FINANCIAL PROGRAMMING

GROUP PROJECT – FINANCIAL BASE TABLE

LUCERO FABRIZIO

SHASHANK GOLLAPALLI

NANDINI GANTAYAT

OVERVIEW OF THE PROJECT

Creating a data science base table with the information extracted from financial dataset and to determine the customers who are eligible to grant loan and credit card.

PROBLEM STATEMENT

The bank wants to improve their services. For instance, the bank managers have only vague idea, who is a good client (whom to offer some additional services) and who is a bad client (whom to watch carefully to minimize the bank loses). Fortunately, the bank stores data about their clients, the accounts (transactions within several months), the loans already granted, the credit cards issued The bank managers hope to improve their understanding of customers and seek specific actions to improve services. A mere application of a discovery tool will not be convincing for them.

DATA EXPLORATION

Data was provided in the form of various tables like credit card, daily transactions, account, loan, demographics, disposition, orders, and client information to create the base table. There are 5369 unique clients' observations with 86 columns in our Base Table. This database used in this project was prepared by Petr Berka and Marta Sochorova.

RAW DATA

- The dataset provided by the bank contains the following tables:
- Account (4500 objects) - each record describes static characteristics of an account in the bank
- Client (5369 objects) - each record describes characteristics of the bank client Disposition (5369 objects) - each record relates client with an account
- Orders (6471 objects) - each record describes characteristics of a payment order
- Transaction (1056320 objects) - each record describes transaction in an account
- Loan (682 objects) - each record describes a loan granted for an account
- Credit Card (892 objects) - each record describes a credit card issued to an account
- Demographic (77 objects) - each record describes demographic characteristics of a district.

COLUMN_NAME	DATA TYPE
account_id	Int64
District_id_branch	Int64
Statement_frequency	Object
Account_creation_date	Object
Account_creation_year	Int64
Account_creation_month	Int64
Disp_id	Int64
Client_id	Int64
Disponents	Int64
District_code	Int64
District_name	object
region	object
Inhabitants	Int64
Municipalities_pop_lt_499	Int64
Municipalities_pop_lt_1999	Int64
Municipalities_pop_lt_9999	Int64
Municipalities_pop_lt_10000	Int64
Numb_cities	Int64
Ratio_urban_inhab	Float64
Avg_salary	Int64
Unemployment_rate__95	Float64
Unemployment_rate_96	Float64
Entr_pr_1k_inhab	Int64
Unemployment_rate__95_flag	bool
Numb_crimes_95	Float64
Numb_crimes_96	Int64
Numb_crimes_95_flag	bool
District_id_client	Int64
Birth_year	Int64
Birth_month	Int64
Birth_day	Int64
Gender	Object
Age	Int64
Age_group	Int64
Collection_from_another_bank	Float64
Credit_card_withdrawal	Float64
Credit_in_cash	Float64
Remittance_to_another_bank	Float64
Withdrawal_in_cash	Float64
Is_periodic	Int_32
Ksymbol_is_household payment	Uint8
Ksymbol_is_insurance payment	Uint8
Ksymbol_is_interest credited	Uint8
Ksymbol_is_loan payment	Uint8
Ksymbol_is_old age pension	Uint8
Last_balance	Uint8
Loan_amount	Float64
Loan_duration Monthly_payments	Float64
Loan_grant_month	Float64
Loan_status_1996	Object

Type_1996	Object
Card_issued_month	Float64
Loan granted 1997	Int32
Card issued 1997	Int32
LOR	Int64

DATA PRE PROCESSING AND CLEANING

CLIENTS TABLE

1. Each record of the dataset read describes the static characteristics of an account.
2. The “Age” and “Age Group” variables were calculated taking current year as 1997.
3. New columns “birthyear”, “birth month”, and “birthday” were extracted from “birth number”
4. New column “Gender” was extracted from “birth month.”
5. There were no missing values.
6. Dropped birth_number column as it was redundant

TRANSACTIONS TABLE

1. There were 2278837 missing values
2. Converting date into a string from int64
3. Using functions to convert transactions into English
4. Renamed 'account' to 'partner_account' for coherency
5. Dropped old columns

