Collaborative Filtering For Banking Products

Should we base our recommendations on customer attributes, or product acquisition history?

Santander Group

Santander

- $1.4 Trillion in assets under management
- More than 100 million customers
- Offer a large variety of products and services
- Possess customer demographic data and product acquisition history data
- How can Santander utilize their data to market additional products to their customers?

The Dataset

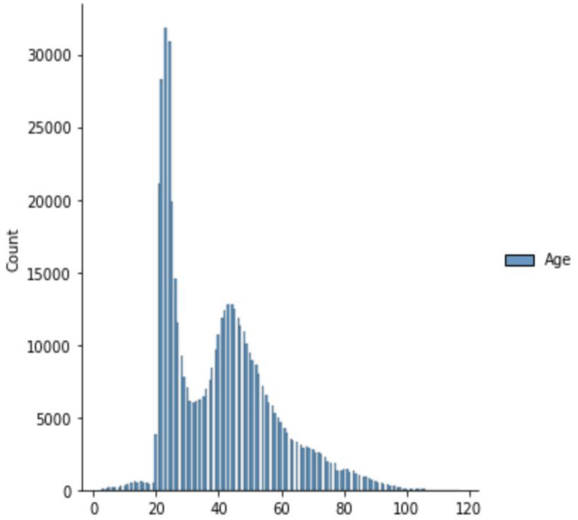
- 48 columns, more than 13 million rows
- 24 String type, 1 Double type, 23 Integer type
- 14 columns with Null/NA values
- Column names in Spanish
- 24 product columns; all product entries are binary id variables (1=customer has product, 0= customer doesn't)

Data Cleaning

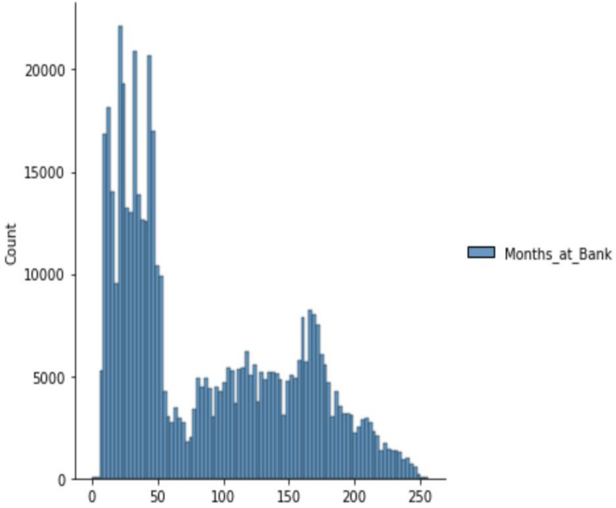
- Only wanted customers with a full range of data, needed 17 timestamps
- Dropped columns that did not offer predictive value
- Dropped columns with little variation amongst outcomes
- Imputed the most frequently occurring category for categorical columns with missing values
- Imputed the median for numeric columns with missing values

