Document Python Group Project

1) Data Preparation

a. Account Table:

We renamed the values in “frequency” column in English and the ‘district_id’ to avoid a conflict at the merging step.

We also extracted the years, months and days of the date to make it easier for analysis and extraction. Then we created a variable to be able to determine since how many days the account exists.

b. Card Table:

We dropped the ‘issued’ column because in our opinion, it will not be that useful to determine wheter a customer is good or not. Then we created a variable to be able to determine the days since the issuance of the card

c. Client table:

As the previous table, we created variable to get the year, month and days of birth of clients and drop the birth_number, which is irrelevant in our case

d. Order Table:

In this table, we just filled the missing values in ‘k_symbol’ column with ‘nocharacterization’ to expose the fact that there is no appropriate label for these debits orders. On top of that, we also replaced the values of ‘k_symbol’ in order for more explicit words.

e. Loan Table:

We replaced the values of ‘status’ in “ok” or “not payed” to make it more explicit in a reporting environment.

f. Demographic Table:

In this table, we just renamed the columns to make it more appropriate and explicit.

g. Transaction Table:

First, we checked the data by just printing the head to get a first overview of the data. Then we started by checking the missing values. It appears that there are 183114 missing values in the “operation” variable (which is more than the 5% or the whole data set so we cannot just delete these rows).

Then we dig a little deeper into the documentation of the data. It appears that some values where not that obvious, indeed, the language was in polish. So, we choose to create some dictionary to make it more appropriate in an international finance context (i.e dicttype, dictop, dictksymb in the jupyter notebook). Hence, the final datamart would be more understandable for most of the people.

In addition, we figured out by checking the values within each variable that 'VYBER', which should be present in operation, was present in the 'type' column. Therefore, we made the appropriate change to replace it with the appropriate and most logical value: withdrawal.

Once the values replacement idone, we came back to the missing values. Before doing any operation on these variables, we choose to mark, by creating a new variable, where values were missing.

K_symbol processing: TBD after the merge, it could be possible to replace it by the correct variables, and not only the NA.

According to the "bank" and "account" variable, they both are correlated (bank represents the name of the partner account).

We deduced that if there is no partner, then there is obviously no bank. At the end there is 21881 "banks" missing, in other words, there is a partner but none of these banks were identified. So, we choosed to flag the "normal" nan by "ZZ" and let the actual missing "bank" values as nan.

i. Function:

The relevant functions related to the transactions:

- NumbertransactionAccount: count the number of transactions for a specific account ID
- MaxAmountAccount: Give the highest transaction amount for a specific account ID
- MinAmountAccount: Give the lowest transaction amount for a specific account ID
- AvgAmountAccount: Give the average transaction amount for a specific account ID
- LatestTransaction: return the date of the latest transaction for a specific account ID
- OldestTransaction: return the date of the oldest transaction for a specific account ID
- CommonOp: return all kind of operation and their count for a specific account ID
- LatestBalance: return the latest balance for a specific account ID
- AverageBalance: return the average balance for a specific account ID