ACCOUNT TABLE

1. There are no missing values
2. Frequency column stands for frequency of issuance of statements
3. Date column is the date of creating of the account
4. Creating Year and Month columns for ease of calculation of LOR

DEMOGRAPHICS TABLE

1. Renamed columns for coherency
2. Replacing the '?' values in umeployment_rate_95, numb_crimes_95 with NaN and converting type to float and creating flags

CARD TABLE

1. There are no missing values
2. Extracted columns like card type, issued_card, issued_year_card, issued_month_card

DISPOSITION TABLE

1. Disp table is used to identify type of owner
2. No missing values

ORDER TABLE

1. Changed the names for all k_symbols
2. There were 1370 NaN values after running the KsymbToEng function initially that were substituted np.NaN with No order info

LOAN TABLE

1. There are no missing values
2. Creating Year and Month columns for ease of calculation and renaming

```
]: INITIAL_BASETABLE
```

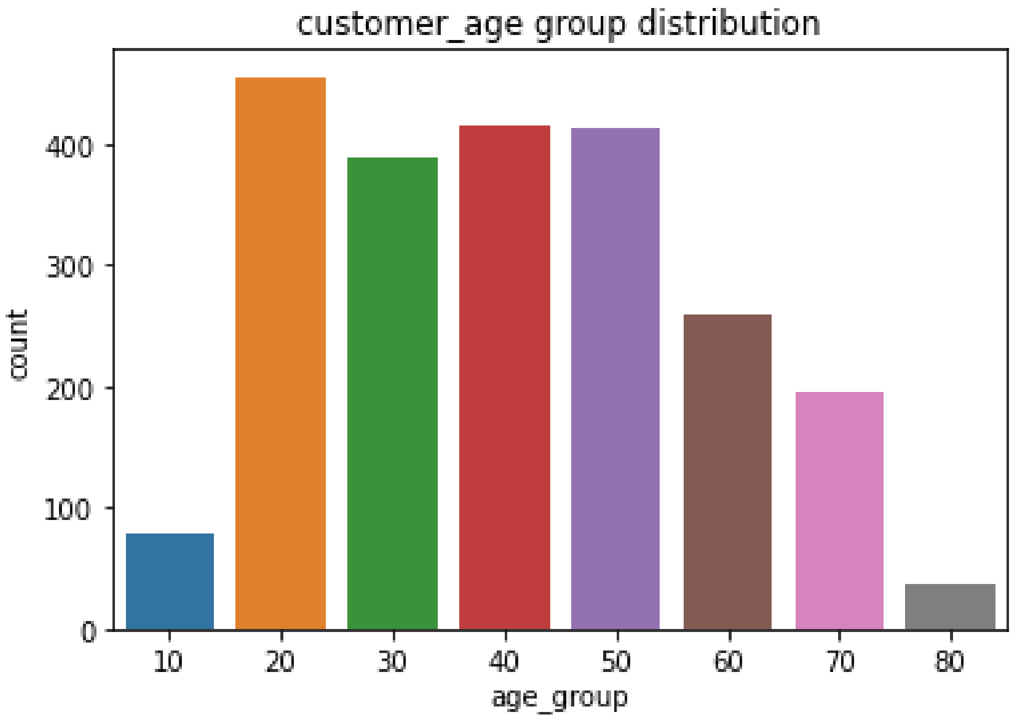
	account_id	district_id_branch	statement_frequency	account_creation_date	account_creation_Year	account_creation_Mont
0	576	55	Monthly Issuance	1993-01-01	1993	
1	3818	74	Monthly Issuance	1993-01-01	1993	
2	704	55	Monthly Issuance	1993-01-01	1993	
3	2378	16	Monthly Issuance	1993-01-01	1993	
4	2632	24	Monthly Issuance	1993-01-02	1993	
...	...	...	...	...	...	.
2234	4462	73	Weekly Issuance	1995-12-27	1995	3
2235	3814	74	Monthly Issuance	1995-12-27	1995	3
2236	2780	63	Monthly Issuance	1995-12-29	1995	3
2237	3273	74	Monthly Issuance	1995-12-29	1995	3
2238	3559	18	Monthly Issuance	1995-12-30	1995	3

2239 rows × 58 columns

Our final base table has 2239 rows and 58 variables.

