontinued
- Recommending products conditionally depends upon puchase history and also the order/sequence of purchase
- Popular known approaches such as Content-based filtering, Collaborative filtering, Similarity matrics are not the only options as they require large diversified information to predict. We can still build a recommendation mechanism just by using user's purchase history (*and location if it is location based recommendation*) especially for mobiles.

7.3 Collaborative Filtering with Temporal Dynamics

A. Sources

1. https://www.cc.gatech.edu/~zha/CSE8801/CF/kdd-fp074-koren.pdf

B. My understanding

- In this paper, Yehuda Koren discusses about one step ahead solution of designing a recommender systems which is a function of time. That is, it models temporal dynamics. For this paper, they have reconditioned collaborative filtering recommendation approaches and evaluation is done on the movie rating dataset by Netflix.
- **Concept drift** : Data can change overtime. The analysis of such data needs to find the right balance between discounting temporary effects that have very low impact on future behavior,while capturing longer-term trends that reflect the inherent nature of the data.
  - Global Concept drift : Concept drift like emergence of new products or services changes the focus of customers globally. For eg. Seasonal changes in shopping pattern.
  - Localized concept drift : These are changes in user behaviour which are driven by localized factors. For eg. a change in the family structure can drastically change shopping patterns. Similiarly, individuals gradually change their taste in movies and music. Hence, for each customer we are looking at different types of concept drifts, each occurs at a distinct time frame and is driven towards a different direction.
- Challenge with modelling time changes at the level of each individual is that it significantly reduces the amount of available data for detecting such changes. Hence, we shouldn't simply abandon or underweight far in time user transactions. Instead, focus on modelling each point in the past allowing us to distinguish persistent signal that should be captured and noise that should be isolated from the longer term parts of the model

- As of now, two main parts of Collaborative Filtering are the *neighborhood approach(i.e. User-User Similarity & Item-Item Similarity) and* *latent factor models such as MF which creates user and item vectors.*

- To solve most common concept drift which is changing customer preferences over time, 3 major approaches are :
  - **Instance selection** : This discards instances that are less relevant to the current state of the system. For eg. time window where only recent instances are considered. However, this approach is reasonable when the time shift is abrupt, but less hence when time shift is gradual as it gives same significance to all instances within the considered time window, while completely discarding all other instances.
  - **Instance weighting** : Here instances are weighted based on their estimated relevance. For eg. a time decay function is used, under-weighting instances as they occur deeper into the past. Disadvantage is that just underweighting past actions loses too much signal along with the lost noise, which is detrimental given the scarcity of data per user.
  - **Ensemble learning** : It maintains a family of predictors which are weighted by their perceived relevance to the present time point and are then combined together to give final results. For e.g. predictors that are more successful on recent instances get higher weights. Disadvantge here is that having multiple models, each of which considers only a fraction of the total behavior may miss on global patterns that can be identified only when considering the full scope of user behavior.
- Guidelines adopted in the research paper for modeling drifting user preferences :
  - models that explain user behavior along the full extent of the time period, not only the present behavior
  - Multiple changing concepts should be captured. For eg. user-dependent changes, item-dependent changes,gradual changes, sudden changes.
  - Combining all seperate drifiting models within a single framework to eventually model interactions crossing users and items hence identifying higher level patterns.
  - Instead of extrapolating future temporal dynamics to estimate future changes in a user’s preferences. Goal is to capture past temporal patterns in order to isolate persistent signal from transient noise, this will intuitively help in predicting future behavior.
- Formulation of predicted rating with baseline predictors known to us is :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

  - *the second term is set of item factors relating item i to a factor vector yi belonging to R^f*
  - *The set R(u) contains the items rated by user u*
- In the above equation, temporal aspects of following terms are studied :
  - **user biases(b_u) change over time**
  - **Item biases (b_i) change over time**
  - **User preferences (p_u) change over time.**