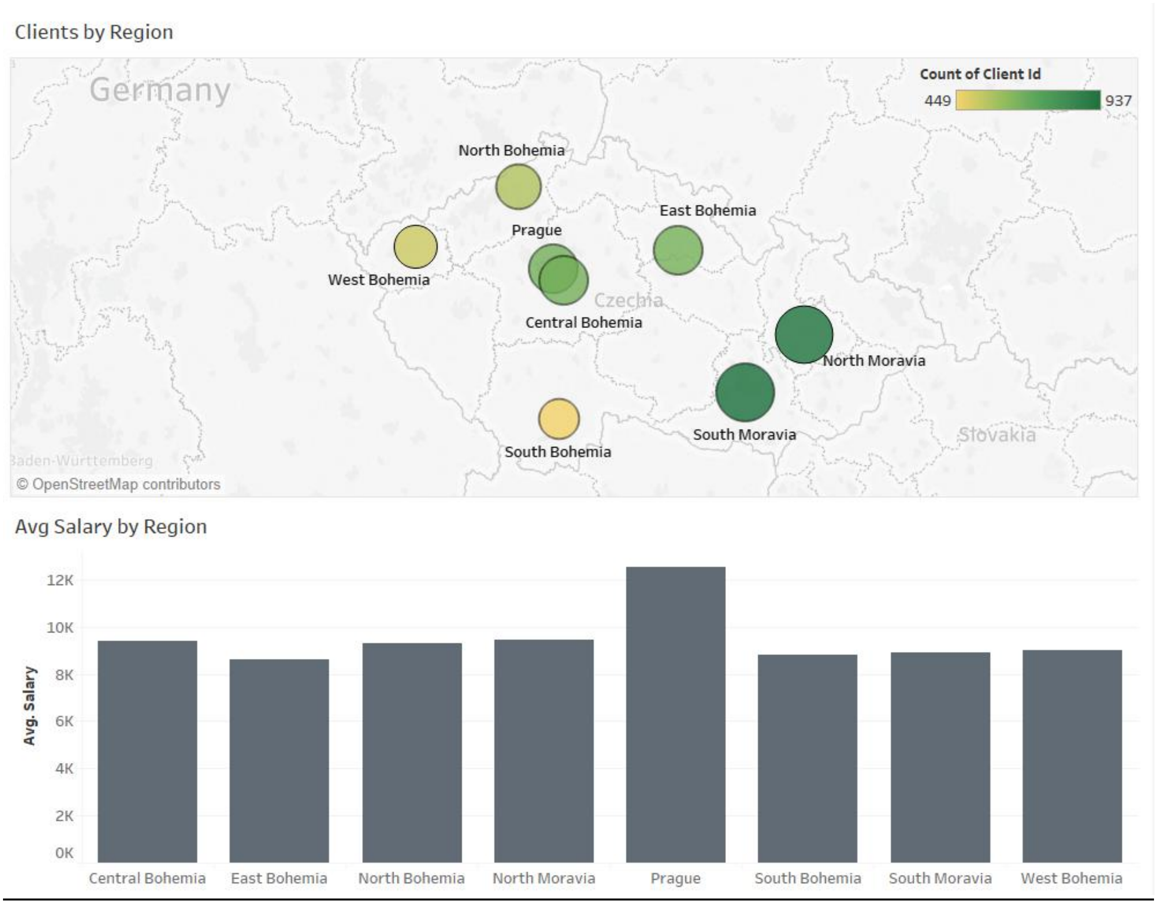
ii. Variable for transaction:

We figured out those applying functions to the transaction table will take a lot of time. So we tried through another method which is combining groupby statement and agg statement. This way, we were able to create the previous variables without using the function (I.e see the jupyter notebook for the details).