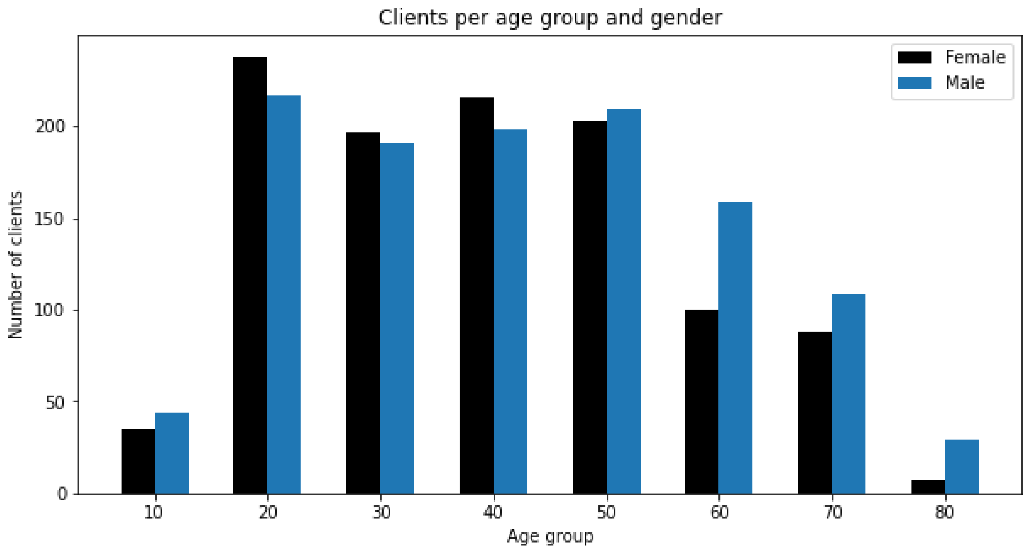
VISUALIZATION

1. Age distribution among customers:



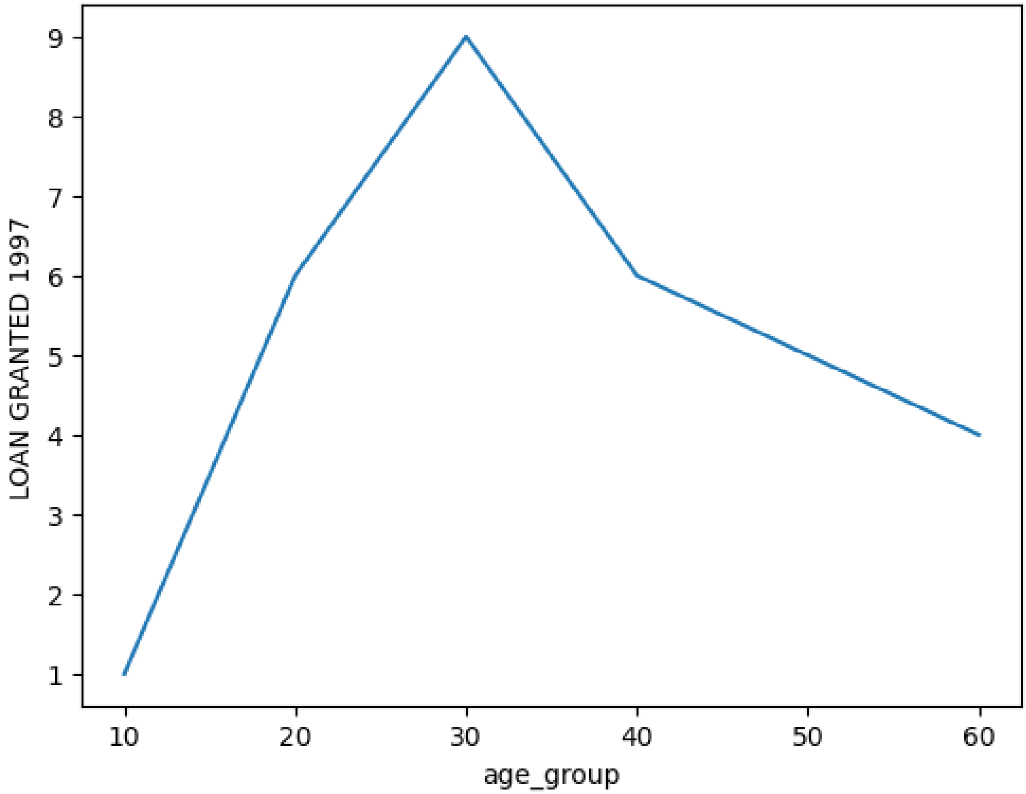
The distribution of the customers is the highest in the age group 20 and sees a gradual decline till the age group 80.

