1. Business Problem

The problem to be solved in this project is how to create a model that predicts which products a bank's customer will buy next month given the available data on a bank's customers. The bank is essentially interested in implementing personalized product recommendations for all of its customers. Under their current system, a small number of the bank's customers receive many recommendations while most of their customers rarely see any. The bank is interested in having a more effective recommendation system in place so that it can better meet the individual needs of all its customers and ensure their satisfaction no matter where they are in life. This project is base on the Santander Product Recommendation Kaggle competition. The evaluation metric for the competition was the mean average precision at seven (MAP @7). Therefore the model should deliver a rank ordered list of up to 7 products for every customer in the test set.

2. Client

Santander Group is a multinational banking conglomerate based in Spain. Its chief holding is Banco Santander the largest bank in Spain and after reviewing the customer data it seems this data is at the least, based on the customer data of Banco Santander. The variety of products to be recommended in this data seem to be typical of a retail banking operation. The products include savings accounts, certificates of deposits, credit cards, mortgages and various other types of accounts. It's worth mention that although they named this a product recommendation challenge its specifically a product prediction challenge and although these two tasks do have plenty of overlap in their approach and implementation there are distinct analytical approaches taken for each.

3. Description Of Data

The data was available as two csv files labeled train and test. Each row is for one customer for a certain month. The test file is just the most recent month with the value for all the products missing. Train file is for the previous 7 month with a column for each product filled with binary values of whether each customer has it or not in that month. There are 24 different product columns and 22 columns that offer data about the customers.

```
train.head()
```

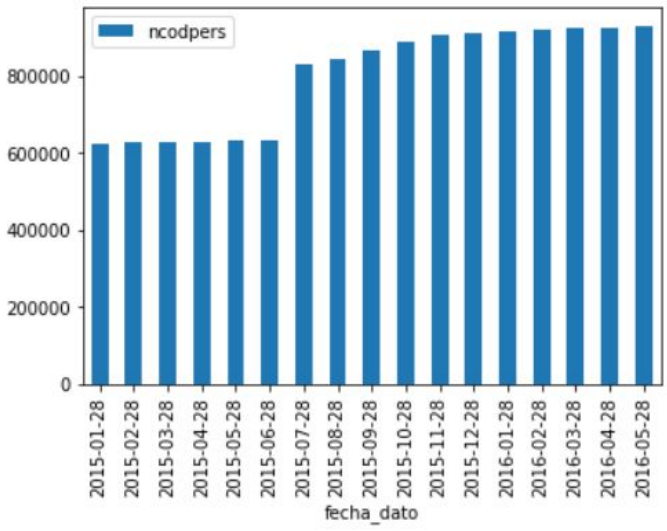
	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	fecha_alta	ind_nuevo	antiguedad	indrel	...	ind_hip_fin_ult1	ind_plan_fin_ult1	ind_pres
0	2015-01-28	1375586	N	ES	H	35	2015-01-12	0.0	6	1.0	...	0	0	
1	2015-01-28	1050611	N	ES	V	23	2012-08-10	0.0	35	1.0	...	0	0	
2	2015-01-28	1050612	N	ES	V	23	2012-08-10	0.0	35	1.0	...	0	0	
3	2015-01-28	1050613	N	ES	H	22	2012-08-10	0.0	35	1.0	...	0	0	
4	2015-01-28	1050614	N	ES	V	23	2012-08-10	0.0	35	1.0	...	0	0	

5 rows × 48 columns

There were plenty of missing values in this data set. I approached the data column by columns to decide how I should deal with this issue. First I noticed what looked to be certain rows that just had multiple missing values across many columns and I just dropped all those rows. For the registration method column and the column which flags for if the customer is a spouse of an employee I imputed the missing values as unknown since most values in these columns where missing. There were 2 columns for the province of the customer, one with the full name and one with an abbreviation, so I just dropped the abbreviation column and imputed the most common province name which happened to be Madrid. There was column for the type of customer with 4 numerical codes for each type and I decided to just replace these with strings for the name of the customer type. Then I imputed the missing values here with what was by far the most common type - primary. For the household income there was pretty good option for smartly imputing values which was to take the median income of the province the customer resided in. Similarly for the account type column I imputed missing values based on the income bracket of the customer. For gender and customer relation at end of month there weren't too many missing values so I imputed those with the most common value. Two of the product columns had some missing values so I just imputing those as 0 ie. the customer did not have the product. And for the customer seniority there were some negative values which I just replaced with the median.