- **Time changing baseline predictors**

$$b_{ui}(t) = \mu + b_i(t) + b_u(t)$$

  - *b_ui(t)* represents the baseline estimate for u’s rating on item 'i' at day 't'
    - *b_u(t) and b_i(t)* are real valued functions that change over time associated with user and item respectively
  - Note that b_u(t) requires finer time resolution as user effects can change on a daily basis, reflecting inconsistencies natural to customer behavior. Wherear, b_i(t) doesn't require finer time resolution we do not expect movie(or item) likeability to fluctuate on a daily basis, but rather to change over more extended periods.
- **Time-changing item biases** : This can be achieved by splitting the item biases into time-based bins. Final item bias as time-function looks like :

$$b_i(t) = b_i + b_{i,\mathrm{Bin}(t)}$$

- **Time-Changing user bias** : It would require finer resolution for users to detect very short lived temporal effects. Also, *we won't have enough ratings per user to produce reliable estimates for isolated bins*. Following different functions are considered in the paper for parameterizing temporal user behavior :
  - Linear function :

$$b_u^{(1)}(t) = b_u + \alpha_u \cdot \mathrm{dev}_u(t)$$

    - α_u = a single new parameter for each user
    - dev_u(t) = deviation of in day on which user rates a movie.

    It is given by formula :

    - $$\mathrm{dev}_u(t) = \mathrm{sign}(t - t_u) \cdot |t - t_u|^{\beta}$$
      - t_u = the mean date of rating for user u

- e |t − tu| measures the time distance (e.g., number of days) between dates t and tu
- β by cross validation

- Using Splines :

  - n_u = numer of ratings given by user u
  - ku time points are spaced uniformly across the dates of u's ratings as kernels that control the following formulation :

$$b_u^{(2)}(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}}$$

  - b_ti^u is learnt from the data
  - here k_u is set to n_u^0.25 letting it grow with the number of available ratings

- Till now we have ***addressed gradual concept drift***. However, there are **sudden drifts emerging as "spikes"** associated with a single day or session.To address this, a single parameter per user and day is assigned which absorbs the day-specific variability. This parameter is denoted by **b_u,t** . Hence our updated Linear and Spline equations are :

  - $$b_u^{(3)}(t) = b_u + \alpha_u \cdot \text{dev}_u(t) + b_{u,t}$$
  - $$b_u^{(4)}(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_{u,t}$$

- **Combining** both time-changing item biases and time-changing user biases(*using linear formulation here in the formula*), our baseline predictors as function of time looks like this :

  - $$b_{ui}(t) = \mu + b_u + \alpha_u \cdot \text{dev}_u(t) + b_{u,t} + b_i + b_{i,\text{Bin}(t)}$$
  - To learn *b_u, α_u, b_u,t, b_i and b_i,Bin(t)* , we solve the below optimization equation which minimizes the associated regularized squared error by using stochastic gradient descent.

$$\min \sum_{(u,i,t)\in\mathcal{K}} \left(r_{ui}(t) - \mu - b_u - \alpha_u \text{dev}_u(t) - b_{u,t} - b_i - b_{i,\text{Bin}(t)}\right)^2 + \lambda(b_u^2 + \alpha_u^2 + b_{u,t}^2 + b_i^2 + b_{i,\text{Bin}(t)}^2)$$

- Let's pin down all kinds of baseline predictors from static to the most temporal inclusive :

> - *static* no temporal effects: $b_{ui}(t) = \mu + b_u + b_i$.
> - *mov* accounting only to movie-related temporal effects: $b_{ui}(t) = \mu + b_u + b_i + b_{i,\text{Bin}(t)}$.
> - *linear* linear modeling of user biases: $b_{ui}(t) = \mu + b_u + \alpha_u \cdot \text{dev}_u(t) + b_i + b_{i,\text{Bin}(t)}$.
> - *spline* spline modeling of user biases: $b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_i + b_{i,\text{Bin}(t)}$.
> - *linear+* linear modeling of user biases and single day effect: $b_{ui}(t) = \mu + b_u + \alpha_u \cdot \text{dev}_u(t) + b_{u,t} + b_i + b_{i,\text{Bin}(t)}$.
> - *spline+* spline modeling of user biases and single day effect: $b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_u} e^{-\gamma|t-d_l|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\gamma|t-t_l^u|}} + b_{u,t} + b_i + b_{i,\text{Bin}(t)}$.
>
>   - Research paper includes RMSE scores of modelling done on all of the above mentioned baseline predictors and it is observed that *sudden changes in user biases, which are captured by the per-day parameters, are most significant*