2. Gender distribution across age groups:



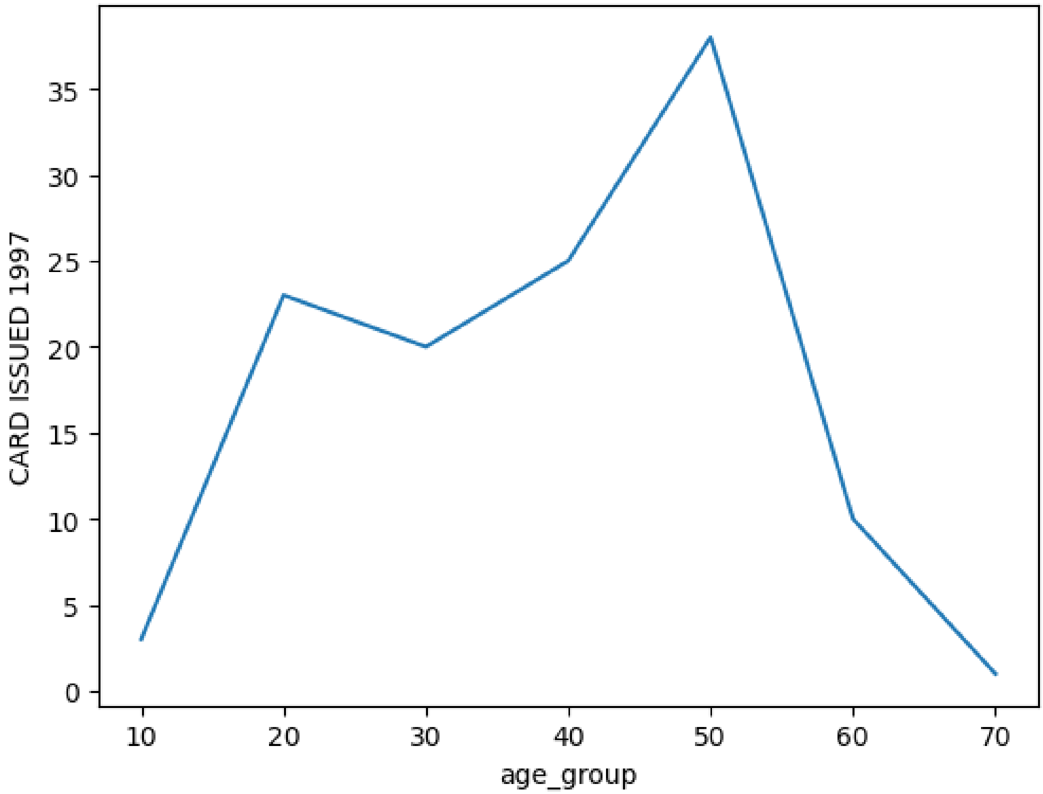
The number of females in the age groups: 20, 30 and 50 is higher than others.