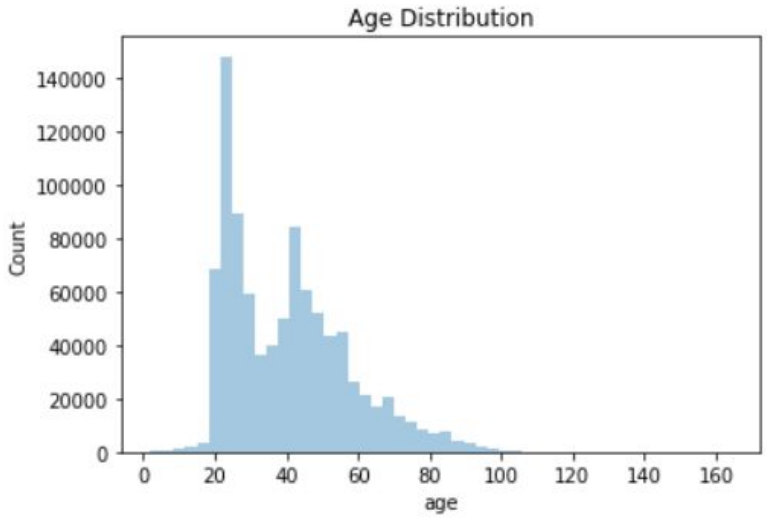
4. Initial Findings

Since the data was sorted by month, each date being the 28th of months in sequential order I wanted to see how the number of customers changed over time and how months of data there were for different customers.



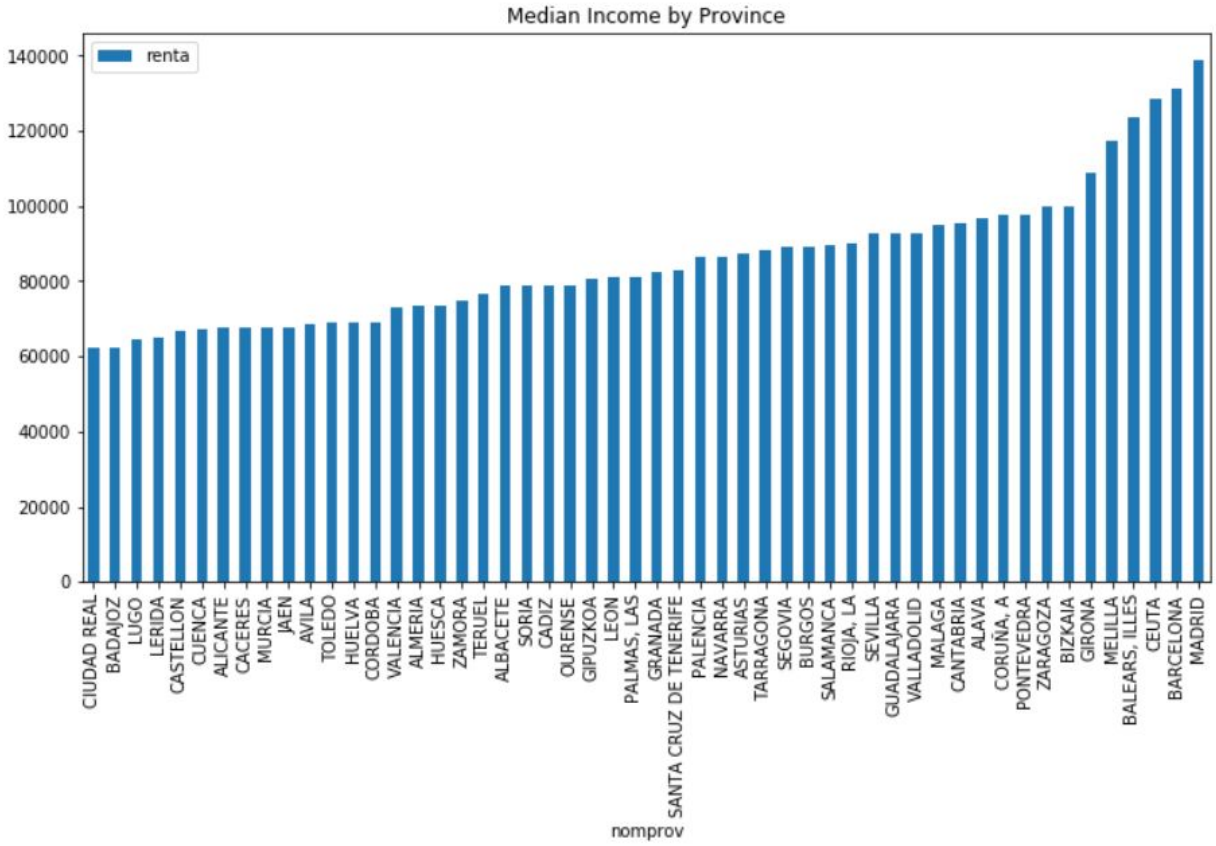
You can see that in June 2015 there was a large jump in the number of customers and increased increased gradually from then. Upon further review there

were a small amount of customers that dropped out over time, though for most part most customer remained in the dataset after entering it and actually there are data points for all months for most of the customers in the data set. Now for analyzing some traits of the customer population.