- Another temporal effect is related to the **changing scale of user ratings**.For e.g. different users employ different rating scales and a single user can change his rating scale over time. Hence, *movie bias is not completely user-independent.* To address this, **a time-dependent scaling feature** is added to the baseline predictors, denoted by **c_u(t).**

- In addition to temporal nature of baseline predictors, **time also affects user's preferences.** Hence, p_u is also a function of time. This is written as :

  - $$p_{uk}(t) = p_{uk} + \alpha_{uk} \cdot \text{dev}_u(t) + p_{uk,t} \qquad k = 1, \ldots, f$$

    - *p_uk captures the stationary portion of the factor*
    - *α_uk ·dev_u(t) approximates a possible portion that changes linearly over time*

    - *p_uk,t absorbs local day-specific variability.*

- Till here, our **predicted rating as function of time** looks like :

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T \left(p_u(t) + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

    - Learning is performed by minimizing the associated squared error function on the training set using a regularized stochastic gradient descent algorithm.

- Lastly, we will **add temporal dynamics to neighborhood** that is user-item interaction part of CF as well. The static model, without any temporal dynamics is :

$$\hat{r}_{ui} = \mu + b_i + b_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} (r_{uj} - b_{uj}) w_{ij} + c_{ij}$$

where, the item-item weights w_ij and c_ij represent the adjustments we need to make to the predicted rating of item i, given a known rating of item j

    - While adding temporal dynamics to the baseline predictors in the above equation, nothing changes. We will just replace it by μ + bi(t) + bu(t).
    - Second term which captures information about user-item interaction requires a different strategy for temporal dynamics though.
    - Firstly, lets understand by an example that how user-item interaction has temporal nature. A user rating both items 'i' and 'j' high in a short time period, is a good indicator for relating them, hence higher the value of w_ij . However, if those two ratings are given five years apart, while the user's taste has considerably change, then this is less of an evidence of any relation between the items. Hence, in addition to time, these interactions are also user-dependent as some users have consistent taste and allow relating their longer term actions while others not.
    - To handle this lets parameterize the decaying relations between two items rated by user u, using exponential decay term e^(−βu·∆t) where ∆t = |t−t_j| and βu > 0 controls the user specific decay rate. Our prediction rule now looks like:

$$\begin{aligned}\hat{r}_{ui}(t) =& \mu + b_i(t) + b_u(t) + \qquad (16)\\ & |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{(j,t_j) \in \mathrm{R}(u)} e^{-\beta_u \cdot |t - t_j|} ((r_{uj} - b_{uj}) w_{ij} + c_{ij})\end{aligned}$$

- Now, we are at the end of the research paper. The **final optimization equation** that we are solving by minimizing the associated regularized squared error, to find parameters

    - b_i(t) = b_i + b_i,Bin(t) ➡ *item bias as function of time*
    - b_u(t) = b_u + α_u ·dev_u(t) + b_u,t ➡ *user bias as function of time*
    - βu,w_ij and c_ij ➡ *user-item interaction as function of time* , is

$$\begin{aligned}\sum_{(u,i,t) \in \mathcal{K}} & \Big(r_{ui}(t) - \mu - b_i - b_{i,\mathrm{Bin}(t)} - b_u - \alpha_u \mathrm{dev}_u(t) - b_{u,t} - \\ & |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{(j,t_j) \in \mathrm{R}(u)} e^{-\beta_u \cdot |t - t_j|} ((r_{uj} - b_{uj}) w_{ij} + c_{ij})\Big)^2 + \\ & \lambda(b_i^2 + b_{i,\mathrm{Bin}(t)}^2 + b_u^2 + \alpha_u^2 + b_{u,t}^2 + w_{ij}^2 + c_{ij}^2) \qquad (17)\end{aligned}$$

- **End note** : addressing temporal dynamics in the data can have a more significant impact on accuracy than designing more complex learning algorithms

C.Usage in this case study

1. This research paper, revolves around the temporal nature of user biases, item biases and user-item interaction, and takes into consideration data from way past to find important signal. It requires huge amount of data, so that we can have enough data to model time dependent functions for each user and item.
2. In this case study, we don't have this amount of data. However, I got a good understanding of taking temporal dynamic nature into account. Hence, I will create time lags as features.