3. Loans granted across age groups



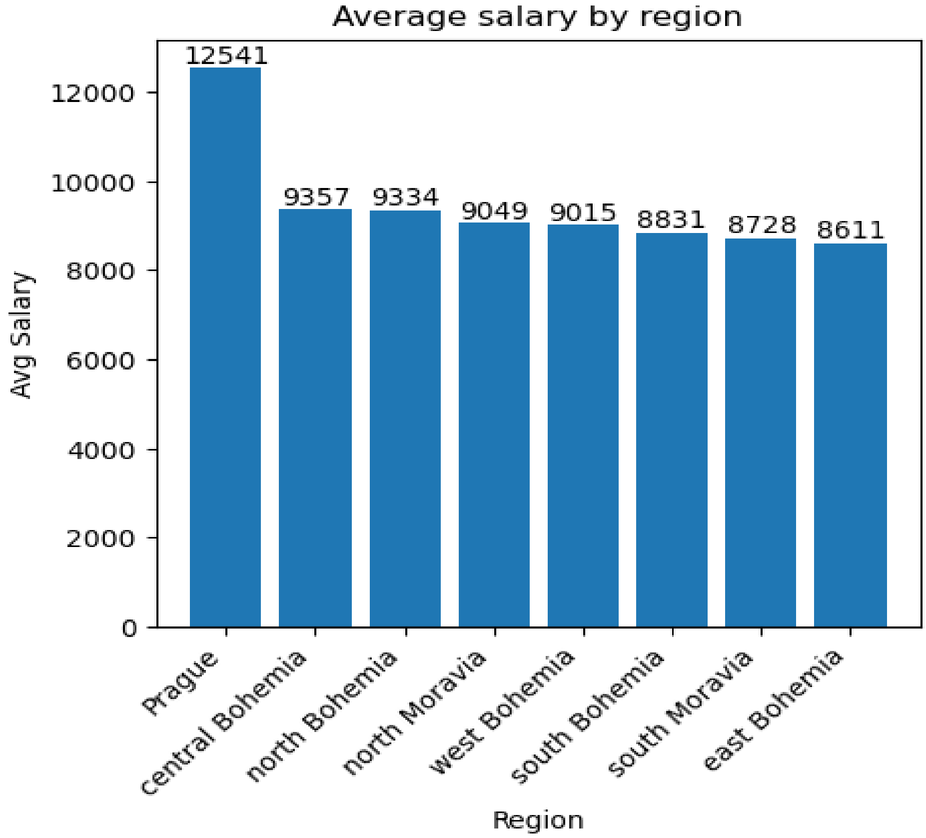
Highest number of loans were granted to age group 30

4. Cards issued across age groups



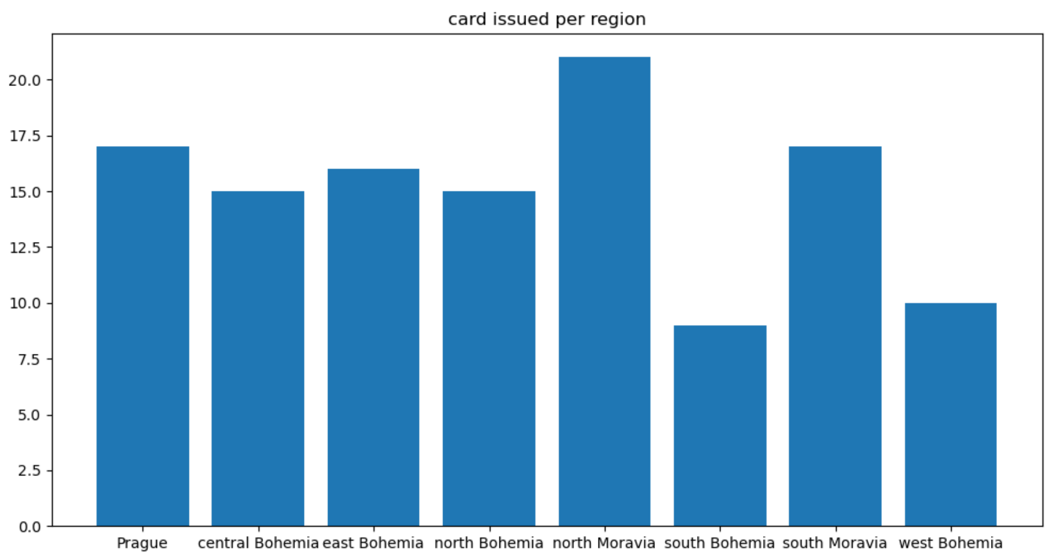
Highest number of cards were issued to age group 50