Looking at the age distribution there does seem to be some outliers with a handful of ages above 150 and probably a few too many young ages for this bank data. Also as seen in the histogram it appears to follow a bimodal distribution around the early 20's and mind 40's.

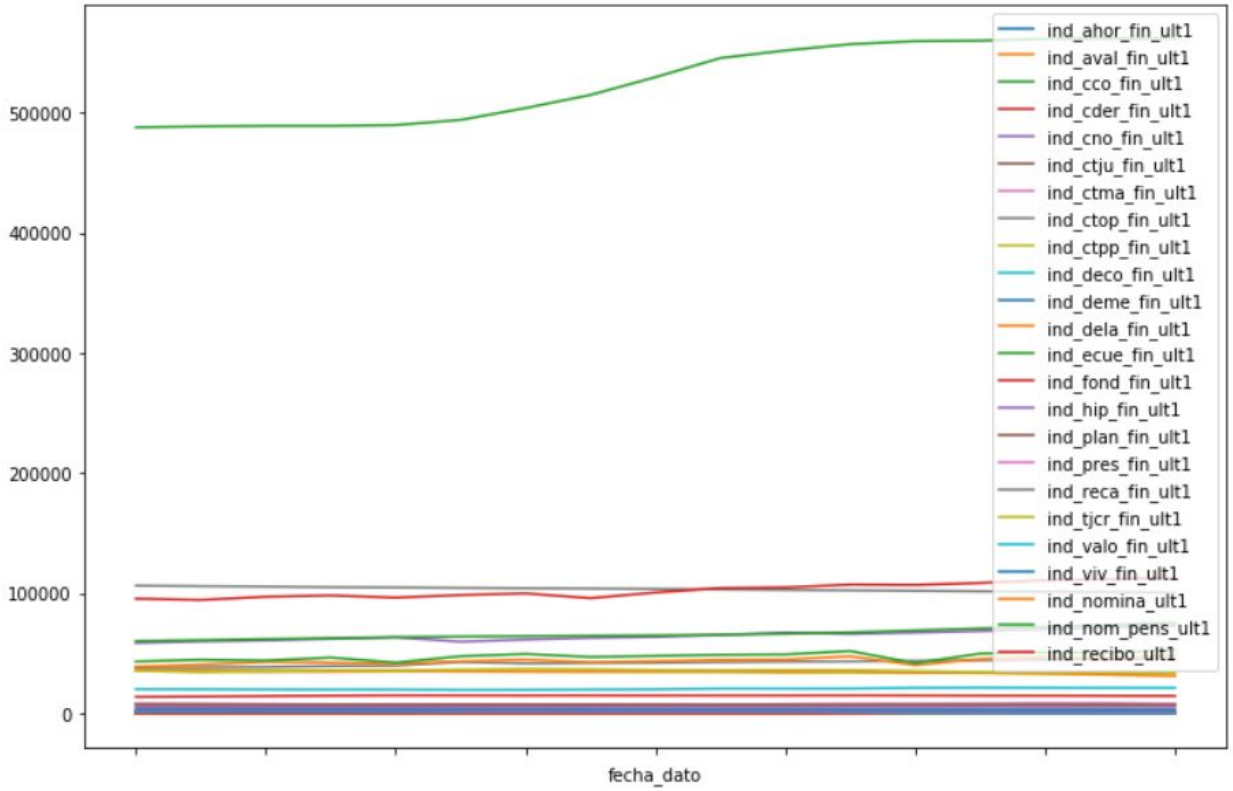
The plot shows the median gross household income by province. This is relevant since I used the median income of the province to impute missing income for customers in that province. You can see Madrid and Barcelona have the highest median incomes.



One of the main things I had to grapple with in this dataset is how to identify when a customer bought a new product. It sounds simple enough, but the way the data was arranged actually made this pretty difficult. Each data row only said whether a given customer owned a given product in a given month, so to find out whether that product was purchased in that month I would have to reference the entry for that customer in the prior month. Combined with the fact there are between 6-9 million customers arranged in no particular order for each of 17 different months with 24 possible products they may have, hopefully you will see that identifying when customers bought a product required a careful approach.