Feature Binning

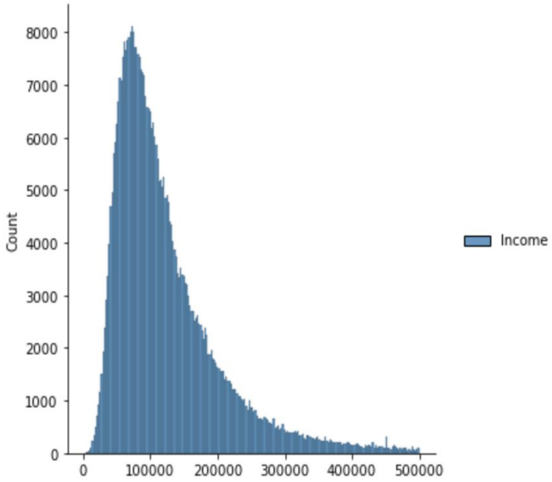
AGE



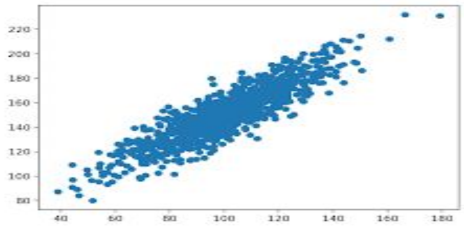
MONTHS AT BANK



INCOME



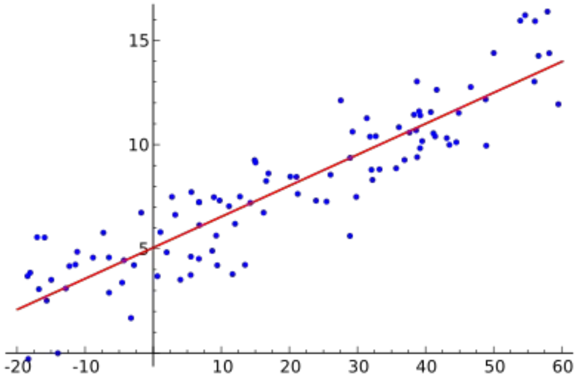
Assessing Correlations




```
Correlation to Payroll_Account for  Gender -0.030316400642578074
Correlation to Payroll_Account for  Foreigner_Index -0.0051768462068024 8
Correlation to Payroll_Account for  Active 0.303077572709932 4
Correlation to Payroll_Account for  Payroll_Account 1.0
```

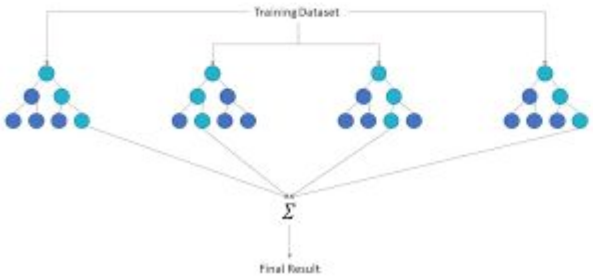
- Correlation for Gender and Foreigner_Index are weak!
- Correlation for Active is strong!
- Upon inspection, it was apparent that Gender and Foreigner_Index were not strongly correlated with any of the other products, while Active had a strong correlation with 8 of them.
- I also dropped 4 product columns that had a miniscule amount of customers

Linear Regression



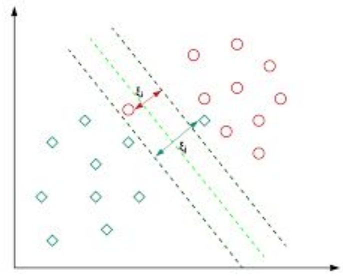
- End up with an R^2 that's close to 0
- High mean squared error
- Need to assume linearity, homoscedasticity, independence and normality
- Harder to interpret than a classification model for rating creation

Random Forest



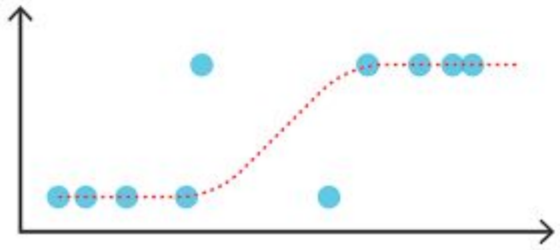
- Significantly lower AUC score than SVM and Logistic Regression
- Less useful for this dataset because there is no need for further dimensionality reduction (have a small amount of input columns)
- Also less useful for this dataset than others because we have such a high volume of data and are less worried about overfitting than we may otherwise be
- Is limited because the numeric features have been pre-binned; the model doesn't get as much control over cutoff points