5. Average Salary by region



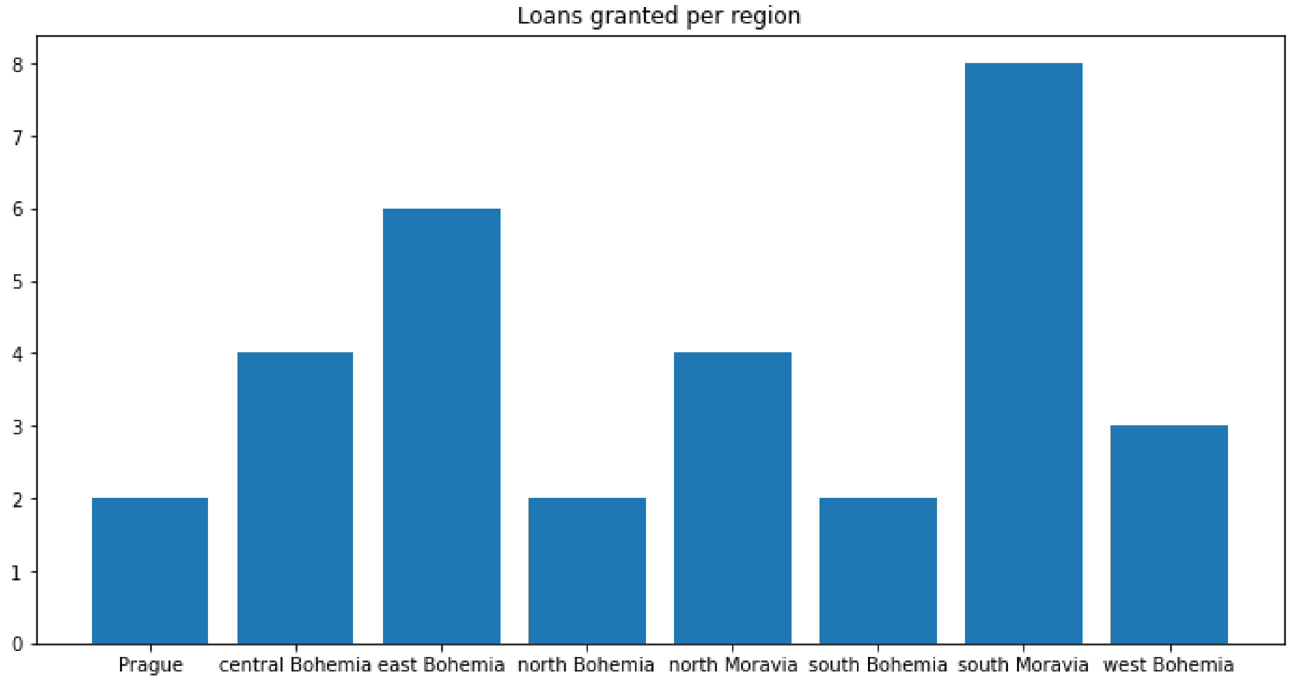
Prague has the highest average salary and East Bohemia the lowest average salary

6. Card Issued by region



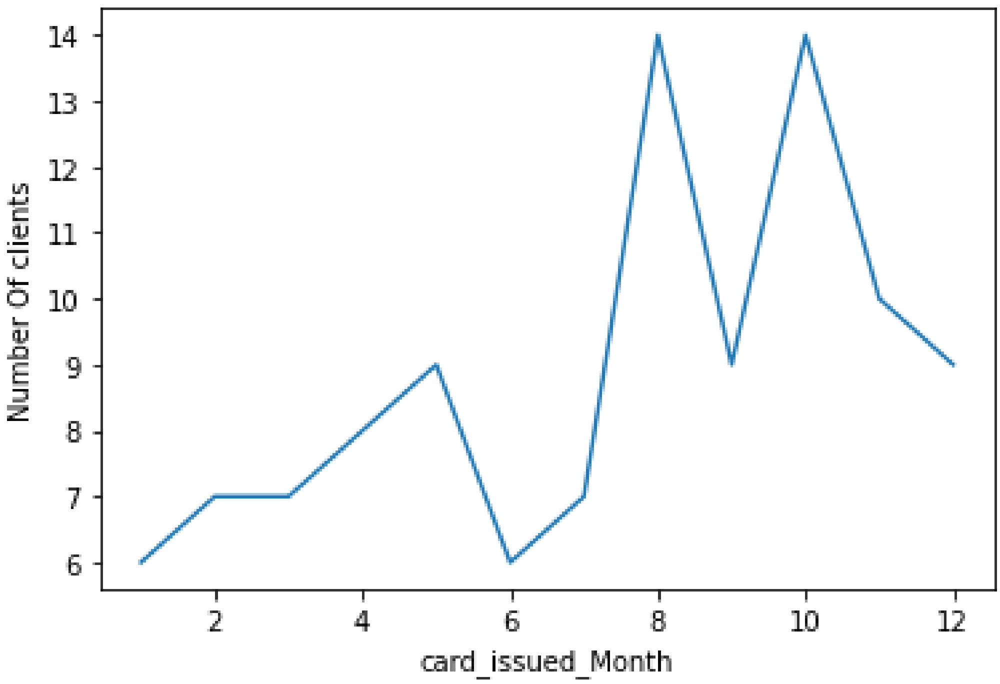
Clients in north Moravia were issued the highest number of cards, whereas clients in South Moravia were granted the highest number of loans.

