For this project, I built two collaborative filtering models that recommend banking products to current customers of Santander Bank. The first model estimates the preferences of the customer based on the other products that the customer currently uses, while the second model estimates their preferences based on the customer's personal attributes (e.g. demographic data that has been made accessible). These models can be beneficial towards the bank and its customers; the bank is better able to understand what its customers want and is thus able to market to them more effectively, while the customer can get a better idea of what products they would benefit from adding. Ultimately, I determined that though both models are effective, for Santander and its customers, a personal-attributes based collaborative filtering model is able to more accurately recommend products than a prior-acquisitions based model.

1. Data

The dataset I used was found on Kaggle.com, and was shared by the official Banco Santander account. Here is a link to the Kaggle page:
https://www.kaggle.com/c/santander-product-recommendation

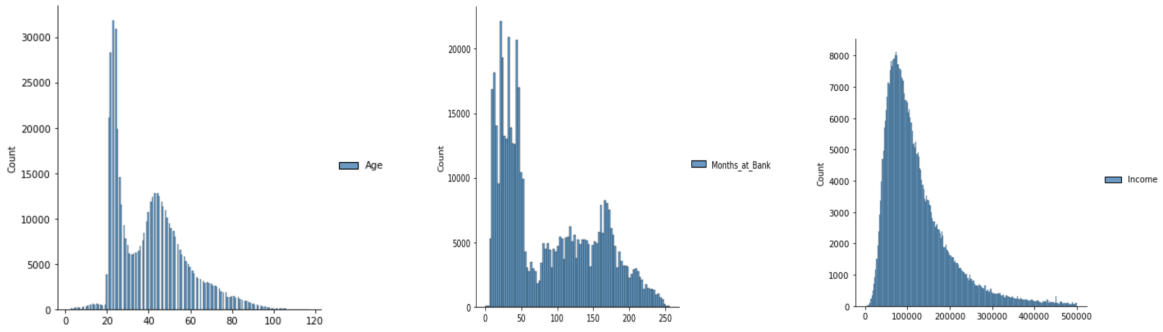
2. Data Cleaning

This project was coded in the Apache Spark language. After using spark.read.option to load the data as a dataframe, I inspected the schema. The data consisted of 48 columns, roughly half of which were string type, while the other half were numeric type. All of the columns were in Spanish, so for clarity's sake, I translated the columns

to English and renamed them. Next, I inspected the columns for NANs and NAs. Most of the columns ended up having some. For the numeric columns with missing values, I imputed the median. For the categorical columns with missing values, I imputed the most frequently occurring category.

Each individual customer had up to 17 observations, which coincided with the 17 months that the dataset measures. We do not want customers with less than 17 observations because we want to look at the same, full range of months for every customer to best optimize our collaborative filtering system. Luckily, there were over 600,000 customers with a full range of data. Because of this, I decided to drop customers who did not have 17 observations. I ended up dropping most of the non-product columns from the dataset because it seemed like they would be of low predictive value. I also feature engineered three new columns by binning numeric variables: Age_Range, Months_Range and Income_Range.

AGE **MONTHS_AT_BANK** **INCOME**



To help me determine what ranges for the bins would be appropriate, In the end, I had whittled the dataset down to 32 columns.

3. EDA

I inspected the 24 product columns to see if there were strong correlations between them and any of the three binary variables (gender, active, and foreigner_index). A few products had strong correlations with the 'active' variable, but none had a strong correlation with gender or foreigner_index. Here's an example:

```
Correlation to Payroll_Account for  Gender -0.030316400642578074
Correlation to Payroll_Account for  Foreigner_Index -0.00517684620680248
Correlation to Payroll_Account for  Active 0.3030775727099324
Correlation to Payroll_Account for  Payroll_Account 1.0
```

I wasn't surprised that gender didn't correlate with anything, but I was surprised that foreigner_index didn't. I also noticed that there were 4 products that fewer than .1% of customers used, so I decided to drop those columns.

4. Rating Development

To develop customer attribute based data, I experimented with four different types of models: Linear Regression, Support Vector Machine, Random Forest and Logistic Regression. I preprocessed the data by running OneHotEncoder on the non-binary variables, and used VectorAssembler to combine the encoded features with the non-encoded features and generate one feature column. Next, I split the data using a 60/40 training/test ratio. After fitting a linear regression model to the Current_Accounts column, it became clear to me that it was not the correct model

type; the r squared score for the column was 0, meaning that the model didn't explain any of the variance. I evaluated the three classification models based on their 'areaUnderROC' score. The logistic regression model performed best under this metric, possibly because many of the feature inputs were correlated with each other. I created 20 logistic regression models (one for each product), fit them to their respective columns, made predictions on the test data, and stored the predictions in a new dataframe to use as implicit ratings for my customer attribute based ALS model.

5. ALS Models

I created two different rating systems to evaluate the bank customers' preferences. The first rating system estimates the preferences of the customer based on the other products that the customer currently uses, while the second rating system estimates their preferences based on the customer's personal attributes. Both rating systems were used for the ratingCol in ALS models. The best ALS model for the first rating system had an RMSE of .25249, while the best ALS model for the second rating system had an RMSE of .19667. Thus, I theorized that although both models were effective, a personal-attributes based collaborative filtering model would be able to more accurately recommend products for Santander and its customers than a prior-acquisitions based model would.

6. Recommendations

The customer attributes/demographics based model had a superior performance compared to the prior product acquisition-based model, but both approaches worked

well, and neither should be disregarded moving forward. Certain attributes, such as gender and country residence should not be considered when making product recommendations for Santander Bank. The models can be used to make recommendations for individual customers, and Santander can take those recommendations and promote them to customers via targeted ads or email.

7. Further Research

There are a few next steps that could be taken in the domain of banking product recommendations. We could start by incorporating PCA on the features instead of whittling them down manually. Instead of using logistic regression as a classifier, a neural network could be incorporated. Additionally, when doing further analysis, it would be helpful to know more about the product types; no product information was provided via the Kaggle dataset aside from the product names. It would also be interesting to work with data from other banks, aside from Santander, to see how the performance of collaborative filtering models would compare.