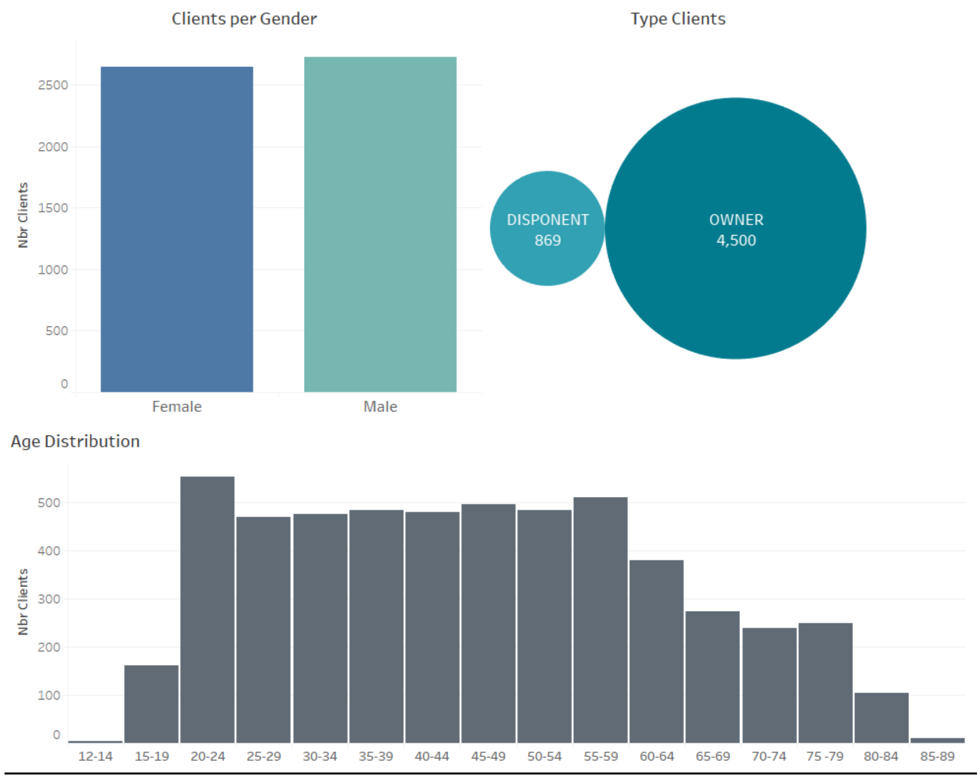
2) Data output and interpretation

Demographic analysis:

Clients are located in eight different regions of Czech. The regions South Moravia and North Moravia are the ones with the highest concentration. In addition, we can observe that the region with the highest average salary is Prague; the other regions have a slight difference in it. It could be a good strategy for the bank increasing the product and customer acquisition in this region.



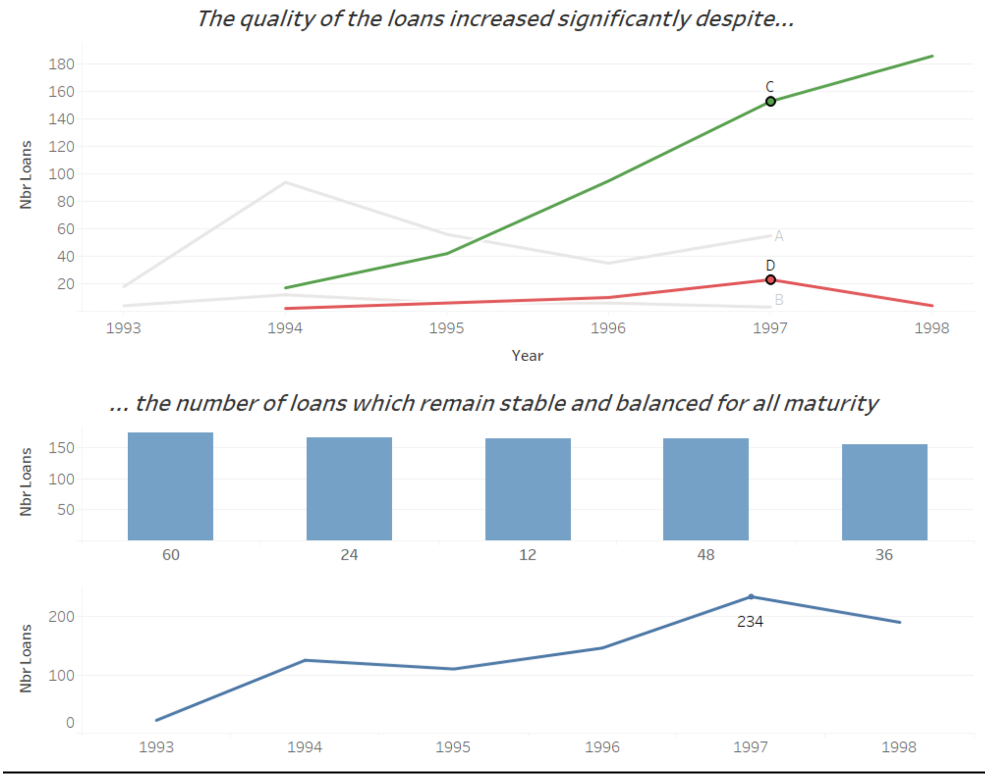
Regarding the gender and age distribution, the clients present a similar division by gender and they are concentrated mainly between the ages of 25 and 59 years. The principal type of clients of the bank are owners since only 869 clients are using the account of others.