7. Loans granted by region



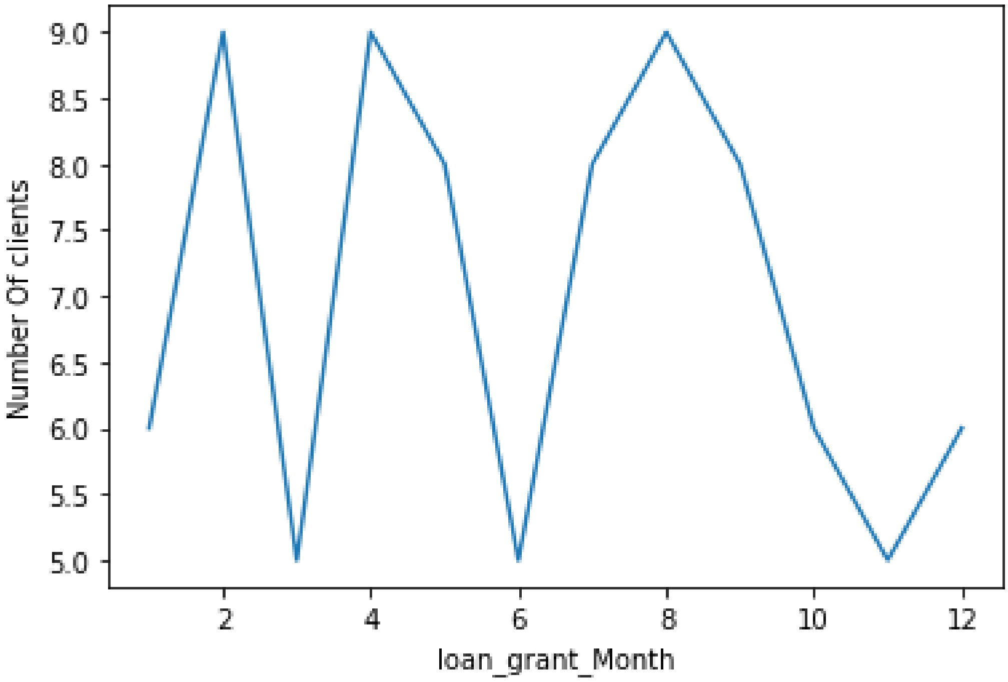
Clients in South Maravia were granted the highest number of loans

8. Distribution of cards issued over the year



Highest number of cards are issued in the months of August and November.

9. Distribution of loans granted over the year



Lowest number of loans were granted in March, June and November.