Support Vector Machine



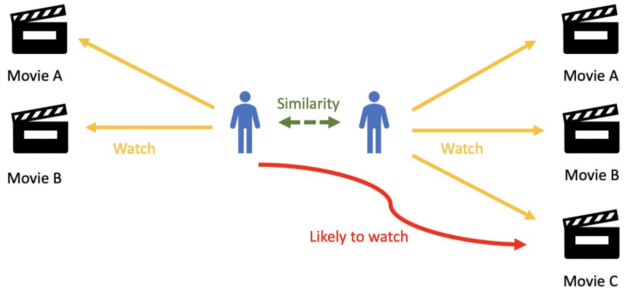
- Slightly lower AUC score than the Logistic Regression models, by a margin of roughly .1
- Not a great fit for datasets with a large number of entries
- More useful for datasets with high dimensionality/a low ratio of columns to rows
- Unlike the other models tried, SVMs can work with non-linear data; this dataset is linear though, so that isn't particularly useful

Logistic Regression



- Best performing model, frequently had an AUC score of over .9
- Fast computation speed; important because there were 20 different products to model
- High bias, low variance model, helps avoid overfitting
- Works well with simple/few features; only 4-5 feature components are being used for the model
- Works well with large samples

Collaborative FIltering



- Make recommendations for a customer based on the preferences of other customers
- If a customer A has similar preferences to customer B, they are more likely to acquire the same products that customer B has than that of another random customer
- Requires precisely 3 input columns: userCol, itemCol and ratingCol
- Spark uses the Alternating Least Squares method to estimate latent factors; can set the ‘rank’ parameter, which represents the number of latent factors used

Comparison of Models - Product History

With product acquisition history data as the input for the Rating Column, the model returns a root mean squared error of .2525

```
alsProduct = ALS(rank=5, maxIter=5, regParam=0.01, userCol="Customer_Code", itemCol="Item", ratingCol="Rating",
         coldStartStrategy="drop")
```

```
rmseProduct = evaluator.evaluate(predictions)
print("Root-mean-square error for Rank 5 = " + str(rmseProduct))
```

```
Root-mean-square error for Rank 5 = 0.2524881406707672
```

Comparison of Models - Customer Attributes

With customer attribute/demographic data as the input for the Rating Column, the model returns a root mean squared error of .1967

```
alsDemog = ALS(rank=10, maxIter=5, regParam=0.01, userCol="Customer_Code", itemCol="Item", ratingCol="Rating",
          coldStartStrategy="drop")
evaluator = RegressionEvaluator(metricName="rmse", labelCol="Rating",
                                predictionCol="prediction")
```

```
rmseDemog = evaluator.evaluate(predictions)
print("Root-mean-square error for Rank 10 = " + str(rmseDemog))
```

```
Root-mean-square error for Rank 10 = 0.19667073659587125
```

Using the models to make recommendations

```
userRecs.loc[userRecs['Customer_Code'] == 18498].iloc[0]['recommendations']
```

```
[Row(Item=1, rating=0.9943267107009888),
 Row(Item=5, rating=0.0),
 Row(Item=20, rating=0.0),
 Row(Item=0, rating=0.0),
 Row(Item=13, rating=0.0),
 Row(Item=2, rating=0.0),
 Row(Item=12, rating=0.0),
 Row(Item=11, rating=0.0),
 Row(Item=14, rating=0.0),
 Row(Item=15, rating=0.0)]
```

Model confidently recommends a Current Account to customer 18498, also confidently recommends that this customer not acquire any additional products

Product Key

1 = Current_Accounts	11 = Mortgage
2 = Payroll_Account	12 = Pensions
3 = Junior_Account	13 = Loans
4 = More_Particular_Account	14 = Taxes
5 = Particular_Account	15 = Credit_Card
6 = Particular_Plus_Account	16 = Securities
7 = Medium_Term_Deposits	17 = Home_Account
8 = Long_Term_Deposits	18 = Payroll
9 = e-Account	19 = Pensions_two
10 = Funds	20 = Direct_Debit

Conclusions

- 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
- Gender and Country Residence should not be considered when making product recommendations
- Age, Income and Months at Bank end up being good predictors, and they are all positively correlated with each other
- Rank did not have much impact on either of the collaborative filtering models' performance

Ideas for Further Research

- Unsupervised learning; could use PCA on a broader number of features as inputs for the logistic regression model or others
- Could build a neural network based classification model
- Would benefit from knowing more about each type of product, some of the column titles are not sufficiently informative
- Compare with data from other banks