7.4 Recommender Systems for Personalized Financial Advice - Aleksandra Chirkinas

A. Sources

A. Sources

- https://youtu.be/H8rWjyZpkGo
- https://www.zhaw.ch/storage/engineering/institute-zentren/iamp/upload/AI_conference_2019/Chirkina_Jeroense_RecommendersInProduction.pdf

B. My understanding

- This is a very informative video of a real-world end to end recommendation system developed by Aleksandra Chirkina, a data scientist @ InCube for J. Safra Sarasin Swiss Private Bank to provide **personalized investment advices to their clients.**

- Collaborative Filtering (MF and Latent features) based recommendation systems have proven efficient in retail, e-commerce and entertainment fields. It can also be used in financial institutions.

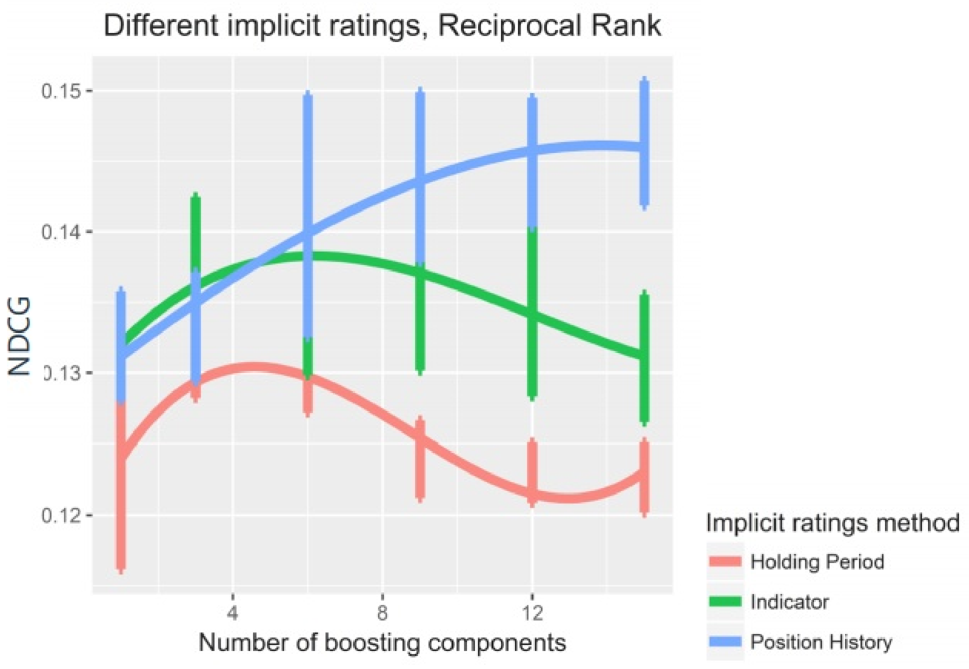
- **Advantages**:

    - It can save a great deal of time and efforts of customer relationship managers who are responsible for recommending relevant investment ideas to their clients.
    - With recommendation system in place, they can now focus on more clients and can also suggest investing ideas with higher acceptance probability.
    - It will increase client's engagement with the bank as now clients can also get involved in the suggestions made.
    - Earlier CRMs were able to focus only on their clients and had little or no knowledge about other clients portfolio. With RS in place, they can tackle each other's clients (in their absence) as they will now have information about what to recommend and why.
- Financial recommendations are *slightly different from recommendations in other fields*.

    - **Interpretability** is very important here.
    - There **isn't any detailed feedback** given by customers. Financial advice about investing in a company is either a hit or miss.
    - Feedback is very very important as our model can't recommend a company which was a miss earlier to the client again. Hence, **including feedback loop** in the pipeline is must.
    - The **penalty of making an error is high**
- *A note : at present, for simplicity let us consider an item to be a company.(it is not though, we will get a clear picture later)*

- The rating matrix for this use case is little different.