MODEL RESULTS

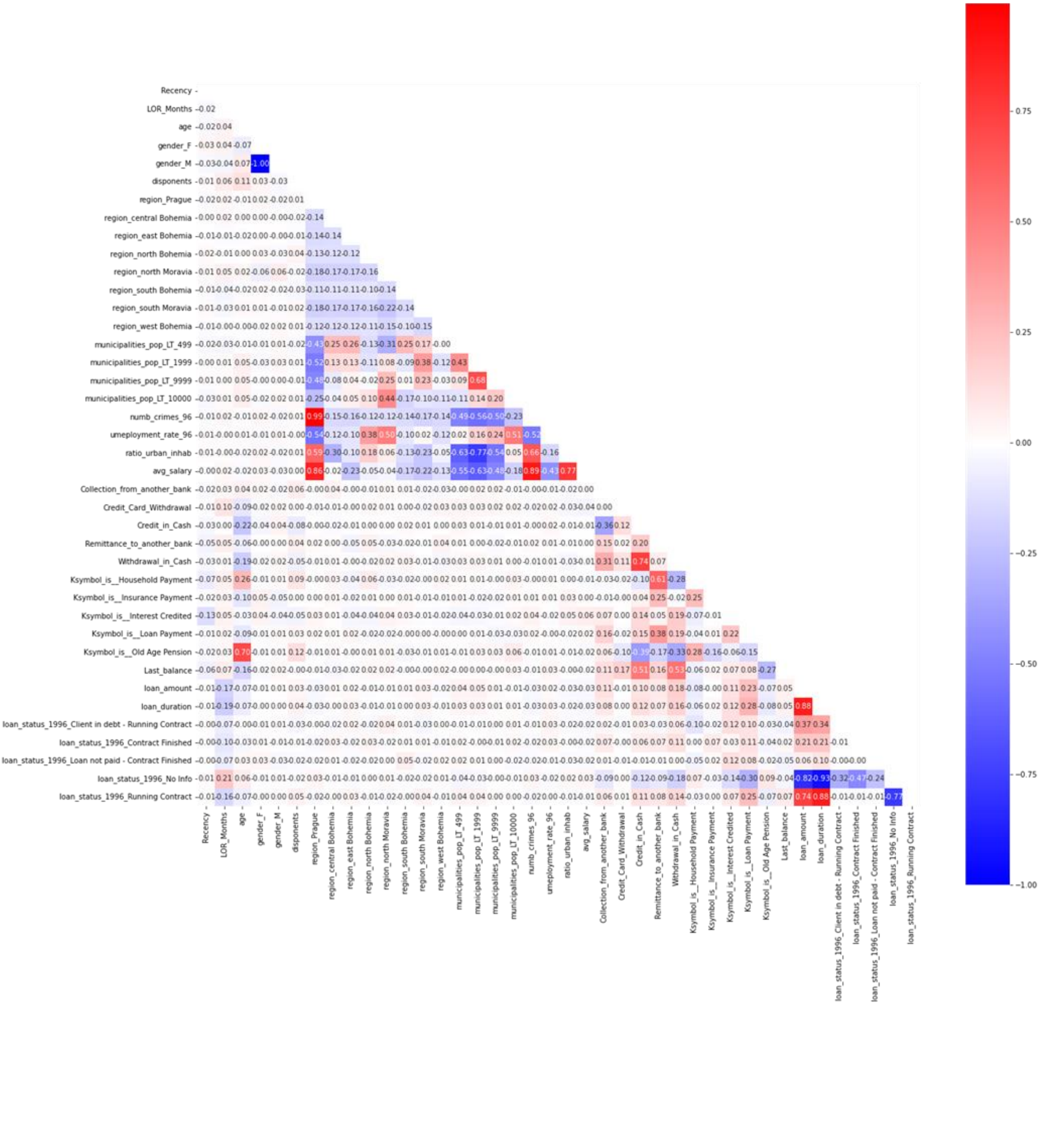
MODEL RESULTS FOR CARD ISSUED

	randomForest	boostedTree	XG
Accuracy	0.935268	0.926339	0.937500
AUC	0.751378	0.769813	0.753683

MODEL RESULTS FOR LOANS GRANTED

	randomForest	boostedTree	XG
Accuracy	0.984375	0.982143	0.984375
AUC	0.859572	0.869453	0.844833

VARIABLE CORRELATION



REFERENCES

[1] Passing_Networks. GRI Research – Passing Networks. Link: https://github.com/fabriziolufe/GRI-Research---Passing-Networks

[2] Pivotting dataframes in pandas. StackOverflow. Link: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.pivot_table.html

[3] Writing Markdown in Python. Earth Data Science. Link: https://www.earthdatascience.org/courses/intro-to-earth-data-science/file-formats/use-text-files/format-text-with-markdown-jupyter-notebook/

[4] Correlation Analysis. Seaborn. Link: https://seaborn.pydata.org/examples/many_pairwise_correlations.html