Clients per Gender

Type Clients

Age Distribution

Loan analysis:


The quality of the loans increased significantly despite…

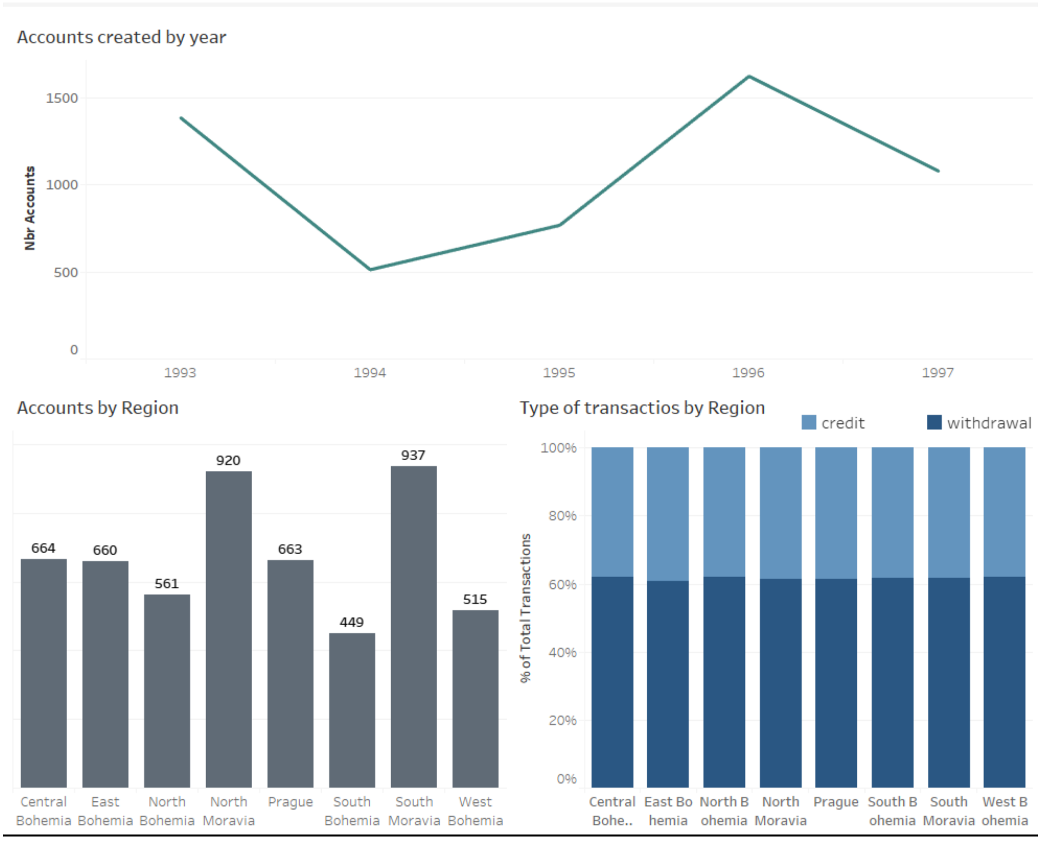
… the number of loans which remain stable and balanced for all maturity

As per the previous graph, we can say assume that the number of loans significantly increased between 1993 and 1997 (from less than 40K to more than 200k) and then slightly decreased in 1998. However, the quality of the loans increased a lot. Indeed, the number of loans in C category, which is “ok” for running contract, was only few times more than the category D that relate to client in debt. However, by 1998, the ratio is higher, about 20 times more quality loans than junk loans. It could be explain by improvements in the loans management or customer’s selection.

Account analysis:

The number of opened accounts had an increase in 1995 and 1996. In general, the bank does not show a stable behavior in this variable. By region, North Moravia and South Moravia are the regions with the highest number of accounts.

Clients show a difference between the type of transaction that they do in their accounts, the proportion of the number of transactions in withdrawal is higher than in cash.