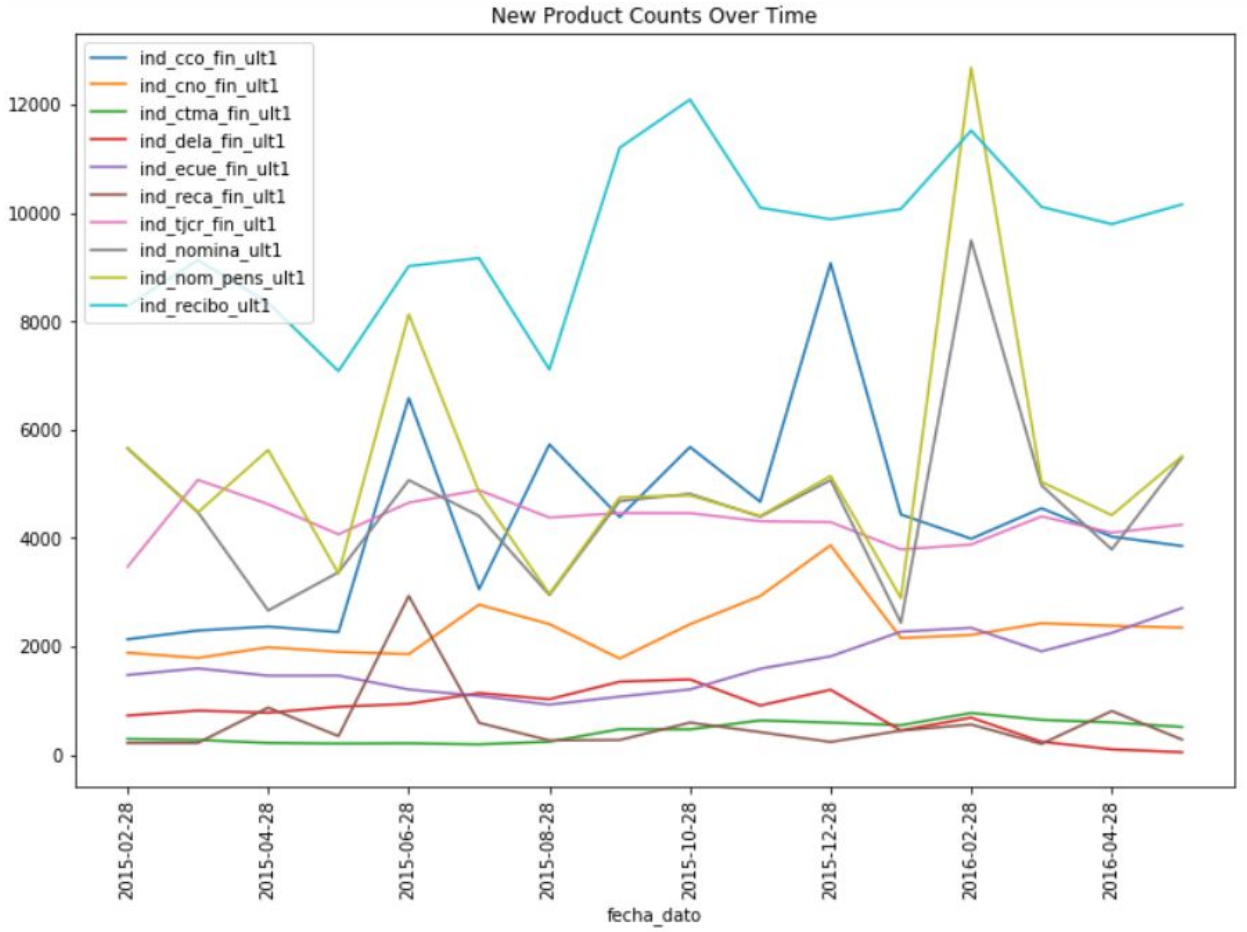
For example one of the first things I wanted to plot was the counts of new products over time so I grouped the products by date and aggregate by sum and quickly

got this plot:



However I realized that this just showed me the total count of each product for each month in the dataset and did nothing to differentiate product that were new for that customer that month.

Later I used a loop to iterate through sequential pairs of months, select only customers that were in both months and subtract the prior months' product values from the current month. Then I filtered out the all the zero and negative one's to get just rows representing a customer buying a new product. This allowed me to create the following plot where I just used the top ten products so that all lines will be distinguishable.



So now I have a pretty good understanding of what my data looks like and have taken steps to clean the data. I am now at good stepping off point to begin feature engineering and model testing and tuning.