    - The only information that we have in the matrix is *whether a user 'i' has held shares of a company 'j' in the past or not.* . This **information is not enough to know about the user's experience**
    - Also, we are not sure **what missing values in the matrix mean**. Did the client didn't like the idea of investing in the company or were they not aware of the company at all, and hence missed out on a potential investment.
- **Handling the rating matrix (P)** : By training multiple models where the *elements p_ij of matrix P* was filled with different values backed by ideas like *holding period, plain indicator, trading activity etc.* data scientists confirmed that filling the matrix with the value of "trading activity within the holding period of shares of company j by user i" gave best results.



- Now, the **trading activity numbers are treated as confidence**
    - Matrix still uses 0's and 1's, where 0 means dissatisfied and 1 means satisfied. But now 0's and 1's are based on confidence.
    - If there is a high trading activity value in a cell, then we are really sure that there must be a 1 and if trading activity is low, we are not sure if there should be a 1 or 0.

    - We have imputed missing values in the matrix with 0 which means assigning a really low confidence value.
    - Now, perform matrix factorization on P to get user and product matrices $\widehat{\boldsymbol{P}} = \widehat{\boldsymbol{X}}_{[|U| \text{ x } k]} \cdot \widehat{\boldsymbol{Y}}^{\boldsymbol{T}}_{[|I| \text{ x } k]}$
    - such that the squared loss is minimized : $min(\sum_{i,j} c_{ij}(p_{ij} - \widehat{p}_{ij})^2 + \lambda(\left\|\widehat{X}\right\|_2^2 + \left\|\widehat{Y}\right\|_2^2))$
      where ;
    - Confidence weights
      $c_{ij} = 1 + \beta \log\left(1 + \frac{r_{ij}}{\epsilon}\right)$
    - $\widehat{\boldsymbol{X}}_{[|U| \text{ x } k]}, \widehat{\boldsymbol{Y}}_{[|I| \text{ x } k]}$ - 'latent features' matrices
      $\widehat{\boldsymbol{P}}$ - imputed ratings matrix

- If we see, the **confidence term of each cell is multiplied to the squared error**
    - If c_ij is low, the algorithm is free to impute any number as it won't contribute to the loss because of multiplication.
    - But if confidence is high, the loss contributed by the predicted number is high.
- Once the recommendation is made, **a complimentary model is incorporated** in the pipeline which explains the recommendation.
    - Explanations in the financial use case are very important as financial losses are involved.
    - Trust is developed by backing up recommendations with explanations.
    - As we know, *interpretation in CF* comes in the form **"users who agreed in the past tend to agree in future".**
    - This complementary model finds co-clusters in the matrix and gives an explanation about recommending a product to a user based on the similar purchase history of other users.
- *Addressing the assumption that we made earlier that the columns of the matrix are shares of a specific company*. <u>This is not true.</u>
    - In the final system, instead of a specific company, they use **Product Bins.**
    - The shares of different companies are grouped together into different asset categories.
    - The features of an asset category can be company name,currency, asset type, market, risk levels etc.
    - Hence assets/shares having the same features are grouped in the same asset category. **And the rating matrix is calculated at asset category level not on the individual asset level**
- **Problems** that can be **solved by Product Binning.**
    - Cold start for the products.
    - As shares of a company change over time, by combining them into bins we make them time invariant.
    - It also reduces dimensionality and sparsity of the matrix.
- However, **output of RS will be an asset category**. But we still need to come up with some companies from that asset category to recommend to the client.
    - This is done by applying multiple business filters to the category.
    - Now,let's say if a client is holding shares of almost all companies in the category, then we won't have anything to recommend. *What to do in this case?*
    - If we just move to the next category we will lose some edit value of the AI recommender system. Instead, we will **expand the same category by dropping some feature** let's say risk value and now will recommend companies from the expanded category.
- Using feedback given by the clients
    - In this RS feedback is binay. During training time feedback was simulated. Once the explicit feedback is given, the components of Loss functions are tweaked in such a way that
        - if the **feedback is negative**,
            - then set the confidence weight, c_ij to be the maximal possible value.
            - The true rating that the algorithm should predict i.e. p_ij is 0.
            - In order to minimize the loss function, p_ij_hat is forced to be zero
            - Hence, the corresponding asset category will be pushed back in the recommendation ranking list for the client who has given negative rating.
        - if the **feedback is positive**,
            - C_ij is set to maximal as this time we are really confident that clients are satisfied with the category.
            - The true rating p_ij should also be high.
            - Now to minimize the loss function, the algorithm will force p_ij_hat to be high as well.
            - Now, this category will be higher in the ranking list of the customer's recommendation batch. It enters the prediction loop for the next batch of recommendations.