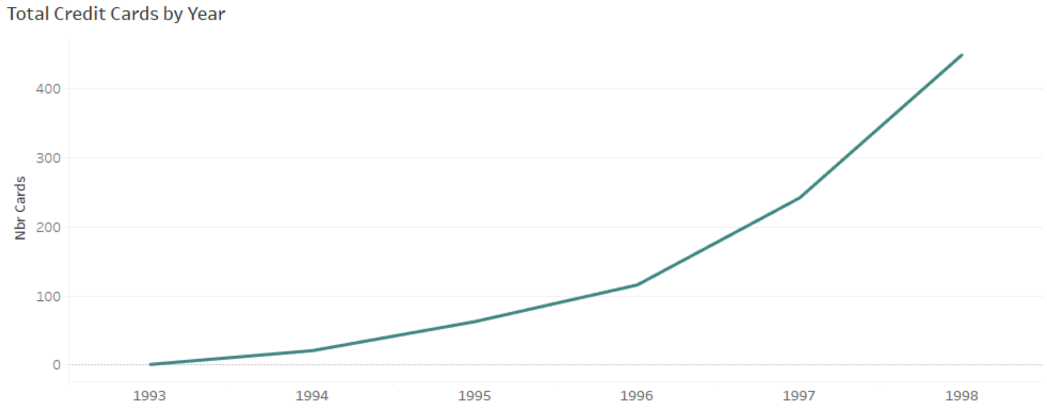
Accounts created by year

Accounts by Region

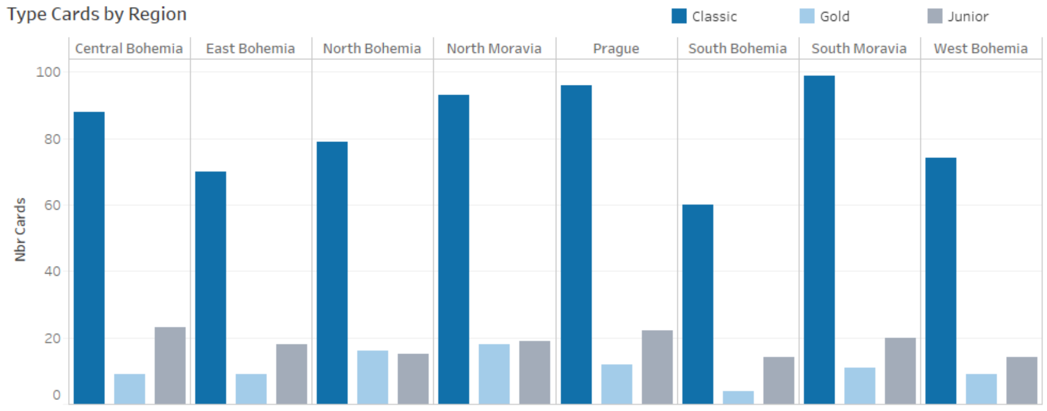
Type of transactios by Region

Credit Card:

The bank has increased the number of credit cards issued every year, by region we can observe that the most common type of card is the Classic.



Total Credit Cards by Year



Type Cards by Region

3) Intermediate Tables

account_cleaned

card_cleaned

client_cleaned

disp_cleaned

district_cleaned

loan_cleaned

order_cleaned

trans_cleaned

order_prep_cleaned

trans_agg_cleaned

maintable

4) Final Table Variables

'client_id',

'district_id',

'birth_year',

'birth_day',

'birth_month',

'gender',

'age',

'disp_id',

'account_id',

'type_x',

'card_id',

'type_y',

'date_issue_card',

'days_issue_card',

'district_id_acc',

'freq_description',

'date_creation_acc',

'year_acc',

'month_acc',

'day_acc',

'days_creation_acc',

'loan_id',

'amount_loan',

'duration_loan',

'payments_loan',

'status_loan',

'status_description',

'date_loan',

'district_name',

'region',

```
'nbr_inhabitants',
'nbr_muni<499_inh',
'nbr_muni_500-1999_inh',
'nbr_muni_2000-9999_inh',
'nbr_muni>10000_inh',
'nbr_cities',
'ratio_urban_inh',
'average_salary',
'unemploymant_rate_95',
'unemploymant_rate_96',
'nbr_entp_1000_inh',
'nbr_crimes_95',
'nbr_crimes_96',
'trans_amt_total',
'trans_amt_min',
'trans_amt_max',
'trans_number',
'date_trans_old',
'days_oldtrans',
'date_trans_late',
'days_latetrans',
'date_balance',
'latest_balance',
'avg_balance',
'household_amount',
'insurance_amount',
'leasing_amount',
'loan_amount',
'nochar_amount',
'nbr_orders',
'order_total_amount'
```