C.Usage in this case study

1. Well this is a specific financial use case of Collaborative Filtering MF recommender system, it can be used as is if I am solving the case study using MF.
2. This work has given me a great idea of real world scenario based use case of recommendation system in Fintech.

7.5 Buy It Again: Modeling Repeat Purchase Recommendations

A. Sources

- https://dl.acm.org/doi/pdf/10.1145/3219819.3219891

B. My understanding

- This paper is about another aspect of recommendations that hasn't been explored much which is **Repeat Purchase Recommendation**.
- Given a customer has purchased a specific set of products *we have to predict if that customer is likely to repeat the purchase and if so, when is the right time to recommend* it to them. These products may serve as reminders and can provide immense value to customers.
- This requires to capture the *time correlations between subsequent repeat purchases of products*. The repeat purchase of such a product depends on the last purchase and how quickly customers run out of it.
- **Approaches discussed** :
  - to rank repeat purchase recommendations of a customer in the descending order of the number of times they repeat purchased the products.
    - However, there may be products which aren't timely relevant to customers anymore.
    - To address this problem, a time-decay based model where the repeat purchase score is decayed based on some pre-specified half-life. But *problem with this approach is that it will assign the highest score to a product right after the repeat purchase, which is not desirable.*
  - to assume that repeat purchasing of products by customers is a periodic phenomenon *to assume that both the above factors i.e. the number of times a customer repeat purchased a product and their repeat purchase periodicity play an important role.

- **Problem formulation** : Given a customer C_j has purchased a product A_i, 'k' times in the past with time intervals t_1, t_2, t_3... t_k. We would like to estimate the purchase probability density (P_Ai) at t = t_k+1. In conditional probability form. this is written as

> $$P_{A_i}(t_{k+1} = t | t_1, t_2, t_3, \ldots t_k)$$

  - Hidden assumptions around the above formulations are :
    - the purchase events of different products are independent of each other
    - the above equation can be decomposed into two possible components

    > $$P_{A_i}(t_{k+1}|t_1, t_2, t_3, \ldots..t_k) \approx Q(A_i) \times R_{A_i}(t_{k+1}|t_1, t_2, t_3, \ldots..t_k, A_i = 1)$$

      - *Q(Ai) is the repeat purchase probability of a customer buying a product a(k + 1)th time given that they have bought it k times*
      - *R_Ai is the time distribution of t_k+1, conditioned on the customer repurchasing that product; indicated by A_i = 1.*

- Models discussed :
  - **Repeat Customer Probability Model (RCP)** :
    - $$\mathrm{RCP}_{A_i} = \frac{\#\text{ customers who bought product } A_i \text{ more than once}}{\#\text{ customers who bought the product } A_i \text{ at least once}}$$
    - a simplifying assumption that is made P_Ai (tk+1|t1,t2,t3...tk) is approximately given by RCP_Ai
    - ignore the time factor altogether, i.e., we assume that R_Ai is a fixed constant for all products A_i
    - to ensure that the quality of repeat purchase recommendations are good threshold is enforced on RCP_Ai such that only the products that satisfy RCP_Ai > r_threshold are deemed repeat purchasable.
    - recommendations are then generated by considering all the repeat purchasable products previously bought by customers and ranking them in the descending order of their estimated probability density P_Ai(t) at a given time t.
    - the *RCP model is a simple probabilistic model* and treated as a baseline model that subsequent models should improve upon
  - **Aggregate Time Distribution Model (ATD)**:
    - This is time based model that uses aggregate repeat purchase behavior of a product across all repeat purchasing customers to determine its repeat purchase characteristics.
    - For this we have to determine the distribution of repeat purchase time intervals (t) of a product across all of its repeat purchasing customers.
    - It is found that log-normal distribution, had the best fit. Hence, **PDF of R_Ai(t)** is given by

    > $$R_{A_i}(t) = \ln\mathcal{N}(t; \bar{\mu}_i, \bar{\sigma}_i) = \frac{1}{\sqrt{2\pi}t\bar{\sigma}_i}\exp\left[-\frac{(\ln t - \bar{\mu}_i)^2}{2\bar{\sigma}_i^2}\right], t > 0.$$

    - Second assumption is that Q(Ai) is a fixed constant q for all products Ai at any given time.
    - Last two assumptions are same as RCP model.

    - **Poisson-Gamma Model (PG)**:
        - It is a NBD i.e. Negative Binomial Distribution model, which is based on following assumptions :
            - a customer's repeat purchases follow a homogeneous Poisson's process with **repeat purchase rate λ** i.e. they assume that *successive repeat purchases are not correlated with each other*
            - a gamma prior on λ assumimg that *λ across all customers follows a* ***Gamma distribution with shape α and rate β***
        - Customer's repeat purchase rate (λ) is calulated by combining the prior distribution with customer's own past purchase history as :

          $$\lambda_{A_i,C_j} = \frac{k + \alpha_{A_i}}{t + \beta_{A_i}}, t > 0$$

          where,
          ** α_Ai = shape &*
          ** β_Ai = rate parameters of the gamma prior of product Ai*
          ** k = number of purchases of product Ai bycustomer Cj*
          ** t = elapsed time between the first purchase of product Ai by customer Cj and the current time*
        - R_Ai is assumed to be a **poisson distribution with rate parameter λ** and the PMF is calculated as :

          $$R_{A_i,C_j}(t) = \sum_{m=1}^{\infty} \frac{\lambda^m_{A_i,C_j} exp(\lambda_{A_i,C_j})}{m!}, t > 0$$

          where,
          m is the number of expected future purchases
        - NBD model is a Bayesian model where the evidence is distributed as a Poisson and the prior on λ is a gamma prior. Hence, this is also called the Poisson-Gamma model (PG)
        - Second assumption that Q_Ai is a fixed constant q for all products Ai at any given time t.
        - Last two assumptions are same as RCP, ATD model.
    - **Modified Poisson-Gamma Model(MPG)**:
        - Issue with λ in PG model is that the estimated purchase rates right after the purchase is larger than right before the purchase. *This is not what we expect in repeat purchase recommendation.*
        - Also,it does not incorporate the product's time-independent repeat customer probability i.e. RCP into the model. To address these issues, following *modifications* are made :
            - **λ for the Modified-poisson process** is dependent on the last time the customer repeat purchased that product.
            - λ across all customers follows a Gamma distribution with shape α and rate β
        - Purchase rate for MPG model is :

          $$\lambda_{A_i,C_j} = \frac{k + \alpha_{A_i}}{t_{purch} + 2 * |t_{mean} - t| + \beta_{A_i}}, t > 0$$

          where,
          ** t_purch = elapsed time interval between the first and last purchase of product Ai by customer Cj*
          ** t = the elapsed time interval between the last purchase of product Ai by customer Cj and the current time*
          ** t_mean = the estimated mean repeat purchase time interval between successive purchases of product Ai by customer Cj*
        - Note that above equation is used when **t < 2 ∗t_mean**
            - when *t ≥ 2 ∗ t_mean, we calualte purchase rate using equation in PG model*
            - hence, **2 ∗ t_mean** as the **partition point after which the MPG model becomes the same as the PG model**
        - In MPG model, **Q(Ai) estimated using its RCP_Ai** i.e.

          > P_Ai(t_k+1|t_1,t_2,t_3...t_k)≈ Q(Ai) ≈ RCP_Ai

        - Last two assumptions are same as RCP, ATD & PG model.

- In **analytical comparisons** of the models, MPG outperformed all -
    - RCP model - which only uses the time independent aggregate repeat purchase behavior of customers will not consider recent purchase into account.
    - PG model - which uses the customer's own prior purchase rate for products
    - ATD model - which uses the aggregate distribution of repeat purchase time intervals across all its repeat purchasing customers + take into account that the most recent purchases as well
    - MPG model - which takes into account the customer's own posterior purchase rate after adjusting for the last purchase time and the product's RCP_Ai
- **Model Performance while training** :
    - A customer and their product purchase were considered as a repeat purchase in *test period (most recent one week)* iff the customer purchased a product in the *training period (y years before the test period)* and also purchased the same product sometime in the test period.

- Metrics used for comparing the performances are **Precision & Recall**. But incase of recommendation ranking is also important hence, a new standard metric to evaluate the quality of recommendations is used which is **Normalized Discounted Cumulative Gain (nDCG)**

- Results were:
  - all *subsequent models are better than the baseline RCP*
  - *RCP and ATD* models are *much worse* than both the PG and MPG models.
  - *MPG model is best overall* - reason for the superiority of the MPG model is that in *addition to the personalized signal*, it *combines the both time and time-independent signals* for generating recommendations while also ensuring that its time-model is adapted for the recommendations use case

- **Testing during production** is done using A/B testing in two phases. Evaluation metric was **CTR.**
  - Phase 1 : *The control did not have the BIA(Buy It Again) recommendations feature and the treatment showed the BIA recommendations feature using ATD model.*
    - Results : Using the ***BIA recommendations using ATD model,resulted in a 7.1% increase in the CTR*** for products on that page in treatment. This lift was statistically significant at a significane level of 0.01 ($p < 0.01$)
  - Phase 2 : *For this A/B test, control showed the BIA recommendations feature using the recommendations generated by the ATD model and treatment showed the BIA recommendations feature using the recommendations generated by the MPG model.*
    - Results : ***1.3% increase in CTR*** for products on the personalized recommendations page of the Amazon.com website in treatment. This lift was also statistically significant @ 0.05 ($p < 0.05$) and confirms the result from the offline experiments that the **MPG model is significantly better than the ATD model.**

C.Usage in this case study

1. This research paper is based on a statistical way of creating a recommender systems.
2. The case study usecase is considerably different from the work. However, it gave me an intuition of solving recommendation problem using probability (Bayesian probability) and known statistical distributions (such as Poisson,Gamma, NBD etc.)
3. Also, in future it will be very helpful as it solves a real-world e-commerce problem. And we can use known parameteric distributions for solving specific type of recommendation problems. E-commerce data largely follows, log normal distribution.

8. First Cut Solution

- As this is a recommendation system, we need to address **Cold Start problem**
- From above researches, I realized that for recommendation recency matters. And also in the case study it is mentioned that we have to predict "which" customer is going to buy "what new" product next month. This means we are mainly interested in recommending to the customers who bought the product in the first place. Hence, we can train on smaller data.
- **EDA** : We have both Categorical and Numerical features.
  - Remove row duplicates from data.
  - Perform Univariate and Bivariate analysis on the featuers to understand their distribution and correlation between them.
  - Outliers to be removed from the features
  - Missing values have to be imputed. For numerical features we can either mean/median (will decide after EDA) and for Categorical features if the missing values are large, we can create a seperate 'missing' category
    - Once, imputing is done, we will Label Encode, categorical features and Normalize numerical features.
- **Feature Selection** :
  - As we are treating products/items as features too, there is a possiblity that some products might have been discontinued overtime. Hence, removing those products will be benefical.
  - Even some features related customer informaation which are redundant, can be dropped.
- **Feature Engineering**:
  - With the help of the research done, I got an idea on how to use past purchase information of the cusstomer. For each customer, we can create each product lags (maybe upto last 6 months or even 12 months, need to decide on this) as features.
- Once we are able to express each **data point as a vector**, we will proceed for modelling.

- **Model Selection**
  - Based on the research work that I done, this case study can be solved either by using Collaborative Filtering MF or by Content Based filtering where I will create user and item vectors from given data about Customer Demographics and Product Information and solve it as multiclass classfication problem. I am choosing the later becasue of large size of data and computational problem.

  - However, I will still want perfrom SVD maybe on smaller set of data for model comparision

  - Create a random model with random labels as predicted values and a baseline model maybe Logistic Regression using OVR before working on Ensemble models such as XGBOOST for good baseline score.

```
In [ ]:
```